UNIVERSITY OF SOUTHAMPTON

FACULTY OF SOCIAL, HUMAN AND
MATHEMATICAL SCIENCES
Social Sciences

Designing Experiments on Networks

by

Vasiliki Koutra

Thesis for the degree of Doctor of Philosophy
September 2017

UNIVERSITY OF SOUTHAMPTON

ABSTRACT

FACULTY OF SOCIAL, HUMAN AND MATHEMATICAL SCIENCES

Social Sciences

Doctor of Philosophy

DESIGNING EXPERIMENTS ON NETWORKS
by Vasiliki Koutra

Designing experiments on networks challenges an assumption common in classical experimental designs, which is that the response observed on a unit is unaffected by treatments applied to other units. This assumption is referred to as 'non-interference'. This thesis aims at improving the design efficiency and validity of networked experiments by relaxing the non-interference assumption, where efficiency stands for low variance of the estimated quantities (precision) and validity for unbiased quantities (accuracy). We develop flexible and effective methods for designing experiments on networks (with a special focus on social networks) by combining the well-established methodology of optimal design theory with the most relevant features of network theory. We provide evidence that conventional designs such as randomised designs are inefficient compared to a systematic approach that accounts for the connectivity structure that underlies the experimental units.

We investigate the impact of the network structure on the efficiency and validity of the experimental design. There is evidence that the experimental design is determined by the small-scale properties of networks. We also develop an algorithmic approach for finding efficient designs by utilising the network symmetry as defined by the automorphism group of the underlying graph. This approach reduces considerably the search time for finding a good design in moderate-sized networks. It works by decomposing the network into symmetric and asymmetric subgraphs and consequently decomposing the design problem into simpler problems on these subgraphs. Moreover, we suggest a framework for finding optimal block designs, while taking into account the interrelations of groups of units within a network. In doing so, the units are initially divided into blocks, using spectral clustering techniques and the concept of modularity, prior to assigning the treatments. We study how the structural properties of the network communities affect the optimal experimental design and its properties. We also make a transition from experiments on social networks to experiments in agriculture showing the diversity of applications this research can address. In particular, we obtain optimal designs with two blocking factors while handling different definitions of neighbour structures related to either the distance among plots or the farmer operations. Throughout this thesis, several optimal designs on networks are obtained using a simple exchange algorithm, which is implemented in the R programming language.

Contents

List of Figures iii

List of Tables vii

Declaration of Authorship ix

Acknowledgements xi

1 Introduction 1
- 1.1 Preliminaries 1
- 1.2 Experiments on networks 5
- 1.3 Related literature 8
  - 1.3.1 Causes of interference and design properties 9
  - 1.3.2 Adjusting for treatment interference 10
  - 1.3.3 Adjusting for response interference and spatial or temporal variation 12
  - 1.3.4 Recent work on spillover effects and modelling influence in networks 13
- 1.4 Contribution 16
- 1.5 Thesis outline 16

2 General framework 19
- 2.1 Preliminaries on optimal designs 19
- 2.2 Standard randomised designs 21
- 2.3 Search algorithms for optimal design 25
- 2.4 Basics of graph theory 28
- 2.5 Discussion 38

3 Optimal designs with network effects 39
- 3.1 Background: modelling interference 39
- 3.2 Designs with linear network effects model (LNM) 43
- 3.3 Some analytical results 48
- 3.4 Patterns of optimally allocated treatments 56
- 3.5 Design efficiency and bias 64
  - 3.5.1 Efficiencies of randomised designs 65
  - 3.5.2 Bias due to model misspecification 66

3.5.3 Misspecification of the network structure 69
3.6 Simple exchange algorithm 73
3.7 Discussion 75

4 Design and network symmetry 77
4.1 Preliminaries on graph automorphisms 77
4.2 Symmetry breaking and design 83
4.3 Algorithmic approach 91
4.4 Discussion 98

5 Optimal block designs with network effects 101
5.1 Spectral clustering and block definition 101
5.2 Clustering algorithm 103
5.3 Designs with network effects block model (NBM) 106
5.4 Comparison of optimal designs under different models 111
5.5 Misspecification of cross-blocking connections 133
5.6 Robustness due to misspecified blocking 135
5.7 Discussion 140

6 Optimal row-column designs with network effects 143
6.1 Introduction 143
6.2 Designs with row-column network effects block model (RCNBM) 145
6.3 Optimisation algorithm 148
6.4 Adjacency matrix: practical issues and extensions 149
6.5 Comparison of optimal designs 151
6.6 Discussion 159

7 Discussion and directions for future research 161
7.1 Summary 161
7.2 Current work: Autoregressive network effects model 163
7.2.1 Autoregressive network effects model (ANM) 164
7.2.2 Maximum likelihood estimation 165
7.2.3 Asymptotic variance of the maximum likelihood estimators . . . 166
7.3 Future work 169

Appendix 173
A Results for Chapter 3 173
B Results for Chapter 5 179
C Optimal designs from Chapter 6 184

Bibliography 195

List of Figures

2.1 A small social network and its adjacency matrix 31
2.2 Degree distribution of the network 32
2.3 Special types of graphs 34
2.4 Popular types of networks 35
2.5 Degree distribution of the random network 36
2.6 Transition from a ring to a random network via small-world network . . 36
2.7 Degree distribution of the small-world network 37
2.8 Degree distribution of the scale-free network 37

3.1 An example social network 47
3.2 Network of Figure 3.1 ($m = 2$) 47
3.3 Network of Figure 3.1 ($m = 3$) 48
3.4 Pairwise comparisons of the criterion values for all balanced designs . . 59
3.5 Optimal designs for Example 3.4.1 for ϕ_1 and ϕ_2 60
3.6 Pairs of units of different distances receiving the same or different treatments 61
3.7 Snapshots of three common types of networks 63
3.8 Snapshots of three common types of networks, optimal designs (ϕ_1) . . . 63
3.9 Snapshots of three common types of networks, optimal designs (ϕ_2) . . . 64
3.10 Boxplots of efficiencies calculated for all balanced designs ignoring network effects 65
3.11 Bias in treatment effects due to network effects 67
3.12 Bias for treatment effects when $\gamma_1 = \gamma_2$ 68
3.13 Degree distribution of the network of Figure 3.1 70
3.14 Design efficiencies for 100 simulated misspecified networks 73
3.15 Design efficiencies for 100 simulated misspecified networks (connected graph) 73

4.1 Graph automorphism corresponding to a relabelling 78
4.2 All six automorphisms of the star graph of Figure 4.1, $Aut(\mathcal{G})$ 80
4.3 (a) Original graph; (b) a graph automorphism 82
4.4 (a) Orbits are indicated by different colours; (b) quotient graph; (c) skeleton 82
4.5 (a) Partition P_1; (b) Partition P_2 83

4.6 Network of size $n = 32$. Different colours indicate different orbits, with white corresponding to the skeleton 85
4.7 Optimal designs ϕ_1 and ϕ_2 86
4.8 Optimal designs ϕ_1 and ϕ_2 for the asymmetric network skeleton 87
4.9 All optimal designs for ϕ_1 for the asymmetric network skeleton 87
4.10 Optimal design for ϕ_1 and ϕ_2 (given fixed allocation for the skeleton) . . 88
4.11 (a) Original graph; (b) Graph skeleton 89
4.12 Optimal designs for ϕ_1 and ϕ_2 (given fixed allocation for the skeleton) . 90
4.13 Optimal design for ϕ_1 and ϕ_2 (without restrictions) 90
4.14 Binary tree 91
4.15 PhD social network 93
4.16 Degree distribution of the PhD network 94
4.17 (a) Original graph; (b) Graph skeleton 94
4.18 Relative design efficiency versus the number of iterations under ϕ_1 . . . 95
4.19 Relative design efficiency versus the number of iterations under ϕ_2 . . . 96
4.20 Boxplots of design efficiencies (100 random starts) 97
4.21 Boxplots of efficiencies for NSYM without iterations 98

5.1 Blocking structure of Example 2.4.1 106
5.2 A Facebook ego-network 109
5.3 Modularity values 110
5.4 Spectral Clustering 110
5.5 Hasse diagram for the collection of models 112
5.6 Boxplots of efficiencies for ϕ_1 114
5.7 Boxplots of efficiencies for ϕ_2 115
5.8 (a) Social network with 3 clusters; (b) degree distribution 116
5.9 Optimal designs (under NBM) for ϕ_1 and ϕ_2 117
5.10 Boxplots of efficiencies for ϕ_1 (assuming NBM) 118
5.11 Boxplots of efficiencies for ϕ_2 (assuming NBM) 118
5.12 Pairs of units of different distances receiving the same or different treatments 119
5.13 Bias in treatment effects due to network effects 124
5.14 Bias for treatment effects when $\gamma_1 = \gamma_2$ 124
5.15 Social network with 5 clusters 125
5.16 Optimal designs (under NBM), ϕ_1 and ϕ_2 126
5.17 Boxplots of efficiencies for ϕ_1 (under NBM) 127
5.18 Boxplots of efficiencies for ϕ_2 (under NBM) 127
5.19 Boxplots of efficiencies for ϕ_1 (under NBM, accounting for all LNDs) . . 131
5.20 Boxplots of relative efficiencies for ϕ_1 and ϕ_2 (1000 simulated SW networks) versus different clustering coefficients 132
5.21 Boxplots of modularity scores corresponding to the 1000 simulated networks for the different clustering coefficients 132

5.22 Boxplots of relative efficiencies for ϕ_1 and ϕ_2 for 100 simulated SW networks versus different clustering coefficients 133
5.23 Removing or adding a connection 133
5.24 Optimal designs, removing connection between units 6 and 12 134
5.25 Optimal designs, adding connection between units 1 and 5 134
5.26 Network misspecification; 100 simulated networks 135
5.27 Blocking misspecification; true (*left*), test (*right*) 137
5.28 Optimal designs for ϕ_1; true (*left*), test (*right*) 138
5.29 Misspecification of unit labelled 13 belonging in a different block: true (*left*), test (*right*) 139
5.30 Optimal designs for ϕ_1; true (*left*), test (*right*) 139

6.1 Field layout of the motivating agricultural experiment and treatments . 145
6.2 Hasse diagram of the model structure (without network effects) 147
6.3 Different connectivity graphs: $\mathcal{G}_1$ (*left*) and $\mathcal{G}_2$ (*right*) 151
6.4 Experimental design in Rothamsted α-RCD (year 2016) 157
6.5 Near-optimal RCNBD1 157
6.6 Near-optimal RCNBD3 158
6.7 Near-optimal RCNBD4 158
6.8 Near-optimal RCNBD5 159

A.1 Optimal designs when removing an edge between units 1 and 5 176
A.2 Optimal designs when removing an edge between units 1 and 10 176
A.3 Optimal designs when removing an edge between units 4 and 10 177
A.4 Optimal designs when adding an edge between units 1 and 4 177
A.5 Optimal designs when adding an edge between units 1 and 12 177
A.6 Optimal designs when adding an edge between units 7 and 9 178
A.7 Optimal designs when removing edges between units 1 and 5 and 1 and 10178
A.8 Optimal designs when removing edges between units 1 and 5, 1 and 10 and 4 and 10 178
B.1 All LNDs 182
B.2 All NBDs 182
C.1 An optimal CRD 184
C.2 An optimal RBD 184
C.3 An optimal RCD 185
C.4 An optimal RCBD 185
C.5 A near-optimal LND 186
C.6 A near-optimal NBD 186
C.7 A near-optimal RCND1 187
C.8 A near-optimal RCNBD1 187
C.9 A near-optimal LND2 188
C.10 A near-optimal NBD2 188
C.11 A near-optimal RCND2 189

C.12 A near-optimal RCNBD2 189
C.13 An optimal RCBD3 190
C.14 A near-optimal NBD3 190
C.15 A near-optimal RCNBD3 191
C.16 A near-optimal LND4 192
C.17 A near-optimal NBD4 192
C.18 A near-optimal RCND4 193
C.19 A near-optimal RCNBD4 193
C.20 A near-optimal NBD5 194
C.21 A near-optimal RCNBD5 194

List of Tables

1.1 Examples of experiments on networks 6

2.1 The general characteristics of the networks 35

3.1 List of designs; top row optimal design for ϕ_1 58
3.2 List of designs; top row optimal design for ϕ_2 58
3.3 The general characteristics of the networks of Figure 3.7 63
3.4 Expectation of bias of balanced designs at each proportion 68
3.5 Efficiencies of the optimal designs under network misspecification 71
3.6 Network measures of the different misspecifications of the network . . . 72

4.1 Approximate time in seconds required to find the optimal designs 90

5.1 Comparisons of the designs for ϕ_1 under different models 113
5.2 Comparisons of the designs for ϕ_2 under different models 113
5.3 Comparisons of the designs for ϕ_1 under different models 117
5.4 Comparisons of the designs for ϕ_2 under different models 118
5.5 Pairs of neighbouring subjects receiving the same/different treatment . . 120
5.6 Table 5.5 with the distribution within blocks 120
5.7 Design bias under model misspecification (ϕ_1) 121
5.8 Design bias under model misspecification (ϕ_2) 122
5.9 Comparisons of the designs for ϕ_1 under different models 126
5.10 Comparisons of the designs for ϕ_2 under different models 126
5.11 Pairs of neighbouring subjects receiving the same/different treatment . . 128
5.12 Table 5.11 with the distribution within blocks 128
5.13 Design bias under model misspecification for the small-world network (ϕ_1)129
5.14 Design bias under model misspecification for the small-world network (ϕ_2)130
5.15 Sets of fixed parameters $\boldsymbol{\beta}$ 136
5.16 Analysis with (a) NBD or (b) RBD (3000 simulations, type (I) of misspecification) 138
5.17 Analysis with (a) NBD or (b) RBD (3000 simulations, type (II) of misspecification) 140

6.1 Candidate classes of designs—King's case 153
6.2 Comparisons of the designs under the different models-class 1 154

6.3 Comparisons of the designs under the different models-class 2 154
6.4 Comparisons of the designs under the different models-class 3 154
6.5 Candidate classes of designs—Farmer's case 155
6.6 Comparisons of the designs under the different models-class 1 155
6.7 Comparisons of the designs under the different models-class 2 156
6.8 Optimality function values under the different models for α-RCD . . . 156

A.1 Table for Figure 3.11 (with unique entries) 175
A.2 Efficiencies of the optimal designs under network misspecification 176
B.1 Near-optimal designs under NBM 179
B.2 Table for Figure 5.13 (with unique entries) 183

Declaration of Authorship

I, Vasiliki Koutra, declare that the thesis entitled "Designing Experiments on Networks" and the work presented in the thesis are both my own, and have been generated by me as the result of my own original research. I confirm that:

- this work was done wholly or mainly while in candidature for a research degree at this University;
- where any part of this thesis has previously been submitted for a degree or any other qualification at this University or any other institution, this has been clearly stated;
- where I have consulted the published work of others, this is always clearly attributed;
- where I have quoted from the work of others, the source is always given. With the exception of such quotations, this thesis is entirely my own work;
- I have acknowledged all main sources of help;
- where the thesis is based on work done by myself jointly with others, I have made clear exactly what was done by others and what I have contributed myself;
- none of this work has been published before submission.

Signed:

Date:

Acknowledgements

I would like to thank my supervisors Prof. Steve Gilmour, Dr. Ben Parker and Prof. Peter Smith for their feedback and advice. I would like to make special mention to Prof. Steve Gilmour for his precious guidance during all stages of my research endeavour and for his timely and spontaneous advice e.g. 'ignore the rules, but remember the principles'.

Additionally, I would like to thank my funding sponsor the Economic and Social Research Council that made this PhD journey possible. I am also grateful to Dr. Rubèn Sánchez-García for our stimulating and inspiring discussions and Prof. Dave Woods for his meticulous feedback at my upgrade.

I also wish to pay tribute to those amazing persons and adept scientists that I came to meet during my just four years at the University of Southampton and Rothamsted Research, who I have the honour to consider as my friends and whose insight and encouragement was far from negligible in the conduct of this research.

Finally, my hearty thanks to friends and family, the unsung heroes, for their enduring love and support in this endeavour.

Chapter 1

Introduction

The main idea of planning an experiment is to compare the effects of alternative interventions (*treatments*) by observing the responses of a number of entities (*experimental units*) to which these interventions are applied. An experiment consists of applying one treatment to each experimental unit, keeping the treatment allocation under the experimenter's control, and making one (or more) observations. The statistical design of experiments ensures that the experiment produces valid and information-rich data. Designing experiments on networks, which consist of interconnected experimental units, raises questions about a common assumption used in classical experimental designs. This assumption is commonly referred to as 'non-interference' and pertains to the treatment effects, stating that the response of a unit remains unaffected by treatments assigned to other units. Relaxing this assumption can improve the design efficiency and the validity of the analysis of the experiment conducted on a network. However, hitherto there is very little published work on designing experiments on networks. The aim of this thesis is to make some progress in this novel area, by combining two major areas of research, namely the design of experiments (DoE) and network theory (NT).

1.1 Preliminaries

It will be convenient at this point to introduce some standard terminology (more details will be given in Chapter 2 and other chapters).

Networks of all kinds drive the modern world, and can be used to represent many different systems: social networks, communication networks, financial networks, biological networks, geographical proximity networks. Networks, either real or virtual, are a collection of entities that are interconnected. In social networks, for instance, the connections can represent some form of friendship, communication, proximity or collaboration. Examples of social networks include acquaintances or friends, networks of colleagues by profession, Facebook friendships or the Twitter follower relationship. These structures outline the (direct and indirect) connections between entities and provide a framework for exploring their interactions and behaviours. Mathematically, networks are represented by *graphs*, with *vertices* (or *nodes*) denoting the entities and *edges* among them indicating the links. Networks are an important source of 'big' data and an area of research in many disciplines such as healthcare, economics, politics and others. Examples of networks and experiments conducted on them will be given in the following section.

Design of experiments (DoE) is a well-developed methodology for obtaining valid and efficient answers to questions that the experiment is intended to answer, given the practical, financial and ethical constraints. Validity stands for unbiased quantities (accuracy) and efficiency for low variance (precision) of the estimated quantities. The pivotal points in experimental design are the choice of experimental units, the determination of a suitable set of treatments that meet the objectives of the experiment, and the allocation method of the treatments to the experimental units. Among others, a standard reference on DoE theory is the book of Cox (1958).

Experimental units and treatments: A well-structured experimental design relies on the precise definition of experimental units as well as on the treatments applied to them. Experimental units can take the form of runs of a process, plots of land, collections of objects or other entities on which different treatments are allocated. They correspond to the smallest division of experimental material such that any two units may receive different treatments in the actual experiment (Cox, 1958, Ch.1). If an experiment is set to run over a period of time, with the observations being collected sequentially, then the collection times of the day can be the experimental units, depending on how the treatments are allocated. For the purpose of this thesis, the concept of 'experimental units' from the design theory perspective coincides with the one of vertices from graph theory. We will mainly refer to them for brevity by the term units. Treatments are what is compared to answer the experimental questions. They may represent different diets, drugs, chemicals, genotypes of crops etc. For a greater insight on the choice of experimental units and treatments refer to Mead *et al.* (2012, Ch.3 and Ch.5). Both units and treatments can possess structures, which the experimenter should consider. The unit structure may arise from the choice of units with regard to a specific study, as for instance units participating in 'cross-over' trial, or units connected in a network, or may be related to suspected sources of variation that are accounted for by grouping of units (blocking). In this thesis the experimental units are connected in a given network. We will discuss the structure of treatments later.

Blocking: The experimental units can be organised into blocks, such that units belonging to the same block are expected to respond similarly to an external stimulus. For example, in clinical trials the units, which may correspond to different patients, can be grouped by gender, age, or other physical characteristics. Grouping patients in this fashion ensures that they will be more similar with respect to all relevant features except for the treatment they received. When the 'natural' variation between units is small, it is easier to identify when a treatment has made a real difference. An example of a blocking system which can be appropriate for experiments on networks will be defined in Chapter 5, while accounting for the connections of units within a network.

Cross-over designs: This type of design is used in various fields including animal science, pharmaceutical studies and clinical trials, where different treatments are applied successively in different time periods to subjects so that an experimental unit corresponds to a subject and period combination. As an example consider a comparison of two different drugs on a chronic condition such as asthma in different consecutive periods of time on the same patients. Thus here lies a structure of units within patients, with different units corresponding to different time periods for the same patient. Notice that the patients essentially act as blocks. One of the standard references on the topic of cross-over designs is that of Jones and Kenward (2003).

Treatment structures: Treatments can be either structured or unstructured. Treatment structures may refer to a single factor or to crossed or nested factors used to define a set of treatments in an experiment. For instance, a factorial structure is a commonly used treatment structure, which involves 'treatment factors', each having a set of possible values called 'levels'. The treatments correspond to different combinations of these levels across all possible factors. Consider, for instance, an agricultural experiment for measuring the crop yields after planting different types of seeds. Suppose that the different seed varieties constitute the first factor and the different quantities of the variety the second. Then the specific combination of factor levels constitutes a treatment whose effect we want to compare with other treatments. In this thesis we focus on the comparison of unstructured treatments, that could mean that for the previous example we compare different types of seeds for a single given fixed quantity.

Experimental assignment and design properties: The final step of the experimental design involves the choice of the appropriate method of allocation of treatments to units. This can take place under a randomised or non-randomised scheme. A repeated application of a treatment to multiple units is called *replication* and is important in order to make treatment comparisons (for more details see Mead *et al.*, 2012, Ch.6). In addition to replication, another fundamental aspect of a good design is balance. Balance is a valuable property of a design and is achieved when the treatments have equal replication, so that any two treatments occur equally often in the design. Additionally, in the case of block designs, each pair of treatments occurs within a block exactly the same number of times over the whole experiment. A characteristic of the balance property is that the precision of all treatment comparisons is equal. Pearce (1963) was the first one to notice this property, by explaining that the design is said to be balanced when the 'sum of weighted concurrences for any two treatments' is equal (pairwise balance) resulting in all simple treatment comparisons being equally precise. Another important property of block designs is orthogonality. Orthogonality of treatments and blocks enables the effects to be considered separately, which allows for independent interpretation of effects of treatments and of blocks (for more details refer to Mead *et al.*, 2012, Ch.7). Although several of the designs we provide are unbalanced, we will give more insight into the properties and other issues of interest in the following chapters.

Additivity and non-interference assumption: The additivity assumption underlies the majority of the standard designs. It implies that a response of a unit is not affected by treatments assigned to other units (Cox, 1958, p. 19). This fundamental assumption is also termed 'non-interference assumption' or 'stable unit treatment value assumption' (SUTVA, as named by Rubin, 1990), and is crucial for the validity and precision of the experimental results. There are three situations where the additivity assumption may be doubtful and unrealistic: when there is time dependence, spatial dependence or network dependence. Time dependence, for instance, may be present when treatments are applied in a sequence (as in cross-over designs) and therefore, there may be effects carried over from one treatment period to the next ('residual effects'). For instance, a treatment applied to a subject at a given period of time may be affected by a treatment that the subject received in the preceding period. Spatial dependence encountered in field experiments can result in a response from one plot which receives a treatment being affected by treatments on adjacent plots ('neighbour effects'). Network dependence, considered for instance in social networks, results in dependent responses due to the interpersonal communication and social influence among the network members ('network effects'). Section 1.3 further discusses this crucial assumption together with some critical steps made in the development of experimental design when potential outcomes or responses are dependent.

Spillover effects (network effects): The propagation effects as a result of units' interference are known under the terms indirect, neighbour, residual and spillover effects, and constitute a violation of the non-interference assumption. Spillover effects can generally produce systematic biases, rendering the removal of those biases necessary in order to obtain reliable comparisons among alternative treatments. Thus some particular precautions may be adopted to protect against problems caused by interference in order to improve the precision of treatment comparisons. In general, we can distinguish two types of spillover effects: one occurring via response interference and one occurring via treatment interference, both of which are discussed later on. This thesis focuses more on treatment interference effects based on a given network structure and therefore we shall refer to them as network effects. Another possible distinction is when spillover effects are regarded as incoming or outgoing effects. This work perceives them as incoming effects, resulting from treatments received by one's immediate neighbours (refer to Sections 3.1 and 3.2). All these issues will be considered in more detail in the following chapters.

Example of a networked experiment: Consider a company, which is interested in conducting a commercial experiment on a social networking site in order to compare the effectiveness of different advertisements concerning a product. The responses are measured based on the purchased quantities during the week immediately following the advertising campaign. The company's goal is to maximise the appeal of the product, and as a result its demand, by effectively using the advertisement as a tool to affect the purchasing decision of potential customers. Our aim is to understand how advertisements affect directly and/or indirectly the purchasing decisions. In other words the advertisements may have an effect not only on the recipient but also on one's (virtual) friends. With the advent of viral marketing, these indirect effects of the advertisements are generally desirable by the marketing campaign and are crucial for the purchasing demand of the product. Typically, from the experimental design point of view, such advertisements can possess complicated treatment structures related to marketing and design characteristics such as the portrayal of the product in the advertisement, the colours used, the recommended price if mentioned, the style and various other features of competitive marketing strategies aiming at making the advertisement more memorable and interesting. The above can be rephrased in the context of having a number of factors at different levels that represent different versions of comparative advertisement structures in an attempt to gain insight to the efficiency of the marketing experiment in a controlled environment. However, if the treatments are unstructured, this could mean that interest lies in making comparisons between defined advertisement formulations with some prespecified appearance characteristics.

Identifying but a few of a growing list of similar examples has given us the incentive to begin this research. The field of experimental design is constantly challenged by the problems that come up in science and industry. The following section discusses more examples with the intention of highlighting the diversity of applications this thesis can address.

1.2 Experiments on networks

A number of experiments and observational studies from various domains have been carried out on networks in an attempt to unveil spillover effects and gain insight into human behaviour, marketing analytics and the spread of diseases. Examples of such topics of research include the spread of behaviours (Centola, 2010; Christakis and Fowler, 2013), the transmission of political beliefs and the resulting different voting turnout (Bond *et al.*, 2012; Green *et al.*, 2013), the word-of-mouth peer-influence, such as purchasing a product or joining a community (Aral and Walker, 2011; Bapna and Umyarov, 2015) and the transmission of diseases (Griffin and Nunn, 2012; Risau-Gusman, 2012; Danon *et al.*, 2012). Our interest lies either in the spillover effects themselves or in the direct comparison of treatments in the presence of nuisance spillover effects. Table 1.1 additionally mentions a few examples of experiments on networks (including but not limited to social networks), showing the diversity of applications across different fields that can be addressed with the theory and methods developed in this thesis. These examples of experiments are synthesised from the literature and share common design features. There is an underlying structure governing the experimental units, resulting in potential spillover effects.

Example (i) is similar to the example of Section 1.1. Advertisers' main goal is to measure the effectiveness of their advertising campaigns in order to achieve product aware-

Table 1.1: Examples of experiments on networks

Field of study	**Network links**	**Intervention**	**Response**	**Spillover effects**
i. Marketing: product recognition	Virtual friendships in a social media platform	Product advertisements	Product awareness	Product awareness among one's friends
ii. Agriculture: plant growth	Geographical proximity of plots	Chemical pesticide	Plant growth	Effect on the plant growth of pesticide applied to adjacent plots
iii. Politics: voting turnout	Interactions of registered voters in a congressional district	Social-pressure mailings	Individual voting behaviour	Influence on voting behavioural change among interacting individuals
iv. Education: healthy eating	Social ties among pupils of a school	Incentivised choices of healthy food	Option of snack	Influence on eating habits among interacting pupils
v. Charities: raising funds	Social ties within the context of a workplace	Different text messages	Donations	Viral effects affecting the neighbouring contacts
vi. Ecology: animal behavioral nudging	Animal interactions	Reward based intervention	Reaction speed	Effect on animals cohabiting the test area
vii. Economics: product price	Business interactions	Decrease in the product price of certain firms	Sales	Potential decrease in the product price of other companies
viii. Health: disease transmission, e.g. MRSA	Contact interactions of hospital inpatients	Targeted information on infection control, e.g. recommended ways of washing hands	Incidence of disease	Positive effect in the infection rate among nearby individuals
ix. Law enforcement: criminal activity	Geographic proximity within a city	Increased surveillance at certain districts	Crime rates	Rollover of criminality to nearby areas due to increased surveillance
x. Technology: newly acquired knowledge	Social ties among students of a university	Education oriented intervention practices, e.g. text books, tablets	Cognitive performance in test scores	Share of knowledge among students

ness. The impact of advertising on consumer behaviour can be measured through the number of views, website visits, downloads, and shares in social networking platforms or various other forms of conversions. The product advertisements can be allocated to connected individuals in a social media platform. There may be viral effects of the advertisements increasing the product awareness to connected interacting individuals. An example of such an experiment in this context was recently conducted by Dai and Luca (2016) on a restaurant review networking platform to assess the overall impact of different advertisements.

Network topology can also describe spatial dependence. Example (ii) outlines an experiment in agriculture, where the response on one plot may be affected by treatments, such as different chemical pesticides, on neighbouring plots as well as by the standard effect of this treatment on the plot. The objective of the experiment is the plant growth and the network structure can be based on the geographical proximity of plots. Examples of such experiments are field trials in Rothamsted Experimental Station (`http://www.rothamsted.ac.uk`), where neighbour effects, soil fertility trends and competition have been studied. Kunin (1998), for instance, acknowledges the potential existence of spillover effects in a well-known long term experiment, the Park Grass experiment. That experiment has monitored the plant species diversity for well over a century. It consists of a series of closely abutting field plots that have been subjected continually to different fertilisation treatments. Section 1.3 mentions work in agricultural field trials with neighbour effects being investigated in addition to the standard effect of the treatment applied to a particular plot.

Example (iii) concerns an experimental study of a voting turnout in political science, where social-pressure mailings can be distributed to registered voters in a congressional district who are eligible to vote. This pressure could be exerted by interacting individuals, household members and neighbours (e.g. within the same zip code), who might also read the mailing. Based on these considerations, Sinclair *et al.* (2012) conducted a multilevel experiment in political science in Chicago during the elections in 2009. They attempted to measure spillover effects in addition to the direct effects of the treatments applied to the voters, which correspond to voter-mobilization messages via postcards, and studied influence on the voting behaviour. In other words they investigated the propagation of the treatment effects within and between households in the same neighbourhood. They discovered that the postcard had a direct effect on the recipient as well as an indirect effect on his household members (at the household level). There was no evidence, whatsoever, of an indirect effect at the neighbourhood level (i.e. friends and acquaintances living in the same neighbourhood) as far as the voting pattern is concerned.

Other experimental studies could involve incentives with the objective of improving behaviours or outcomes in a specific domain such as education, health, labour market, and pro-social behaviours (e.g recycling, charitable giving, blood donation etc.). Examples (iv), (v), and (vi) fall in this category. Consider for instance an experimental study which aims to incentivise the choice of healthy food for improving the habitual decision to consume nutritious foods (Example (iv)). Angelucci *et al.* (2015) conducted an experiment on this basis, which involved incentives for choosing the healthier option between two different snack choices 'grapes and cookies'. The field experiment took place in school cafeterias in nine different schools during lunchtime, with the lunch tables that peers were sharing constituting the peer networks. The pupils were given the choice of alternative snacks based on the observations of their peers' incentivised choices. The study highlighted that both the direct and spillover effects of the incentives were substantial. The website `http://www.dietdetective.com` is also based on this idea of texting reminders to eat healthy via different interventions. It provides different types of reminders at different times, which can correspond to different treatments. Following a similar rationale, Example (v) describes an experiment where a humanitarian campaign aims to compare different text messages with the object of raising funds. As such a study may be conducted at various workplaces in order to compare a number of different candidate messages. The responses of the subjects can be measured by means of their donations. The subjects work together and might hold discussions with each other. Therefore, the text messages could have viral effects, as such affecting not only the person receiving the text message but also that person's friends. Moreover, it is worth considering that the act of one's donation, can be observed by others in the contact network leading to a 'contagion' effect. Similarly, Example (vi) involves animal behavioral nudging for accomplishing a task faster. Animal networks are networks of contacts between animals. They are the animal equivalent to human social networks. The experimental intervention in this setting is some kind of reward to encourage the animals to act/respond faster. The proximity of the animals may be regarded as a way to define the social network between them. The spillover effects may be based on the observation of nearby animals receiving treatments or responding to these treatments.

The remaining examples, (vii)–(x), displayed in the table can be easily understood and regarded in a real life context. The 'true' network organisational structure is sometimes difficult to define or obtain. In this thesis, the focus lies exclusively on static networks, i.e. networks in which units are assumed to be pre-specified and fixed during the time of experimentation. Section 2.4 discusses this issue in more detail. It is important to point out that in all these examples the non-interference assumption is untenable. Therefore, more elaborate procedures are required rather than standard experimental designs, which ignore potential interactions and by extension spillover effects. This necessitates the development of new methodological approaches in the field of design of experiments with connected experimental units, building upon the existing literature.

1.3 Related literature

Designing experiments for the special cases where there is time, spatial or network dependence challenges the 'non-interference' assumption (briefly discussed in Section

1.1). In this section, we will review the literature on investigating spillover effects resulting from interference. The section is arranged in a thematic order. We will provide a view of the critical steps in the development of design and analysis of experiments, when interference is present, but also literature on interference in networks for capturing (social) influence and network interdependencies. More technical issues and different interference models are discussed in Chapter 3.

1.3.1 Causes of interference and design properties

Common attempts in the literature to prevent interference involve allowing for 'guard' areas around plots when it comes to spatial dependence or resting periods ('wash-out' time) in cross-over designs when it comes to time dependence. Other methods for reducing or eliminating the spillover effects resulting from interference are related to modifications on the treatment allocation or restrictions on randomisation procedures (methods and examples are discussed in Mead *et al.*, 2012, Ch.8 and Ch.11).

An important work that incorporates different developments in the analysis of field experiments accommodating interference is that of Besag and Kempton (1986), which investigated different causes of association between neighbouring plots and provided appropriate models for better capturing each cause. Besides the method of widely separated plots receiving different treatments and other systematic arrangements, they discussed spatial techniques based on the differences of plot responses in one direction for removing from the data local trends of soil fertility (i.e. adjustment for fertility effects). Another cause of association they considered, arises from 'interplot competition', where plot responses are directly related to the values of neighbouring plots (i.e. response interference). They suggested an autoregressive formulation of an interference model to adjust for competition between plots. They also considered the case where plot responses are affected by the treatments applied to neighbouring plots (i.e. treatment interference).

From the perspective of experimental designs, work on interference has mainly concentrated on the construction of block and neighbour-balanced designs with appropriate randomisation schemes (David and Kempton, 1996). Neighbour-balanced designs were introduced by Rees (1967) in serology, and these are designs where each treatment has every other treatment as a neighbour, on either side, an equal number of times (Wilkinson *et al.*, 1983). In other words, neighbour-balanced designs have the property that for each ordered pair of treatments there is one plot that has the first chosen treatment as a left neighbour and the second chosen treatment as a right neighbour (Azaïs *et al.*, 1993). In block designs, on the other hand, it is desirable that all pairs of treatments occur in the same block an approximately equal number of times. Thus neighbour designs take account of the concurrence of treatments in adjacent units rather than units in the same block.

Subsequently, we discuss the literature relevant to the types of spillover effects related to the different causes of association among neighbouring units.

1.3.2 Adjusting for treatment interference

One of the most well known examples in the literature where, when applying a treatment to one unit part of its effect spills on to the other units, is the cross-over design with carry-over effects (or residual effects) (see Section 1.1). In such a design, the drug the patient received in the first period may have residual effects on the second drug's performance during the second period. Commonly used, especially in clinical trials, cross-over designs assume a time dependence only in one direction (influence by past values only). Experimentation where sequences of treatments are allocated over time can also be found in other areas of research, such as industrial research, animal feeding trials and psychology. For more information about and practical examples of cross-over designs see Jones and Kenward (2003) and Senn (2002) and further references therein. The time dependence results in making further assumptions about the parameters of the model describing adequately the experimental situation, by capturing carry-over effects or period effects or even some interactions among them which can affect the response. A typical model used for this situation is presented in Chapter 3 along with further discussion.

As opposed to time dependence which extends only in one direction (forwards), spatial dependence can be multi-directional. Cliff and Ord (1981, p.9) highlighted that space should be treated quite differently from time as there is no natural ordering. The development of design and inference on neighbouring experimental units was especially motivated by applications in agricultural and horticultural research fields, e.g. spatial field experiments. In a spatial experiment, units occupy fixed locations distributed throughout a region, e.g. agricultural field experiments. In such an experiment, the response on a given plot can be affected by treatments applied on neighbouring plots as well as by the treatment applied to that plot. As an example, the effect of a chemical pesticide applied to a plot can be potentially affected, in terms of plant growth, by its surrounding plots' treatments (see Example (i) of Table 1.1).

A noteworthy relevant published work is that of Pearce (1957), who mentioned the importance of taking into account both 'local' (i.e. direct) and 'remote' (i.e. indirect) effects. In one of his experiments on the effects of pollen on the size of apples, each treatment had a local effect on the tree branch (experimental unit) to which it is applied and an equal remote effect on all other branches of the same tree (block). He suggested efficient block designs and provided estimates for treatment comparisons within and between blocks. He also allowed for interactions between the remote effect of a treatment on a plot and the treatment applied locally to that plot. For that he considered an additional effect, which captures the total effect resulting from the treatments in a block, which differs from the one applied to that specific plot. Building upon the work of Pearce (1957), Draper and Guttman (1980) considered the neighbour effects to be proportional to the direct treatment effects multiplied by a coefficient of interference, which can be positive or negative depending on the ‘aggressiveness’ of each treatment. Besag and Kempton (1986) discussed the ideas of those authors and provided relevant applications on the use of neighbouring plot responses in the experimental analysis.

In contrast to the approach followed by the above-mentioned work, which primarily focused on the analysis rather than the design of experiments, Azaïs *et al.* (1993) and Druilhet (1999) considered neighbour-balanced designs with one- or two- sided neighbour effects. Azaïs *et al.* (1993) also provided a list of efficient circular neighbour-balanced designs. Note that in circular designs, each one of the plots has a left and right neighbour and the first and last plots are considered as neighbours. Research in this field has given rise to design and model issues that are related to: (1) the presence of guard plots (also known as border plots, which differ from the interior plots); (2) the arrangement of plots in one or two dimensions (typically with a regular structure of neighbours); (3) the presence of neighbour effects in one or both directions (left and/or right); and (4) the left-neighbour effects being equal to or different from the right-neighbour effects. Special mention has also been made of block designs where plots are grouped into blocks with experiment-specific characteristics, which are expected to behave similarly, e.g. geographical blocks of similarly situated plots, groves of trees with specific morphological characteristics etc.

Based on these considerations, a wide variety of possible models has been suggested for the case of treatment interference. Examples include the work of Druilhet (1999), Kunert and Martin (2000), Bailey and Druilhet (2004) and Kunert and Mersmann (2011), who provided models and efficient designs that concern experiments where units are arranged in a circle or a line for neighbours one apart. Their suggested designs were in their majority neighbour-balanced in the sense that all pairs of treatments occur in adjacent plots equally often. Druilhet (1999), in particular, considered designs with few blocks and guard plots at each end of the block, so that each interior plot has two neighbours. Kunert and Martin (2000) relaxed the assumption of guard plots and also proposed having different left and right neighbour effects with block sizes of three and four extending the model postulated by Druilhet (1999) and improving the design given that extended model. Consequently Kunert and Mersmann (2011) extended the work of Kunert and Martin (2000) by adapting their model with different interference from two sides to more than four plots per block. More recently, Parker *et al.* (2016) adopted the conceptual approach of Pearce (1957) and introduced a model, called the linear network effects model (LNM). The LNM bears many similarities with the model of Kunert and Martin (2000). However, it differs by relaxing the assumption of neighbour effects being controlled in only one way and allows for a network setting without the inclusion of blocks. Chapter 3 discusses the LNM and provides some new analytical results on designs for this model. Moreover we extend the LNM to experiments with blocks in Chapters 5 and 6.

1.3.3 Adjusting for response interference and spatial or temporal variation

As we have already pointed out, designs for correlated observations are common when data form a series in time or space (in violation of the non-interference assumption). Typically the nature of this heterogeneity is such that there is a significant correlation among units that are neighbours, either in time or space. Patterns of autocorrelation in time have been investigated in cross-over designs to gain insight into some time dependencies analogous to time-series (Matthews, 1987). A discussion on this topic with an extensive list of references can be also found in Hinkelmann and Kempthorne (2005, Sec. 19.8). However, specification of modelling approaches considering autocorrelation structures has been mostly seen in spatial experiments. Cliff and Ord (1981, p.16-9) mentioned the need to characterise a priori the form of spatial dependence of interest with a 'spatial proximity matrix' based on an appropriately defined set of weights related to the study at hand. The weights, for instance, may depend on the length of common boundaries between the plots or other appropriate metrics for controlling environmental or other differences in the field. We discuss some of these issues in Chapter 6, where we focus on a particular agricultural field experiment.

It was first suggested by Papadakis (1937) that the response from a plot is closely related to the responses from its immediate neighbours due to the spatial heterogeneity in the responses. In his 'nearest method' presented for control of soil fertility, the residual for one plot is given by both the difference between the observed yield for the variety in that plot and the average yield for the same variety in the whole experiment. In an early approach, accounting for spatial correlation by modelling the error structure, Williams (1952) considered neighbour designs with correlated plot effects. In particular, he discussed a low-order autoregressive process (at distance one or two) on plots allocated in a one-dimensional layout (line) or two dimensional layout (plane) with a few border plots. Butcher (1956) also discussed one-dimensional neighbour-balanced designs but extending to longer distances. The spatial analysis using the Papadakis method was further discussed by Bartlett (1978), who proposed the traditional use of blocks while adjusting the responses (yields) by accounting for the inherent positive correlation between the fertility of neighbouring plots (i.e. covariance on the residuals of neighbouring plots). Although not strictly concerned with correlation effects, designs with neighbour-balance in two-dimensions date back to Freeman (1979), who considered designs balanced for nearest neighbours in the context of 'polycross designs' for seed production and provided possible methods of construction of one- and two-dimensional neighbour-balance designs. In polycross designs, the neighbours are the immediate adjacent plots positioned in a cross arrangement. Kiefer and Wynn (1981) suggested one-dimensional block designs where the correlation occurs between immediate adjacent plots only within the blocks.

A fundamental source on measures of spatial autocorrelation and on spatial regression models is Cliff and Ord (1981). A body of literature associated with this work grew rapidly in the context of social networks (Leenders, 2002), as is discussed further below. Coming back to the subject of nearest neighbour methods for spatial statistics, Wilkinson *et al.* (1983) suggested a method in which local trends are removed by considering the second differences or partial differences. They also proposed alternative models for the analysis of field variety trials. Street and Street (1985) discussed the construction of designs that satisfy the balance conditions in Wilkinson *et al.* (1983). Gill and Shukla (1985) considered a different nearest neighbour method of analysis of block experiments with no neighbour effects but with fertility trends (autoregressive correlation model) and inter-plot correlation (moving average model). As we earlier mentioned, Besag and Kempton (1986) also suggested a nearest neighbour model for the analysis of soil fertility based on the first differences of plot responses. In that work, Besag and Kempton (1986) presented two competition models: the response interference model (for inter-plot competition) of Kempton (1982) and the treatment interference model of Pearce (1957). The response interference model assumed observations adjusted for the general mean and ignored the block effect, whereas the treatment interference model assumed one can jointly model competition effects and spatial variability as part of the treatment structure, while modelling the trend in only one dimension as part of the structure of errors. Efficient neighbour-balanced designs were also introduced by Kunert (1987) for positive correlated structure. Martin and Eccleston (2004) obtained block designs under a linear variance model (with autocorrelation in errors).

All these studies suggest methods for improving the treatment contrast estimates by removing from the data any locally linear trends, which correspond for instance to the pattern of fertility, or by adjusting for the response interference among neighbouring plots. In this thesis, the spillover effect depends on the treatment applied to the neighbours. However, later we discuss another setting where the spillover effect depends on the actual response from the neighbouring units. In particular, building on the autoregressive formulation for the responses in the work of Besag and Kempton (1986), we present a model in the context of networks in Section 7.2.1, where units' responses are directly related to the responses of neighbouring connected units.

1.3.4 Recent work on spillover effects and modelling influence in networks

In recent years a new wave of scientific work has emerged that strives to identify and even measure spillover effects, due to their increasing importance.

In certain studies the structure of the spillover effects can be arbitrary, either known or unknown. Aronow and Samii (2013) and Coppock and Sircar (2013), both focused on estimating average unit level causal effects of exposures in the presence of interference, which has either a known or unknown structure. The average causal effect (of a binary variable) on a study population is typically defined by a contrast of means of potential outcomes generated in the experiment (one for each level of that variable). Aronow and Samii (2013) and Coppock and Sircar (2013) both suggested inferential frameworks for carrying out analyses to run experiments when there are spillover effects (SUTVA violation). A brief review on recently developed frameworks for statistical inference is written by Bowers *et al.* (2013) which focuses on the application in networks. In other studies, however, spillover effects are governed by a known structure. For instance, Sinclair *et al.* (2012) and de Miguel Luken and Tranmer (2010) both conducted experiments in multilevel settings. Other recent literature in the field of experimentation on networks in the context of estimating the average treatment effect is that of Eckles *et al.* (2014) and Basse and Airoldi (2015). This work follows conceptually the idea of treatment interference and response interference. In particular, Eckles *et al.* (2014) dealt with social interference where the expected responses are a function of the treatment assignments to the units, while Basse and Airoldi (2015) focused on the network-correlated outcomes where the expected responses of a unit is a function of that unit's covariates but also those of other units.

However, most of this work follow the framework of the Rubin causal model and potential outcome framework (Rubin, 1974), with the interest in measuring a quantity, which is the difference of the effects of two treatments on the potential average outcomes where the treatments are applied to all units each time. Thus the causal effect of the treatment involves the comparison of the two potential outcomes one for each treatment. A thorough book on this topic with a comprehensive presentation of these conceptual issues is that of Imbens and Rubin (2015). Aral (2016) provided a review of recent advances in networked experimentation, highlighting the importance of developing modern design approaches that study the propagation of network effects with the objective of drawing causal inference. Another relevant work is that of Bowers *et al.* (2016) that showed that standard designs used under non-interference schemes are inefficient when used in networked experiments and highlighted the importance of incorporating the potential network effects in the model. However, this potential outcome framework is beyond the scope of the current text. This thesis mainly focuses on the computation of optimal experimental designs on networks, where the treatment assignment to units is governed by a deterministic rule rather than by a completely random mechanism. Moreover, the spillover effects are governed by a known network structure, within which the potential interactions among experimental units take place.

Apart from the above mentioned studies, there is a vast literature on modelling influence on networks and capturing network interdependences. These include:

- network autocorrelation models (Doreian, 1980; Cliff and Ord, 1981; Anselin, 1988). These models are the most widely used in social network analysis as a way of measuring correlation in the responses of members of a social network. A detailed presentation of these models is given in Section 3.1 (for an extensive review, refer to Leenders, 2002).
- Multilevel models (Sinclair, 2011; Sinclair *et al.*, 2012; de Miguel Luken and Tran-

mer, 2010), where the population under investigation is considered as having a hierarchical structure, in which each lower-level unit is nested within a single higher level unit, e.g. individuals (at level-1) within households (at level-2) nested within municipalities (at level-3). Sinclair (2011) suggested strategies for designing randomised experiments in multilevel settings. This work emphasised the importance of accounting for the hierarchical structure, when present, as well as considering the randomisation of assigning treatments to the population (see Example (iii) of Table 1.1). Another study on a multilevel context was conducted by de Miguel Luken and Tranmer (2010) who fitted a multilevel logistic regression model to investigate the personal support networks of recent immigrants to Spain.

- Multiple membership models (Hill and Goldstein, 1998; Browne *et al.*, 2001), which treat sources of dependences jointly with the dependences in a social network, e.g. geographical areas or groups to which an individual belongs. More specifically, they allow for variations that occur as a result of dependences within a social network, as well as within other groups to which an individual belongs, to be concurrently assessed. In recent research, Tranmer *et al.* (2014) claim that network autocorrelation models are quite problematic and can lead to misleading and inconsistent results since they ignore other possible underlying levels of the population structure. They show that multiple membership models are more suitable for estimating social network dependences on one unit's responses in the context of other group dependences in a multilevel framework, reflecting better the relationships among correlated observations. For example, each individual can belong to a number of overlapping social settings, e.g. family, neighbourhoods, workplace, and as a result there is a relative variation in his/her response at the individual, network and group levels. This kind of structure is not purely hierarchical but follows a multiple membership (i.e. crossed) structure, since the groups may overlap at the same level. In other words multiple membership models are, in a way, a combination of autocorrelation and multilevel models. Tranmer *et al.* (2014) used such models to analyse the individual responses of academic performance and self-assessed health status. Thus, their study was conducted by taking into account the social network dependences together with other group dependences, namely school and areas to which an individual belongs, based on a subsample from the Adolescent Health dataset from the US.
- Stochastic actor-based models for analysing the dynamics of networks and/or changing attributes of the network members over time (Snijders *et al.*, 2010; Veenstra and Steglich, 2012). These models are relatively new and more complicated than most statistical models and are useful for longitudinal network data. Dahl and Van Zalk (2014) conducted a longitudinal study for examining the role of peer influence on adolescents' political behaviour, by fitting a stochastic actor-based model.

Within the scope of this thesis, we do not assume a hierarchical structure or any other additional complications. Our main focus lies in accounting, to an extent, for treatment interference among units resulting from the network dependences, using a parsimonious and easily justifiable model for such an experimental situation. At this introductory stage in the design field on networks we believe we first need to focus our research on simpler models and to gain experience in their application and an in-depth understanding of associated design issues. Nevertheless, it seems reasonable to expect that any model that somehow gauges spillover effects will serve to produce substantial gains in the accuracy and precision of an experiment in a network.

1.4 Contribution

A large proportion of experiments throughout the world is conducted without sophisticated experimental design; this should become a practical concern in networked experiments where the experimental units are connected. When the underlying network structure is ignored this could lead to invalid inferences and unreliable conclusions. Thus, for the sake of improving the accuracy in experiments, it is very important for the statistician to control for the network heterogeneity by adjusting for the interference between neighbours' responses or neighbours' treatments. This work makes an original contribution by developing novel methods for designing experiments on networks with a special focus on social networks, which can be immediately put into practice. The main contributions in the optimal experimental design methodology are: (*i*) investigating the impact of different network structures on the experimental design properties, (*ii*) developing and appropriately modifying algorithms in order to find optimal designs for networked data in different experimental contexts, (*iii*) investigating methods of taking into account the graph structure of the network and especially its symmetries so as to substantially reduce the computational time of finding optimal designs in complex real-world networks (e.g. from weeks to seconds); (*iv*) developing flexible methods for finding block designs with appropriately defined blocks, while taking into account the interrelations of groups of units within the network (an application is given for a subset of a real-world social network); (*v*) determining optimal row-column designs with network effects for use in agriculture; and (*vi*) discussing a model that account for network heterogeneity by including an autocorrelation of the responses component. Throughout this thesis, several optimal designs are obtained, many of which are compared in terms of efficiency to conventional and other more elaborate designs, justifying that accounting for network effects results in increased precision of the treatment contrasts' estimates. Various issues of interest such as design robustness to misspecification of the model or to misspecification of the assumed network are also investigated.

1.5 Thesis outline

The thesis is organised as follows. Chapter 2 discusses fundamental principles and technical concepts of experimental design and graph theories.

Chapter 3 focuses on the recent work of Parker *et al.* (2016), which introduced the linear network effects model (LNM) and developed optimal designs on connected units. We extend this work and provide proof of some analytical results. Moreover, we construct several optimal designs on networks where we explore how connections affect the optimal allocation of the treatments to the network members. We also compare designs that account for network effects to standard designs by means of efficiencies and biases. The generic structure of the exchange algorithm used throughout this thesis for finding optimal designs is also described here.

Chapter 4 provides a computationally fast way for designing experiments for large and complex networks. In particular, we provide an algorithmic approach for designing experiments efficiently while utilising the network structure and its properties (i.e. network symmetry).

Chapter 5 considers flexible blocking structures that exploit the community structure of a network. Particularly, it provides an adaptable framework for obtaining optimal block designs, where blocks have been defined using spectral clustering methods and graph techniques. We assume that network members that belong to a community may respond similarly to an external stimulus and therefore we define the blocks of the experimental design to correspond to these communities. Comparisons of the performance of different efficient designs for given example networks are also provided in this chapter.

Chapter 6 makes a transition to a particular agricultural experiment where the network structure is related to the distance among the plots or to the farm operations. There is often great variation in the yields of plots of land, and therefore by grouping plots of low geographical proximity in blocks we reduce the effect of this variation. This specific idea is investigated in this chapter, where we consider a double blocking structure of the units and construct row-column designs with network effects.

Chapter 7 concludes this thesis with a summary and further research directions. Moreover, it discusses some model diversification where the response of a particular unit is adjusted by the responses of its connected units and it suggests the future steps for the construction of the designs for that model.

Chapter 2

General framework

This chapter provides some of the fundamental elements and technical notation of experimental design theory and network theory, which are used throughout this thesis. In particular, Sections 2.1 and 2.2 discuss respectively the optimal design theory and models used for standard designs, whereas Section 2.4 contains the essentials of network theory. Section 2.3 outlines some commonly used algorithmic approaches for finding optimal designs. The chapter ends with a concluding discussion and provides insight into the methodology of designing experiments on connected units within a network setting. As a note, we maintain a consistent notation throughout the thesis although at times this means deviating from some of the conventions used in the corresponding literatures.

2.1 Preliminaries on optimal designs

Scientific experimentation (introduced by Fisher, 1935) includes the set up of a statistical model to adequately describe the experimental situation at hand and the development of the statistical design associated with that model. However, the model can hardly ever be exact, but can only be a good approximation of the real, yet unknown, relationship between the experimental factors and response variable(s). Let us assume that we have a linear model, which describes a relationship between the response (outcome) and the independent variables (factors) plus the error component (noise). The matrix formulation of the model is

$$\boldsymbol{y} = X\boldsymbol{\beta} + \boldsymbol{\epsilon},$$

where $\boldsymbol{y}$ is the $n \times 1$ vector of responses (with n denoting the total number of units), $\boldsymbol{\beta}$ is a vector of p unknown parameters corresponding to the model terms, X denotes the $n \times p$ extended design matrix, each row consisting of the independent variables at each data point, and each column to a parameter of the model and $\boldsymbol{\epsilon}$ is a vector of random variables with expectation $\mathbb{E}\left[\boldsymbol{\epsilon}\right] = \mathbf{0}$ and variance $var\left(\boldsymbol{\epsilon}\right) = \sigma^2 I$ (where I is the identity matrix), which represents the random variation among experimental units.

The design matrix is the key component in the planning process as it is linked with inference properties. The Fisher information matrix, M, is a $p \times p$ matrix given by $X^T X$. This matrix is proportional to the inverse of the variance-covariance matrix for the ordinary least squares estimators $\hat{\boldsymbol{\beta}}$ of the model's parameters (i.e. $\hat{\boldsymbol{\beta}} = M^{-1} X^T \boldsymbol{y}$). In other words,

$$var\left(\hat{\boldsymbol{\beta}}\right) = \sigma^2 M^{-1}.$$

Our primary interest is to estimate a set of the model parameters as precisely as possible. If the rank of M is less than p, that is if the p columns of M are linearly dependent, then the least squares equations ($X^T X \hat{\boldsymbol{\beta}} = X^T \boldsymbol{y}$) cannot be solved directly unless additional constraints are imposed. This issue will be further discussed in the following section. In order to obtain precise parameter estimates we can minimise the variances of estimators of a certain linear combination of the elements of $\boldsymbol{\beta}$. However, depending on the experimental objectives, different interests on aspects of the model parameters imply different criteria for optimising the design involving evaluating different functions of the Fisher information matrix.

Experimental designs for a chosen statistical model are defined as optimal with respect to a specific design criterion among a set of all possible competing designs of the same size that define the *design space* or study region Ξ. Design criteria are based on an objective function (ϕ) and are either maximised or minimised. There are several design criteria proposed in the literature, e.g. distance criteria, compound criteria and other. However, the most common criteria are the information-based criteria, whose calculation is based on a function of the information matrix (Kiefer, 1959). We present below some of them. For a more in depth review of these and other criteria as well as their application please refer to Pukelsheim (1993, Ch.6) and Atkinson *et al.* (2007, Ch.10).

– D-optimality maximises the determinant of the information matrix. This is equivalent to minimising the generalised variance of the parameter estimators.

– A-optimality minimises the trace of the inverse of the information matrix. This is equivalent to minimising the average of the variances of the parameter estimators.

– E-optimality maximises the minimum eigenvalue of the information matrix. This implies the minimisation of the maximum variance of all possible normalised linear combinations of parameter estimators.

In order to determine the optimal designs in this thesis a 'modified A-optimality' criterion is used, the L-optimality criterion (introduced by Fedorov, 1972). L-optimality ('linear optimality') minimises the average variance of a pre-specified set of linear functions of the estimators of the parameters $\boldsymbol{\beta}$ (Atkinson *et al.*, 2007, Ch.10). Thus q parameters or linear functions of the p parameters in the model are of interest, with $q \leq p$. Let the q linear combinations be $S^T \boldsymbol{\beta}$, where S^T is a $q \times p$ matrix. If ξ is a design for which $M(\xi)$ is non-singular, the variance matrix of the least squares estimator of $S^T \boldsymbol{\beta}$ is proportional to $S^T \left\{M(\xi)\right\}^{-1} S$. However, if $M(\xi)$ is singular, the variance matrix of the least squares estimator of $S^T\boldsymbol{\beta}$ is proportional to $S^T\left\{M(\xi)\right\}^- S$, where M^- is any generalised inverse of M, that is any matrix such that $MM^-M = M$. Thus the minimisation of the criterion function

$$\phi(\xi) = \operatorname{tr}\left(S^T\left\{M(\xi)\right\}^- S\right) = \operatorname{tr}\left(\left\{M(\xi)\right\}^- L\right), \tag{2.1}$$

where $L = SS^T$, leads to an L-optimal design. Thus for the L-optimal design ξ^*

$$\phi(\xi^*) = \min_{\xi\in\Xi}\phi(\xi), \tag{2.2}$$

where $\phi(\xi^*) = \phi^*$ is the *optimal function value.* The L-optimality criterion is an extension of A-optimality and they are identical when $L = I$. Moreover, L-optimality also includes the special case of c-optimality when $q = 1$, for designs with one linear combination of the model parameters to be estimated. For a discussion of these two special cases refer to Atkinson *et al.* (2007, Sec.10.5).

Comparative measures of design performance also involve the information matrix. In particular, we often compare the performance of two designs using their relative efficiency. The *relative efficiency* with respect to the objective function ϕ (see Equation (2.1)) of a design ξ_2 compared with a design ξ_1 is given as

$$\text{Eff}(\xi_1, \xi_2) = \frac{\phi(\xi_1)}{\phi(\xi_2)}. \tag{2.3}$$

We can also define the *L-efficiency* of a design ξ as $\text{Eff}(\xi) = \text{Eff}(\xi^*, \xi)$, where ξ^* is L-optimal design.

In the following chapters, we come back to these definitions when assessing the performance of different optimal designs based on different models for a number of given networks.

2.2 Standard randomised designs

In the simplest form of design the aim is to compare different treatments by randomly assigning them to experimental units. There are n experimental units in the experiment and m treatments to compare. In a Completely Randomised Design (CRD), the m treatments are randomly applied on the experimental units. The randomisation procedure is based on a pseudo-random number sequence that can be easily generated via computer software and is used for the allocation of the treatments to units (see, for example, Hinkelmann and Kempthorne, 2007, Sec. 6.2 and Mead *et al.*, 2012, Sec. 2.5). Randomisation ensures that any possible treatment allocation occurs equally often, removing subjectivity in the allocation of treatments to units ('selection effects') in order to eliminate biases. Assuming that the effect of the treatment is additive, the model derived from the randomisation scheme is provided by the following equation

$$\text{CRM}: \quad y_j = \mu + \tau_{r(j)} + \epsilon_j, \tag{2.4}$$

where $j = 1, 2, \ldots, n$, y_j is the response from unit j, μ is the overall mean, $r(j)$ is a function defining the treatment applied to unit j with $r(j) = s \in \{1, 2, \ldots, m\}$, τ_s is the deviation of the average response of the treatment s from the average value of the response from the set of treatments (direct effect of the treatment s applied to unit j) and ϵ_j is the deviation of unit j from the average response (error term). We assume that the errors are independently distributed with mean value equal to 0 and some fixed variance σ^2. Additionally, we often assume normality of the error term, i.e. $\epsilon_j \sim N(0, \sigma^2)$. Note that the abbreviation CRM corresponds to the model for a CRD.

If the n experimental units are arranged in κ blocks and m treatments are to be applied on these units, the response from the unit j in block i receiving some treatment is denoted by y_{ij} and the equation for this model is

$$\text{RBM: } y_{ij} = \mu + b_i + \tau_{r(ij)} + \epsilon_{ij}, \tag{2.5}$$

where $i = 1, 2, \ldots, \kappa$; $j = 1, 2, \ldots, n$, μ, as previously mentioned, represents the general mean (average response of all units), $r(ij)$ is a function defining the treatment applied to unit j in block i with $r(ij) = s \in \{1, 2, \ldots, m\}$ and τ_s represents the deviation of each treatment from the average response of the set of treatments in the experiment, i.e. it is the (direct) effect of the treatment s applied to unit j in the i-th block. The extra term b_i corresponds to the effect of the block i, which is the average deviation of the set of units in block i from μ, and ϵ_{ij} is the deviation of unit j from the average response of the units in block i (error term). As before, we assume that the errors are independently distributed with a common variance and we can additionally make the assumption of normality. The abbreviation RBM corresponds to the model for a Randomised Block Design (RBD).

By re-writing models (2.4) and (2.5) in matrix notation we obtain

$$\begin{aligned} &\text{CRM:} \quad && \mathbb{E}\left[\boldsymbol{y}\right] = \mu\mathbf{1} + X_\tau\boldsymbol{\tau} = \left(\mathbf{1} \quad X_\tau\right)\left(\mu \quad \boldsymbol{\tau}^T\right)^T; \\ &\text{RBM:} \quad && \mathbb{E}\left[\boldsymbol{y}\right] = \mu\mathbf{1} + X_\tau\boldsymbol{\tau} + X_b\boldsymbol{b} = \left(\mathbf{1} \quad X_\tau \quad X_b\right)\left(\mu \quad \boldsymbol{\tau}^T \quad \boldsymbol{b}^T\right)^T. \end{aligned}$$

where X_τ and X_b are the incidence matrices for treatment and block effects respectively with $X_\tau = (\boldsymbol{u}_1 \ldots \boldsymbol{u}_m)$ and $X_b = (\boldsymbol{w}_1 \ldots \boldsymbol{w}_\kappa)$. Quoting Bailey (2008, Sec. 2.7), vector $\mathbf{u}_s$ corresponds to an $n \times 1$ vector where for each of the treatments $(s = 1, 2, \ldots, m)$ all the elements of the vector equal zero, except for those that correspond to the units which receive that treatment and are equal to one. For instance $\mathbf{u}_1$ is the indicator vector with ones for the unit(s) receiving treatment $s = 1$ and zeros elsewhere. Likewise the vector $\mathbf{w}_i$ corresponds to an $n \times 1$ vector where for each one of the blocks $(i = 1, 2, \ldots, \kappa)$, all the elements of the vector equal zero except the j-th (which is equal to one) corresponding to the units $(j = 1, 2, \ldots, n)$ which belong to that block.

We can write the information matrices for models (2.4) and (2.5) respectively as

$$M_1 = \begin{pmatrix} \mathbf{1}^T \\ {X_\tau}^T \end{pmatrix} \begin{pmatrix} \mathbf{1} & X_\tau \end{pmatrix} = \begin{pmatrix} \mathbf{1}^T\mathbf{1} & \mathbf{1}^T X_\tau \\ {X_\tau}^T\mathbf{1} & {X_\tau}^T X_\tau \end{pmatrix}$$

$$= \begin{pmatrix} \mathbf{1}^T\mathbf{1} & \mathbf{1}^T\mathbf{u}_1 & \mathbf{1}^T\mathbf{u}_2 & \dots & \mathbf{1}^T\mathbf{u}_m \\ \mathbf{u}_1^T\mathbf{1} & \mathbf{u}_1^T\mathbf{u}_1 & \mathbf{u}_1^T\mathbf{u}_2 & \dots & \mathbf{u}_1^T\mathbf{u}_m \\ \mathbf{u}_2^T\mathbf{1} & \mathbf{u}_2^T\mathbf{u}_1 & \mathbf{u}_2^T\mathbf{u}_2 & \dots & \mathbf{u}_2^T\mathbf{u}_m \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \mathbf{u}_m^T\mathbf{1} & \mathbf{u}_m^T\mathbf{u}_1 & \mathbf{u}_m^T\mathbf{u}_2 & \dots & \mathbf{u}_m^T\mathbf{u}_m \end{pmatrix} = \begin{pmatrix} n & n_1 & n_2 & \dots & n_m \\ n_1 & n_1 & 0 & \dots & 0 \\ n_2 & 0 & n_2 & \dots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ n_m & 0 & 0 & \dots & n_m \end{pmatrix}$$

and

$$M_2 = \begin{pmatrix} \mathbf{1}^T \\ {X_\tau}^T \\ {X_b}^T \end{pmatrix} \begin{pmatrix} \mathbf{1} & X_\tau & X_b \end{pmatrix} = \begin{pmatrix} \mathbf{1}^T\mathbf{1} & \mathbf{1}^T X_\tau & \mathbf{1}^T X_b \\ {X_\tau}^T\mathbf{1} & {X_\tau}^T X_\tau & {X_\tau}^T X_b \\ {X_b}^T\mathbf{1} & {X_b}^T X_\tau & {X_b}^T X_b \end{pmatrix}$$

$$= \begin{pmatrix} \mathbf{1}^T\mathbf{1} & \mathbf{1}^T\mathbf{u}_1 & \mathbf{1}^T\mathbf{u}_2 & \dots & \mathbf{1}^T\mathbf{u}_m & \mathbf{1}^T\mathbf{w}_1 & \mathbf{1}^T\mathbf{w}_2 & \dots & \mathbf{1}^T\mathbf{w}_\kappa \\ \mathbf{u}_1^T\mathbf{1} & \mathbf{u}_1^T\mathbf{u}_1 & \mathbf{u}_1^T\mathbf{u}_2 & \dots & \mathbf{u}_1^T\mathbf{u}_m & \mathbf{u}_1^T\mathbf{w}_1 & \mathbf{u}_1^T\mathbf{w}_2 & \dots & \mathbf{u}_1^T\mathbf{w}_\kappa \\ \mathbf{u}_2^T\mathbf{1} & \mathbf{u}_2^T\mathbf{u}_1 & \mathbf{u}_2^T\mathbf{u}_2 & \dots & \mathbf{u}_2^T\mathbf{u}_m & \mathbf{u}_2^T\mathbf{w}_1 & \mathbf{u}_2^T\mathbf{w}_2 & \dots & \mathbf{u}_2^T\mathbf{w}_\kappa \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots \\ \mathbf{u}_m^T\mathbf{1} & \mathbf{u}_m^T\mathbf{u}_1 & \mathbf{u}_m^T\mathbf{u}_2 & \dots & \mathbf{u}_m^T\mathbf{u}_m & \mathbf{u}_m^T\mathbf{w}_1 & \mathbf{u}_m^T\mathbf{w}_2 & \dots & \mathbf{u}_m^T\mathbf{w}_\kappa \\ \mathbf{w}_1^T\mathbf{1} & \mathbf{w}_1^T\mathbf{u}_1 & \mathbf{w}_1^T\mathbf{u}_2 & \dots & \mathbf{w}_1^T\mathbf{u}_m & \mathbf{w}_1^T\mathbf{w}_1 & \mathbf{w}_1^T\mathbf{w}_2 & \dots & \mathbf{w}_1^T\mathbf{w}_\kappa \\ \mathbf{w}_2^T\mathbf{1} & \mathbf{w}_2^T\mathbf{u}_1 & \mathbf{w}_2^T\mathbf{u}_2 & \dots & \mathbf{w}_2^T\mathbf{u}_m & \mathbf{w}_2^T\mathbf{w}_1 & \mathbf{w}_2^T\mathbf{w}_2 & \dots & \mathbf{w}_2^T\mathbf{w}_\kappa \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots \\ \mathbf{w}_\kappa^T\mathbf{1} & \mathbf{w}_\kappa^T\mathbf{u}_1 & \mathbf{w}_\kappa^T\mathbf{u}_2 & \dots & \mathbf{w}_\kappa^T\mathbf{u}_m & \mathbf{w}_\kappa^T\mathbf{w}_1 & \mathbf{w}_\kappa^T\mathbf{w}_2 & \dots & \mathbf{w}_\kappa^T\mathbf{w}_\kappa \end{pmatrix}$$

$$= \begin{pmatrix} n & n_1 & n_2 & \dots & n_m & n_{(1)} & n_{(2)} & \dots & n_{(\kappa)} \\ n_1 & n_1 & 0 & \dots & 0 & n_{(1)1} & n_{(2)1} & \dots & n_{(\kappa)1} \\ n_2 & 0 & n_2 & \dots & 0 & n_{(1)2} & n_{(2)2} & \dots & n_{(\kappa)2} \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots \\ n_m & 0 & 0 & \dots & n_m & n_{(1),m} & n_{(2),m} & \dots & n_{(\kappa),m} \\ n_{(1)} & n_{(1)1} & n_{(1)2} & \dots & n_{(1),m} & n_{(1)} & 0 & \dots & 0 \\ n_{(2)} & n_{(2)1} & n_{(2)2} & \dots & n_{(2),m} & 0 & n_{(2)} & \dots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots \\ n_{(\kappa)} & n_{(\kappa)1} & n_{(\kappa)2} & \dots & n_{(\kappa),m} & 0 & 0 & \dots & n_{(\kappa)} \end{pmatrix},$$

where n_s is the number of units given treatment $s \in \{1, 2, \dots, m\}$, $n_{(i)}$ is the number of units in block $i \in \{1, 2, \dots, \kappa\}$ and $n_{(i)s}$ is the number of units in block i receiving treatment s. To conclude the definitions, we assume that every unit in the experiment receives a treatment. Therefore, the sum of the number of units given the different treatments amount to the total number of units, i.e. $\sum_{s=1}^{m} n_s = n$.

Estimability and constraints. The treatment models are overparameterised. To overcome this problem requires imposing some constraints, otherwise the normal equations have an infinite number of solutions and our parameters will not be uniquely estimated. In our examples, without loss of generality, we assume the set-last-to-zero linear constraints with the m-th treatment effect τ_m and κ-th block effect b_κ of the binary indicators to be set equal to zero. As a result the X_τ (likewise X_b) loses the last column so as to be of full rank. Apart from the set-to-zero constraints another commonly used type of linear constraints are the sum-to-zero constraints (Yandell, 1997, Ch.7), e.g. $\sum_{s=1}^{m} \tau_s = 0$ and $\sum_{i=1}^{\kappa} b_i = 0$. Note that the arbitrary assumption of setting the last effect equal to zero or other similar linear constraint does not affect the design, as the interest lies in estimating the treatment effects' differences under ϕ_1 and ϕ_2.

Linear optimality. Focusing on the L-optimal (unblocked) design the interest lies in estimating the linear combinations of the parameters $S^T\boldsymbol{\beta}$ with minimum variance, where S corresponds to multiple $p \times 1$ vectors of known constants (as we define below). In particular, the number of parameters p under this model is $m+1$, that is the length of $\boldsymbol{\beta}$ (corresponding to the column of constants and columns of the m treatment effects).

Definition. Let $\boldsymbol{s}^\star = \boldsymbol{s}^\star(\alpha_1, \alpha_2)$ with $1 \leq \alpha_1 \leq \alpha_2 \leq m+1$. Let $s_{\alpha_1} = 1$, $s_{\alpha_2} = -1$ and $s_i = 0$, for $i \neq \alpha_1, \alpha_2$. Then $\boldsymbol{s}^\star(\alpha_1, \alpha_2) = (s_1, \ldots, s_{m+1})^T$. Let $\boldsymbol{s}(\alpha_1, \alpha_2)$ be the vector, which contains elements s_1 to s_m of $\boldsymbol{s}^\star$ (to account for the constraints).

Algorithmically each vector $\boldsymbol{s} = \boldsymbol{s}(\alpha_1, \alpha_2)$ is formed by considering a vector of zeroes of length $m+1$. We consider the linear function (treatment contrasts) of the model parameters corresponding to the treatment effects. To obtain these differences we replace the zeroes of this vector that correspond to the α_1-th element and α_2-th with 1 and -1 respectively (with α_1 and α_2 corresponding to the summation indices of v and h respectively as shown below to obtain $\boldsymbol{s}^\star$). The $(m+1)$-st element from each vector is then removed to account for the constraint $\tau_m = 0$, for uniquely estimating the treatment effects. Note that we have also removed the $(m+1)$-st column/row of the information matrix. The resulting $m \times 1$ vectors are pre- and post-multiplied by the $m \times m$ matrix M^- and then summed up over all treatment contrasts.

The design criterion to be minimised is thus the $var\left(S^T\hat{\boldsymbol{\beta}}\right)$ which is proportional to $S^T\left\{M(\xi)\right\}^- S$, where $S = (\boldsymbol{s}_1, \boldsymbol{s}_2, \ldots, \boldsymbol{s}_q)^T$ is a $q \times p$ matrix and q corresponds to the number of pairwise differences of treatment effects. We seek to minimise the average variance of all pairwise differences of treatment effects

$$\frac{2}{m(m-1)} \sum_{s=1}^{m-1} \sum_{s'=s+1}^{m} var(\widehat{\tau_s - \tau_{s'}}).$$

This is proportional to

$$\phi = \sum_{v=2}^{m} \sum_{h=v+1}^{m+1} \boldsymbol{s}^T(v,h) M^- \boldsymbol{s}(v,h).$$

We are looking for a design which minimises the value of the optimality criterion, i.e. $\phi^* = \min \phi$. For numerical examples refer to Section 3.2.

As already mentioned in Section 1.1, if all treatment comparisons are of equal importance, and there are no restrictions on the possible replication of any of the treatments then the precision of treatment comparisons will be optimised when each treatment has the same replication (see Mead *et al.*, 2012, Sec. 6.5, for a further discussion of balance). Given n_s units receiving treatment s, we can estimate τ_s independently with variance σ^2/n_s. Thus assuming equal replication of treatments in a CRD such that $n_s = n/m$ for all s, the variance for each pairwise treatment contrast $\tau_s - \tau_{s'}$ with $s \neq s'$ is $var\,(\hat{\tau}_s - \hat{\tau}_{s'}) = \sigma^2/n_s + \sigma^2/n_{s'} = 2\sigma^2/n_s = 2\sigma^2 m/n$.

The treatment models (2.4) and (2.5) are linear in the parameters and are governed by the 'non-interference' (additivity) assumption which states that the observation on one unit is unaffected by the treatment assignment to other units (discussed in Section 1.1). This is crucial for the validity of the statistical analysis and the precision of the estimation of the average treatment difference of the units used in the experiment. Normality of the distribution of random errors, on the other hand, is not strictly required but is suggested when the usual tests, such as significance testing and calculation of confidence intervals are to be performed for making inferences about the means, and which rely on the properties of the normal distribution. For more details on the additivity assumption see Hinkelmann and Kempthorne (2005, Sec. 1.6), who derived the model (2.5), assuming additivity under the randomisation scheme for an incomplete block design (without assuming normality).

The CRD and RBD are probably the simplest of all experimental designs and constitute the foundation of other designs. A RBD, as an extension of a CRD, is suitable when units have a block structure, that is when they can be grouped into blocks of homogeneous units. In Chapter 6 we will see that if the units exhibit heterogeneity in two directions (as perhaps in a field experiment), then a row-column design (with two blocking factors) may be the most suitable. In the following chapters, we provide some evidence that when dealing with connected units the standard randomised designs perform poorly and can result in misleading interpretations (see, for example, Chapters 5 and 6). Thus it is important to question the 'non-interference' assumption in settings where a connectivity structure among units is present (implying spillover effects) and appropriately adapt the statistical model before moving on to the treatment allocation stage.

2.3 Search algorithms for optimal design

Finding the globally optimal design among all possible designs is difficult and most of the time computationally infeasible since the number of possible designs increases substantially with the number of treatments and units (due to the large size of the design space $|\Xi| = m^n$). This computationally challenging problem for large design sizes falls within the class of NP-hard optimisation problems. These are the problems that are at least as hard as any Non-deterministic Polynomial time problem with no known solution (Arora and Barak, 2009).

For example when there are $m = 2$ unstructured treatments and $n = 60$ experimental units, then all the possible choices of those treatments amount to m^n which is approximately 11.5×10^{17}. For the simplest design type with no blocking structure, a completely randomised configuration gives the size of the design space as

$$\frac{n!}{\prod_{s=1}^{m} r_s!},$$

where r_s indicate the replicates of the treatments. When assuming equal replication and that m divides n, the above can be re-written as

$$\frac{n!}{[(n/m)!]^m}.$$

Recall that a balanced design has the property that its treatments appear an equal number of times (see Section 1.1). By assuming a balanced design we decrease our design space. For our example, we will have $r_s = 30$ replicates for each treatment and therefore all the possible combinations in the design space are approximately 1.18×10^{17}. If each computation of the criterion, for instance, took one micro-second, an exhaustive search would require more than 3750 years. However, we should point out that for the models we have assumed so far and the methods employed, the computational time required is much less. This is due to the relative small design space of the problems we consider for exhaustive search, but also due to significant advancements in technology and development of fast software.

In real-life situations, we do not need the absolutely best designs, but only ones that are adequately efficient. In the literature, there exist a number of computationally efficient algorithms for finding *near-optimal* designs in a practical time frame using iterative methods. The main steps involved in the majority of those algorithms are the following: (i) initialisation of the search, e.g. random generation of a non-singular design (i.e. the matrix $S^T M^- S$ must be non-singular to ensure that the parameters in ϕ are estimable); (ii) modification of the current solution, e.g. make exchanges/interchanges in the treatment set/design points; (iii) assessment of new solution, e.g. design is assessed with respect to an objective function. Steps (ii)-(iii) are repeated until no change improves the design value; (iv) termination of the search process and return of the final design (which is assumed to be optimal).

There are many widely-used algorithms for the construction of optimal designs, especially for D-optimal designs (Fedorov, 1972; Fedorov and Hackl, 1997). Comparisons of the performance of some popular algorithms for seeking D-optimal designs is provided by Cook and Nachtsheim (1980), who also introduced a generic simple exchange algorithm which roughly follows the steps we outlined earlier. More specifically, they considered each design point in turn, in a random order, carrying out any beneficial exchange as soon as it was discovered. They named the resulting procedure 'a modified Fedorov exchange'. While the original Federov exchange algorithm only performs the 'best' exchange (Fedorov, 1972), the modified Fedorov (point exchange) algorithm considers exchanges of design points with random points from a candidate list, accepting an exchange at each iteration which leads to an improvement of the objective function. As opposed to the point exchange algorithm, the co-ordinate exchange algorithm is based on the idea of possible exchanges of coordinates within each design point (Meyer and Nachtsheim, 1995). However, the algorithm of Meyer and Nachtsheim (1995) does not require a candidate set of points, making it applicable and effective for 'large' designs (large number of points and variables). Expanding the traditional modified Fedorov and coordinate exchange algorithms, Huang (2016) proposed a new algorithm which incorporates a continuous optimisation method and numerical integration techniques to obtain optimal designs that decrease the computational cost required. Note that in this thesis, we focus on comparing unstructured treatments and therefore there is no difference between point and coordinate exchange.

There exist many other studies for improving the speed of convergence of proposed algorithms. Among others, Nguyen and Miller (1992) suggested that the speed of convergence of these algorithms depends on the starting (initial) designs. For an extensive discussion of different algorithms and several modifications and improvements for speeding up the design search refer to Atkinson *et al.* (2007, Ch.12) and included references. A slightly different algorithmic approach is that of simulated annealing (see, e.g., Kirkpatrick *et al.*, 1983 and Haines, 1987), whose search involves stochastic transitions between designs. A useful review of simulated annealing is given by Spall (2003). The optimisation scheme of the simulated annealing algorithm has been implemented not only for the design search problem but also in various network problems. Examples include colouring graphs (Chams *et al.*, 1987) and community detection (Liu and Liu, 2010). Another stochastic algorithm is the particle swarm optimisation (Kennedy and Eberhart, 2001). This algorithm is inspired by the social behaviour of bird flocking and it searches for the best solution in a population ('swarm') updating a candidate solution ('particle') at every iteration. There is a growing application of the particle swarm optimisation algorithm to generate optimal designs mainly due to the ease of implementation (only a few key tuning parameters required related to the flock size, i.e. number of designs, and number of iterations) without requiring any assumption on the objective function (e.g. Chen *et al.*, 2014, Wong *et al.*, 2015 and Liu *et al.*, 2015).

However, the most commonly used algorithms in the design of experiments are the exchange algorithms, mainly due to the simplicity of their implementation, adaptability for any design criterion and computational efficiency even for large design spaces. We develop a simple ad-hoc exchange algorithm in order to meet the purposes of this thesis which can generate near-optimal designs for the networks at our disposal (see Section 3.6). We seek designs that depend on a chosen statistical model and are optimal with respect to the L-criterion for estimating the direct and indirect treatment effects separately. The advantage of this approach is that the resulting designs are customised to our specific problems, thereby increasing the efficiency of estimation and, as a result, of the inferences drawn.

Computational note. The suggested exchange algorithm is implemented in R statistical computing environment (R Development Core Team, 2017). We use the package 'igraph' (Csárdi and Nepusz, 2006) to generate different networks (graph snapshots) and we explore the optimal properties of designs obtained on them. igraph is an open source C library for the analysis of large-scale complex networks, with embedding interfaces for R, Python and Ruby. Moreover, we occasionally implement our exchange algorithm in a parallel fashion, where different chains of designs will be produced resulting from different starting designs. This results in significant runtime speedups in the search space, which enable us to address larger design problems. When evaluation of the designs becomes too computationally intensive due to the size of the design space and algebraic operations such as matrix inversions, we make use of the supercomputing facility Iridis 4, which is the central resource for high performance computing at the University of Southampton (`cmg.soton.ac.uk/iridis`). MATLAB and Mathematica have been also used in sections of this thesis, the former for verifying analytical results using the Symbolic Math Toolbox, the latter for attempting to solve a constraint satisfaction problem (see Section 3.4). Note that a constraint satisfaction problem relies on a combinatorial optimisation algorithm, which assigns values to each variable from a set of variables subject to a list of constraints, with the aim of finding a solution that satisfies every constraint (if one exists).

2.4 Basics of graph theory

Suppose that n experimental units are available for experimentation and that they form a network, which is represented by a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$. The graph $\mathcal{G}$ consists of $n = |\mathcal{V}|$ vertices, where $\mathcal{V} = \{v_1, \ldots, v_n\}$ and $l = |\mathcal{E}|$ edges, where $\mathcal{E} \subset V \times V$. The vertices may represent individuals or organisations connected through edges which may represent some form of connection such as friendship, collaboration, or communication. The *adjacency matrix* of a graph (also known as a connectivity matrix) is an $n \times n$ matrix $A = [A_{jh}]_{j,h \in V}$ which is a compact way to represent the collection of edges (connections). We focus on undirected and unweighted graphs with $A_{jh} = A_{hj} \in \{0, 1\}$ representing the presence of an edge between the vertices j and h (mutual connections). Thus the resulting adjacency matrix A is symmetric and binary consisting solely of zeros and ones. However, in Chapter 6, we also consider directed networks, where the edges have a direction and therefore $A_{jh} \neq A_{hj}$ as well as weighted networks, where a number (weight) is assigned to each edge. In an adjacency matrix the diagonal elements represent the self-links, which in this thesis are redundant. By convention we set these values to zero, i.e. $A_{jj} = 0$. We should also note that the identity of every unit in the network has been fixed by assigning a unique label to each one (given that a labelling is arbitrary and every choice leads to an equivalent description of the same network).

In reality the specification of this adjacency matrix may not be straightforward and this matrix will likely be a proxy for the actual dynamics and influences that take place between the units. The structure of the interaction network describing who is interacting with whom, how frequently and with which intensity is quite difficult to define or obtain. Different methods have been devised over the years including network proxies such as neighbourhood indicators or spatial distances, historical or hypothetical interactions. There is a large literature on the specification of the adjacency matrix: in the spatial context (Cliff and Ord, 1981; Anselin, 1988); in the social network context (Leenders, 1995, 2002); but also in other related fields, where the adjacency matrix specification is a consequence of a specific similarity matrix based on a similarity measure or function (Von Luxburg, 2007). The latter can prove especially helpful for multivariate and clustering problems, which can be reformulated using a similarity graph. Since many real networks are not static but evolving, with new nodes entering dynamically and establishing connections to already existing nodes, there can also be time varying adjacency matrices. Models for generating graphs or sequences of evolving graphs have also been introduced, some of which we discuss in this section. In this thesis, we assume that the adjacency matrix is pre-specified (non-stochastic) and corresponds to a snapshot of observed interactions or connections which reflect the experimental setup and experimental objectives. In the context of this thesis we occasionaly come back to this subject when required.

Some basic notions in graph theory are *walks* and *paths.* A walk is a sequence of links (edges) connected to a sequence of vertices, where vertices can appear more than once, while a path is a walk without the repetition of vertices (i.e. each vertex in the sequence is distinct). In general a walk can be a longer sequence than a path, since in the path the vertices appear only once. The special case where a walk begins and ends at the same vertex, is called a *closed walk.* If all vertices are distinct but the first vertex coincides with the last one (end vertices are the same) then this is a 'closed path', which is called a *cycle.* A graph that contains no cycles is called *acyclic.* A path length is the number of edges on the path. The *shortest path* (or *geodesic path*) between two vertices, as the name implies, is the shortest sequence of edges between them in the (undirected) graph. For a more detailed description of these network concepts refer to the books of Wasserman and Faust (1994) and Newman (2003). An undirected graph is said to be *connected* if every pair of vertices in the network are connected by some path in the network.

In an undirected network, the *degree*, d_j, of the vertex $v_j \in \mathcal{V}$ denotes the number of vertices that the vertex v_j is connected to (number of neighbours) or in other words is the number of edges (in $\mathcal{E}$) of v_j. It is mathematically defined as $d_j = \sum_{h=1}^{n} A_{jh} = \sum_{h=1}^{n} A_{hj}$. The square of an adjacency matrix A^2 has the property that each of its elements represents the number of two-walks (i.e. walks of length two or walks with two edges) from node j to node h. Thus $\sum_{q=1}^{n} A_{jq}A_{qh}$, with one being added to the sum only when A_{jq} and A_{qj} are both 1. That is, when the edges $v_j v_q$ and $v_q v_k$ are in $\mathcal{G}$, which corresponds to the two-walk from v_j to v_k through v_q.

Measures of network topology

Different networks have different general characteristics that define their connectivity structure and their behaviour as a whole. According to Barabási and Albert (2002) there are three prominent robust measures of a network's topology: degree distribution (or average degree), clustering coefficient, and average path length. We provide below the basic definitions of these measures. However, further discussion will follow in the coming chapters.

Degree distribution and average degree. A network $\mathcal{G}$ is characterised by its degree distribution function $P(d)$, which is specified by the individual degrees of its vertices. In particular, given a graph $\mathcal{G}$, we define $P(d)$ to be the fraction of vertices $v_j \in V$ with degree $d_j = d$. Equivalently, $P(d)$ gives the probability that a randomly selected vertex has degree d (exactly d edges). Thus the collection $\{P(d)\}_{d \geq 0}$ is called the degree distribution of $\mathcal{G}$ and is simply a rescaling of the set of degree frequencies, formed from the original degree sequence. A plot of $\{P(d)\}_{d \geq 0}$ for any given network can be formed by plotting a barchart of the degrees of vertices. We can also define the average degree of the vertices as

$$\delta = \frac{1}{n} \sum_{j=1}^{n} d_j. \tag{2.6}$$

Clustering coefficient of a network. A common attribute in many graphs is that the neighbours of a vertex are connected to each other. In social networks, for example, there is a tendency for one's friends to become friends with each other forming cliques. This is captured by the clustering coefficient, proposed by Watts and Strogatz (1998), which reflects the topological connections among a vertex's neighbours. The *clustering coefficient*, $\mathcal{C}_j$ of vertex v_j, is the number of edges around vertex j and is calculated as the ratio of the number E_j of edges that actually exist among the neighbours of the same vertex v_j over the total number of possible edges of these neighbours (if they are themselves all connected). The clustering coefficient, $\mathcal{C}$, of the entire network is the average of $\mathcal{C}_j$ over all n vertices, that is

$$\mathcal{C} = \frac{1}{n} \sum_{j=1}^{n} \mathcal{C}_j = \frac{1}{n} \sum_{j=1}^{n} \frac{E_j}{\frac{1}{2} d_j (d_j - 1)}, \tag{2.7}$$

where d_j is the degree of vertex v_j. The value of $\mathcal{C}$ can range between $\mathcal{C} = 1$ when the network is fully connected ($\delta = n - 1$) and $\mathcal{C} = 0$ when the network is empty ($\delta = 0$). More details on the definition can be found in Barabási and Albert (2002, p.49) and Caldarelli (2007, p.16), while an extensive discussion and an alternative definition is provided by Newman *et al.* (2006, Ch.4).

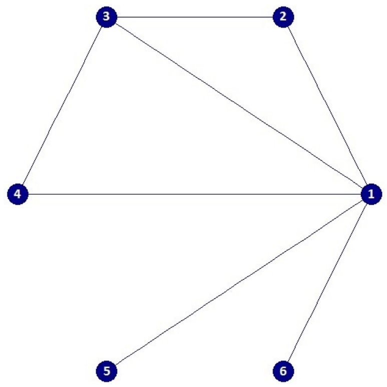
Average path length. This is the average distance between two vertices in the network across all pairs of distinct vertices. The distance, $\ell(u, v)$, is the length of the shortest path connecting the vertices u and v. This path does not necessarily need to be unique. Recall that path refers to an alternating sequence of edges over all pairs of distinct vertices connecting any two vertices, while path length is the number of these edges along the path. We can calculate the distance $\ell(u, v)$ from the adjacency matrix A as the smallest power p of A, such that the (u, v)-element of A^p is different from zero. Thus in a connected graph, the average path length, denoted as ℓ, is given as

$$\ell = \frac{1}{n(n-1)} \sum_{u \neq v \in \mathcal{V}} \ell(u, v). \tag{2.8}$$

Thus, this quantity provides the average number of required links to traverse along the shortest path connecting two vertices, which in most real life networks increases at most logarithmically with the number n of vertices (Garlaschelli *et al.*, 2010). Therefore, it describes how 'globally connected' a graph is. The idea of this measure was conceived by Milgram (1967) who discovered in his study that there exist short paths through social networks between apparently distant individuals, a discovery made by the distribution of letters that were passed on from person to person in an attempt to reach a desired target individual. This is also known as the 'small-world phenomenon' (see Watts and Strogatz (1998) and references within).

Example 2.4.1. Figure 2.1 depicts a small network of 6 vertices and 7 edges together with its adjacency matrix. We can calculate the average degree δ, the average path length ℓ and the clustering coefficient $\mathcal{C}$. The degrees are $d_1 = 5$, $d_3 = 3$, $d_2 = d_4 = 2$, $d_5 = d_6 = 1$ with average degree $\delta = 2.3$. By counting the number of vertices according to their degree we form the degree distribution $P(d)$. Figure 2.2 shows a barchart of the distribution of vertex degrees of the given network.

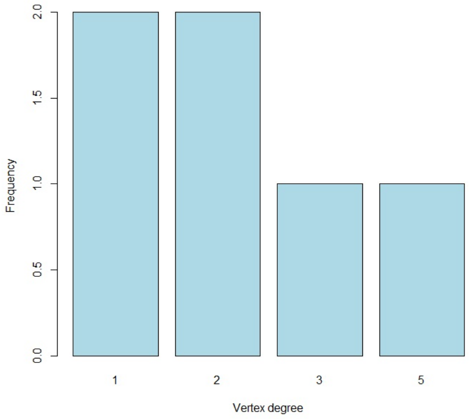
Figure 2.1: A small social network and its adjacency matrix



$$A = \begin{pmatrix} 0 & 1 & 1 & 1 & 1 & 1 \\ 1 & 0 & 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}$$

The most highly connected vertex corresponds to vertex 1 in the network. The typical separation between any two vertices in the graph (average path length) is $\ell = 1.53$. The clustering coefficient of the network is 0.40. To understand the calculation of this measure we can calculate the clustering coefficient for vertex 1. That vertex has five neighbours and the fully connected neighbourhood is composed of 10 $(= 5(5-1)/2)$ mutual pairs. Two out of ten possible pairs of these neighbours are connected with each

Figure 2.2: Degree distribution of the network



other (i.e. vertices 2 with 3 and 3 with 4). Thus the resulting clustering coefficient for this vertex is 0.2. Then the clustering coefficient for the entire network is the average quantity over all six vertices.

Spectral clustering

A graph $\mathcal{G}$ can be represented by different connectivity matrices. Apart from the adjacency matrix we consider Laplacian matrices, which are the basic tools for spectral clustering. The area that makes use of those matrices is most commonly known as spectral graph theory. Spectral clustering, as the name implies, makes use of the graph spectrum, i.e. the the set of eigenvalues and eigenvectors, to group vertices into κ clusters. The set of eigenvalues plays a major role in the understanding of the structure and dynamics occurring in the network since they are linked with numerous graph invariants (i.e. graph properties that remain unchanged under transformations). In order to implement spectral clustering one has to calculate the first κ eigenvectors (corresponding to the κ smallest eigenvalues) of a specific graph Laplacian that best meets the objectives of the research (Chung, 1997).

To begin with, we define the standard Laplacian, namely the *unnormalised graph Laplacian* (without loops and multiple edges) as $L = D - A$, where D is the $n \times n$ diagonal degree matrix (where the entries are the degrees of all vertices, i.e. $d_1, \dots, d_n$ on the diagonal) and A the adjacency matrix of the same graph. In other words the (j, h)-th entry of the symmetric unnormalised Laplacian matrix is:

$$L = [L]_{jh} = \begin{cases} d_j, & j = h. \\ -A_{jh}, & j \neq h. \end{cases}$$

Some important properties (as mentioned in Von Luxburg, 2007) of the unnormalised Laplacian L are:

(i) it is a positive semi-definite matrix ($\forall i$, the eigenvalues $\lambda_i \geq 0$);

(ii) it has non-negative and real eigenvalues $0 = \lambda_1 \leq ... \leq \lambda_n$, where $n = |\mathcal{V}|$, (i.e. eigenvalues are in ascending order where the smallest value is 0); and

(iii) the multiplicity of the zero eigenvalue equals the number of connected components in the graph (in particular $\lambda_2 > 0$ if and only if the graph is connected; λ_2 is called the *algebraic connectivity* of the network).

We focus on connected graphs. Therefore, there is only one zero eigenvalue corresponding to a constant eigenvector (i.e. the second smallest eigenvalue is different from zero). The literature also makes mention of two normalised graph Laplacians, namely

$$\begin{aligned} L_{sym} &= D^{-1/2} L D^{-1/2}; \\ L_{rw} &= D^{-1} L. \end{aligned}$$

The subscripts denote that the first normalised graph Laplacian (L_{sym}) is symmetric and the second (L_{rw}) is a non-symmetric matrix which is related to a random walk on $\mathcal{G}$. A random walk is, as the name implies, a walk across a network created by taking repeated random steps. Starting at some specified vertex, it chooses uniformly at random the edges to go along in order to move on to a random neighbour attached to the current vertex and repeats this process. It can visit vertices and go along edges more than once. Since $L_{rw} = D^{-1}L = D^{-1}(D - A) = I - D^{-1}A$, where $D^{-1}A$ is in fact the transition matrix of a standard random walk on the given graph, it is often used to study the diffusion process on that graph. Moreover, the vertices with high degree are more likely to be visited by the random walk because there are more ways of reaching them. It is apparent that L_{rw} is a weighted version of L. However, the eigenvalues of the normalised graph Laplacian matrices are scale-independent ranging from 0 to 2.

Revisiting Example 2.4.1, we can calculate the degree matrix D and the different Laplacians.

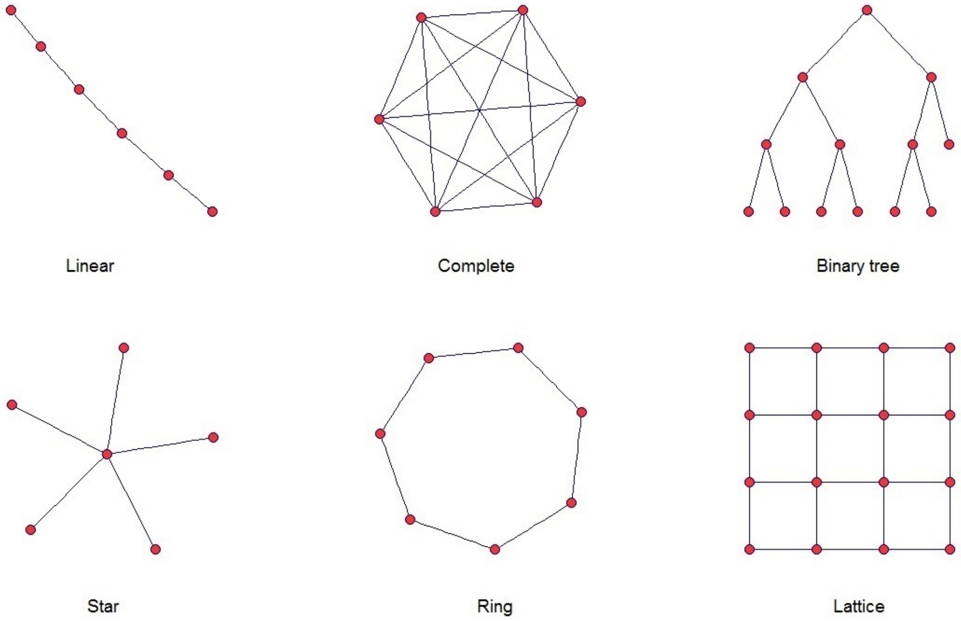
$$D = \begin{pmatrix} 5 & 0 & 0 & 0 & 0 & 0 \\ 0 & 2 & 0 & 0 & 0 & 0 \\ 0 & 0 & 3 & 0 & 0 & 0 \\ 0 & 0 & 0 & 2 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix} \quad L_{rw} = \begin{pmatrix} 1 & -0.2 & -0.2 & -0.2 & -0.2 & -0.2 \\ -0.5 & 1 & -0.5 & 0 & 0 & 0 \\ -0.33 & -0.33 & 1 & -0.33 & 0 & 0 \\ -0.5 & 0 & -0.5 & 1 & 0 & 0 \\ -1 & 0 & 0 & 0 & 1 & 0 \\ -1 & 0 & 0 & 0 & 0 & 1 \end{pmatrix}$$

$$L = \begin{pmatrix} 5 & -1 & -1 & -1 & -1 & -1 \\ -1 & 2 & -1 & 0 & 0 & 0 \\ -1 & -1 & 3 & -1 & 0 & 0 \\ -1 & 0 & -1 & 2 & 0 & 0 \\ -1 & 0 & 0 & 0 & 1 & 0 \\ -1 & 0 & 0 & 0 & 0 & 1 \end{pmatrix} \quad L_{sym} = \begin{pmatrix} 1 & -0.32 & -0.26 & -0.32 & -0.45 & -0.45 \\ -0.32 & 1 & -0.41 & 0 & 0 & 0 \\ -0.26 & -0.41 & 1 & -0.41 & 0 & 0 \\ -0.32 & 0 & -0.41 & 1 & 0 & 0 \\ -0.45 & 0 & 0 & 0 & 1 & 0 \\ -0.45 & 0 & 0 & 0 & 0 & 1 \end{pmatrix}$$

Special graphs

Graphs come in all shapes and sizes but there are certain types of graphs that we more often encounter in practice, especially as parts of bigger graphs. The connectivity structure of these special graphs follows a common pattern (as opposed to a random graph, which is generated by a random process or probabilistic rules). Examples of such graphs include: linear, complete, tree, star, ring and lattice graphs (see Figure 2.3).

Figure 2.3: Special types of graphs



Linear

Complete

Binary tree

Star

Ring

Lattice

The *linear graph* is the simplest form of graph topology, where all vertices are connected with a single line. A *complete graph* (or clique) is a fully connected graph, i.e. there is an edge between every pair of vertices. The number of connections is $n(n-1)/2$ corresponding to all possible choices of pairs of vertices. In a *tree graph* any two vertices are connected by exactly one path and are arranged in the form of a tree, having a branching structure leading from branch vertices (i.e. vertices of degree at least two) to leaves (i.e. vertices of degree one). Trees can be drawn such that one vertex is at the top, with all edges going down to subsequent vertices. Figure 2.3 illustrates a binary tree, where each branch vertex has at most two edges. A connected graph of n vertices is a tree if and only if it has $n-1$ edges. A *star graph* is a special tree graph, where every vertex is connected to a central vertex (only one branch vertex) also called the hub. The *ring graph* consists of a single cycle. In a ring graph the number of vertices equal the number of edges and every vertex has exactly two edges connected to it (i.e. every vertex has degree two). A q*-dimensional lattice* is a graph in which the vertices can be placed at the integer coordinate points of the q-dimensional Euclidean space, and each vertex connects to vertices which are exactly one unit away from it. In the figure the lattice is two-dimensional with length 4 along each dimension. In this type of graph two vertices are connected if the difference of one of their coordinates is plus or minus one and all their other coordinates are exactly the same.

Three types of networks as generated by graph models

Graph models are mathematical rules which generate either a single graph (graph instance or a graph snapshot), or an entire graph sequence. The most popular graph models are the Erdös-Rényi (random graph) model, the Watts-Strogatz (small-world) model and Barabási-Albert (preferential attachment) model. These models are used as reference models in network theory, because they possess characteristic and distinguishable properties. Extensive introductory overviews of these networks are provided in the papers by Barabási and Albert (2002) and Newman (2003), with many references to research papers providing further considerations and technical details.

For illustrative purposes, we generate graphs of size $n = 200$ by each one of the different mentioned network formation models as illustrated in Figure 2.4. Additionally, Table 2.1 indicates the number of vertices and edges, the average degree, the average path length, and the clustering coefficient for each network. We obtain two barcharts corresponding to the degree distribution of each graph on two scales for summarising each graph's behaviour (i.e. Figures 2.5, 2.7 and 2.8 corresponding to each network). Given the nature of a distribution, sometimes it is more effective to present the degree information on log-log scale. This is especially useful for distinguishing a scale-free network. Figure 2.8, for instance, illustrates a linear decay in the log-frequency as a function of log-degree, which as we will see tend to a power law form.

Figure 2.4: Popular types of networks: random, small-world and scale-free

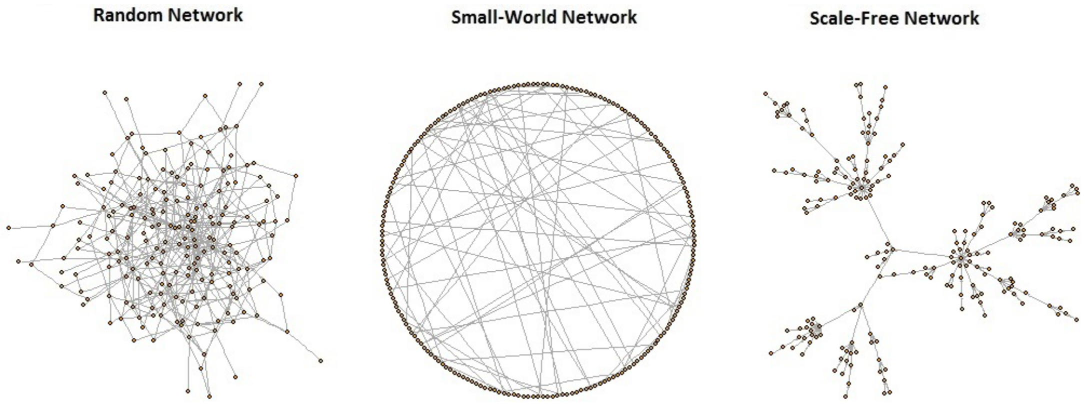
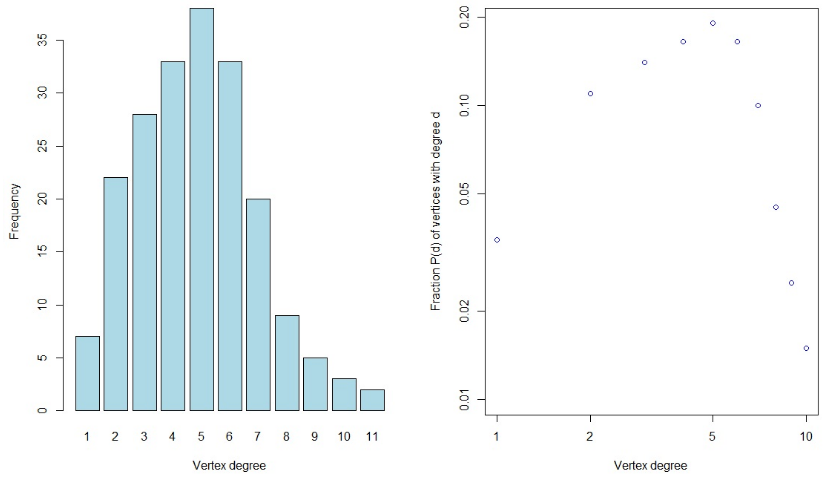


Table 2.1: The general characteristics (number of vertices, number of edges, average degree, clustering coefficient and average path length) of the networks of Figure 2.4

Network	$\|V\|$	$\|E\|$	δ	$\mathcal{C}$	ℓ
Random	200	391	3.998	0.016	3.910
Small-world	200	800	8	0.478	3.737
Scale-free	200	199	1.990	0	5.521

Random Erdös-Rényi graphs (Erdös and Rényi, 1959) are generated by a random process, in which a set of edges are added at random between pairs of vertices belonging to the network. In particular every pair of nodes is chosen randomly and independently from $n(n-1)/2$ possible configurations and is connected with equal probability p. Thus the expected number of edges under the ER random graph model is $\binom{n}{2}p$ and the distribution of the number of edges follows a binomial distribution. The measurements on real networks are usually compared against those on 'random networks'. Random graphs have short average paths and low clustering.

Figure 2.5: Degree distribution of the random network on two scales



Small-world networks (Watts and Strogatz, 1998) have a higher clustering and almost the same average path as random networks of the same number of vertices and edges. As their name implies, almost every unit in the network is close in proximity to almost every other unit. This model is based on the idea of 'rewiring' a fraction of edges within a regular lattice (most frequently a ring). Consider a ring where each vertex is initially linked to its k closest neighbours. An edge is rewired when that edge is disconnected from some of the attached vertices and then randomly connected to an other vertex anywhere in the network. A parameter which needs to be specified is the rewiring probability, where each edge is chosen to be rewired to the graph as a random edge with probability p. A transition from order to randomness, i.e. regular lattice to (an almost) random graph, is observed when we vary p from 0 to 1 (see Figure 2.6).

Figure 2.6: Transition from a ring of $n = 20$ units with $k = 4$ neighbours to a random network via small-world network (cited from Watts and Strogatz, 1998, Figure 1, p.441)

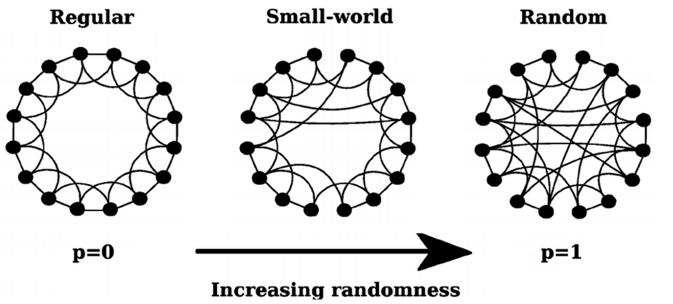
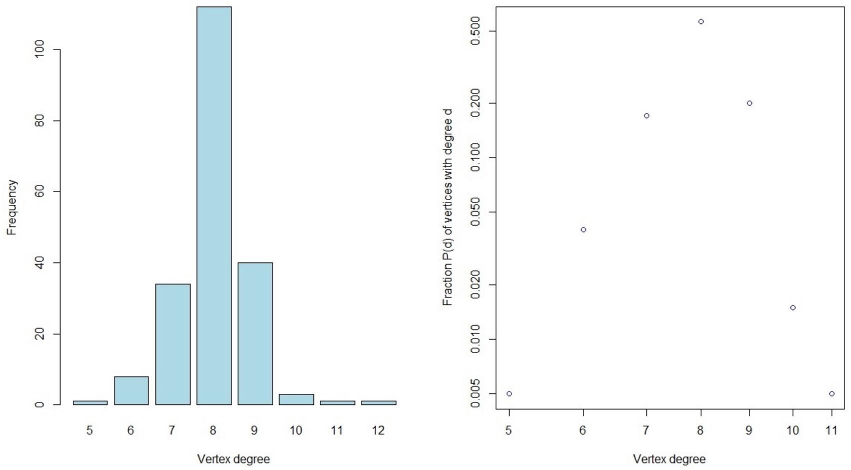
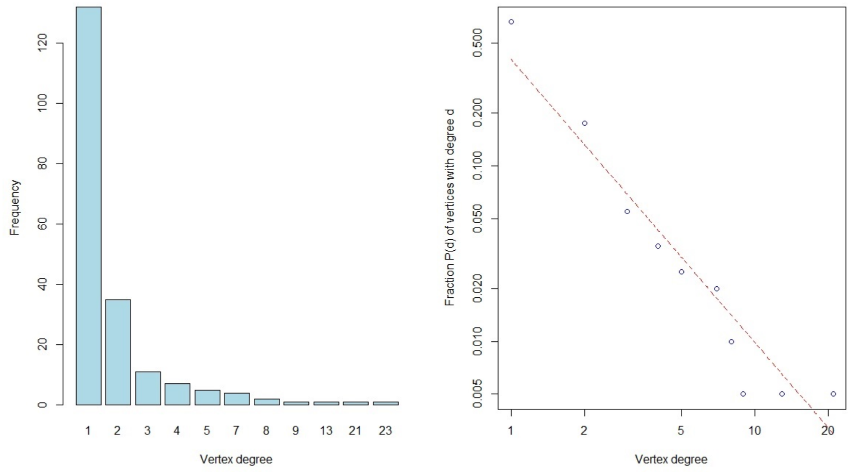


Figure 2.7: Degree distribution of the small-world network on two scales



Scale-free networks are characterised by a highly heterogeneous degree distribution and they are generated by the Barabási-Albert preferential attachment model (Barabási and Albert, 1999). In this model, an edge is most likely to attach to vertices with higher degrees (preferential attachment). The shape of the distribution of a scale-free network, as the name implies, is independent of the level of detail: most vertices have only a small number of neighbours (small degree), but there are some vertices with very high degree (hubs). This distribution function roughly follows a 'power-law' degree distribution, $P(d) \propto d^{-\gamma}$, where the fraction of vertices of degree d is proportional to $d^{-\gamma}$ for some exponent $\gamma > 1$ which determines the 'fatness' of the tail of the curve (Caldarelli, 2007). Figure 2.8 illustrates the degree distribution of the scale-free network of Figure 2.4. The fitted red line represents the fitted power law distribution with $\gamma = 1.612$ (Kolmogov-Smirnov test $R^2 = 0.921$). As we can discern the distribution is right-skewed with a heavy-tail, which implies that the network is heterogeneous as opposed to a random graph, where most vertices have approximately the same degree that is close to the average degree.

Figure 2.8: Degree distribution of the scale-free network on two scales



2.5 Discussion

Based on minimal assumptions there is an elegant general theory for balanced designs with linear analysis that is based on an assumption of 'non-interference'. This is the assumption that treatment effects are additive and a randomisation scheme justifies the validity of experimental analysis. However, randomisation methods cannot effectively capture network heterogeneity as we will see in the following chapters. An optimal design approach can be proved more effective in generating efficient designs that account for network effects. It is of importance to note at this point that the model, on which the design will be based, requires some assumptions to be made about the nature of the spillover effects and their structure. In a network setting, where there is network interference, the response of interest measured on a particular unit may depend either on interventions assigned on neighbouring connected units or on the neighbours' responses. The following chapters will investigate these issues separately, by modelling interference and optimising the design based on the corresponding model. Ideally, we seek designs which are robust to model and network misspecifications. Careful planning of an experiment reduces the experimental burden and the required resources and provides useful and reliable data compared to unplanned experiments.

Chapter 3

Optimal designs with network effects

As earlier discussed in Chapter 1 there are various approaches for accounting for the interference among different units. Some of the early ideas have been employed in agricultural experiments, where the potential interference effects among plots have been taken into account in the formulation of the statistical model and in the development of the experimental design. This chapter starts by presenting some of the models considered in the design field with neighbouring units but also in the network field under the presence of social influence (Section 3.1). Section 3.2 focuses on the linear network effects model, which models interference by means of the (observed) network structure among the units (Parker *et al.*, 2016). Subsequently, Section 3.3 provides the analytical expressions of the optimality criteria as a function of the information matrix and the derivation of the design bias in the estimators of the model parameters due to model misspecification. Some patterns of the optimally allocated treatments, as discovered by means of different example designs, are illustrated in Section 3.4. Additional example designs are computed by an exhaustive search in Section 3.5, while other issues associated with the design efficiency and bias are also explored. Section 3.6 proposes a simple exchange algorithm for constructing L-optimal designs, when exhaustive search is computationally prohibitive. Note that the notation introduced in this chapter is used throughout this thesis, and certain elements of the methodology are restated, or extended in subsequent chapters.

3.1 Background: modelling interference

When we suspect that interference effects may be present among the experimental units, it may be possible to take them into account in the design and analysis by including additional terms in the response model. In this section we provide some of the most important interference models considered in various fields (some discussion was already provided in Section 1.3). The model notation has been streamlined for comparative purposes.

In cross-over designs, commonly used in medical research, each subject enrolled in the trial receives some treatments over consecutive periods of time. The treatments exhibit effects beyond the period in which they are applied; these are the spillover effects which in this setting are mostly known under the names residual or carryover effects. The common assumption in these designs is that carryover effects last only for one period. A simple linear model summarising this idea, as provided by Atkinson *et al.* (2007, Ch. 25), is

$$y_{ij} = \mu + \tau_{r(i,j)} + \lambda_{r(i-1,j)} + \pi_i + s_j + \epsilon_{ij}, \tag{3.1}$$

where $i = 1, 2, \ldots, p; j = 1, 2, \ldots, n$, y_{ij} is the response from subject j in period i, μ is the overall mean, $r(i, j)$ is the treatment applied to subject j in period i, $\tau_{r(i,j)}$ is the direct effect of the treatment $r(i, j)$, $\lambda_{r(i-1,j)}$ is the carry-over effect of treatment $r(i-1, j)$ applied in period $i-1$ for subject j with $\lambda_{r(0,j)} = 0$, π_i is the effect of period i, s_j is the effect of subject j and ϵ_{ij} are independent and identically normally distributed errors, with expectation 0 and unknown variance σ^2. This model also applies in sensory evaluation and animal experiments. Other appropriate models describing this framework can be found in Jones and Donev (1996) and Jones and Kenward (2003).

Apart from cross-over trials, spillover effects have been extensively studied and modelled in the agricultural field with the aim of estimating interference effects. Pearce (1957) considered a model incorporating both the direct and neighbour effects in a block design. The cause of interference is associated with the average treatment value of the nearest neighbouring treatments, i.e. treatment interference. Based on Besag and Kempton (1986), the general model formulation (in matrix form) is

$$\boldsymbol{y} = X_b\boldsymbol{b} + X_\tau\boldsymbol{\tau} + WX_\tau\boldsymbol{\gamma} + \boldsymbol{\epsilon}, \tag{3.2}$$

where $\boldsymbol{b}$ is a vector of block effects with incidence matrix X_b, $\boldsymbol{\tau}$ is a vector of centred treatment effects (about their means) with design matrix X_τ. The effects of $\boldsymbol{\tau}$ can be interpreted as centred mean responses in the absence of neighbour effects. The 'neighbour incidence matrix', W, is of dimension $n \times n$ composed of 0 and 1 for the neighbours. As earlier pointed out in Section 1.3, Pearce (1957) considered all plots from the same block to be neighbours, irrespectively of whether they are adjacent or not. The difference from the adjacency matrix defined Section 2.4, is that W equals one for any plot belonging to the same block as the plot under investigation. The neighbour effects are the centred indirect treatment effects produced by neighbours and are represented by the vector $\boldsymbol{\gamma}$. Moreover, according to Pearce (1957), it is assumed that the 'remote' effects (i.e. neighbour effects) do not influence other blocks and that the remote effect is the same on all other plots of the block. The errors are denoted by $\boldsymbol{\epsilon}$ and in a certain setting can also have an autoregressive structure capturing potential spatial variation. We should note that the neighbour effect is outgoing for this model, in the sense that each treatment has an effect on the plot to which it is applied and a neighbour effect on each neighbouring plot in the same block. In this thesis we are mostly interested in the incoming effect, that is the total network effect from all the immediate connected neighbours in any direction (see following section).

Draper and Guttman (1980) explored a special case of the model postulated by Pearce (1957), with a common coefficient of interference, ρ, for all treatments. The model as presented by Besag and Kempton (1986) is

$$\boldsymbol{y} = X_b\boldsymbol{b} + G\boldsymbol{\tau} + \boldsymbol{\epsilon}, \tag{3.3}$$

where $G = (I + \rho W)X_\tau$.

Another model also mentioned in the paper of Besag and Kempton (1986) is that of Kempton (1982), where the cause of interference in an array of plots is attributed to the responses of the two neighbouring plots (left and right).

$$\boldsymbol{y} = X_b\boldsymbol{b} + X_\tau\boldsymbol{\tau} + \rho W\boldsymbol{y} + \boldsymbol{\epsilon}, \tag{3.4}$$

where W is a weight matrix that has off-diagonal elements 1/2 $(j, j \pm 1)$ or otherwise 0. Similarly to the previous model, ρ is the nearest-neighbour 'competition' coefficient, which is common to all treatments. Thus, the product $W\boldsymbol{y}$ corresponds to the mean of the observed responses of the two neighbouring plots considered. The model assumes observations adjusted for the general mean. We can re-write this model as

$$y_{ij} = b_i + \tau_{r(ij)} + \frac{1}{2}\rho\left(y_{r(i,j-1)} + y_{r(i,j+1)}\right) + \epsilon_{ij}, \tag{3.5}$$

where $i = 1, 2, \ldots, \kappa; j = 1, 2, \ldots, n_{(i)}$, y_{ij} is the response from unit j in block i and $r(ij)$ indicates the treatment applied to subject j in block i, $\tau_{r(ij)}$ is the centred direct effect of treatment $r(ij)$.

This model can be extended to include effects from q > 2 neighbouring plots by averaging across all the observed responses of the neighbouring plots considered, i.e. $\sum_{j'=1}^{\mathrm{q}} y_{r(ij')}/\mathrm{q}$ where $y_{r(ij')}$ is the observed value given the treatment $r(ij')$ assigned to plot j' (neighbour to plot j) belonging to block i and q is the number of neighbouring plots considered. For instance, when q corresponds to the number of plots in the same block i as plot j this will be similar to the idea of Pearce's model but focusing on the neighbouring responses rather than the neighouring effects.

A model used for designing experiments with interfering units is that of Kunert and Martin (2000),

$$y_{ij} = \mu + b_i + \tau_{r(i,j)} + \lambda_{r(i,j-1)} + \theta_{r(i,j+1)} + \epsilon_{ij}, \tag{3.6}$$

where $i = 1, 2, \ldots, \kappa; j = 1, 2, \ldots, n$, y_{ij} is the response from plot (i, j) in the j-th position of block i, μ is the overall mean, b_i is the effect of block i, $r(i, j) \in \{1, \ldots, m\}$ is the treatment applied to the plot (i, j), $\tau_{r(i,j)}$ is the direct effect of the treatment $r(i, j)$, and $\lambda_{r(i,j-1)}$ and $\theta_{r(i,j+1)}$ are the left and right neighbour effects of the treatments assigned to the left $(i, j - 1)$ and right $(i, j + 1)$ adjacent plots respectively and ϵ_{ij} is the random error, which is assumed to be independent and identically normally distributed (i.i.d) with expectation 0 and constant variance, i.e. $\epsilon_{ij} \sim N(0, \sigma^2)$. The set of neighbours is restricted, since there are no border plots, and therefore the terms $\lambda_{r(i,j-1)}$ and $\theta_{r(i,j+1)}$ should appear if and only if $j \neq 1$ and $j \neq \kappa$. Thus $\lambda_{r(i,0)} = \theta_{r(i,\kappa+1)} = 0$, where κ is the block size.

The interference in the model postulated by Kunert and Martin (2000) is restricted to two-directional adjacent plots. Recent work which extends this setting is by Parker *et al.* (2016), who considered the linear network effects model (LNM). Their modelling approach adopts the viewpoint of Pearce (1957) as presented in the paper of Besag and Kempton (1986) (see model (3.2)), but in a multi-directional setting without the inclusion of blocks and with incoming neighbour effects. In particular, the LNM is a simplified representation of the propagation of treatment effects in connected units that form a network structure, where the response is a function of the effect of the treatment a unit receives and the effects of the treatments that its (connected) neighbours receive. The detailed mathematical description is presented in Section 3.2 along with further discussion.

Interference has also been modelled in the context of social networks. Network autocorrelation models (Doreian, 1980; Cliff and Ord, 1981; Anselin, 1988) are considered as the predominant models in networks of interdependencies and can be divided into two wide classes: the network effects model (NEM) (also known as spatial effects model or regressive-autoregressive model); and the network disturbance model (NDM) (also known as spatial disturbances or spatial moving average model). These classes consider the autocorrelation of the response or of the error term separately, reflecting the different mechanisms governing social influence (see the relevant literature presented in Section 1.3). An extensive review on this wide class of models is provided by Leenders (2002). These classes of models are formulated (in matrix notation) as:

$$\text{NEM: } \boldsymbol{y} = \rho W \boldsymbol{y} + X\boldsymbol{\beta} + \boldsymbol{\epsilon}, \qquad \boldsymbol{\epsilon} \sim \mathcal{N}\left(\mathbf{0}, \sigma_\epsilon^2 I\right), \tag{3.7}$$

$$\text{NDM: } \boldsymbol{y} = X\boldsymbol{\beta} + \boldsymbol{\epsilon}, \qquad \boldsymbol{\epsilon} = \rho W \boldsymbol{\epsilon} + \boldsymbol{\nu}, \qquad \boldsymbol{\nu} \sim \mathcal{N}\left(\mathbf{0}, \sigma_\nu^2 I\right), \tag{3.8}$$

where $\boldsymbol{y}$ is a $n \times 1$ vector of responses and W is a predefined $n \times n$ weight matrix representing the influence structure among the n individuals (occurring either by means of the autocorrelation of the responses or of the errors/disturbances), whilst ρ is a scalar describing the strength of association between a unit's response and its neighbours' responses. Moreover, X is the $n \times p$ extended design matrix and $\boldsymbol{\epsilon}$ and $\boldsymbol{\nu}$ are the vectors of error terms, which are assumed to be normally distributed with zero means and constant variances. The matrix of weights, W, is highly important in the study of the estimated social influence model since it incorporates the structure of the network with its influence patterns (Leenders, 2002). In both models what is referred to as 'intrinsic opinion' (Leenders, 1995) is measured by the term $X\boldsymbol{\beta}$. The difference between the two models is that the response of a unit in the case of the NEM is given as an aggregate of both its intrinsic opinion and the opinion of its neighbours, while for NDM it is adapted to the change of the responses of its neighbours from their innate opinion.

More specifically, for the NEM, the response of an individual is given by the weighted combination of the responses of its neighbours, where the strength of the influence of each one on that individual is denoted by the specific element of the weighted matrix W (i.e. $y_j = \rho w_{j1} y_1 + \rho w_{j2} y_2 + \ldots + \rho w_{jn} y_n + \epsilon_j$), plus the 'local' (direct) effects. In other words the response of an individual is a weighted function of the responses of its neighbours, represented by the autocorrelation term $\rho W \boldsymbol{y}$, plus the local effects as given by the term $X\boldsymbol{\beta}$. For instance, an individual's behaviour may be influenced by social interaction with its neighbours, but may also be influenced by local effects related to that intrinsic individual behaviour (e.g. income, education or other structural factors which could determine one's behaviour). From the design perspective this response may be caused by the treatment applied. Note that when $\rho = 0$ then the NEM reduces to the standard regression model while when $\boldsymbol{\beta} = \boldsymbol{0}$ it reduces to a standard spatial model. Section 7.2.1 further investigates this topic, with special focus on the NEM, similar to the viewpoint of Kempton (1982) and the model (3.4).

On the other hand, the NDM is a spatial autoregressive model. The response of a unit depends both on the local effects ($X\boldsymbol{\beta}$) and on the deviation of its neighbours' responses from their supposed innate responses, represented by the autocorrelation residuals, i.e. $\boldsymbol{\epsilon} = \rho W \boldsymbol{\epsilon} + \boldsymbol{\nu}$. Thus one's response does not depend on its neighbours' responses but on the change of one's neighbours' responses from their intrinsic opinions. For instance, an individual's behaviour may be adapted to one's neighbours' actions by imitation frequently occurring in situations involving insecurity or risk (Leenders, 2002).

3.2 Designs with linear network effects model (LNM)

An ideal model should account for the relationships between the subjects and be parsimonious at the same time. The linear network effects model (LNM) (Parker *et al.*, 2016) differs from the CRM (2.4) by accounting for network effects from the application of treatments to the connected units. It is defined as

$$\text{LNM: } y_j = \mu + \tau_{r(j)} + \sum_{h=1}^{n} A_{jh} \gamma_{r(h)} + \epsilon_j, \tag{3.9}$$

where $j = 1, 2, \ldots, n$, y_j is the response from unit j, μ is the average response for the whole set of units (overall mean), $r(j) = s \in \{1, 2, \ldots, m\}$ indicates the treatment applied to subject j, $\tau_{r(j)}$ is the direct effect of treatment $r(j)$ applied to unit j, A_{jh} is the adjacency (or connectivity) matrix indicating the presence of mutual connections (in a network structure) between units j and h (see Section 2.4), $\gamma_{r(h)}$ is the network effect of the treatment applied to the connected subject h when there is a connection between units j and h (neighbouring or indirect effect) and ϵ_j are the error terms, which are assumed to be independent with mean 0 and constant variance σ^2. For uniquely estimating the treatment effects we assume that $\tau_m = 0$ (see Section 2.2 for more details). The spillover effects or network effects correspond to the (incoming) effect from the total number of immediate neighbours who receive some treatment. This capacity of the model to measure, to a certain extent, the spillover effects, renders it a useful mathematical tool for experiments, where 'the non-interference assumption' does not hold. The main assumptions underlying this model are the following:

- the response of a unit is dependent on the treatment applied to that unit and on the treatments applied to the immediate neighbouring units, when there is a relationship between the two units.
- the structure among units is known and is captured by the connectivity matrix;
- the model does not account for any influence from external (out-of-network) units;
- all units receive (only) one treatment (note that a treatment could also mean the absence of a treatment).

The generic matrix formation of the linear model is $E(\boldsymbol{y}) = X\boldsymbol{\beta}$, where $\boldsymbol{y}$ is the $n \times 1$ column vector of responses, X is the extended design matrix comprising n rows and p columns (including the column of ones corresponding to the constant) and $\boldsymbol{\beta}$ is a column that contains p unknown regression parameters (see Sections 2.1 and 2.2). By expanding the model (2.4), the expectation of the linear model (3.9), for the case of m treatments, has the following form

$$\begin{aligned} E(\boldsymbol{y}) &= (\mathbf{1} \quad X_\tau{}^\star \quad AX_\tau)\,(\mu \quad \boldsymbol{\tau} \quad \gamma)^T \\ &= (\mathbf{1} \quad \boldsymbol{u_1} \ldots \boldsymbol{u_{m-1}} \quad A\boldsymbol{u_1} \ldots A\boldsymbol{u_m})\,(\mu \quad \tau_1 \ldots \tau_{m-1} \quad \gamma_1 \ldots \gamma_m)^T, \end{aligned}$$

where $\boldsymbol{\beta} = (\mu \quad \boldsymbol{\tau}^T \quad \boldsymbol{\gamma}^T)^T = (\mu \quad \tau_1 \ldots \tau_{m-1} \quad \gamma_1 \ldots \gamma_m)^T$ is the vector parameter. There is no column corresponding to the m-th treatment effect τ_m, since we assumed it to be zero. We can write the Fisher information matrix for LNM (3.9) as

$$M = X^TX = \begin{pmatrix} n & \mathbf{1}^T X_\tau{}^\star & \mathbf{1}^T AX_\tau \\ X_\tau{}^{\star T}\mathbf{1} & X_\tau{}^{\star T} X_\tau{}^\star & X_\tau{}^{\star T} AX_\tau \\ X_\tau^T A\mathbf{1} & X_\tau^T AX_\tau{}^\star & X_\tau^T A^2 X_\tau \end{pmatrix}.$$

The information matrix M is a symmetric $(2m) \times (2m)$ matrix. The first column (or row) corresponds to the mean μ, the following $m-1$ columns (or rows) to the (direct) treatment effects and the final m columns (or rows) to the network effects. By expanding the information matrix, we obtain

$$M = \begin{pmatrix} \mathbf{1}\mathbf{1}^T & \mathbf{1}\mathbf{u}_1^T & \mathbf{1}\mathbf{u}_2^T & \ldots & \mathbf{1}\mathbf{u}_{m-1}^T & \mathbf{1}A\mathbf{u}_1^T & \mathbf{1}A\mathbf{u}_2^T & \ldots & \mathbf{1}A\mathbf{u}_m^T \\ \mathbf{u}_1\mathbf{1}^T & \mathbf{u}_1\mathbf{u}_1^T & \mathbf{u}_1\mathbf{u}_2^T & \ldots & \mathbf{u}_1\mathbf{u}_{m-1}^T & \mathbf{u}_1A\mathbf{u}_1^T & \mathbf{u}_1A\mathbf{u}_2^T & \ldots & \mathbf{u}_1A\mathbf{u}_m^T \\ \mathbf{u}_2\mathbf{1}^T & \mathbf{u}_2\mathbf{u}_1^T & \mathbf{u}_2\mathbf{u}_2^T & \ldots & \mathbf{u}_2\mathbf{u}_{m-1}^T & \mathbf{u}_2A\mathbf{u}_1^T & \mathbf{u}_2A\mathbf{u}_2^T & \ldots & \mathbf{u}_2A\mathbf{u}_m^T \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots \\ \mathbf{u}_{m-1}\mathbf{1}^T & \mathbf{u}_{m-1}\mathbf{u}_1^T & \mathbf{u}_{m-1}\mathbf{u}_2^T & \ldots & \mathbf{u}_{m-1}\mathbf{u}_{m-1}^T & \mathbf{u}_{m-1}A\mathbf{u}_1^T & \mathbf{u}_{m-1}A\mathbf{u}_2^T & \ldots & \mathbf{u}_{m-1}A\mathbf{u}_m^T \\ \mathbf{u}_1A\mathbf{1}^T & \mathbf{u}_1A\mathbf{u}_1^T & \mathbf{u}_1A\mathbf{u}_2^T & \ldots & \mathbf{u}_1A\mathbf{u}_{m-1}^T & \mathbf{u}_1A^2\mathbf{u}_1^T & \mathbf{u}_1A^2\mathbf{u}_2^T & \ldots & \mathbf{u}_1A^2\mathbf{u}_m^T \\ \mathbf{u}_2A\mathbf{1}^T & \mathbf{u}_2A\mathbf{u}_1^T & \mathbf{u}_2A\mathbf{u}_2^T & \ldots & \mathbf{u}_2A\mathbf{u}_{m-1}^T & \mathbf{u}_2A^2\mathbf{u}_1^T & \mathbf{u}_2A^2\mathbf{u}_2^T & \ldots & \mathbf{u}_2A^2\mathbf{u}_m^T \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots \\ \mathbf{u}_{m-1}A\mathbf{1}^T & \mathbf{u}_{m-1}A\mathbf{u}_1^T & \mathbf{u}_{m-1}A\mathbf{u}_2^T & \ldots & \mathbf{u}_{m-1}A\mathbf{u}_{m-1}^T & \mathbf{u}_{m-1}A^2\mathbf{u}_1^T & \mathbf{u}_{m-1}A^2\mathbf{u}_2^T & \ldots & \mathbf{u}_{m-1}A^2\mathbf{u}_m^T \end{pmatrix}$$

$$= \begin{pmatrix} n & n_1 & n_2 & \dots & n_{m-1} & l_1 & l_2 & \dots & l_m \\ n_1 & n_1 & 0 & \dots & 0 & l_{11} & l_{12} & \dots & l_{1m} \\ n_2 & 0 & n_2 & \dots & 0 & l_{21} & l_{22} & \dots & l_{2m} \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots \\ n_{m-1} & 0 & 0 & \dots & n_{m-1} & l_{\{m-1,1\}} & l_{\{m-1,2\}} & \dots & l_{\{m-1,m\}} \\ l_1 & l_{11} & l_{12} & \dots & l_{\{1,m-1\}} & l_{11}^{\langle 2\rangle} & l_{12}^{\langle 2\rangle} & \dots & l_{1m}^{\langle 2\rangle} \\ l_2 & l_{21} & l_{22} & \dots & l_{\{2,m-1\}} & l_{21}^{\langle 2\rangle} & l_{22}^{\langle 2\rangle} & \dots & l_{2m}^{\langle 2\rangle} \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots \\ l_m & l_{m1} & l_{m2} & \dots & l_{\{m,m-1\}} & l_{m1}^{\langle 2\rangle} & l_{m2}^{\langle 2\rangle} & \dots & l_{mm}^{\langle 2\rangle} \end{pmatrix},$$

where n_s is the number of units given treatment s $(= 1, 2, \dots, m)$, l_s is the number of links of units given treatment s, $l_{ss'}$ is the number of links between units given treatment s and those given treatment s' with $s, s' \in \{1, 2, \dots m\}$, and ${l_{ss'}}^{\langle 2\rangle}$ the number of links between units given treatment s and those given treatment s' with a walk of length 2 (including closed walks - see Section 2.4 for more details). Recall at this point that A^2 is the squared adjacency matrix, which represents the number of two-walks (walks with two edges).

Recall that L-optimality (Section 2.1) is used for seeking designs that minimise the variance of linear combinations of the model parameters. The main interest here lies in minimising the variance of two different functions of the parameter estimates in the model (3.9) for estimating the treatment and network effects respectively. For any possible design ξ in our design space $\Xi_{\{n,m,A\}}$ it is reasonable to minimise the average variance of

– all pairwise differences of (direct) treatment effects,

$$\phi_1 = \sum_{v=2}^{m} \sum_{h=v+1}^{m+1} \boldsymbol{s}^T(v,h) M^{-1} \boldsymbol{s}(v,h);$$

– all pairwise differences of network effects (indirect treatment effects),

$$\phi_2 = \sum_{v=m+2}^{2m} \sum_{h=v+1}^{2m+1} \boldsymbol{s}^T(v,h) M^{-1} \boldsymbol{s}(v,h),$$

where $\boldsymbol{s}(\alpha_1, \alpha_2)$ is a vector of length $2m$. This is a vector of zeroes of length $2m+1$ (corresponding to the column of constants, m treatment effects and m network effects), except the α_1 and α_2 elements which are 1 and -1 respectively, before removing the $(m+1)$-th element (to account for the constraint $\tau_m = 0$). We want to find that design which minimises the value of the optimality criterion ϕ_1 or ϕ_2, i.e. ${\phi_1}^* = \min_{\xi \in \Xi} \phi_1$ and ${\phi_2}^* = \min_{\xi \in \Xi} \phi_2$.

For seeking optimal designs in the following examples, we use an exhaustive search. It is worth noting that the experimental design which is optimal for a particular criterion is not necessarily optimal for a different criterion. As we will see in the following sections the second optimality criterion is highly influenced by the network at hand and its specific network features (e.g. small-world effect, clustering, degree distribution etc.) as opposed to the first criterion. If the experimenter is interested in finding an optimal design for estimating the total effect then a weighted sum of the two above mentioned criteria may be appropriate.

Special cases of the optimality criteria formulae for the cases of two and three treatments respectively are given below. Recall that the information matrix is symmetric.

Treatment case: $m = 2$

$$\begin{aligned}\phi_1 &= \boldsymbol{s}^T(2,3)M^{-1}\boldsymbol{s}(2,3) = (0\ 1\ 0\ 0)M^{-1}(0\ 1\ 0\ 0)^T\\ \phi_2 &= \boldsymbol{s}^T(4,5)M^{-1}\boldsymbol{s}(4,5) = (0\ 0\ 1\ -1)M^{-1}(0\ 0\ 1\ -1)^T,\end{aligned}$$

where

$$M = \begin{pmatrix} n & n_1 & l_1 & l_2 \\ n_1 & n_1 & l_{11} & l_{12} \\ l_1 & l_{11} & l_{11}^{\langle 2\rangle} & l_{12}^{\langle 2\rangle} \\ l_2 & l_{12} & l_{12}^{\langle 2\rangle} & l_{22}^{\langle 2\rangle} \end{pmatrix}.$$

Treatment case: $m = 3$

$\phi_1 = \boldsymbol{s}^T(2,3)M^{-1}\boldsymbol{s}(2,3) + \boldsymbol{s}^T(2,4)M^{-1}\boldsymbol{s}(2,4) + \boldsymbol{s}^T(3,4)M^{-1}\boldsymbol{s}(3,4)$

$$= (0\ 1\ -1\ 0\ 0\ 0)\,M^{-1}\begin{pmatrix}0\\1\\-1\\0\\0\\0\end{pmatrix} + (0\ 1\ 0\ 0\ 0\ 0)\,M^{-1}\begin{pmatrix}0\\1\\0\\0\\0\\0\end{pmatrix} + (0\ 0\ 1\ 0\ 0\ 0)\,M^{-1}\begin{pmatrix}0\\0\\1\\0\\0\\0\end{pmatrix}$$

$\phi_2 = \boldsymbol{s}^T(5,6)M^{-1}\boldsymbol{s}(5,6) + \boldsymbol{s}^T(5,7)M^{-1}\boldsymbol{s}(5,7) + \boldsymbol{s}^T(6,7)M^{-1}\boldsymbol{s}(6,7)$

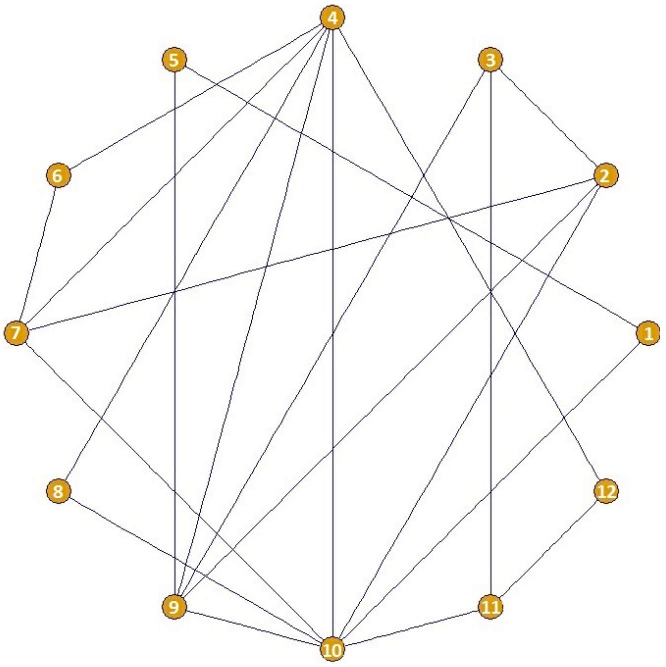
$$= (0\ 0\ 0\ 1\ -1\ 0)\,M^{-1}\begin{pmatrix}0\\0\\0\\1\\-1\\0\end{pmatrix} + (0\ 0\ 0\ 1\ 0\ -1)\,M^{-1}\begin{pmatrix}0\\0\\0\\1\\0\\-1\end{pmatrix} + (0\ 0\ 0\ 0\ 1\ -1)\,M^{-1}\begin{pmatrix}0\\0\\0\\0\\1\\-1\end{pmatrix},$$

where

$$M = \begin{pmatrix} n & n_1 & n_2 & l_1 & l_2 & l_3 \\ n_1 & n_1 & 0 & l_{11} & l_{12} & l_{13} \\ n_2 & 0 & n_2 & l_{12} & l_{22} & l_{23} \\ l_1 & l_{11} & l_{12} & l_{11}^{\langle 2\rangle} & l_{12}^{\langle 2\rangle} & l_{13}^{\langle 2\rangle} \\ l_2 & l_{12} & l_{22} & l_{12}^{\langle 2\rangle} & l_{22}^{\langle 2\rangle} & l_{23}^{\langle 2\rangle} \\ l_3 & l_{13} & l_{23} & l_{13}^{\langle 2\rangle} & l_{23}^{\langle 2\rangle} & l_{33}^{\langle 2\rangle} \end{pmatrix}.$$

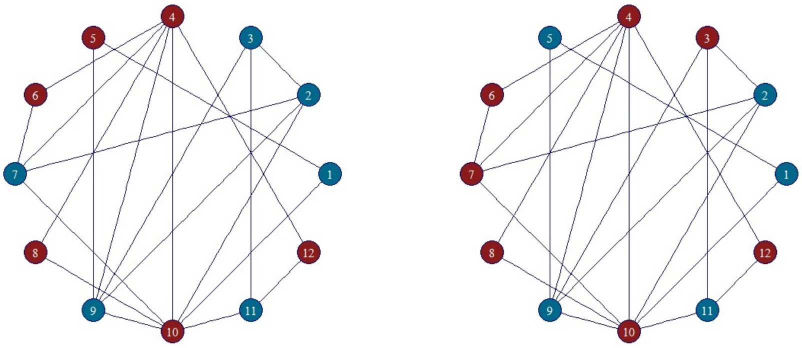
Example 3.2.1. Suppose we have a small social network of twelve subjects connected by friendship relationships represented by the graph $\mathcal{G}$ as displayed in Figure 3.1, with $|\mathcal{V}| = 12$ and $|\mathcal{E}| = 21$. For example, subject 1 is connected to subjects 5 and 10 and subject 2 is connected to subjects 3, 7, 9 and 10.

Figure 3.1: An example social network of 12 subjects and 21 links



By exhaustive search over all possible designs, we find that the optimal design for the $m = 2$ treatment case is $\{1, 1, 1, 2, 2, 2, 1, 2, 1, 2, 1, 2\}$ for optimally estimating τ_1 via criterion ϕ_1 (i.e. we give treatment 1 to subjects $1, 2, 3, 7, 9$ and 11 and treatment 2 to the other subjects). The design $\{1, 1, 2, 2, 1, 2, 2, 2, 1, 2, 1, 2\}$ is L-optimal for estimating the difference in the network effects $\gamma_1 - \gamma_2$ via criterion ϕ_2. The optimal designs on the given network are illustrated in Figure 3.2, with ${\phi_1}^* = 0.3359$ and ${\phi_2}^* = 0.0866$. We can observe that the first design is balanced; i.e. treatments 1 and 2 are applied to an equal number of subjects. However, the design for estimating the network effects (ϕ_2) is not. This is an unusual property in optimal design. Another observation is that the designs that are optimal for ϕ_1 and ϕ_2 differ.

Figure 3.2: Network of Figure 3.1 ($m = 2$); different colours indicate the optimal designs for estimating τ_1 and $\gamma_1 - \gamma_2$ for two treatments respectively



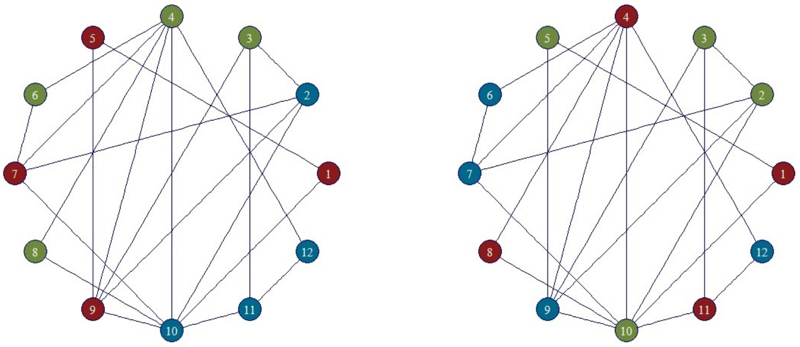
The information matrices for the different optimal designs are

$$M_1 = \begin{pmatrix} 12 & 6 & 21 & 21 \\ 6 & 6 & 10 & 11 \\ 21 & 10 & 53 & 34 \\ 21 & 11 & 34 & 59 \end{pmatrix} \qquad M_2 = \begin{pmatrix} 12 & 7 & 26 & 16 \\ 7 & 7 & 16 & 10 \\ 26 & 16 & 80 & 31 \\ 16 & 10 & 31 & 38 \end{pmatrix}.$$

From M_1, we can see that the design is balanced for the number of units receiving the different treatments ($n_1 = n_2 = 6$) who have also a balanced number of first order connections ($l_1 = l_2 = 21$). The features of the design for ϕ_2 are a bit more complicated to decipher from M_2. Section 3.4 explores this idea by identifying patterns of the optimally allocated treatments under the two optimality criteria.

We also obtain the optimal designs for comparing three distinct treatments for this network. By exhaustive search over all possible designs, we find the L-optimal design for the $m = 3$ treatment case: $\{2, 1, 3, 3, 2, 3, 2, 3, 2, 1, 1, 1\}$ and $\{2, 3, 3, 2, 3, 1, 1, 2, 1, 3, 2, 1\}$ for ϕ_1 and ϕ_2, respectively. The designs are illustrated in Figure 3.3, with ${\phi_1}^* = 1.5361$ and ${\phi_2}^* = 0.4925$ respectively. As for the case of two treatments, the designs that are optimal for ϕ_1 and ϕ_2 are different. Both designs are balanced; i.e. treatments 1, 2 and 3 are applied to an equal number of subjects.

Figure 3.3: Network of Figure 3.1 ($m = 3$); different colours indicate different treatments



The corresponding information matrices for the different designs are

$$M_1 = \begin{pmatrix} 12 & 4 & 4 & 16 & 13 & 13 \\ 4 & 4 & 0 & 6 & 5 & 5 \\ 4 & 0 & 4 & 5 & 4 & 4 \\ 16 & 6 & 5 & 28 & 17 & 21 \\ 13 & 5 & 4 & 17 & 25 & 15 \\ 13 & 5 & 4 & 21 & 15 & 21 \end{pmatrix} \qquad M_2 = \begin{pmatrix} 12 & 4 & 4 & 13 & 13 & 16 \\ 4 & 4 & 0 & 2 & 5 & 6 \\ 4 & 0 & 4 & 5 & 2 & 6 \\ 13 & 2 & 5 & 29 & 16 & 15 \\ 13 & 5 & 2 & 16 & 27 & 13 \\ 16 & 6 & 6 & 15 & 13 & 36 \end{pmatrix}.$$

3.3 Some analytical results

Building upon the work of Parker *et al.* (2016), we derive analytical expressions for the case of two treatments for the optimality criterion functions and the bias under model misspecification. We study the implications of these results and we investigate their interpretation in some common network structures. However, before doing so, we should note that for any given social network, comprising n units and l connections, there are some underlying constraints that apply to all experiments on networks:

(i) All units receive a treatment. The sum of the number of units receiving different treatments amount to the total number of units in the network, i.e. $\sum_{s=1}^{m} n_s = n$.

(ii) The sum of the degrees of all nodes equals the total number of links multiplied by two, i.e. $\sum_{j=1}^{n} d_j(\mathcal{G}) = 2\,|E| = 2l = c_1$. This is equivalent to the sum of degrees of units receiving treatments, i.e. $\sum_{s=1}^{m} l_s = c_1$, where l_s is the number of links between units one of which is receiving treatment s. Note if A_{jh} is the (j,h)-th element of A then

$$c_1 = 2l = \sum_{j=1}^{n} \sum_{\substack{h=1,\\ h\neq j}}^{n} A_{jh}.$$

(iii) The term $l_{ss'}$ denotes the number of links between units receiving treatment s and those receiving treatment s' with $s, s' \in \{1, 2, \dots, m\}$. By adding the links between units receiving a specific treatment s and all their neighbours, independently of the treatments they receive, we obtain the degree of units receiving treatment s, i.e. $\sum_{s'=1}^{m} l_{ss'} = l_s$.

(iv) The sum of the squared degrees of all nodes is $\sum_{j=1}^{n} d_j^2 = c_2$. An upper bound based on de Caen (1998) is $c_2 \leq l\left(\frac{2l}{n-1} + n - 2\right)$. As $l_s^{\langle 2\rangle}$ is the number of links between units one of which is receiving treatment s with a walk of length 2, then $\sum_s l_s^{\langle 2\rangle} = c_2$. Work is continuing in discrete mathematics to find stricter upper bounds of c_2, see for example Das (2004). However, for the purposes of our work the presented upper bound is adequate. Note that

$$c_2 = \sum_{j=1}^{n} \sum_{\substack{h=1,\\ h\neq j}}^{n} A_{jh}^2.$$

(v) The term $l_{ss'}^{\langle 2\rangle}$ denotes the number of links between units receiving treatment s and those receiving treatment s' with $s, s' \in \{1, 2, \dots, m\}$ and walk of length 2. By adding the second order connections (links of length 2) between a unit receiving a specific treatment s and all its neighbours, independently of the treatments they receive, we obtain the second degree of units receiving treatment s, i.e. $\sum_{s'=1}^{m} l_{ss'}^{\langle 2\rangle} = l_s^{\langle 2\rangle}$.

For the case of two treatments the above constraints can be re-written as

$$\begin{aligned} n_1 + n_2 &= n && (i)\\ l_1 + l_2 &= c_1 = 2l && (ii)\\ l_{11} + l_{12} &= l_1 && (iii)\\ l_{22} + l_{21} &= l_2 && (iv) \end{aligned}$$

$$
\begin{aligned}
&l_1^{\langle 2\rangle} + l_2^{\langle 2\rangle} = c_2, \quad \text{with } c_2 \le l\left(\frac{2l}{n-1} + n - 2\right) && (v)\\
&l_{11}^{\langle 2\rangle} + l_{12}^{\langle 2\rangle} = l_1^{\langle 2\rangle} && (vi)\\
&l_{22}^{\langle 2\rangle} + l_{21}^{\langle 2\rangle} = l_2^{\langle 2\rangle} && (vii)
\end{aligned}
$$

Optimality criteria

We obtain the analytical formulae of the optimality criteria, which can be useful for the construction of efficient designs and can also be beneficial for speeding up the optimisation algorithm when dealing with irregular and complicated neighbour structures (e.g. by avoiding the inversion of the information matrix).

Lemma 1. The analytical expressions of the optimality criteria for the $m = 2$ treatment case are

$$\phi_1 = \frac{l_1^2 l_2^{\langle 2\rangle} + l_2^2 l_1^{\langle 2\rangle} - l_{12}^{\langle 2\rangle}\left(c_1^2 - nc_2\right) - n l_1^{\langle 2\rangle} l_2^{\langle 2\rangle}}{D} \tag{3.10}$$

and

$$\phi_2 = \frac{n_2 l_1^2 + n_1 l_2^2 - n_1 n_2 c_2}{D} \tag{3.11}$$

where

$$
\begin{aligned}
D = {} & l_1^{\langle 2\rangle} n_1 l_2 (l_2 - 2l_{12}) + l_2^{\langle 2\rangle} n_2 l_1 (l_1 - 2l_{12}) - l_{12}^{\langle 2\rangle}\left(n_1 l_2^2 + n_2 l_1^2 - n_1 n_2 c_2\right)\\
& - n_1 n_2 l_1^{\langle 2\rangle} l_2^{\langle 2\rangle} - l_{12}^2\left(c_1^2 - nc_2\right) - l_1 l_2 \left(l_1 l_2 - 2c_1 l_{12}\right).
\end{aligned}
$$

Most of the calculations of the proof of Lemma 1 can be found in Appendix A. We provide here an overview of the proof. The information matrix, M, and its inverse, M^{-1}, are

$$
M = \begin{pmatrix} n & n_1 & l_1 & l_2 \\ n_1 & n_1 & l_{11} & l_{12} \\ l_1 & l_{11} & l_{11}^{\langle 2\rangle} & l_{12}^{\langle 2\rangle} \\ l_2 & l_{12} & l_{12}^{\langle 2\rangle} & l_{22}^{\langle 2\rangle} \end{pmatrix}
\qquad
M^{-1} = \frac{1}{D}\begin{pmatrix} M_{11} & M_{12} & M_{13} & M_{14} \\ M_{12} & M_{22} & M_{23} & M_{24} \\ M_{13} & M_{23} & M_{33} & M_{34} \\ M_{14} & M_{24} & M_{34} & M_{44} \end{pmatrix}
$$

where

$$
\begin{aligned}
M_{11} &= l_{22}^{\langle 2\rangle} l_{11}^2 - 2 l_{11} l_{12} l_{12}^{\langle 2\rangle} + l_{11}^{\langle 2\rangle} l_{12}^2 + n_1 \left(l_{12}^{\langle 2\rangle}\right)^2 - l_{11}^{\langle 2\rangle} l_{22}^{\langle 2\rangle} n_1\\
M_{12} &= -\left(l_{12}^{\langle 2\rangle}\right)^2 n_1 + l_2 l_{12} l_{11}^{\langle 2\rangle} - l_1 l_{12} l_{12}^{\langle 2\rangle} - l_2 l_{11} l_{12}^{\langle 2\rangle} + l_1 l_{11} l_{22}^{\langle 2\rangle} - l_{11}^{\langle 2\rangle} l_{22}^{\langle 2\rangle} n_1\\
M_{13} &= -l_1 l_{12}^2 - l_2 l_{11} l_{12} + l_2 l_{12}^{\langle 2\rangle} n_1 - l_{12} l_{12}^{\langle 2\rangle} n_1 - l_1 l_{22}^{\langle 2\rangle} n_1 + l_{11} l_{22}^{\langle 2\rangle} n_1\\
M_{14} &= -l_2 l_{11}^2 - l_1 l_{11} l_{12} - l_2 l_{11}^{\langle 2\rangle} n_1 + l_1 l_{12}^{\langle 2\rangle} n_1 + l_{12} l_{11}^{\langle 2\rangle} n_1 - l_{11} l_{12}^{\langle 2\rangle} n_1\\
M_{22} &= l_{22}^{\langle 2\rangle} l_1^2 - 2 l_1 l_2 l_{12}^{\langle 2\rangle} + l_{11}^{\langle 2\rangle} l_2^2 + n \left(l_{12}^{\langle 2\rangle}\right)^2 - l_{11}^{\langle 2\rangle} l_{22}^{\langle 2\rangle} n\\
M_{23} &= -l_2^2 l_{11} - l_1 l_2 l_{12} - l_2 l_{12}^{\langle 2\rangle} n_1 + l_{12} l_{12}^{\langle 2\rangle} n + l_1 l_{22}^{\langle 2\rangle} n_1 - l_{11} l_{22}^{\langle 2\rangle} n\\
M_{24} &= -l_1^2 l_{12} - l_1 l_2 l_{11} + l_2 l_{11}^{\langle 2\rangle} n_1 - l_1 l_{12}^{\langle 2\rangle} n_1 - l_{12} l_{11}^{\langle 2\rangle} n + l_{11} l_{12}^{\langle 2\rangle} n
\end{aligned}
$$

$$\begin{aligned}
M_{33} &= l_2^2 n_1 - 2l_2 l_{12} n_1 + n l_{12}^2 + l_{22}^{\langle 2\rangle} n_1^2 - l_{22}^{\langle 2\rangle} n n_1 \\
M_{34} &= -l_{12}^{\langle 2\rangle} n_1^2 + l_1 l_2 n_1 - l_1 l_{12} n_1 - l_2 l_{11} n_1 + l_{11} l_{12} n - l_{12}^{\langle 2\rangle} n n_1 \\
M_{44} &= l_1^2 n_1 - 2l_1 l_{11} n_1 + n l_{11}^2 + l_{11}^{\langle 2\rangle} n_1^2 - l_{11}^{\langle 2\rangle} n n_1
\end{aligned}$$

and

$$\begin{aligned}
D = &-l_1^2 l_{12}^2 + l_{22}^{\langle 2\rangle} l_1^2 n_1 + 2l_1 l_2 l_{11} l_{12} - 2l_1 l_2 l_{12}^{\langle 2\rangle} n_1 - 2l_{22}^{\langle 2\rangle} l_1 l_{11} n_1 + 2l_1 l_{12} l_{12}^{\langle 2\rangle} n_1 - l_2^2 l_{11}^2 \\
&+ l_{11}^{\langle 2\rangle} l_2^2 n_1 + 2l_2 l_{11} l_{12}^{\langle 2\rangle} n_1 - 2l_{11}^{\langle 2\rangle} l_2 l_{12} n_1 + l_{22}^{\langle 2\rangle} n l_{11}^2 - 2n l_{11} l_{12} l_{12}^{\langle 2\rangle} + l_{11}^{\langle 2\rangle} n l_{12}^2 \\
&- \left(l_{12}^{\langle 2\rangle}\right)^2 n_1^2 + n\left(l_{12}^{\langle 2\rangle}\right)^2 n_1 + l_{11}^{\langle 2\rangle} l_{22}^{\langle 2\rangle} n_1^2 - l_{11}^{\langle 2\rangle} l_{22}^{\langle 2\rangle} n n_1.
\end{aligned}$$

Aggregating the expression of the determinant of the information matrix, D, and implementing some of the above-mentioned constraints, it can be re-written either as a function of the first (immediate) and second order neighbouring relationships or as a function of the first and second order degrees of units receiving one of the different treatments. In particular

$$\begin{aligned}
D = &\, l_{11}^{\langle 2\rangle}\left(n_1 l_{22}^2 + n_2 l_{12}^2\right) + l_{22}^{\langle 2\rangle}\left(n_2 l_{11}^2 + n_1 l_{12}^2\right) - 2l_{12} l_{12}^{\langle 2\rangle}\left(n_1 l_{22} + n_2 l_{11}\right) \\
&+ n_1 n_2\left[\left(l_{12}^{\langle 2\rangle}\right)^2 - l_{11}^{\langle 2\rangle} l_{22}^{\langle 2\rangle}\right] - \left({l_{12}}^2 - l_{11} l_{22}\right)^2 \qquad (3.12) \\
= &\, l_1^{\langle 2\rangle} n_1 l_2 (l_2 - 2l_{12}) + l_2^{\langle 2\rangle} n_2 l_1 (l_1 - 2l_{12}) - l_{12}^{\langle 2\rangle}\left(n_1 l_2^2 + n_2 l_1^2 - n_1 n_2 c_2\right) \\
&- n_1 n_2 l_1^{\langle 2\rangle} l_2^{\langle 2\rangle} - l_{12}^2\left(c_1^2 - n c_2\right) - l_1 l_2\left(l_1 l_2 - 2c_1 l_{12}\right). \qquad (3.13)
\end{aligned}$$

The derivation of the D for the two different ways is presented in Appendix A. Having calculated the inverse of the information matrix, we proceed by pre- and post- multiplying it by the appropriate vectors for obtaining the functions of the parameters we want to estimate (corresponding to the treatment and network effects), i.e.

$$\phi_1 = \boldsymbol{s}^T(2,3) M^{-1} \boldsymbol{s}(2,3) = (0\;1\;0\;0) M^{-1} (0\;1\;0\;0)^T = \frac{M_{22}}{D},$$

$$\begin{aligned}
\phi_2 &= \boldsymbol{s}^T(4,5) M^{-1} \boldsymbol{s}(4,5) = (0\;0\;1\;-1) M^{-1} (0\;0\;1\;-1)^T = \frac{M_{33} - M_{34} - M_{43} + M_{44}}{D} \\
&= \frac{M_{33} - 2M_{34} + M_{44}}{D}.
\end{aligned}$$

After simple manipulations, provided in Appendix A, we can write the expressions of the optimality criteria as

$$\phi_1 = \frac{l_1^2 l_2^{\langle 2\rangle} + l_2^2 l_1^{\langle 2\rangle} - l_{12}^{\langle 2\rangle}\left(c_1^2 - n c_2\right) - n l_1^{\langle 2\rangle} l_2^{\langle 2\rangle}}{D} \qquad (3.14)$$

$$\phi_2 = \frac{n_2 l_1^2 + n_1 l_2^2 - n_1 n_2 c_2}{D} \qquad (3.15)$$

where D is of the form of Equation (3.13).

The denominator comprises both first and second order quantities. Since the numerator of ϕ_1 also comprises second order quantities by division their effects are reduced and as such this criterion is less dominated by them. The opposite holds for ϕ_2. As already mentioned $l_s^{\langle 2\rangle}$ denotes the second degree given a specific treatment (i.e. number of links of units given treatment s with a walk of length two). It is formed as a sum of the number of links of length two connected to units receiving treatment s, which incorporates the sum of the degrees of those units. The sum of those second degrees, indicated by c_2, is present in both expressions (3.14) and (3.15). A logical corollary is that c_2 increases with the number of connections of a unit. One can deduce that both expressions depend on the number of first and second order pathways and that the experimental design is partially determined by the specific structure of the network (c_1 and c_2 subject to the constraints mentioned above).

Conditions of the lower bound for ϕ_1

When the units are not connected in a network as in the case of a CRD, the optimal design is an equi-replicate design. For instance when there are two unstructured treatments this means that $\phi_1 = 4\sigma^2/n$. Thus under the non interference assumption we achieve that lowest bound of ϕ_1. When units are connected in a network this bound helps us to identify an efficient design and also helps us with the algorithms of the design (see Section 3.6). In particular, for $n_1 = n_2 = n/2, l_1 = l_2 = c_1/2 = l, l_{12} = c_1/4 = l/2$ and by implementing the constraint $l_2^{\langle 2\rangle} = c_2 - l_1^{\langle 2\rangle}$ in Equation (3.14) we obtain $\phi_1 = 4/n$, which is the same as the minimum average variance possible for the unrestricted case when having independent units (with $\sigma^2 = 1$ without loss of generality). This topic will be further discussed in Section 3.4 together with the features of the properties of a good design in order to estimate the direct treatment effects.

Special graphs

Lemma 2. No optimal design exists with respect to the LNM for complete, star and ring networks.

For complete, star and ring graphs (presented in Section 2.4), the LNM cannot be implemented due to the fact that the expressions related to the first and second order connections are multiples of each other leading to a singular information matrix. It is quite straightforward to notice that the denominator of the expressions of the optimality criteria is zero. This means that both ϕ_1 and ϕ_2 are incalculable (i.e. they cannot be defined). In other words the designs on these networks lead to the LNM being inestimable. More specifically:

- For the complete network:
  $l_1/n_1 = l_2/n_2 = l_1^{\langle 2\rangle}/l_1 = l_2^{\langle 2\rangle}/l_2 = n-1,\ \ {l_{12}}^{\langle 2\rangle}/l_{12} = n-2,\ \ n_1 n_2 = l_{12}.$
- For the star network:
  $n_1 = l_1 = l_{12},\ \ l_{11} = 0,\ \ {l_{12}}^{\langle 2\rangle}/l_{12} = n_2 - 1$ or

$n_2 = l_2 = l_{12},\ l_{22} = 0,\ l_{12}{}^{\langle 2\rangle}/l_{12} = n_1 - 1$

- For the ring network:
  $l_1^{\langle 2\rangle} = 2l_1 = 2\,(2n_1)\,,\ l_2^{\langle 2\rangle} = 2l_2 = 2\,(2n_2).$

The proof of Lemma 2 follows directly from the application using the constraints in the Equations (3.14) and (3.15). For instance, focusing on the denominator of the expressions of the criteria we can see that, for the star and ring cases,

$$\begin{aligned} l_1^{\langle 2\rangle} n_1 l_2 (l_2 - 2l_{12})^2 + l_2^{\langle 2\rangle} n_2 l_1 (l_1 - 2l_{12})^2 &= -n_1 n_2 l_1^{\langle 2\rangle} l_2^{\langle 2\rangle} - l_1 l_2 (l_1 l_2 - 2c_1 l_{12}) \\ n_1 l_2^2 + n_2 l_1^2 &= n_1 n_2 c_2 \\ c_1^2 &= n c_2. \end{aligned}$$

At this point we will focus on a tree network in its simplest form. This is a network comprising an even number of vertices, where each vertex has an even number of branch vertices (which is constant across the tree). For the case where the number of treatments is a multiple of the number of leaves, the best design under ϕ_1 is balanced. For instance for the $m = 2-$treatment case we have that $n_1 = n_2 = n/2,\quad l_1 = l_2$, but also $l_{11} = l_{22}$. Notice that the tree-network is symmetric. Thus when we have a balanced treatment allocation on the vertices in each branch, we anticipate that the design will be optimal. We further examine this result in Section 3.4.

Those topologies are commonplace in large complex networks. In Chapter 4 we see how we can benefit from identifying these structures for substantially reducing the search time for finding efficient designs. In the same chapter we also formulate the conjecture under which we claim that LNM is not appropriate for modelling these types of special graphs. In particular, we notice that if a given network can be partitioned into a maximum of two sets of structurally equivalent vertices (i.e. vertex orbits- see Definition 5 in Section 4.1), then the LNM cannot be implemented due to the singular information matrix. More details will be provided in the relevant chapter.

Design bias due to model misspecification

The statistical properties of the parameters depend on how well the model describes the true process under study. Serious misspecifications may give biased and/or inefficient parameter estimators. Work that discuss the potential impact of model misspecification are that of Larson and Bancroft (1963) on the parameter estimators and Box and Draper (1959) on the choice of optimal design. Box and Draper (1959) suggested a design search strategy for precisely estimating the terms of the reduced fitted model while considering some lack of fit components of the suitable complete model. Note that a complete model is the assumed true model with the extended design matrix, while the reduced model is a sub-model obtained by deleting some of the model parameters. The bias of an estimator $\hat{\beta}$ is defined as the deviation of the expectation from the true value, i.e. $\mathbb{E}\left[\hat{\beta}\right] - \beta$. All else being equal, the estimator with the smaller bias is preferable. Note that the bias has not been taken into account in the design optimality criterion used in this thesis.

Work that explore methods of incorporating the possibility of model misspecification in compound criteria are those of Goos *et al.* (2005), Gilmour and Trinca (2012), Egorova (2017) and others. However, we evaluate the bias in the treatment effects' estimates due to model misspecification via simulations. Some numerical examples will follow in Section 3.5. Moreover, in Section 5.4 we obtain optimal designs under different models and the corresponding design biases with respect to the unknown parameters for each model.

We consider performing an experiment on a network, where the adjacency matrix, A, is given. We assume that the true model is the LNM (3.9). However, the postulated model for the experiment is the CRM (2.4), which ignores network effects (i.e. $\boldsymbol{\gamma} = \mathbf{0}$). We want to obtain the bias in the parameter estimates due to the model misspecification as a function of the unknown model parameters. This will be accomplished by using generalised inverse matrices (Harville, 1997, Ch.9). Let ${X_\tau}^{\star}$ be the $n \times (m-1)$ optimal design matrix for the true model (implementing the standard constraints discussed in Section 3.2, so X_τ loses its last column). The extended design matrices under the LNM (3.9) and CRM (2.4) are then of the form $X_R = (\mathbf{1}\ {X_\tau}^{\star}\ \mathbf{0}_{n\times m})$ and $X_C = (\mathbf{1}\ {X_\tau}^{\star}\ AX_\tau)$ each column of which contains the intercept, the treatment effects and the network effects (R and C denote the 'reduced' and 'complete' respectively). Note that $\mathbb{E}\left[\boldsymbol{\epsilon}\right] = \mathbf{0}$, therefore $\mathbb{E}\left[{X_C}^T\boldsymbol{\epsilon}\right] = \mathbf{0}$ and $\mathbb{E}\left[{X_R}^T\boldsymbol{\epsilon}\right] = \mathbf{0}$, i.e. the observed values of X_C and X_R respectively are uncorrelated with the corresponding residuals (X_C and X_R are fixed). Note that $\hat{\boldsymbol{\beta}}_C$ is the best linear unbiased estimator for $\boldsymbol{\beta}$ so that $\mathbb{E}\left[\hat{\boldsymbol{\beta}}_C\right] = \boldsymbol{\beta}$. With the necessary algebraic calculations on generalised inverse matrices, it follows that the bias of the design, $\mathcal{W}$, under the assumption that there are network effects but we do not take them into account, is

$$
\begin{aligned}
\mathcal{W} &= \mathbb{E}\left[\hat{\boldsymbol{\beta}}_R - \hat{\boldsymbol{\beta}}_C\right] = \mathbb{E}\left[\left({X_R}^T X_R\right)^{-}{X_R}^T\boldsymbol{y} - \left({X_C}^T X_C\right)^{-}{X_C}^T\boldsymbol{y}\right] \\
&= \left(\left({X_R}^T X_R\right)^{-}{X_R}^T - \left({X_C}^T X_C\right)^{-}{X_C}^T\right)\mathbb{E}\left[\boldsymbol{y}\right] \\
&= \left(\left({X_R}^T X_R\right)^{-}{X_R}^T - \left({X_C}^T X_C\right)^{-}{X_C}^T\right)X_C\boldsymbol{\beta} \\
&= \left(\left({X_R}^T X_R\right)^{-}{X_R}^T X_C - \left({X_C}^T X_C\right)^{-}{X_C}^T X_C\right)\boldsymbol{\beta} \\
&= \left(\left({X_R}^T X_R\right)^{-}{X_R}^T X_C - I_{2m}\right)\boldsymbol{\beta} \\
&= \left(\left((\mathbf{1}\ {X_\tau}^{\star}\ \mathbf{0}_{n\times m})^T\begin{pmatrix}\mathbf{1}\\ {X_\tau}^{\star}\\ \mathbf{0}_{n\times m}\end{pmatrix}\right)^{-}(\mathbf{1}\ {X_\tau}^{\star}\ \mathbf{0}_{n\times m})^T\begin{pmatrix}\mathbf{1}\\ {X_\tau}^{\star}\\ AX_\tau\end{pmatrix} - I_{2m}\right)\boldsymbol{\beta} \\
&= \left(\begin{pmatrix}\mathbf{1}^T\mathbf{1} & \mathbf{1}^T{X_\tau}^{\star} & \mathbf{0}\\ {X_\tau}^{\star T}\mathbf{1} & {X_\tau}^{\star T}{X_\tau}^{\star} & \mathbf{0}\\ \mathbf{0} & \mathbf{0} & \mathbf{0}\end{pmatrix}^{-}\begin{pmatrix}\mathbf{1}^T\mathbf{1} & \mathbf{1}^T{X_\tau}^{\star} & \mathbf{1}^T AX_\tau\\ {X_\tau}^{\star T}\mathbf{1} & {X_\tau}^{\star T}{X_\tau}^{\star} & {X_\tau}^{\star T}{X_\tau}^{\star}\\ \mathbf{0} & \mathbf{0} & \mathbf{0}\end{pmatrix} - I_{2m}\right)\boldsymbol{\beta},
\end{aligned}
$$

where I_{2m} is the $2m \times 2m$ identity matrix. Let B and Γ represent the $m \times m$ matrices

$$B = \begin{pmatrix} \mathbf{1}^T\mathbf{1} & \mathbf{1}^T X_\tau{}^\star \\ X_\tau{}^{\star T}\mathbf{1} & X_\tau{}^{\star T} X_\tau{}^\star \end{pmatrix} \quad \text{and} \quad \Gamma = \begin{pmatrix} \mathbf{1}^T A X_\tau \\ X_\tau{}^{\star T} A X_\tau \end{pmatrix}.$$

Then for the $2m \times 2m$ block-diagonal matrix

$$A = \begin{pmatrix} B & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{pmatrix},$$

we have that B is a non-singular $m \times m$ matrix (has full rank), and defining a $2m \times 2m$ matrix

$$G = \begin{pmatrix} G_{11} & G_{12} \\ G_{21} & G_{22} \end{pmatrix}$$

(where G_{11} is of dimension $m \times m$), we obtain

$$AGA = \begin{pmatrix} BG_{11}B & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{pmatrix},$$

implying that G is a generalised inverse of A if and only if $BG_{11}B = B$, or if and only if $G_{11} = B^{-1}$. Hence, we have

$$\mathcal{W} = \left(\begin{pmatrix} B^{-1} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{pmatrix} \begin{pmatrix} B & \Gamma \\ \mathbf{0} & \mathbf{0} \end{pmatrix} - I_{2m} \right) \boldsymbol{\beta} = \left(\begin{pmatrix} B^{-1}B & B^{-1}\Gamma \\ \mathbf{0} & \mathbf{0} \end{pmatrix} - I_{2m} \right) \boldsymbol{\beta}$$

$$= \left(\begin{pmatrix} I & B^{-1}\Gamma \\ \mathbf{0} & \mathbf{0} \end{pmatrix} - I_{2m} \right) \boldsymbol{\beta} = \begin{pmatrix} \mathbf{0} & B^{-1}\Gamma \\ \mathbf{0} & -I_m \end{pmatrix} \boldsymbol{\beta}.$$

Thus the bias introduced in the estimates of the parameters, $\hat{\boldsymbol{\beta}}_R$, under the false assumption that there are no network effects is given by the quantity $B^{-1}\Gamma$. This quantity is the result of ignoring the network effects, which is represented by an adjustment of the intercept and the treatment effect estimates. Observe that $\mathbb{E}\left[\hat{\boldsymbol{\beta}}_R\right] \neq \mathbb{E}\left[\hat{\boldsymbol{\beta}}_C\right]$ unless $B^{-1}\Gamma = \mathbf{0}$, which results from $\Gamma = \mathbf{0}$ or $\boldsymbol{\gamma} = \mathbf{0}$.

We can easily obtain the analytical expression of the bias in the treatment effect estimates as a function of the unknown parameters, for the case of two treatments (under the constraint $\tau_2 = 0$). We have

$$B = \begin{pmatrix} \mathbf{1}^T\mathbf{1} & \mathbf{1}^T\boldsymbol{u}_1 \\ u_1{}^T\mathbf{1} & \boldsymbol{u}_1^T\boldsymbol{u}_1 \end{pmatrix} = \begin{pmatrix} n & n_1 \\ n_1 & n_1 \end{pmatrix} \quad \text{with} \quad B^{-1} = \frac{1}{nn_1 - n_1^2} \begin{pmatrix} n_1 & -n_1 \\ -n_1 & n \end{pmatrix}$$

and

$$\Gamma = \begin{pmatrix} \mathbf{1}A\boldsymbol{u}_1 & \mathbf{1}A\boldsymbol{u}_2 \\ \boldsymbol{u}_1A\boldsymbol{u}_1 & \boldsymbol{u}_1A\boldsymbol{u}_2 \end{pmatrix} = \begin{pmatrix} l_1 & l_2 \\ l_{11} & l_{12} \end{pmatrix}.$$

Recall that $nn_1 - n_1^2 = n_1(n - n_1) = n_1 n_2$. Therefore, we have

$$B^{-1}\Gamma = \frac{1}{n_1 n_2}\begin{pmatrix} n_1 l_1 - n_1 l_{11} & n_1 l_2 - n_1 l_{12} \\ \\ -n_1 l_1 + n l_{11} & -n_1 l_2 + n l_{12} \end{pmatrix} = \begin{pmatrix} \dfrac{l_1 - l_{11}}{n_2} & \dfrac{l_2 - l_{12}}{n_2} \\ -\dfrac{n_1 l_1 - n l_{11}}{n_1 n_2} & -\dfrac{n_1 l_2 - n l_{12}}{n_1 n_2} \end{pmatrix},$$

which comes down to the expression of bias when we attempt to fit a model which ignores the network effects although they exist, of

$$\mathcal{W} = \begin{pmatrix} 0 & 0 & \dfrac{l_1 - l_{11}}{n_2} & \dfrac{l_2 - l_{12}}{n_2} \\ 0 & 0 & -\dfrac{l_1 n_1 - l_{11} n}{n_1 n_2} & -\dfrac{l_2 n_1 - l_{12} n}{n_1 n_2} \\ 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & -1 \end{pmatrix}\begin{pmatrix} \mu \\ \tau_1 \\ \gamma_1 \\ \gamma_2 \end{pmatrix}.$$

Thus the bias in the treatment effects as a function of the unknown parameters $\boldsymbol{\beta}$ for $m = 2$ is

$$\begin{aligned} \text{Bias}(\widehat{\tau}_1) &= \beta_{\gamma_1}\gamma_1 + \beta_{\gamma_2}\gamma_2 & (3.16) \\ &= -\frac{l_1 n_1 - l_{11} n}{n_1 n_2}\gamma_1 - \frac{l_2 n_1 - l_{12} n}{n_1 n_2}\gamma_2, & (3.17) \end{aligned}$$

or by implementing the constraints (i), (iii) and (iv) we have

$$\text{Bias}(\widehat{\tau}_1) = \left(\frac{l_{11}}{n_1} - \frac{l_{12}}{n_2}\right)\gamma_1 - \left(\frac{l_{22}}{n_2} - \frac{l_{12}}{n_1}\right)\gamma_2. \qquad (3.18)$$

3.4 Patterns of optimally allocated treatments

This section will serve as a rough guide for obtaining efficient designs for estimating the direct treatment effects or the network effects. This will be achieved by exploring, by means of explicit examples, the optimally allocated treatments' patterns. These patterns can enable us to find efficient designs rapidly when dealing with large sized networks comprising hundreds of vertices and edges without the need of a computer software. We will then draw some general guidelines which can be used by practitioners when designing experiments on networks. We have gathered evidence which suggests that designs that present certain patterns are more efficient for the majority of cases. To this end, we provide additional example networks, which consist of some of the most commonly found network topologies, and we verify some of the observed general patterns. Another aim of this section is to discuss elements related to the search for the optimal design. When the optimal design cannot be found by an exhaustive search, we resort to approximate methods as in the case of our exchange algorithm that will be introduced in Section 3.6. By incorporating the general treatment allocation patterns that will be discussed in the current section, we can efficiently speed up our algorithmic approach for finding efficient designs. More computational details of this issue will follow in Chapter 4.

Expanding the analytical results presented in the previous section, we initially attempted to optimise analytically the criteria of Equations (3.14) and (3.15). To achieve this optimisation, we implemented a search algorithm to solve the constraint satisfaction problem under the linear constraints $(i) - (vii)$ (see Sections 2.3 and 3.3). Our approach was based on notions considered in the work of Brailsford *et al.* (1999). However, the bound constraints were not sufficient to obtain optimal solutions. Subject to those constraints we cannot choose freely our quantities, as they are all partially determined by the specific structure of the network. Following another unsuccessful attempt we focused on elements that dominate these expressions, i.e. the second order degrees. As a way to limit the range of the possible values, which stem mostly from the network sizes and shapes, we attempted to assign specific values to some of the fixed quantities such as the total number of units and connections. The desired solution of a given optimal interval could not be attained. Thus implementing a local search we end up in a few small-sized networks, where trials of different values in the criteria expressions enabled us to compare a number of candidate sub-optimal designs with respect to the optimal design. Exploring this issue by means of tables and graphs helps us investigate what network properties induce particular designs and to identify those features that make a design good enough.

Revisiting Example 3.2.1, let us first investigate how the different treatment allocations affect the efficiency for estimating the treatment and network effects. Tables 3.1 and 3.2 show properties of the optimal design obtained from an exhaustive search (first line) and ten sub-optimal designs as obtained by increasing or decreasing some of the values in the criteria expressions. Each row corresponds to a design, and breaks down the information matrix for each design into its components.

From a first inspection of Table 3.1 the optimal design for ϕ_1 is balanced on the number of units receiving the different treatments ($n_1 = n_2 = n/2$) and also balanced on the number of first and second order connections ($l_1 = l_2 = l$ and $l_1^{\langle 2 \rangle} \approx l_2^{\langle 2 \rangle}$). We should note that the number of treatments is not divisible by the number of connections. Moreover the links connecting same or different treatments are approximately the same ($l_{11} \approx l_{12} \approx l_{22}$), which tends to be true also for the second order links connecting the same treatments ($l_{11}^{\langle 2 \rangle} \approx l_{22}^{\langle 2 \rangle}$). The first few rows of Table 3.1 show balanced designs with equal numbers of treatments allocated to units of similar numbers of connections, but with different second order connections. On the other hand, looking at Table 3.2 the optimal design for ϕ_2 is neither balanced for the number of units receiving the

Table 3.1: List of designs with the corresponding calculated quantities of the information matrix. Top row optimal design for ϕ_1

ϕ_1	ϕ_2	n_1	n_2	l_1	l_2	l_{11}	l_{12}	l_{22}	$l_{11}^{\langle 2\rangle}$	$l_{12}^{\langle 2\rangle}$	$l_{22}^{\langle 2\rangle}$	$l_{1}^{\langle 2\rangle}$	$l_{2}^{\langle 2\rangle}$
0.33594	0.09395	6	6	21	21	10	11	10	53	34	59	87	93
0.33902	0.20465	6	6	21	21	10	11	10	49	40	51	89	91
0.34143	0.29139	6	6	21	21	10	11	10	53	41	45	94	86
0.38182	0.21382	6	6	22	20	10	12	8	50	39	52	89	91
0.40805	0.12010	6	6	25	17	14	11	6	71	35	39	106	74
0.45374	0.27298	3	9	10	32	2	8	24	14	29	108	43	137
0.54063	0.24911	7	5	23	19	10	13	6	63	38	41	101	79
0.71764	0.16259	8	4	29	13	16	13	0	89	28	35	117	63
0.87558	0.17093	6	6	22	20	6	16	4	58	28	66	86	94
1.57150	0.52174	6	6	25	17	10	15	2	63	35	47	98	82
2.00000	0.46875	6	6	24	18	8	16	2	64	32	52	96	84

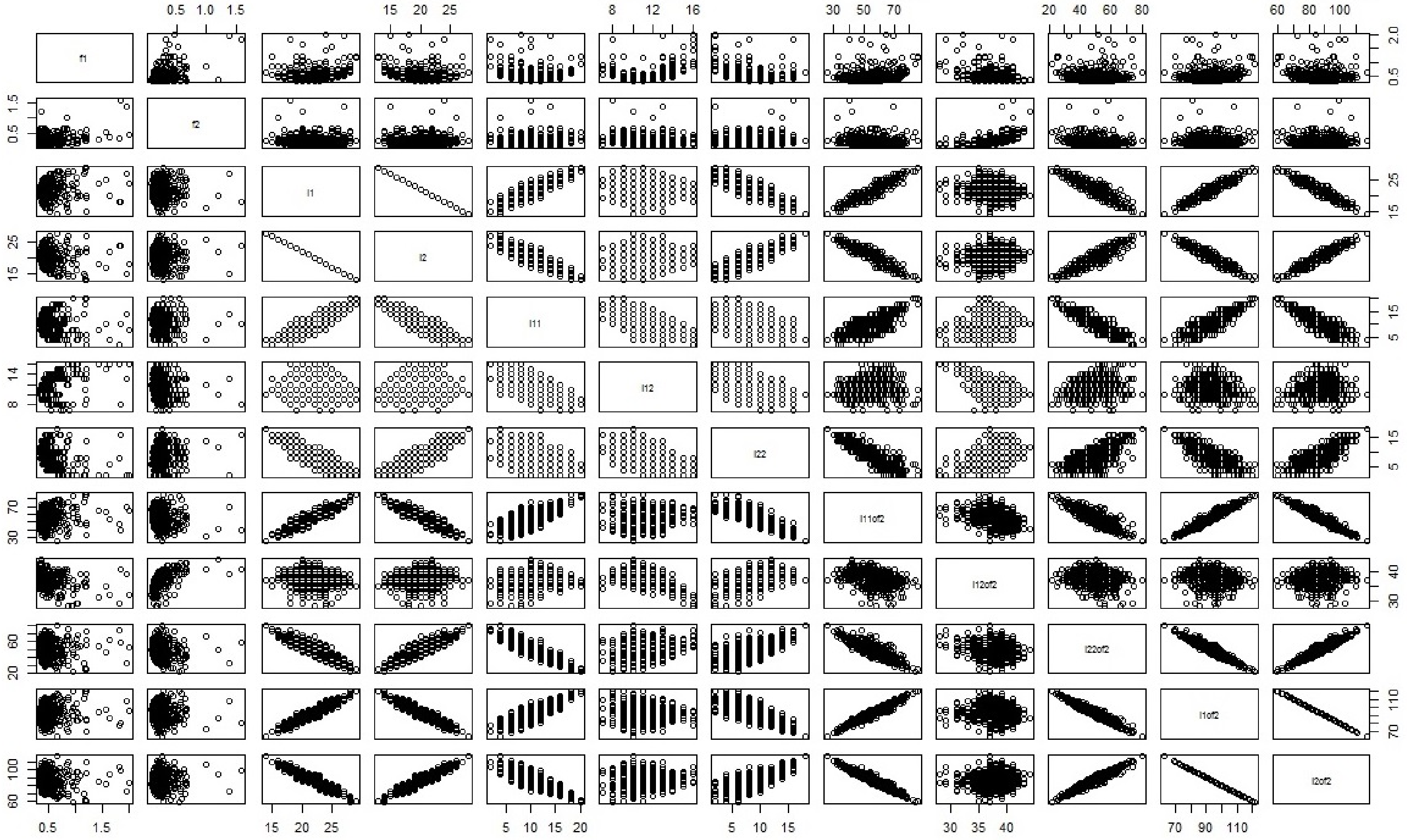
Table 3.2: List of designs with the corresponding calculated quantities of the information matrix. Top row optimal design for ϕ_2

ϕ_1	ϕ_2	n_1	n_2	l_1	l_2	l_{11}	l_{12}	l_{22}	$l_{11}^{\langle 2\rangle}$	$l_{12}^{\langle 2\rangle}$	$l_{22}^{\langle 2\rangle}$	$l_{1}^{\langle 2\rangle}$	$l_{2}^{\langle 2\rangle}$
0.35112	**0.08662**	7	5	26	16	16	10	6	80	31	38	111	69
0.38566	0.08721	6	6	24	18	12	12	6	66	32	50	98	82
0.36401	0.10909	6	6	22	20	10	12	8	58	35	52	93	87
0.33740	0.14650	6	6	21	21	10	11	10	59	37	47	96	84
0.59482	0.21336	5	7	23	19	10	13	6	53	38	51	91	89
0.39301	0.27511	4	8	16	26	6	10	16	28	38	76	66	114
0.40989	0.37643	7	5	27	15	16	11	4	71	39	31	110	70
0.73141	0.46323	6	6	23	19	10	13	6	59	40	41	99	81
0.83333	0.50000	2	10	6	36	2	4	32	6	18	138	24	156
0.84055	0.66970	6	6	20	22	12	8	14	48	41	50	89	91
1.84211	1.57895	6	6	18	24	10	8	16	40	41	58	81	99

different treatments nor for the number of first and second order connections. As we will see in the following example the optimality of the design is greatly affected by the shape of the network. Figure 3.4 highlights all the pairwise combinations of the components of the information matrix in scatterplots (e.g. ϕ_1, ϕ_2, l_1, l_2, l_{11} etc.). The first two rows (or columns) show scatterplots between each of the two optimality criteria related to each one of the the information matrix quantities, indicating that there are no apparent patterns (this result is a corollary of the non-linear expressions (3.14) and (3.15)). The remaining rows show the relationships among each of the elements of the information matrix, which arise as the result of constraints in the experiment on the network (discussed in Section 3.3). As an example when l_1 is increasing we expect l_2 to decrease. Similar correlations also can be inferred from the other rows.

Example 3.4.1. For this example the given network comprises 32 units with 36 connections (the example network is taken from MacArthur *et al.*, 2008). The optimal function values under LNM are ${\phi_1}^* = 0.12500$ and ${\phi_2}^* = 0.02375$ with the corresponding optimal allocations illustrated in Figure 3.5.

Figure 3.4: Pairwise comparisons of the optimal function values and each of the information matrix elements for all balanced designs, i.e. ϕ_1, ϕ_2, l_1, l_2, l_{11}, l_{12}, l_{22}, $l_{11}^{\langle 2\rangle}$, $l_{12}^{\langle 2\rangle}$, $l_{22}^{\langle 2\rangle}$, $l_1^{\langle 2\rangle}$ and $l_2^{\langle 2\rangle}$



For ϕ_1, units with the same number of (first and second) connections receive the same treatment, retaining balance overall (Figure 3.5(*left*)). Additionally, $\phi_1 = 0.125$ equals the minimum average variance when the units are independent (with $\sigma^2 = 1$ without loss of generality). Note that there may be more than one way to equally allocate the treatments to units of the same degree. For ϕ_2, there is a tendency to equal allocation among vertices of the same first and second degree (see Figure 3.5(*right*)). This holds for units belonging to a tree-like structure, with leaves receiving the same treatment which is different from that received by immediately connected vertices in each branch. For instance units $21, 22, 24$, and 25 receive treatment 1 while 23 and 20 receive treatment 2 and unit 5 treatment 1. Similarly units in the star structure $12, 13, 14$ and 15 receive treatment 2 while unit 2 receives treatment 1. Another observation is that units belonging to a clique $(26, 27, 28)$ receive the same treatment. The optimal design under ϕ_2 is unique (up to relabelling of treatments), making some of the patterns in the treatment allocation easier to identify.

Figure 3.5: Optimal design for Example 3.4.1 for ϕ_1 (*left*) and ϕ_2 (*right*); different colours indicate different treatments

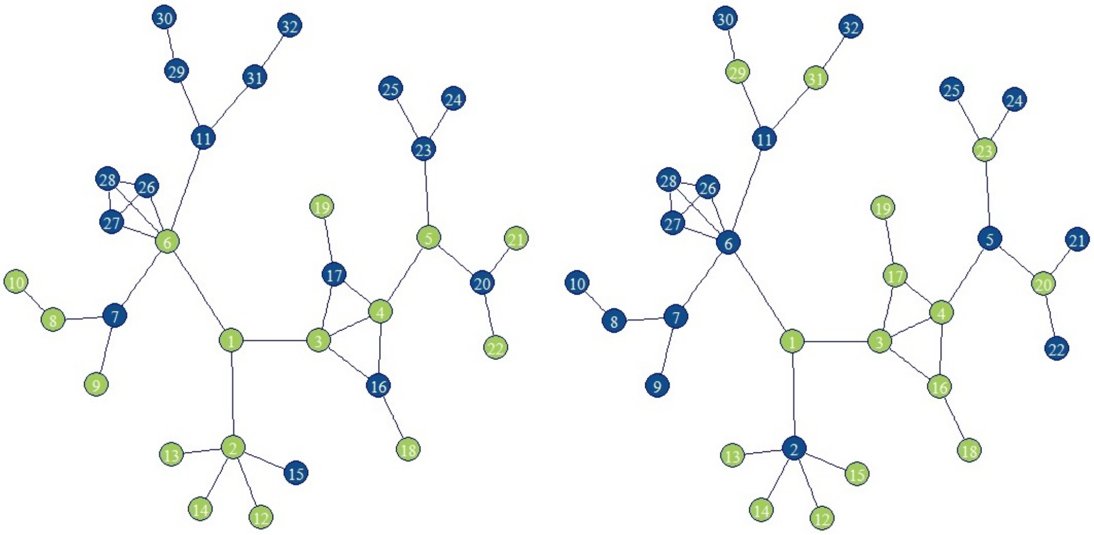
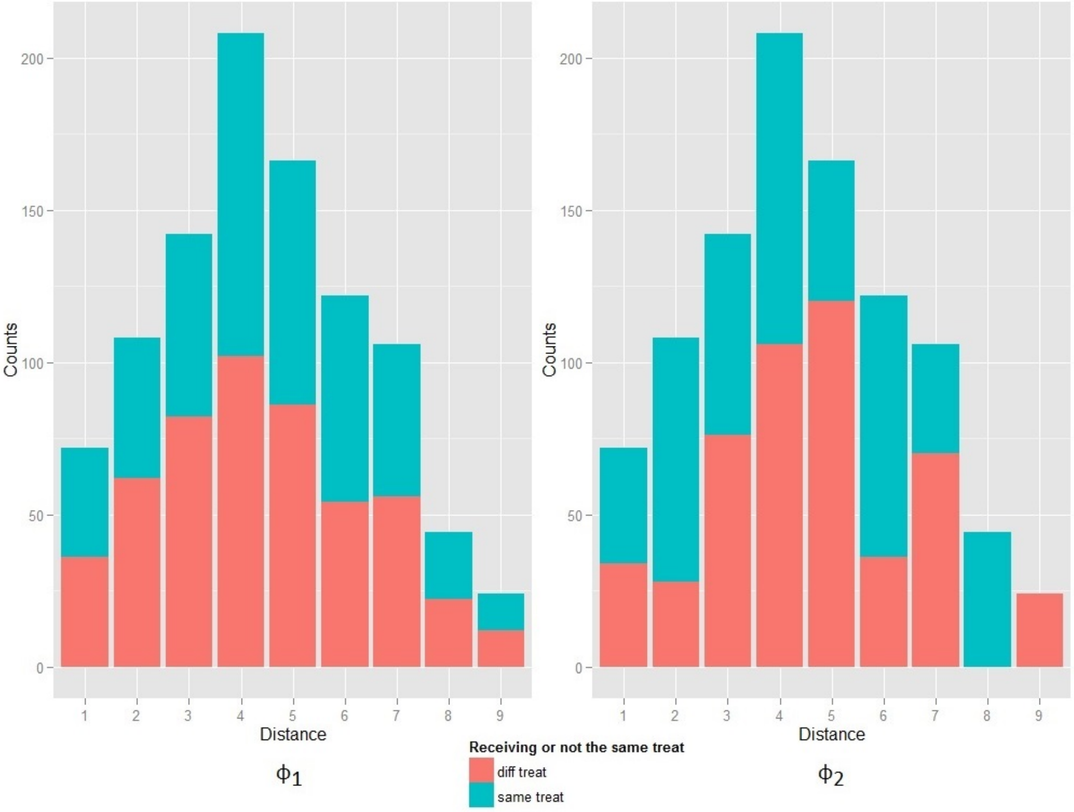


Figure 3.6 illustrates the number of links connecting the same (blue) or different (pink) treatments regarding different distances and different optimality criteria. In particular, for ϕ_1, there are 36 links connecting immediate neighbours receiving the same treatments (as indicated by the 'blue' bar only at distance 1). Note that a mutual link between every pair of units is counted twice. This means that half of the links in the network connect pairs of neighbouring units receiving the same treatment. There are 108 second order connections of which 46 links connect neighbours of distance two that receive the same treatments. Whereas for ϕ_2 there are 38 links connecting immediate neighbours that receive the same treatments and 80 links connecting neighbours of distance two that receive the same treatments.

We provide guidelines for choosing designs on experiments in networks, thereby ensur-

Figure 3.6: Pairs of units of different distances receiving the same (*blue*) or different (*pink*) treatments for ϕ_1 (*left*) and ϕ_2 (*right*)



ing good quality designs that account for the network effects generated by the network under experimentation. The intent of these guidelines is also to speed up the convergence of the search algorithm introduced in Chapter 4, which is developed for finding near-optimal designs in networks that present high degrees of symmetry.

- *for* ϕ_1. The design tends to be balanced with all the treatments being equally allocated to units ($n_1 \approx n_2$), with similar first and second degrees receiving proportionally the different treatments ($l_1 \approx l_2$ and $l_{11} \approx l_{12} \approx l_{22}$ and $l_1^{\langle 2 \rangle} \approx l_2^{\langle 2 \rangle}$ and $l_{11}^{\langle 2 \rangle} \approx l_{22}^{\langle 2 \rangle}$). The optimal function values tend to be very close to the minimum average variance possible for the unstructured case when having independent units under SUTVA ($\sigma^2/n_1 + \sigma^2/n_2 = 2\sigma^2 m/n$ with $n_1 = n_2 = n/2$). The allocation of the treatments to subjects with the same degree tends to be balanced (within each block in the case of block designs, see Chapter 5). Units located at the ends of the networks (e.g. leaves) tend to receive an equal number of the two treatments. Overall there seems to be an interconnection between equal replication of treatments and balanced number of first and/or second order connections.
- *for* ϕ_2. The design is greatly affected by the degree distribution of the units and may be nearly balanced when units have similar degrees. Recall that the degree distribution is formed by counting the number of units by their number of connections. Units forming a clique (complete sub graph) tend to receive

> the same treatment, whereas units located at the ends (e.g. leaves) receive the same treatment, and that treatment is different from the one received from their immediate neighbour who is of higher degree. A crude approximation of the lower bound for ϕ_2 is $1/l$ (where l is the total number of links of the network). For instance for the case of two treatments we have $1/2l_1 + 1/2l_2 = 1/l$ with $l_1 = l_2 = l$. Ideally this bound should be somehow weighted by a quantity related to the degree distribution. One objective for future work is to determine a lower bound on ϕ_2 for the optimal design (if this is possible). In general a balanced design may be near-optimal. However, the degrees of the units given each of the different treatments differ. There also seems to be a pattern underlying the quantities l_{12} and ${l_{12}}^{\langle 2 \rangle}$ under that criterion, where the former decreases as the design gets worse and the latter increases.

Thus in practice, a constructive way to find ϕ_1-optimal designs for massive networks will be to have an equal-replication of the treatments (balanced design) and allocate them to units with equal number of connections (balanced degrees). If additionally m divides n and m divides l, where m, n and l are the number of treatments, units and links respectively in the network, the above will be satisfied exactly, if allowed by the degree distribution of the network. For instance, in the case of comparing two treatments in a network we expect: half of all edges to go to treatment A and half to treatment B; treatments A and B to be assigned to an equal number of units; and the sum of degrees from treatment A to be equal to the sum of edges from treatment B. However, ϕ_2-optimal designs are highly dependent on the particular network structure. Thus for obtaining such designs, we propose to focus on subgraphs of the network identified as having special structures (e.g. complete, tree, star etc). For instance, we would give the same treatment to units belonging to a complete subgraph. Consider the $m = 2$-treatment case. If less connected units, as for instance units located at leaves of a tree-like structure, will receive treatment A, then the hubs that these units are connected to will receive treatment B. Synthesising the design for the whole network, it will be prudent to try to retain an overall balance on average, in the sense that not all cliques across the network will receive the same treatment. However, keep in mind that with high degree heterogeneity expected in real networks, ϕ_2-optimal designs will tend to be unbalanced.

Example 3.4.2. In this example, we consider three different graphs depending on three different graph models, i.e. the random graph (Erdös-Rényi) model, the small-world (Watts-Strogatz) model and the scale-free (Barabási-Albert, preferential attachment) model (for details and a better understanding of graph concepts see Section 2.4). We want to investigate if our observations on the patterns of the optimally allocated treatments are valid for these particular types of networks with system-specific features (see Figure 3.7). Each of the graphs comprise 24 vertices with the number of edges being 30, 48 and 23 corresponding to the random, small-world and scale-free graphs respectively. For these generated graphs, the vertices are connected with the same rewiring probability of 0.1. Table 3.3 provides the individual network's structural properties for each network. We find the optimal designs by exhaustive search for comparing two different (unstructured) treatments for each network (see Figures 3.8 and 3.9).

Figure 3.7: Snapshots of three common types of networks; random, small-world and scale-free graphs - $n = 24$

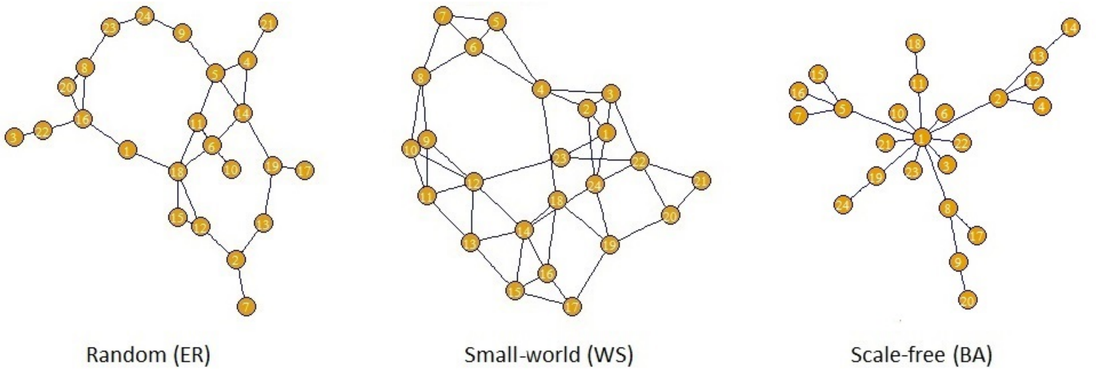
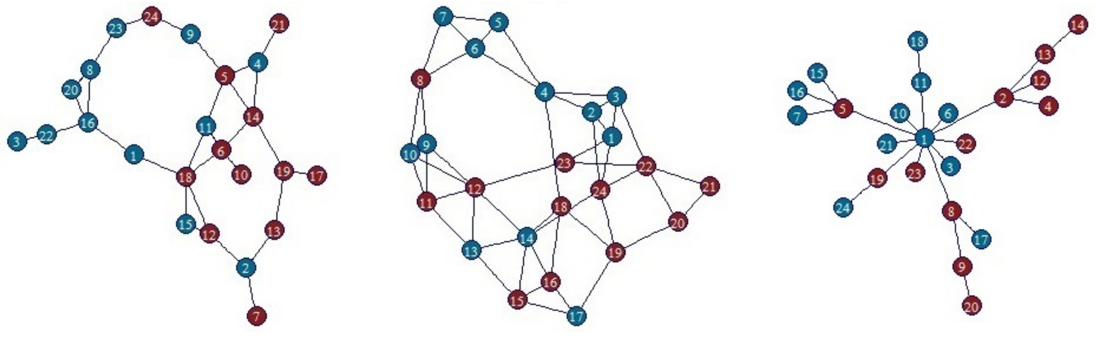


Table 3.3: The general characteristics of the networks of Figure 3.7. For each network we have indicated the number of vertices, the number of edges, the average degree δ, the clustering coefficient $\mathcal{C}$ and the average path length ℓ

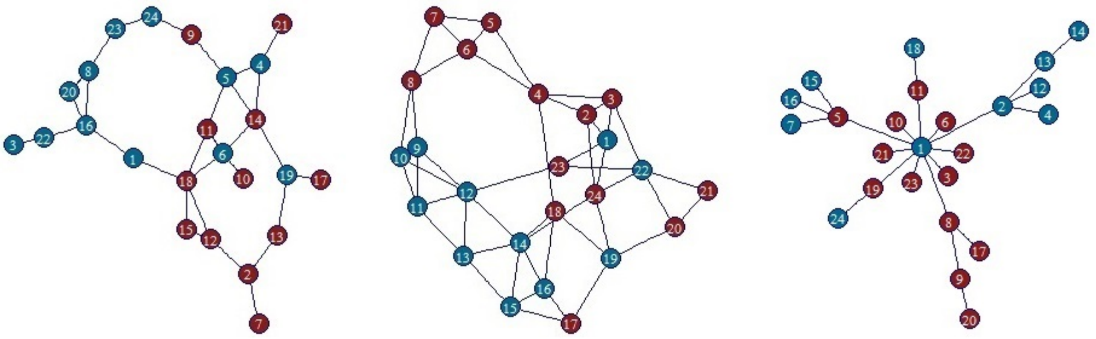
Network	$\lvert V \rvert$	$\lvert E \rvert$	δ	$\mathcal{C}$	ℓ
Random (ER)	24	30	2.5	0.2	3.619
Small-world	24	48	4	0.35	2.636
Scale-free	24	23	1.9	0	2.894

Figure 3.8: Snapshots of three common types of networks, optimal designs for estimating τ_1 (ϕ_1)



By observing the resulting optimal designs for ϕ_1 and ϕ_2, in Figures 3.8 and 3.9 respectively, we can confirm some of the patterns of the optimally allocated treatments. For the random network (first graph) there are no apparent patterns to discern, due to the random degree distribution of the edges (variety of degrees, no highly connected units). The small-world network (middle graph) has high clustering coefficient while the weak connections produce the branching structure that reaches many vertices in a few steps. Under ϕ_1, groups of vertices that are more densely connected to each other compared to the rest of the network tend to receive an equal number of treatments (balanced

Figure 3.9: Snapshots of three common types of networks,optimal designs for estimating $\gamma_1 - \gamma_2$ (ϕ_2)



allocation within groups), while under ϕ_2 the groups of vertices tend to be dominated by the same treatment. This observation is not so clear in this figure but it will become more concrete in Chapter 5, where we investigate in detail the idea of clustering in networks. The scale-free graph (third graph) is characterised by a highly heterogeneous degree distribution. By construction new vertices create connections with existing vertices with probability proportional to their degree (preferential attachment rule). This results in a tree structure (with lack of cycles), with highly connected vertices (hubs). We can observe that some patterns already mentioned for tree structures are repeated here. For example, under ϕ_2, neighbours of units with few social ties tend to receive the same treatment and their immediate neighbours of a larger degree tend to receive a different treatment. Small-world and scale-free networks are very close structurally to many real-life social networks. Having said that, we can benefit from these observed patterns to find good experimental designs for use in practice given the properties of the network topology under experimentation.

3.5 Design efficiency and bias

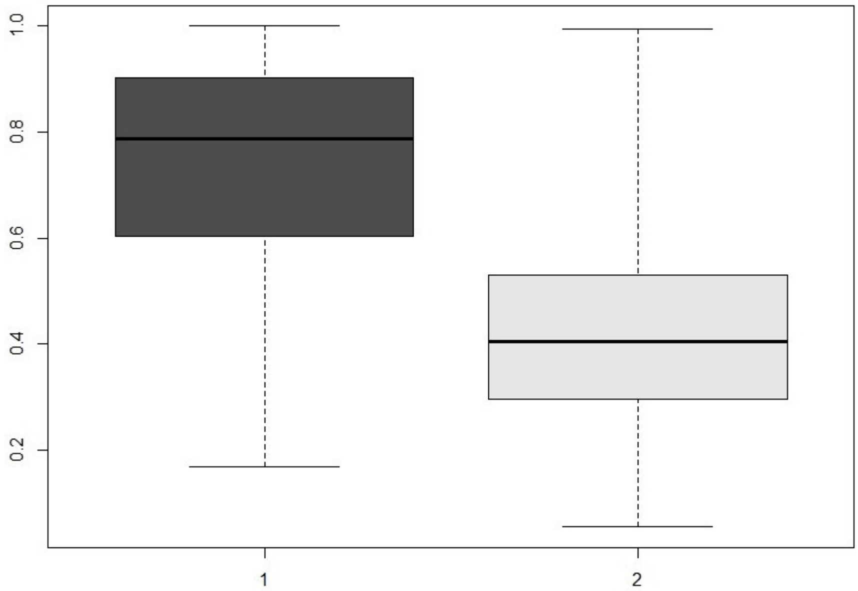
We aim to construct experimental designs on networks that are efficient in estimating treatment and/or network effects and that ideally lead to a small bias of the estimated model parameters. The desirable design properties we are seeking concern the efficiency, i.e. to minimise average, over $1 \leq s < s' \leq m$, of the estimated variance of all pairwise treatment contrasts $\hat{\tau}_s - \hat{\tau}_{s'}$ (that is the L-optimality defined in Section 2.1) and the lack of bias, i.e. we want $\mathbb{E}[\hat{\tau}_s] = \tau_s$ for each treatment s. The experimental designs we obtain in this thesis rely on minimising variances not biases. We obtain the bias with respect to the model parameters and investigate its impact via simulations for an example network. In order to make comparisons with other designs we use the ϕ-efficiency, Eff_ϕ, of some arbitrary design ξ with respect to the (near-) optimal design ξ^*. Recall that this is defined as the ratio $\phi(\xi^*)/\phi(\xi)$, where ϕ is the optimality function.

3.5.1 Efficiencies of randomised designs

We explore if experiments should be randomised, when there is strong evidence of network structure underlying the experimental units. In doing so we compare the optimal designs with network effects to the randomised balanced designs (equal replication), providing evidence that when we randomise, the balanced design is typically not very good. In the majority of cases designs that account for the network structure have higher efficiency than the standard designs.

Revisiting Example 3.2.1 (see Figure 3.1), we obtain all the randomised balanced designs of which there are 462. These are all the possible combinations of 12 subjects comparing 2 treatments, i.e. $n!/((n/m)!)^m = 12!/((12/2)!)^2$ dividing the result by 2 to exclude the symmetry of treatment labels in the design. For instance, we find that our optimal design for this example is $\{1, 1, 1, 2, 2, 2, 1, 2, 1, 2, 1, 2\}$ for optimally estimating τ_1 via criterion ϕ_1, which is equivalent to $\{2, 2, 2, 1, 1, 1, 2, 1, 2, 1, 2, 1\}$. Suppose that we choose a randomised design where we wrongly assume that there is no network effect. We calculate the efficiencies of all possible balanced designs under the LNM compared to the optimal design (when there is a network effect). The results are shown in Figure 3.10.

Figure 3.10: Boxplots of efficiencies calculated for all balanced designs ignoring network effects; efficiency (vertical axis) for the criteria ϕ_1 (*left*) and ϕ_2 (*right*)



The mean optimal function values of all balanced designs are 0.5118 and 0.2547 for ϕ_1 and ϕ_2 respectively. Note that the distributions of the criterion values are positively skewed (median values are 0.4264 and 0.2136 for ϕ_1 and ϕ_2 respectively). The relative efficiencies of the standard designs (i.e. randomised designs) with respect to the optimal LND that account for network effects are $0.3359/0.5118 = 0.66$ and $0.0866/0.2547 = 0.34$ for ϕ_1 and ϕ_2 respectively. Thus the standard randomised designs do not perform well on average under the scenario that the experimental units are connected according to some network structure, especially in the case of ϕ_2. In subsequent sections we provide further evidence that standard designs, e.g. completely randomised designs, are inefficient when experimental units are connected in a network, regardless of the network characteristics (see Chapter 5).

3.5.2 Bias due to model misspecification

In this section we employ the same example (Example 3.2.1) to thoroughly investigate the potential impact of model bias on the design. Under the false assumption that that there is no network effect (i.e. assuming CRM) we calculate the expectation of the bias introduced in the parameter estimates for all possible balanced designs (which are optimal when there is no network effect) in terms of the true unknown parameters $\boldsymbol{\beta} = (\mu\ \tau_1\ \gamma_1\ \gamma_2)^T$ as

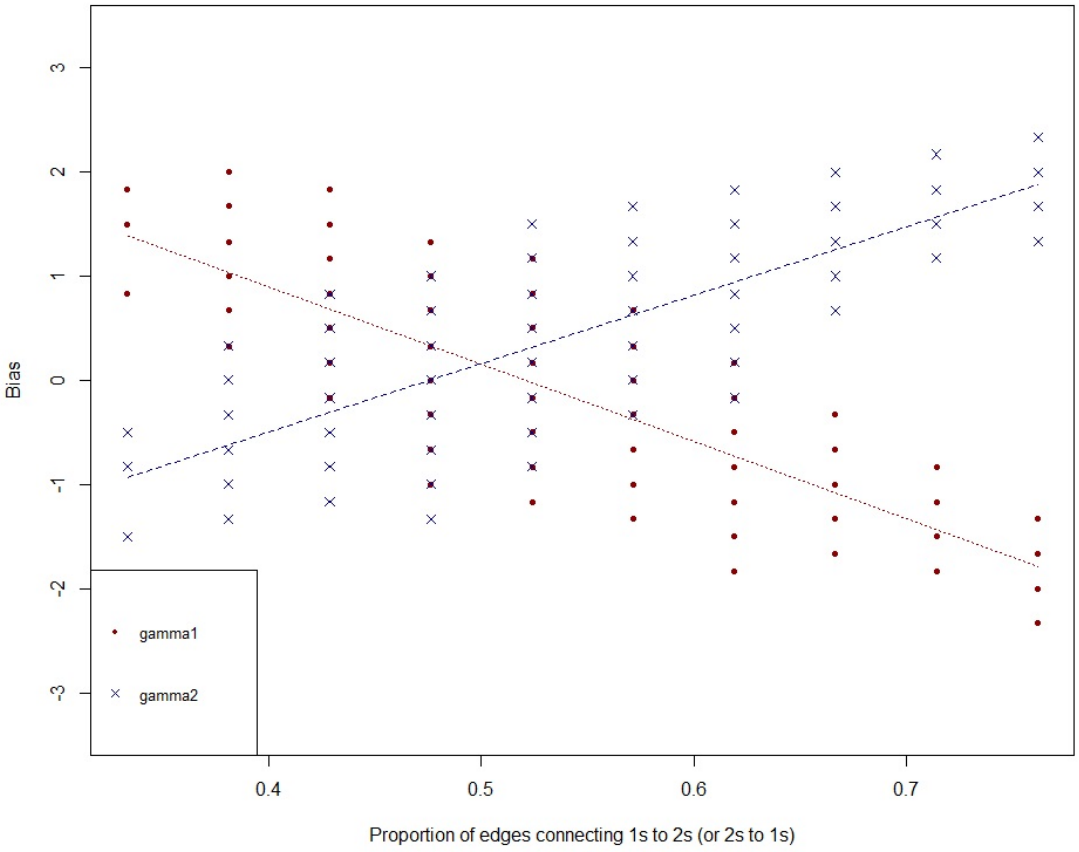
$$\begin{pmatrix} 0 & 0 & 1.91 & 1.45 \\ 0 & 0 & -0.18 & 0.45 \\ 0 & 0 & -1.00 & 0.00 \\ 0 & 0 & 0.00 & -1.00 \end{pmatrix} \begin{pmatrix} \mu \\ \tau_1 \\ \gamma_1 \\ \gamma_2 \end{pmatrix}. \tag{3.19}$$

As shown earlier in Section 3.3, the bias depends on the chosen design. The bias for the direct treatment effects estimator is then given as: $\mathbb{E}[\hat{\tau}_1] - \tau_1 = -0.18\gamma_1 + 0.45\gamma_2$. If the true values of the network effects are zero (so that $\gamma_1 = \gamma_2 = 0$), the balanced design will produce unbiased estimators. In a completely randomised design there is 50% chance of any two units equally receiving treatments 1 and 2. To investigate if the bias from treatment effects results from network effects passed on from 1s (i.e. units receiving treatment 1) to 2s (i.e. units receiving treatment 2) or inversely, we focus on the quantity l_{12} which reflects how many links connect the different treatments. We assume that γ_1 and γ_2 are of similar magnitude, i.e. $\gamma_1 = \gamma_2 = \gamma$. The plot in Figure 3.11 illustrates the bias in treatment effect estimates stemming from network effects against the proportion of edges, which connect pairs of units receiving different treatments. The locations of the plotting symbols are related to the obtained coefficients of the network effects in the bias equation under each design and are dependent on the size of the true γ (see Equation (3.16)). Thus they correspond to the bias of balanced designs (CRDs) for estimating the treatment effects due to the network effects under the assumption that the underlying parameters are both one. In other words, we assume that γ_1 and γ_2 are equal, and without loss of generality are one, i.e. $\gamma_1 = \gamma_2 = \gamma = 1$.

It should be noted that many designs are overlapping and each location of the plotting symbols can represent the bias due to network effects for alternative CRDs. In particular, the number of balanced designs for each proportion of edges (connecting 1s to 2s) is given below:

0.33	**0.38**	**0.43**	**0.48**	**0.52**	**0.57**	**0.62**	**0.67**	**0.71**	**0.76**
4	18	41	82	100	85	67	44	12	9

Figure 3.11: Bias in treatment effects due to network effects



The plot suggests that there is a relationship between the biases due to γ_1 and γ_2, which depends on the proportion l_{12}/l. On average across the randomisation, given the condition γ_1 and γ_2 are of equal size, the bias cancels out (like for instance the carry over effects in cross-over designs which are equal). In order to obtain the least expected bias in the treatment effects, the number of pairs of connected units who receive different treatments should roughly equal half the number of the total edges of the network (see intersection point).

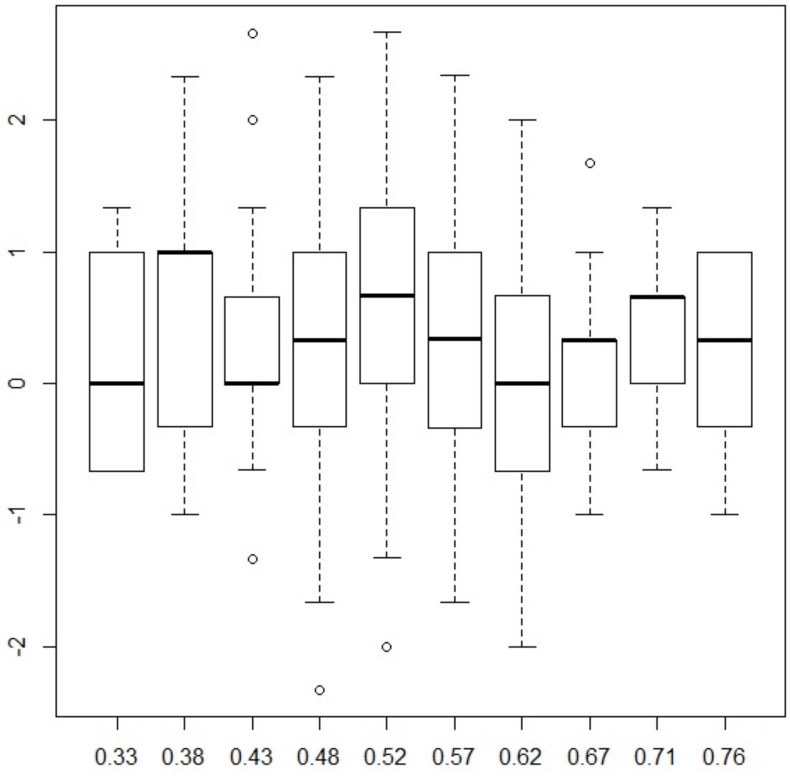
Table 3.4 shows the average bias introduced in treatment effects over all balanced designs corresponding to each proportion. The highlighted row shows that for the proportion of 0.48 the average bias across all balanced designs is the same for γ_1 and γ_2, backing the aforementioned finding. Hence, in practice, an efficient design for estimating with minimum variance the treatment effects on a network will be achieved by half of the mutual links connecting 1s to 2s (if this is possible). In other words, the experimenter should aim towards imposing restrictions on the allocation of treatment combinations such that the number of connected pairs of units receiving different treatments roughly equals half of the total number of edges in the network. Restrictions on the randomisation in such a way enables us to protect the experimental results against bias in treatment effects stemming from potential network effects. For more information about restricted randomisation refer to Mead *et al.* (2012, Ch. 11).

Table 3.4: Expectation of bias of balanced designs at each proportion

l_{12}/l	Bias(γ_1)	Bias(γ_2)
0.33	1.39	-0.94
0.38	1.17	-0.50
0.43	0.83	-0.17
0.48	**0.17**	**-0.17**
0.52	0.00	0.33
0.57	-0.33	0.67
0.62	-0.83	0.83
0.67	-1.00	1.33
0.71	-1.33	1.67
0.76	-1.83	1.83

Figure 3.12 depicts the bias in the estimation of treatment effects due to the network effects over all possible balanced CRDs under the assumption that $\gamma_1 = \gamma_2$ for every proportion of links. The limits of the boxplots are related to the coefficients of the network effects and are dependent on the size of the true γ. Evidently complete randomisation is not a good idea.

Figure 3.12: Bias for treatment effects when $\gamma_1 = \gamma_2$



Additionally, Table A.1 in Appendix A provides the precise information of the coordinates of Figure 3.11 for each proportion of edges (connecting 1s to 2s) as captured by l_{12}/l (with unique entries). By regressing β_{γ_2} on β_{γ_1} and l_{12}/l in R programming language, we obtain the bias due to γ_2, β_{γ_2}, as a function of the bias due to γ_1, β_{γ_1}, plus a constant multiplied by the difference in the proportions. In particular,

$$\begin{aligned}\beta_{\gamma_2} &= \beta_{\gamma_1} + 14\,\frac{l_{12}}{l} - 7\\ &= \beta_{\gamma_1} + \psi\left(\frac{l_{12}}{l} - 0.5\right)\end{aligned}$$

where ψ is a constant. This relationship suggests that the biases differ by a constant that is related to the size of the network or related to an association between the number of units and number of connections (recall that $n = 21$ and $l = 12$). This provides us with an interesting relationship of the biases due to network effects.

3.5.3 Misspecification of the network structure

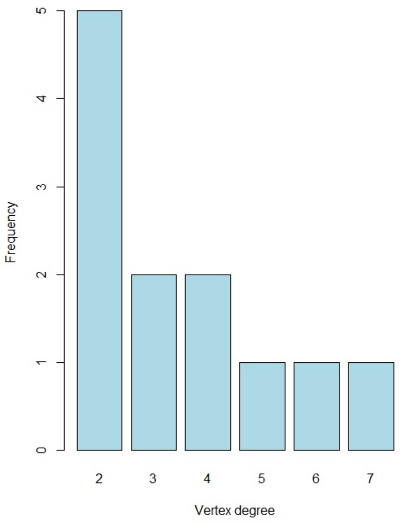
Another important issue is whether it is possible to find designs which are robust to misspecification of the network structure. Given that we generally assume that the number of units is fixed under experimentation, our focus lies on the misspecification of the edges, rather than of the vertices. Firstly, we investigate the robustness of the network after removing some of its edges, where the network's robustness relates to its connectedness and specific properties. Menger (1927) was the first to consider this issue when dealing with the propagation of effects through a network. In particular, he showed (later proved by Harary, 1969) that the minimum number of edges in the network that must be removed in order for two vertices to become disconnected is exactly equal to the number of edge-independent paths between the two vertices (Menger's theorem). By the concept 'edge-independent paths' between two vertices we are refering to all (distinct) paths that have no edges in common. This result triggered further research in graph theory as an attempt to evaluate the robustness of the network when a certain fraction of edges (or vertices) are removed or added in some way (work on this topic is well described in the book by Newman, 2010, p.424-513). Measures like the average degree δ, the clustering coefficient $\mathcal{C}$ and the average path length ℓ (discussed in Section 2.4) are robust performance measures of the network topology to be used for calculating quantities relevant to the connectivity structure of the graph and its specific structural characteristics (Barabási and Albert, 2002).

We explore the robustness of the design when observing the resilience of the network's topology to different changes. The robustness of the design and robustness of network are interrelated. This is intuitively reasonable, since the network topology has a great impact on the properties of the optimal design as we already saw in the previous sections. Thus it is interesting to investigate to what extent the network remains robust against a random or targeted removal of edges. This issue has been especially explored in the epidemiological setting, where there is an underlying diffusion process in the network, such as the spread of a disease. A treatment such as a vaccination of the network members can be regarded as the removal from the network under experimentation of some particular set of edges (or vertices). Some discussion in this direction can be found in the text of Newman (2003). The key question that arises is how robust the design will be against such change.

We showed earlier that the randomised design does not perform sufficiently well in situations when units are connected independently of the exact nature of their connections. Trials suggest that even if we slightly misspecify the network concerning some connections, it is still better to use designs that take into account some structure among units rather than using a random balanced design that completely ignores it.

Revisiting Example 3.2.1, we obtain the degrees of the vertices (the number of edges each has) corresponding to the vertices 1 to n as: $\{2, 4, 3, 6, 2, 2, 4, 2, 5, 7, 3, 2\}$. For example, subject 1 has two friends, while subject 10 has seven friends. Figure 3.13 illustrates the degree distribution of this network. We compare the optimal designs produced under a number of misspecified networks by means of their efficiencies with respect to the optimal design under the true network structure of Figure 3.1. Recall that the optimal function values under the true LNM (with the true network structure) are $\phi_1 = 0.3359$ and $\phi_2 = 0.0866$. The efficiencies of optimal designs under different misspecifications are provided in Table 3.5 (the optimal designs for each misspecified network can be found in Appendix A).

Figure 3.13: Degree distribution of the network of Figure 3.1



The specific cases of network misspecifications we focus on concern edges between: subjects 1 and 5 and similarly 1 and 12, who both have few friends; subjects 1 and 10 and similarly 1 and 4, one of which has many friends while the other only a few; subjects 4 and 10 and similarly 7 and 9, who both have many friends. This particular choice of removal of edges is mainly related to the vertex degree of each unit, a quantity which can be crucial when it comes to propagation of effects in a network. It appears that it makes a big difference in the efficiency for ϕ_2, when removing an edge between a hub (i.e. one with many friends) and an isolated member. Moreover, in the case of removing an edge between subjects both of which have few friends, ϕ_2 tends to be more robust

Table 3.5: Efficiencies of the optimal designs under network misspecification

	Edge(s) removed between units	Figure	ϕ_1	ϕ_2	Eff$_{\phi_1}$	Eff$_{\phi_2}$
(i)	1 and 5	(A.1)	0.3490	0.0884	0.96	0.98
(ii)	1 and 10	(A.2)	0.3428	0.0986	0.98	0.88
(iii)	4 and 10	(A.3)	0.3370	0.0875	1	0.99
(iv)	1 and 5 and 1 and 10	(A.7)	0.3368	0.0994	0.99	0.87
(v)	1 and 5, 1 and 10, and 4 and 10	(A.8)	0.3478	0.1023	0.96	0.84
	Edge(s) included between units					
(vi)	1 and 4	(A.4)	0.3417	0.0875	0.98	0.99
(vii)	1 and 12	(A.5)	0.3406	0.0952	0.98	0.90
(viii)	7 and 9	(A.6)	0.3417	0.0866	0.98	1

than ϕ_1. A possible reason for this is that both units have similar degrees. In general, the efficiencies appear to be high. Moreover, the design accounting for the connections among units performs better than the corresponding random balanced designs. Recall that the mean of the efficiencies of all balanced designs for the two criteria ϕ_1 and ϕ_2 are 0.5118 and 0.2547 respectively. Thus, if we misspecify the network, the design we obtain might no longer be optimal but it would be much better than taking a random balanced design. Therefore, this example is helpful in informing us that we might actually be better off by specifying the network, even if we could be slightly wrong about some of its connections.

The next question we should ask is how much misspecification we can allow for. For this reason, we try removing/adding more than one connection simultaneously and observe if it makes a big difference. We start by removing two edges at the same time, followed by three edges (see Table 3.5). We observe that the efficiencies still remain high. Moreover, we can see that the optimal design that accounts for the network effects is still preferable to a standard (randomised) design. An immediate question that arises is how many edges we can remove before the design becomes worse than a randomised one. As future work, we would like to think about a threshold of misspecification of the network, below which the design remains better than a randomised design. However, bear in mind that our example has only 12 subjects, so if we remove too many edges the connectivity will be destroyed and the social network will collapse. As we pointed out earlier in this section, by removing edges we change the shape and topology of the network; thus the misspecified network may have different properties that define it; for instance, the degree distribution may alter. Table 3.6 illustrates some network measures of the different misspecifications of the network of Figure 3.1. In particular, we obtained the the average degree δ, the clustering coefficient $\mathcal{C}$ and the average path length ℓ for each network. We see that these measures do not substantially vary for the different network misspecifications, which indicates that the network is robust to these changes. In practice, if one is working with a relatively small network, one should have confidence about the structure, as opposed to a larger scale network where we are prone to misspecify the network structure.

Table 3.6: Network measures of the different misspecifications of the network of Figure 3.1; average degree, the clustering coefficient and the average path length

Network	δ	$\mathcal{C}$	ℓ
Original	3.5	0.304	1.863
(i)	3.3	0.313	1.893
(ii)	3.3	0.338	2.045
(iii)	3.3	0.206	1.893
(iv)	3.6	0.311	1.818
(v)	3.6	0.287	1.818
(vi)	3.6	0.384	1.833
(vii)	3.1	0.344	1.818
(viii)	3.0	0.235	1.836

We investigated how the network misspecification affects the optimality of the design, when removing a targeted set of edges from the network. An alternative approach would be to randomly remove edges, with some probability. For this reason, we simulate 100 different networks misspecifying an increasing number of edges: from 1 up to 7. Then we calculate the efficiencies of the optimal designs for those misspecified networks when the true network is the one shown in Figure 3.1. These efficiencies are illustrated in Figure 3.14. An approach for examining design robustness to network misspecification is by removing sequentially an increasing number of edges. This can be performed by randomly selecting and removing sets of vertices. In Figure 3.15 we follow the same approach, but with the additional constraint that the graph will remain connected. In other words, we randomly select and remove sets of vertices in such a way that we do not increase the number of components of the graph. The pattern of efficiencies is very similar to that in Figure 3.14. However, by employing this check, so as to avoid ending up with isolated members, we can discern that there is a slight tendency to reaching higher design efficiencies in the majority of cases as compared to not employing it at all. We see the efficiency decreases when increasing the number of edges that are removed or in other words when we increase the network misspecification, which is true especially for ϕ_2. It is important to note that there is a large variation in the efficiencies even when we misspecify the network for a few edges. Obtaining a large number of simulated misspecified networks and choosing an 'intermediate' design among the obtained designs, could serve as an approach for finding the best design which will be quite robust to misspecification. For a discussion of this approach see Section 7.3.

In general, when we randomly alter the topology of a given a network (keeping the size fixed), its vertex specific features are expected also to change. This can be justified by considering that vertices have originally some topological properties associated with them, such as degree, clustering coefficient, average path length and by reforming the network, deliberately or not, the resulting vertices will no longer be equivalent to the original ones (under a somewhat generalised notion of topological equivalence). However, if we make changes to the edges attached to structurally equivalent vertices (which are sets of vertices with the same external properties), then there may not be

Figure 3.14: Design efficiencies for 100 simulated misspecified networks; removing 1 to 7 edges: Eff_{ϕ_1} (*left*) and Eff_{ϕ_2} (*right*)

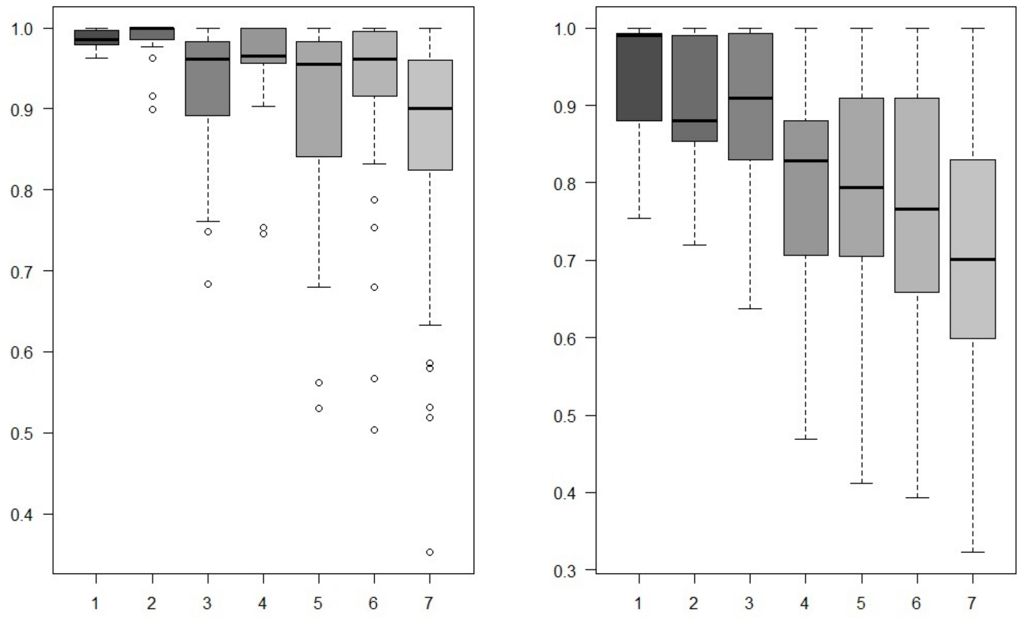
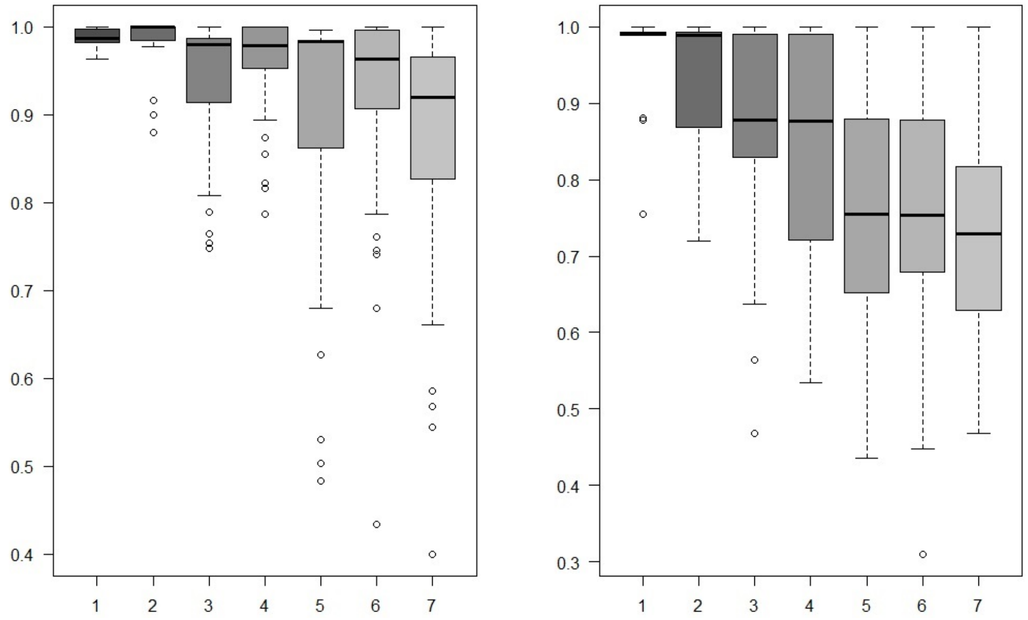


Figure 3.15: Design efficiencies for 100 simulated misspecified networks (connected graph); removing 1 to 7 edges: Eff_{ϕ_1} (*left*) and Eff_{ϕ_2} (*right*)



major topological changes to the network. This issue is discussed in Chapter 4 in the context of network symmetry.

3.6 Simple exchange algorithm

When the exhaustive search to find an optimal design is not possible we implement an approximate method (see discussion in Section 2.3). We develop a simple exchange algorithm for finding near-optimal designs for unstructured treatments. In the first phase, an initial design is generated by randomly allocating a number of treatments to units. The aim is to find the 'best' possible allocation. Then the problem becomes one of optimisation. A simple way that this algorithm could evolve is by exchanging the treatments systematically. Thus the algorithm can be summarised as follows:

Systematic Exchange Algorithm–SE

– Step 0: Enter the adjacency matrix (A) (indicating the relations in the network). Units are labelled from 1 to n.

– Step 1: Generate a random initial design (i.e. random allocation of treatment labels to the units).

– Step 2: Calculate the optimality criterion $\phi_{(0)}$ (ϕ_1, ϕ_2 or any other defined criterion) for the arbitrary design of step 1.

– Step 3: Iterate from $i = 1$ to n to find a better design (where i corresponds to the i-th subject). For the i-th iteration:

  (i) Exchange the treatment applied to subject i with another treatment.

  (ii) Calculate the information matrix. If it is non-singular, compute the new value of the chosen criterion. Otherwise set $i \leftarrow i + 1$.

  (iii) If the new criterion value $\phi(i) \leq \phi(i-1)$ exchange the treatments. This is a provisional solution. Otherwise set $i \leftarrow i + 1$.

– Step 4: Repeat step 3 until no further exchange can be made on the current design and then go to step 5.

– Step 5: Rerun all the above steps for several randomly generated initial designs and return the design with the lowest criterion function value ϕ_{min} (i.e. the best solution X_{opt}). This is the L-optimal design, which will be (if not globally) the locally optimal solution in the numerical (global) optimisation. End.

This means that the algorithm iterates by exchanging the treatments, one at each iteration, subject to improving the design based on a chosen criterion. This criterion is calculated as a function of the information matrix based on a pre-specified model. Thus at each iteration, the procedure combines the provisional design with a systematic exchange of treatments on selected units, forming a new design if this improves the criterion function value. The algorithm terminates if a complete pass of all the vertices yields no further exchanges. It is important to mention that the algorithm starts by generating random designs until a nonsingular starting design is found. Moreover to obtain an efficient final design (and address the problem of becoming stuck in a local optimum) multiple random initialisations are used. Other future work could focus on finding an optimal design via a different algorithm, for instance simplex methods, simulated annealing or colouring algorithms (for further background see Section 2.3).

For the particular problems of interest in this thesis, the demonstrated algorithm appears to be powerful enough and simple to implement. The main advantage is that it has been appropriately adjusted to satisfy the exact needs of the particular design at hand. However, the computational requirements increase proportionally or even more with the size of the problem (i.e. number of units and treatments involved in the experiment). Examples of the construction of near-optimal designs using the algorithm presented here are given in the following chapters.

3.7 Discussion

In this chapter we provided the main ingredients of optimal experimental designs on networks: the response model for a given network interference structure; the optimality criteria appropriately chosen so as to reflect the objectives of the experimenter (either estimating the treatment or network effects); and an iterative algorithm to search across the design space for obtaining an efficient design when an exhaustive search is not possible. We derived the formulae of the optimality criteria and highlighted the impact of the first and second order connections on the optimality of the design and how changes in their values can potentially affect the overall optimal allocation. The expressions of the optimality criteria are generally complicated even for the $m = 2$ treatment case. Heuristically the design for estimating the treatment effects tends to be balanced on the units having a similar number of connections, while for estimating the network effects the design is greatly influenced by the network connections. A useful rule of thumb when constructing designs on networks is to find designs that present (near-) equal replication of the treatments allocated to units with equal first and second order connections. It may also be sensible to consider the allocation of treatments separately in special subgraphs that make up the graph as a whole. Further consideration of this issue will be given in the following chapter.

Other issues we comment upon are the design bias and efficiencies due to the model or network misspecification. After deriving the formula for bias, we give an intuitive feel for this bias and how it ranges with the number of connected pairs of units with different treatments. By means of an example, we explored the bias over a population of randomisations which is non-trivial. A conclusion from that exploration is that the least expected bias in the treatment effect estimates is achieved when we balance the treatments so that the number of pairs of connected units who receive different treatments is roughly half of the number of the total edges. This conclusion provides further evidence on the effectiveness of the treatment allocation patterns and our suggested general guidelines for the experimental design.

From our discussion, so far, it is clear that a statistical model should accommodate the network dependence. We also showed that the higher the level of network misspecification the bigger the loss of robustness of our optimal designs to misspecification, especially under ϕ_2.

Chapter 4

Design and network symmetry

This chapter develops a novel algorithmic approach for speeding up our exploration of the design space for finding an efficient design by understanding and utilising the topology in a network and particularly of its symmetries. The driving force behind this chapter is the paper of MacArthur *et al.* (2007). The concepts that are going to be presented here are used in the literature but sometimes with slightly different definition. For reasons of consistency of this thesis, the definitions have been adjusted appropriately. Section 4.1 reviews the main concepts of network symmetry via graph automorphism groups. In Section 4.2, we sketch an example that combines some common symmetric motifs typically found in large networks (the example is from MacArthur *et al.*, 2008). We detect and utilise those symmetric motifs in an attempt to avoid exhaustive exploration in the design space and to reduce the computational burden. In Section 4.3, we consider an algorithmic approach to obtain optimal designs on networks of high degrees of symmetry. Such an approach can in practice be very efficient for finding designs in real-world networks, which are in general large-sized and highly symmetric, as it can substantially reduce the search time while maintaining the design efficiency at a sufficient level. Towards this direction, we discuss several synthetic and real-world examples.

4.1 Preliminaries on graph automorphisms

In mathematics, symmetry is the invariance of an object's properties under a set of transformations. For instance, a rotation of a circle about its center leaves the circle unchanged and preserves distances for any points of the circle. Symmetry in networks entails invariance of network adjacency under permutations on a vertex set. Most definitions on graph automorphisms and structural equivalence are adapted from Harary (1969). A good review of complex networks and symmetry can be found in the work of Garlaschelli *et al.* (2010).

Let $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ be a graph with vertex set $\mathcal{V}$ and edge set $\mathcal{E} \subset \mathcal{V}^2$, where two vertices are said to be adjacent when there is an edge between them (see Chapter 2). Intuitively an automorphism on a graph is a mapping that relabels the network vertices so that the edges among the vertices appear unchanged. This relabelling is called a permutation and is given by a function which is a bijection from $\mathcal{G}$ to itself (that is $\mathcal{V}$ to $\mathcal{V}$ and $\mathcal{E}$ to $\mathcal{E}$). Hence, a graph automorphism is a structure-preserving permutation of the vertices that does not affect network adjacency. This means that if two vertices v_1 and v_2 are joined by an edge, so are their images $\eta(v_1)$ and $\eta(v_2)$ under the permutation η. We can re-phrase the above using the following Definitions $1-3$.

Definition 1. A vertex *permutation* η, acting on a finite set $\mathcal{V}$, is a bijective mapping $\eta : \mathcal{V} \mapsto \mathcal{V}, v \mapsto \eta(v)$.

Definition 2. Two graphs $\mathcal{G}_1 = (\mathcal{V}_1, \mathcal{E}_1)$ and $\mathcal{G}_2 = (\mathcal{V}_2, \mathcal{E}_2)$ are *isomorphic*, $\mathcal{G}_1 \approx \mathcal{G}_2$, if there exists a permutation $\eta : \mathcal{V}_1 \mapsto \mathcal{V}_2$ such that for all vertices $v, v' \in \mathcal{V}_1$ so that $vv' \in \mathcal{E}_1$ if and only if $\eta(v)\eta(v') \in \mathcal{E}_2$. (Recall that vv' is the edge between vertices v and v' and likewise $\eta(v)\eta(v')$ is the edge between $\eta(v)$ and $\eta(v')$). In other words, $\eta(\mathcal{G}_1) \approx \mathcal{G}_2$ means that $\mathcal{G}_1$ and $\mathcal{G}_2$ are topologically equivalent).

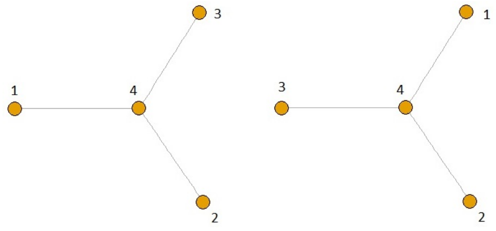
Graph isomorphism is the problem of testing whether two graphs are the same. The *graph automorphism problem* is the analysis of the isomorphisms of a graph with itself or in other words the problem of testing whether a graph has a nontrivial automorphism (Harary, 1969). Note that every graph has a trivial symmetry (the identity) that maps each vertex to itself.

Definition 3. An *automorphism* of a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ is a self-isomorphism (from $\mathcal{G}$ onto itself), i.e. $\eta : \mathcal{G} \mapsto \mathcal{G}$.

Figure 4.1, for example, illustrates a graph automorphism which is obtained by swapping the vertices' labels 1 and 3 of a star graph, through a bijection η; $\eta(1) = 3,\ \eta(2) = 2,\ \eta(3) = 1,\ \eta(4) = 4$. Another simple way to write the same permutation is $\eta = (1\ 3)$, which is called cycle-notation where vertex 1 goes to vertex 3 and 3 goes back to 1. Summarising the domain and image of this function η, this particular permutation can be represented by

$$\eta = \begin{pmatrix} 1 & 2 & 3 & 4 \\ 3 & 2 & 1 & 4 \end{pmatrix}.$$

Figure 4.1: Graph automorphism corresponding to the relabelling of 1 by 3, 3 by 1



This relabelling constitutes an automorphism and the adjacencies between vertices in the two graphs are identical. For this particular network, we can interchange vertex labels 1, 2 and 3 in any way and they still stay adjacent to 4. The set of all automorphisms, including the trivial one (that moves no labels at all), is called the automorphism group of the graph (see Definition 4). Any automorphism of a given graph is a permutation, since it permutes the vertices of the graph and can therefore be represented using permutation matrices. The permutation matrix P_η corresponding to the permutation η is

$$P_\eta = \begin{pmatrix} 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}.$$

Recall that the information of a graph is contained in the adjacency matrix. Thus we can re-write Definition 2 using the adjacency matrices of the corresponding graphs (Definition 2*).

Definition 2*. Two graphs $\mathcal{G}_1 = (\mathcal{V}_1, \mathcal{E}_1)$ and $\mathcal{G}_2 = (\mathcal{V}_2, \mathcal{E}_2)$ are isomorphic, $\mathcal{G}_1 \approx \mathcal{G}_2$, if there exists a permutation matrix P_η such that $A_{\mathcal{G}_1} = P_\eta A_{\mathcal{G}_2} {P_\eta}^T$.

For the example of the star network the corresponding adjacency matrix, where vertices 1 and 3 are structurally equivalent, is unchanged after exchanging its 1st and 3rd row, and its 1st and 3rd column. In doing so, we are not interchanging the identity of 1 and 3, which still represents the original vertices, (for instance two particular persons in a social network). In other words, if $A_{\mathcal{G}_1}$ and $A_{\mathcal{G}_2}$ correspond to the left and right graphs of Figure 4.1 respectively, then we have that $A_{\mathcal{G}_1} = P_\eta A_{\mathcal{G}_2} {P_\eta}^T$.

Definition 4. The *automorphism group* of $\mathcal{G}$, denoted by $Aut(\mathcal{G})$, is the set of all permutations of the vertex set that preserves adjacency (all automorphisms), i.e. $Aut(\mathcal{G}) = \{\eta : \eta(\mathcal{E}) = \mathcal{E}\}$.

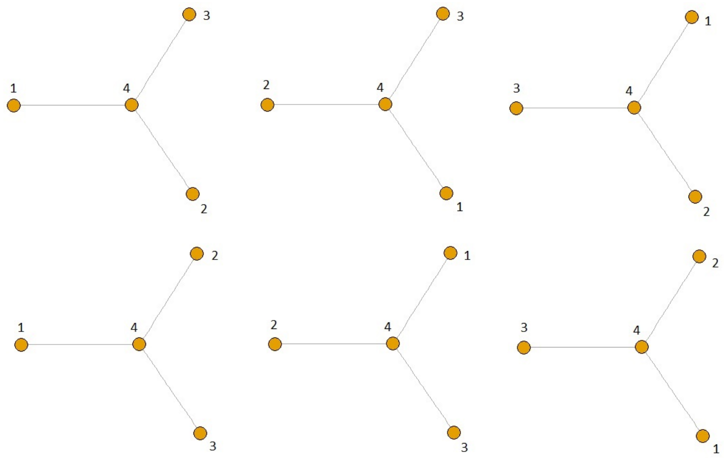
From Definition 4, it follows that $Aut(\mathcal{G})$ comprises all adjacency-preserving bijections of $\mathcal{V}$ formed under composition of permutations. We can easily identify all the automorphisms of the previous example with the star graph. There are 6 (= 3!) automorphisms as found by considering all possible permutations of the vertex labels 1, 2 and 3 (which do not alter the adjacencies). Figure 4.2 illustrates all six automorphisms, $Aut(\mathcal{G})$, of the star graph.

The automorphisms also define a relationship between the vertices of the graph: two vertices are *structurally equivalent* (Wasserman and Faust, 1994) if there is an automorphism taking one to the other. For our example, vertices 1 and 3 are structurally equivalent since there is an automorphism (1 3), which takes 1 onto 3. The sets of structurally equivalent vertices are called vertex orbits; in the example there are two (vertex) orbits $\{1; 2; 3\}$ and $\{4\}$.

Definition 5. The equivalence classes of the vertices of a graph $\mathcal{G}$ under the action of the automorphisms are called *vertex orbits* and they are defined as $orb_{\mathcal{G}}(P, \mathcal{V}) = \{P(v) : v \in \mathcal{V}\}$ where $P(v) = \{\eta(v) : \eta \in Aut(\mathcal{G})\}$ is called the orbit of vertex v.

In simple terms, the orbit of a vertex v is the set of vertices where v can be mapped to under some automorphism. Therefore, the automorphism partition of the vertex

Figure 4.2: All six automorphisms of the star graph of Figure 4.1, $Aut(\mathcal{G})$



set $\mathcal{V}$ of a graph $\mathcal{G}$ is the set $orb_{\mathcal{G}}(Aut(\mathcal{G}), \mathcal{V})$ of the orbits of its automorphism group. In our indicative example of the star graph, all vertices except the central one are in the same orbit, which renders them structurally equivalent because they have the same set of neighbours. Permutations of structurally equivalent vertices lead to exactly the same topology and are therefore automorphisms (exact symmetries) of the graph. Note that groups of equivalence vertices can be called *symmetric motifs*. In other words, a symmetric motif is a synthesis of more than one vertex orbit (Milo *et al.*, 2002; Alon, 2007). These symmetric subgraphs occur much more frequently than would be expected in a similar random network. We can now define the notion of network symmetry of a graph.

Definition 6. A network is said to be *symmetric* if its underlying graph has a nontrivial automorphism group; otherwise it is said to be asymmetric.

The *symmetry breaking* problem concerns the decomposition of the vertices of the graph into its collection of orbits of the automorphism group. Vertices on the same orbit may be permuted without altering the network structure and are indistinguishable from one another. This is a recent area of research (MacArthur *et al.*, 2008; MacArthur and Sánchez-García, 2009; Xiao *et al.*, 2008) which deals with real-world networks, since traditionally, the graph automorphism problem has only been dealt with for specific classes of graphs generated according to deterministic rules (see, e.g., Harary, 1969).

Adopting the approach of MacArthur *et al.* (2007) we can quantify the degree of symmetry in a network by the means of structural repetition of its equivalent vertices, calculated as

$$r_{sym}(\mathcal{G}) = 1 - \frac{|orb_{\mathcal{G}}(Aut(\mathcal{G}), \mathcal{V})| - 1}{|\mathcal{V}|} = 1 - \frac{n_{\mathcal{O}} - 1}{n}, \tag{4.1}$$

where $n_{\mathcal{O}}$ denote the number of vertex orbits, i.e. the vertices that play identical structural role, and n is the total number of vertices in the graph. Note that $0 < r_{sym} \leq 1$: the smaller the value of r_{sym} the more the network is constructed from structurally unique elements; while the larger the value the more the network is constructed from repetition of structurally identical elements and therefore the higher the degree of network symmetry. This measure is a useful yardstick for the algorithmic approach we introduce in Section 4.3.

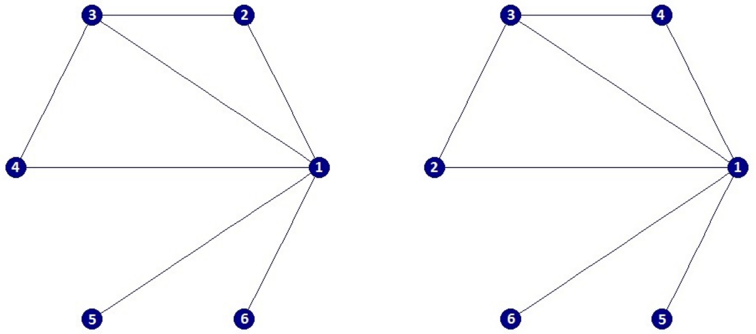
Graph automorphisms have been used to simplify the topology of real networks by collapsing redundant information and obtaining *network quotients* (MacArthur *et al.*, 2007; MacArthur and Sánchez-García, 2009). The network quotient is simply the decomposition of this network's automorphism group to its distinct symmetric motifs. The quotients of real networks are found to preserve various structural properties (MacArthur and Sánchez-García, 2009). They are therefore a reduction or simplification of the original network. In this thesis we focus on a part of the resulting network quotient, which we call the skeleton, which is asymmetric and focuses on the fixed and distinct vertices (omitting the vertex orbits).

Definition 7. The *skeleton* of a graph consists of vertices with vertex orbit of size one.

In the next section, we describe how the skeleton of a given graph can be taken into account in the design process. This approach initially requires finding the vertex orbits. In order to obtain the vertex orbits of a graph we use the graph isomorphism testing program *nauty* (which is short for No AUTomorphisms, Yes?), by McKay (1981). *nauty* is a set of very efficient C language procedures for determining the automorphism group of a graph (which takes only a few seconds for networks with hundreds of vertices) and it is one of the most powerful and best known programs (publicly available at `http://cs.anu.edu.au/~bdm/nauty/`). Up to date information can be also found in the article by McKay and Piperno (2014). The required time for decomposing the network to its symmetric motifs and its skeleton for some thousands of vertices takes between 2 and 6 seconds using *nauty* on a standard computer. However, there is an increasing number of modifications of this algorithm, examples of which include *bliss* (Darga *et al.*, 2004) and *saucy* (Junttila and Kaski, 2007). For sparse and small-sized networks considered in this chapter, *nauty* requires less than a second to obtain the vertex orbits of a graph. Recall that a sparse network is one in which the density of the edges that are actually present in the network relative to all possible edges approaches zero as the number of total vertices approaches infinity, or simply there are many zeros in the adjacency matrix. Note that real-world networks such as friendship networks are usually regarded as sparse. However, for regular graphs such as a lattice network this would not be the case. For a review of the graph automorphism problem from the practical point of view with an extensive description of the recent advances refer to McKay and Piperno (2014) and references therein. We focus on the operation of *nauty*, which provides the orbits of the automorphism group and the size of the automorphism group. This is performed by making use of the structural information of the graph as defined through the degree partition. This operation is related to incrementally refining a partition by using iteratively some degree information. The following example serves as a guide for understanding this idea after summing up all the previous definitions.

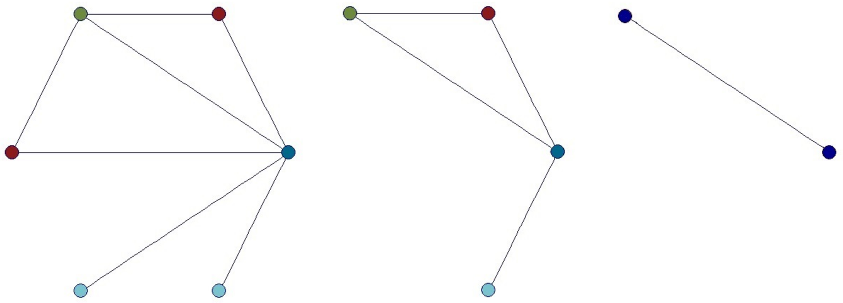
Revisiting Example 2.4.1, we consider the graph $\mathcal{G}$ as illustrated in Figure 4.3(a). The automorphism group of $\mathcal{G}$ is a permutation group acting on $\mathcal{V}$. If $\eta \in Aut(\mathcal{G})$, then η preserves adjacency and hence sends the neighbours of v to the neighbours of $\eta(v)$. Therefore, if η sends v to v', then v and v' have the same degree (the same number of neighbours). Figure 4.3(b) shows a graph automorphism which is obtained by swapping the vertex labels 2 with 4 and 5 with 6, through the bijection $\eta = (24)(56)$. Recall that this cycle-notation means that η maps vertices 2 to 4 and 5 to 6 as well as 4 to 2 and 6 to 5. This relabelling constitutes an automorphism as the set of edges remains the same. Note that (24) and (56) are two automorphisms, and when combined one after the other they also make an automorphism.

Figure 4.3: (a) Original graph; (b) a graph automorphism



Recall that the automorphisms also define an equivalence relationship on the vertices of the graph: two vertices are equivalent if there is an automorphism taking one to the other. The sets of equivalent vertices are called orbits; in the example there are four orbits: $\{1\}, \{2; 4\}, \{3\}$ and $\{5; 6\}$ with two fixed points as shown in Figure 4.4(a). Hence, the graph contains symmetries and we can quantify the degree of symmetry by calculating the ratio $r_{sym} = 0.5$. A degree of symmetry of 50% implies that the graph contains a good amount of structural redundancy, which indicates that exploiting network symmetry is beneficial for reducing combinatorial searches in the design space. Moreover, Figures 4.4(b) and 4.4(c) illustrate the quotient and skeleton respectively of the corresponding graph.

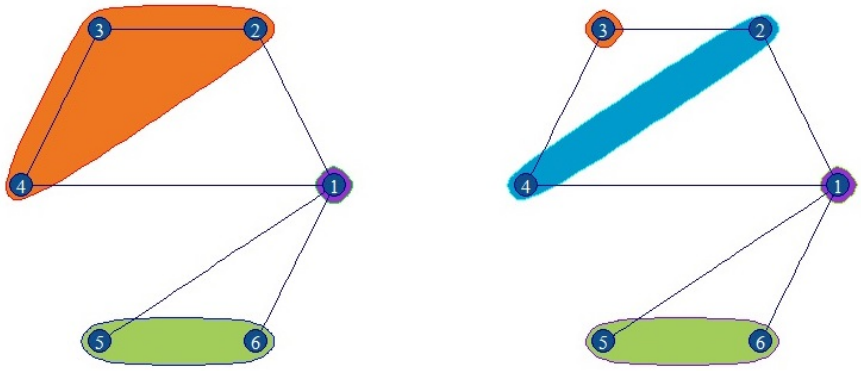
Figure 4.4: (a) Orbits are indicated by different colours; (b) quotient graph; (c) skeleton



To better understand the partition refinement as implemented by *nauty*, consider the slightly altered graph of Figure 4.3(a), by removing the edge between vertices v_1 and v_3.

The vertex orbits are again $\{1\}, \{2; 4\}, \{3\}$ and $\{5; 6\}$, which are found with a two-stage degree partition between the vertices. In particular, we have that the initial partition, P_0, consists of the vertices of the entire graph, i.e. $P_0 = \langle\{v_1, v_2, v_3, v_4, v_5, v_6\}\rangle$. Then ordering the vertices by degree, we compute partition $P_1 = \langle\{v_2, v_3, v_4\}, \{v_1\}, \{v_5, v_6\}\rangle$ (with first degrees $\langle d_1, d_2, d_3\rangle = \langle 2, 4, 1\rangle$). In other words vertices 2, 3 and 4 have each two edges, while vertex 4 has 4 edges and 5 and 6 have each one edge. Figure 4.5(a) illustrates this degree partition with different colours indicating the different degrees. Then the vertices are further distinguished into more restricted cells based on the degrees of those within each partition. In our example, this partial ordering of the vertices by degree will take the form $P_2 = \langle\{v_1\}, \{v_2, v_4\}, \{v_3\}, \{v_5, v_6\}\rangle$. This is because the degree of vertex v_3 within the partition P_1 differs from v_2 and v_4. We can calculate $deg(v, P) = \langle \boldsymbol{d}\rangle$ where $d = |\{u \in P | (u, v) \in \mathcal{E}\}|$, $\forall d \in \boldsymbol{d}$, where $\boldsymbol{d}$ is a vector of degrees that correspond to partition P. Thus $deg(v_3, P_1) = \langle 2, 0, 0\rangle$ which is different from $deg(v_2, P_{1)}) = deg(v_4, P_1) = \langle 1, 1, 0\rangle$. In other words vertex 3 is separated from vertices 2 and 4. Partition P_2 is illustrated in Figure 4.5(b).

Figure 4.5: (a) Partition P_1; (b) Partition P_2



4.2 Symmetry breaking and design

As recent research has revealed, real-world networks are richly symmetric (MacArthur and Anderson, 2006; MacArthur *et al.*, 2007). Their inherent complexity can be simplified to an extent by understanding the structural features in some quantitative way and/or by excluding structural redundancy as captured by the structural repetition of symmetric motifs. Detecting and utilising network symmetry proves to be very useful in accelerating the search for the optimal design in the design space. We make some conjectures based on several examples that provide a fruitful road-map for our future work in this area. Conjecture 1 is related to the linear network effects model (LNM) and the circumstances under which it cannot be implemented. Conjecture 2 is related to the symmetry breaking and design problem.

Conjecture 1. Let $\mathcal{G}$ be a graph and $Aut(\mathcal{G})$ be the set of its automorphisms. If the vertices of the graph can be partitioned into two or fewer vertex orbits under the group of automorphism, then the LNM cannot be fitted (due to the singular information matrix).

Conjecture 1 summarises what is already demonstrated in Section 3.3. For the special graphs, i.e. complete, star and ring networks (described in Section 2.4), the vertices of the corresponding graph can be partitioned into two, one and two vertex orbits respectively. Thus the LNM cannot be applied in these cases due to the singular information matrix. For instance, a k-star network has two orbits, k vertices of degree one that fall in one orbit of size k and the central one of degree k that falls in another orbit of size one. Similarly the complete and ring networks have one orbit each (because given any pair of vertices u and v, we can find an automorphism sending u to v). Another interesting observation is that the information matrix tends to be singular in network with a high degree of symmetry.

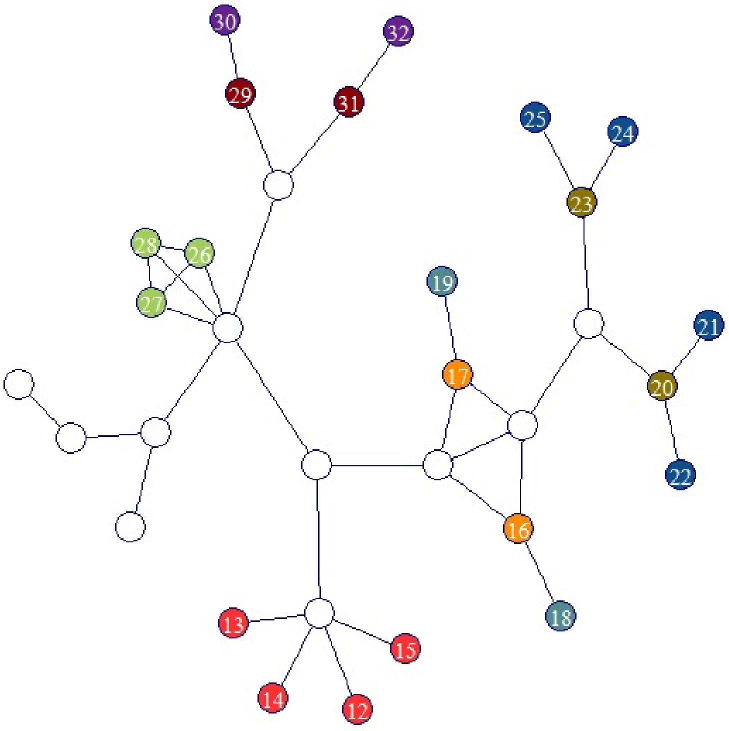
Conjecture 2. Let A be the adjacency matrix of a given network and let $\xi^*_{skeleton}$ be the optimal design under ϕ_1 corresponding to the skeleton. Then, there is an optimal design of the original network under ϕ_1, which contains $\xi^*_{skeleton}$.

Thus we conjecture that the optimal design for the case of ϕ_1 includes the optimal design on the skeleton, where ϕ_1 corresponds to the L-optimality criterion for minimising the average variance of all pairwise differences of treatment effects (see Chapter 2). The useful aspect of this conjecture is that we can obtain an (near-) optimal design to the original graph by decomposing it to subgraphs resulting in a reduction of the computational burden to find a good design. Recall that the skeleton is defined as the maximum asymmetric subgraph of the original network and can be considerably smaller than the original network. This finding is only a conjecture based on a large number of trial networks of different shapes and sizes. However, we could not prove this claim but we provide supportive evidence via some illustrative examples. From our experience, there is no counterexample to discredit this. We explore the benefits of this conjecture algorithmically and leave its proof for future work.

For the case of ϕ_2 the introduction of a skeleton is not adequate and a different definition of the skeleton may be more appropriate potentially related to a weighted quotient graph. However, we can still get highly or moderately efficient designs in one tenth of the normal time when utilising the symmetries of a graph. There seems to be a close connection between the first design criterion and the graph automorphism, which leads us to the point of calling them ϕ_1-automorphisms. In the future, more research should be devoted to defining ϕ_2-automorphisms, taking into account that ϕ_2 is especially influenced by the second order connections (see Section 3.4).

Revisiting Example 3.4.1, we illustrate the symmetry breaking problem and how it is considered in the design process. Our interest here lies in designs for the comparison of two unstructured treatments. Figure 4.6 depicts a network presenting some common symmetric motifs (star, complete graph, tree-structure etc.). The example network was taken from the paper of MacArthur *et al.* (2008). The different colours of the vertices in this figure indicate the different vertex orbits. The white vertices correspond to the fixed vertices for which the orbit is of size one. The skeleton comprises these white nodes and therefore is asymmetric, which means that any exchange on its labels will alter the connectivity structure. Quantifying the degree of symmetry, as defined in Equation (4.1), we have that $r_{sym} = 0.4375$ ($n_{\mathcal{O}} = 19$), and therefore the degree of network symmetry is approximately 44%. Moreover, $|Aut(\mathcal{G})| = 4608$. We find the (globally) optimal designs by an exhaustive search within the design space $\Xi_{\{n,m\}} = \Xi_{\{32,2\}}$, where $|\Xi| = m^n$. The computational time required to find the optimal design for this 32-subject network was estimated to exceed 20 days with a conventional computer (Intel Core $i5$ processor running 64 bits Windows 7 operating system at 1.90 GHz). For this reason Iridis 4 was used and the optimal designs for ϕ_1 and ϕ_2 were found (with running time being approximately 135 hours). For more details about Iridis 4 refer to the computational note in Section 2.3.

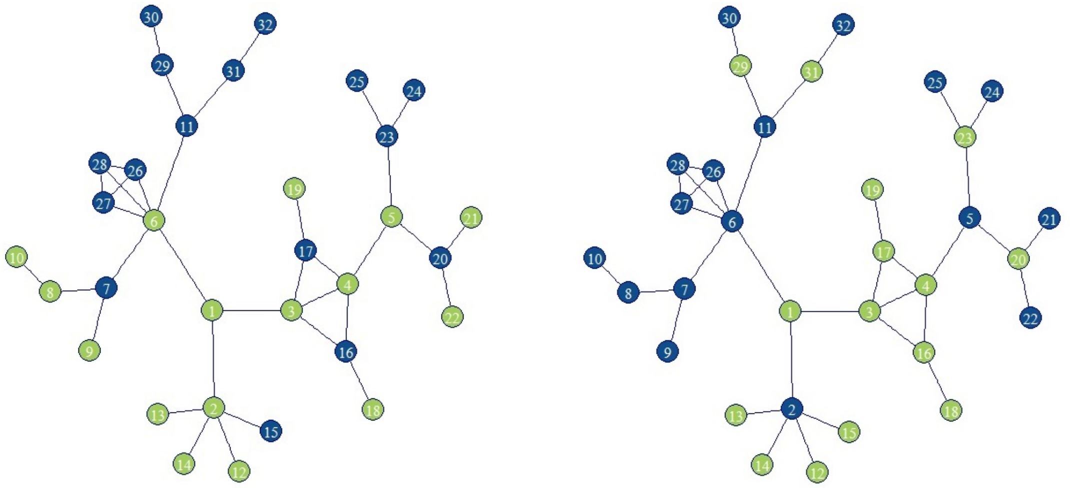
Figure 4.6: Network of size $n = 32$. Different colours indicate different orbits, with white corresponding to the skeleton



The exhaustive search for the optimal designs was achieved with a complete enumeration using the binary number system. More specifically given n (number of units) we obtain the i-th allocation (corresponding to the i-th row of a $2^n \times n$ matrix of all possible combinations of zeros and ones of length n), which is the binary representation of the integer i. Therefore, the algorithm generates step-by-step the binary representations of the sequence $1 : 2^n$. Comparisons are made to the consecutive designs (evaluation of each row separately) and then the algorithm returns the best. Recall that the treatments are equivalent with respect to relabelling, in the sense that one sequence of treatment labels can be obtained from the other one by relabelling. Thus, this symmetry of labels reduces the search time in the design space and as a result the algorithm concludes at $2^{(n-1)}$.

The optimal designs under LNM are found and illustrated in Figure 4.7, with optimal function values $\phi_1{}^* = 0.1250$ and $\phi_2{}^* = 0.0237$ respectively. Note that the optimal design for ϕ_1 is not unique as opposed to that of ϕ_2. There can be more than one ϕ_1-optimal design in relation to the graph automorphisms. More discussion on this issue will be given later in this chapter. The optimal allocations for ϕ_1 and ϕ_2 are $\{0,0,0,0,0,0,1,0,0,0,1,0,0,0,1,1,1,0,0,1,0,0,1,1,1,1,1,1,1,1,1,1\}$ and $\{0,1,0,0,1,1,1,1,1,1,1,0,0,0,0,0,0,0,0,0,1,1,0,1,1,1,1,1,0,1,0,1\}$ respectively.

Figure 4.7: Optimal designs ϕ_1 (*left*) and ϕ_2 (*right*); different colours indicate different treatments



Our aim is to find the optimal design (or a near-optimal design) on this network of high degree of symmetry faster. For this reason we find the optimal designs on the skeleton of the graph, which corresponds to the white vertices (omitting all the symmetric motifs of the graph) in Figure 4.6. The processing time (or CPU time) in seconds (secs) required to solve both optimisation problems (corresponding to the two different criteria) is less than 2 seconds. Figure 4.8 illustrates the optimal designs for this subnetwork under each design criterion. The optimal function values for the skeleton are $\phi_1{}^*_{sk} = 0.3668$ and $\phi_2{}^*_{sk} = 0.1265$ respectively. Notice that there are four optimal design for ϕ_1 (see Figure 4.9), whilst in the case of ϕ_2 there is only one.

Considering the optimal design on the whole network and extracting the design on the skeleton, a key questions is how good this sub-design is compared to the optimal design obtained on that skeleton. To make this comparison, we put our focus on the treatment allocation on the vertices of the skeleton of the 32-subject network of Figure 4.7. The values of the criteria for the skeleton are $\phi_1 = 0.6843$ and $\phi_2 = 0.3944$ as opposed to optimal function values, which were previously found to be $\phi_1{}^*_{sk} = 0.3668$ and $\phi_2{}^*_{sk} = 0.1265$ respectively as illustrated in Figure 4.8. The design efficiencies with

Figure 4.8: Optimal designs for ϕ_1 (*left*) and ϕ_2 (*right*) for the asymmetric network skeleton

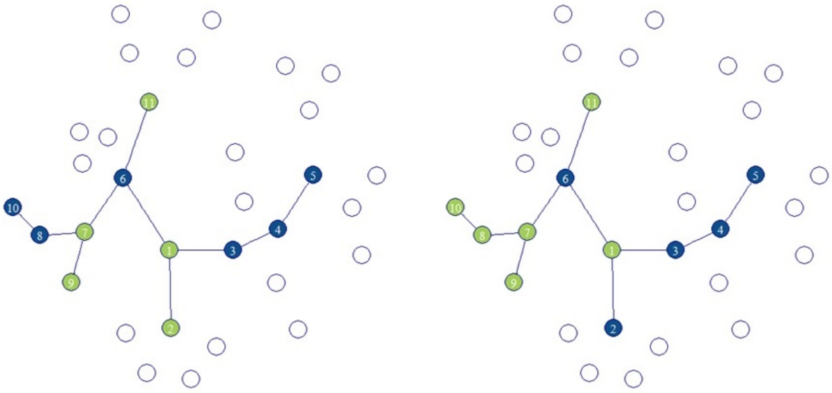
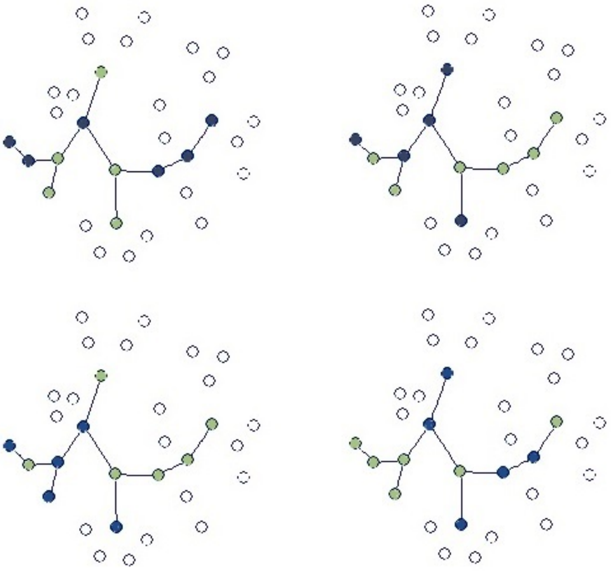


Figure 4.9: All optimal designs for ϕ_1 for the asymmetric network skeleton



respect to the optimal designs on the skeleton, under ϕ_1 and ϕ_2, are

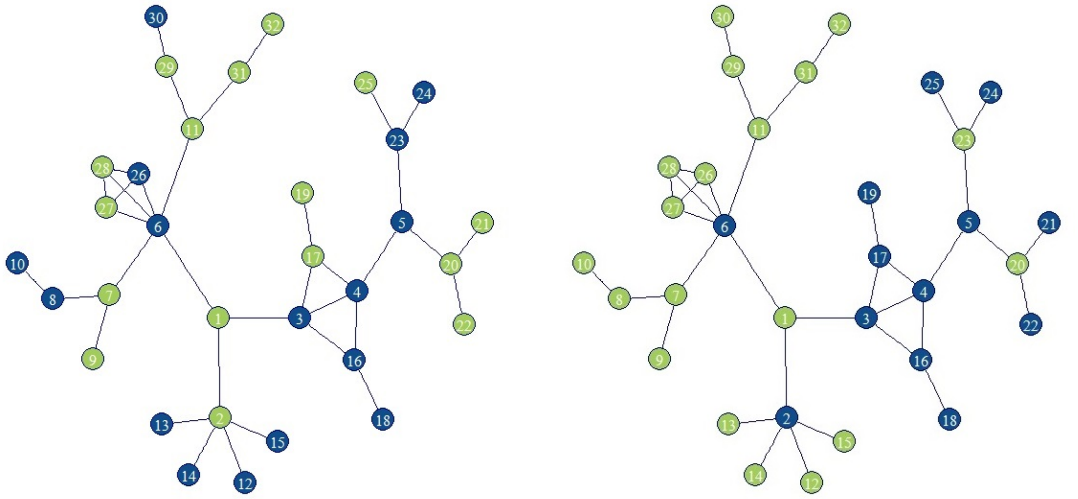
$$\mathrm{Eff}_{\phi_1} = \frac{0.3668}{0.6843} = 0.5360 \quad \text{and}$$

$$\mathrm{Eff}_{\phi_2} = \frac{0.1265}{0.3944} = 0.3207.$$

The resulting efficiencies are quite low, indicating that the optimal design on the whole network is not necessarily optimal for the skeleton of that network. However, it is important to note at this point that the optimal design under ϕ_1 is not unique. This brings us to the second key question, that is if we can find the optimal design on the 32-subject network of the original network conditional on fixing the optimal skeleton design, i.e. the optimal design on the particular subgraph consisting of 11 subjects. The answer is affirmative and led us to Conjecture 2.

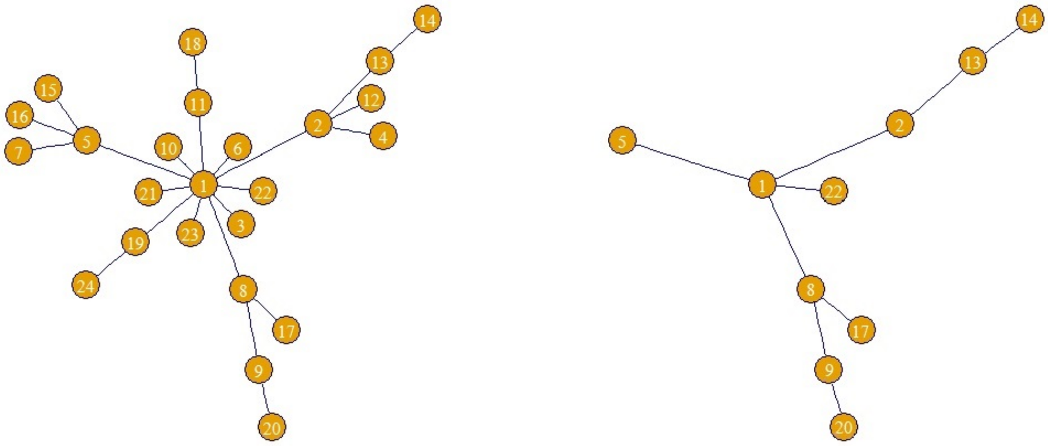
The computational (CPU) time in seconds (secs) required to solve this optimisation problem of finding an optimal design on the whole network having fixed the allocation on the skeleton is less than 3 seconds. The optimal function values are ${\phi_1}^* = 0.1250$ and ${\phi_2}^* = 0.0261$. The efficiencies compared to the corresponding designs under ϕ_1 and ϕ_2 are 100% and 91% respectively. The designs obtained following this approach are shown in Figure 4.10. This result is in agreement with Conjecture 2 which states that we can find the optimal design for ϕ_1 on a network by fixing the optimal allocation to its skeleton. The consequence of the results is that we can reduce considerably the size of the search region, given a restriction on the number of runs by considering the symmetries of the graph, and by finding the optimal design or at least a near-optimal design on its skeleton. This conjecture implies that for any network there is at least one optimal design which is also optimal for the skeleton. However, it is worth noting that even if this conjecture does not hold, it can still be very useful for finding near-optimal designs for large-sized networks within a small time frame. It should be noted that when m does not divide n or l some deviation occurs from the optimal function value.

Figure 4.10: Optimal design for ϕ_1 (*left*) and ϕ_2 (*right*) (given fixed allocation for the skeleton)



Let us now revisit Example 3.4.2. We investigate the degree of symmetry for each one of the different common types of snapshot networks of size 24 illustrated in Figure 3.7. Implementing *nauty* for these networks we have, as anticipated, that the random network is asymmetric, the small-world network has just two equivalent vertices (i.e. 9 and 10) and the scale-free network has 14 orbits. Bear in mind that locally tree-like structures are commonly present in real-world networks, which makes this example an interesting one to consider. Thus focusing on the scale-free network we obtain the optimal designs for the two optimality criteria. The CPU time required was calculated to be 2563.62 seconds for ϕ_1 and 2564.64 seconds for ϕ_2. The resulting vertex orbits for that network are $\{1\}$, $\{2\}$, $\{3; 6; 10; 21; 22; 23\}$,$\{4; 12\}$,$\{5\}$, $\{7; 15; 16; \}$, $\{8\}$, $\{9\}$, $\{11; 19\}$, $\{13\}$, $\{14\}$, $\{17\}$, $\{18; 24\}$, $\{20\}$. The run time required for obtaining the orbits was less than 0.01 seconds. Note that the $|Aut(\mathcal{G})| = 17280$. The original graph has 24 vertices and 23 edges, whilst the skeleton of the graph consist of 10 vertices and 9 edges (see Figure 4.11).

Figure 4.11: (a) Original graph-scale-free network of Figure 3.7(c) ($n = 24, l = 23$); (b) Graph skeleton ($n = 10, l = 9$)



Next we obtain the optimal designs on the skeleton. Having fixed these treatment allocations to the vertices of the skeleton we obtain the optimal designs for the original network for each criterion (time required less than 0.04 seconds). Figure 4.12 illustrates those optimal designs. More specifically, the optimal treatment allocations are $\{1, 0, 0, 0, 0, 1, 1, 0, 0, 0, 0, 1, 0, 0, 1, 1, 0, 1, 1, 0, 1, 1, 1, 1\}$ with $\phi_1 = 0.1672$ and $\{0, 1, 1, 0, 1, 1, 0, 0, 0, 1, 1, 0, 0, 1, 1, 1, 0, 1, 1, 0, 1, 1, 1, 1\}$ with $\phi_2 = 0.0505$. The design efficiencies are 100% and 80.2% respectively. The optimal designs found by exhaustive search for this network have ${\phi_1}^* = 0.1672$ and ${\phi_2}^* = 0.0405$, corresponding to the optimal allocations $\{0, 1, 0, 1, 1, 0, 0, 1, 1, 0, 0, 1, 1, 1, 0, 0, 0, 0, 1, 1, 0, 1, 1, 0\}$ and $\{0, 0, 1, 0, 1, 1, 0, 1, 1, 1, 1, 0, 0, 0, 0, 0, 1, 0, 1, 1, 1, 1, 1, 0\}$ respectively (see Figure 4.13).

It is important to highlight some key points of our research in symmetries. Firstly, when designing experiments on real-world networks, which are in generally richly symmetric then we strongly believe that partitioning the graph into its symmetric subgraphs (symmetric motifs) and then into its orbits proves to be an efficient way to proceed. This can save us considerable time and can help us potentially design experiments on moderate or large networks. Table 4.1 summarises the relevant computational time required to find the optimal designs on the two considered networks when utilising (or not) the symmetries of the network (supporting Conjecture 2 and illustrating its potential power).

Secondly, another point of interest is the 'fork' structure, which corresponds to identical branches of vertices of a tree-like structure (see, for example, vertices $29 - 32$ in Figure 4.6). The automorphism group of a tree is more complicated conceptually due to the presence of these 'fork' structures. In general, the automorphism group of a graph

Figure 4.12: Optimal designs for ϕ_1 (*left*) and ϕ_2 (*right*) (given fixed allocation for the skeleton)

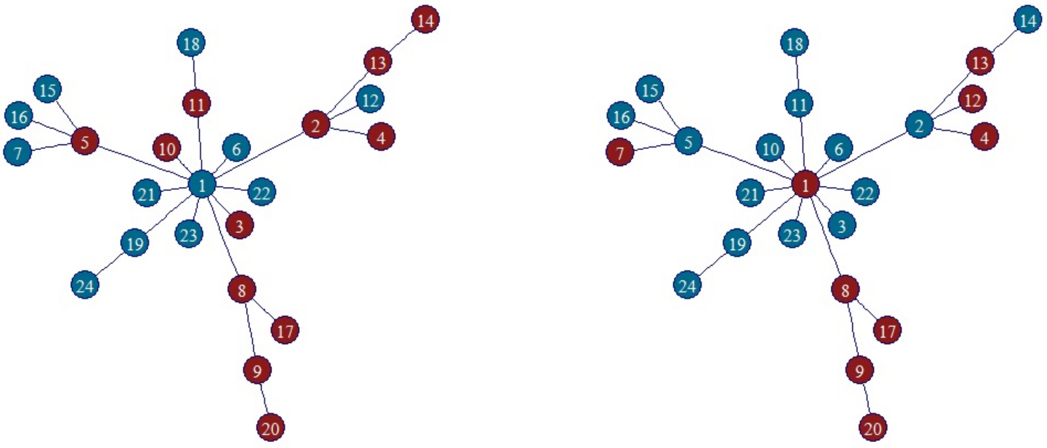


Figure 4.13: Optimal design for ϕ_1 (*left*) and ϕ_2 (*right*) (without restrictions); different colours indicate different treatments

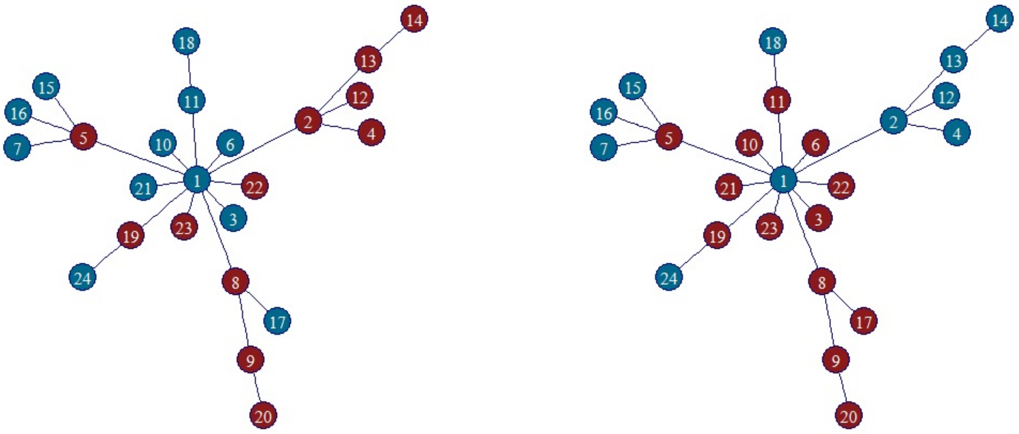
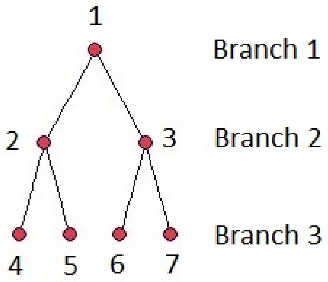


Table 4.1: Approximate time in seconds required to find the optimal designs

Network	Time	Time	Design efficiency	
	(ignoring symmetries)	(utilising symmetries)	ϕ_1	ϕ_2
Ex. 3.4.1	1728000	6	100%	91%
Ex. 3.4.2c	5128.26	0.05	100%	80%

can typically be decomposed into direct and wreath products of symmetric groups, the latter is a mild generalization of direct products (see, for instance, Kerber, 1971 or Rotman, 1994 for representation of wreath problems and examples). To understand what a wreath product is consider the binary tree illustrated in Figure 4.14. The tree automorphism group acts transitively on each branch of the tree. For instance, we can consider the sets of automorphisms switching the nodes 4 and 5 or 6 and 7 at the lowest branch (Branch 3 vertices). However, the vertices labelled 2 and 3 (Branch 2 vertices) can be switched in addition to the permutations on the leaves (Branch 3 vertices). This is represented by the wreath product of permutations of groups (which can be considered as 'dependent permutations' of the nesting branch structure). This has implications when treatments are allocated to units belonging to a tree like structure. When permuting the treatments assigned to units in a branch among the units in that branch this should be considered in addition to the treatment permutations associated with units at the lower branch (if that exists). In the research presented in this chapter we consider the vertices of the same branch to belong to the same vertex orbit, allowing freely for permutations among them without considering making restrictions based on the connected branches. Thus we anticipate the permutations on the corresponding vertex orbits comprising the symmetries of tree structures to impact on the ϕ_1-automorphism group and therefore on the optimality of the designs. Future work should better accommodate tree-structures when breaking down the design problem, by either assigning weights or altering the skeleton definition, which can lead to an improvement on the design efficiency.

Figure 4.14: Binary tree



It is clear that there is much to be done in this area. The close association between the network symmetry and the design of experiments stands out as very promising for future research and needs to be elaborated further. In this chapter we deal only with a limited part of this combined area of the two different fields, displaying the computational benefits for large complex networks. The following section recommends a basic algorithmic approach which can be readily used by researchers and practitioners for obtaining efficient designs on large networks with high degrees of symmetry, in one tenth of a second without substantial losses in the design efficiency.

4.3 Algorithmic approach

In this section we describe the algorithmic approach in the search for an optimal design in a network with a high degree of symmetry. We discuss some modification of this algorithm by an illustrative example and make comparisons by means of computational time as well as design performance for a given real-world social network. Recall from Section 3.4, the optimal designs on (symmetric) special graphs are balanced under ϕ_1, whilst under ϕ_2 optimal designs tend to be dominated by the same treatment for units of the same degree. We incorporate these patterns in our approach and we explore the results. Below we sketch the algorithmic approach (NSYM) for finding optimal designs while accounting for network symmetry. Before implementing NSYM, we calculate the preliminary measure r_s, that quantifies the degree of network symmetry by counting the frequency of the symmetric motifs in the entire graph. If the r_s measure is relatively low ($r_s \to 0$) then NSYM requires equivalent computational time as the systematic exchange (SE) algorithm suggested in Section 3.6, for finding an optimal design for the original network. However, large values of r_s indicate that utilising the network symmetries for finding the optimal design reduces significantly the search time leading to valuable gain in computational time. We assume arbitrarily that when $r_s \geq 0.4$ then taking into account the graph automorphism in our search for the optimal design can be very beneficial. This holds especially when dealing with a large number of treatments and a large number of experimental units that reflect the size of the network.

Algorithm that accounts for network symmetry (NSYM)

(I) (*Skeleton*). In order to obtain the skeleton, we employ the graph isomorphism program *nauty*, which makes use of the structural information of the graph derived from the degrees of the vertices and partitions the vertices into symmetric subgraphs.

- Step 0: Input non empty graph by entering the adjacency matrix (A) (indicating the connectivity structure). Units are labelled from 1 to n.
- Step 1: Obtain the graph skeleton (i.e. asymmetric subgraph omitting the vertex orbits).
- Step 2: Find an (near-) optimal design on the graph skeleton using the systematic exchange algorithm.

(II) (*Vertex orbits*). Different allocation of the treatments under ϕ_1 and ϕ_2.

- ϕ_1: Have a balanced allocation of the treatments within each vertex orbit, and ultimately achieve an overall balance along all orbits (if possible);
- ϕ_2: Have the same treatment (which will be randomly selected among the different treatments) allocated to all the units within the same vertex orbit.

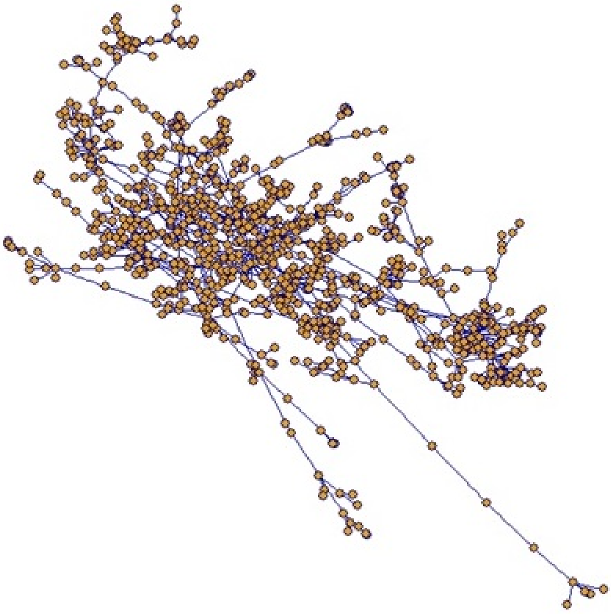
(III) (*Original network*). The design is constructed by fixing the allocation on the skeleton (Step *I*) as well as on the vertex orbits (Step *II*). The resulting design is expected to be good enough (optimal or near-optimal). (Additional systematic exchanges can be allowed, leading to a further improvement of the overall design.)

An alternative to the algorithm in Steps *I* or *II* is to consider a random exchange algorithm, by relaxing the constraint of systematically exchanging the treatments on the ordered units, but instead exchanging the treatments on randomly selected units (iterating for a specified number of times). Other alterations could be related to the occurrences of pairs of same or different treatments in a vertex orbit or in the skeleton.

We can also consider a quotient graph in Step I, as an extension of the skeleton for better capturing the network information. Recall that in a quotient graph, vertices represent groups and edges are induced by connectivity between groups. This can serve as a very interesting topic for experimentation. The following example illustrates the proposed approach.

Example 4.3.1. This is an example of a social network consisting of 1025 vertices and 1043 edges and is illustrated in Figure 4.15. The edges indicate the ties among PhD students and their advisors in theoretical computer science (Johnson, 1984). This network has been obtained from `http://vlado.fmf.uni-lj.si/pub/networks/Data/esna/CSPhD.htm`. Calculating some network measures we conclude that the average degree is approximately $\delta = 2.035$, the clustering coefficient $\mathcal{C} = 0.002$ and the average path length $\ell = 11.748$. The degree distribution of the network is illustrated on two scales in Figure 4.16. The figure shows that the distribution follows power-law vertex degree distribution (scale-free) as indicated by the fitted red line, which represents the fitted power law distribution with $\gamma = 1.924$ (Kolmogov-Smirnov test $R^2 = 0.897$). We want to find the optimal designs for estimating with minimum variance the treatment effects and network effects for the $m = 2$ treatment case.

Figure 4.15: PhD social network

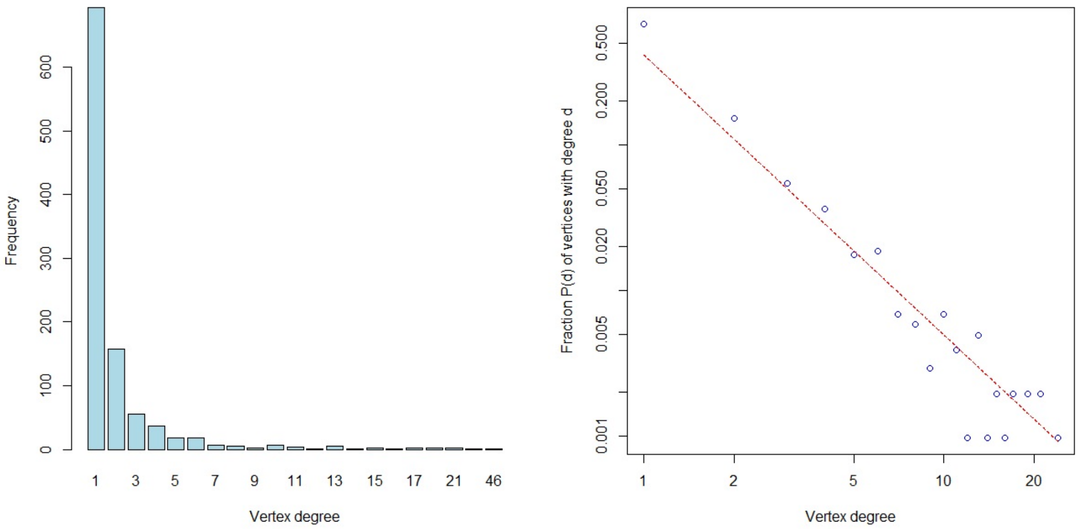


We obtain the orbits of the graph, of which there are 511 (CPU time = 0.45 seconds). Figure 4.17 illustrates the original graph and its skeleton. We can then quantify the degree of network symmetry as:

$$r_s = 1 - \frac{511 - 1}{1025} = 0.4976.$$

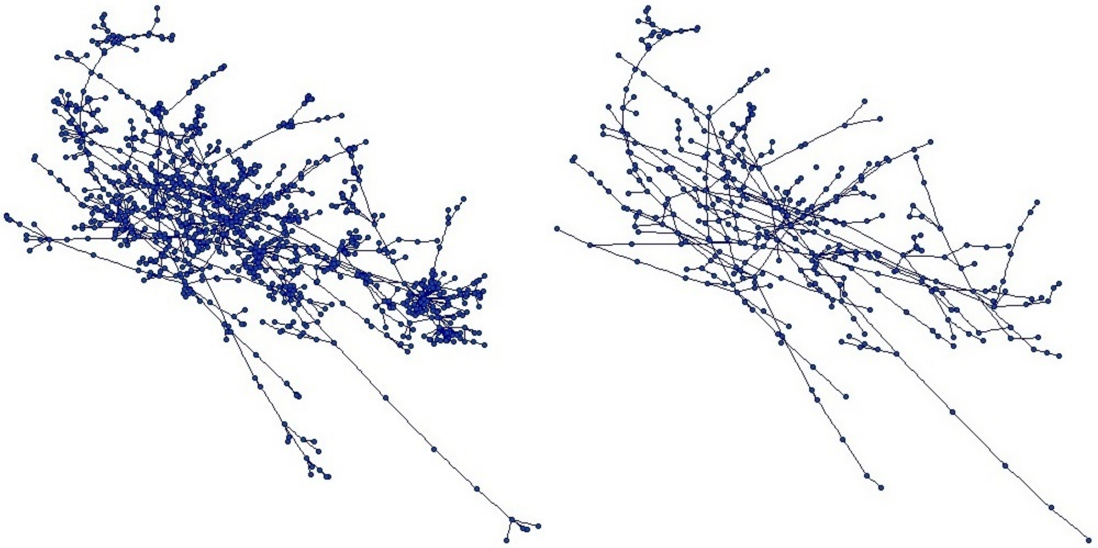
This value indicates that 50% of the vertices of the graph play the same structural role as at least one other vertex (structural repetition of the vertices). Recall that the

Figure 4.16: Degree distribution of the PhD network. Original (*left*); log-log (*right*)



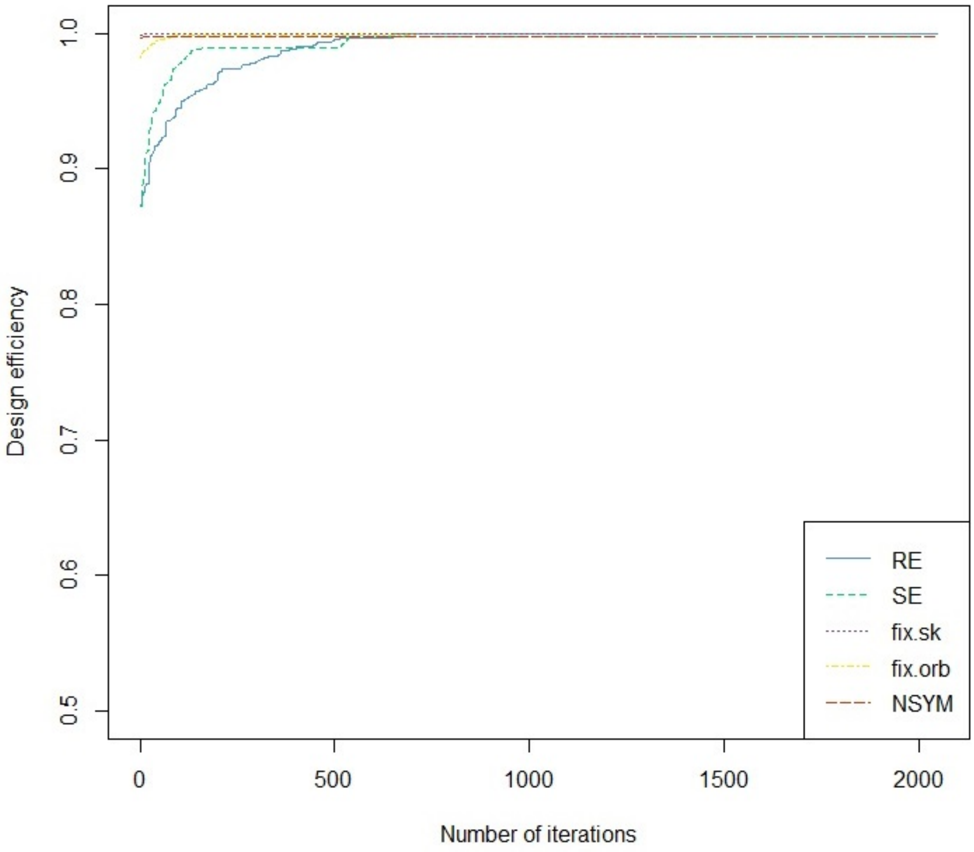
larger the value of r_s the more the network is constructed from repetition of structurally identical elements, and therefore the higher its degree of symmetry.

Figure 4.17: (a) Original graph ($n = 1025, l = 1043$); (b) Graph skeleton ($n = 358, l = 374$)



Consequently, we investigate different algorithmic update scenarios, which correspond to the components that compose the suggested algorithm. On the grounds of comparison among those strategies we set as the starting design the same random balanced design (one random start). The algorithms make exchanges to the treatment arrangements on the skeleton and/or vertex orbits, if those improve the overall design. Figures 4.18 and 4.19 illustrate all the considered algorithms in an attempt to compare them. The vertical axis represents the design efficiency under each criterion and the horizontal axis the number of iterations corresponding to the consecutive design evaluations required for the algorithms to converge. The efficiencies were calculated with reference to the best design achieved among all those algorithms (this design may or may not coincide with the optimal design). The different lines correspond to the five different algorithmic scenarios: random exchange ('RE'), systematic exchange ('SE'), fixing the allocation only on the skeleton ('fix.sk') or only on the vertex orbits ('fix.orb'), while allowing for systematic exchanges to the remaining vertices and fixing the allocation on both skeleton and vertex orbits ('NSYM'). The RE algorithm makes a specified number of exchanges at random as opposed to the SE algorithm (first ascent). In order for the two to be comparable, the number of iterations specified for the RE algorithm are on average as many as the SE needed to converge. We would expect that by increasing the number of iterations for RE, given the available computational time, the probability of finding a design that is close to the (globally) optimal would increase. In general those two alternative exchange algorithms perform equally well.

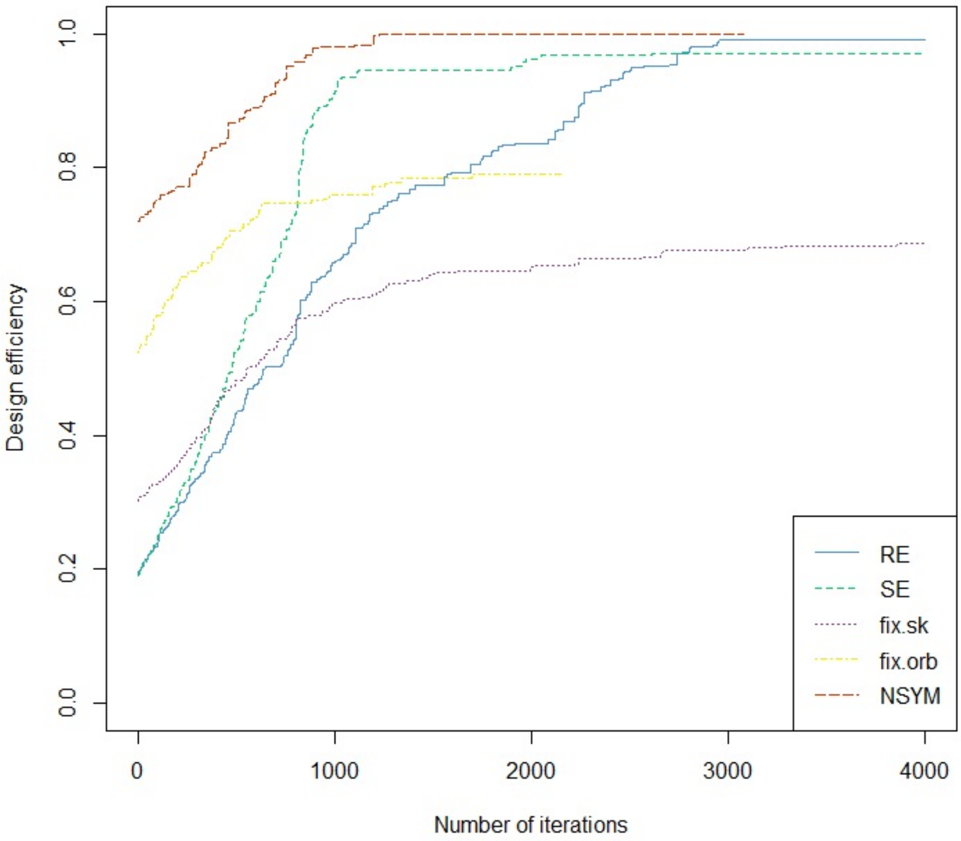
Figure 4.18: Relative design efficiency versus the number of iterations under ϕ_1



For NSYM in Figures 4.18 and 4.19, we allowed for a few complete passes through all the vertices for potential exchanges. Another observation is that ϕ_2 requires a larger number of iterations as opposed to ϕ_1 under all algorithms. This larger number of iterations for ϕ_2 is needed to account for the complexity and the absence of balance in the design (see Section 3.4).

The optimal designs were found to have $\phi_1^* = 3.902\ \times 10^{-3}$ and $\phi_2^* = 0.350\ \times 10^{-3}$. For

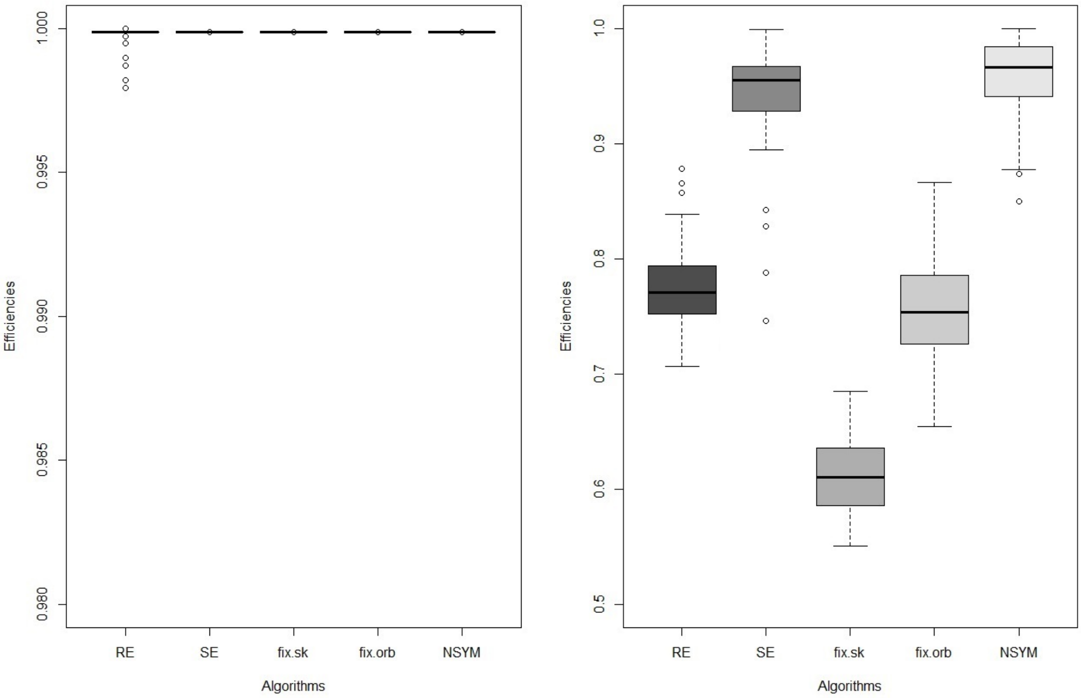
Figure 4.19: Relative design efficiency versus the number of iterations under ϕ_2



the case of ϕ_1, we can observe that all algorithmic scenarios converge to the optimal design. The starting balanced design is approximately 87% efficient ($\phi_1 = 4.474 \times 10^{-3}$) and a few iterations seem to be sufficient for achieving a good design for this example (less than n design evaluations). When we fix the allocations on the skeleton and the vertex orbits (NSYM), the resulting designs are 100% ($\phi_1 = 3.917 \times 10^{-3}$) efficient for ϕ_1 and 70% ($\phi_2 = 0.497 \times 10^{-3}$) efficient for ϕ_2 (without exchanges). If we decide to allow for some additional design evaluations (desirable for ϕ_2) then it converges to the best design. NSYM seems to require significantly fewer successful iterations compared to all other algorithms for reaching the best design under both criteria. When we fix the allocation in either the skeleton or vertex orbits, allowing for iterations on the remaining experimental units, the algorithms under ϕ_2 produce relatively efficient designs (skeleton: $\phi_2 = 0.523 \times 10^{-3}$ and orbits: $\phi_2 = 0.454 \times 10^{-3}$). It is worth mentioning that the best found ϕ_1-optimal design contains the optimal design found in the skeleton (Conjecture 2). Reproducing these figures having different initial designs, we obtain qualitatively similar results, with NSYM outperforming on average the other approaches in finding the optimal design while running fewer iterations. Allowing for multiple random initialisations, all different strategies produce better designs in terms of efficiency.

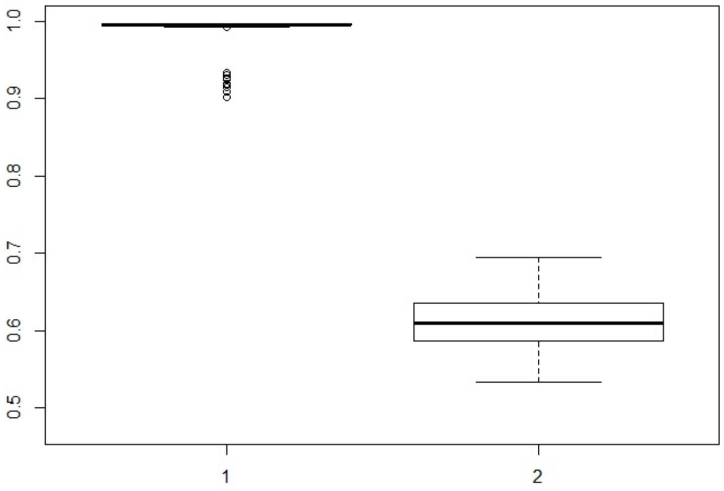
To illustrate this, Figure 4.20 shows boxplots of design efficiencies with respect to the best found design having 100 random starts (both balanced and unbalanced), corresponding to the n-th iteration for ϕ_1 and to the $2n$-th iteration for ϕ_2 (where $n = 1025$ is the total number of units). The number of iterations is not the same for the two criteria and is dependent on the required quality of the design (see Figure 4.20). This allows us to get an idea of how well the different algorithms perform when we have multiple starting designs, having fixed the number of runs (total number of evaluations). Whilst for ϕ_1 all approaches obtain the best designs, for ϕ_2 only SE and NSYM achieve the optimal design, with the latter obtaining it more frequently. In addition, Figure 4.21 presents the boxplots of design efficiencies for ϕ_1 and ϕ_2 corresponding to NSYM without any iterations. The median efficiencies are 100% and around 60% respectively, indicating that for ϕ_2 it is sensible to allow for extra exchanges overall.

Figure 4.20: Boxplots of design efficiencies (100 random starts)



A key observation here is that NSYM performs consistently better than the other algorithms for this particular example. When fixing the allocation on either the skeleton or orbits we have a significant improvement in the computational time required, due to the simpler algebraic computations (requiring smaller sized matrix inversion). SE requires more time than the other approaches, but is likely to find good designs. However, there is a limitation in this comparison of the algorithms due to the variability of the results stemming from the choice of sub-design on the skeleton or orbits. Consider, for instance, that the 'fork' structure previously discussed has not being accounted for appropriately when allocating the treatments in the sense that immediately connected vertices that belong in different orbits (e.g. branches of a tree-like structure) may receive the same treatment. Moreover, it could be fruitful particularly to consider the same type of symmetric motifs and treatment allocation on them in turn, e.g. all star struc-

Figure 4.21: Boxplots of efficiencies for NSYM without iterations; ϕ_1 (*left*) and ϕ_2 (*right*)



tures together etc. Overall, the NSYM approach greatly increases the computational efficiency of the design search without sacrificing any appreciable design efficiency.

4.4 Discussion

This chapter investigates how the network symmetry affects the properties of the design and particular its optimality and how we can utilise it to our benefit in the design search in an effective way. Decomposition of the graph based on its symmetries can be regarded as an essential step in the search for an optimal design on experimental units that are connected in a network formation. We provided evidence of the beneficial effects of symmetry breaking on the experimental design, decreasing the computational complexity and size of the design space for speeding up the search for an efficient design. We presented an example of a scale-free network, which is a common type of real-world network. We believe that in order to understand the allocation of the design as a whole, on the original network, we should first decompose the network and try to understand the allocation of the design to its network constituents. To the best of our knowledge, this study is the first to provide a strategy for finding experimental designs on networks while accounting for its symmetries. The main advantage of our approach is that it relies on building up sequentially the design on the subgraphs, reducing repeated evaluations of equivalent designs on the original graph and alleviating limitations due to memory shortage of a computer when dealing with large networks. Thus it can reduce computational burden of the optimality criterion calculation substantially (due to small network and design sizes), enabling us to obtain good designs within a practical time frame (due to the small number of evaluations required). This flexible approach could be implemented irrespective of the main algorithm for finding particular optimal design.

It is worth noting that the graph isomorphism problem, that is the determination if two graphs (of different targets and domains) are isomorphic or not, belongs to the class of NP computational complexity (see Köbler *et al.*, 1993 for an extended discussion). The graph isomorphism problem is computationally equivalent to the problem of computing the automorphism group of a graph. However, Babai (2015) claims that the graph isomorphism problem and thereby also the graph automorphism problem can be solved in quasipolynomial time (i.e. quantity is exponential in some power of a logarithm). This finding suggests the existence of efficient deterministic algorithms for this problem.

Our approach points to the right direction for future research, with possible developments involving the identification of ϕ_2-automorphisms. Experimental units are interwoven with their defining network characteristics and therefore imposing additional restrictions on the randomisation process related to the units on the quotient graph may be more appropriate. An algorithmic improvement could relate to determining occurrences for particular types of network motif. Accounting for the different motif frequencies can further enable the design search to be carried out effectively in practice for large sized networks. Investigating alternative connectivity matrices such as a Laplacian or the squared adjacency matrix could also be a way to move forward.

Chapter 5

Optimal block designs with network effects

In this chapter we develop a methodology for finding optimal block designs with network effects. We incorporate a blocking structure, based on the underlying connectivity structure, as an attempt to reduce unexplained variation. In Section 5.1 we introduce the basic elements of spectral graph theory with graph Laplacians (most of which is given in Section 2.4). In Section 5.2, we provide the SM (Shi and Malik, 2000) spectral clustering algorithm used to define the blocks of the experimental design. Subsequently, in Section 5.3, we extend the LNM (Parker *et al.*, 2016) incorporating the notion of blocks. We then provide a working example following the recommended step-by-step methodology for the special case of two unstructured treatments. Issues associated with the design efficiency and bias underlying the analysis of dependent data are explored in Sections 5.4–5.6, where we also compose some optimal designs under different models. We also touch upon some practical concerns emerging from the suggested methodology such as the choice of graph Laplacian and further issues of interest.

5.1 Spectral clustering and block definition

Spectral clustering techniques play a major role in determining data clusters (also termed communities or partitions), when there are weak connections between clusters and strong connections within them. To detect the community structure in a network, a clustering algorithm will be employed, which basically partitions the corresponding graph into different groups of subgraphs, with dense connections within groups and only sparser connections between them. From the design point of view, we expect the densely clustered units of strong connections to exhibit similar response patterns within the same communities and dissimilar ones from units of different communities irrespective of the presence or absence of viral effects of specific treatments. Consider for instance the advertisement example described in Section 1.1, clusters may contain cliques of close friends who engage in similar behaviours, e.g. following a similar decision pattern as of what quantity of product to purchase. In this thesis, we try to discover existing clusters which we use to define blocking structures. It could be of interest to use another covariate on the basis of non-structural information, e.g. males versus females in the case of social networks, or different age groups. However, this is outside the scope of this thesis.

Suppose that n units are available for experimentation and they form a network which is represented by means of a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, with vertex set $\mathcal{V}$ (of size n) and edge set $\mathcal{E}$ (of size l) (see Section 2.4). These units are divided into κ clusters (communities) which have been obtained using spectral clustering techniques (more techinical details will be given in Section 5.2). We assume that units within the same cluster are expected to behave similarly to an external stimulus because of shared unobserved characteristics and attributes. For this reason, we define the blocks to be the dense clusters found by exploiting the units' network connections. There is a group effect separately from the individual network effect, and irrespective of the applied treatments. In other words, the purchasing decisions might be similar for units belonging to the same blocks independently of the particular treatment allocation or network connections (i.e. block effects). In that way we establish more precise comparisons of treatment effects within blocks than when making comparisons between blocks. Motivated by this natural definition of blocking we suggest the network effects block model (NBM) with fixed effects for blocks for design purposes in Section 5.3.

In Chapter 2 we described the main tools used for spectral clustering, i.e. the graph Laplacian matrices, and provided their detailed mathematical formulae. The main differences between spectral clustering techniques lie in the choice of the Laplacian matrix. There are several arguments in favour of using normalised rather than unnormalised spectral clustering (e.g. see Von Luxburg and Bousquet, 2004 and Cheng and Shen, 2010). The objective function employed to optimise graph clustering in the normalised case is based on the degree of a cluster instead of its size as in the unnormalised case, which is preferable when dealing with irregular graphs of different degree distributions. Another argument against unnormalised graph clustering is that it is inclined to produce partitions containing only one vertex. We refer to Von Luxburg (2007) for a more detailed justification of why normalised spectral clustering performs better than the unormalised one, but also specifically for the normalised case why L_{rw} is more fitting than L_{sym}. The defence for using the eigenvectors of L_{rw} is that they are cluster indicator vectors $\mathbf{1}_{C_i}$ as opposed to those of L_{sym} which are additionally multiplied by $D^{1/2}$ (see Section 2.4). We chose to obtain our results using the normalised Laplacian graph L_{rw}, which is related to a random walk rendering it a useful tool for capturing a diffusion process, as is the treatment propagation effects, in a network (see Section 2.4). Recall that the normalised random walk Laplacian is given as $L_{rw} = I - D^{-1}A$. For an in-depth coverage of these issues related to graph Laplacians and principles of spectral clustering refer to Von Luxburg (2007) as well as Chung (1997), who both provide extensive introductory overviews, with many references to research papers providing further technical details.

5.2 Clustering algorithm

We implement a spectral clustering algorithm for detecting communities in a (social) network. We justify our choice due to the simplicity of its implementation (standard linear algebra), without requiring any explicit model of the data distribution and without making any kind of assumption on the formation of the clusters. There is a growing literature of clustering algorithms that are based on optimisation methods such as greedy algorithms, simulated annealing, or spectral optimisation, with different approaches offering different tradeoffs between speed and accuracy (for a comparison see Danon *et al.*, 2005). An interesting critique on a number of different applications of clustering, their 'usefulness' and corresponding evaluation procedures was made by Guyon *et al.* (2009). Our suggested methodological approach performs satisfactorily.

Consequently, we provide the normalised spectral clustering algorithm, namely the Shi and Malik (SM) algorithm (Shi and Malik, 2000) as based on Von Luxburg (2007) with appropriate adjustments for the purposes of this thesis:

1. **Compute the normalised graph Laplacian** L_{rw} and its spectrum (as based on the known adjacency matrix of the network).
2. **Dimensionality reduction**: using the κ first eigenvectors $\boldsymbol{v}_1, \ldots, \boldsymbol{v}_\kappa$ of the graph Laplacian that correspond to the first κ eigenvalues sorted in ascending order, let $U \in R_{n \times \kappa}$ be the matrix containing $\boldsymbol{v}_1, \ldots, \boldsymbol{v}_\kappa$ as columns. (Note that the eigenvector-space varies based on the chosen dimensionality κ.)
3. **Clustering step**: treating each row of U as a data point, $(y_i)_{i=1,\ldots,n} \in R^\kappa$, group them via the (standard) k-means algorithm into κ (dimensionality of the eigenvector space) clusters, $C_1, \ldots, C_\kappa$. Therefore, the vertices of the network are projected into a κ-dimensional space, where κ is the number of the first nontrivial eigenvectors of the L_{rw}. As a result, each unit is allocated to one of the produced clusters.

We perform the *clustering step* for various numbers of clusters κ, $\kappa = 2, \ldots, n/2$, the choice of which is specified in order to limit the numerical search given that there should be at least two units within a cluster for achieving treatment comparisons. Note that the partition method can rely on a different clustering step rather than k-means if required. However, we used this standard one, which is the most used partitioning method. Spectral clustering is derived as an approximation to a graph partitioning problem. Without going into detailed technicalities, it is based on the objective function *Ncut* (Shi and Malik, 2000), defined as

$$Ncut(C_1, \ldots, C_\kappa) = \frac{1}{2} \sum_{i=1}^{\kappa} \frac{A(C_i, \bar{C}_i)}{vol(C_i)},$$

where A is the adjacency matrix of the graph, with $A(C_i, \bar{C}_i) = \sum_{j \in C_i, h \in \bar{C}_i} A_{jh}$, denoting the edges in the cut (i.e. edges between highly connected subgraphs). Note at this point that given a subset of vertices $C \subset \mathcal{V}$, we denote its *complement* $\mathcal{V} - C$ by $\bar{C}$ and therefore $\bar{C}_i$ is the complement of C_i. $vol(C)$ measures the size of C, calculated by summing up the edges attached to the vertices belonging to C, which is equivalent to the summation of the degrees of the vertices in C, i.e. $vol(C) = \sum_{j \in C} d_j$. The objective function takes a small value if the clusters are too small; therefore the obtained clusters are 'reasonably large' (Von Luxburg, 2007). The factor 1/2 is introduced for consistency reasons, otherwise we would count each edge twice since we are dealing with undirected graphs.

However, this clustering algorithm gives no hint about the choice of the number of clusters or 'goodness' of each partition. Various methods have been suggested to address these practical concerns (see, e.g., Von Luxburg, 2007). For instance, the eigengap heuristic is a common way to choose the number of clusters such that all eigenvalues $\lambda_1, \ldots, \lambda_\kappa$ are very small but $\lambda_{\kappa+1}$ is relatively large. As a result the $|\lambda_{\kappa+1} - \lambda_\kappa|$ gap indicates that the dataset contains κ clusters. While this heuristic works well if the clusters are well pronounced, its effectiveness decreases the more noisy or overlapping the clusters become. In this thesis, in order to quantify the quality of partitions and to choose the number of communities based on the 'best' partition, we use the concept of modularity (Newman and Girvan, 2004).

Modularity is a measure which quantifies the strength (degree) of the obtained community structure in a network by comparing it with a possible arrangement of the edges in an equivalent network (of the same size and same degree sequence), where the edges are placed at random (Newman, 2006). Hence by this comparison, and bearing in mind that the random network is not expected to have a cluster structure, the possible presence of clusters is revealed. The mathematical formation of this measure is

$$Q = \frac{1}{2l} \sum_{jh} \sum_{i} \left(A_{jh} - \frac{d_j d_h}{2l} \right) s_{ji} s_{hi},$$

where l is the total number of edges in the network, A_{jh} has value 1 if vertices j and h are connected and 0 otherwise, and s_{ji} and s_{hi} are binary indicators of whether vertices j and h belong to group i or not (membership vectors). d_j is the degree of vertex j and $2l = \sum_{jh} A_{jh} = \sum_j d_j$ is the total degree of all the vertices. In other words, modularity is defined as the difference between the fraction of the edges that fall within clusters and the fraction of the edges that would be expected to fall within the communities if the edges were assigned randomly but keeping the degrees of the vertices unchanged. Looking at the terms in more detail:

- The observed fraction of edges in each community; that is the total number of edges that run between the vertices in the same cluster i,

$$\frac{1}{2} \sum_{jh} A_{jh} s_{ji} s_{hi}.$$

- The expected fraction of edges in each community; that is the expected number of edges between all pairs of vertices j and h in the same cluster if edges are placed at random,

$$\frac{1}{2}\sum_{jh}\frac{d_j d_h}{2l}s_{ji}s_{hi}.$$

To better understand this expression consider that the chances of a particular edge in one of the d_j being attached to vertex j is $d_j/2l$ where $2l$ denotes the number of the edges in the entire network. Counting all d_j edges attached to j, the probability that the vertices j and h, with degrees d_j and d_h respectively, are connected is given by $(d_j d_h)/(2l)$, since edges are placed independently of each other. Thus the expected degree distribution coincides with that of the graph, which is equivalent to the configuration model that is the model of a random graph (Molloy and Reed, 1995).

Note that the factor of 1/2 in both expressions accounts for the fact that every vertex pair j, h is counted twice. Taking the difference of the above mentioned expressions, summing over all communities and taking the fraction of such edges (i.e. dividing by l) we obtain the expression for Q. The value of modularity lies in the range $[-1/2, 1]$ and high (positive) values indicate that more actual edges are within a cluster than what one would expect to have by chance, indicating possible presence of community structure (Newman, 2006). When the communities are not better than the random partition or when the network does not exhibit any community structure, Q is negative or zero. The upper limit can be reached if the communities have been perfectly detected. The maximisation of modularity has been proposed as a possible method for detecting communities (Newman and Girvan, 2004); however, in this thesis, modularity measurements are simply used to find the best partition among all the possible partitions of the network produced using the suggested spectral clustering algorithm. We identify the best densely connected cluster with $\kappa = \arg\max Q$. As such κ is regarded as the appropriate candidate for indicating the number of intrinsic communities which will define the blocks to be used in the design process, if the κ-th is the highest modularity score over all produced partitions.

Prior to implementing the clustering algorithm on a real-life network in the following section, we revisit the simple Example 2.4.1, which serves to better understand the general concepts provided in this section. Its blocking structure obtained by implementing the suggested spectral clustering algorithm with optimal modularity is illustrated in Figure 5.1. To detect this clustering, we firstly calculate the spectrum of L_{rw} (see p.31). The eigenvalues sorted in an ascending order are: $0, 0.68, 1, 1, 1.54, 1.78$. Employing modularity for evaluating the quality of clusters of 2 and 3 units, we choose $\kappa = 2$ where the modularity reaches its maximum. In particular, we have $Q = 0.07$ for $\kappa = 2$ and $Q = 0.03$ for $\kappa = 3$. For example, the calculation of the modularity score for the given configuration of two clusters is

$$Q = \frac{1}{14}\Big[A_{11} - \frac{25}{14} + A_{22} - \frac{4}{14} + A_{33} - \frac{9}{14} + A_{44} - \frac{4}{14} + A_{55} - \frac{1}{14} + A_{66} - \frac{1}{14}$$

$$+ 2\left(A_{23} - \frac{6}{14} + A_{34} - \frac{6}{14} + A_{24} - \frac{4}{14} + A_{15} - \frac{5}{14} + A_{16} - \frac{5}{14} + A_{56} - \frac{1}{14}\right)\Bigg]$$

$$= \frac{1}{14}\Big[-1.79 - 0.29 - 0.64 - 0.28 - 0.07 - 0.07 + 2(0.57 + 0.57 - 0.28 + 0.64$$

$$+ 0.64 - 0.07)\Big] = 0.07\,.$$

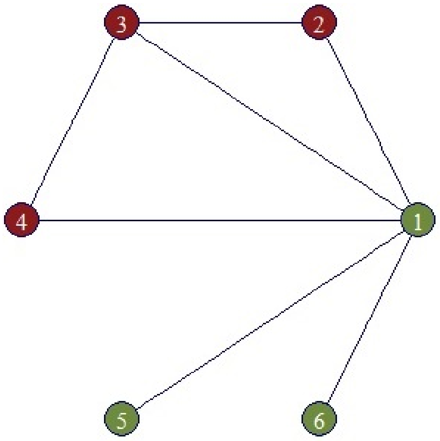
The matrix $U_{n\times\kappa}$ is therefore

$$U = \begin{pmatrix} 0.408 & 0.188 \\ 0.408 & -0.264 \\ 0.408 & -0.356 \\ 0.408 & -0.264 \\ 0.408 & 0.591 \\ 0.408 & 0.591 \end{pmatrix}.$$

The cluster means, from the k-means clustering step, are

$$\begin{pmatrix} 0.408 & -0.294 \\ 0.408 & 0.457 \end{pmatrix}.$$

Figure 5.1: Blocking structure of Example 2.4.1



However, we should note that the values of Q are relatively low indicating that fewer actual edges are within a cluster than what one would expect to have by chance. In other words, there is no strong evidence of community structure and therefore it is not sensible to have clusters for this small network.

5.3 Designs with network effects block model (NBM)

In this section we review the main ingredients of optimal block designs on networks with network effects, which are the response model and the optimality criteria. We introduce the network effects block model (NBM), which will form the primary tool for finding optimal block designs for various observed social networks. The NBM is an extension of LNM (see Equation (3.9)) and is described by the equation

$$\text{NBM: } y_{ij} = \mu + \tau_{r(ij)} + b_i + \sum_{g=1}^{\kappa} \sum_{h=1}^{n_{(g)}} A_{\{ij,gh\}} \gamma_{r(gh)} + \epsilon_{ij} \tag{5.1}$$

where $i = 1, 2, \ldots, \kappa; j = 1, 2, \ldots, n_{(i)}$, y_{ij} is the response from unit j in i-th block receiving the treatment $s = r(ij) \in \{1, \ldots, m\}$, μ is the average response for the whole set of units (overall mean), b_i is the effect of block i, $\tau_{r(ij)}$ is the (direct) treatment effect, $A_{\{ij,gh\}}$ is the adjacency matrix indicating the edge between units j and h belonging to blocks i and g respectively, $\gamma_{r(gh)}$ is the network effect (neighbour or indirect treatment effect) and ϵ_{ij} are the errors, which we assume to be independent and identically distributed with mean 0 and constant variance σ^2. This model is a simplified representation of the propagation of treatment effects in clustered units within a network. Note that the neighbour effects can affect units in other blocks (as opposed to what is considered in the work of Pearce, 1957).

The expectation of this model in matrix formation is

$$\mathbb{E}\left[\boldsymbol{y}\right] = \left(\mathbf{1}\ \boldsymbol{u_1} \ldots \boldsymbol{u_{m-1}}\ \boldsymbol{w_1} \ldots \boldsymbol{w_{\kappa-1}}\ A\boldsymbol{u_1} \ldots A\boldsymbol{u_m}\right) \left(\mu\ \tau_1 \ldots \tau_{m-1}\ b_1 \ldots b_{\kappa-1}\ \gamma_1 \ldots \gamma_m\right)^T,$$

where $\boldsymbol{\beta} = \left(\mu \quad \boldsymbol{\tau}^T \quad \boldsymbol{b}^T \quad \boldsymbol{\gamma}^T\right)^T = \left(\mu \quad \tau_1 \ldots \tau_{m-1} \quad b_1\ \ldots\ b_{\kappa-1} \quad \gamma_1 \ldots \gamma_m\right)^T$ is the vector parameter. There are no columns corresponding to the m-th treatment effect, τ_m, and κ-th block effect, b_κ, since we assume them to be zero (see Section 2.2). The symmetric information matrix M is

$$M = \begin{pmatrix} n & \mathbf{1}^T X_\tau{}^\star & \mathbf{1}^T X_b{}^\star & \mathbf{1}^T A X_\tau \\ X_\tau{}^{\star T}\mathbf{1} & X_\tau{}^{\star T} X_\tau{}^\star & X_\tau{}^{\star T} X_b{}^\star & X_\tau{}^{\star T} A X_\tau \\ X_b{}^{\star T}\mathbf{1} & X_b{}^{\star T} X_\tau{}^\star & X_b{}^{\star T} X_b{}^\star & X_b{}^{\star T} A X_\tau \\ X_\tau{}^T A\mathbf{1} & X_\tau{}^T A X_\tau{}^\star & X_\tau{}^T A X_b{}^\star & X_\tau{}^T A^2 X_\tau \end{pmatrix}$$

$$= \begin{pmatrix} n & & & & & & & & & \\ n_1 & n_1 & & & & & & & & \\ \vdots & \vdots & \ddots & & & & & & & \\ n_{m-1} & 0 & \ldots & n_{m-1} & & & & & & \\ n_{(1)} & n_{(1)1} & \ldots & n_{(1),m-1} & n_{(1)} & & & & & \\ \vdots & \vdots & \vdots & \vdots & \vdots & \ddots & & & & \\ n_{(\kappa-1)} & n_{(\kappa-1)1} & \ldots & n_{(\kappa-1),m-1} & 0 & \ldots & n_{(\kappa-1)} & & & \\ l_1 & l_{11} & \ldots & l_{1,m-1} & l_{(1)1} & \ldots & l_{(\kappa-1)1} & l_{11}{}^{\langle 2\rangle} & & \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \ddots & \\ l_m & l_{m1} & \ldots & l_{m,m-1} & l_{(1)m} & \ldots & l_{(\kappa-1)m} & l_{m1}{}^{\langle 2\rangle} & \ldots & l_{mm}{}^{\langle 2\rangle} \end{pmatrix},$$

where $n_{(i)s}$ is the number of units given treatment s $(= 1, 2, \ldots, m)$ belonging to block

$i\ (= 1, 2, \ldots, \kappa)$, $l_{(i)s}$ is the number of links of units given treatment s belonging to block i, $l_{ss'}$ and $l_{ss'}{}^{\langle 2 \rangle}$ correspond to the number of links between units given treatment s and those given treatment s' of walks of length one and two respectively (including closed walks, see Section 2.4). It holds that $\sum_{i=1}^{\kappa} \sum_{s=1}^{m} n_{(i)s} = \sum_{i=1}^{\kappa} n_{(i)} = \sum_{s=1}^{m} n_s = n\,.$

As in Section 3.2, the L-optimality criteria (ϕ_1 and ϕ_2 respectively) are

$$\phi_1 = \sum_{v=2}^{m} \sum_{h=v+1}^{m+1} \boldsymbol{s}^T(v,h) M^{-1} \boldsymbol{s}(v,h)$$

and

$$\phi_2 = \sum_{v=w+m+2}^{w+2m} \sum_{h=v+1}^{w+2m+1} \boldsymbol{s}^T(v,h) M^{-1} \boldsymbol{s}(v,h),$$

where $\boldsymbol{s}(\alpha_1, \alpha_2)$ is a vector of zeroes of length $2m + w$, with $w = \kappa - 1$. This vector is formed by considering a vector of zeroes with length $2m + w + 1$ (corresponding to the column of constants, columns of the m treatment effects, columns of the $\kappa - 1$ block effects, columns of the m network effects) except the α_1 and α_2 elements which are 1 and -1 respectively (for more details refer to Section 3.2). The $(m+1)$-st element from each vector is removed before it is pre- and post-multiplied by the matrix M^{-1}. This accounts for the constraint $\tau_m = 0$, for uniquely estimating the treatment effects. For instance, for the case of two treatments and two blocks the criteria become

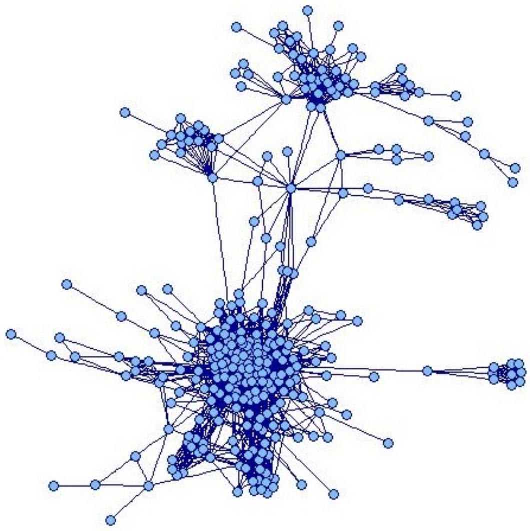
$$\phi_1 = \boldsymbol{s}^T(2,3)\; M^{-1}\; \boldsymbol{s}(2,3)$$
$$\phi_2 = \boldsymbol{s}^T(5,6)\; M^{-1}\; \boldsymbol{s}(5,6),$$

where $\boldsymbol{s}\,(2,3) = (0 \quad 1 \quad 0 \quad 0 \quad 0)^T$ and $\boldsymbol{s}\,(5,6) = (0 \quad 0 \quad 0 \quad 1 \quad -1)^T$.

For finding the near-optimal block designs based on the NBM and network at our disposal we implement the algorithm (presented in Section 3.6) with the appropriate adjustments. As a result we are in search for L-optimal block designs for explicitly estimating the direct and indirect treatment effects.

Example 5.3.1. The dataset was obtained from the large Stanford network dataset collection (`snap.stanford.edu/data/egonets-Facebook.html`) and consists of 4039 vertices and 88234 edges. The vertices denote the Facebook-members and the edges the virtual friendships between them. Based on recent Facebook analytics (`http://newsroom.fb.com`), there are approximately two billion active monthly users on average who seek interpersonal connectivity but also many entrepreneurs and marketing professionals who opt to promote their products or business (as of March 2017). The dataset was collected by McAuley and Leskovec (2012) as part of their research on testing the efficiency of a novel model for identifying users' social circles on several networking sites, including Facebook. The dataset is freely distributed for both academic and commercial use. In this thesis, however, a small subset was used containing 324 vertices and 2514 undirected edges (mutual ties) forming an ego-network, which is a subgraph of the original network (see Figure 5.2). In particular, the 'centre' vertex of an ego-network (the 'ego') is not included in graph $\mathcal{G}$, but $\mathcal{G}$ rather consists of only ego's friends (its neighbouring contacts). Note that we are interested in the connected part and therefore twenty isolated members have been omitted.

Figure 5.2: A Facebook ego-network



Step one: Blocking Structure. Expressing the topology of the above network of 324 vertices through the normalised Laplacian matrix L_{rw}, we produce a number of possible partitions of the network for different values of κ (dimensionality of the eigenvector space) using the spectral clustering algorithm SM presented in Section 5.2. Then we assess all these produced partitions using the quality function of modularity and choose the number of communities (fixed κ) to be used for the block design, which maximises the value of modularity. For the social network at hand the maximum modularity value over all possible partitions is found for $\kappa = 24$ clusters. The vertical and horizontal axes represent the modularity score and number of clusters respectively. Figure 5.3 illustrates the modularity scores for different number of communities obtained via graph partition. This screening stage helps us to choose the number of communities, where the modularity score takes its maximum value. The resulting clustering is illustrated in Figure 5.4.

Step two: Optimal design with block and network effects. We wish to compare two unstructured treatments, which could correspond, for instance, to advertisements as discussed in the Example of Section 1.1. The allocation of the treatments to units is achieved by implementing our simple exchange algorithm for finding L-optimal designs. Depending on the objectives of the experiment we may be interested in one of the two criteria for either estimating the direct treatment effects (ϕ_1) or the indirect treatment effects (ϕ_2). Initially, the two different treatments will be randomly allocated to the clustered units belonging to the social network of Figure 5.2. In doing so, and by

Figure 5.3: Modularity values for partitions of $\kappa = 2, \ldots, n/2$

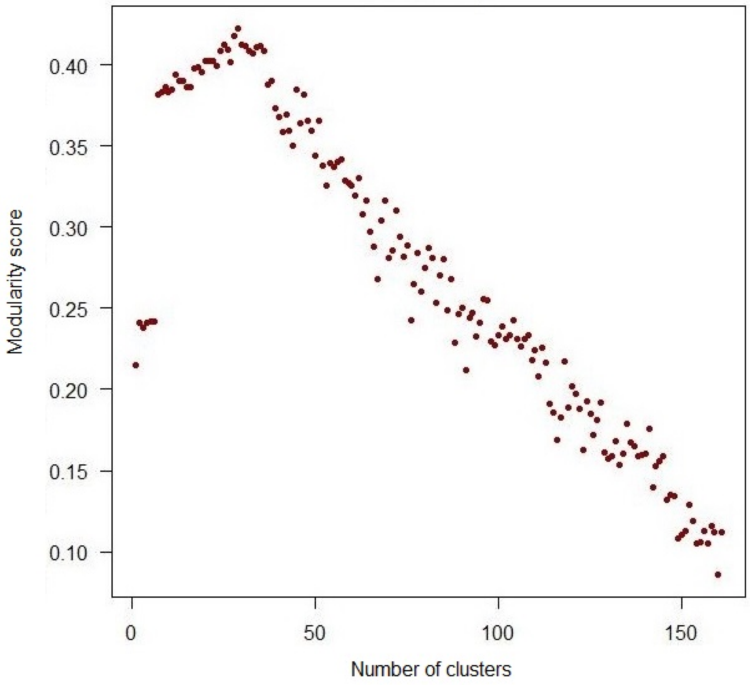
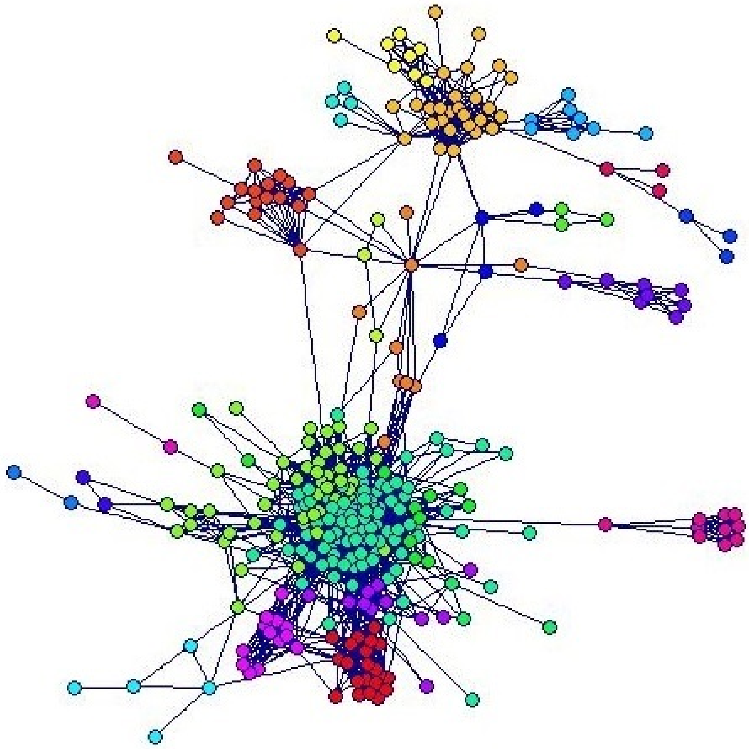


Figure 5.4: Spectral Clustering



removing all restrictions, the algorithm generates random numbers of ones and twos corresponding to the two treatments from a uniform distribution. Then it computes the information matrix which summarises this arbitrary design and calculates the design criteria. The search continues by systematically exchanging treatments, if the new treatment, which will replace the existing one, improves the design. The optimal function values resulting from this search (allowing for multiple initial designs) were found to be ${\phi_1}^* = 0.0124$ and ${\phi_2}^* = 0.0002$. Similar to the results presented in Section 3.4, ${\phi_1}^*$ is approximately equal to the estimated variance of the difference of the two treatments when the units are independent and the model does not contain network effect or blocking (i.e. $2\sigma^2 m/n = 0.0123\sigma^2$). Whilst the optimal design for estimating τ_1 is approximately balanced (with 161 and 163 units receiving treatment 1 and 2 respectively), the optimal design for estimating the difference in network effects is far from balanced (with 117 and 207 units receiving treatment 1 and 2 respectively). The optimal allocations are provided in Table B.1 in Appendix B.

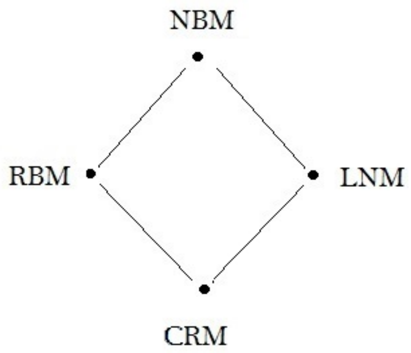
5.4 Comparison of optimal designs under different models

In this section we provide comparisons of optimal designs for estimating the direct and network effects under different models. In doing so we obtain the optimal function values of the near-optimal designs, each one corresponding to one of the two optimality criteria, on a number of different networks considering different models. Recall that the models considered in this thesis so far are

$$\begin{aligned}
\text{CRM: } y_j &= \mu + \tau_{r(j)} + \epsilon_j \\
\text{RBM: } y_{ij} &= \mu + \tau_{r(ij)} + b_i + \epsilon_{ij} \\
\text{LNM: } y_j &= \mu + \tau_{r(j)} + \sum_{h=1}^{n} A_{jh}\gamma_{r(h)} + \epsilon_j \\
\text{NBM: } y_{ij} &= \mu + \tau_{r(ij)} + b_i + \sum_{g=1}^{\kappa}\sum_{h=1}^{n_{(g)}} A_{\{ij,gh\}}\gamma_{r(gh)} + \epsilon_{ij}\,.
\end{aligned}$$

In all cases, we assume that the errors are independent and random with zero mean and constant variance. The CRM and RBM are the standard treatment models discussed in Section 2.2. The LNM (see Section 3.2) and NBM (see Section 5.3) are the extensions of CRM and RBM respectively, including a network term for capturing the connections among units. Figure 5.5 illustrates the hierarchy among the models by means of a Hasse diagram. The Hasse diagrams, named after the German mathematician Hasse, are graphic representations of ordered sets. The use of Hasse diagrams for visualising the structure in experiments in order to determine the analysis strategy were described by Taylor and Hilton (1981). Bailey (2008, Ch.10.4) and Goos and Gilmour (2012) use Hasse diagrams from a rather different perspective explaining the relationships between factors in a designed experiment and calculating the corresponding degrees of freedom. Here, we use the Hasse diagram to describe the collection of the considered models. In particular, the diagram of Figure 5.5 is a simple graph with dots representing models and lines representing nested relationships between models. If the true model is a submodel of the assumed model, the bias in estimating treatment effects will be zero. This will be further explained in following design examples.

Figure 5.5: Hasse diagram for the collection of models



Prior to moving to the design examples, we can obtain the analytical form of the bias for the treatment effect estimates, under the false assumption that there are no network effects. Let us assume that we perform an experiment where we wrongly assume that there are either no network effects or no blocking effects, while the true model is of the form NBM. The reduced model is either RBM or LNM respectively and the design matrices are of the form $X_{RBM} = (\mathbf{1} \quad {X_\tau}^\star \quad {X_b}^\star)$ and $X_{LNM} = (\mathbf{1} \quad {X_\tau}^\star \quad AX_\tau)$ with ${X_\tau}^\star$ and ${X_b}^\star$ being the treatment and block design matrices respectively imposing the standard constraints. The true (extended) design matrix is that of the NBM, i.e. $X_C = (\mathbf{1} \quad {X_\tau}^\star \quad {X_b}^\star \quad AX_\tau)$. The least squares estimator for $\boldsymbol{\beta}$ (mean, treatment and network effects) under NBM is $\hat{\boldsymbol{\beta}}_C = ({X_C}^T X_C)^{-1} {X_C}^T \boldsymbol{y}$, with $\mathbb{E}\left[\hat{\boldsymbol{\beta}}_C\right] = \boldsymbol{\beta}$. $\hat{\boldsymbol{\beta}}_R$ is an estimator of $\boldsymbol{\beta}$ fitted by OLS for either RBM (without network effects) or LNM (without block effects), i.e. $\hat{\boldsymbol{\beta}}_R = \left[({X_R}^T X_R)^{-1} {X_R}^T \boldsymbol{y} \ \mathbf{0}_m\right]$, where

$$X_R = \begin{cases} (\mathbf{1} \quad {X_\tau}^\star \quad {X_b}^\star \quad \mathbf{0}_{n\times m}) & \text{under the RBM,} \\ (\mathbf{1} \quad {X_\tau}^\star \quad \mathbf{0}_{n\times(\kappa-1)} \quad AX_\tau) & \text{under the LNM.} \end{cases}$$

The difference between the estimators under the wrong assumption is

$$\begin{aligned} \mathbb{E}\left[\hat{\boldsymbol{\beta}}_R - \hat{\boldsymbol{\beta}}_C\right] &= \mathbb{E}\left[({X_R}^T X_R)^- {X_R}^T \boldsymbol{y} - ({X_C}^T X_C)^- {X_C}^T \boldsymbol{y}\right] \\ &= \left(({X_R}^T X_R)^- {X_R}^T - ({X_C}^T X_C)^- {X_C}^T\right) \mathbb{E}\left[\boldsymbol{y}\right] \\ &= \left(({X_R}^T X_R)^- {X_R}^T - ({X_C}^T X_C)^- {X_C}^T\right) X_C \boldsymbol{\beta} \\ &= \left(({X_R}^T X_R)^- {X_R}^T X_C - I\right) \boldsymbol{\beta}. \end{aligned}$$

For the special case of $m = 2$ treatments and $\kappa = 3$ blocks, for instance, we can easily write the bias in treatment effects, under the assumption that network effects are present although we do not account for them (i.e. RBM). For the derivation of this result refer to Appendix B, where we also provide the bias in treatment effects for κ blocks and two treatments as a function of the network effects. We have

$$\text{Bias}(\hat{\tau}_1) = \beta_{\gamma_1}\gamma_1 + \beta_{\gamma_2}\gamma_2 = \frac{\Gamma_1}{D}\gamma_1 + \frac{\Gamma_2}{D}\gamma_2, \tag{5.2}$$

where

$$\begin{aligned}\Gamma_1 &= l_{(1)1}n_{(1)1}n_{(2)}\left(n-n_{(2)}\right)+l_{(2)1}n_{(2)1}n_{(1)}\left(n-n_{(1)}\right)+n_{(1)}n_{(2)}\big[l_1n_1-l_{11}n\\ &\quad -l_1n_{(1)1}-l_1n_{(2)1}+l_{11}(n_{(1)}+n_{(2)})-n_1\left(l_{(1)1}+l_{(2)1}\right)+l_{(1)1}n_{(2)1}+l_{(2)1}n_{(1)1}\big],\\ \Gamma_2 &= l_{(1)2}n_{(1)1}n_{(2)}\left(n-n_{(2)}\right)+l_{(2)2}n_{(2)1}n_{(1)}\left(n-n_{(1)}\right)+n_{(1)}n_{(2)}\big[l_2n_1-l_{12}n\\ &\quad -l_2n_{(1)1}-l_2n_{(2)1}+l_{12}(n_{(1)}+n_{(2)})-n_1(l_{(1)2}+l_{(2)2})+l_{(1)2}n_{(2)1}+l_{(2)2}n_{(1)1}\big]\end{aligned}$$

and

$$\begin{aligned}D &= n_{(1)}n_{(2)}\left(n_1^2-2n_1n_{(1)1}-2n_1n_{(2)1}+n_1n_{(1)}+n_1n_{(2)}-nn_1+2n_{(1)1}n_{(2)1}\right)\\ &\quad +n_{(1)1}^2n_{(2)}\left(n-n_{(2)}\right)+n_{(2)1}^2n_{(1)}\left(n-n_{(1)}\right).\end{aligned}$$

Revisiting Example 5.3.1 (the working example of this chapter), we obtain optimal designs assuming each of the earlier mentioned models for the social network of Figure 5.2. Designs for these models are labelled as CRD, RBD, LND and NBD. We will then compare the optimal properties of each design based on the underlying model. Under the assumption of independent errors that have a common variance σ^2, the variance-covariance matrix of the least squares estimator $\hat{\boldsymbol{\beta}}$ is $var(\hat{\boldsymbol{\beta}}) = \sigma^2(X^TX)^{-1}$. Given that interest lies in the comparisons of the designs, the value σ^2 is not relevant since the value is the same if the model is identical for all proposed designs for a particular experiment. The comparisons of the (near-) optimal designs under ϕ_1 (for estimating the direct treatment effect) and under ϕ_2 (for estimating the indirect treatment effect or network effect) are given in Tables 5.1 and 5.2. We are able to estimate the network effects only under the LNM and NBM. Moreover, the values for CRD and RBD are the mean values over a large number of possible randomisations (we have taken into account 50000 balanced designs out of a total number $n!/((n/m)!)^m = 324!/((324/2)!)^2$). The values for LND and NBD are unique.

Table 5.1: Comparisons of the designs for ϕ_1 under different models

Models	**Optimal designs for $\phi_1(\times 10^2)$**			
	CRD	**RBD**	**LND**	**NBD**
CRM	**1.2346**	1.2347	1.2346	1.2346
RBM	1.3298	**1.2432**	1.3042	1.2432
LNM	1.2481	1.2747	**1.2346**	1.2348
NBM	1.3749	1.2907	1.3177	**1.2432**

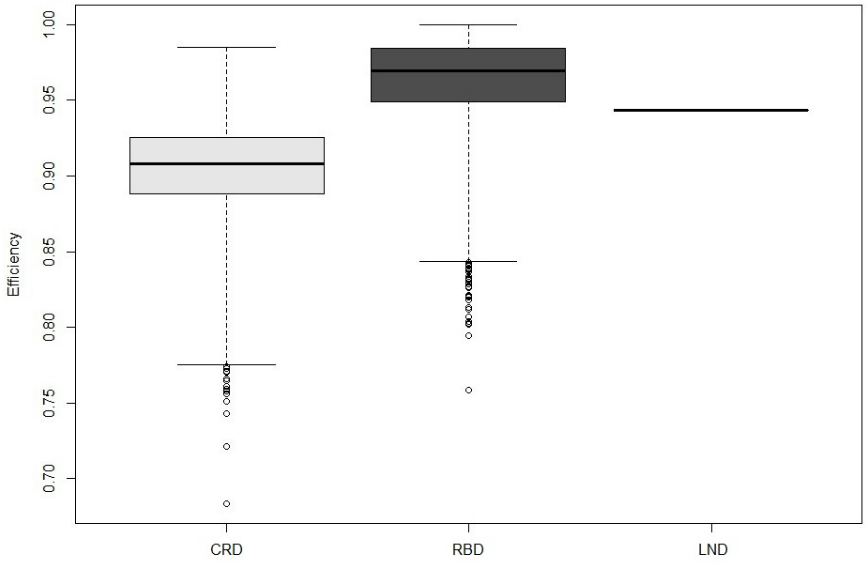
Table 5.2: Comparisons of the designs for ϕ_2 under different models

Models	**Optimal designs for $\phi_2(\times 10^2)$**			
	CRD	**RBD**	**LND**	**NBD**
LNM	0.1121	0.1474	**0.0119**	0.0149
NBM	0.1553	0.1647	0.0312	**0.0230**

Tables 5.1 and 5.2 illustrate the optimality function values for each design (first row) under the different models (first column). We can obtain the efficiencies of the designs with respect to the optimal designs. The criterion values of the optimal designs are on the diagonal of the matrix. Recall that the smaller the criterion value the better

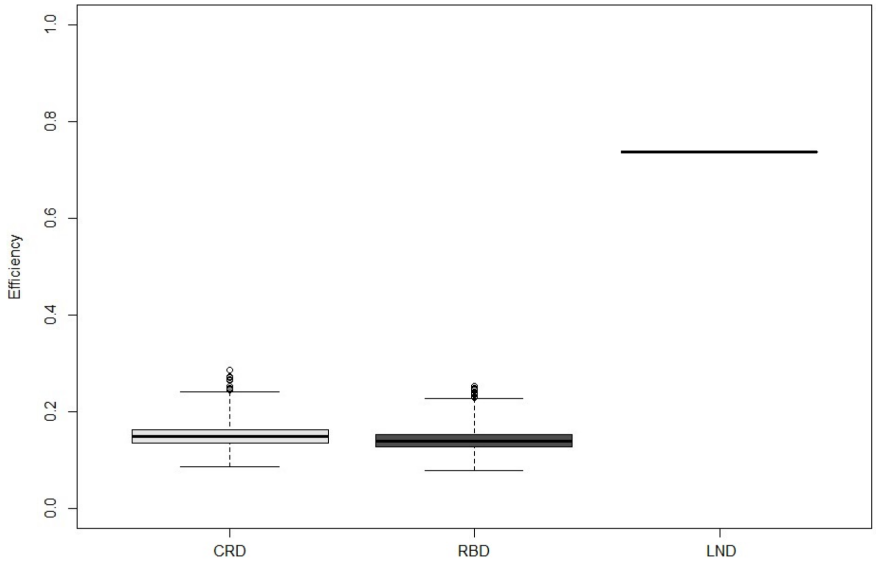
the design is. A key observation derived from these tables is that a randomised design which ignores the network effects is on average highly inefficient, at least with respect to ϕ_2. We can observe that when assuming LNM to be true, the NBD, which is completely balanced within blocks, is almost as efficient as the LND (with efficiency 0.012346/0.012348 = 99.9%). Under NBM the LND, which is balanced overall but unbalanced with respect to blocks, is 93.9% efficient. Moreover, under NBM, a random balanced RBD which is balanced within blocks performs better than LND on average, with the latter being as efficient as a random balanced CRD on average. In Table 5.2 the random balanced designs CRD and RBD perform poorly under both models with their efficiencies ranging on average between 10.6% and 14.8% respectively when ignoring blocks and 8.1% and 13.9% respectively when accounting for blocks. These low efficiencies can be justified given that these standard designs ignore the spillover effects resulting from the network structure among units.

Figure 5.6: Boxplots of efficiencies for ϕ_1



The boxplots in Figure 5.6 depict the efficiencies under L-optimality of the design based on ϕ_1 for a number of possible balanced designs (as chosen at random) for CRD and RBD which do not take into account network effects, although they exist. The outliers are detected by setting the upper/lower ends of the whiskers at three standard deviations. For ϕ_1 CRD and RBD have median efficiencies 0.92 and 0.97 respectively, with minimum efficiencies 0.68 and 0.76 respectively. On the other hand for ϕ_2 in Figure 5.7, CRD and RBD perform similarly with median efficiencies between 0.10 and 0.25, lower quartiles between 0.15 and 0.17 and upper quartiles between 0.27 and 0.29. This results from not taking into account the network effects. For ϕ_1 LND performs on average better than most of the balanced CRDs, but worse than the majority of balanced RBDs, whilst for ϕ_2 LND is approximately 70% as efficient as NBD. On closer inspection of the values of Tables 5.1 and 5.2 under NBM together with the corresponding boxplots, we can make the following observations:

Figure 5.7: Boxplots of efficiencies for ϕ_2

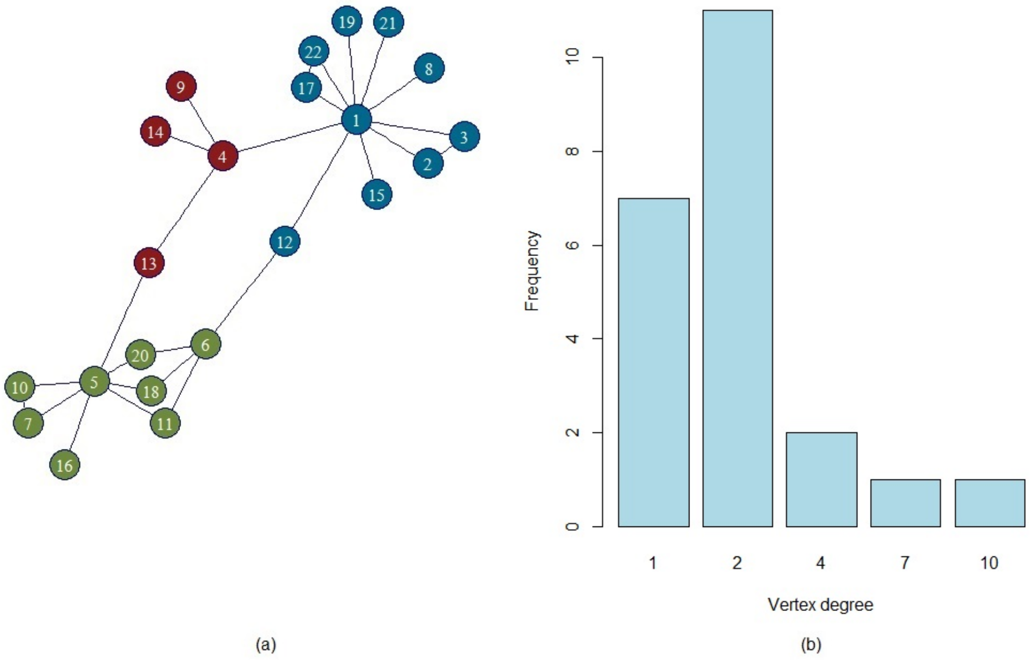


- On average the randomisation process leads to fairly poor designs, i.e. high variance (this is especially true under the second optimality criterion).
- When ignoring both blocking and network effects, the design can perform poorly.
- When taking into account the network effects but not the blocking effects, the design performs satisfactorily well.
- By using blocks and ignoring network effects, with a 25% chance (upper quartile) one can do just as well as when taking into account the network effects, but on average one does worse.
- Evidence suggests that we will be better off by using blocks instead of ignoring them independently of whether we are taking into account network effects.

Similarly to the findings presented in Section 3.4, the optimal designs under the second optimality criterion are generally unbalanced. Therefore, it is not surprising that the standard designs perform poorly with median local efficiencies lower than 30%. Thus in practice, when there is a strong belief that the units are governed by a network structure and/or when community structure is detected, then it is of importance to account for those in order to obtain an efficient design.

Example 5.4.1. The social network in Figure 5.8(a) is a co-authorship network, comprising 22 subjects (PhD students, supervisors and co-authors) and 27 edges, which indicate the ties between individuals with common publications or ties between PhD students and their supervisors in University of Southampton in the field of experimental design during a certain period in year 2015. The three blocks have been defined using spectral clustering techniques (at the maximum modularity value of 0.51). We also calculate some network measures; the average degree $\delta = 2.45$, the clustering coefficient $\mathcal{C} = 0.10$ and the average path length $\ell = 2.87$. The degree distribution is illustrated in Figure 5.8(b). We want to conduct an experiment on this network to compare two distinct treatments.

Figure 5.8: (a) Social network with 3 clusters; different colours indicate different clusters; (b) degree distribution

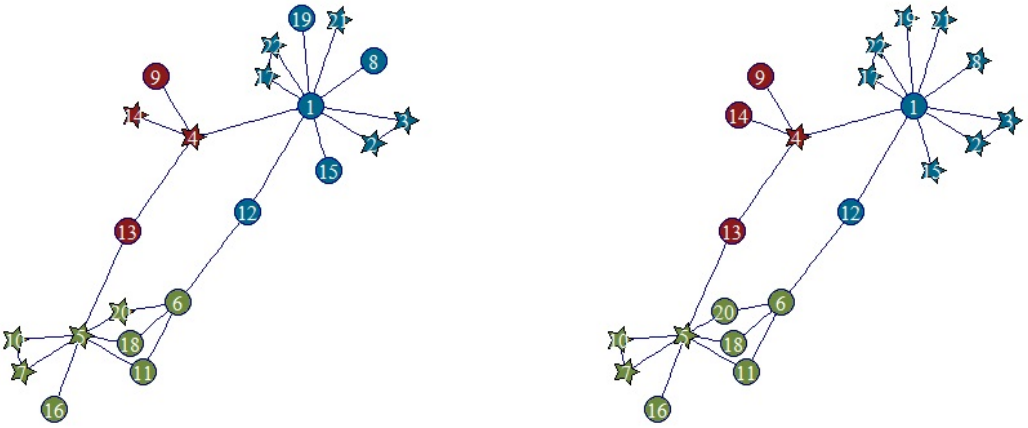


Design efficiency

Similarly to the previous example, we obtain the L-optimal designs assuming the earlier mentioned models for this network. The optimal values for CRD and RBD are the mean values (mean variances) over all possible balanced randomisations (i.e. the number of CRD and RBD are $\frac{1}{2}\binom{22}{11} = 352716$ and $\frac{1}{2}\binom{10}{5}\binom{8}{4}\binom{4}{2} = 52920$ respectively) while the values for LND and NBD are unique. Notice at this point that there are 16 optimal treatment arrangements for LND and 26 for NBD under ϕ_1. These are related to the ϕ_1- automorphisms we discussed in Chapter 4, e.g. there are $\binom{4}{1}\binom{4}{1}$ combinations for swapping the vertices in the star subgraph without altering its adjacencies resulting in 16 global designs for LND. Likewise one can observe that there are 27 $(= \binom{4}{1}\binom{3}{1}\binom{2}{1} + 3)$ combinations for NBD (see Appendix B). Whilst in the case of ϕ_2 there is only one optimal design.

Figure 5.9 illustrates the optimal designs for the two criteria, providing supportive evidence about the patterns of the optimally allocated treatments discussed in Section 3.4. In particular, overall the design for ϕ_1 is balanced with all treatments being equally allocated to the subjects ($n_1 = n_2$), i.e. 11 subjects receiving each of the two treatments, who have similar degree ($l_1 \approx l_2$ and $l_{11} \approx l_{12} \approx l_{22}$) (equality does not hold here because the number of treatments is not divisible by the number of connections). The optimal design for ϕ_1, is also balanced within each block. Moreover the optimal function values, which are 0.1822 (under LNM) and 0.1828 (under NBM) are very close to the minimum average variance possible for the unstructured case when having independent subjects under SUTVA (that is $0.1818\sigma^2$). Note that in the case that the number of subjects within each block is not divisible by the number of treatments, then the design is not perfectly balanced. For ϕ_2, the subjects located at the ends (leaves) receive the same treatment that is opposite to the treatments of their immediate neighbours of higher degree.

Figure 5.9: Optimal designs (under NBM), ϕ_1 (left) and ϕ_2 (*right*); different colours indicate different blocks and different vertex' shapes different treatments



Tables 5.3 and 5.4 illustrate the optimal function values for each design under the different models. Similar to the results of the previous example, the values highlight that, when we randomise, our designs on average perform worse. When assuming LNM to be true, the NBD is only just slightly worse than optimal with 99.7% efficiency. On the contrary when assuming NBM to be true, LND is only 50.5% efficient (see also Figure 5.10). Recall that NBD for ϕ_1 is balanced overall but also balanced within blocks. The situation is similar to the design efficiencies under ϕ_2, meaning that under NBM, LND has a low efficiency of 38.9% as a result of ignoring the blocks, while under LNM, NBD is 64.8% efficient. As we will show later, the optimisation process for LNM drives the balance property away from having an equal treatment allocation within blocks. This means that, if we account for the blocks, we almost certainly do better. Figures 5.10 and 5.11 depict the efficiencies under ϕ_1 and ϕ_2 respectively, for CRD, RBD and LND where the first two do not take into account the network effects, although we assume that they exist and act according to the NBM.

Table 5.3: Comparisons of the designs for ϕ_1 under different models

Models	**Optimal designs**			
	CRD	**RBD**	**LND**	**NBD**
CRM	**0.1818**	0.1818	0.1818	0.1818
RBM	0.2034	**0.1818**	0.2685	0.1818
LNM	0.2126	0.2191	**0.1822**	0.1827
NBM	0.2500	0.2270	0.3621	**0.1828**

Figure 5.10: Boxplots of efficiencies for ϕ_1 (assuming NBM)

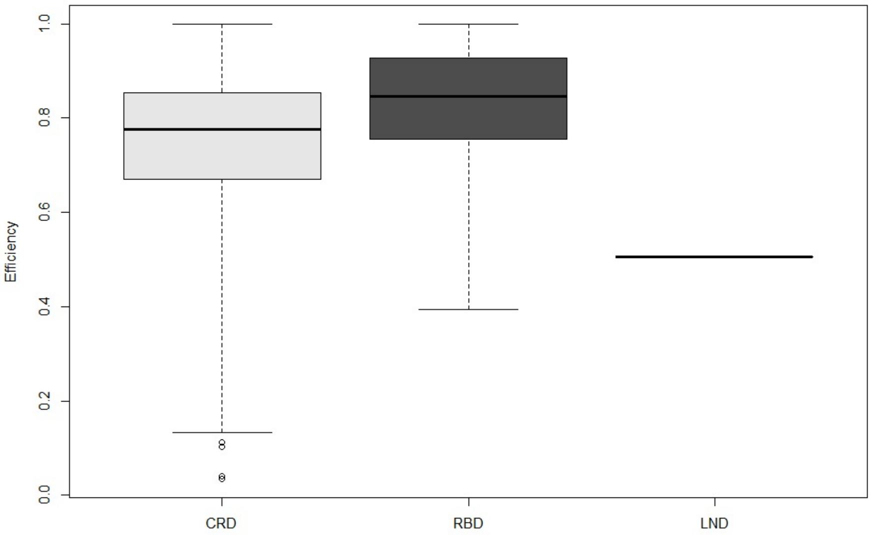
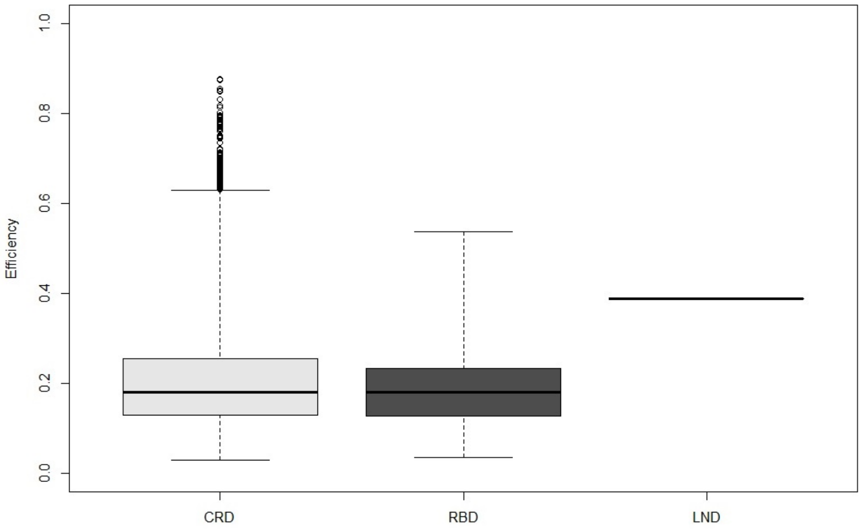


Table 5.4: Comparisons of the designs for ϕ_2 under different models

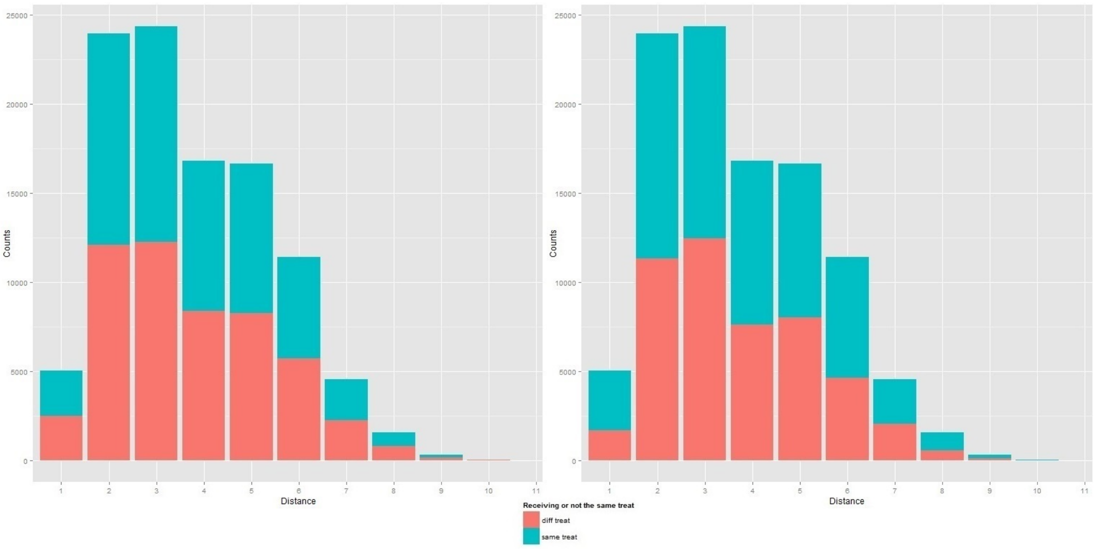
Models	**Optimal designs**			
	CRD	**RBD**	**LND**	**NBD**
LNM	0.1447	0.1927	**0.0237**	0.0366
NBM	0.2354	0.2503	0.0998	**0.0388**

Figure 5.11: Boxplots of efficiencies for ϕ_2 (assuming NBM)



The poor design performance of LND relatively to NBD under NBM results from having a completely different structure and different allocation to clusters (which define the blocks). However, we want to explore further this low efficiency. For this reason, we investigate the occurrences of treatments between neighbours. As a first attempt to explore this, we obtain Figure 5.12 which illustrates how the distance between neighbours affects the design. It provides the distances among units receiving the same (blue colour) or different (pink colour) treatments. For example, for ϕ_1 there are as many pairs of immediate neighbours receiving the same treatment as those receiving different treatments, whilst for ϕ_2 there are twice as many pairs of immediate neighbours receiving the same treatment as those receiving different treatments. In other words, immediate neighbours tend to receive the same treatment in the case of ϕ_1, which does not hold in general for ϕ_2. A similar pattern holds for the pairs of neighbours of distance two (note that the model takes into account distances up to two).

Figure 5.12: Pairs of units of different distances receiving the same (*blue*) or different (*pink*) treatments for ϕ_1 (*left*) and ϕ_2 (*right*)



Tables 5.5 and 5.6 reveal the distribution of these patterns within blocks and for every distance. In particular, these tables provide a better insight into the patterns of neighbours sharing a treatment when increasing the order of the mutual distance from one to two. Distance 0 represents replication of the treatments. The advantage of NBD is not in terms of replication of treatments or first order connections. The benefit of NBD over LND may be related to the second order connections. Table 5.5 explores the optimal pattern of the allocated treatments when ignoring or accounting for blocks. A key observation under ϕ_1 for distance 2 is that for the LND the number of pairs of neighbours receiving the same treatment is almost equal to the number of pairs of units receiving different treatments, whilst for NBD there are twice as many pairs of neighbours receiving different treatments compared to the pairs of neighbours receiving the same treatment. It is important to note that the blocks are not equally-sized and are relative small. Thus, because we have relatively small blocks and given that for NBD we have balanced treatment allocation within blocks, there are more vertices in the close neighbourhood that receive the opposite treatment. In other words, under NBD there is an equal number of the two different treatments in the neighbourhood, which forces more paths (of length two) connecting units receiving treatment one to units receiving treatment two. This does not hold for LND, since the balance is retained overall but not in close neighbourhoods and therefore many paths of length two

actually connect units receiving the same treatment. Hence, one of the main reasons for the poor performance of LND is the unequal and relatively small-sized blocks.

Table 5.5: Pairs of neighbouring subjects receiving the same/different treatment

	LND						**NBD**					
	ϕ_1			ϕ_2			ϕ_1			ϕ_2		
	$\circ\,\circ$	$\circ\,\star$	$\star\,\star$	$\circ\,\circ$	$\circ\,\star$	$\star\,\star$	$\circ\,\circ$	$\circ\,\star$	$\star\,\star$	$\circ\,\circ$	$\circ\,\star$	$\star\,\star$
Distance 0	11		11	13		9	11		11	10		12
Distance 1	7	13	7	13	13	1	7	13	7	5	17	5
Distance 2	30	31	26	55	8	24	22	42	23	20	23	44

Table 5.6: Table 5.5 with the distribution within blocks

		LND						**NBD**					
		ϕ_1			ϕ_2			ϕ_1			ϕ_2		
		$\circ\,\circ$	$\circ\,\star$	$\star\,\star$	$\circ\,\circ$	$\circ\,\star$	$\star\,\star$	$\circ\,\circ$	$\circ\,\star$	$\star\,\star$	$\circ\,\circ$	$\circ\,\star$	$\star\,\star$
D. 0	Block 1	3		1	1		3	2		2	3		1
	Block 2	1		7	2		6	4		4	5		3
	Block 3	7		3	10		0	5		5	2		8
D. 1	Block 1	0	1	2	0	3	0	0	2	1	0	2	1
	Block 2	0	3	4	1	5	1	2	3	2	3	3	1
	Block 3	7	6	0	10	3	0	4	7	2	1	11	1
	btw blocks	0	3	1	2	2	0	1	1	2	1	1	2
D. 2	Block 1	0	2	1	2	1	0	2	1	0	1	2	0
	Block 2	5	7	5	9	1	7	6	9	2	6	5	6
	Block 3	12	16	11	28	1	10	5	19	15	7	11	21
	btw blocks	13	6	9	16	5	7	9	13	6	6	5	17

Design bias

We obtain the biases under the different models with respect to the model parameters in Tables 5.7 and 5.8. We can discern that if the network effects are substantial compared to the (direct) treatment effects, then ignoring them can potentially lead to over-/under-estimated treatment effects. Thus by not taking into account network effects in our design, we produce an experiment which can have higher variance than necessary and biased estimates.

Consider the following example in order to understand how we can interpret these quantities. We can extract the bias related to the treatment effects (second line in each matrix). Assuming that we wrongly ignore network effects, we have the following two cases

$$\text{CRD} \mid \text{LNM:} \qquad \mathbb{E}\left[\hat{\tau}_1\right] = -0.48\gamma_1 - 0.24\gamma_2, \tag{5.3}$$

$$\text{RBD} \mid \text{NBM:} \qquad \mathbb{E}\left[\hat{\tau}_1\right] = -0.72\gamma_1 - 0.05\gamma_2. \tag{5.4}$$

Two situations can occur in the network. One is the interference among neighbours (network effects) and the other relates to grouping of responses according to the clus-

Table 5.7: Design bias under model misspecification (ϕ_1)

True model	**Optimal designs**			
	CRD	**RBD**	**LND**	**NBD**
CRM	$\mathbf{0}_{2\times 2}$	$\mathbf{0}_{2\times 2}$	$\mathbf{0}_{2\times 2}$	$\mathbf{0}_{2\times 2}$
RBM	$\begin{bmatrix} 0 & 0 & 0.17 & 0.34 \\ 0 & 0 & 0.02 & 0.03 \\ 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & -1 \end{bmatrix} \begin{pmatrix} \mu \\ \tau_1 \\ b_1 \\ b_2 \end{pmatrix}$	$\mathbf{0}_{4\times 4}$	$\begin{bmatrix} 0 & 0 & 0.32 & 0.08 \\ 0 & 0 & -0.18 & 0.52 \\ 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & -1 \end{bmatrix} \begin{pmatrix} \mu \\ \tau_1 \\ b_1 \\ b_2 \end{pmatrix}$	$\mathbf{0}_{4\times 4}$
LNM	$\begin{bmatrix} 0 & 0 & 1.28 & 1.53 \\ 0 & 0 & -0.48 & -0.24 \\ 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & -1 \end{bmatrix} \begin{pmatrix} \mu \\ \tau_1 \\ \gamma_1 \\ \gamma_2 \end{pmatrix}$	$\begin{bmatrix} 0 & 0 & 1.18 & 1.60 \\ 0 & 0 & -0.72 & -0.05 \\ 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & -1 \end{bmatrix} \begin{pmatrix} \mu \\ \tau_1 \\ \gamma_1 \\ \gamma_2 \end{pmatrix}$	$\mathbf{0}_{4\times 4}$	$\mathbf{0}_{4\times 4}$
NBM	$\begin{bmatrix} 0 & 0 & 0.17 & 0.34 & 1.28 & 1.53 \\ 0 & 0 & 0.02 & 0.03 & -0.48 & -0.24 \\ 0 & 0 & -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 0 & 0 & -1 \end{bmatrix} \begin{pmatrix} \mu \\ \tau_1 \\ b_1 \\ b_2 \\ \gamma_1 \\ \gamma_2 \end{pmatrix}$	$\begin{bmatrix} 0 & 0 & 0 & 0 & 1.18 & 1.60 \\ 0 & 0 & 0 & 0 & -0.72 & -0.05 \\ 0 & 0 & 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 0 & 0 & -1 \\ 0 & 0 & 0 & 0 & 0.05 & -0.45 \\ 0 & 0 & 0 & 0 & 0.56 & -0.21 \end{bmatrix} \begin{pmatrix} \mu \\ \tau_1 \\ b_1 \\ b_2 \\ \gamma_1 \\ \gamma_2 \end{pmatrix}$	$\begin{bmatrix} 0 & 0 & 0.32 & 0.08 & 0 & 0 \\ 0 & 0 & -0.18 & 0.52 & 0 & 0 \\ 0 & 0 & -0.01 & 0.15 & 0 & 0 \\ 0 & 0 & -0.02 & -0.13 & 0 & 0 \\ 0 & 0 & -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & -1 & 0 & 0 \end{bmatrix} \begin{pmatrix} \mu \\ \tau_1 \\ b_1 \\ b_2 \\ \gamma_1 \\ \gamma_2 \end{pmatrix}$	$\mathbf{0}_{6\times 6}$

Table 5.8: Design bias under model misspecification (ϕ_2)

True model	**Optimal designs**			
	CRD	**RBD**	**LND**	**NBD**
LNM	$\begin{bmatrix} 0 & 0 & 1.28 & 1.53 \\ 0 & 0 & -0.48 & -0.24 \\ 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & -1 \end{bmatrix}\begin{pmatrix} \mu \\ \tau_1 \\ \gamma_1 \\ \gamma_2 \end{pmatrix}$	$\begin{bmatrix} 0 & 0 & 1.18 & 1.60 \\ 0 & 0 & -0.72 & -0.05 \\ 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & -1 \end{bmatrix}\begin{pmatrix} \mu \\ \tau_1 \\ \gamma_1 \\ \gamma_2 \end{pmatrix}$	$\mathbf{0}_{4\times 4}$	$\mathbf{0}_{4\times 4}$
NBM	$\begin{bmatrix} 0 & 0 & 0.17 & 0.34 & 1.28 & 1.53 \\ 0 & 0 & 0.02 & 0.03 & -0.48 & -0.24 \\ 0 & 0 & -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 0 & 0 & -1 \end{bmatrix}\begin{pmatrix} \mu \\ \tau_1 \\ b_1 \\ b_2 \\ \gamma_1 \\ \gamma_2 \end{pmatrix}$	$\begin{bmatrix} 0 & 0 & 0 & 0 & 1.18 & 1.60 \\ 0 & 0 & 0 & 0 & -0.72 & -0.05 \\ 0 & 0 & 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 0 & 0 & -1 \\ 0 & 0 & 0 & 0 & 0.05 & -0.45 \\ 0 & 0 & 0 & 0 & 0.56 & -0.21 \end{bmatrix}\begin{pmatrix} \mu \\ \tau_1 \\ b_1 \\ b_2 \\ \gamma_1 \\ \gamma_2 \end{pmatrix}$	$\begin{bmatrix} 0 & 0 & 0.08 & -0.02 & 0 & 0 \\ 0 & 0 & 0.26 & 0.64 & 0 & 0 \\ 0 & 0 & 0.02 & 0.16 & 0 & 0 \\ 0 & 0 & -0.01 & 0.01 & 0 & 0 \\ 0 & 0 & -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & -1 & 0 & 0 \end{bmatrix}\begin{pmatrix} \mu \\ \tau_1 \\ b_1 \\ b_2 \\ \gamma_1 \\ \gamma_2 \end{pmatrix}$	$\mathbf{0}_{6\times 6}$

tering of units. The connections among units affect the interference among them, which implies that the network effects are the same for units irrespective of falling within the same block. The treatment allocation affects the bias of the estimated (direct) treatment effects due to the network effects. We should note that the network effects are not in general of the same size. However, for our investigation we assume that the network effects are of the same magnitude ($\gamma_1 = \gamma_2 = \gamma$) and without loss of generality we set $\gamma = 1$ (see Section 3.5.2).

For the case of Equations (5.3) and (5.4), if we average out the coefficients corresponding to the network effects the bias in (5.4) is slightly larger than the corresponding one in (5.3), i.e. $|0.39| > |0.36|$. Note that the biases of the estimated treatment effects are unequal, which may be attributed to the block sizes. We would expect the opposite to hold, that is the bias for the case of (5.4) (which is the same to that of RBD | LNM) to be much less than for the case of (5.3), since we still take account of the clustering as specified by the network structure and therefore we protect against bias from losing the network effects. A large difference in bias for the case of (5.4) compared to that of (5.3) would suggest that we lose more in the presence of the blocks than in their absence. A conclusion drawn from this example is that if we use a model that ignores network effects there is no obvious benefit in terms of bias from including blocks. Thus having blocks does not insure us against the bias introduced by wrongly excluding network effects.

Figure 5.13 illustrates the bias (when assuming an optimal RBD under the true NBM) in the treatment effect estimates introduced from network effects against the proportion of edges, which connect pairs of subjects receiving different treatments (when we assume that the underlying parameters are one). The expectation of bias over all possible balanced designs (RBDs) for the intersection point, which corresponds to approximately half for the proportion of links, is equal to zero. Moreover, the boxplots in Figure 5.14 depict the bias in the estimation of treatment effects due to the network effects over all possible balanced RBDs in the case that we assume that $\gamma_1 = \gamma_2$ for every proportion of links. The limits of the boxplots are relative to the size of the true network effect. This plot suggests that in general the complete randomisation does not perform well. In combination with the previous observation, we can conclude that the least expected bias in the treatment effects is achieved when the number of pairs of connected units who receive different treatments equals approximately half of the total number edges of the network (see the boxplot at the proportion 0.56 in Figure 5.14). The number of balanced designs for each proportion of edges (connecting units receiving treatment 1 to units receiving treatment 2) is given below:

0.37	**0.41**	**0.44**	**0.48**	**0.52**	**0.56**	**0.59**	**0.63**	**0.67**	**0.7**	**0.74**
38	322	1532	4710	9460	12742	11838	7674	3540	952	112

Table B.2 in Appendix B provides the precise information of the coordinates of Figure 5.13 for each proportion of edges (connecting 1s to 2s) as captured by l_{12}/l. Similarly

Figure 5.13: Bias in treatment effects due to network effects

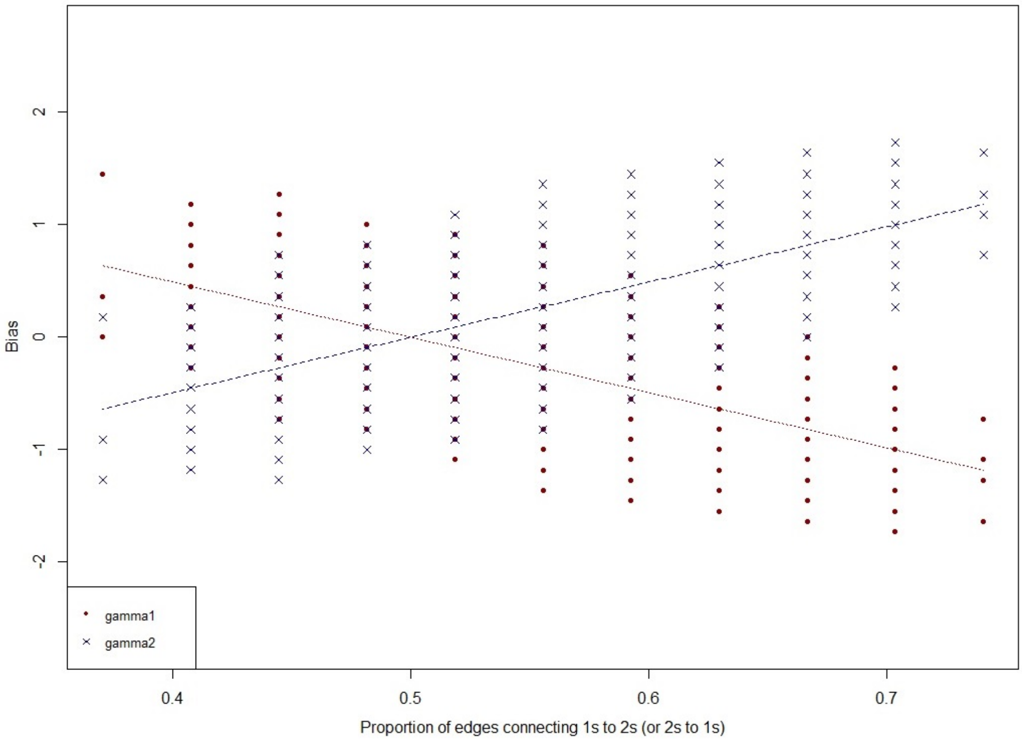
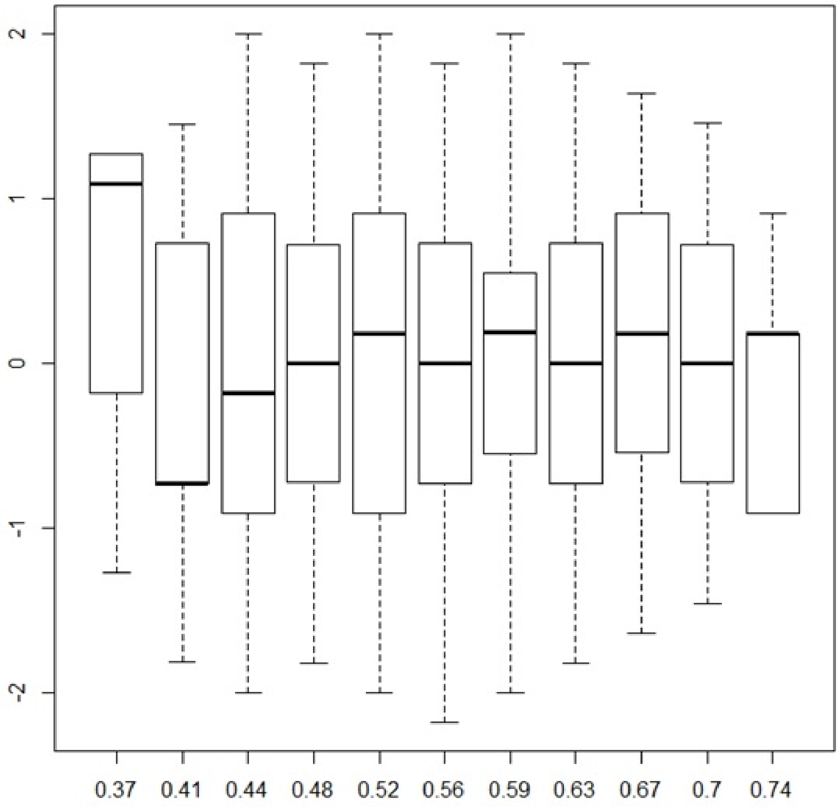


Figure 5.14: Bias for treatment effects when $\gamma_1 = \gamma_2$



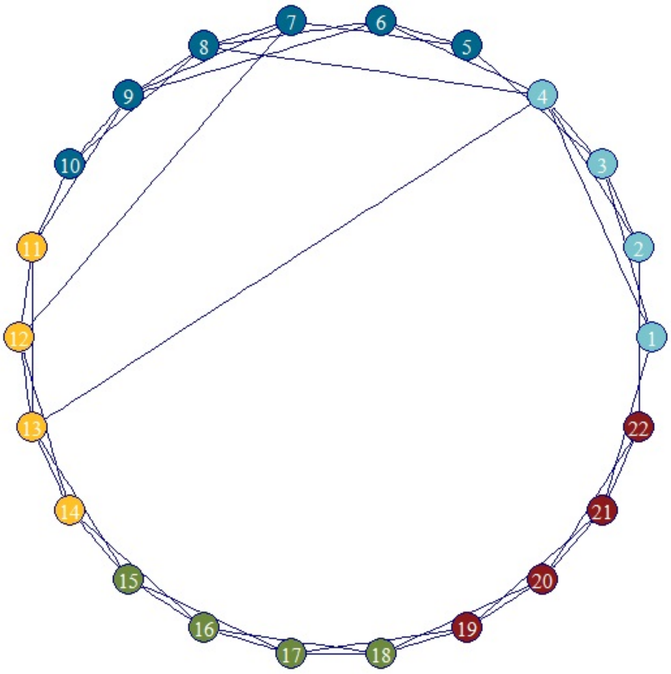
to Section 3.5.2, the bias due to γ_2 equals

$$\beta_{\gamma_2} = \beta_{\gamma_1} + \psi \left(\frac{l_{12}}{l} - 0.5 \right)$$

where ψ is a constant related to the given network.

Example 5.4.2. In this example, we generate a small-world network using the Watts-Strogatz model. The network is of equal size to the one in the previous example comprising 22 nodes and 44 edges. This example serves to illustrate that the efficiencies and biases are of the same scale depending on the size of the network rather than the connections, but also there is an interesting observation associated with this type of network. The typical features underlying such network are the small average shortest path length and a clustering coefficient significantly larger than expected by random chance ($\delta = 4$, $\mathcal{C} = 0.37$ and $\ell = 2.64$). The latter implies that the graph is highly clustered around a few vertices which result in well-defined blocks rendering block designs with network effects preferable.

Figure 5.15: Social network with 5 clusters; different colours indicate different clusters



We obtain the L-optimal designs based on the different models respectively for the social network of Figure 5.15 (modularity value 0.45). The values for CRD and RBD are the mean values (mean variances) over all possible randomisations, where the number of CRDs and RBDs are $\frac{1}{2}\binom{22}{11} = 352716$ and $\frac{1}{2}\binom{4}{2}^4\binom{6}{3} = 2160$ respectively. Figure 5.16 illustrates the optimal designs for the two criteria, providing supportive evidence about the patterns of the allocated treatments discussed in Section 3.4. We obtain the optimal function values for the different designs when assuming different models, which are provided in Tables 5.9 and 5.10. The criterion values in the diagonal correspond to the optimal values computed under the true model. Moreover, Figures 5.17 and 5.18 show boxplots of the L-efficiencies based on ϕ_1 and ϕ_2 for CRD, RBD and LND with respect to the NBM (which is assumed to be the true model).

Figure 5.16: Optimal designs (under NBM), ϕ_1 (*left*) and ϕ_2 (*right*); different colours indicate different blocks, and different vertex' shapes the different treatments

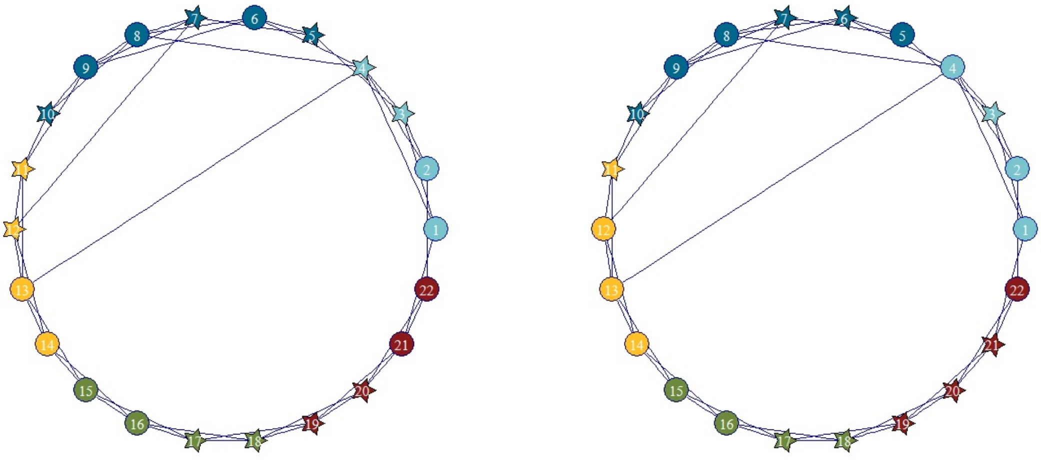


Table 5.9: Comparisons of the designs for ϕ_1 under different models

Models	**Optimal designs**			
	CRD	**RBD**	**LND**	**NBD**
CRM	**0.1818**	0.1818	0.1818	0.1818
RBM	0.2311	**0.1818**	0.2791	0.1818
LNM	0.2212	0.2980	**0.1818**	0.1818
NBM	0.3890	0.3346	0.4224	**0.1818**

Table 5.10: Comparisons of the designs for ϕ_2 under different models

Models	**Optimal designs**			
	CRD	**RBD**	**LND**	**NBD**
LNM	0.0715	0.1214	**0.0234**	0.0326
NBM	0.1597	0.1625	0.1022	**0.0517**

An interesting observation from Table 5.9 is that under LNM, ϕ_1 for NBD is (almost) the same as the optimal function value for LND (rounded), which is mainly due to the second order distances and specific network structure. As an attempt to disentangle the influence of the blocking structure, in addition to the network structure among units, on the optimal allocations of LND and NBD we provide Tables 5.11 and 5.12. At first glance, it seems that the number of pairs of units receiving different treatments is roughly the same as the number of pairs of units receiving the same treatments for both optimal designs (for distances 0 and 1) (see Table 5.11). This renders NBD fairly good compared to the LND. Since for NBD we have balanced treatment allocation within blocks, there are more vertices in the close neighbourhood that receive the

Figure 5.17: Boxplots of efficiencies for ϕ_1 (under NBM)

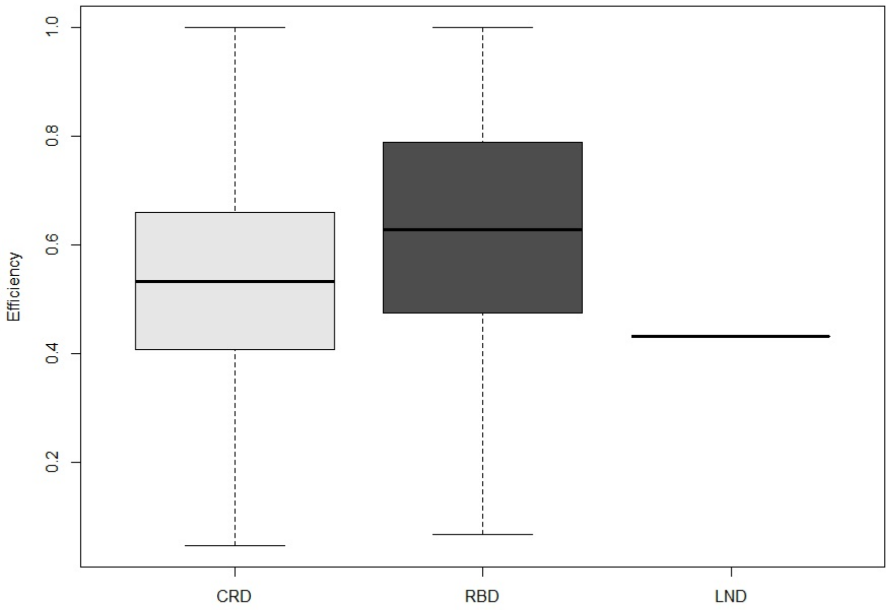
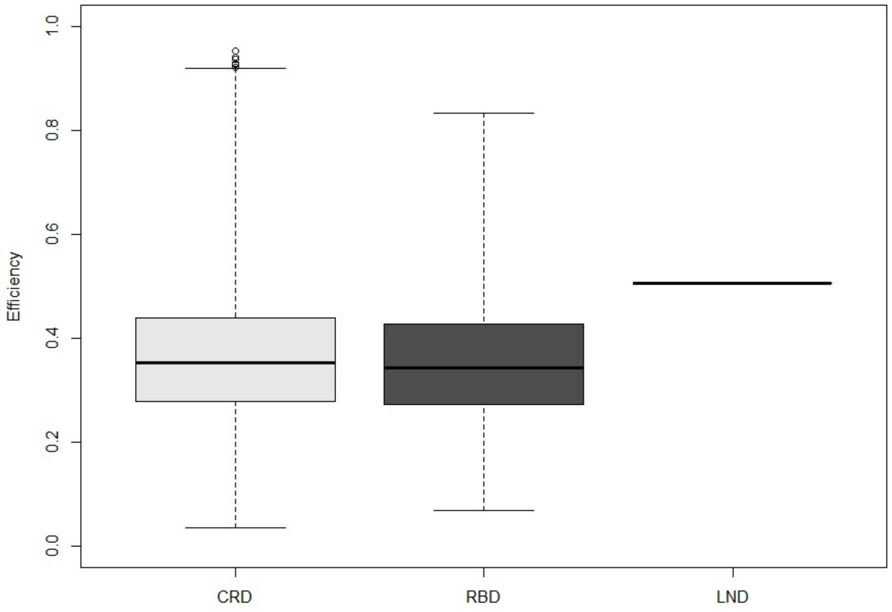


Figure 5.18: Boxplots of efficiencies for ϕ_2 (under NBM)



opposite treatment (which result in LND performing poorly under NBM). Similarly to the results discussed in the previous examples of this section, we can see that LND performs on average better than most standard designs for ϕ_2, which is not surprising if we consider that the LNM is 'more correct' in that it accounts for the network structure. However, LND is not as adequate as NBD (see both Figures 5.17 and 5.18). In practice, for networks that have high clustering coefficient, such as the small-world networks, the optimal NBD that accounts for both blocking and network effects is expected to have higher design efficiency compared to any standard design. This is because a high clustering coefficient implies a large number of clusters in the network, which lead to more vertices in the close neighbourhood receiving different treatments. As a result an LND that is balanced overall (but not balanced in neighbourhoods) performs poorly, since many paths lead to the same treatment.

Table 5.11: Pairs of neighbouring subjects receiving the same/different treatment

	LND						**NBD**					
	ϕ_1			ϕ_2			ϕ_1			ϕ_2		
	$\circ\circ$	$\circ\star$	$\star\star$	$\circ\circ$	$\circ\star$	$\star\star$	$\circ\circ$	$\circ\star$	$\star\star$	$\circ\circ$	$\circ\star$	$\star\star$
Distance 0	11		11	13		9	11		11	12		10
Distance 1	11	22	11	15	20	9	11	22	11	13	23	8
Distance 2	17	35	16	27	26	15	15	42	11	22	38	8

Table 5.12: Table 5.11 with the distribution within blocks

		LND						**NBD**					
		ϕ_1			ϕ_2			ϕ_1			ϕ_2		
		$\circ\circ$	$\circ\star$	$\star\star$	$\circ\circ$	$\circ\star$	$\star\star$	$\circ\circ$	$\circ\star$	$\star\star$	$\circ\circ$	$\circ\star$	$\star\star$
D. 0	Block 1	4		2	2		4	3		3	3		3
	Block 2	1		3	4		0	2		2	1		3
	Block 3	1		3	3		1	2		2	2		2
	Block 4	4		0	3		1	2		2	3		1
	Block 5	1		3	1		3	2		2	3		1
D. 1	Block 1	4	4	1	0	6	3	3	5	1	1	8	0
	Block 2	0	3	2	5	0	0	1	3	1	0	2	3
	Block 3	0	3	2	3	2	0	1	3	1	1	3	1
	Block 4	5	0	0	2	3	0	0	4	1	2	3	0
	Block 5	0	2	3	0	3	2	1	3	1	3	2	0
	btw blocks	2	10	3	5	6	4	5	4	6	6	5	4
D. 2	Block 1	2	1	1	0	3	1	1	3	0	0	4	0
	Block 2	0	1	2	3	0	0	1	2	0	0	2	1
	Block 3	0	1	1	1	1	0	0	2	0	0	2	0
	Block 4	4	0	0	2	2	0	0	4	0	2	2	0
	Block 5	0	1	2	0	1	2	0	2	1	1	2	0
	btw blocks	11	31	10	21	19	12	13	29	10	19	26	7

Tables 5.13 and 5.14 provide the expected bias with respect to the parameters under the true model for each design, after multiplying these matrices by the corresponding vector of parameters, which: under RBM is $(\mu \quad \tau_1 \quad b_1 \quad b_2 \quad b_3)^T$; under LNM is $(\mu \quad \tau_1 \quad \gamma_1 \quad \gamma_2)^T$; and under NBM is $(\mu \quad \tau_1 \quad b_1 \quad b_2 \quad b_3 \quad \gamma_1 \quad \gamma_2)^T$. For instance, if we assume LNM to be the true model, the expected bias introduced in the treatment effects of all the corresponding balanced designs CRD and RBD are positively weighted by the network effect due to the second treatment, i.e.

$$\text{CRD} \mid \text{LNM:} \quad \mathbb{E}\left[\hat{\tau}_1\right] - \tau_1 = -0.14\gamma_1 + 0.24\gamma_2; \tag{5.5}$$

$$\text{RBD} \mid \text{LNM:} \quad \mathbb{E}\left[\hat{\tau}_1\right] - \tau_1 = -0.71\gamma_1 + 0.83\gamma_2. \tag{5.6}$$

We emphasise at this point that similar to Example 5.4.1 there are more than one optimal treatment arrangements for LND and for NBD under ϕ_1. In particular there are 16695 LNDs and 86 NBDs that return the same optimal function value. On the

Table 5.13: Design bias under model misspecification for the small-world network (ϕ_1)

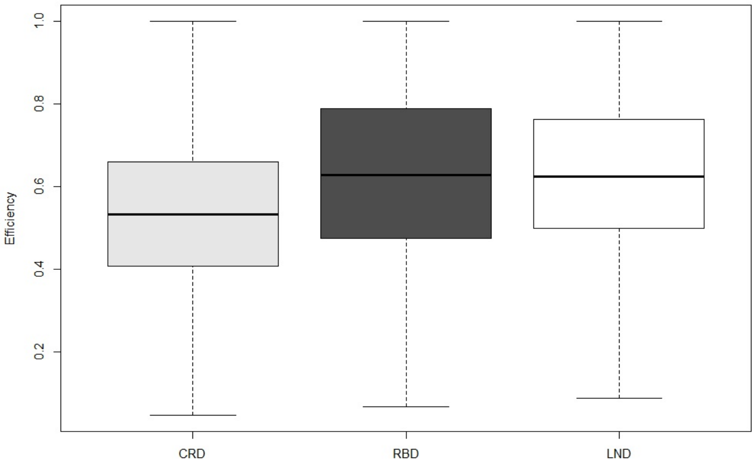
True model	**Optimal designs**			
	CRD	**RBD**	**LND**	**NBD**
CRM	$\mathbf{0}_{2\times 2}$	$\mathbf{0}_{2\times 2}$	$\mathbf{0}_{2\times 2}$	$\mathbf{0}_{2\times 2}$
RBM	$\begin{bmatrix} 0 & 0 & 0.26 & 0.17 & 0.17 & 0.22 \\ 0 & 0 & 0.03 & 0.02 & 0.02 & -0.08 \\ 0 & 0 & -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 0 & 0 & -1 \end{bmatrix}$	$\mathbf{0}_{5\times 5}$	$\begin{bmatrix} 0 & 0 & 0.36 & 0.42 & 0.09 & 0.36 \\ 0 & 0 & -0.18 & 0.18 & 0.18 & -0.36 \\ 0 & 0 & -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 0 & 0 & -1 \end{bmatrix}$	$\mathbf{0}_{5\times 5}$
LNM	$\begin{bmatrix} 0 & 0 & 2.09 & 1.86 \\ 0 & 0 & -0.14 & 0.24 \\ 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & -1 \end{bmatrix}$	$\begin{bmatrix} 0 & 0 & 2.52 & 1.67 \\ 0 & 0 & -0.71 & 0.83 \\ 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & -1 \end{bmatrix}$	$\mathbf{0}_{4\times 4}$	$\mathbf{0}_{4\times 4}$
NBM	$\begin{bmatrix} 0 & 0 & 0.26 & 0.17 & 0.17 & 0.22 & 2.09 & 1.86 \\ 0 & 0 & 0.03 & 0.02 & 0.02 & -0.08 & -0.14 & 0.24 \\ 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 \end{bmatrix}$	$\begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 2.52 & 1.67 \\ 0 & 0 & 0 & 0 & 0 & 0 & -0.71 & 0.83 \\ 0 & 0 & 0 & 0 & 0 & 0 & -0.08 & -0.17 \\ 0 & 0 & 0 & 0 & 0 & 0 & -0.38 & -0.12 \\ 0 & 0 & 0 & 0 & 0 & 0 & -0.17 & -0.08 \\ 0 & 0 & 0 & 0 & 0 & 0 & -0.08 & -0.17 \\ 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 \end{bmatrix}$	$\begin{bmatrix} 0 & 0 & 0.36 & 0.42 & 0.09 & 0.36 & 0 & 0 \\ 0 & 0 & -0.18 & 0.18 & 0.18 & -0.36 & 0 & 0 \\ 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 \\ 0 & 0 & -0.06 & -0.05 & 0.04 & -0.08 & 0 & 0 \\ 0 & 0 & 0.06 & 0.11 & -0.04 & 0.08 & 0 & 0 \end{bmatrix}$	$\mathbf{0}_{8\times 8}$

Table 5.14: Design bias under model misspecification for the small-world network (ϕ_2)

True model	**Optimal designs**			
	RD	**RBD**	**LND**	**NBD**
LNM	$\begin{bmatrix} 0 & 0 & 2.09 & 1.86 \\ 0 & 0 & -0.14 & 0.24 \\ 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & -1 \end{bmatrix}$	$\begin{bmatrix} 0 & 0 & 2.52 & 1.67 \\ 0 & 0 & -0.71 & 0.83 \\ 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & -1 \end{bmatrix}$	$\mathbf{0}_{4\times 4}$	$\mathbf{0}_{4\times 4}$
NBM	$\begin{bmatrix} 0 & 0 & 0.26 & 0.17 & 0.17 & 0.22 & 2.09 & 1.86 \\ 0 & 0 & 0.03 & 0.02 & 0.02 & -0.08 & -0.14 & 0.24 \\ 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 \end{bmatrix}$	$\begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 2.52 & 1.67 \\ 0 & 0 & 0 & 0 & 0 & 0 & -0.71 & 0.83 \\ 0 & 0 & 0 & 0 & 0 & 0 & -0.08 & -0.17 \\ 0 & 0 & 0 & 0 & 0 & 0 & -0.38 & -0.12 \\ 0 & 0 & 0 & 0 & 0 & 0 & -0.17 & -0.08 \\ 0 & 0 & 0 & 0 & 0 & 0 & -0.08 & -0.17 \\ 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 \end{bmatrix}$	$\begin{bmatrix} 0 & 0 & 0.48 & 0.32 & 0.09 & 0.11 & 0 & 0 \\ 0 & 0 & 0.28 & -0.26 & -0.11 & -0.12 & 0 & 0 \\ 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & -0.09 & 0 & 0.01 & 0 & 0 \\ 0 & 0 & -0.14 & 0.05 & 0.06 & 0.05 & 0 & 0 \end{bmatrix}$	$\mathbf{0}_{8\times 8}$

other hand, the optimal treatment arrangements under ϕ_2 are unique. For this example, these treatment arrangements return different values under the block models RBM and NBM. The mean criterion values for LND under RBM and NBM are respectively 0.2371 (that is lower than 0.2791) and 0.3237 (that is lower than 0.4224). We also recomputed the boxplot of LNDs under NBM for ϕ_1, which is illustrated in Figure 5.19. The outliers are detected by setting the upper/lower ends of the whiskers at three standard deviations. The median efficiency for LND is 0.624, with minimum and maximum efficiencies 0.088 and 1 respectively.

Figure 5.19: Boxplots of efficiencies for ϕ_1 (under NBM) (accounting for all LNDs)



Moreover, the bias of the average of all LNDs under NBM is reformulated as

$$\begin{bmatrix} 0 & 0 & -0.12 & 0.16 & 0.43 & 0.21 & 0 & 0 \\ 0 & 0 & 0.01 & 0.02 & 0.03 & -0.09 & 0 & 0 \\ 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 \\ 0 & 0 & 0.08 & 0.01 & -0.07 & -0.02 & 0 & 0 \\ 0 & 0 & 0.07 & -0.01 & -0.06 & 0.03 & 0 & 0 \end{bmatrix} \begin{pmatrix} \mu \\ \tau_1 \\ b_1 \\ b_2 \\ b_3 \\ b_4 \\ \gamma_1 \\ \gamma_2 \end{pmatrix}.$$

Suppose we perform an experiment where we wrongly assume at the design stage that there are no blocking effects. We can investigate how LND performs with respect to NBD for different clustering coefficient values in a number of simulated small-world networks. Figure 5.20 illustrates this idea. Recall that the clustering coefficient can take the extreme values of 0 and 1 for the random and complete network respectively. This explains the high efficiencies of LND for the boxplots around those values. However, for the 3rd, 4th and 5th boxplots where it is preferable to account for blocking (as implied from the high modularity scores in Figure 5.21) LND performs poorly. In particular, the median efficiencies under ϕ_1 range from 65% to 70% while under ϕ_2 they range from 50% to 65%. This descending pattern of efficiencies when ignoring the blocking effects, although there is a strong evidence of community structure underlying the connected units, is also verified for larger small-world networks. In particular, we obtain the relative efficiencies of the optimal LND compared to the optimal NBD when we assumed that the true treatment model is described by the NBM, for 100 simulated small-world networks of size $n = 100$ vertices. Figure 5.22 shows the results. We observe a slightly different behaviour for larger clustering coefficients with a significant decrease of the efficiencies. This is probably due to the fewer simulated networks that fall in that last category.

Figure 5.20: Boxplots of relative efficiencies (LND to NBD) for ϕ_1 (*left*) and ϕ_2 (*right*) for 1000 simulated SW networks (with $n = 22$) versus different clustering coefficients

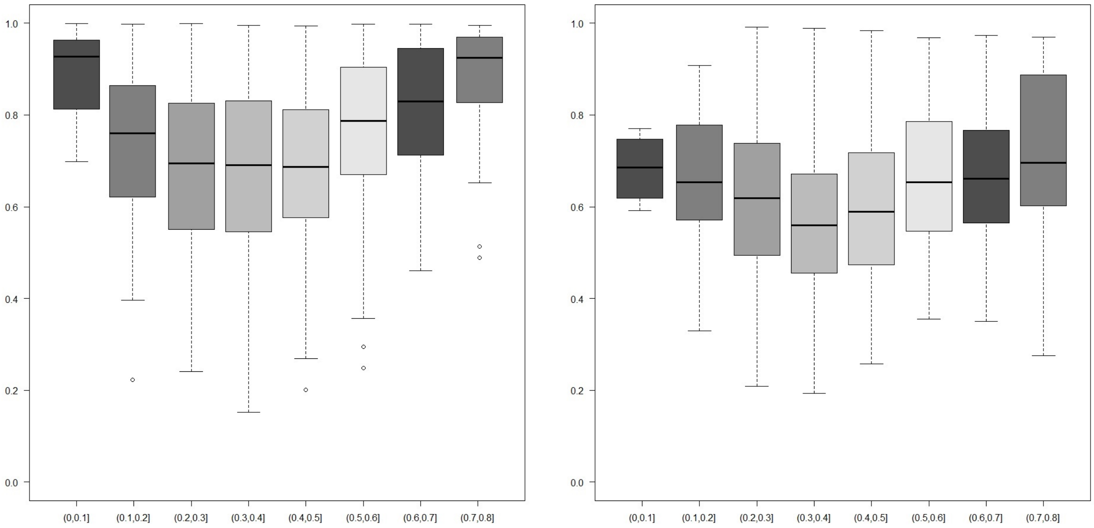


Figure 5.21: Boxplots of modularity scores corresponding to the 1000 simulated networks for the different clustering coefficients

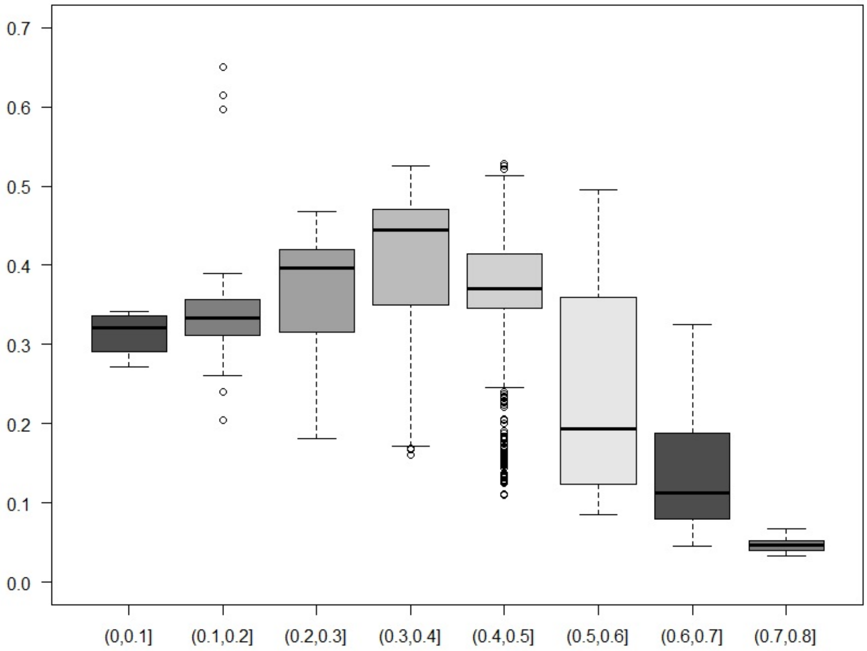
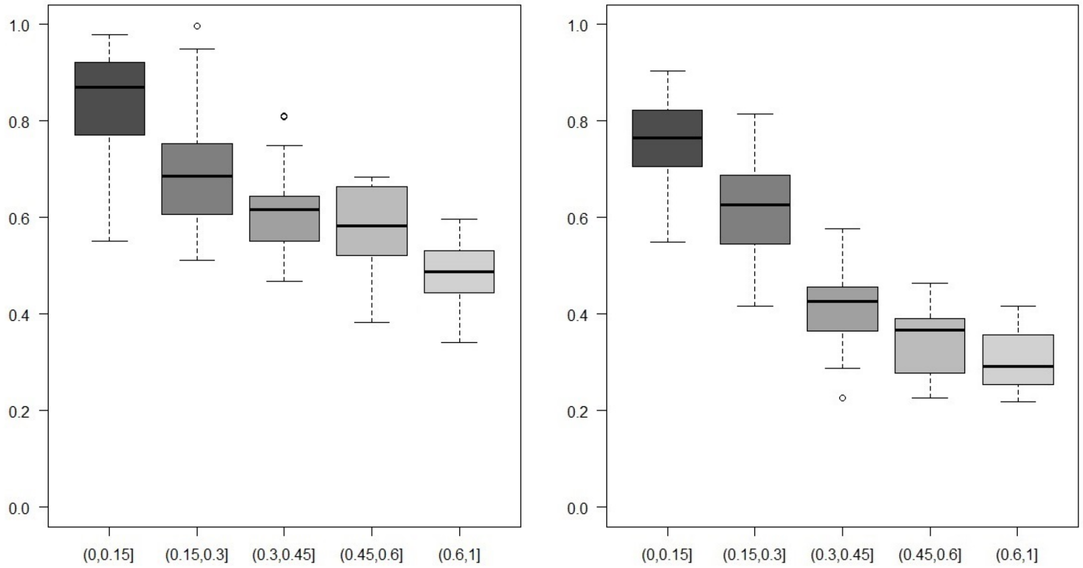


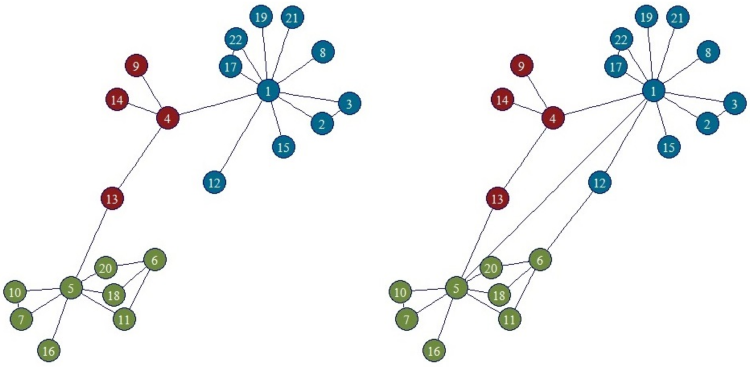
Figure 5.22: Boxplots of relative efficiencies (LND to NBD) for ϕ_1 (*left*) and ϕ_2 (*right*) for 100 simulated SW networks (with $n = 100$) versus different clustering coefficients



5.5 Misspecification of cross-blocking connections

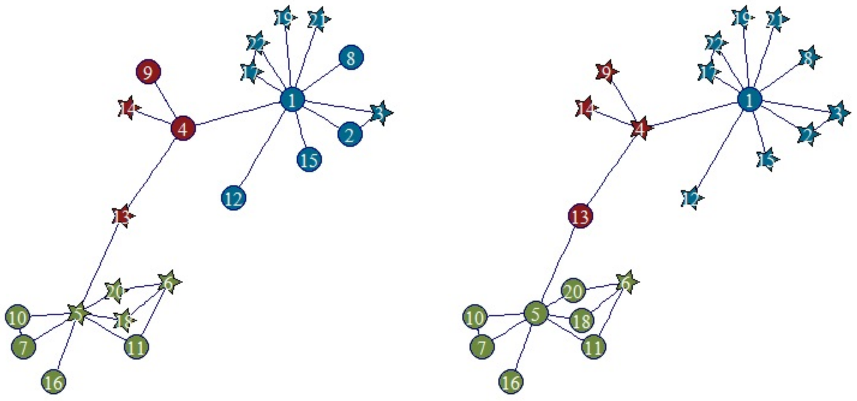
In this section, we focus on the case where we misspecify the network structure and the impact of such misspecification on the design efficiency (this subject matter was also discussed in Chapter 3 when misspecifying the connections of a network for an unblocked design). Let us consider two different cases of misspecification for the particular network of Figure 5.8(a) of Example 5.4.1 and obtain their corresponding efficiencies. The cases we consider are illustrated in Figure 5.23.

Figure 5.23: Removing (*left*) or adding (*right*) a connection



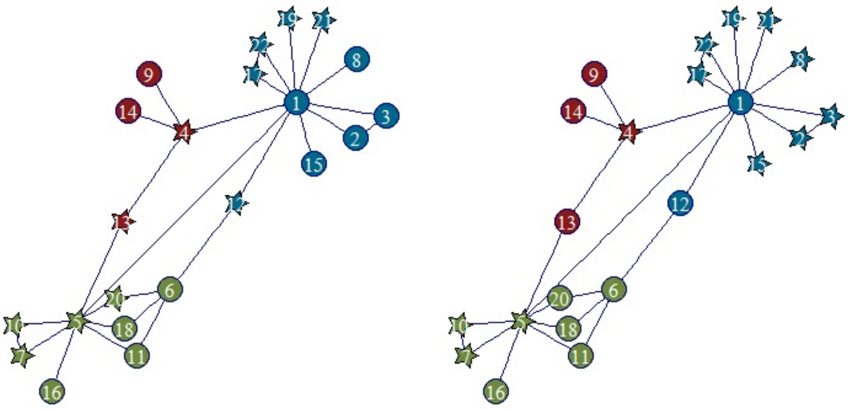
When we remove an edge connecting units 6 and 12, which are located in different blocks the design efficiencies are 0.99 and 0.92 for ϕ_1 and ϕ_2 respectively. The optimal treatment allocations are illustrated for this case in Figure 5.24. We are also interested in the network measures of the misspecified network; the average degree $\delta = 2.36$, the clustering coefficient $\mathcal{C} = 0.11$ and the average path length $\ell = 3.19$. By removing this edge the average path length increases by approximately 10%.

Figure 5.24: Optimal designs for ϕ_1 (*left*) and ϕ_2 (*right*), removing connection between units 6 and 12



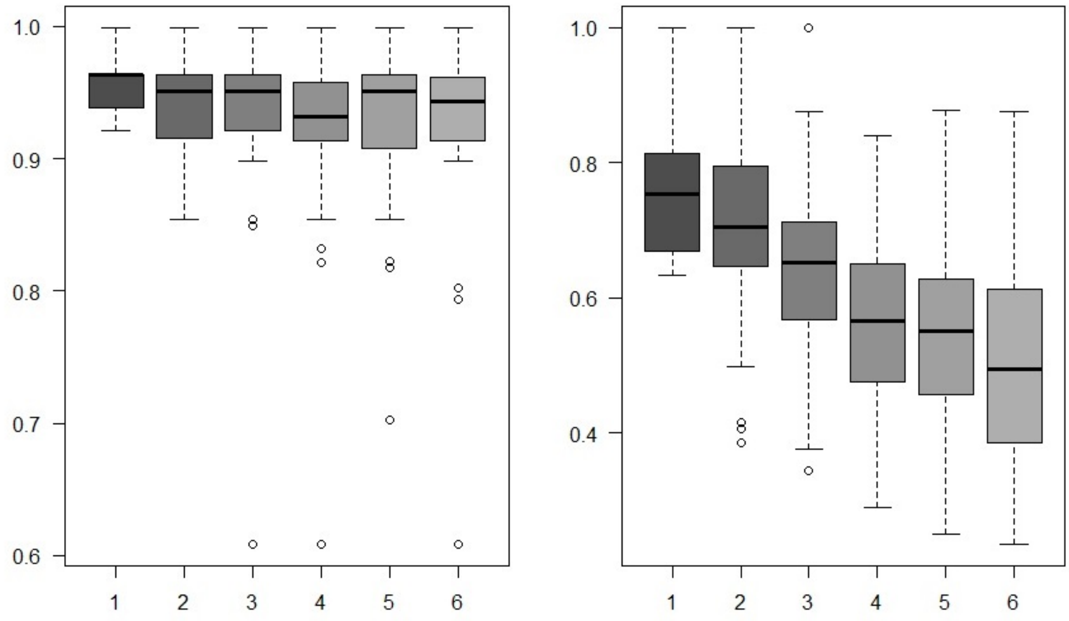
When we add an edge among the highly connected units 1 and 5, the design efficiencies are 0.99 and 0.99 for ϕ_1 and ϕ_2 respectively. The optimal treatment allocations are illustrated in Figure 5.25. We compute the network measures of this misspecified network; the average degree $\delta = 2.54$, the clustering coefficient $\mathcal{C} = 0.08$ and the average path length $\ell = 2.42$.

Figure 5.25: Optimal designs for ϕ_1 (*left*) and ϕ_2 (*right*), adding connection between units 1 and 5



An approach for examining design robustness to network misspecification is by removing sequentially an increasing number of edges. This is done by randomly selecting and removing sets of a specified number of edges. In particular, we simulate 100 different networks misspecifying an increasing number of edges, removing from 1 edge up to 6 edges simultaneously. Then we calculate the efficiencies of the optimal designs for those misspecified networks when the true network is the one shown in Figure 5.8 (see Figure 5.26). For ϕ_1 the efficiencies relative to the optimal design are greater that 90% on average. However, for ϕ_2 the efficiencies decrease substantially when the network is misspecified, especially for large number of connections. In reality we should be cautious when dealing with relatively small networks and large degree of network misspecification, since the network can collapse after the removal of a large number of edges.

Figure 5.26: Network misspecification; removing 1 to 6 edges, 100 simulated networks. Boxplots of efficiencies for ϕ_1 (*left*) and ϕ_2 (*right*)



5.6 Robustness due to misspecified blocking

In this section, we revisit Example 5.4.1 and we investigate the design robustness to the different choice of blocking for two types of misspecification: (I) number of blocks and (II) number of units within blocks. Recall that designs for which the loss of information is 'small' may be considered as robust. We consider two different scenarios: we design the experiment with NBM, and we either (a) carry out the analysis with the same model or (b) switch to RBM, which ignores network effects. We investigate the variability and bias introduced to treatment effects due to the different misspecifications.

We choose different values for the parameters in order to examine a number of different possible cases that can be true in practice. We generally expect the network effects to be less than or equal to the direct treatment effects, e.g. $\tau_1 \geq \gamma_1$. The different cases we investigated occur when $\tau_{r(ij)} \geq \gamma_{r(ij)} \geq b_i$ or $\tau_{r(ij)} \geq b_i \geq \gamma_{r(ij)}$ for $j = 1, 2, \ldots, n$ and $i = 1, 2, \ldots, \kappa$, including some zeros to differentiate inactive effects from the active effects. Note that the size of the block effects is not related to the size of the treatment or the network effects. For example, there could be no treatment or network effects but large block effects. However, we introduce these restrictions to consider only a subset, since the results will be similar. The various sets of fixed parameters $\boldsymbol{\beta}$ we have considered correspond to the seventeen different rows of Table 5.15. To put these values into context, consider the advertising example of Section 1.1. Let the network of Figure 5.27 (*left*) be the social network under experimentation, where the different blocks correspond to three different cliques of friends. If subject 1 is given advert 1 on purchasing a product, it will have an effect τ_1 and an individual network effect γ_1 on all connected friends (including subject 4, which belongs to a different block), to which subject 1 may show the advert causing them potentially to buy larger quantities of the product. Separately to those individual network effects and irrespective of the advertisements applied, subject 1 tends to buy a similar amount of product to subjects belonging to the same clique due to the block effect b_1 ('blue' block). The different parameter values may indicate that the block effect is stronger or weaker than the network effect.

For our simulation study we assumed the errors follow a $N(0, 1)$ distribution. Note that μ was set equal to 1. These cases have been chosen quite arbitrarily in an attempt to include possible experimental conditions. For example, the block effects in some cases are larger or smaller than the network effects, indicating that the blocks play a major or minor role in the formation of the responses respectively. For each case we generated 3000 datasets. For the first case we assume that all effects are equal to 1. This special case enables us to quantify the increase of the variance when neglecting the effect of the third block. Bear in mind that due to the small size of clusters and the network as a whole, we would expect any misspecification to cause a significant change to both the design and analysis of the experiment compared to a misspecification occurring in a larger-sized network.

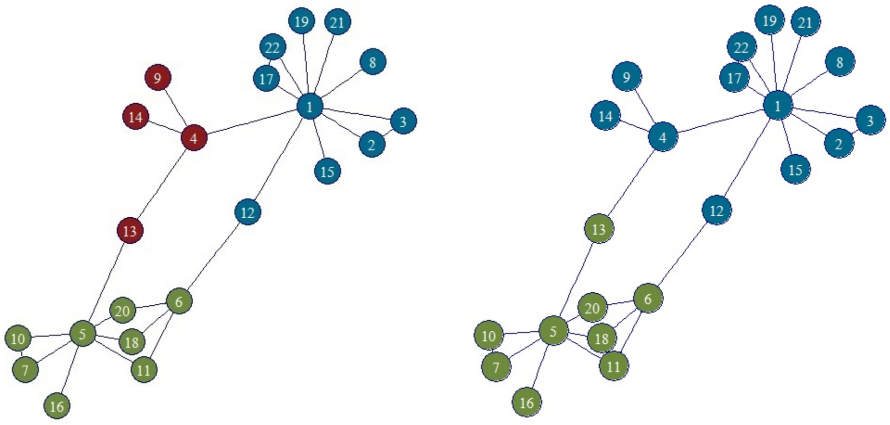
Table 5.15: Sets of fixed parameters $\boldsymbol{\beta}$

No.	μ	τ	b_1	b_2	γ_1	γ_2
1	1	1	1	1	1	1
2	1	3	0	1	1	1
3	1	3	1	0	1	1
4	1	3	1	1	0	1
5	1	3	1	1	1	0
6	1	3	1	2	1	2
7	1	3	-1	0	-3	2
8	1	3	1	2	3	-3
9	1	3	1	1	0	0
10	1	10	6	3	8	9
11	1	10	-5	1	6	1
12	1	10	8	5	2	2
13	1	10	3	6	5	2
14	1	10	3	2	10	8
15	1	10	0	-3	-4	5
16	1	10	6	1	-9	10
17	1	10	3	3	2	0

I. Misspecifying the number of blocks

We begin by investigating the design robustness when varying the number of blocks. The actual blocking structure is illustrated in Figure 5.27 (*left*), with blue corresponding to the first block, red to the second and green to the third respectively. If we design the experiment according to a different blocking structure, e.g. Figure 5.27 (*right*), then the between-block variance will remain almost the same. However, the within-block variance will be higher than before, since in the case of two blocks part of the between-block variation will be contained in the residual variation, instead of block effects. We investigate through Monte Carlo simulations the design robustness if the assumed model is wrong (adopting the wrong blocking structure) and compare it in terms of the variances and biases to the design under the true model (adopting the true blocking structure). Thus if the true within-block variance is hugely inflated then the impact of misspecifying the number of blocks will be large.

Figure 5.27: Blocking misspecification; true (*left*), test (*right*)



The optimal designs for ϕ_1 are provided in Figure 5.28. The individual variances of the simulated treatment effect estimates under the correct (with three blocks) and wrong (with two blocks) design are presented in Table 5.16. This table also shows the case when we analyse the experiment assuming a wrong (with two blocks) randomised balanced design, without having network effects.

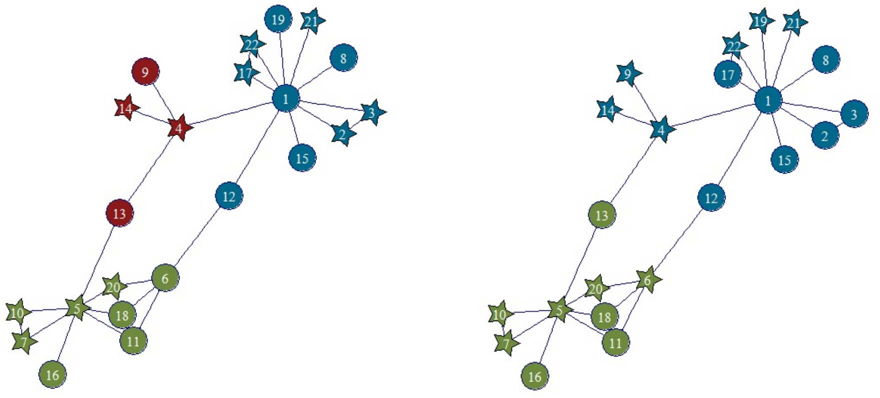
The variance increase for the 'test' blocking structure (second column) is consistently higher in Table 5.16 for both test(a) and test(b) compared to the 'true' blocking structure (first column). From Table 5.16, we can calculate the average of the variances. The variance increases are $\mathbb{E}\left[var(\hat{\tau}_{true})\right]/\mathbb{E}\left[var(\hat{\tau}_{test})\right] = 0.1814/0.2395 = 0.76$ and $\mathbb{E}\left[var(\hat{\tau}_{true})\right]/\mathbb{E}\left[var(\hat{\tau}_{test})\right] = 0.1805/0.2099 = 0.86$ on average for test(a) and test(b) respectively. The larger variance increase can mainly be attributed to the small size of the network and/or blocks, or to the absence of the network effects. Looking at the ratios between the two cases (ratio of test variance to true variance of each row) of the individual variances, we can see that they do not vary much. Moreover, compared to the magnitude of the variances the biases are negligible. The main effect of fitting the wrong model is the variance increase, whereas the biases are less important. However, for the second scenario the bias is larger. This happens because the largest part of the bias results from ignoring network effects. Ignoring block effects, just like in the standard cases inflates the variance, but it does not introduce bias. Therefore, block and network effects play different roles, such that the experimenter cannot use the one to compensate for the other, i.e. they are not interchangeable.

Table 5.16: Analysis with (a) NBD or (b) RBD (3000 simulations, type (I) of misspecification)

	true	test(a)		test(b)	
No.	$var(\hat{\tau}_{true})$	$var(\hat{\tau}_{test})$	$\widehat{\text{Bias}}$	$var(\hat{\tau}_{test})$	$\widehat{\text{Bias}}$
1	0.1862	0.2356	-0.0009	0.2075	-0.0016
2	0.1857	0.2369	0.0038	0.2067	0.0021
3	0.1839	0.2432	-0.0020	0.2123	-0.0008
4	0.1801	0.2340	0.0090	0.2069	0.0067
5	0.1824	0.2398	-0.0007	0.2108	-0.0029
6	0.1817	0.2383	-0.0167	0.2107	-0.0141
7	0.1815	0.2381	0.0143	0.2049	0.0124
8	0.1723	0.2271	0.0112	0.1984	0.0041
9	0.1801	0.2467	0.0026	0.2222	0.0044
10	0.1780	0.2381	0.0141	0.2136	0.0124
11	0.1685	0.2427	0.0071	0.2087	0.0060
12	0.1836	0.2486	-0.0114	0.2169	-0.0114
13	0.1820	0.2360	-0.0105	0.2099	-0.0106
14	0.1782	0.2453	0.0167	0.2129	0.0093
15	0.1843	0.2408	0.0161	0.2097	0.0146
16	0.1884	0.2446	-0.0121	0.2110	-0.0121
17	0.1865	0.2364	-0.0103	0.2055	-0.0059

We conclude that when misspecifying the number of blocks the variance increase is approximately of the same order of magnitude, indicating that the problem of block misspecification is not network specific. The mean variances across the simulations of these estimates support this conclusion. Thus in practice, misspecifying the blocks makes the properties of a good design worse but up to approximately the same extent as the standard designs without network effects. Additionally, the results also show that we cannot fix the situation with post blocking. Thus, we need to know the correct block structure at the time we design the experiment, as we cannot rectify it at a later stage by analysing it by using blocks indirectly.

Figure 5.28: Optimal designs for ϕ_1; true (*left*), test (*right*)



II. Misspecifying units within blocks

Here, we assume that we have three blocks, but subject 13 is misspecified to be in the 'green' block instead of the 'red' as shown in Figure 5.29 (*right*). Recall that the implemented clustering algorithm, which was defined on the basis of graph structure, ensures that clusters are not too small. By implementing a different community detection method, this intermediate subject, which is loosely connected to the two clusters, is likely to be clustered in a different way, which justifies our choice of investigating that particular subject. Using the same set of parameters $\boldsymbol{\beta}$ as before, we investigate the two previously discussed scenarios. In particular we design the experiment wrongly, having misclassified the subject labelled 13 and analysing it either correctly or wrongly by assuming the NBM or RBM respectively. Our results after running 3000 simulations are presented in Table 5.17. In addition, the optimal designs for ϕ_1 are provided in Figure 5.30.

Figure 5.29: Misspecification of unit labelled 13 belonging in a different block: true (*left*), test (*right*)

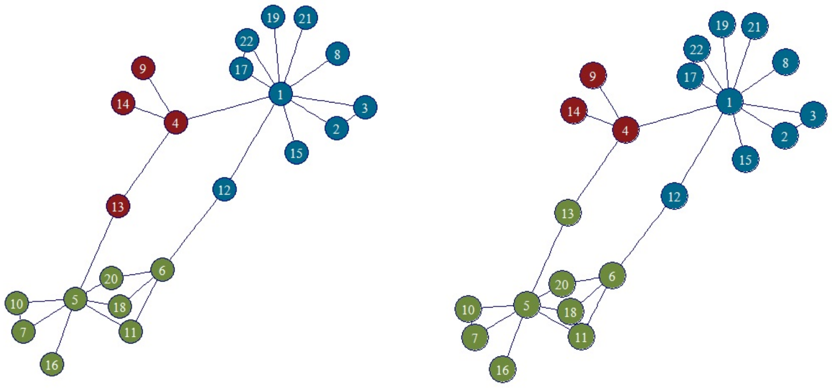
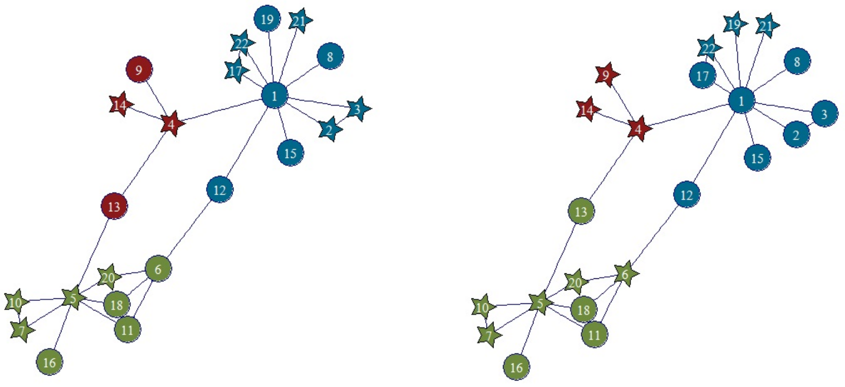


Figure 5.30: Optimal designs for ϕ_1; true (*left*), test (*right*)



Focusing on the cases where the effect of the second block (i.e. 'red' block) is zero (rows No. 3 and 7), we can observe that the variance increase is relatively small for both cases (assuming either NBM or RBM). Thus this type of misspecification leads to an increase of the variances but up to a similar order of magnitude under both scenarios, when analysing the experiment by either accounting for or ignoring network

Table 5.17: Analysis with (a) NBD or (b) RBD (3000 simulations, type (II) of misspecification)

	true	test(a)		test(b)	
No.	$var(\hat{\tau}_{true})$	$var(\hat{\tau}_{test})$	$\widehat{\text{Bias}}$	$var(\hat{\tau}_{test})$	$\widehat{\text{Bias}}$
1	0.1862	0.1992	-0.0035	0.1991	-0.0034
2	0.1857	0.2023	-0.0111	0.2022	-0.0109
3	0.1839	0.1980	-0.0094	0.1980	-0.0092
4	0.1801	0.1982	-0.0059	0.1979	-0.0063
5	0.1824	0.1859	-0.0110	0.1855	-0.0112
6	0.1817	0.1961	-0.0106	0.1955	-0.0105
7	0.1815	0.1984	0.0016	0.1982	0.0018
8	0.1723	0.2017	0.0008	0.2015	0.0008
9	0.1801	0.1952	-0.0054	0.1952	-0.0054
10	0.1780	0.1981	0.0052	0.1978	0.0048
11	0.1685	0.2004	-0.0005	0.2001	-0.0003
12	0.1836	0.2009	0.0067	0.2004	0.0068
13	0.1820	0.1962	-0.0076	0.1961	-0.0075
14	0.1782	0.1958	0.0007	0.1959	0.0007
15	0.1843	0.2036	-0.0020	0.2033	-0.0020
16	0.1884	0.2021	0.0076	0.2019	0.0074
17	0.1865	0.1970	0.0098	0.1966	0.0096

effects, which implies that the problem of misspecifying units in different blocks is not network specific. From Table 5.17, we can calculate the average of the variances. The average variance increases are $\mathbb{E}\left[var(\hat{\tau}_{true})\right]/\mathbb{E}\left[var(\hat{\tau}_{test})\right] = 0.1814/0.1982 = 0.91$ and $\mathbb{E}\left[var(\hat{\tau}_{true})\right]/\mathbb{E}\left[var(\hat{\tau}_{test})\right] = 0.1805/0.1979 = 0.91$ for test(a) and test(b) respectively. Thus in practice, similarly to the misspecification of number of block, allocating a unit to a different block causes problems to the analysis but approximately to the same extent as of misspecifying units in standard designs without the inclusion of network effects. Moreover, the bias induced in the treatment effects resulting from this type of misspecification is larger than in the case of misspecifying the number of blocks, especially when we assume RBM to be true. This increase in bias results from not accounting for network effects, when they are actually important.

5.7 Discussion

To sum up, the design-based methodological framework proposed in the current chapter consists of three main components: (i) we use spectral clustering techniques of the given, or discovered, social network (its topology is expressed through the Laplacian matrix) to project the vertices of the network onto an eigenvector-space of changeable dimensionality (where κ is the number of first nontrivial eigenvectors of the L_{rw}). (ii) Subsequently, we choose the 'best' partition of the network over all produced partitions (given the dimension κ of eigenvector-space at each time), which reveals the densely connected group of vertices with only sparse connections between groups. This is obtained by using the quality function of modularity. In particular we choose that partition, over all possible partitions for every considered dimension κ, where the modularity takes its maximum value. (*iii*) Finally we find the near-optimal block designs, for that number of blocks, which will provide the optimally allocated treatments to the groups of subjects within the social network. The designs obtained will enable the researcher to make comparisons between the different (unstructured) treatments both directly and indirectly.

An issue of concern with the suggested methodology is that when misspecifying the number of blocks, the true blocks may not be partitions of the assumed blocks. The spectral clustering algorithm, implemented in this chapter, differs from other community detection algorithms to the point that the whole network is not recursively partitioned. The difference lies in the sense that other divisive approaches iteratively cut the edges, dividing the network progressively into smaller and smaller disconnected subnetworks identified as the communities. However, the clusters obtained using the spectral clustering algorithm are preferable because the process iterates independently for different numbers of clusters. For instance, for $\kappa = 2$ and $\kappa = 3$, we have two and three clusters respectively, where the three clusters are not sub-partitions of the two clusters in the sense that they may contain units from the two clusters of the previously produced two clusters. Moreover, given that the different blocking structures are produced by a stochastic clustering algorithm and the optimal block designs are obtained by means of a non-exhaustive search, the results initially presented a certain variability. To correct for that, we allowed 2000 multiple initial configurations for the k-means clustering step and 100 starting designs for the exchange algorithm in order to avoid getting stuck in local optima. The former action stabilises the partition obtained by enhancing the convergence of the k-means algorithm to the optimal solution (if there is no automatic convergence), while the latter action increases the chance of convergence to an efficient design for the given community structure. This also enhances the reproducibility of the results.

Some of the comparisons of optimal designs were obtained in an ego-network, part of Facebook. Ego-networks, either traditional or virtual ones, will become increasingly important for estimating network effects and network properties due to the simplicity of data collection, and the cost effectiveness compared to the collection of full network data.

In the vast majority of cases, using blocks does not incur a negative effect on the design. Non orthogonality in block effects inflates variance, but this was not an issue of concern for this thesis due to the relatively large-sized blocks we considered compared to the number of treatments which amounted to only two. Moreover, we show that misspecifying the blocks or units within blocks compromises the properties of a good design up to roughly the same extent as the randomised designs.

Chapter 6

Optimal row-column designs with network effects

This chapter makes a transition from experiments in social networks to agricultural experiments. The idea is to view the potential interference among plots as a network structure. Recall that the network structure is defined by means of the connectivity matrix, which can pertain to two main categories of specifications: capturing the spatial structure to reflect distances across space and controlling for farmer operations. To this end we consider a field trial running at Rothamsted Experimental Station (`http://www.rothamsted.ac.uk`) and we obtain different row-column designs accounting for the neighbour effects with different pre-specified network structures implementing a simple interchange algorithm. A comparison of various optimal designs under different models, including the commonly used designs in such situations, is provided.

6.1 Introduction

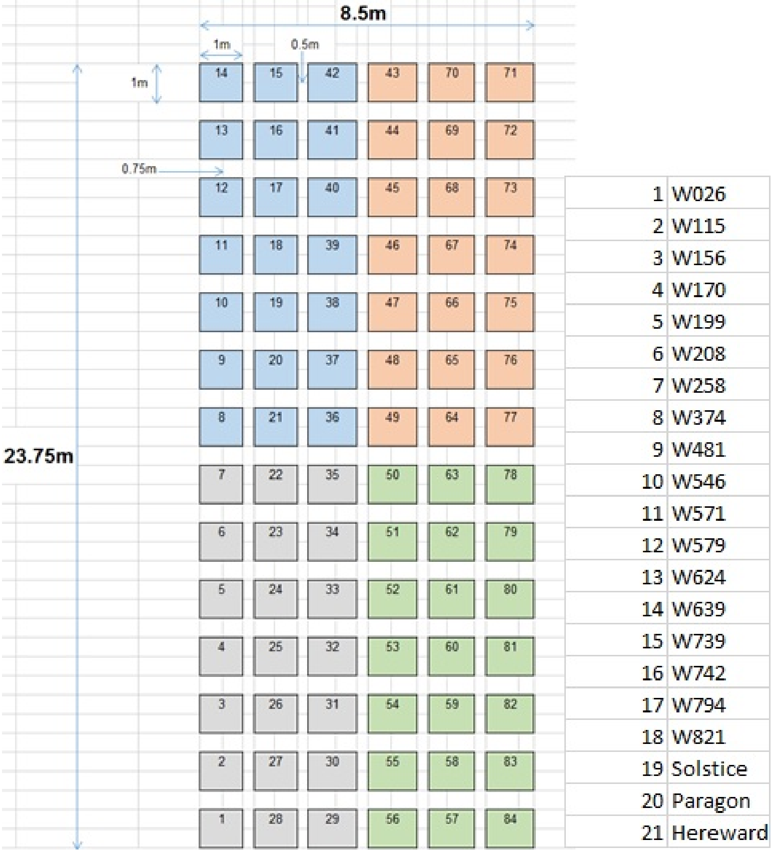
Row-column designs (RCDs) are commonly used in agricultural field experiments where the experimental units are arranged in two-dimensional space. A RCD may be required to allow for possible differences between both rows and columns and such a crossed block structure is likely to be much more efficient than the one-dimensional blocking structure for agricultural experiments (John and Williams, 1995, Ch.5-6). In this chapter we consider an agricultural experiment which aims to assess the natural cereal aphid colonisation of selected lines from the Watkins collection compared to three elite hexaploid wheat lines. There are two other factors associated with the farm operations which can influence the design structure related to the drilling and spraying of the field. Addressing these farm constraints and considering a few complete replicates of the treatments, a conventional design will be a two-dimensional arrangement of rows and columns nested within each superblock. The most common nested row-column designs in use are the resolvable row-column designs (R-RCD), where treatments are arranged in complete superblocks and occur in each one of them exactly once (John and Williams, 1995, Ch.4-6). A R-RCD is usually specified as $(m, \kappa, \kappa_1, \kappa_2)$, where

m denotes the number of treatments, κ the number of superblocks, each consisting of κ_1 rows and κ_2 columns (with each superblock containing a complete replicate of $m = \kappa_1\kappa_2$ treatments). Thus all treatments occur κ times, where κ is the number of superblocks and m divides n, because n/m is the number of plots per superblock.

Several authors have given resolvable row-column designs for comparing different treatments for example Bose (1947), Singh and Dey (1979), Ipinyomi and John (1985) and Bailey (1993). In general, R-RCD are preferred when dealing with a large number of treatments (e.g. a large number of varieties) and small number of replications. These designs allow for adjustment of field trend and constraints in two directions. Other such designs are α-designs (introduced by Patterson and Williams, 1976), which are resolvable block designs with respect to the column component (see, for instance, John and Eccleston, 1986, who gave a class of orthogonal row-column designs based on the α-designs). The software package CycDesigN (Whitaker *et al.*, 1997) is a practical tool for constructing efficient resolvable row-column designs and is frequently used in Rothamsted Experimental Station. A detailed discussion of R-RCD and methods of construction can be found in John and Williams (1995, Ch.6).

For the case of our reference example, the R-RCD will be $(m, \kappa, \kappa_1, \kappa_2) = (21, 4, 7, 3)$, where the 21 different treatments (wheat lines) will be allocated to $1\text{m} \times 1\text{m}$ (see Figure 6.1) plots. The blocking structures are related to the physical location (e.g. similar type of soil) and farmer operations; the drilling of long lines of plots (along the columns) and spraying which can accommodate six plots in a row (along the rows). The different colours denote the $4/(7 \times 3)$ crossed block structure as used in Rothamsted (where each treatment appears once in every superblock). Thus each replication is accommodated by a compact superblock consisting of a 7×3 array of plots: the use of a row-and-column arrangement within each superblock allows the variations of the aphid colonizations to be eliminated in two directions. Given that for the current experiment the complete superblocks are already defined, we call the design resolved rather than resolvable. By convention, the m treatments are specified in the form of $\{1, 2, \ldots, m\}$ like in the previous chapters. We then assign treatment labels to the m treatments such that there is a one-to-one correspondence between the numbers and the names as shown in Figure 6.1. The resulting design of this type is a resolved nested row-column design.

The adjacent plots are assumed to be more alike due to e.g. the similar type of soil and by accommodating the underlying neighbour structure among them we can effectively eliminate sources of variation which could influence the treatment comparisons. An efficient experimental design will ultimately maximise the separation of treatment information from field variation. In agricultural experiments, the response on a given plot is often affected by the treatments applied to the neighbouring plots (see e.g. Besag and Kempton, 1986). For instance, considering treatments such as fertiliser, irrigation, or pesticide applied to one plot may cause spillover effects to its adjacent plots (for more details see Chapter 1). The underlying structure can be accounted for by means of a network. Thus we aim to improve the accuracy of the experiment by controlling



14	15	42	43	70	71
13	16	41	44	69	72
12	17	40	45	68	73
11	18	39	46	67	74
10	19	38	47	66	75
9	20	37	48	65	76
8	21	36	49	64	77
7	22	35	50	63	78
6	23	34	51	62	79
5	24	33	52	61	80
4	25	32	53	60	81
3	26	31	54	59	82
2	27	30	55	58	83
1	28	29	56	57	84

1	W026
2	W115
3	W156
4	W170
5	W199
6	W208
7	W258
8	W374
9	W481
10	W546
11	W571
12	W579
13	W624
14	W639
15	W739
16	W742
17	W794
18	W821
19	Solstice
20	Paragon
21	Hereward

Figure 6.1: Field layout of the motivating agricultural experiment and treatments

for the network heterogeneity by adjusting for the interference between neighbouring treatments or between neighbours' responses. In this chapter, we focus on the former and we construct optimal resolved row-column designs accounting for the treatment interference between neighbouring units. Section 6.2 provides the model on which the optimal designs will be based, incorporating the network effects. The formulation of L-optimality for estimating the treatment effects' differences in this case is also discussed in this section. Consequently, Section 6.3 develops an interchange algorithm for the construction of optimal resolved row-column designs with network effects. The specification of the adjacency matrix is considered in Section 6.4. Several optimal designs are provided in Section 6.5 with a detailed comparison among them. Finally, Section 6.6 discusses relevant practical issues and extensions.

6.2 Designs with row-column network effects block model (RCNBM)

The determination of the optimal resolved row-column designs with network effects (R-RCND) will be based on an extension of the LNM (3.9). Note that we are designing the networked experiment with respect to fixed effects for blocks. There will be a general row-column set up and then we group rows into big rows and columns into big columns.

In other words, we have (big rows | rows)$*$(big columns | columns), where big rows and big columns correspond to set of rows and set of columns respectively. Let y_{ikgh} denote the expected response from the experimental unit in the g-th row and h-th column within the i-th big row and k-th big column. Thus, the quadruple (i, k, g, h) identifies the experimental units which correspond to the vertices $v \in \{1, \ldots, n\}$ and $s = r(ikgh)$ corresponds to the the set of treatments applied on these units with $s \in \{1, \ldots, m\}$. The model is

$$\begin{aligned} y_{ikgh} &= \mu + \tau_{r(ikgh)} + R_i + C_k + (RC)_{ik} + r_{ig} + c_{kh} + (rC)_{ikg} + (Rc)_{ikh} \\ &\quad + \sum_{i'=1}^{b_1} \sum_{k'=1}^{b_2} \sum_{g'=1}^{\kappa_1} \sum_{h'=1}^{\kappa_2} A_{\{ikgh,i'k'g'h'\}} \gamma_{r(i'k'g'h')} + \epsilon_{ikgh}, \qquad (6.1) \end{aligned}$$

$$\Rightarrow y_v = \mu + \tau_s + R_i + C_k + (RC)_{ik} + r_{ig} + c_{kh} + (rC)_{ikg} + (Rc)_{ikh} + \sum_{v' \neq v} A_{\{v,v'\}} \gamma_{s'} + \epsilon_v,$$

where $i = 1, 2, \ldots, b_1; k = 1, 2, \ldots, b_2; g = 1, 2, \ldots, \kappa_1; h = 1, 2, \ldots, \kappa_2$, μ denotes the overall mean effect, $s = r(ijkl)$ indicates the treatment applied to unit $v = (i, k, g, h)$, τ_s is the (direct) effect of treatment s applied to unit v, R_i and C_j denote the i-th and j-th big row and big column effects respectively, while r_{ig} and c_{kh} denote the row and column effects nested within the big rows and big columns respectively, $(RC)_{ik}$ denotes the interaction effects of big rows and big columns adjusting for big rows and big columns, $(rC)_{ikg}$ denotes the interaction effects of rows and the big columns adjusting for rows and big columns, and similarly $(Rc)_{ikh}$ denotes the interaction effects of columns and the big rows, adjusting for columns and big rows. Note that (big row)+(big column)+(big row$*$big column) correspond to the (complete) superblocks (where $\kappa = b_1 b_2$). Moreover, $A_{\{v,v'\}}$ is the adjacency (neighbour) matrix indicating the weighted connections between the units v and v' and $\gamma_{s'}$ is the network effect of the treatment $s' = r(i'k'g'h')$ applied to the connected unit $v' = (i', k', g', h')$ when there is connection between v and v' (neighbour effect) (see Section 6.4 for more details). Notice that $\sum_{v' \neq v}$ corresponds to the multiple summations $\sum_{i'k'g'h'} = \sum_{i'=1}^{b_1} \sum_{k'=1}^{b_2} \sum_{g'=1}^{\kappa_1} \sum_{h'=1}^{\kappa_2}$. The ϵ_{ikgh} are assumed to be independent random variables (i.i.d), each with $E(\epsilon_v) = 0$ and $E({\epsilon_v}^2) = \sigma^2$.

The model in matrix notation can be written as

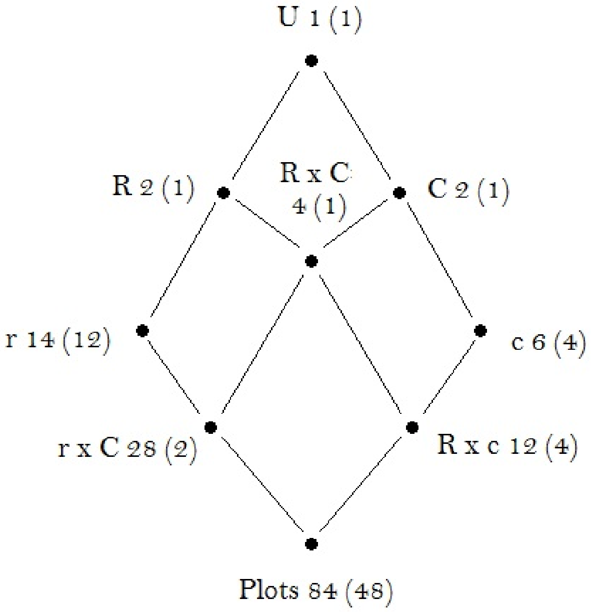
$$\mathbb{E}\left[\boldsymbol{y}\right] = \mu\mathbf{1} + X_\tau\boldsymbol{\tau} + X_R\boldsymbol{R} + X_C\boldsymbol{C} + X_{RC}(\boldsymbol{RC}) + X_r\boldsymbol{r} + X_c\boldsymbol{c} + X_{rC}(\boldsymbol{rC}) + X_{Rc}(\boldsymbol{Rc}) + AX_\gamma\boldsymbol{\gamma},$$

where $\boldsymbol{\tau}$, $\boldsymbol{R}$, $\boldsymbol{C}$, $(\boldsymbol{RC})$, $\boldsymbol{r}$, $\boldsymbol{c}$, $(\boldsymbol{rC})$, $(\boldsymbol{Rc})$ and $\boldsymbol{\gamma}$ denote the vector of treatment, big row, big column, (big row)$*$(big column) interaction, row, column, two crossed blocking interactions and network effects respectively.

Similarly to the previous chapters, the extended information matrix (including the column of ones corresponding to the constant) for RCNBM (6.1) is $M = X^T X$, where $X = (\mathbf{1} \quad X_\tau \quad X_R \quad X_C \quad X_{RC} \quad X_r \quad X_c \quad X_{rC} \quad X_{Rc} \quad AX_\gamma)$ with the corresponding incidence matrices for the treatments, big rows, big columns, (big row)$*$(big column) interactions, rows, columns, the two crossed blocking interactions and network effects respectively.

Returning to our motivating example, the entire experiment is broken down into big rows and big columns of lengths $b_1 = b_2 = 2$. The big rows by big columns combinations are equal to the $\kappa = 4$ superblocks, which are also broken down to $\kappa_1 = 14$ rows and $\kappa_2 = 6$ columns. From the crossing of the two combinations of rows and big columns and big rows and columns we obtain the $n = 84$ experimental units. Following Bailey (2008, Ch.10.4) and Goos and Gilmour (2012), we construct a Hasse diagram to visualise the structure in the units. The Hasse diagram in Figure 6.2 describes this experimental setup with the corresponding degrees of freedom in each stratum (not considering the network effects in this figure). Recall that according to the management of the trial, spraying and drilling processes are done row by row and column by column respectively. Note that the effects of these processes might be confounded with the positional effects (i.e. the row, column, big row and big column effects).

Figure 6.2: Hasse diagram of the model structure (without network effects)



The design will be obtained following the optimisation algorithm described in the next section, by generating an initial treatment arrangement with some particular properties (e.g. resolved row-column design) and by making random pairwise interchanges within the superblocks (restricting randomisation to retain the resolvability property). The optimality criterion employed for this optimisation problem is the L-optimality criterion (see Section 2.1), which minimises the average variance of the pairwise differences of treatment comparisons. In particular, we have

$$\phi = \sum_{v=2}^{m} \sum_{h=v+1}^{m+1} \boldsymbol{s}^T(v,h) M^{-1} \boldsymbol{s}(v,h), \tag{6.2}$$

where $\boldsymbol{s}(\alpha_1; \alpha_2)$ is a vector of zeroes of length $b_1 + b_2 + \kappa + \kappa_1 + \kappa_2 + b_1\kappa_2 + b_2\kappa_1 + 2m - 7$, except the α_1 and α_2 elements which are 1 and -1 respectively, following the removal of the $(m+1)$-st element (for more details refer to Section 5.3). This vector will be premultiplied and post-multiplied by the information matrix for obtaining the treatment effects' contrasts of which there are 210 in total in this experiment. Note at this point that the various effects of the RCNM (6.1) are assumed to satisfy some necessary standard constraints to make the treatment contrasts estimable.

6.3 Optimisation algorithm

Some early algorithmic attempts in search for optimal RCD were discussed in detail by Eccleston and Jones (1980) and Jones and Eccleston (1980). Their optimal designs were based on an evaluation of designs using the L-optimality criterion for minimising the sum of the weighted variances of a set of treatment contrasts of interest. The potential weights and the particular set of contrasts were specified by the experimenter to reflect his perception of the distances among plots. Moreover, a number of computer algorithms have also been suggested for obtaining RCD, for instance, the design generation package ALPHA+ for obtaining α-designs developed by Williams and Talbot (1993), described in detail by Nguyen and Williams (1993) and nested simulated annealing algorithm developed by John and Whitaker (1993). They all use some form of interchange procedure where pairs of treatments are interchanged in the design, subject to an iterative improvement procedure with respect to a chosen optimality criterion. To address the problem of the procedure getting stuck at a local optimum, Nguyen and Williams (1993) suggested repeated runs of the algorithm using different starting designs, and then choosing the best design over all runs. John and Whitaker (1993) also addressed this problem by accepting with low probability some randomly chosen interchanges that do not result in an improvement in the chosen optimality criterion.

For the objectives of the working example, we implement a simple interchange algorithm. We have a design with a crossed block structure with fixed number and sizes of blocks. Given that we restrict the design to be resolved, the swaps take place only within superblocks. The algorithm begins with the choice of a starting R-RCD. The L-optimality criterion introduced in the previous section determines the decision rule of either allowing the interchange to occur or leaving the design unchanged. In order to preserve resolvability the pairwise exchanges between units are made within the same superblock, accepting those interchanges that improve the criterion value for the overall design. The interchanges of pairs of treatments are chosen at random. An alternative approach would be to identify the pair of treatments that lead to the biggest improvement of the optimal function value. Thus the algorithm focuses on each superblock in turn, keeping the treatment allocation fixed in the remaining superblocks. It visits each superblock until a pass yields no changes to the design. The optimisation of the design is based on the network model of Equation (6.1) for minimising the variance of treatment comparisons by using interchanges rather than exchanges. The steps in the algorithm are as follows:

Simple interchange algorithm (SIA)

- Step 0: Enter the number of replicates, rows, columns, big rows, big columns, treatments and the adjacency matrix.
- Step 1: Generate a random (resolved if exists) row-column design and calculate the optimality criterion $\phi^{(0)}$ for the starting design.
- Step 2: Make a pairwise interchange of treatments within the current superblock keeping the arrangement of the treatment combinations fixed for the remaining superblocks. Calculate the optimality criterion $\phi^{(i)}$ for the current design that corresponds to that specific interchange (in run i). If an interchange improves the criterion value of the overall design, accept it and continue; otherwise, undo the interchange and continue.
- Step 3: Repeat Step 2 until no further interchanges in the current design result in an improvement (or if the above holds for at least a large number of iterations) then move on to the next superblock.
- Step 4: Repeat Steps 2 and 3 until a pass through all superblocks yields no changes or for a specific number of times.
- Step 5: Repeat Steps 1–4 for several randomly generated initial designs.

For Step 2 we choose a unit following the ascending order of units and swap its treatment to that of another randomly chosen unit belonging to the same superblock. Thus the candidate treatment swaps are made at random subject to constraints such as resolvability, where interchanges are restricted to be within the superblocks. Step 5 is required, as different starting designs may lead to different local optima. The final selected design is the best from all the trials. Moreover, if interchanges between units result in a singular information matrix it will be rejected. In Section 6.5, we provide different optimal designs adjusting this algorithm appropriately. For instance, we can relax the constraint of having a resolved design with the treatment interchanges occurring between cross-blocked plots nested within replicates, and let them occur across the whole field (non-resolved but equally replicated), or the algorithm allows for exchanges rather than interchanges allowing for non-equireplicate designs. For the exchanges the algorithm moves along all the units and exchanges a treatment with one of the competitive ones at random if this results in an improvement of the criterion (similar to the systematic exchange algorithm presented in Section 3.6).

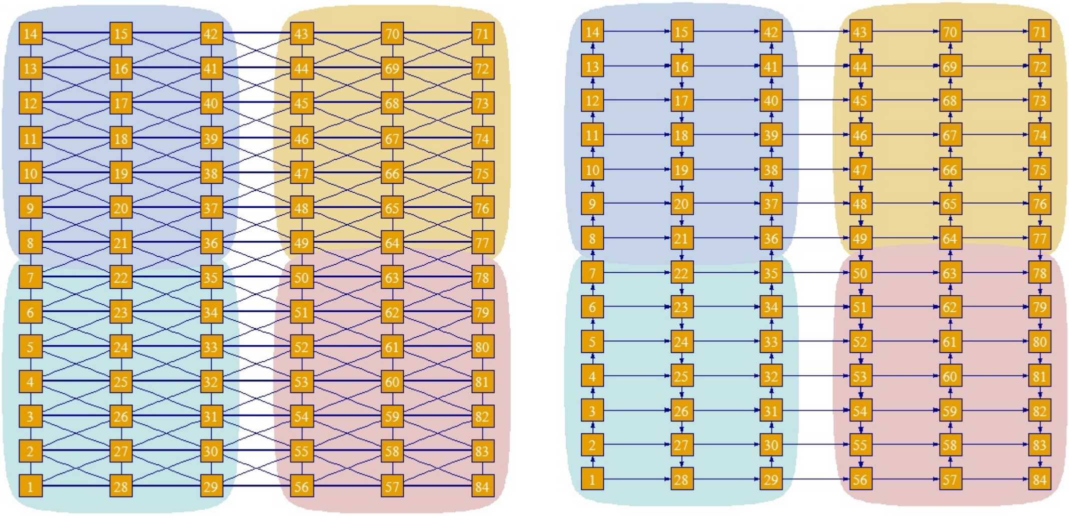
6.4 Adjacency matrix: practical issues and extensions

The aim of the experiment is to compare the number of selected wheat lines on the basis of their responses to natural cereal aphid colonisation. The n experimental units correspond to plots (pieces of land), which form a network structure pre-specified by the experimenter, so as to address the particular needs of the field trial. Different causes of association among neighbouring units can result in different specification of the adjacency matrix and appropriate models. Examples can be related to a non-directional or directional neighbouring matrix (e.g. considering the shade or wind), unweighted or weighted (e.g. capturing geographical distances or importance of upward neighbours, steep gradients in the spatial surface etc.). When dealing with spatial data there are several methods to specify the neighbour structure. For example, polygons that form square or rectangular lattices can be used to construct the spatial adjacency. We should note that we do not have spatial contiguity among the plots but they are rather separated with small spatial distances. These adjoining boundaries define the spatial relationships among the square plots. The adjacency-based spatial relationships can correspond to very different ranges of spatial similarity and the verification of the methods to construct the spatial adjacency is related to the threshold distance and number of neighbours. These methods, however, mainly depend on the rules used to define the notions of spatial neighbour: neighbourhood distance-based or border-based sharing common boundaries as neighbours. There is a considerable amount of literature on this issue, with many proposed methods for defining the spatial weight matrix (see, for instance, Cliff and Ord, 1981 and Anselin, 1988). For the current experiment the plots constitute the vertices of the network instead, for instance, of plot centroids (i.e. focusing on their geometric centre). Different cases could include an appropriate definition of geometrical contiguity among polygonal plots based on some distance rule (e.g. Euclidean distance). It is important to note that the adjacency matrix can appropriately be defined for capturing the spatial patterns even of an irregular arrangement of plots.

Our primary tool for describing the relationships among the plots in this chapter is an appropriately chosen adjacency matrix, where its edges are defined following an approach proposed by Cliff and Ord (1981). To understand our first choice for the adjacency matrix, one can think of the allowable moves of chess pieces and particularly king's moves. Recall that the king moves one space vertically, diagonally or horizontally. The second choice for the adjacency matrix relates to the farmer operations. More specifically, for our first specification the adjacency matrix is weighted, with weights based on the inverse spatial distance between plots including plots that are very close diagonally to one another (A_1), while for the second specification the adjacency matrix is constructed without imposing any weights and it is related to the farm operations and specifically the directions of drilling and spraying methods implemented by the farmer (A_2). These adjacency specifications can be used for controlling for unwanted geographical differences in the site and simplification of the farm operations. Figure 6.3 shows the two specifications of predetermined graph structures. Loosely speaking and for brevity we refer to these adjacency specifications as King's case (for $\mathcal{G}_1$) and Farmer's case (for $\mathcal{G}_2$) respectively.

These adjacency matrices have been specifically chosen to explore two common situations that can occur in this specific agricultural trial, serving the objectives of the experiment and the particular interests of the experimenter. We might expect the

Figure 6.3: Different connectivity graphs: $\mathcal{G}_1$ (*left*) and $\mathcal{G}_2$ (*right*)



spillover effects to stem from the adjacent neighbouring plots or from the particular farm operations. This led us to define a weighted and undirected network, with the weights being the reciprocals of the distances between plots, and a directed but unweighted network, where the directions indicate the order of the particular spraying and drilling operations that are expected to happen. The following section explores the optimality of the designs for the two alternative scenarios of treatment interference among neighbouring plots.

6.5 Comparison of optimal designs

In this section we provide comparisons of optimal designs for estimating the treatment effects under different models for the two different pre-specified adjacency matrices. We consider resolved and non-resolved designs allowing for unequal and equal replication. This will allow us to measure the efficiency loss by imposing additional restrictions on the randomisation process so as to quantify more accurately the efficiency of choosing for instance a resolved RCNBD with equal replication compared to that of RCND, which is non-resolved and without equal replication. In doing so we obtain the optimality function values of the designs with respect to the different models as

$$\text{CRM: } y_j = \mu + \tau_{r(j)} + \epsilon_j,\quad (j = 1, 2, \ldots, 84)$$

$$\text{RBM: } y_{ikj} = \mu + \tau_{r(ikj)} + R_k + C_l + (RC)_{ik} + \epsilon_{ikj},\quad (i = 1, 2; k = 1, 2; j = 1, 2, \ldots, 21)$$

$$\text{RCM: } y_{ik} = \mu + \tau_{r(ik)} + r_i + c_k + \epsilon_{ik},\quad (i = 1, 2, \ldots, 14; k = 1, 2, \ldots, 6)$$

$$\text{RCBM: } y_{ikgh} = \mu + \tau_{r(ikgh)} + R_i + C_k + (RC)_{ik} + r_{ig} + c_{kh} + (rC)_{kig} + (Rc)_{ikh} + \epsilon_{ikgh},$$
$$(i = 1,2; k = 1,2; g = 1,2,\ldots,7; h = 1,2,3)$$

$$\text{LNM: } y_j = \mu + \tau_{r(j)} + \sum_{j'=1}^{84} A_{jj'}\gamma_{r(j')} + \epsilon_j,$$
$$(j = 1,2,\ldots,84)$$

$$\text{NBM: } y_{ikj} = \mu + \tau_{r(ikj)} + R_i + C_k + (RC)_{ik} + \sum_{i'=1}^{2}\sum_{k'=1}^{2}\sum_{j'=1}^{21} A_{\{ikj,i'k'j'\}}\gamma_{r(i'k'j')} + \epsilon_{ikj},$$
$$(i = 1,2; k = 1,2; j = 1,2,\ldots,21)$$

$$\text{RCNM: } y_{ik} = \mu + \tau_{r(ik)} + r_i + c_k + \sum_{i'=1}^{14}\sum_{k'=1}^{6} A_{\{ik,i'k'\}}\gamma_{r(i'k')} + \epsilon_{ik},$$
$$(i = 1,2,\ldots,14; k = 1,2,\ldots,6)$$

$$\text{RCNBM: } y_{ikgh} = \mu + \tau_{r(ikgh)} + R_i + C_k + (RC)_{ik} + r_{ig} + c_{kh} + (rC)_{ijk} + (Rc)_{ijl}$$
$$+ \sum_{i'=1}^{2}\sum_{k'=1}^{2}\sum_{g'=1}^{7}\sum_{h'=1}^{3} A_{\{ikgh,i'k'g'h'\}}\gamma_{r(i'k'g'h')} + \epsilon_{r(ikgh)},$$
$$(i = 1,2; k = 1,2; g = 1,2,\ldots,7; h = 1,2,3)\,.$$

These models are functions of the treatment factors plus the error terms. We assume that, in all cases, the errors are independent and random with zero mean and constant variance. Similarly to comparisons of previous chapters, our interest lies in the comparisons of designs under the same model, making σ^2 redundant as it is the same for all proposed designs under the same experiment. Designs for these models are labelled CRD, RBD, CRD, RCBD, LND, NBD, RCND, RCBD.

We follow three steps in constructing the optimal designs under the above models. In the fist step we relax the restriction of having a resolved design and allow for unequal replication of the treatments. In the next step we allow for non-resolvability but restrict to equal replication. In the final step we restrict the optimal designs to be resolved and equally replicated. Table 6.1 provides the full list of candidate designs' labels. We investigate the benefits of imposing equal replication and resolvability. As will be seen later, the arguments against equal replication and resolvability are weaker if we include network effects, in the sense that we are not losing much in efficiency by imposing further restrictions on the optimisation process. This is an important observation given that algorithmically it is better to impose more restrictions since we reduce the design space leading to a faster convergence to an efficient design. We should also note that the algorithm presented in Section 6.3 has been adjusted appropriately for producing each class of designs imposing additional restrictions. For the unrestricted case the algorithm is an interchange-exchange algorithm running two nested computations sequentially: allowing for a large number of exchanges of the treatments with the competitive ones on the ordered units and then interchanging the treatments until we reach convergence. Note that the optimal designs for the first step under CRM, RBM, RCM and RCBM always have equal replication. The detailed optimal designs can be found in Appendix C. Note also that we always assume that competing block designs have the same superblock partition. A key observation from our comparisons below is that designs that account for network effects perform consistently better than the standard designs. Thus if we believe that there exists an underlying network structure then it is highly important to incorporate the network effects which capture that structure in our model.

Table 6.1: Candidate classes of designs—King's case

Class 1 **Unrestricted**	**Class 2** **Equal-replication**	**Class 3** **Resolvability**
CRD		
RBD		
RCD		
RCBD1		RCBD3
LND1	LND2	
NBD1	NBD2	NBD3
RCND1	RCND2	
RCNBD1	RCNBD2	RCNBD3

Adjacency matrix - King's case ($\mathcal{G}_1$)

We first look at the case where the adjacency matrix reflects distances across space. Thus it has weights corresponding to the reciprocal of the distances between the plots (accounting also for the edges at the corners). The optimality function values for all the designs obtained are shown in Tables 6.2–6.4. The optimality criterion is ϕ as presented in Equation (6.2). Recall that the smaller the criterion function value the better the design is.

Table 6.2 shows the optimal designs for the unrestricted case, when we do not impose either equal replication or resolvability. The optimal designs under the standard models have been chosen arbitrarily and as we can see they are much worse compared to the models that account for the network effects. More specifically, focusing on the last row where the true model is assumed to be the RCNBM, we can see that all randomised designs perform poorly with approximate relative efficiencies 35%, 40%, 44% and 45% for the optimal CRD, RBD, RCD and RCBD respectively. Moreover, when we account for the network effects, additionally to the block effects, the design performs slightly better than when ignoring them; efficiency increases to 51% compared to 48%. Accounting additionally for the row and column effects the design is 53% efficient. Thus we can see that with respect to this criterion all designs perform poorly under RCNBM, which means that if we strongly believe that these effects are present, we should account for them in the model. Another interesting observation from this table is that if we do not believe that there are block effects, by including them the efficiency decreases by 46.7% (see ϕ_{RCNBD1} under RCNM).

As we can observe from the class of designs in Table 6.3, if we impose equal replication we lose slightly in terms of efficiency compared to not doing so. It is interesting to note that the differences in the efficiencies for designs that account for the network structure are relatively small. Forcing additionally resolvability we obtain the class of optimal designs illustrated in Table 6.4. For the case of the randomised design RCBD3, we do not do much worse by imposing resolvability. Another observation in this last table is that NBD performs well under the RCNBM with approximately 75% efficiency, implying that when accounting for the network and block effects, we generally obtain a good design. Another interesting observation is that the NBD1 performs almost as well as the LND1 (the same holds for NBD2 compared to LND2) under LNM indicating that we do not do much worse by including block effects in the network model in terms of the design efficiencies. This observation has been also made in Chapter 5.

Table 6.2: Comparisons of the designs under the different models-class 1

Models	**Optimal designs**							
	CRD	**RBD**	**RCD**	**RCBD1**	**LND1**	**NBD1**	**RCND1**	**RCNBD1**
CRM	**105.0000**	105.0000	105.0000	105.0000	105.6667	107.0000	107.6667	108.3333
RBM	111.1778	**105.0000**	108.0804	106.4626	110.3307	109.6485	111.6093	113.8296
RCM	145.6674	129.3118	**125.6508**	145.4067	143.8451	154.6404	136.3127	150.0122
RCBM	158.8660	141.8568	143.1771	**125.7446**	152.8750	153.2337	156.1715	139.7849
LNM	185.7967	217.8699	195.3904	197.8486	**123.8627**	128.2574	144.0940	139.8225
NBM	231.5575	234.6434	231.2569	229.1827	148.0148	**134.3943**	160.1571	151.2984
RCNM	627.6495	518.7476	644.0838	479.0471	404.2810	496.8882	**212.8154**	455.8680
RCNBM	608.5364	541.5796	494.7243	477.3583	453.5195	422.1597	405.4540	**215.5461**

Table 6.3: Comparisons of the designs under the different models-class 2

Models	**Optimal designs**			
	LND2	**NBD2**	**RCND2**	**RCNBD2**
CRM	105.0000	105.0000	105.0000	105.0000
RBM	107.0332	106.0848	108.3879	107.1786
RCM	144.4564	149.7779	137.7795	141.7552
RCBM	143.0036	151.2253	151.9807	133.6267
LNM	**129.0965**	131.1138	147.4536	148.1472
NBM	148.5842	**138.3800**	164.1907	162.1072
RCNM	392.3432	462.7354	**230.4923**	344.0345
RCNBM	408.2686	429.5068	436.9890	**240.8215**

Table 6.4: Comparisons of the designs under the different models-class 3

Models	**Optimal designs**		
	RCBD3	**NBD3**	**RCNBD3**
CRM	105.0000	105.0000	105.0000
RBM	105.0000	105.0000	105.0000
RCM	138.1757	145.2837	145.3975
RCBM	**126.0997**	142.4339	135.9004
LNM	208.3804	139.3841	165.9144
NBM	224.6607	**142.2910**	171.0041
RCNM	503.0577	466.4401	509.4884
RCNBM	526.0848	398.1695	**298.3764**

Adjacency matrix - Farmer's case ($\mathcal{G}_2$)

At this point, we focus on the adjacency matrix related to the farmer operations. The optimality function values for all the obtained designs are illustrated in Tables 6.6–6.7. We note that in the first class of candidate designs where we allow for non-resolvability and unequal replication, the obtained optimal designs are all equally replicated (including the ones accounting for network effects). The labels of the designs are presented in Table 6.5.

Table 6.5: Candidate classes of designs—Farmer's case

Class 1	**Class 2**
Unrestricted	**Resolvability and equal replication**
CRD	
RBD	
RCD	
RCBD	RCBD5
LND4	
NBD4	NBD5
RCND4	
RCNBD4	RCNBD5

We see that the results here follow similar patterns to the results in the King's case. One difference stemming from the different network specification is that the optimal function values of the standard designs are slightly better than before. Assuming, for instance, that RCNBM is true, the randomised designs have approximate relative efficiencies 51%, 64%, 57% and 68% for the CRD, RBD, RCD and RCBD respectively. Accounting additionally for the network effects, the optimal designs are 65%, 78% and 78% efficient for the LND4, NBD4 and RCND4 respectively, implying that accounting for the block effects is as good as accounting for the row and column effects. When we additionally restrict for resolvability we lose just 1% in efficiency, but the NBD5 compared to the RCNBD5 performs very similarly in terms of efficiency under all models.

Table 6.6: Comparisons of the designs under the different models-class 1

Models	**Optimal designs**							
	CRD	**RBD**	**RCD**	**RCBD**	**LND4**	**NBD4**	**RCND4**	**RCNBD4**
CRM	**105.0000**	105.0000	105.0000	105.0000	105.0000	105.0000	105.0000	105.0000
RBM	111.1778	**105.0000**	108.0804	106.4626	109.9506	106.2476	108.0953	106.5304
RCM	145.6674	129.3118	**125.6508**	145.4067	139.9116	140.9631	130.7638	138.4211
RCBM	158.8660	141.8568	139.1012	**125.7446**	143.1771	144.6677	142.0032	130.6123
LNM	164.2687	159.3360	166.5983	162.9767	**129.6608**	130.0774	134.1759	134.1177
NBM	182.9066	161.1575	176.2992	167.2595	142.5026	**131.8274**	140.5860	137.6629
RCNM	305.5027	249.2212	249.0576	291.5403	222.5107	239.8034	**170.4020**	232.5888
RCNBM	343.4111	272.6616	305.6709	266.3991	265.5920	223.2733	222.4862	**173.6894**

Table 6.7: Comparisons of the designs under the different models-class 2

Models	**Optimal designs**		
	RCBD3	**NBD5**	**RCNBD5**
CRM	105.0000	105.0000	105.0000
RBM	105.8226	105.0000	105.0000
RCM	139.9143	145.6178	144.3373
RCBM	**125.8301**	148.1647	133.0591
LNM	161.8658	135.0065	139.0961
NBM	167.5187	**135.5649**	140.1965
RCNM	274.6812	237.0845	243.7944
RCNBM	254.8142	242.3752	**189.0563**

In general, we can infer that when we believe that there may be spillover effects due to a structure governing the plots under experimentation, it is sensible to incorporate them in the model. For this second network specification (of the Farmer's case) accounting for the network effects even when we question their true existence does not do much harm in the design efficiencies (we are not losing more than 12% efficiency), which means that we are better off accounting for them than ignoring them.

Experimental design implemented in Rothamsted

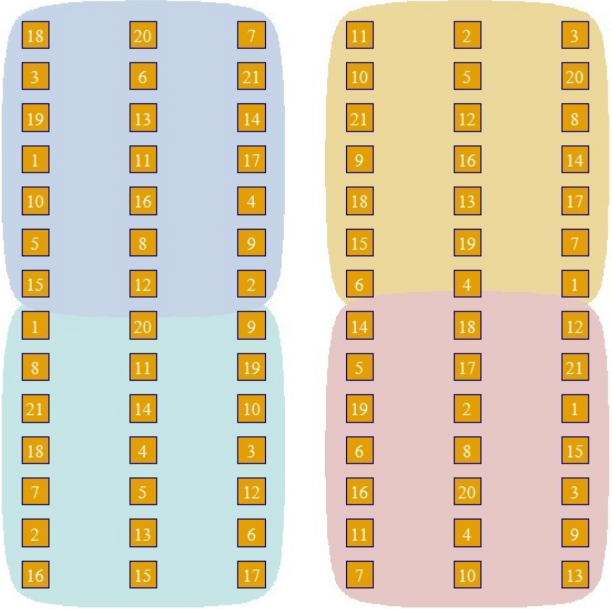
We obtain the optimality function values for the resolved α-RCD, which has been used for the relevant agricultural experiment conducted in the year 2016 under the different models (illustrated in Table 6.8). The design is provided in Figure 6.4. By wrongly ignoring the network effects although they exist, we observe that the design efficiency, with respect to the corresponding model which describes them which is a RCNBM, is 298.3764/518.0070 = 57.6% for the Farmer's case and 189.0563/281.5975 = 67.1% for the King's case. This implies that when we account for the network effects we considerably increase the design efficiency when we believe in the underlying network structure governing the plots.

Table 6.8: Optimality function values under the different models for α-RCD

Models	**Optimal designs**	
	Farmer's case	**King's case**
CRM	105.0000	105.0000
RBM	105.0000	105.0000
RCM	131.9278	131.9278
RCBM	136.5855	136.5855
LNM	153.4577	174.5538
LNBM	154.5439	185.9997
RCNM	258.4022	535.2381
RCNBM	281.5975	518.0070

From this agricultural trial we decide which treatments are allocated to the plots while accounting for potential important neighbour effects that are likely to affect the plot responses through an appropriately pre-specified adjacency matrix. In practice, the adjacency matrix is tailor-made reflecting the suspected underlying interference structure

Figure 6.4: Experimental design in Rothamsted α-RCD (year 2016)



among plots. The choice of this matrix can also address irregular regions demanding further potential constraints. According to the specific problem at hand, the experimenter should appropriately choose the adjacency matrix, suggest a suitable model to fit and optimise the design for that model for estimating the important parameters of interest. A conclusion drawn from the comparison of the optimal designs in this section is that unsystematic designs that ignore network effects may lead to poor results.

We present the non-resolved and resolved row column designs: RCNBD1, RCNBD3, RCNBD4 and RCNBD5 in Figures 6.5, 6.6, 6.7 and 6.8 respectively.

Figure 6.5: Near-optimal RCNBD1

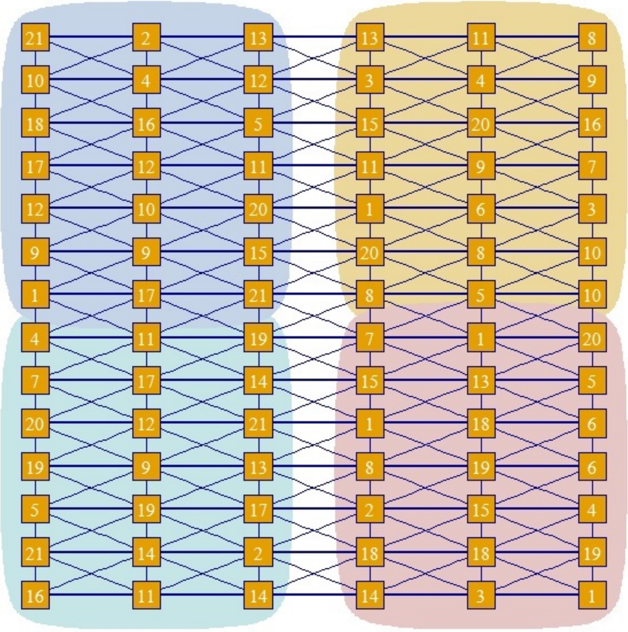


Figure 6.6: Near-optimal RCNBD3

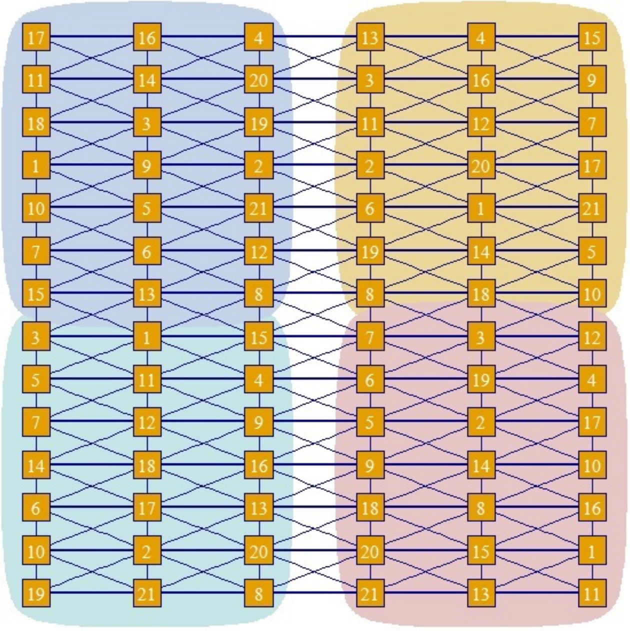
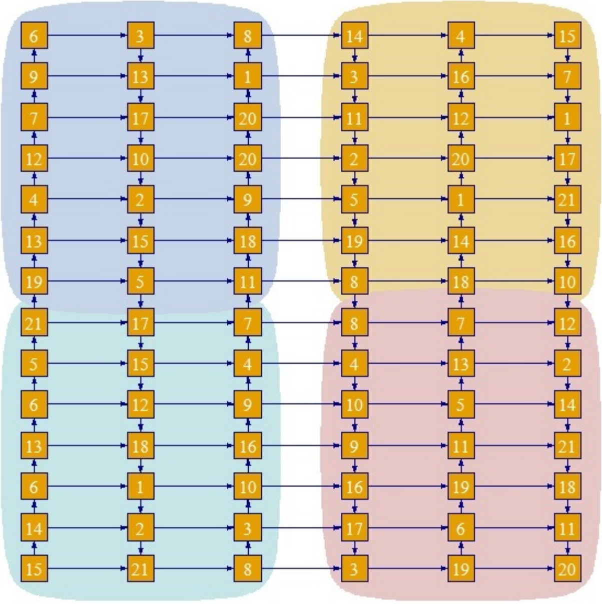
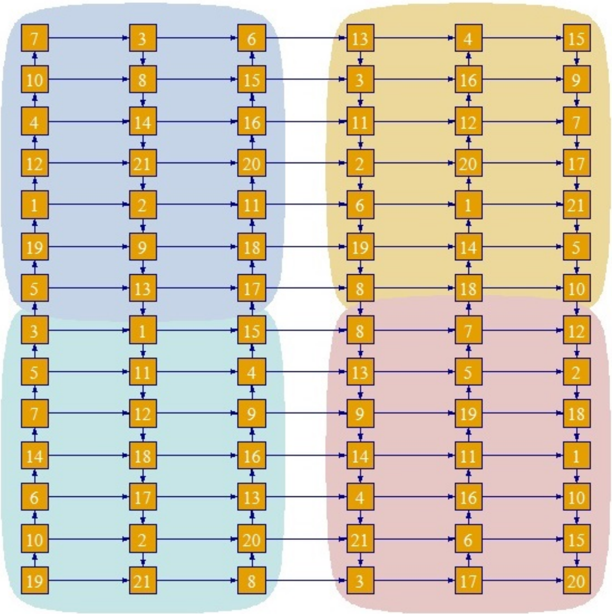


Figure 6.7: Near-optimal RCNBD4



In constructing designs for large numbers of treatments and when there may be treatment interference due to the neighbour structure underlying the plots, it is sensible to have each replicate of every treatment close to at least one replicate of all the other treatments. Moreover, it is desirable that adjacent plots receive the same treatment at least once. This observation is true also when we require resolvability. The designs of this chapter can be found in Appendix C.

Figure 6.8: Near-optimal RCNBD5



6.6 Discussion

In this chapter, we attempted to control for the potential variation resulting from treatment interference or farm operations in order to improve the precision of treatment comparisons. We show that optimal designs with network effects outperform conventional designs in terms of efficiency as they maximise the separation of treatment information from other sources of variation. Another remark is that including network effects that might be important is better than ignoring them and still the resulting optimal designs are good enough under the conventional models CRM, RBM, RCM etc. When we account for network effects, the optimal designs presented a common feature: each replicate of every treatment is very close to at least one replicate of each other treatment. To verify this, refer to the optimal designs presented in Appendix C. This is desirable and its importance has been highlighted by Freeman (1979) for the case of row-column designs. For the construction of designs with many treatments in networked experiments, a desirable pattern is that pairs of closely connected units receive the same treatment. This crucial observation seems to play a major role in the design efficiency of experiments with connected units.

We considered observations collected from connected units to be dependent, with the degree of dependence decreasing along network distance between units. It is important to note that various alternative specifications of the adjacency matrix could be also investigated that will be relevant to the interest of the experimenter: including measurements of neighbourhood properties among the plots to better reflect the spatial structure or a different ‘arbitrary’ perception of the influences among plots based on the experiment at hand. In practice, the specification of this matrix might be a result of the scientific knowledge of the experimenter or elicitation of information from farmers on the suspected sources of spillover effects based on experience which is likely to strongly affect the treatment effects' differences.

In practice, there is an increasing demand for such designs (especially for use in agricultural field trials) in situations where we assume an underlying structure which suggests neighbouring treatment interference. The advantage of the suggested design approach is that we can control for the neighbouring environment according to the causes, directions and weights of the neighbouring interference. For instance, neighbour effects may depend on the speed and direction of wind or periods in shade, which can result in an appropriate definition of the connectivity matrix imposing suitable weights for the left or right neighbours, etc. This highlights the importance of defining the adjacency matrix based on requirements of the experiment at hand. If the network structure is adequately modelled, this design procedure may be expected to cause an increase in precision of the treatment contrasts.

The specification of modelling approaches considering autocorrelation structures has also been of great practical importance in spatial experiments (e.g. agricultural field experiments). Section 7.2.1 builds on fundamental approaches towards this direction and aims to encourage future research on designing experiments on networks when there is sufficient network correlation among neighbouring units (including, but not limited to, spatial correlation).

Chapter 7

Discussion and directions for future research

7.1 Summary

Although a vast number of networked experiments has been performed over the last decade, little light has been cast on the statistical design of experiments. Underestimating the importance of experimental design, could lead to invalid experimental conclusions for the scientific questions of interest. This research contributes to filling this gap on designing experiments in environments that comprise interacting entities (including but not limited to social networks). Examples of experiments that can benefit from the suggested design techniques are presented in Table 1.1.

A successful experimental design relies on the careful balancing of several features including applicability, simplicity, time and cost efficiency. This thesis provided flexible and effective methods to improve the statistical design of experiments on connected units, considering both artificial and real-world networks of different shapes and sizes, and investigated their impact on the properties of the design. We showed that designs that account for network effects outperform standard designs in terms of efficiency and design bias, while at the same time being fast to compute. One can find an efficient design by exploiting the structural properties of the network at hand, which describes the relationships among units. This method suggests the decomposition of the optimal design problem into a set of nested sub problems. The novelty of this research is justified by the very few contributions to the literature in this area as well as the lack of methods for designing experiments on networks, for instance, when they present high degrees of symmetry or in the presence of community structure. Many systems and problems can be regarded from the network perspective by appropriate specification of the connectivity matrix. Thus our approaches are easily extendable and adaptable to a wide class of experiments with interconnected units. This research has potential impact on how experiments on networks are set up in the future.

After we provided the necessary framework in Chapter 2, we built on the work of

Parker *et al.* (2016) in Chapter 3, deriving some analytical results and investigating how different topologies influence the patters of the optimally allocated treatments on the network members for the case of two unstructured treatments. Heuristically the experimental design for estimating the treatment effects tends to be balanced on the units having a similar number of connections, while for estimating the network effects the design is greatly influenced by the network's first and second order connections. To this end we provided key observations that facilitate the construction of optimal designs in real-networks. Moreover, analytical formulae of the optimality criteria were also obtained, which can be beneficial for speeding up the optimisation algorithm when dealing with irregular and complicated neighbour structures, by avoiding the inversion of the information matrix. Issues related to design efficiency and bias were also explored, showing that designs that account for network effects are fairly robust to some misspecification of the model or network under experimentation. However, since the micro-structure of the network has a significant impact on the design, changing the structure of the network radically (and consequently its properties), by either adding or removing a large number of edges, will affect the optimality of the designs.

Chapter 4 described a practical algorithmic approach in the search for an optimal design in a network having a high degree of symmetry. We saw that an automorphism of a graph is a form of symmetry in which the graph is mapped onto itself while preserving the edge-vertex connectivity. This automorphism can also be called a ϕ_1-automorphism, which means that the connectivity structure will be invariant to the automorphisms under ϕ_1. We showed that permutations of structurally equivalent vertices are exact symmetries of the graph (i.e. automorphisms), and by utilising those symmetries we can speed up the search for optimal designs for moderate network sizes. This is based on the idea of breaking down the networks into symmetric and asymmetric components and by obtaining simultaneously sub-designs for each one of them (taking into account observed patterns of the optimally allocated treatments on the symmetric motifs). By using the *nauty* program, we can find all the automorphism of large graphs rapidly (sub second processing time). Subsequently these sub-designs make up the overall design for the original network. If the degree of symmetry is high in the network then the skeleton can be significantly smaller than the original network from which it is derived, resulting in a faster design optimisation process. We encourage future researchers to further investigate the impact of network symmetries on the design of experiments, for instance by embedding additionally network information related to its topological properties (e.g. degree distribution, shortest paths etc.) into its network components.

Considering social networks, people have a strong tendency to associate with others whom they perceive as being similar to themselves by means of traits and this tendency is called homophily (or assortative mixing) (Newman, 2010, Ch. 7.13). This led us to the natural definition of blocking, with experimental units that are highly associated with each other and we anticipate them to behave similarly to an external stimulus. Following this idea, in Chapter 5 we constructed optimal block designs with network effects, where blocks were defined using spectral clustering techniques achieving optimal modularity. Spectral clustering has proven to be a useful tool in designing an experiment and determining, if it exists, a natural division of its vertices into non-overlapping groups (these groups may be of any size). By detecting communities one can get a further insight into the inherent structure of the network population under investigation as it is equivalent to investigating statistical properties of a graph, ignoring the roles played by specific subgraphs, and hence identifying substructures which could influence important functions. We showed that designs are robust to network misspecifications to the extent that the network retains its characteristic properties. Moreover, simulations have shown that when we misspecify the number of blocks or units in blocks, the properties of the design become worse but only up to the same extent of the standard designs without network effects. Another important finding from Chapter 5 is that when the clustering coefficient is large, such as in small-world networks, then the model ignoring the block effects leads to highly inefficient designs.

In Chapter 6, we showed that the methods considered can be easily adapted to agricultural field experiments with regular or even irregular layouts. When plots of land are close to each other, this can result in treatment interference. A comparison of several optimal designs showed that designs that control for more sources of potential variation, such as transmission of treatment effects due to the topological properties or farmer operations, perform consistently better that the conventional designs used in such experiments.

Another methodological approach for adjusting for response interference rather than treatment interference is explored in the next section, where we account for potential interference among units by modelling the autocorrelation structure of the responses. Thus the responses of the units are correlated and this correlation is attributed to neighbouring connected units on a multidirectional network. The optimal designs will be based on conditional criteria of the asymptotic variance covariance structure of the treatment effects.

7.2 Current work: Autoregressive network effects model

In this section we assume that the ‘nearer’ the units are the more correlated their responses are. Previously we assumed the existence of neighbour effects, captured by the network term in the model, where each unit is influenced not only by the actual treatment received, but also by treatments present on neighbouring units. A different approach in a regression context, as we discussed in Chapters 1 and 3 (see autoregressive models NEM and NDM), is when influence occurs through the autocorrelation of the responses or through the autocorrelation of the errors (either in the form of non-constant error variances in a regression model or in the form of variable regression coefficients). Focusing on the former, Section 7.2.1 introduces the Autoregressive Network effects Model (ANM) which directly accounts for the network heterogeneity, and incorporates both a regression component for the mean and an autoregressive component in the responses to reflect the network structure. After estimating the model parameters by maximum likelihood in Section 7.2.2, we derive the large sample asymptotic approximation for the variance-covariance matrix of the MLE of the regression coefficients of that extended model in Section 7.2.3. We finish with a discussion of the ongoing research in the field of experimental design and network autoregressive models.

7.2.1 Autoregressive network effects model (ANM)

The model considered here incorporates both a regression component for the mean and an autoregressive component for the responses to reflect the network structure. Hence, we can write

$$\text{ANM: } y_j = \mu + \tau_{r(j)} + \rho \sum_{k=1}^{n} A_{jk} y_k + \epsilon_j \tag{7.1}$$

for $j = 1, 2, \ldots, n$, where y_j is the resulting response from unit j receiving treatment $r(j) = s \in \{1, 2, \ldots, m\}$, μ is the overall mean, $\tau_{r(j)}$ is the (direct) effect of treatment $r(j)$ applied to unit j, the scalar ρ is the autocorrelation parameter, A_{jk} is the adjacency matrix indicating the presence of connections between units j and k (see Section 2.4), y_k is the response of the neighbouring unit k when there is a connection between units j and k. The error terms, denoted by ϵ_j, are assumed to be independent and identically distributed (i.i.d.) with mean 0 and constant variance σ^2. Note that when $\rho = 0$, the expression (7.1) reduces to the CRM (2.4).

Assuming normality of the errors we can re-write the model (7.1) in matrix notation

$$\boldsymbol{y} = \rho A \boldsymbol{y} + X\boldsymbol{\beta} + \boldsymbol{\epsilon}, \quad \boldsymbol{\epsilon} \sim \mathcal{N}_n\left(0, \sigma_\epsilon^2 I\right), \tag{7.2}$$

where $\boldsymbol{\beta} = \begin{pmatrix}\mu & \boldsymbol{\tau}^{*T}\end{pmatrix}^T = \begin{pmatrix}\mu & \tau_1 \ldots \tau_{m-1}\end{pmatrix}^T$ is the vector of parameters and $X = \begin{pmatrix}\boldsymbol{1} & X_{\tau}{}^{\star}\end{pmatrix}$ is the extended design matrix. There are no columns corresponding to the m-th treatment effect τ_m, since we assumed that to be zero (for more details on the notation refer to Chapter 3.2). Letting $G = I - \rho A$, we have that $G\boldsymbol{y} \sim \mathcal{N}\left(X\boldsymbol{\beta}, \sigma_\epsilon^2 I\right)$. Assuming that $|\rho|$ is less than one, and that the matrix elements are such that $(I - \rho A)$ is non singular, we have that $\mathbb{E}\left[\boldsymbol{y}|X\right] = G^{-1}X\boldsymbol{\beta}$ and $var\left(\boldsymbol{y}|\boldsymbol{\beta}\right) = G^{-1}\sigma_\epsilon^2 I \left(G^{-1}\right)^T = \sigma_\epsilon^2 G^{-2}$. Note that G is symmetric for non-directed graphs (as A is symmetric) and therefore $G^T = G$. Thus $\boldsymbol{y} \sim \mathcal{N}_n\left(G^{-1}X\boldsymbol{\beta}, \sigma_\epsilon^2 G^{-2}\right)$.

Theorem 7.1 (Harville, 1997, Th. 18.2.16.): Let $M_{n\times n}$ be a square matrix. Then $I + M + M^2 + \cdots$ converges if and only if $\lim_{k\to\infty} M^k = 0$, in which case $I - M$ is non singular and $(I - M)^{-1} = I + M + M^2 + \cdots$ (where $M^0 = I$).

Therefore, we can take $M = \rho A$ where $|\rho| < 1$ and since $0 \leq A_{jk}^k \leq 1$, $\forall j, k$ then $|M|$<1 which results in $\lim_{q\to\infty} M^q = 0$. For more details on the relevant matrix algebra refer to Harville (1997, Ch.18). Thus we have

$$\boldsymbol{y} = (I - \rho A)^{-1} X\boldsymbol{\beta} + (I - \rho A)^{-1}\boldsymbol{\epsilon} \tag{7.3}$$

in which each inverse can be expanded into an infinite series, including both the explanatory variables and the error terms. In the following section we see that the method of ordinary least squares proves to be biased and inconsistent in this setting. This is because OLS ignores the term $log|I - \rho A|$ resulting from the nature of the spatial dependence. For this reason we employ maximum likelihood methods.

7.2.2 Maximum likelihood estimation

For the estimation of the parameters of the ANM (7.2) we perform maximum likelihood (ML) estimation. Maximum likelihood estimation of (spatial) autoregressive models was first outlined by Ord (1975). The joint loglikelihood for the ANM does not equal the sum of the loglikelihoods associated with the individual observations. This is due to the multi-directional nature of the network dependence, which results in a Jacobian term that is the determinant of a full $n \times n$ matrix, i.e. $|I - \rho A|$. This makes ML estimators the preferred estimators in such setting. In particular, for the ANM, the loglikelihood is based on the multivariate normal case. Since $\boldsymbol{\epsilon} \sim \mathcal{N}\left(0, \sigma^2 I\right)$ where $\boldsymbol{\epsilon} = G\boldsymbol{y} - X\boldsymbol{\beta}$ and $G = I - \rho A$ it follows that $G\boldsymbol{y} \sim \mathcal{N}\left(X\boldsymbol{\beta}, \sigma^2\right)$ or $\boldsymbol{y} \sim \mathcal{N}\left(G^{-1}X\boldsymbol{\beta}, \sigma^2 G^{-2}\right)$. Note that $|\sigma^2 G^{-2}| = \sigma^{2n}|G^{-2}|$. The likelihood function for ρ, σ^2 and $\boldsymbol{\beta}$ is

$$L = (2\pi)^{-n/2}\left(\sigma^{2n}\right)^{-1/2}|G| \exp\left(-\frac{1}{2\sigma^2}(G\boldsymbol{y} - X\boldsymbol{\beta})^T (G\boldsymbol{y} - X\boldsymbol{\beta})\right).$$

Thus the log-likelihood $\ell = \log L$ is given as

$$\begin{aligned}\ell\left(\sigma^2, \boldsymbol{\beta}, \rho\right) &= -\frac{n}{2}\log(2\pi) - \frac{n}{2}\log\left(\sigma^2\right) + \log|G| - \frac{1}{2\sigma^2}(G\boldsymbol{y} - X\boldsymbol{\beta})^T (G\boldsymbol{y} - X\boldsymbol{\beta}) \\ &= -\frac{n}{2}\log(2\pi) - \frac{n}{2}\log(\omega) + \log|G| - \frac{1}{2\omega}(G\boldsymbol{y} - X\boldsymbol{\beta})^T (G\boldsymbol{y} - X\boldsymbol{\beta}), \quad (7.4)\end{aligned}$$

where $\sigma^2 = \omega$ for ease of notation. We calculate the first order derivatives

$$\frac{\partial \ell}{\partial \omega} = -\frac{n}{2}\frac{1}{\omega} + \frac{1}{2\omega^2}(G\boldsymbol{y} - X\boldsymbol{\beta})^T (G\boldsymbol{y} - X\boldsymbol{\beta}).$$

It follows that the maximum likelihood estimate of ω will be given by setting the above expression equal to zero:

$$\frac{\partial \ell}{\partial \omega} = 0 \Rightarrow n\hat{\omega} = \left(G\boldsymbol{y} - X\hat{\boldsymbol{\beta}}\right)^T \left(G\boldsymbol{y} - X\hat{\boldsymbol{\beta}}\right). \tag{7.5}$$

Moreover, we have that

$$\begin{aligned}\epsilon^T\epsilon &= (G\boldsymbol{y} - X\boldsymbol{\beta})^T (G\boldsymbol{y} - X\boldsymbol{\beta}) \\ &= (G\boldsymbol{y})^T G\boldsymbol{y} - (G\boldsymbol{y})^T X\boldsymbol{\beta} - (X\boldsymbol{\beta})^T G\boldsymbol{y} + (X\boldsymbol{\beta})^T X\boldsymbol{\beta} \\ &= (G\boldsymbol{y})^T G\boldsymbol{y} - 2(X\boldsymbol{\beta})^T G\boldsymbol{y} + \boldsymbol{\beta}^T X^T X\boldsymbol{\beta}, \qquad (7.6)\end{aligned}$$

where the transpose of the scalar is the scalar, i.e. $(G\boldsymbol{y})^T X\boldsymbol{\beta} = (X\boldsymbol{\beta})^T G\boldsymbol{y}$. Also

$$\frac{\partial \ell}{\partial \boldsymbol{\beta}} = -\frac{1}{2\omega}\left(-2X^T G\boldsymbol{y} + 2X^T X\boldsymbol{\beta}\right) = \frac{1}{\omega}\left(X^T G\boldsymbol{y} - X^T X\boldsymbol{\beta}\right).$$

Setting this equal to zero gives the ML estimators

$$\frac{\partial \ell}{\partial \boldsymbol{\beta}} = 0 \Rightarrow \hat{\boldsymbol{\beta}} = \left(X^T X\right)^{-} X^T G\boldsymbol{y}. \qquad (7.7)$$

For a known ρ, the GLS or ML is determined from the usual first-order conditions:

$$\begin{aligned}\hat{\boldsymbol{\beta}} &= \left(X^T X\right)^{-} X^T G\boldsymbol{y} \\ \hat{\sigma}^2 &= \frac{\left(G\boldsymbol{y} - X\hat{\boldsymbol{\beta}}\right)^T \left(G\boldsymbol{y} - X\hat{\boldsymbol{\beta}}\right)}{n}\end{aligned}$$

with $var\left(\hat{\boldsymbol{\beta}}\right) = \hat{\sigma}^2\left(X^T X\right)^{-}$. However, ρ is generally unknown and may be estimated by ML. Following Besag and Kempton (1986, p.243-4), from an iterative estimation of $\boldsymbol{\beta}$, ω and ρ, we compute $\hat{\rho}$ by a search on the function $const - \frac{n}{2}\log(\hat{\omega}) + \log|G|$ (maximisation) and then substitute to the log-likelihood ℓ and to the expressions of the ML estimators (7.7) and (7.5) (see the computational method employed by Mead, 1967, p.192). For additional discussion on existing suggested iterative procedures to obtain the estimators refer to Cliff and Ord (1981, p.232) and references therein.

The partial derivative of ℓ with respect to ρ is

$$\frac{\partial \ell}{\partial \rho} = -A|G|^{-1} - \frac{1}{2\omega}2\left(G\boldsymbol{y} - X\boldsymbol{\beta}\right)^T \boldsymbol{y}A. \qquad (7.8)$$

7.2.3 Asymptotic variance of the maximum likelihood estimators

According to Kendall and Stuart (1979, p.59-60) and Cliff and Ord (1981, p.241-2), the large sample variance-covariance matrix (denoted by V) is given by

$$V^{-1} = -\mathbb{E}\left[\frac{\partial^2 \ell}{\partial \vartheta_r \partial \vartheta_s}\right] \quad \text{or} \quad V = -\mathbb{E}\left[\frac{\partial^2 \ell}{\partial \vartheta_r \partial \vartheta_s}\right]^{-1}$$

where ϑ_r and ϑ_s are the typical parameters.

Calculating the second order derivatives, we obtain

$$\frac{\partial^2 \ell}{\partial \omega^2} = \frac{n}{2\omega^2} - \frac{2\omega}{2\omega^4} n\hat{\omega},$$

$$\frac{\partial^2 \ell}{\partial \boldsymbol{\beta}^2} = -\frac{1}{\omega} X^T X,$$

$$\frac{\partial^2 \ell}{\partial \rho^2} = A^2 |G|^{-2} - \frac{1}{\omega} \boldsymbol{y}^T \boldsymbol{y} A^2,$$

$$\frac{\partial^2 \ell}{\partial \boldsymbol{\beta} \partial \omega} = \frac{X^T G \boldsymbol{y}}{\omega^2} - \frac{X^T X \boldsymbol{\beta}}{\omega^2} = \frac{-X^T \left(G\boldsymbol{y} - X\boldsymbol{\beta} \right)}{\omega^2},$$

$$\frac{\partial^2 \ell}{\partial \boldsymbol{\beta} \partial \rho} = -\frac{X^T A \boldsymbol{y}}{\omega} \quad \text{and}$$

$$\frac{\partial^2 \ell}{\partial \omega \partial \rho} = \frac{1}{\omega^2} (G\boldsymbol{y} - X\boldsymbol{\beta})^T \boldsymbol{y} A.$$

Letting $B = AG^{-1}$ and $\boldsymbol{y}_L = A\boldsymbol{y}$ we have

$$\mathbb{E}\left[\boldsymbol{y}_L\right] = BX\boldsymbol{\beta},$$

$$\mathbb{E}\left[\epsilon^T \boldsymbol{y}_L\right] = \omega \operatorname{tr}(B) \quad \text{and}$$

$$\mathbb{E}\left[\boldsymbol{y}_L^T \boldsymbol{y}\right] = \omega \operatorname{tr}\left(B^T B\right) + \mathbb{E}\left[\boldsymbol{y}_L\right]^T \mathbb{E}\left[\boldsymbol{y}_L\right] = \omega \operatorname{tr}\left(B^T B\right) + (BX\boldsymbol{\beta})^T (BX\boldsymbol{\beta}).$$

Note that $\mathbb{E}\left[\hat{\omega}\right] = \omega$ asymptotically. Implementing the above equations we have

$$\mathbb{E}\left[-\frac{\partial^2 \ell}{\partial \omega^2}\right] = -\mathbb{E}\left[\frac{n}{2\omega^2} - \frac{2\omega}{2\omega^4} n\hat{\omega}\right] = -\mathbb{E}\left[\frac{n\omega - 2n\omega}{2\omega^3}\right] = \frac{n}{2\omega^2},$$

$$\mathbb{E}\left[-\frac{\partial^2 \ell}{\partial \boldsymbol{\beta}^2}\right] = \frac{1}{\omega} X^T X,$$

$$\begin{aligned}\mathbb{E}\left[-\frac{\partial^2 \ell}{\partial \rho^2}\right] &= \mathbb{E}\left[-\left(A|G|^{-1}\right)^2 + \frac{1}{\omega} \boldsymbol{y}_L^T \boldsymbol{y}_L\right] = -\operatorname{tr}\left(A|G|^{-1}\right)^2 + \frac{1}{\omega} \mathbb{E}\left[\boldsymbol{y}_L^T \boldsymbol{y}_L\right] \\ &= \operatorname{tr}(B)^2 + \operatorname{tr}\left(B^T B\right) + \frac{1}{\omega} (BX\boldsymbol{\beta})^T (BX\boldsymbol{\beta}),\end{aligned}$$

$$\mathbb{E}\left[-\frac{\partial^2 \ell}{\partial \boldsymbol{\beta} \partial \omega}\right] = 0,$$

$$\mathbb{E}\left[-\frac{\partial^2 \ell}{\partial \boldsymbol{\beta} \partial \rho}\right] = \frac{1}{\omega} X^T \mathbb{E}\left[\boldsymbol{y}_L\right] = \frac{1}{\omega} X^T B X \boldsymbol{\beta} \quad \text{and}$$

$$\mathbb{E}\left[-\frac{\partial^2 \ell}{\partial \omega \partial \rho}\right] = \mathbb{E}\left[\frac{1}{\omega^2} \epsilon^T \boldsymbol{y}_L\right] = \frac{1}{\omega} \operatorname{tr}(B).$$

Writing the parameters in order $\omega, \boldsymbol{\beta}, \rho$ the asymptotic variance is

$$\text{Asymptotic } \; V\left(\omega, \boldsymbol{\beta}, \rho\right) = \begin{pmatrix} \mathbb{E}\left[-\frac{\partial^2 \ell}{\partial \omega^2}\right] & \mathbb{E}\left[-\frac{\partial^2 \ell}{\partial \omega \partial \boldsymbol{\beta}}\right] & \mathbb{E}\left[-\frac{\partial^2 \ell}{\partial \omega \partial \rho}\right] \\ & \mathbb{E}\left[-\frac{\partial^2 \ell}{\partial \boldsymbol{\beta}^2}\right] & \mathbb{E}\left[-\frac{\partial^2 \ell}{\partial \boldsymbol{\beta} \partial \rho}\right] \\ & & \mathbb{E}\left[-\frac{\partial^2 \ell}{\partial \rho^2}\right] \end{pmatrix}^{-1}$$

$$= \begin{pmatrix} \frac{n}{2\omega^2} & 0 & \frac{\operatorname{tr}(B)}{\omega} \\ & \frac{X^TX}{\omega} & \frac{X^TBX\boldsymbol{\beta}}{\omega} \\ & & \operatorname{tr}\left(B\right)^2 + \operatorname{tr}\left(B^TB\right) + \frac{(BX\boldsymbol{\beta})^T(BX\boldsymbol{\beta})}{\omega} \end{pmatrix}^{-1}$$

$$= \frac{1}{\alpha} \begin{pmatrix} 2\omega^2\left(\operatorname{tr}\left(B\right)^2 + \operatorname{tr}\left(B^TB\right)\right) & 2\operatorname{tr}\left(B\right)^2 BX\boldsymbol{\beta}\omega X^- & -2\operatorname{tr}\left(B\right)^2\omega \\ & Z\left(X^TX\right)^- & -n(BX\boldsymbol{\beta})^T\left(X^T\right)^- \\ & & n \end{pmatrix},$$

where $Z = \operatorname{tr}\left(B^TB\right)n\omega - 2\operatorname{tr}\left(B\right)^2\omega + n(BX\boldsymbol{\beta})^T\left(BX\boldsymbol{\beta}\right) + \operatorname{tr}\left(B\right)^2 n\omega$ and $\alpha = \operatorname{tr}\left(B^TB\right)n + \operatorname{tr}\left(B\right)^2(n-2)$. The lower half is filled by symmetry. Note also that while the covariance between $\boldsymbol{\beta}$ and ω is 0 in the standard regression model, this is not the case for ρ and ω. Moreover, $|I - \rho A| = \prod_{i=1}^{n}(1-\lambda_i)$ where $\lambda_1 \geq \lambda_2 \geq \ldots \geq \lambda_n$ are the eigenvalues of A and therefore $\operatorname{tr}\left(B\right) = \sum_{i=1}^{n} \frac{\lambda_i}{1-\lambda_i}$ (Ord, 1975, p.121).

The autocorrelation structure leads to an information matrix that is a function of ρ, a generally unknown parameter (see Matthews, 1987, for an explicit expression for the joint information matrix for direct and residual effects discussed in the context of correlated errors). As a consequence it becomes much more difficult to obtain general results for optimality of designs. Typically optimal designs are chosen to optimise (either minimise or maximise) the value of a given function of the expected information matrix associated with the estimation problem (see Section 2.1). The importance of the information matrix stems from its role in maximum likelihood estimation, where it is proportional to the inverse of the asymptotic variance matrix of the parameter estimators (Davison, 2003, p.118). It can thus be thought of as a measure of the likely precision of the estimators resulting from the experiment. A locally L-optimal design minimises the trace of the asymptotic variance-covariance matrix of the MLE at a specified value of $\boldsymbol{\beta}$, which is

$$V(\hat{\boldsymbol{\beta}}) = \frac{\operatorname{tr}\left(B^T B\right) n\omega - 2\operatorname{tr}(B)^2 \omega + n(BX\boldsymbol{\beta})^T(BX\boldsymbol{\beta}) + \operatorname{tr}(B)^2 n\omega}{\operatorname{tr}(B^T B)\, n + \operatorname{tr}(B)^2 (n-2)} \left(X^T X\right)^-. \quad (7.9)$$

The optimal designs will rely on the asymptotic variance of the ML estimators (Equation (7.9)). Ongoing work concerns the construction of such locally optimal designs evaluated with local criteria or pseudo-Bayesian criteria. The optimality criteria will be defined conditional on a set of parameters $\boldsymbol{\beta}$. In particular, we can define $\phi_1(X; \boldsymbol{\beta}, \omega, \rho)$ as the average variance of differences between direct treatment effects and $\phi_2(X; \boldsymbol{\beta}, \omega, \rho)$ as the variance of ρ (or some function of ρ). These 'conditional criteria' depend on the unknown parameters, so cannot be used to compare designs. However, we can use them to define some unconditional criteria. The simplest are the local optimality criteria, which can take some point prior estimates for the unknown parameters, in order to obtain in this way locally L-optimal designs on given networks. Thus we substitute $\boldsymbol{\beta} = \boldsymbol{\beta}_0$, $\omega = \omega_0$ and $\rho = \rho_0$ to get ${\phi_1}^{(0)}(X)$ and ${\phi_2}^{(0)}(X)$ for some point prior estimates (or guesses). Alternatively, we can consider more robust criteria, such as the expectation of $\phi_1(X; f(\boldsymbol{\beta}, \omega, \rho))$ over a prior distribution of $\boldsymbol{\beta}$, ω and ρ.

7.3 Future work

There is great scope for future research in various fields such as: the specification of the connectivity matrix (discussed in Section 2.4); the choice of an appropriate model capturing the network dependencies (e.g. autoregresive model briefly discussed earlier); the inclusion of more complicated treatment structures (e.g. factorial designs with network effects); additional types of algorithmic approaches (some of which are discussed in Sections 2.3, 3.6, 4.3 and 6.3); tracking additional properties of the network structure; compound criteria that take into account possible misspecification of the model when designing an experiment on networks; capturing the dynamics of the networks; developing techniques to handle complicated random effect structures and others. Some of the possibilities for future work and improvements have been highlighted in different parts of this thesis. There are also various levels of sophistication one can add. Without going into much detail we discuss some of them and provide some hints to future research directions.

Criterion for estimating the total treatment effects: The experimenter might be interested in finding an optimal design for estimating the total treatment effect by a weighted sum of the two criteria mentioned in this thesis. Following Bailey and Druilhet (2004), another suggestion of an (unweighted) optimality criterion for estimating the total effects is: $\phi_3 = \boldsymbol{s}^T M^- \boldsymbol{s}$ where $\boldsymbol{s} = (0\ n_1 \ldots\ n_{m-1}\ \sum_{s=1}^m l_{s1} \ldots\ \sum_{s=1}^m l_{sm})^T$. This criterion can be derived using the formula $1/m \sum_{s=1}^m var\boldsymbol{\beta}_s$, where the total treatment effect of a particular treatment s is given by $\boldsymbol{\beta}_s$, where $\boldsymbol{\beta}_s = n_s\tau_s + \gamma_s(\sum_{s'=1}^m l_{ss'})$.

Recall that n_s is the number of units given treatment s, τ_s is the treatment effect, γ_s is the network effect, $l_{ss'}$ is the number of connections between units given treatment s and those given treatment s'.

A/B or multivariate testing: In the comparative experiment discussed in the first section of this thesis, Section 1.1, the effectiveness of the advertisements can be quantified by the quantity of products purchased the week following the advertising or the profit generated due to the advertisements. Another measurement of the responses could be related to the clicks of the advertisement in a test period time. Thus the model could be a logistic or a log-linear one.

Experiments on stochastic networks: Networks in real life evolve over time, with vertices or edges either appearing or disappearing. Consider a network with connections changing over time. Let $A = A(t)$, where $A_{ij}(t) > 0$ denotes the particular weight corresponding to the element (i, j) at time t. Organising the weights at each given time we get a stochastic adjacency matrix $A(t)$. This has many dynamical and structural implications. The connections targeted for the experiment can correspond for instance to frequent interactions. Note that the degree of contact between neighbouring units has also been investigated in spatial applications. Thus we can consider the neighbouring contacts which have 'active ties' between them (i.e. frequency of interactions over a long period of time). Thus a researcher should consider methodologies from artificial intelligence and computer science such as machine learning and data mining, for describing the structure of social network and contagion pathways and as such to better understand treatment propagation or spread of behaviour along the shortest paths. A possible approach could be to consider a model for designing an experiment on a network, conditional on a graph model which generates that network.

Experiments in multilayer networks: Real-world networks are becoming more complicated. An individual may belong to several networks at the same time, for instance in two social networking platforms. In that case we can consider the two ego-networks as two different connectivity structures independently from one another (incorporating them in the model), or merging them to one single network adopting the methods we have already considered, or the two networks can correspond to different layers of connectivity. One can distinguish different types of multilayer networks depending on the interaction between the different layers. This poses a challenge to the current experimental design theory. A possible way to regard this problem is by considering a multistage experiment with blocking.

Most methods discussed in this thesis can be adapted to a Bayesian framework. For instance, consider that for a particular experiment there may exist different possible adjacency matrices that can explain the various scenarios of interference instead of an assumed fixed adjacency matrix. Time may also be accounted for in the design of an experiment resulting in an association of the responses during the course of the experiment. For example, extending the framework to a time series analysis might be appropriate. Moreover, revisiting Section 3.5, a pseudo-Bayesian approach can be applied for constructing robust designs when there is a level of misspecification of the network. A possible way to proceed involves the construction of an 'intermediate design among an adequate number of designs obtained on simulated misspecified networks. This 'median' design is robust by definition to a number of different misspecifications of the network and therefore performs well in situations in which there is an uncertainty about the network connections. This uncertainty about the network connections could be described with appropriate 'priors'. The execution of this suggested computational method is time efficient and easily adaptable to any experiment.

Future research could involve exploring a wider classes of models, extending also to non-linear models, e.g. where time is incorporated in a quadratic term, or involving complicated variance structures. Another possible research question is how the design will be affected when choosing a subset of units in a network. Designing experiments in sampled networks is a broad area of research since it can substantially reduce the required time and cost. It is important to capture appropriately the statistical properties of the sampled population while sustaining the properties of the sampled network. In this thesis, we discussed issues related to the robustness of the design to network misspecification but also issues related to the impact of the network symmetries on the design efficiency. Some of these ideas and techniques might be useful when performing experiments on sampled networks. Moreover, on sampled networks clustering and symmetry-breaking techniques, such as determining the frequency of network motifs compared to their frequency in a random ensemble of networks with similar properties to the original, might be useful. Practical aspects on sampling methodologies are discussed in the work of Aral (2016), suggesting that typically preserving the graph properties (e.g. degree distribution) is preferable to ignoring it. Some of those important considerations are covered by Lee *et al.* (2006). When conducting large-scale experiments, it is essential to advance the statistical design of experiments to the era of Big Data. The topology based sampling or representation of the graph by its quotient point to the right future direction in order to maintain the representative graph properties and structures when designing experiments on them.

Our computational methods for the construction of optimal designs can be implemented, improved and extended by researchers and practitioners in any field that involves networks. It is worth mentioning that our on-going work aims at incorporating some of our methods into a practical package for the R statistical environment so as to aid practitioners in designing networked experiments. By further exploring and exploiting the network topology, we can better understand how the connections between experimental units affect the design of experiments. Designing experiments on networks is a complex area of research but at the same time a very promising one for dealing with practical problems. Therefore more attention should be given as many of the possibilities have yet to be explored in depth.

Appendix

A Results for Chapter 3

The determinant D of Equations (3.12) and (3.13) are obtained respectively as

$$
\begin{aligned}
D = {} & l_{11}^{\langle 2\rangle}\left(n_1 l_2^2 - 2l_2 l_{12} n_1 + n l_{12}^2\right) + l_{22}^{\langle 2\rangle}\left(n_1 l_1^2 - 2l_1 l_{11} n_1 + n l_{11}^2\right) + 2l_{12}^{\langle 2\rangle}(-l_1 l_2 n_1 + l_1 l_{12} n_1 + l_2 l_{11} n_1 \\
& - n l_{11} l_{12}) - n_1^2\left(l_{12}^{\langle 2\rangle}\right)^2 + n n_1\left(l_{12}^{\langle 2\rangle}\right)^2 - l_1^2 l_{12}^2 + 2l_1 l_2 l_{11} l_{12} - l_2^2 l_{11}^2 + l_{11}^{\langle 2\rangle} l_{22}^{\langle 2\rangle} n_1^2 - l_{11}^{\langle 2\rangle} l_{22}^{\langle 2\rangle} n n_1 \\
& = l_{11}^{\langle 2\rangle}\left(n_1 l_2^2 - 2l_2 l_{12} n_1 + n_1 l_{12}^2 + n_2 l_{12}^2\right) + {l_{22}}^{\langle 2\rangle}\left(n_1 l_1^2 - 2l_1 l_{11} n_1 + n_1 {l_{11}}^2 + n_2 {l_{11}}^2\right) \\
& + 2{l_{12}}^{\langle 2\rangle}\left[-n_1 l_1\left(l_2 - l_{12}\right) + n_1 l_{11}(l_2 - l_{12}) - n_2 l_{11} l_{12}\right] + n_1 n_2\left(l_{12}^{\langle 2\rangle}\right)^2 - l_{11}^{\langle 2\rangle} l_{22}^{\langle 2\rangle} n_1 n_2 - \left(l_1 l_{12} - l_2 l_{11}\right)^2 \\
& = l_{11}^{\langle 2\rangle}\left(n_1(l_2 - l_{12})^2 + n_2 l_{12}^2\right) + l_{22}^{\langle 2\rangle}\left[n_1(l_1 - l_{11})^2 + n_2 l_{11}^2\right] + 2l_{12}^{\langle 2\rangle}\left(-n_1 l_1 l_{22} + n_1 l_{11} l_{22} - n_2 l_{11} l_{12}\right) \\
& + n_1 n_2\left(l_{12}^{\langle 2\rangle}\right)^2 - l_{11}^{\langle 2\rangle} l_{22}^{\langle 2\rangle} n_1 n_2 - \left[\left(l_{11} + l_{12}\right) l_{12} - \left(l_{22} + l_{12}\right) l_{11}\right]^2 \\
& = l_{11}^{\langle 2\rangle}\left(n_1 l_{22}^2 + n_2 l_{12}^2\right) + l_{22}^{\langle 2\rangle}\left(n_1 l_{12}^2 + n_2 l_{11}^2\right) + 2l_{12}^{\langle 2\rangle}\left(-n_1 l_{22}(l_1 - l_{11}) - n_2 l_{11} l_{12}\right) \\
& + n_1 n_2\left(l_{12}^{\langle 2\rangle}\right)^2 - l_{11}^{\langle 2\rangle} l_{22}^{\langle 2\rangle} n_1 n_2 - \left({l_{12}}^2 - l_{11} l_{22}\right)^2 \\
& = l_{11}^{\langle 2\rangle}\left(n_1 l_{22}^2 + n_2 l_{12}^2\right) + l_{22}^{\langle 2\rangle}\left(n_2 l_{11}^2 + n_1 l_{12}^2\right) + 2l_{12}^{\langle 2\rangle}\left(-n_1 l_{22} l_{12} - n_2 l_{11} l_{12}\right) \\
& + n_1 n_2\left[\left(l_{12}^{\langle 2\rangle}\right)^2 - l_{11}^{\langle 2\rangle} l_{22}^{\langle 2\rangle}\right] - \left({l_{12}}^2 - l_{11} l_{22}\right)^2 \\
& = l_{11}^{\langle 2\rangle}\left(n_1 l_{22}^2 + n_2 l_{12}^2\right) + l_{22}^{\langle 2\rangle}\left(n_2 l_{11}^2 + n_1 l_{12}^2\right) - 2l_{12} l_{12}^{\langle 2\rangle}\left(n_1 l_{22} + n_2 l_{11}\right) \\
& + n_1 n_2\left[\left(l_{12}^{\langle 2\rangle}\right)^2 - l_{11}^{\langle 2\rangle} l_{22}^{\langle 2\rangle}\right] - \left({l_{12}}^2 - l_{11} l_{22}\right)^2
\end{aligned}
$$

and

$$
\begin{aligned}
D = {} & \left(l_1^{\langle 2\rangle} - l_{12}^{\langle 2\rangle}\right)\left(n_1 l_{22}^2 + n_2 l_{12}^2\right) + \left(l_2^{\langle 2\rangle} - l_{12}^{\langle 2\rangle}\right)\left(n_2 l_{11}^2 + n_1 l_{12}^2\right) - 2l_{12} l_{12}^{\langle 2\rangle}\left(n_1 l_{22} + n_2 l_{11}\right) \\
& + n_1 n_2\left[\left(l_{12}^{\langle 2\rangle}\right)^2 - \left(l_1^{\langle 2\rangle} - l_{12}^{\langle 2\rangle}\right)\left(l_2^{\langle 2\rangle} - l_{12}^{\langle 2\rangle}\right)\right] - \left({l_{12}}^2 - l_{11} l_{22}\right)^2 \\
& = l_1^{\langle 2\rangle}\left(n_1 l_{22}^2 + n_2 l_{12}^2\right) + l_2^{\langle 2\rangle}\left(n_1 l_{12}^2 + n_2 l_{11}^2\right) - l_{12}^{\langle 2\rangle}\left[2n_1 l_{22} l_{12} + 2n_2 l_{11} l_{12} + n_1 l_{22}^2 + n_2 l_{12}^2 + n_1 l_{12}^2\right. \\
& \left.+ n_2 l_{11}^2\right] + n_1 n_2\left\{\left(l_{12}^{\langle 2\rangle}\right)^2 - \left[l_1^{\langle 2\rangle} l_2^{\langle 2\rangle} - l_{12}^{\langle 2\rangle}\left(l_1^{\langle 2\rangle} + l_2^{\langle 2\rangle}\right) + \left(l_{12}^{\langle 2\rangle}\right)^2\right]\right\} - \left({l_{12}}^2 - l_{11} l_{22}\right)^2 \\
& = l_1^{\langle 2\rangle}\left(n_1(l_2 - l_{12})^2 + n_2 l_{12}^2\right) + l_2^{\langle 2\rangle}\left[n_1 l_{12}^2 + n_2(l_1 - l_{12})^2\right] - l_{12}^{\langle 2\rangle}\left[n_1(l_{22} + l_{12})^2 + n_2(l_{11} + l_{12})^2\right] \\
& - n_1 n_2\left[l_1^{\langle 2\rangle} l_2^{\langle 2\rangle} - l_{12}^{\langle 2\rangle}\left(l_1^{\langle 2\rangle} + l_2^{\langle 2\rangle}\right)\right] - \left(l_{12}^2 - (l_1 - l_{12})(l_2 - l_{12})\right)^2 \\
& = l_1^{\langle 2\rangle} n_1 l_2(l_2 - 2l_{12}) + l_2^{\langle 2\rangle} n_2 l_1(l_1 - 2l_{12}) + \left(n_1 + n_2\right) l_{12}^2\left(l_1^{\langle 2\rangle} + l_2^{\langle 2\rangle}\right) \\
& - l_{12}^{\langle 2\rangle}\left(n_1 l_2^2 + n_2 l_1^2\right) - n_1 n_2 l_{12}^{\langle 2\rangle}\left(l_1^{\langle 2\rangle} + l_2^{\langle 2\rangle}\right) - n_1 n_2\left(l_1^{\langle 2\rangle} l_2^{\langle 2\rangle}\right) - \left[\left(l_1 + l_2\right) l_{12} - l_2 l_1\right]^2 \\
& = l_1^{\langle 2\rangle} n_1 l_2(l_2 - 2l_{12}) + l_2^{\langle 2\rangle} n_2 l_1(l_1 - 2l_{12}) - l_{12}^{\langle 2\rangle}\left(n_1 l_2^2 + n_2 l_1^2 - n_1 n_2 c_2\right) + n l_{12}^2 c_2 \\
& - n_1 n_2\left(l_1^{\langle 2\rangle} l_2^{\langle 2\rangle}\right) - \left(c_1 l_{12} - l_2 l_1\right)^2 \\
& = l_1^{\langle 2\rangle} n_1 l_2(l_2 - 2l_{12}) + l_2^{\langle 2\rangle} n_2 l_1(l_1 - 2l_{12}) - l_{12}^{\langle 2\rangle}\left(n_1 l_2^2 + n_2 l_1^2 - n_1 n_2 c_2\right) \\
& - n_1 n_2 l_1^{\langle 2\rangle} l_2^{\langle 2\rangle} - l_{12}^2\left(c_1^2 - n c_2\right) - l_1 l_2\left(l_1 l_2 - 2c_1 l_{12}\right).
\end{aligned}
$$

The optimality criterion on function values of Equations (3.14) and (3.15) respectively are

$$
\begin{aligned}
\phi_1 &= \frac{l_{22}^{\langle 2\rangle} {l_1}^2 - 2l_1 l_2 {l_{12}}^{\langle 2\rangle} + l_{11}^{\langle 2\rangle} {l_2}^2 + n {{l_{12}}^{\langle 2\rangle}}^2 - l_{11}^{\langle 2\rangle} l_{22}^{\langle 2\rangle} n}{D} \\
&= \frac{l_1^2 l_{22}^{\langle 2\rangle} + l_2^2 l_{11}^{\langle 2\rangle} - 2l_1 l_{12}^{\langle 2\rangle} l_2 + n {{l_{12}}^{\langle 2\rangle}}^2 - n\left(l_1^{\langle 2\rangle} - {l_{12}}^{\langle 2\rangle}\right)\left(l_2^{\langle 2\rangle} - l_{12}^{\langle 2\rangle}\right)}{D} \\
&= \frac{\left\{\begin{array}{r} l_1^2 l_2^{\langle 2\rangle} - l_1^2 l_{12}^{\langle 2\rangle} + l_2^2 l_1^{\langle 2\rangle} - l_2^2 l_{12}^{\langle 2\rangle} - 2l_1 l_2 l_{12}^{\langle 2\rangle} + n\Big(l_{12}^{\langle 2\rangle}\Big)^2 \\ -n l_1^{\langle 2\rangle} l_2^{\langle 2\rangle} + n l_1^{\langle 2\rangle} l_{12}^{\langle 2\rangle} + n l_{12}^{\langle 2\rangle} l_2^{\langle 2\rangle} - n\Big(l_{12}^{\langle 2\rangle}\Big)^2 \end{array}\right\}}{D} \\
&= \frac{l_1^2 l_2^{\langle 2\rangle} + l_2^2 l_1^{\langle 2\rangle} - l_{12}^{\langle 2\rangle} \left(l_1 - l_2\right)^2 - n l_1^{\langle 2\rangle} l_2^{\langle 2\rangle} + n l_{12}^{\langle 2\rangle} \left(l_1^{\langle 2\rangle} + l_2^{\langle 2\rangle}\right)}{D} \\
&= \frac{l_1^2 l_2^{\langle 2\rangle} + l_2^2 l_1^{\langle 2\rangle} - l_{12}^{\langle 2\rangle} \left(c_1^2 - n c_2\right) - n l_1^{\langle 2\rangle} l_2^{\langle 2\rangle}}{D}
\end{aligned}
$$

and

$$
\begin{aligned}
\phi_2 &= \frac{\left\{\begin{array}{r} -n_1 {l_1}^2 - 2n_1 l_1 l_2 + 2n_1 l_1 l_{11} + 2n_1 l_1 l_{12} - n_1 {l_2}^2 + 2n_1 l_2 l_{11} + 2n_1 l_2 l_{12} \\ -n {l_{11}}^2 - 2n l_{11} l_{12} - n {l_{12}}^2 + l_{11}^{\langle 2\rangle} n_1 n_2 + l_{12}^{\langle 2\rangle} 2n_1 n_2 + l_{22}^{\langle 2\rangle} n_1 n_2 \end{array}\right\}}{D} \\
&= \frac{\left\{\begin{array}{r} \left(l_{11} + l_{12}\right)^2 n + \big[\left(l_1 + l_2\right)\left(l_1 - 2\left(l_{11} + l_{12}\right) + l_2\right) \\ -\left(l_{11}^{\langle 2\rangle} + 2l_{12}^{\langle 2\rangle} + l_{22}^{\langle 2\rangle}\right) n\big] n_1 + \left(l_{11}^{\langle 2\rangle} + 2l_{12}^{\langle 2\rangle} + l_{22}\right) n_1^2 \end{array}\right\}}{D} \\
&= \frac{l_1^2 n + \left[c_1\left(l_1 - 2l_1 + l_2\right) - \left(l_{11}^{\langle 2\rangle} + l_{12}^{\langle 2\rangle} + l_{22}^{\langle 2\rangle} + l_{12}^{\langle 2\rangle}\right) n\right] n_1 + \left(l_{11}^{\langle 2\rangle} + l_{12}^{\langle 2\rangle} + l_{22}^{\langle 2\rangle} + l_{12}^{\langle 2\rangle}\right) n_1^2}{D} \\
&= \frac{l_1^2 n + \left[c_1\left(l_2 - l_1\right) - \left(l_1^{\langle 2\rangle} + l_2^{\langle 2\rangle}\right) n\right] n_1 + \left(l_1^{\langle 2\rangle} + l_2^{\langle 2\rangle}\right) n_1^2}{D} \\
&= \frac{l_1^2 n + c_1(l_2 - l_1) n_1 - c_2 n n_1 + c_2 n_1^2}{D} = \frac{n_2 l_1^2 + n_1\left[l_1^2 + c_1\left(c_1 - 2l_1\right)\right] - c_2 n_1 n_2}{D} \\
&= \frac{n_2 l_1^2 + n_1\left(l_1 - c_1\right)^2 - c_2 n_1 n_2}{D} = \frac{n_2 l_1^2 + n_1 l_2^2 - n_1 n_2 c_2}{D}
\end{aligned}
$$

where

$$
\begin{aligned}
D = {} & l_1^{\langle 2\rangle} n_1 l_2(l_2 - 2l_{12}) + l_2^{\langle 2\rangle} n_2 l_1(l_1 - 2l_{12}) - l_{12}^{\langle 2\rangle}\left(n_1 l_2^2 + n_2 l_1^2 - n_1 n_2 c_2\right) - n_1 n_2\left(l_1^{\langle 2\rangle} l_2^{\langle 2\rangle}\right) \\
& - l_{12}^2\left(c_1^2 - n c_2\right) - l_1 l_2\left(l_1 l_2 - 2c_1 l_{12}\right).
\end{aligned}
$$

Table of proportions of edges (connecting 1s to 2s) connecting different treatments. The columns two and three give the coordinates for the locations of the plotting symbols of Figure 3.11.

Table A.1: Table for Figure 3.11 (with unique entries)

l_{12}/l	β_{γ_1}	β_{γ_2}	l_{12}/l	β_{γ_1}	β_{γ_2}	l_{12}/l	β_{γ_1}	β_{γ_2}
0.33	0.83	-1.50	0.48	-1.00	-1.33	0.62	-1.50	0.17
0.33	1.50	-0.83	0.48	-0.33	-0.67	0.62	0.17	1.83
0.33	1.83	-0.50	0.48	1.33	1.00	0.62	-0.17	1.50
0.38	1.33	-0.33	0.52	-0.17	0.17	0.62	-1.83	-0.17
0.38	1.00	-0.67	0.52	-1.17	-0.83	0.67	-1.67	0.67
0.38	1.67	0.00	0.52	0.83	1.17	0.67	-1.00	1.33
0.38	0.33	-1.33	0.52	0.50	0.83	0.67	-1.33	1.00
0.38	0.67	-1.00	0.52	-0.50	-0.17	0.67	-0.67	1.67
0.38	2.00	0.33	0.52	0.17	0.50	0.67	-0.33	2.00
0.43	1.17	0.17	0.52	-0.83	-0.50	0.71	-1.17	1.83
0.43	0.50	-0.50	0.52	1.17	1.50	0.71	-1.50	1.50
0.43	-0.17	-1.17	0.57	-0.67	0.33	0.71	-0.83	2.17
0.43	0.17	-0.83	0.57	-1.00	0.00	0.71	-1.83	1.17
0.43	0.83	-0.17	0.57	-1.33	-0.33	0.76	-2.00	1.67
0.43	1.50	0.50	0.57	-0.33	0.67	0.76	-2.33	1.33
0.43	1.83	0.83	0.57	0.00	1.00	0.76	-1.33	2.33
0.48	0.00	-0.33	0.57	0.33	1.33	0.76	-1.67	2.00
0.48	1.00	0.67	0.57	0.67	1.67			
0.48	0.67	0.33	0.62	-0.50	1.17			
0.48	0.33	0.00	0.62	-0.83	0.83			
0.48	-0.67	-1.00	0.62	-1.17	0.50			

The optimal designs for ϕ_1 and ϕ_2 for the example of Figure 3.1 under different misspecifications are provided below.

Table A.2: Efficiencies of the optimal designs under network misspecification

Edge(s) removed between units	Figure	Eff_{ϕ_1}	Eff_{ϕ_2}
1 and 5	(A.1)	0.96 (= 0.3359/0.3490)	0.98 (= 0.0866/0.0876)
1 and 10	(A.2)	0.98 (= 0.3359/0.3406)	0.88 (= 0.0866/0.0984)
4 and 10	(A.3)	1 (= 0.3359/0.3370)	0.99 (= 0.0866/0.0875)
1 and 5 and 1 and 10	(A.7)	0.99 (= 0.3359/0.3368)	0.87 (= 0.0866/0.0994)
1 and 5, 1 and 10 and 4 and 10	(A.8)	0.96 (= 0.3359/0.3478)	0.84 (= 0.0866/0.1023)
Edge(s) included between units			
1 and 4	(A.4)	0.98 (= 0.3359/0.3417)	0.99 (= 0.0866/0.0875)
1 and 12	(A.5)	0.98 (= 0.3359/0.3406)	0.90 (= 0.0866/0.0952)
7 and 9	(A.6)	0.98 (= 0.3359/0.3417)	1 (= 0.0866/0.0866)

Figure A.1: Optimal designs when removing an edge between units 1 and 5; for ϕ_1 (*left*) and ϕ_2 (*right*)

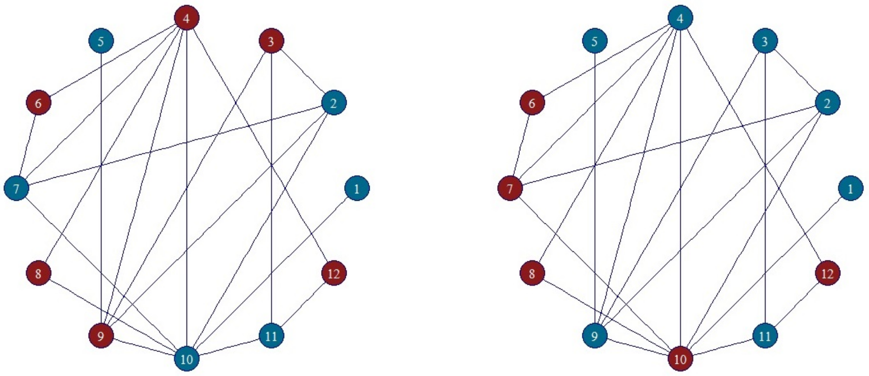


Figure A.2: Optimal designs when removing an edge between units 1 and 10; for ϕ_1 (*left*) and ϕ_2 (*right*)

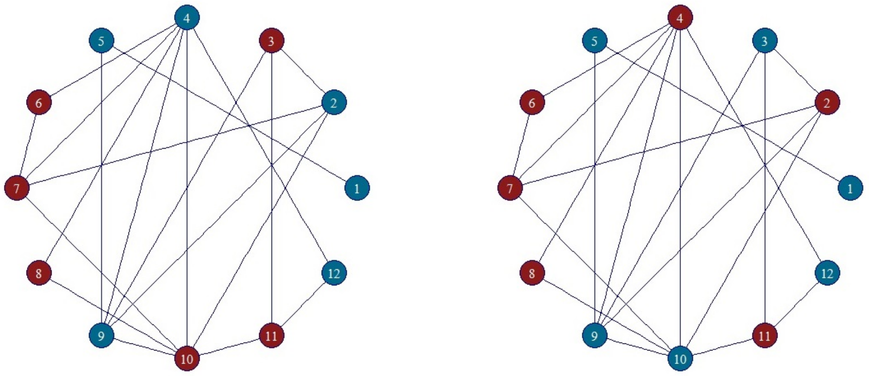


Figure A.3: Optimal designs when removing an edge between units 4 and 10; for ϕ_1 (*left*) and ϕ_2 (*right*)

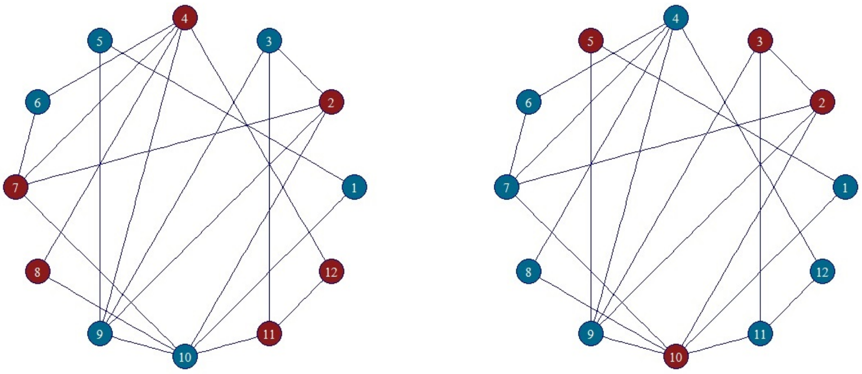


Figure A.4: Optimal designs when adding an edge between units 1 and 4; for ϕ_1 (*left*) and ϕ_2 (*right*)

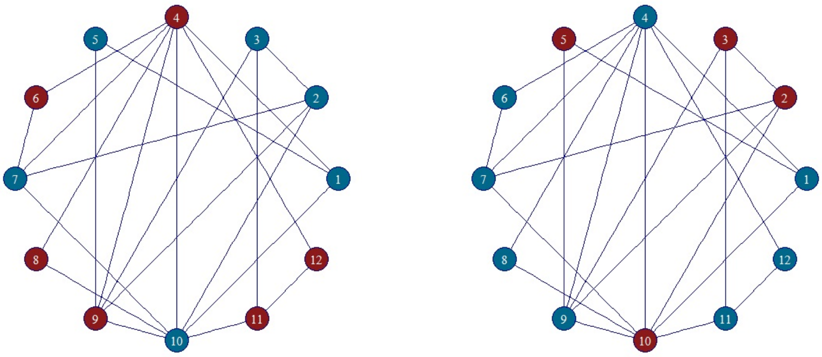


Figure A.5: Optimal designs when adding an edge between units 1 and 12; for ϕ_1 (*left*) and ϕ_2 (*right*)

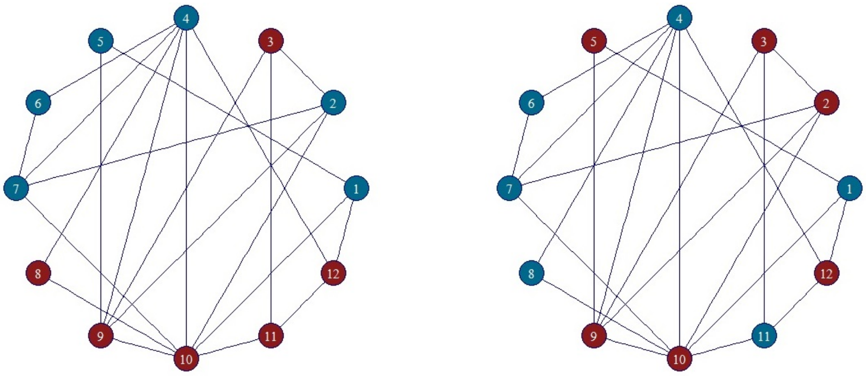


Figure A.6: Optimal designs when adding an edge between units 7 and 9; for ϕ_1 (*left*) and ϕ_2 (*right*)

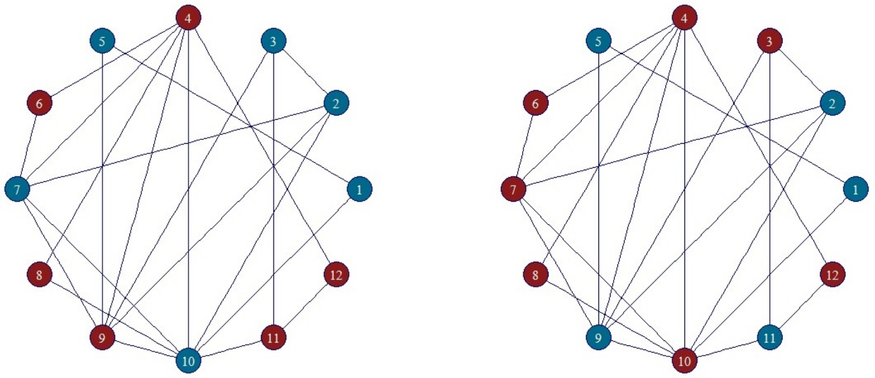


Figure A.7: Optimal designs when removing edges between units 1 and 5 and 1 and 10; for ϕ_1 (*left*) and ϕ_2 (*right*)

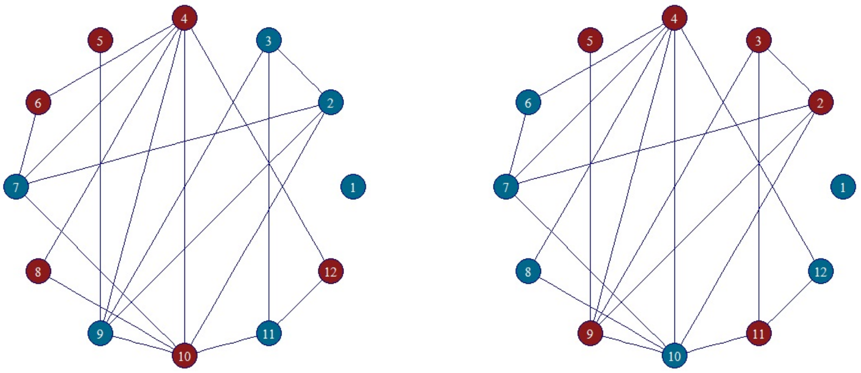
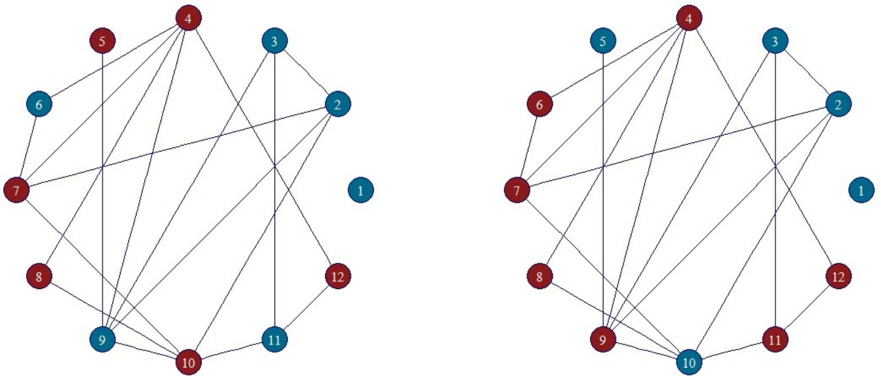


Figure A.8: Optimal designs when removing edges between units 1 and 5, 1 and 10 and 4 and 10; for ϕ_1 (*left*) and ϕ_2 (*right*)



B Results for Chapter 5

For the optimal block designs for Example 5.3.1 we have 161 and 163 units receiving treatments 1 and 2 respectively for ϕ_1 units, and 117 and 207 units receiving treatments 1 and 2 respectively for ϕ_2 units. The optimal allocations are provided in Table B.1.

Table B.1: Near-optimal designs under NBM

Optimal allocation for ϕ_1											Optimal allocation for ϕ_2										
2	2	1	2	2	1	1	1	1	1	2	2	2	1	2	2	2	2	2	2	1	2
1	2	1	1	2	2	1	1	1	1	2	2	2	2	2	1	2	2	1	1	2	2
2	2	1	2	2	1	1	1	1	1	1	1	2	2	1	1	1	2	2	2	1	1
2	1	1	1	2	1	2	2	2	2	2	1	2	1	1	1	1	2	2	2	2	2
1	1	1	1	2	1	1	1	1	2	1	2	2	2	2	2	2	2	2	2	1	2
2	2	2	2	2	2	2	2	1	1	2	2	2	1	2	2	2	1	2	1	1	2
1	2	2	2	2	1	2	1	2	2	2	2	1	2	2	1	1	1	1	2	2	2
2	2	2	1	2	1	1	2	2	2	2	1	2	1	1	1	1	2	2	2	2	2
2	2	1	2	1	1	2	1	1	1	2	2	1	2	1	2	2	2	2	1	2	1
2	2	2	1	1	2	1	2	1	1	2	2	2	1	2	1	1	1	2	2	2	2
2	1	2	2	2	2	1	1	1	1	1	1	1	2	2	1	2	2	1	2	2	2
2	2	1	2	1	1	2	1	1	2	1	2	2	2	1	1	1	1	1	1	1	1
2	1	2	1	1	2	1	1	2	2	2	1	2	1	1	2	2	2	2	2	2	2
1	2	2	1	1	1	2	1	1	2	2	2	1	2	2	2	1	2	2	2	2	2
2	1	2	2	1	1	1	2	1	2	1	2	2	1	2	2	2	2	2	1	1	1
1	1	1	2	2	1	1	2	1	1	2	2	2	1	2	1	2	1	2	1	2	2
1	1	2	2	2	1	2	2	1	2	2	1	1	2	1	2	2	1	2	1	1	2
2	2	1	2	2	2	1	1	1	2	2	2	1	2	1	2	1	2	2	1	1	1
2	2	1	1	1	2	1	1	2	1	1	1	2	2	2	2	1	2	2	2	1	1
1	1	1	2	2	2	1	1	1	2	1	2	2	1	2	2	2	2	2	2	1	2
1	1	2	1	2	1	2	2	2	2	2	1	1	1	1	2	2	2	2	2	1	1
2	1	1	1	1	2	1	2	1	2	1	1	2	2	1	1	2	2	1	2	1	2
1	1	1	1	2	1	2	2	1	1	2	2	2	2	2	2	1	2	2	2	2	1
2	2	1	2	1	1	2	2	1	2	2	2	1	2	2	1	2	2	1	2	2	2
2	1	2	2	2	1	2	2	2	1		1	1	2	2	1	2	2	2	1	2	
2	1	1	1	2	1	1	1	2	2		2	2	2	2	1	2	1	2	1	1	
2	2	1	1	2	1	2	2	1	2		2	1	2	1	2	2	1	2	2	1	
1	2	2	1	2	1	1	1	2	1		2	1	1	2	2	2	2	2	2	2	
2	1	2	2	2	1	1	2	1	1		2	1	2	2	1	2	2	1	1	2	
1	2	2	1	1	2	2	1	2	1		2	2	2	2	2	2	2	1	2	2	

For the $m = 2$ treatment case and $\kappa = 3$ we can write the bias in terms of unknown parameters $\boldsymbol{\beta}$, under the assumption there are network effects and we do not account for them (i.e. RBM), as

$$\begin{pmatrix} 0 & 0 & 0 & 0 & I_1 & I_2 \\ 0 & 0 & 0 & 0 & I_3 & I_4 \\ 0 & 0 & 0 & 0 & I_5 & I_6 \\ 0 & 0 & 0 & 0 & I_7 & I_8 \\ 0 & 0 & 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 0 & 0 & -1 \end{pmatrix} \begin{pmatrix} \mu \\ \tau_1 \\ b_1 \\ b_2 \\ \gamma_1 \\ \gamma_2 \end{pmatrix}$$

where

$$\begin{aligned}
I_1 = {} & D(n_{(2)1}^2 n_{(1)}(l_1 - l_{(1)1}) + n_{(1)1}^2 n_{(2)}(l_1 - l_{(2)1}) + n_{(1)}n_{(2)}\big[n_1(l_{11} - l_1) - (n_{(1)1} + n_{(2)1})l_{11} \\
& + n_1(l_{(1)1} + l_{(2)1})\big] + n_{(1)1}(n_{(2)1} - n_1)n_{(2)}l_{(1)1} + n_{(2)1}n_{(1)}(n_{(1)1} - n_1)l_{(2)1}) \\
I_2 = {} & D(n_{(2)1}^2 n_{(1)}(l_2 - l_{(1)2}) + n_{(1)1}^2 n_{(2)}(l_2 - l_{(2)2}) + n_{(1)}n_{(2)}\big[n_1(l_{12} - l_2) - (n_{(1)1} + n_{(2)1})l_{12} \\
& + n_1(l_{(1)2} + l_{(2)2})\big] + n_{(1)1}(n_{(2)1} - n_1)n_{(2)}l_{(1)2} + n_{(2)1}n_{(1)}(n_{(1)1} - n_1)l_{(2)2}) \\
I_3 = {} & D(n_{(1)}n_{(2)}(n_{(1)} + n_{(2)})l_{11} + n_{(1)1}n_{(2)}(n - n_{(2)})l_{(1)1} + n_{(2)1}n_{(1)}(n - n_{(1)})l_{(2)1} \\
& + n_{(1)}n_{(2)}\ \big[n_1(l_1 - (l_{(1)1} + l_{(2)1})) + l_{(2)1}n_{(1)1} - l_1 n_{(1)1} + l_{(1)1}n_{(2)1} - l_1 n_{(2)1} - l_{11}n\big]) \\
I_4 = {} & D(n_{(1)}n_{(2)}(n_{(1)} + n_{(2)})l_{12} + n_{(1)1}n_{(2)}(n - n_{(2)})l_{(1)2} + n_{(2)1}n_{(1)}(n - n_{(1)})l_{(2)2} \\
& + n_{(1)}n_{(2)}\ \big[n_1(l_2 - (l_{(1)2} + l_{(2)2})) + l_{(2)2}n_{(1)1} - l_2 n_{(1)1} + l_{(1)2}n_{(2)1} - l_2 n_{(2)1} - l_{12}n\big]) \\
I_5 = {} & D(l_{(1)1}n_1 n_{(2)}(n_{(2)} + n_1 - n - 2n_{(2)1}) + n_{(1)1}n_{(2)}(l_{11}n + l_1 n_{(2)1} - l_1 n_1 - l_{11}n_{(2)} + l_{(2)1}n_1) \\
& + n_{(2)1}l_{(2)1}(n_1 n_{(1)} - nn_{(1)1}) + n_{(2)1}^2(l_{(1)1}n - n_{(1)}l_1) + n_{(1)}n_{(2)}(l_1 n_1 - l_{11}n_1 + l_{11}n_{(2)1} - l_{(2)1}n_1)) \\
I_6 = {} & D(l_{(1)2}n_1 n_{(2)}(n_{(2)} + n_1 - n - 2n_{(2)1}) + n_{(1)1}n_{(2)}(l_{12}n + l_2 n_{(2)1} - l_2 n_1 - l_{12}n_{(2)} + l_{(2)2}n_1) \\
& + n_{(2)1}l_{(2)2}(n_1 n_{(1)} - nn_{(1)1}) + n_{(2)1}^2(l_{(1)2}n - n_{(1)}l_2) + n_{(1)}n_{(2)}(l_2 n_1 - l_{12}n_1 + l_{12}n_{(2)1} - l_{(2)2}n_1)) \\
I_7 = {} & D(l_{(2)1}nn_{(1)1}^2 - l_1 n_{(1)1}^2 n_{(2)} + n_{(1)}n_{(2)}(l_1 n_1 - l_{11}n_1 + l_{11}n_{(1)1} - l_{(1)1}n_1) \\
& + n_{(1)}l_{(2)1}n_1(n_1 + n_{(1)} - n - 2n_{(1)1}) + n_{(1)}(l_1 n_{(1)1}n_{(2)1} - l_{11}n_{(2)1}n_{(1)} - l_1 n_1 n_{(2)1} + l_{11}nn_{(2)1}) \\
& + n_{(1)}l_{(1)1}n_1 n_{(2)1} + l_{(1)1}n_1 n_{(1)1}n_{(2)} - l_{(1)1}nn_{(1)1}n_{(2)1})) \\
I_8 = {} & D(l_{(2)2}nn_{(1)1}^2 - l_2 n_{(1)1}^2 n_{(2)} + n_{(1)}n_{(2)}(l_2 n_1 - l_{12}n_1 + l_{12}n_{(1)1} - l_{(1)2}n_1) \\
& + n_{(1)}l_{(2)2}n_1(n_1 + n_{(1)} - n - 2n_{(1)1}) + n_{(1)}(l_2 n_{(1)1}n_{(2)1} - l_{12}n_{(2)1}n_{(1)} - l_1 n_1 n_{(2)1} + l_{12}nn_{(2)1}) \\
& + n_{(1)}l_{(1)2}n_1 n_{(2)1} + l_{(1)2}n_1 n_{(1)1}n_{(2)} - l_{(1)2}nn_{(1)1}n_{(2)1})
\end{aligned}$$

and where

$$D = \frac{1}{\begin{array}{l} n_{(1)}n_{(2)}n_1(n_1 - 2n_{(1)1} - 2n_{(2)1} + n_{(1)} + n_{(2)} - n) + n_{(1)1}n_{(2)}(-n_{(1)1}n_{(2)} + nn_{(1)1}) \\ \quad + n_{(2)1}n_{(1)}(-n_{(2)1}n_{(1)} + nn_{(2)1}) + n_{(1)}n_{(2)}2n_{(1)1}n_{(2)1} \end{array}}.$$

We conjecture that for κ blocks and two treatments the bias in τ_1 from γ_1 will be

$$\mathrm{Bias}(\hat{\tau}_1) = \beta_{\gamma_1}\gamma_1 + \beta_{\gamma_2}\gamma_2 = \frac{\Gamma_1}{D}\gamma_1 + \frac{\Gamma_2}{D}\gamma_2,$$

where

$$\begin{aligned}
\Gamma_1 = {} & \sum_{i=1}^{\kappa-1}\Bigg[\left(n - \sum_{q=1,q\neq i}^{\kappa-1} n_{(q)}\right) l_{(i)1} n_{(i)1} \prod_{q=1,q\neq i}^{\kappa-1} n_{(q)}\Bigg] + \prod_{i=1}^{\kappa-1} n_{(i)}\Bigg[n_1 l_1 - n l_{11} \\
& - \sum_{i=1}^{\kappa-1} n_{(i)1} l_1 + \sum_{i=1}^{\kappa-1} n_{(i)} l_{11} - \sum_{i=1}^{\kappa-1} n_1 l_{(i)1} + \sum_{i=1}^{\kappa-1} n_{(i)1} \sum_{q=1,q\neq i}^{\kappa-1} l_{(q)1}\Bigg] \\
D = {} & \sum_{i=1}^{\kappa-1}\Bigg[\left(n - \sum_{q=1,q\neq i}^{\kappa-1} n_{(q)}\right) {n_{(i)1}}^2 \prod_{q=1,q\neq i}^{\kappa-1} n_{(q)}\Bigg] + n_1 \prod_{i=1}^{\kappa-1} n_{(i)}\Bigg[n_1 - 2\sum_{i=1}^{\kappa-1} n_{(i)1} \\
& - \sum_{i=1}^{\kappa-1}\sum_{q=1,q\neq i}^{\kappa-1} n_{(i)1} n_{q1} + \sum_{i=1}^{\kappa-1} n_{(i)} - n\Bigg].
\end{aligned}$$

This holds for

$$\begin{aligned}
\kappa = 2:\ \Gamma_1 &= n n_{(1)1} l_{(1)1} + n_{(1)}\left(n_1 l_1 - n l_{11} - n_{(1)1} l_1 + n_{(1)} l_{11} - n_1 l_{(1)1}\right) \\
D &= n {n_{(1)1}}^2 + n_1 n_{(1)}\left(n_1 - 2n_{(1)1} + n_{(1)} - n\right);
\end{aligned}$$

$$\begin{aligned}
\kappa = 3:\ \Gamma_1 = {} & \left(n - n_{(2)}\right) n_{(1)1} l_{(1)1} n_{(2)} + \left(n - n_{(1)}\right) n_{(2)1} l_{(2)1} n_{(1)} + n_{(1)} n_{(2)}\big[n_1 l_1 - n l_{11} \\
& - \left(n_{(1)1} + n_{(2)1}\right) l_1 + \left(n_{(1)} + n_{(2)}\right) l_{11} - n_1\left(l_{(1)1} + l_{(2)1}\right) + l_{(1)1} n_{(2)1} + l_{(2)1} n_{(1)1}\big] \\
D = {} & \left(n - n_{(2)}\right) {n_{(1)1}}^2 n_{(2)} + \left(n - n_{(1)}\right) {n_{(2)1}}^2 n_{(1)} + n_1 n_{(1)} n_{(2)}\big[n_1 - 2\,\big(n_{(1)1} \\
& + n_{(2)1} - n_{(1)1} n_{(2)1}/n_1 + \left(n_{(1)} + n_{(2)}\right) - n\big];
\end{aligned}$$

$$\begin{aligned}
\kappa = 4:\ \Gamma_1 = {} & \left(n - n_{(3)} - n_{(2)}\right) n_{(1)1} l_{(1)1} n_{(2)} n_{(3)} + \left(n - n_{(1)} - n_{(3)}\right) n_{(2)1} l_{(2)1} n_{(1)} n_{(3)} \\
& + \left(n - n_{(1)} - n_{(2)}\right) n_{31} l_{(3)1} n_{(1)} n_{(2)} + n_{(1)} n_{(2)} n_{(3)}\big[n_1 l_1 - n l_{11} \\
& - \left(n_{(1)1} + n_{(2)1} + n_{31}\right) l_1 + \left(n_{(1)} + n_{(2)} + n_{(3)}\right) l_{11} - n_1\left(l_{(1)1} + l_{(2)1} + l_{(3)1}\right) \\
& + \left(l_{(1)1} + l_{(3)1}\right) n_{(2)1} + \left(l_{(2)1} + l_{(3)1}\right) n_{(1)1} + \left(l_{(1)1} + l_{(2)1}\right) n_{31}\big] \\
D = {} & \left(n - n_{(2)} - n_{(3)}\right) {n_{(1)1}}^2 n_{(2)} n_{(3)} + \left(n - n_{(1)} - n_{(3)}\right) {n_{(2)1}}^2 n_{(1)} n_{(3)} \\
& + \left(n - n_{(1)} - n_{(2)}\right) {n_{31}}^2 n_{(1)} n_{(2)} + n_1 n_{(1)} n_{(2)} n_{(3)}\big[n_1 \\
& - 2\left(n_{(1)1} + n_{(2)1} + n_{31} - \left(n_{(1)1} n_{(2)1} + n_{(1)1} n_{31} + n_{31} n_{(2)1}\right)/n_1\right) \\
& + \left(n_{(1)} + n_{(2)} + n_{(3)}\right) - n\big].
\end{aligned}$$

Likewise, by replacing treatment 1 to 2 in the expression of β_{γ_1} for the links, we conjecture that the bias in τ_1 from γ_2 will be

$$\begin{aligned}
\Gamma_2 = {} & \sum_{i=1}^{\kappa-1}\big[\left(n - \sum_{q=1,q\neq i}^{\kappa-1} n_{(q)}\right) l_{(i)2} n_{(1)2} \prod_{q=1,q\neq i}^{\kappa-1} n_{(q)}\big] + \prod_{i=1}^{\kappa-1} n_{(i)}\big[n_1 l_1 - n l_{12} \\
& - \sum_{i=1}^{\kappa-1} n_{(i)1} l_2 + \sum_{i=1}^{\kappa-1} n_{(i)} l_{11} - \sum_{i=1}^{\kappa-1} n_1 l_{(i)2} + \sum_{i=1}^{\kappa-1} n_{(i)1} \sum_{q=1,q\neq i}^{\kappa-1} l_{(q)2}\big].
\end{aligned}$$

We present all optimal LNDs and NBDs for ϕ_1 in Figures B.1 and B.2 respectively for the Example 5.4.1. The different optimal arrangements are related to the ϕ_1-automorphisms.

Figure B.1: All LNDs

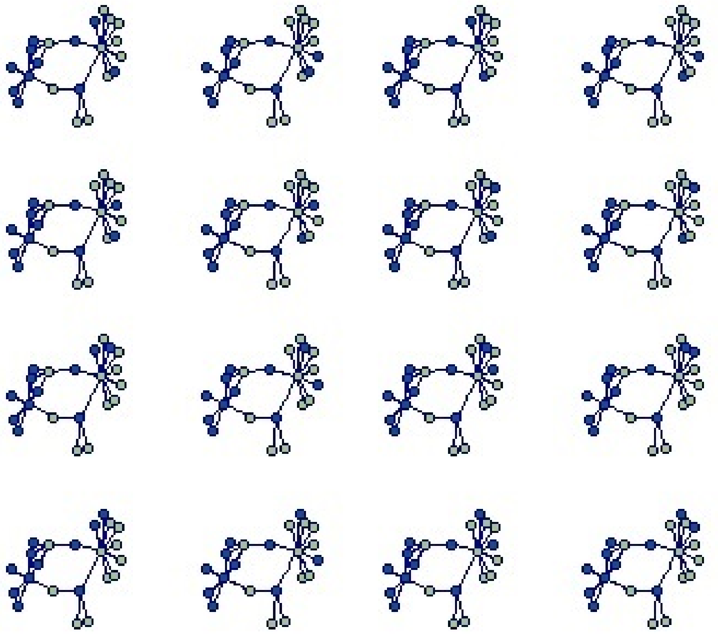


Figure B.2: All NBDs

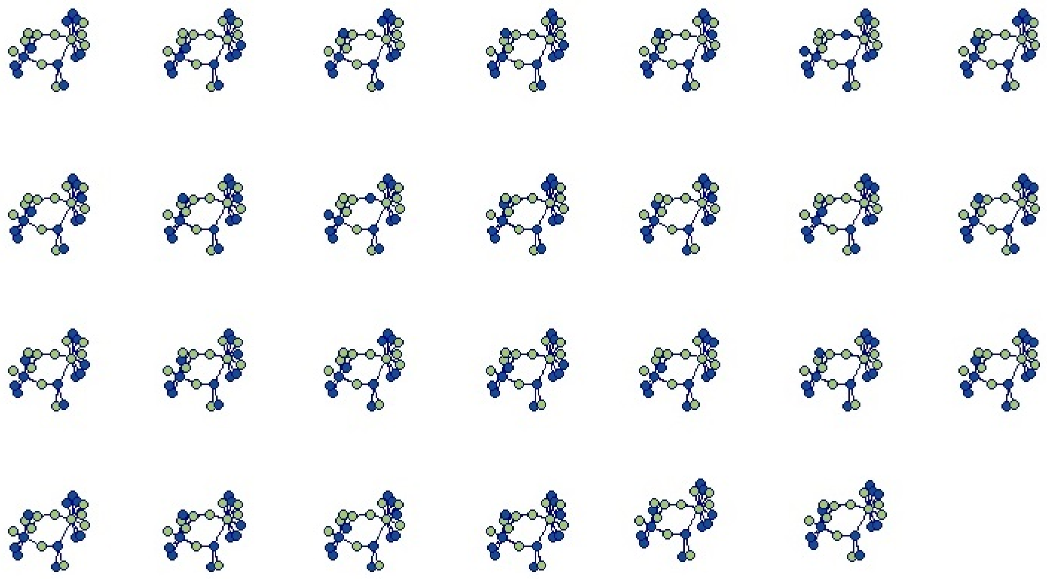


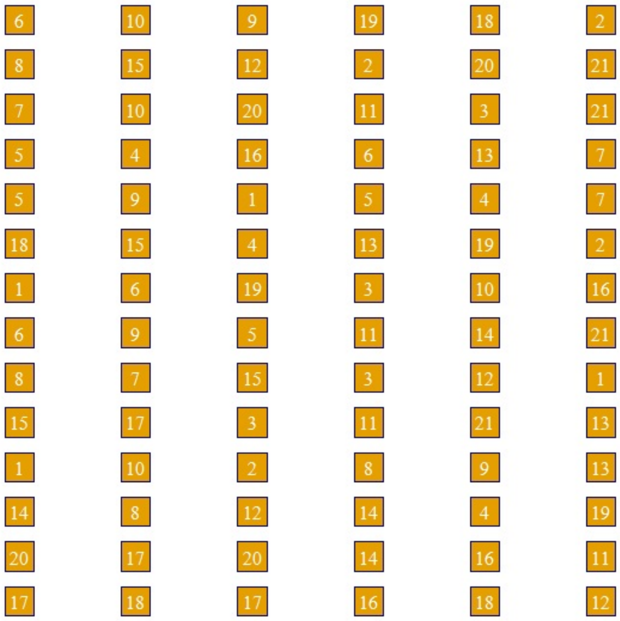
Table of proportions of edges (connecting 1s to 2s) connecting different treatments. The columns two and three give the coordinates for the locations of the plotting symbols of Figure 5.13.

Table B.2: Table for Figure 5.13 (with unique entries)

l_{12}/l	β_{γ_1}	β_{γ_2}
0.37	0	-1.27
0.37	-0.18	-1.45
0.37	0.18	-1.09
0.37	0.36	-0.91
0.41	-0.09	-1
0.41	-0.27	-1.18
0.41	-0.45	-1.36
0.41	0.09	-0.82
0.41	0.27	-0.64
0.41	0.45	-0.45
0.44	-0.73	-1.27
0.44	-0.36	-0.91
0.44	-0.55	-1.09
0.44	-0.18	-0.73
0.44	0	-0.55
0.44	0.18	-0.36
0.44	0.36	-0.18
0.48	-0.82	-1
0.48	-1	-1.18
0.48	-0.64	-0.82
0.48	-0.45	-0.64
0.48	-0.27	-0.45
0.48	-0.09	-0.27
0.48	0.09	-0.09
0.48	0.27	0.09
0.48	0.45	0.27
0.52	-1.09	-0.91
0.52	-0.91	-0.73
0.52	-0.73	-0.55
0.52	-0.55	-0.36
0.52	-0.36	-0.18
0.52	-0.18	0
0.52	0	0.18
0.52	0.18	0.36
0.56	-1.18	-0.64
0.56	-1.36	-0.82
0.56	-1	-0.45
0.56	-0.64	-0.09
0.56	-0.82	-0.27
0.56	-0.45	0.09
0.56	-0.27	0.27
0.56	-0.09	0.45
0.56	0.09	0.64
0.59	-1.27	-0.36
0.59	-1.45	-0.55
0.59	-0.91	0
0.59	-0.73	0.18
0.59	-1.09	-0.18
0.59	-0.55	0.36
0.59	-0.36	0.55
0.59	-0.18	0.73
0.63	-1.55	-0.27
0.63	-1.36	-0.09
0.63	-1	0.27
0.63	-1.18	0.09
0.63	-0.82	0.45
0.63	-0.64	0.64
0.63	-0.45	0.82
0.67	-1.64	0
0.67	-1.27	0.36
0.67	-1.45	0.18
0.67	-0.91	0.73
0.67	-1.09	0.55
0.67	-0.73	0.91
0.7	-1.55	0.45
0.7	-1.18	0.82
0.7	-1.73	0.27
0.7	-1.36	0.64
0.7	-1	1
0.74	-1.64	0.73
0.74	-1.27	1.09

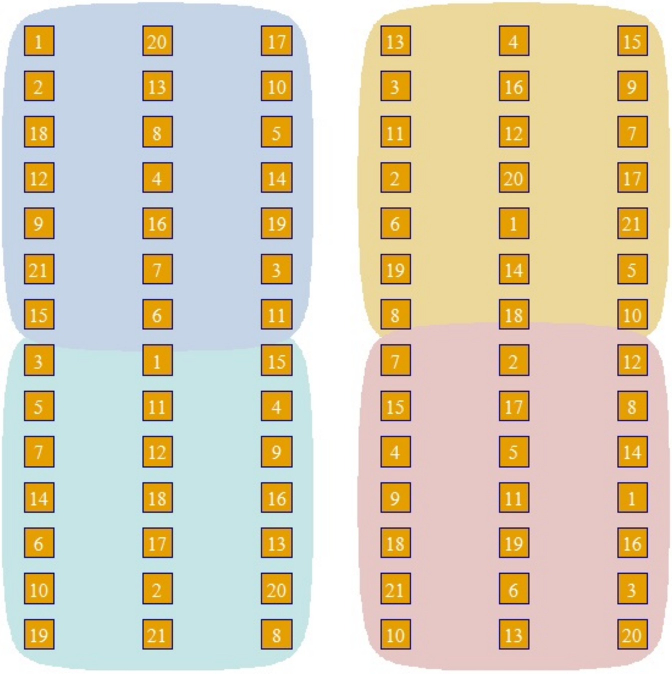
C Optimal designs from Chapter 6

Figure C.1: An optimal CRD



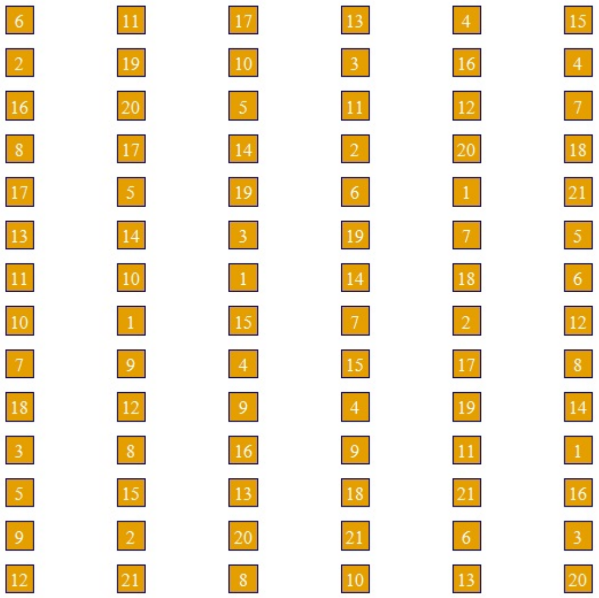
6	10	9	19	18	2
8	15	12	2	20	21
7	10	20	11	3	21
5	4	16	6	13	7
5	9	1	5	4	7
18	15	4	13	19	2
1	6	19	3	10	16
6	9	5	11	14	21
8	7	15	3	12	1
15	17	3	11	21	13
1	10	2	8	9	13
14	8	12	14	4	19
20	17	20	14	16	11
17	18	17	16	18	12

Figure C.2: An optimal RBD



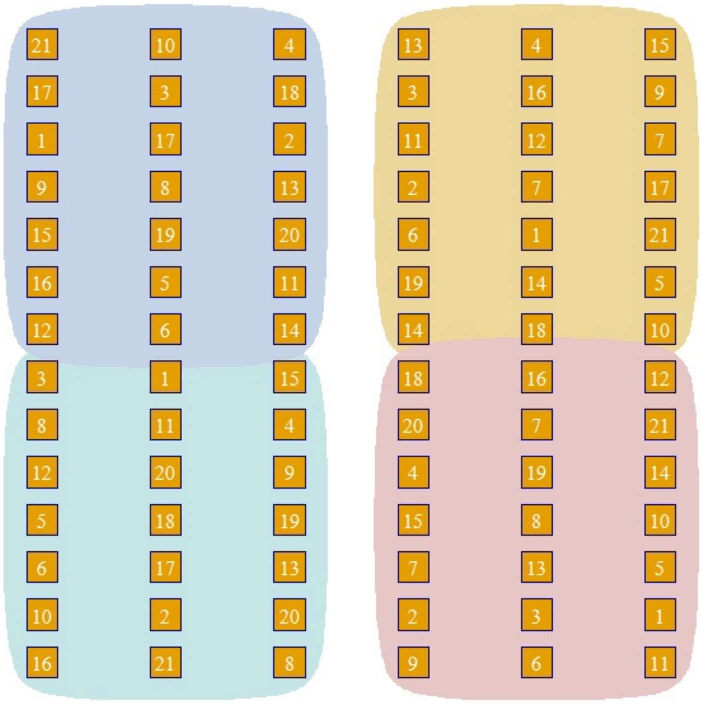
1	20	17	13	4	15
2	13	10	3	16	9
18	8	5	11	12	7
12	4	14	2	20	17
9	16	19	6	1	21
21	7	3	19	14	5
15	6	11	8	18	10
3	1	15	7	2	12
5	11	4	15	17	8
7	12	9	4	5	14
14	18	16	9	11	1
6	17	13	18	19	16
10	2	20	21	6	3
19	21	8	10	13	20

Figure C.3: An optimal RCD



6	11	17	13	4	15
2	19	10	3	16	4
16	20	5	11	12	7
8	17	14	2	20	18
17	5	19	6	1	21
13	14	3	19	7	5
11	10	1	14	18	6
10	1	15	7	2	12
7	9	4	15	17	8
18	12	9	4	19	14
3	8	16	9	11	1
5	15	13	18	21	16
9	2	20	21	6	3
12	21	8	10	13	20

Figure C.4: An optimal RCBD



21	10	4	13	4	15
17	3	18	3	16	9
1	17	2	11	12	7
9	8	13	2	7	17
15	19	20	6	1	21
16	5	11	19	14	5
12	6	14	14	18	10
3	1	15	18	16	12
8	11	4	20	7	21
12	20	9	4	19	14
5	18	19	15	8	10
6	17	13	7	13	5
10	2	20	2	3	1
16	21	8	9	6	11

Figure C.5: A near-optimal LND

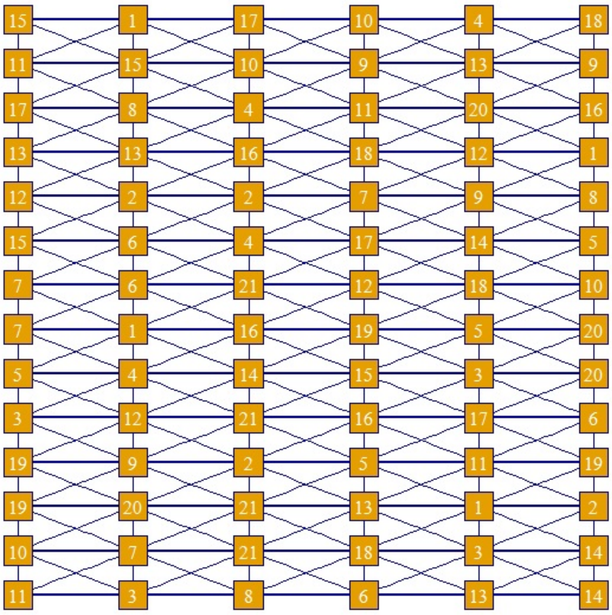


Figure C.6: A near-optimal NBD

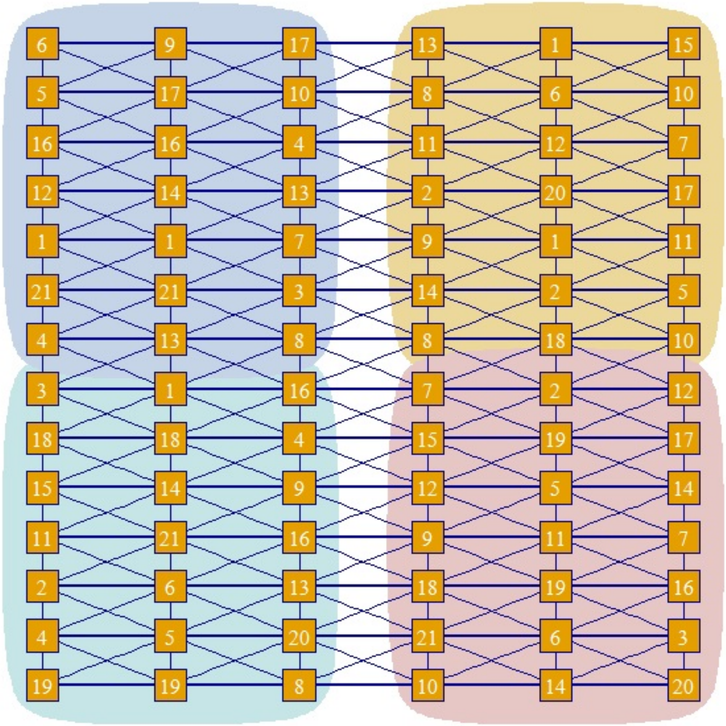


Figure C.7: A near-optimal RCND1

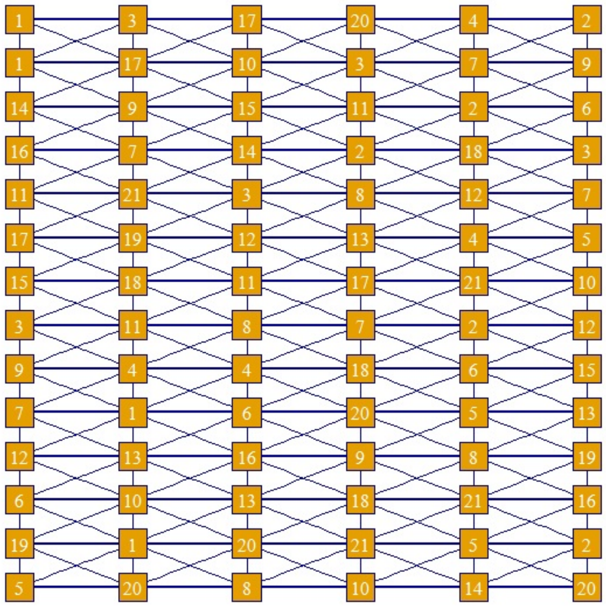


Figure C.8: A near-optimal RCNBD1

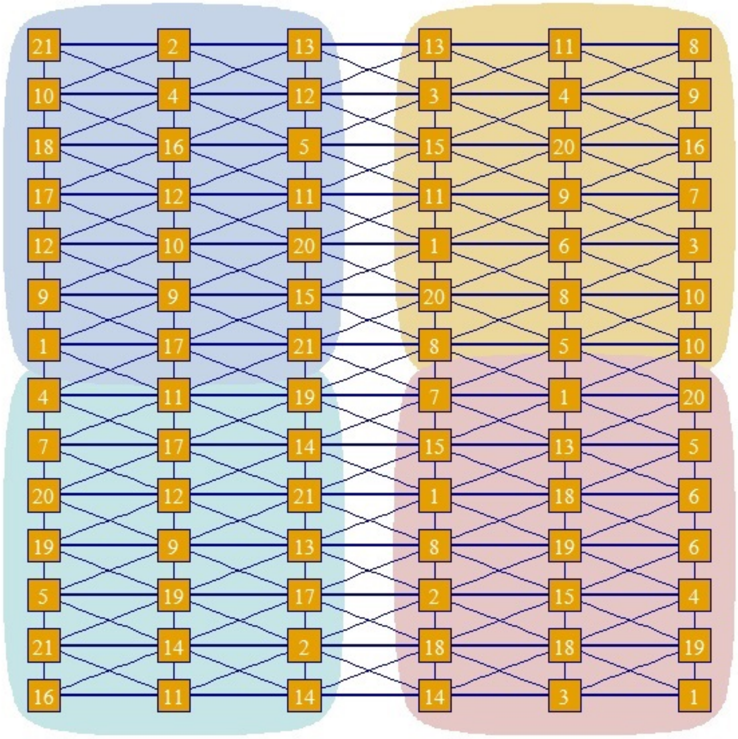


Figure C.9: A near-optimal LND2

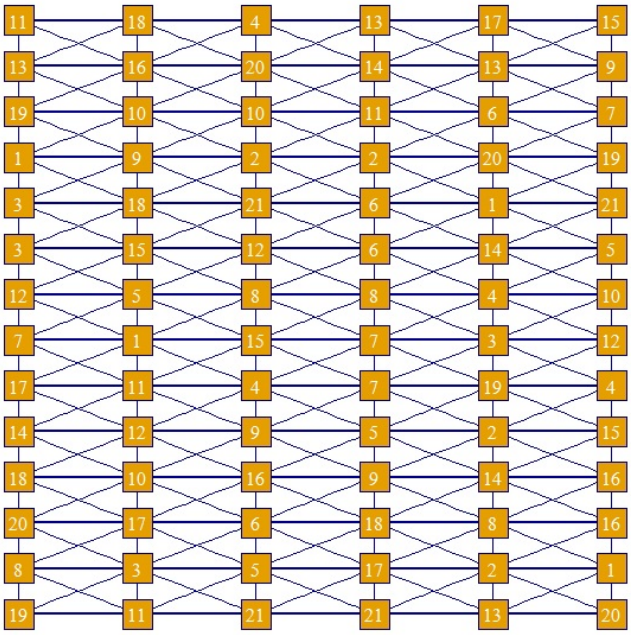


Figure C.10: A near-optimal NBD2

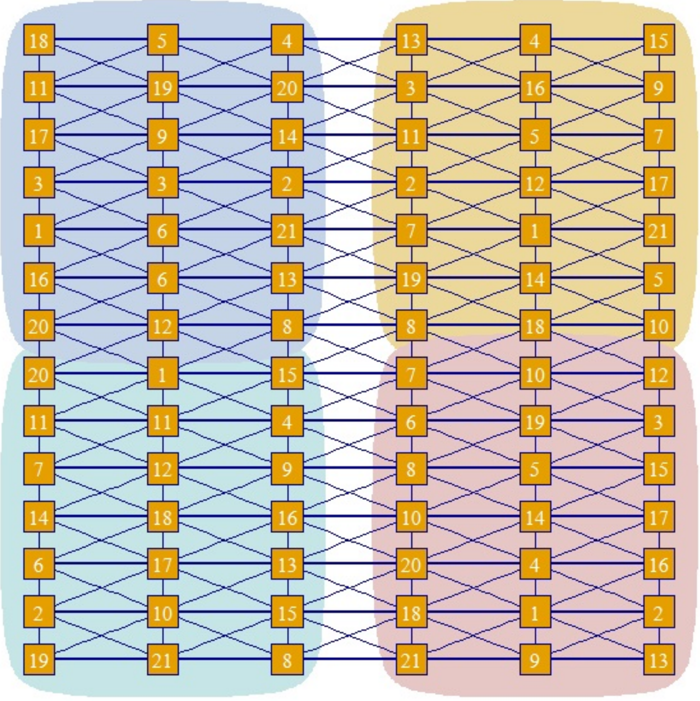


Figure C.11: A near-optimal RCND2

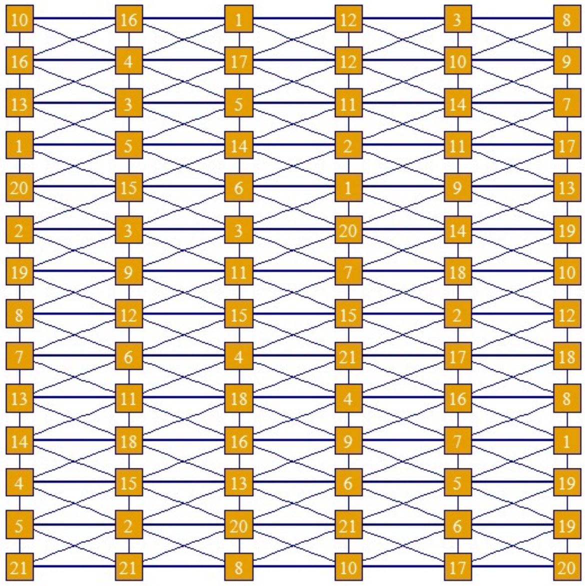


Figure C.12: A near-optimal RCNBD2

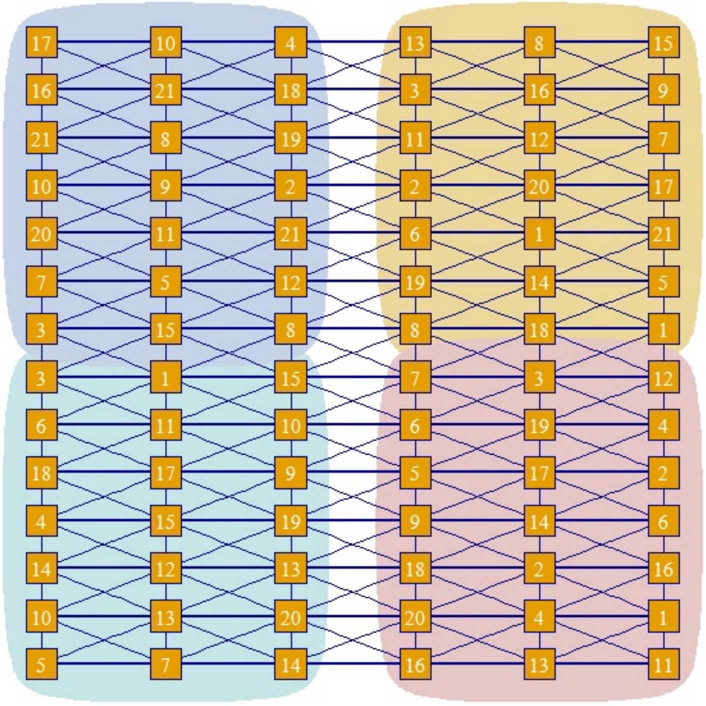
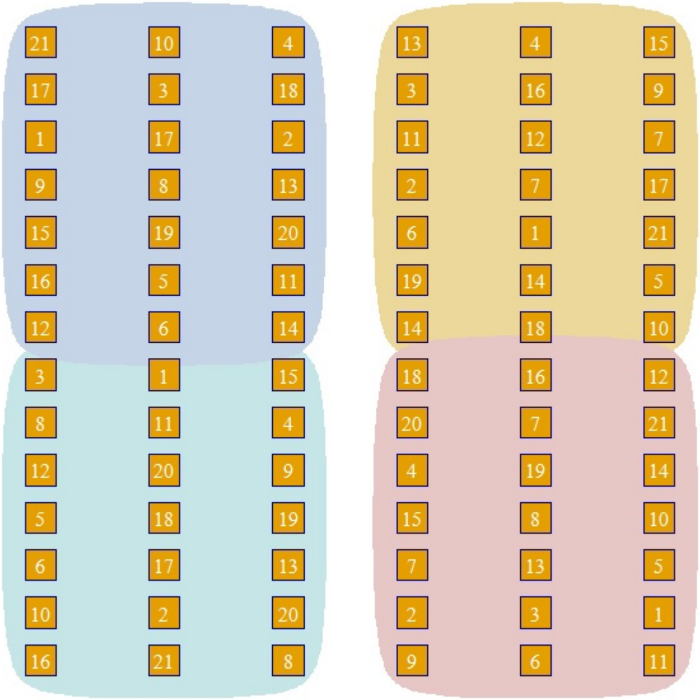
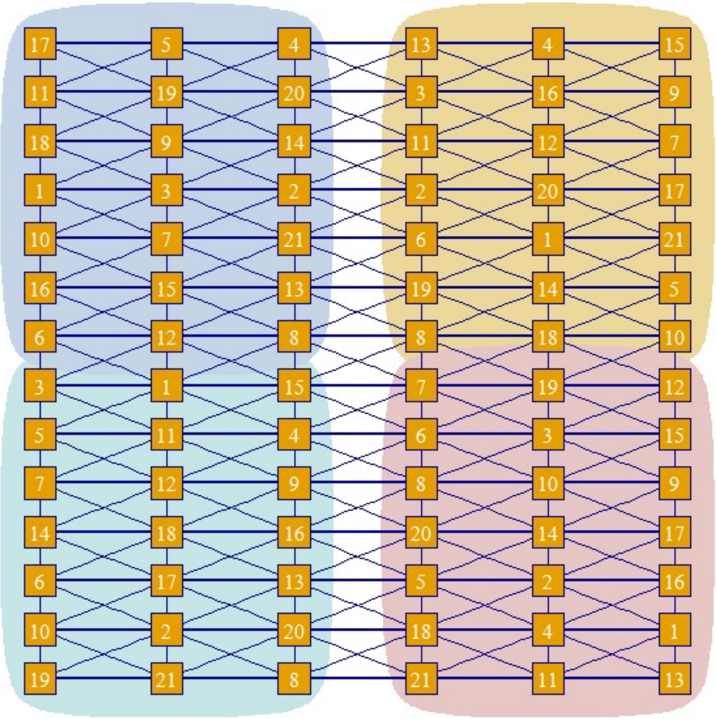


Figure C.13: An optimal RCBD3



21	10	4	13	4	15
17	3	18	3	16	9
1	17	2	11	12	7
9	8	13	2	7	17
15	19	20	6	1	21
16	5	11	19	14	5
12	6	14	14	18	10
3	1	15	18	16	12
8	11	4	20	7	21
12	20	9	4	19	14
5	18	19	15	8	10
6	17	13	7	13	5
10	2	20	2	3	1
16	21	8	9	6	11

Figure C.14: A near-optimal NBD3



17	5	4	13	4	15
11	19	20	3	16	9
18	9	14	11	12	7
1	3	2	2	20	17
10	7	21	6	1	21
16	15	13	19	14	5
6	12	8	8	18	10
3	1	15	7	19	12
5	11	4	6	3	15
7	12	9	8	10	9
14	18	16	20	14	17
6	17	13	5	2	16
10	2	20	18	4	1
19	21	8	21	11	13

Figure C.15: A near-optimal RCNBD3

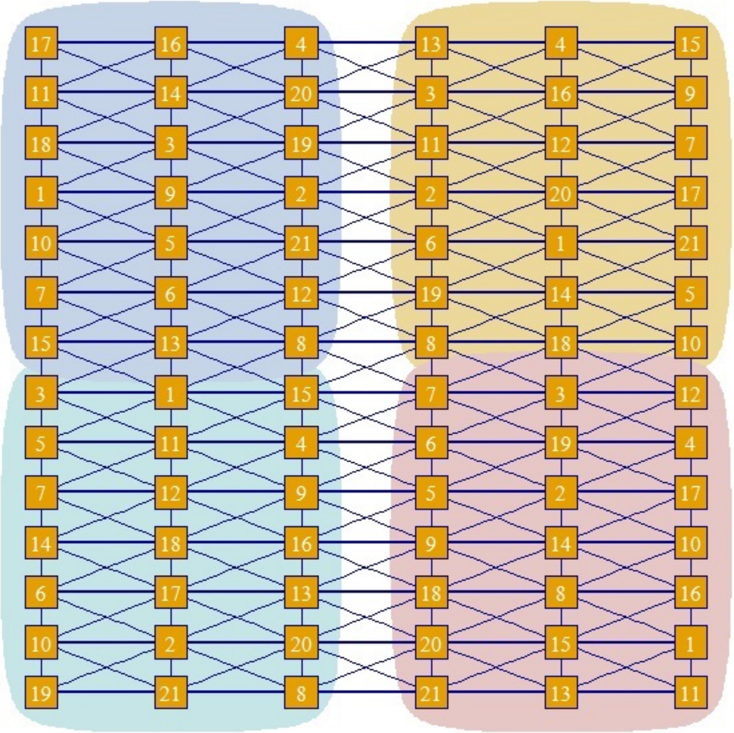


Figure C.16: A near-optimal LND4

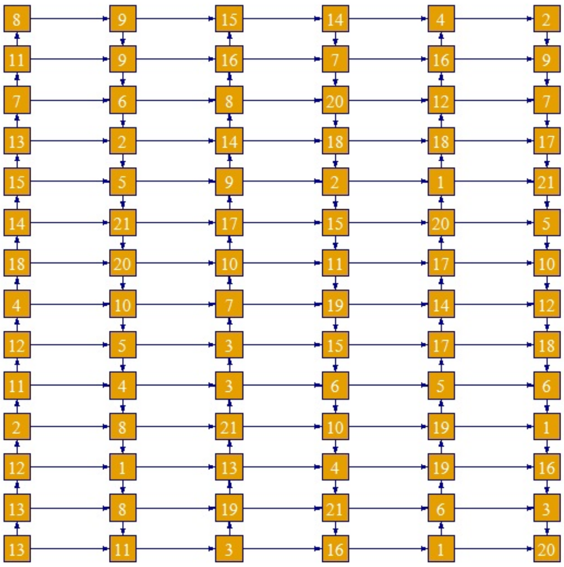


Figure C.17: A near-optimal NBD4

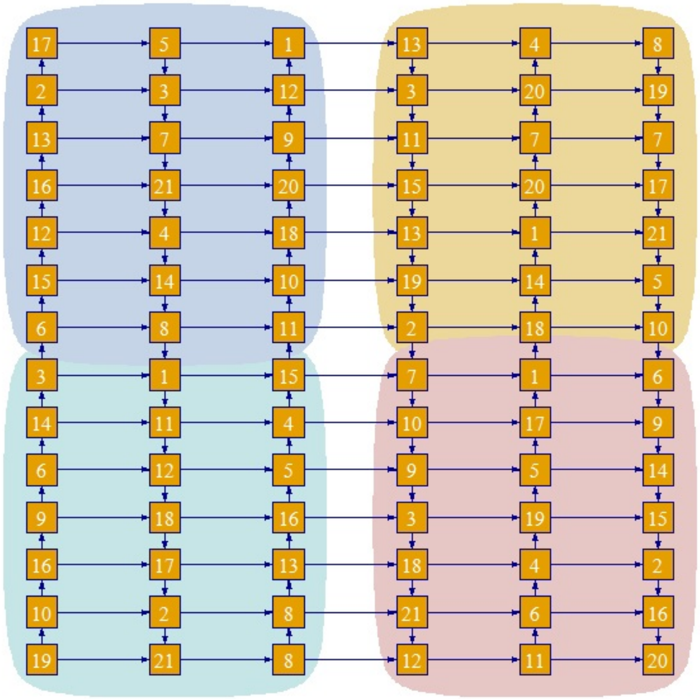


Figure C.18: A near-optimal RCND4

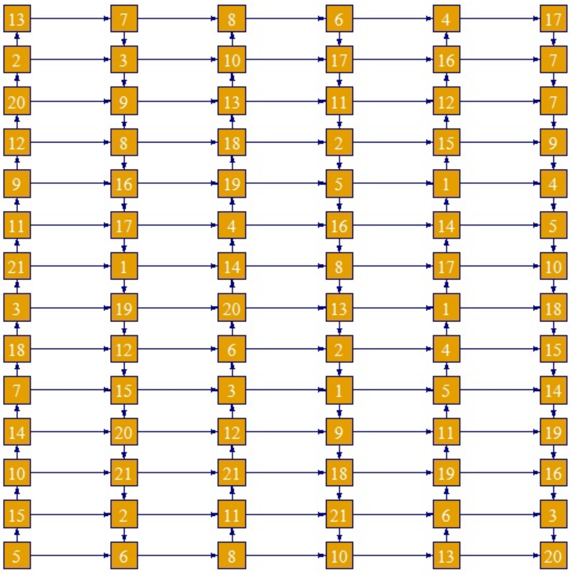


Figure C.19: A near-optimal RCNBD4

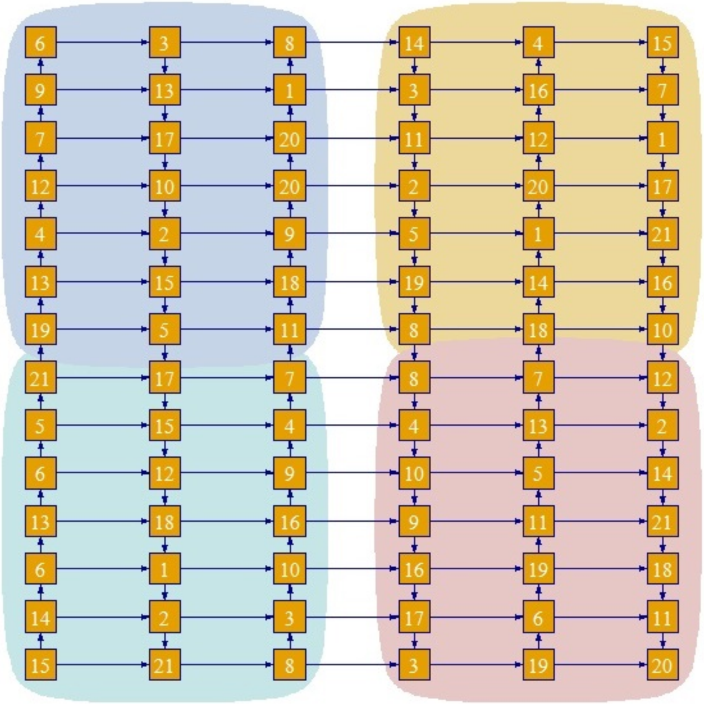


Figure C.20: A near-optimal NBD5

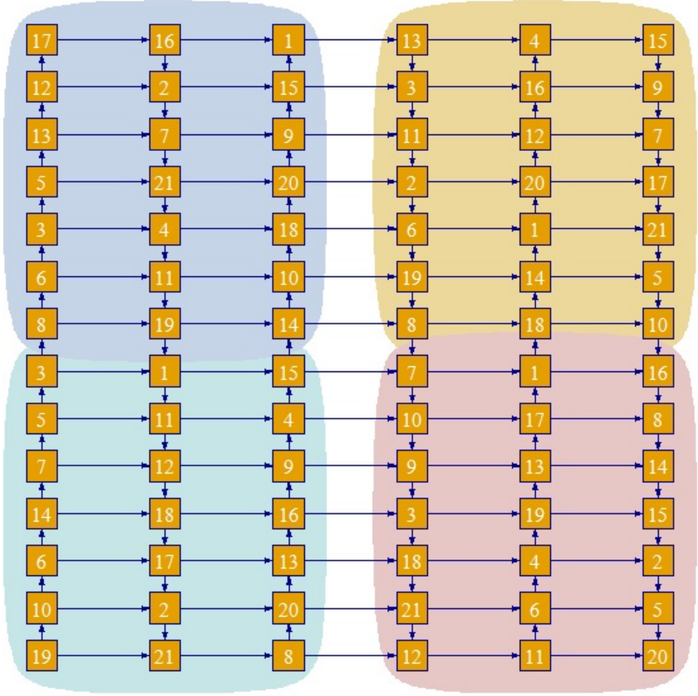
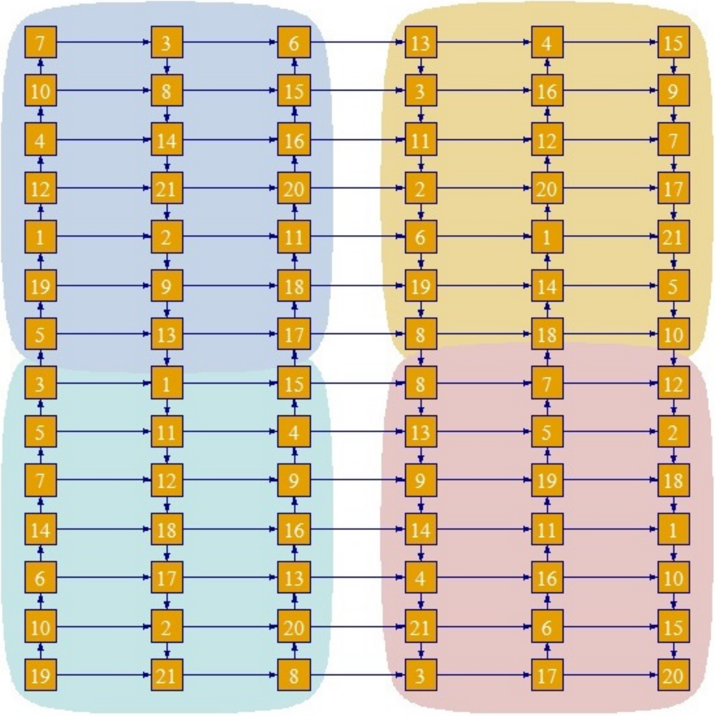


Figure C.21: A near-optimal RCNBD5



Bibliography

Alon, U. (2007) Network motifs: theory and experimental approaches. *Nature Reviews Genetics*, **8**, 450–461.

Angelucci, M., Prina, S., Royer, H. and Samek, A. (2015) When incentives backfire: spillover effects in food choice. Working paper no. 21481. Appeared in the August 2015 issue of the National Bureau of Economic Researh.

Anselin, L. (1988) *Spatial Econometrics: Methods and Models*. Dordrecht: Springer.

Aral, S. (2016) *The Oxford Handbook of the Economics of Networks*, chap. Networked experiments. New York: Oxford University Press.

Aral, S. and Walker, D. (2011) Creating social contagion through viral product design: a randomized trial of peer influence in networks. *Management Science*, **57**, 1623–1639.

Aronow, P. M. and Samii, C. (2013) Estimating average causal effects under interference between units. arXiv: 1305.6156.

Arora, S. and Barak, B. (2009) *Computational Complexity: A Modern Approach*. New York: Cambridge University Press.

Atkinson, A. C., Donev, A. N. and Tobias, R. D. (2007) *Optimum Experimental Designs, with SAS*. Oxford: Oxford University Press.

Azaïs, J. M., Bailey, R. A. and Monod, H. (1993) A catalogue of efficient neighbour - designs with border plots. *Biometrics*, **49**, 1252–1261.

Babai, L. (2015) Graph isomorphism in quasipolynomial time. arXiv: 1512.03547 [cs.DS].

Bailey, R. A. (1993) Recent advances in experimental design in agriculture. *Bulletin of the International Statistical Institute*, **55**, 179–93.

Bailey, R. A. (2008) *Design of Comparative Experiments*. Cambridge: Cambridge University Press.

Bailey, R. A. and Druilhet, P. (2004) Optimality of neighbor-balanced designs for total effects. *The Annals of Statistics*, **32**, 1650–1661.

Bapna, R. and Umyarov, A. (2015) Do your online friends make you pay? A randomized field experiment in an online music social network. *Management Science*, **61**, 1902–1920.

Barabási, A. L. and Albert, R. (1999) Emergence of scaling in random networks. *Science*, **286**, 509–512.

Barabási, A. L. and Albert, R. (2002) Statistical mechanics of complex networks. *Reviews of Modern Physics*, **74**, 47–97.

Bartlett, M. S. (1978) Nearest neighbour models in the analysis of field experiments (with discussion). *Journal of the Royal Statistical Society, Series B*, **40**, 147–174.

Basse, G. W. and Airoldi, E. M. (2015) Optimal design of experiments in the presence of network-correlated outcomes. arXiv: 1507.00803.

Besag, J. and Kempton, R. A. (1986) Statistical analysis of field experiments using neighbouring plots. *Biometrics*, **42**, 231–251.

Bond, R. M., Fariss, C. J., Jones, J. J., Kramer, A. D. I., Marlow, C., Settle, J. E. and Fowler, J. H. (2012) A 61-million-person experiment in social influence and political mobilization. *Nature*, **489**, 295–298.

Bose, R. C. (1947) On a resolvable series of balanced incomplete block designs. *Sankhyā: The Indian Journal of Statistics*, **8**, 249–256.

Bowers, J., Desmarais, B. A., Frederickson, M., Ichino, N., a. L. H. W. and Wang, S. (2016) Models, methods and network topology: experimental design for the study of interference. arXiv: 1601.00992v1.

Bowers, J., Fredrickson, M. M. and Panagopoulos, C. (2013) Reasoning about interference between units: a general framework. *Political Analysis*, **21**, 97–124.

Box, G. E. P. and Draper, N. R. (1959) A basis for the selection of a response surface design. *Journal of the American Statistical Association*, **54**, 622–654.

Brailsford, S. C., Potts, C. N. and Smith, B. M. (1999) Constraint satisfaction problems: algorithms and applications. *European Journal of Operational Research*, **119**, 557–581.

Browne, W. J., Goldstein, H. and Rasbash, J. (2001) Multiple membership multiple classification (MMMC) models. *Statistical Modelling*, **1**, 103–124.

Butcher, J. C. (1956) Treatment variances for experimental designs with serially correlated observations. *Biometrika*, **43**, 208–212.

Caldarelli, G. (2007) *Scale-Free Networks: Complex Webs in Nature and Technology.* Oxford: Oxford University Press.

Centola, D. (2010) The spread of behavior in an online social network experiment. *Science*, **329**, 1194–1197.

Chams, M., Hertz, A. and de Werra, D. (1987) Some experiments with simulated annealing for coloring graphs. *European Journal of Operational Research*, **32**, 260–266.

Chen, R. B., Chang, S. P., Wang, W., Tung, H. C. and Wong, W. K. (2014) Minimax optimal designs via particle swarm optimization methods. *Statistics and Computing*, **25**, 975–988.

Cheng, X. Q. and Shen, H. W. (2010) Uncovering the community structure associated with the diffusion dynamics on networks. *Journal of Statistical Mechanics: Theory and Experiment*, **2010**, 4–24.

Christakis, N. A. and Fowler, J. H. (2013) Social contagion theory: Examining dynamic social networks and human behavior. *Statistics in Medicine*, **32**, 556–577.

Chung, F. R. K. (1997) Spectral graph theory. In *CBMS Regional Conference Series in Mathematics*, no. 92. American Mathematical Society.

Cliff, A. D. and Ord, J. K. (1981) *Spatial Processes: Models and Applications.* London: Pion Limited.

Cook, R. D. and Nachtsheim, C. J. (1980) A comparison of algorithms for constructing exact Doptimal designs. *Technometrics*, **22**, 315–324.

Coppock, A. and Sircar, N. (2013) Design of field experiments under unknown interference structures. Working paper [Online] (Accessed: 22 September 2013).

Cox, D. R. (1958) *The Planning of Experiments.* New York: Wiley.

Csárdi, G. and Nepusz, T. (2006) The igraph software package for complex network research. *InterJournal, Complex Systems*, **1695**. URL `http://igraph.org`.

Dahl, B. V. and Van Zalk, M. H. W. (2014) Peer networks and the development of illegal political behavior among adolescents. *Journal of Research on Adolescence*, **24**, 399–409.

Dai, W. D. and Luca, M. (2016) Effectiveness of paid search advertising: experimental evidence. Working paper no. 17-025. Appeared in the March-April 2017 issue (pp.26-27) of Harvard Business Review.

Danon, L., Albert, D. G., Jordi, D. and Arenas, A. (2005) Comparing community structure identification. *Journal of Statistical Mechanics: Theory and Experiment*, **2005**, 219–228.

Danon, L., House, T. A., Read, J. M. and Keeling, M. J. (2012) Social encounter networks: collective properties and disease transmission. *Journal of Royal Society Interface*, **9**, 2826–2833.

Darga, P. T., Liffiton, M. H., Sakallah, K. A. and Markov, I. L. (2004) Exploiting structure in symmetry detection for CNF. In *Proceedings of the 41st Design Automation Conference*, 530–534.

Das, K. C. (2004) Maximizing the sum of the squares of the degrees of a graph. *Discrete Mathematics*, **285**, 57–66.

David, O. and Kempton, R. A. (1996) Designs for interference. *Biometrics*, **52**, 597–606.

Davison, A. C. (2003) *Statistical Models.* Cambridge: Cambridge University Press.

de Caen, D. (1998) An upper bound on the sum of squares of degrees in a graph. *Discrete Mathematics*, **185**, 245–248.

de Miguel Luken, V. and Tranmer, M. (2010) Personal support networks of immigrants to spain: a multilevel analysis. *Social Networks*, **32**, 253–262.

Doreian, P. (1980) Linear models with spatially distributed data: Spatial disturbances or spatial effects? *Sociological Methods and Research*, **9**, 29–60.

Draper, N. R. and Guttman, I. (1980) Incorporating overlap effects from neighbouring units into response surface models. *Journal of the Royal Statistical Society, Series C*, **29**, 128–134.

Druilhet, P. (1999) Optimality of neighbor-balanced designs. *Journal of Statistical Planning and Inference*, **81**, 141–152.

Eccleston, J. A. and Jones, B. (1980) Exchange and interchange procedures to search for optimal designs. *Journal of the Royal Statistical Society, Series B*, **42**, 238–243.

Eckles, D., Karrer, B. and Ugander, J. (2014) Design and analysis of experiments in networks: reducing bias from interference. arXiv: 1404.7530.

Egorova, O. (2017) Optimal design of experiments for multiple objectives. University of Southampton, Faculty of Social, Human and Mathematical Sciences. Doctoral Thesis.

Erdös, P. and Rényi, A. (1959) On random graphs. *Publicationes Mathematicae*, **6**, 290–297.

Fedorov, V. V. (1972) *Theory of Optimal Experiments.* New York: Academic Press.

Fedorov, V. V. and Hackl, P. (1997) *Model-Oriented Design of Experiments.* New York: Springer.

Fisher, R. A. (1935) *The Design of Experiments.* New York: Hafner Publishing Company, 8th edn.

Freeman, G. H. (1979) Some two-dimensional designs balanced for nearest neighbours. *Journal of the Royal Statistical Society, Series B*, **41**, 88–95.

Garlaschelli, D., Ruzzenenti, F. and Basosi, R. (2010) Complex networks and symmetry I: a review. *Symmetry*, **2**, 1683–1709.

Gill, P. S. and Shukla, G. K. (1985) Efficiency of nearest neighbour balanced block designs for correlated observations. *Biometrika*, **72**, 539–544.

Gilmour, S. G. and Trinca, L. A. (2012) Optimum design of experiments for statistical inference (with discussion). *Journal of the Royal Statistical Society, Series C*, **61**, 345–401.

Goos, P. and Gilmour, S. G. (2012) A general strategy for analyzing data from split-plot and multistratum experimental designs. *Technometrics*, **54**, 340–354.

Goos, P., Kobilinsky, A., O'Brien, T. E. and Vandebroek, M. (2005) Model-robust and model-sensitive designs. *Computational Statistics and Data Analysis*, **49**, 201–216.

Green, D. P., McGrath, M. C. and Aronow, P. M. (2013) Field experiments and the study of voter turnout. *Journal of Elections, Public Opinion and Parties*, **23**, 27–48.

Griffin, R. H. and Nunn, C. L. (2012) Community structure and the spread of infectious disease in primate social networks. *Evolutionary Ecology*, **26**, 779–800.

Guyon, I., Von Luxburg, U. and Williamson, R. C. (2009) Clustering: science or art? In *Neural Information Processing Systems (NIPS 2009) Workshop on Clustering Theory.*

Haines, L. M. (1987) The application of the annealing algorithm to the construction of exact optimal designs for linear-regression models. *Technometrics*, **29**, 439–447.

Harary, F. (1969) *Graph Theory.* Reading, Massachusetts: Addison-Wesley.

Harville, D. A. (1997) *Matrix Algebra from a Statistician's Perspective.* New York: Springer.

Hill, P. W. and Goldstein, H. (1998) Multilevel modeling of educational data with cross-classification and missing identification for units. *Journal of Educational and Behavioral Statistics*, **23**, 117–128.

Hinkelmann, K. and Kempthorne, O. (2005) *Design and Analysis of Experiments. Advanced Experimental Design*, vol. 2. New Jersey: Wiley.

Hinkelmann, K. and Kempthorne, O. (2007) *Design and Analysis of Experiments. Introduction to Experimental Design*, vol. 1. New Jersey: Wiley, 2nd edn.

Huang, Y. (2016) Optimal design of non-linear multi-factor experiments. University of Southampton, Faculty of Social, Human and Mathematical Sciences. Doctoral Thesis.

Imbens, G. W. and Rubin, D. B. (2015) *Causal Inference for Statistics, Social and Biomedical Sciences. An Introduction.* Cambridge: Cambridge University Press.

Ipinyomi, R. A. and John, J. A. (1985) Nested generalized cyclic row-column designs. *Biometrika*, **72**, 403–9.

John, J. A. and Eccleston, J. A. (1986) Row-column α-designs. *Biometrika*, **73**, 301–306.

John, J. A. and Whitaker, D. (1993) Construction of resolvable row-column designs using simulated annealing. *Australian Journal of Statistics*, **35**, 237–245.

John, J. A. and Williams, E. R. (1995) *Cyclic and Computer Generated Designs.* London: Chapman and Hall, 2nd edn.

Johnson, D. S. (1984) The genealogy of theoretical computer science: a preliminary report. *The Association for Computing Machinery: Special Interest Group on Algorithms and Computation Theory (ACM SIGACT News)*, **16**, 36–44.

Jones, B. and Donev, A. N. (1996) Modelling and design of cross over trials. *Statistics in Medicine*, **15**, 1435–1446.

Jones, B. and Eccleston, J. A. (1980) Exchange and interchange procedures to search for optimal row-and-column designs. *Journal of the Royal Statistical Society, Series B*, **42**, 372–376.

Jones, B. and Kenward, M. G. (2003) *Design and Analysis of Cross-over Trials.* London: Chapman and Hall, 3rd edn.

Junttila, T. and Kaski, P. (2007) Engineering an efficient canonical labeling tool for large and sparse graphs. In *Proceedings of the Ninth Workshop on Algorithm Engineering and Experiments*, 135–149.

Kempton, R. A. (1982) Adjustment competition for between varieties in plant breeding trials. *Journal of Agricultural Science*, **98**, 599–611.

Kendall, M. and Stuart, A. (1979) *The Advanced Theory of Statistics*, vol. 2. London: Charles Griffin, 4th edn.

Kennedy, J. and Eberhart, R. C. (2001) *Swarm Intelligence.* San Francisco: Morgan Kaufmann.

Kerber, A. (1971) *Representations of Permutation Groups, I*, vol. 240. New York: Springer.

Kiefer, J. (1959) Optimum experimental designs (with discussion). *Journal of the Royal Statistical Society, Series B*, **21**, 272–319.

Kiefer, J. and Wynn, H. P. (1981) Optimum balanced block and latin square designs for correlated observations. *The Annals of Statistics*, **9**, 737–757.

Kirkpatrick, S., Gelatt, C. D. and Vecchi, M. P. (1983) Optimisation by simulated annealing. *Science, Series B*, **220**, 671–680.

Köbler, J., Schöning, U. and Torán, J. (1993) *The Graph Isomorphism Problem. Its Structural Complexity.* New York: Springer.

Kunert, J. (1987) Neighbour balanced block designs for correlated errors. *Biometrika*, **74**, 717–724.

Kunert, J. and Martin, R. J. (2000) On the determination of optimal designs for an interference model. *The Annals of Statistics*, **28**, 1728–1742.

Kunert, J. and Mersmann, S. (2011) Optimal designs for an interference model. *Journal of Statistical Planning and Inference*, **141**, 1623–1632.

Kunin, W. E. (1998) Biodiversity at the edge: a test of the importance of spatial 'mass effects' in the rothamsted park grass experiments. *Proceedings of the National Academy of Sciences of the United States of America*, **1**, 207–212.

Larson, H. J. and Bancroft, T. A. (1963) Biases in prediction by regression for certain incompletely specified models. *Biometrika*, **50**, 391–402.

Lee, S. H., Kim, P. J. and Jeong, H. (2006) Statistical properties of sampled networks. *Physical Review E*, **73**, 016102.

Leenders, R. T. A. J. (1995) Structure and influence: statistical models for the dynamics of actor attributes, network structure and their interdependence. *Amsterdam: Thela Thesis Publishers.*

Leenders, R. T. A. J. (2002) Modeling social influence through network autocorrelation: constructing the weight matrix. *Social Networks*, **24**, 21–47.

Liu, J. and Liu, T. (2010) Detecting community structure in complex networks using simulated annealing with k-means algorithms. *Physica A: Statistical Mechanics and its Applications*, **389**, 2300–2309.

Liu, Z., Zhu, P., Chen, W. and Yang, R. J. (2015) Improved particle swarm optimization algorithm using design of experiment and data mining techniques. *Structural and Multidisciplinary Optimization*, **52**, 813–826.

MacArthur, B. D. and Anderson, J. W. (2006) Symmetry and self-organization in complex systems. arXiv: cond-mat/0609274.

MacArthur, B. D. and Sánchez-García, R. J. (2009) Spectral characteristics of network redundancy. *Physical Review E*, **80**, 026117.

MacArthur, B. D., Sánchez-García, R. J. and Anderson, J. W. (2007) On automorphism groups of networks. arXiv: 0705.3215v2.

MacArthur, B. D., Sánchez-García, R. J. and Anderson, J. W. (2008) Symmetry in complex networks. *Discrete Applied Mathematics*, **156**, 3525–3531.

Martin, R. J. and Eccleston, J. A. (2004) Variance-balanced designs under interference for dependent observations. *Journal of Statistical Planning and Inference*, **119**, 207–223.

Matthews, J. N. S. (1987) Optimal crossover designs for the comparison of two treatments in the presence of carry over effects and autocorrelated errors. *Biometrika*, **74**, 311–320.

McAuley, J. and Leskovec, J. (2012) Learning to discover social circles in ego networks. [Online] via NIPS. URL `http://cs.stanford.edu/people/jure/pubs/circles-nips12.pdf`.

McKay, B. D. (1981) Practical graph isomorphism. *Congressus Numerantium*, **30**, 45–87.

McKay, B. D. and Piperno, A. (2014) Practical graph isomorphism, II. *Journal of Symbolic Computation*, **60**, 94–112.

Mead, R. (1967) A mathematical model for the estimation of inter-plant competition. *Biometrics*, **23**, 189–205.

Mead, R., Gilmour, S. G. and Mead, A. (2012) *Statistical Principles for the Design of Experiments: Applications to Real Experiments.* Cambridge: Cambridge University press.

Menger, K. (1927) Zue allgemeinen kurventheorie. *Fundamenta Mathematicae*, **10**, 96–115.

Meyer, R. K. and Nachtsheim, C. J. (1995) The coordinate-exchange algorithm for constructing exact optimal experimental designs. *Technometrics*, **37**, 60–69.

Milgram, S. (1967) The small world problem. *Psychology Today*, **2**, 60–67.

Milo, R., Shen-Orr, S., Itzkovitz, S., Kashtan, N., Chklovskii, D. and Alon, U. (2002) Network motifs: simple building blocks of complex networks. *Science*, **298**, 824–827.

Molloy, M. and Reed, B. (1995) A critical point for random graphs with a given degree sequence. *Random Structures and Algorithms*, **6**, 161–180.

Newman, M., Barabási, A. L. and Watts, D. J. (2006) *The Structure and Dynamics of Networks.* New Jersey: Princeton University Press.

Newman, M. E. J. (2003) The structure and function of complex networks. *Society for Industrial and Applied Mathematics (SIAM) Reviews*, **45**, 167–256.

Newman, M. E. J. (2006) Modularity and community structure in networks. *Proceedings of the National Academy of Sciences of the United States of America*, **103**, 8577–8582.

Newman, M. E. J. (2010) *Networks: An introduction.* New York: Oxford University Press.

Newman, M. E. J. and Girvan, M. (2004) Finding and evaluating community structure in networks. *Physical Review E*, **69**, 026113.

Nguyen, N. K. and Miller, A. J. (1992) A review of some exchange algorithms for constructing discrete d-optimal designs. *Computational Statistics and Data Analysis*, **14**, 489–498.

Nguyen, N. K. and Williams, E. R. (1993) An algorithm for constructing optimal resolvable row-column designs. *Australian Journal of Statistics*, **35**, 363–370.

Ord, J. K. (1975) Estimation methods for models of spatial interaction. *Journal of the American Statistical Association*, **70**, 120–126.

Papadakis, J. S. (1937) Méthode statistique pou des expériences sur champ. *Bulletin Institut d' Amélioration des Plantes à Salonique (Gréce)*, **23**, 13–21.

Parker, B., Gilmour, S. G. and Schormans, J. (2016) Optimal design of experiments on connected units with application to social networks. *Journal of the Royal Statistical Society, Series C*, **66**, 455–480.

Patterson, H. D. and Williams, E. R. (1976) A new class of resolvable incomplete block designs. *Biometrika*, **63**, 83–90.

Pearce, S. C. (1957) Experimenting with organisms as blocks. *Biometrika*, **44**, 141–149.

Pearce, S. C. (1963) The use and classification of non-orthogonal designs. *Journal of the Royal Statistical Society, Series A*, **126**, 353–377.

Pukelsheim, F. (1993) *Optimal Design of Experiments.* New York: Wiley.

R Development Core Team (2017) R: A language and environment for statistical computing. R Foundation for Statistical Computing, Vienna, Austria. URL `http://www.R-project.org`.

Rees, D. H. (1967) Some designs of use in serology. *Biometrics*, **23**, 779–791.

Risau-Gusman, S. (2012) Influence of network dynamics on the spread of sexually transmitted diseases. *Journal of Royal Society Interface*, **9**, 1363–1372.

Rotman, J. J. (1994) *An Introduction to the Theory of Groups*, vol. 148. New York: Springer, 4th edn.

Rubin, D. B. (1974) Estimating causal effects of treatments in randomized and non-randomized studies. *Journal of Educational Psychology*, **74**, 688–701.

Rubin, D. B. (1990) Comment: Neyman (1923) and causal inference in experiments and observational studies. *Statistical Science*, **5**, 472–480.

Senn, S. (2002) *Cross-over Trials in Clinical Research.* Chichester: John Wiley and Sons, 2nd edn.

Shi, J. and Malik, J. (2000) Normalized cuts and image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **22**, 888–905.

Sinclair, B. (2011) *Cambridge Handbook of Experimental Political Science*, chap. Design and analysis of experiments in multilevel populations, 481–493. New York: Cambridge University Press.

Sinclair, B., McConnell, M. and Green, D. P. (2012) Detecting spillover in social networks: Design and analysis of multilevel experiments. *The American Journal of Political Science*, **56**, 1055–1069.

Singh, M. and Dey, A. (1979) Block designs with nested rows and columns. *Biometrika*, **66**, 321–6.

Snijders, T. A. B., van de Bunt, G. G. and Steglich, C. E. G. (2010) Introduction to stochastic actor-based models for network dynamics. *Social Networks*, **32**, 44–60.

Spall, J. P. (2003) *Introduction to Stochastic Search and Optimization.* New York: Wiley.

Street, D. J. and Street, A. P. (1985) Designs with partial neighbour balance. *Journal of Statistical Planning and Inference*, **12**, 47–59.

Taylor, W. H. and Hilton, H. G. (1981) A structure diagram symbolization for analysis of variance. *The American Statistician*, **35**, 85–93.

Tranmer, M., Steel, D. and Browne, W. J. (2014) Multiple-membership multiple-classification models for social network and group dependences. *Journal of the Royal Statistical Society, Series A*, **177**, 439–455.

Veenstra, R. and Steglich, C. (2012) *Handbook of Developmental Research Methods*, chap. Actor-based model for network and behavior dynamics, 598–618. New York: Guilford Press.

Von Luxburg, U. (2007) A tutorial on spectral clustering. *Statistics and Computing*, **17**, 395–416.

Von Luxburg, U. and Bousquet, O. (2004) Limits of spectral clustering. In *Neural Information Processing Systems (NIPS 2004) Workshop on Advances in NIPS 17.*

Wasserman, S. and Faust, K. (1994) *Social Network Analysis: Methods and Applications.* Cambridge: Cambridge University Press.

Watts, D. J. and Strogatz, S. H. (1998) Collective dynamics of small-world networks. *Nature*, **393**, 440–442.

Whitaker, D., Williams, E. R. and John, J. A. (1997) Cycdesign: a package for the computer generation of experimental designs (Version 1.0). *Canberra: CSIRO.*

Wilkinson, G. N., Eckert, S. R., Hancock, T. W. and Mayo, O. (1983) Nearest neighbour (NN) analysis of field experiments (with discussion). *Journal of the Royal Statistical Society, Series B*, **45**, 151–178.

Williams, E. R. and Talbot, M. (1993) *ALPHA+: Experimental Designs for Variety Trials. Design User Manual.* Canberra: CSIRO and Edinburg: SASS.

Williams, R. M. (1952) Experimental designs for serially correlated observations. *Biometrika*, **39**, 151–167.

Wong, W. K., Chen, R. B., Huang, C. C. and Wang, W. (2015) A modified particle swarm optimization technique for finding optimal designs for mixture models. *PLoS ONE*, **10**, e0124720.

Xiao, Y., MacArthur, B. D., Wang, H., Xiong, M. and Wang, W. (2008) Network quotients: structural skeletons of complex systems. *Physical Review E*, **78**, 046102.

Yandell, B. S. (1997) *Practical Data Analysis For Designed Experiments.* New York: Chapman and Hall.