UNIVERSITY OF SOUTHAMPTON

2-PERIOD TRAVELLING SALESMAN PROBLEM

Martin Butler, B.E., M.Sc., M.P.A.

<u>Doctor of Philosophy</u>

Department of Mathematics

Faculty of Mathematical Studies.

January, 1997

UNIVERSITY OF SOUTHAMPTON

ABSTRACT

FACULTY OF MATHEMATICAL STUDIES

MATHEMATICS

Doctor of Philosophy

2-Period Travelling Salesman Problem

By Martin Butler, B.E., M.Sc., M.P.A.

The 2-period travelling salesman problem originates from the collection of milk from dairy farmers in County Dublin, Ireland. Specifically, a group of dairy farms is allocated to a milk tanker. Of these farms some require every day collection, and others require collection every other day. The problem is to identify two tours with a combined distance that is minimised such that each farm requiring collection every day is visited by both tours, and each farm requiring collection every other day is visited by exactly one tour.

Optimal solution procedures are developed for examples of the problem. These procedures are based on integer programming formulations. These formulations are solved directly for small problems. The solution of medium sized problems, up to 100 nodes, require LP relaxation, subtour and comb constraints, and ultimately the solution of a considerably constrained $\{0,1\}$ model. The solution process identifies an important group of inequalities whose explicit presence in the model dramatically improves our ability to solve medium sized problems.

For problems with over 100 nodes the search time for an optimal solution becomes excessive. In these cases heuristic procedures, which provide good, but not necessarily optimal solutions, are used. A range of heuristic procedures are developed and empirically analysed. In the absence of an optimal answer, the heuristic solutions are compared with a lower bound on the optimal answer. Three classes of bounds are developed. The first class is based on increasingly constrained LP relaxations. The second class is based on an extension of the 1-tree concept. The third class is based on Lagrangian relaxation.

2-Period TSP

		Page
Table of Contents		i

List of Figures

List of Figures (Continued)

List of Tables

Acknowledgements

The author would like to thank Professor Paul Williams, University of Southampton, for all of his help and advice over many years. Professor Williams has an unique insight into the world of integer programming and combinatorial optimisation, and it was through his wisdom that the author found a problem that was both stimulating and challenging.

I would like to thank Mr Pat Doyle, Premier Dairies, who first introduced me to the intricacies of milk collection.

The support from my colleagues in the Commerce Faculty, University College Dublin, is much appreciated, and their help made a long road not as strenuous as it might have been. A special word of thanks to Dr Derek O'Connor for his many hours of technical help.

Most importantly, I would like to thank my family for all of their understanding and love over the long haul.

Chapter 1 2-Period Travelling Salesman Problem

1.1 Background in the Irish Milk Industry

The fresh milk output from Irish dairy farmers is either collected once a day or once every second day by large bulk milk tankers. The preferred option is to call once to each farmer every second day. However, the required collection frequency for a particular farmer depends on the capacity of the milk storage tank on the farm. If the on-farm tank capacity is not sufficient to hold two days milk output, then that farmer must be called to every day. Various schemes are in operation to encourage farmers who require collection every day to upgrade their on-farm milk storage tanks.

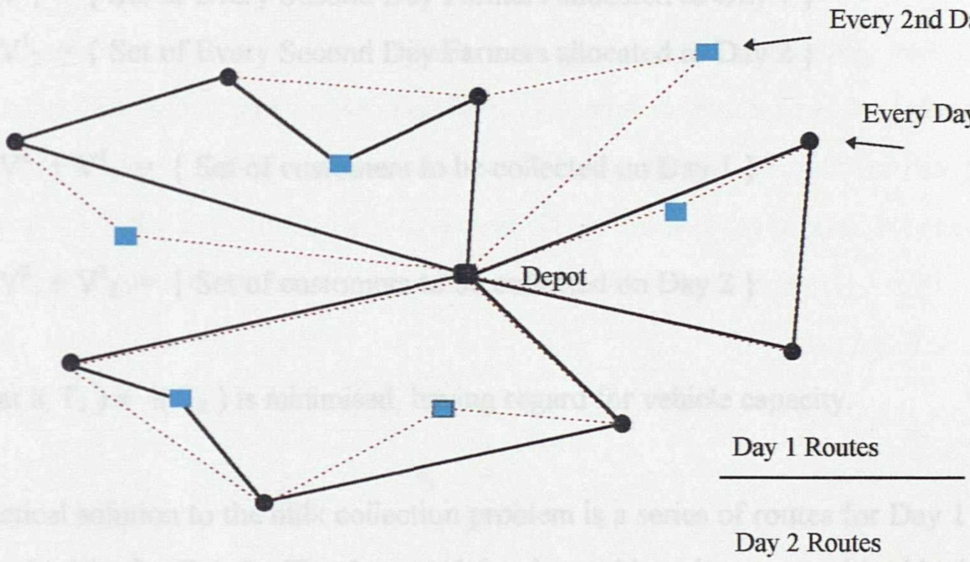


Figure 1.1 : Day 1 and Day 2 Milk Collection Routes

At present the scheduler of the milk tanker fleet has two problems. The first problem is to decide on how best to allocate the farmers requiring collection every second day to either a Day_1 route or a Day_2 route. A Day_1 route is driven on Mondays, Wednesdays, etc. The Day_2 route is driven on Tuesdays, Thursdays, etc. Having allocated the every second day farmers, the scheduler must now, having regard for vehicle capacity, generate efficient tours to service both the every day and every second day farmers. The solution to a typical milk collection problem is shown in Figure 1.1.

The problem of the milk scheduler can be described as follows :

Let V^1 = { Farmers Requiring Every Second Day Collection }

V^2 = { Farmers Requiring Every Day Collection }

T_1 = { Set of Routes Driven on Day 1, of total length $l(T_1)$ }

T_2 = { Set of Routes Driven on Day 2, of total length $l(T_2)$ }

Find the partition of V^1 into $V^1{}_1$ and $V^1{}_2$, where:

$V^1{}_1$ = { Set of Every Second Day Farmers allocated to Day 1 }

$V^1{}_2$ = { Set of Every Second Day Farmers allocated to Day 2 }

$V^2 + V^1{}_1$ = { Set of customers to be collected on Day 1 }

$V^2 + V^1{}_2$ = { Set of customers to be collected on Day 2 }

Such that $l(T_1) + l(T_2)$ is minimised, having regard for vehicle capacity.

The practical solution to the milk collection problem is a series of routes for Day 1 and a series of routes for Day 2. The data used for the problem, later termed in this thesis as the 42 node problem, is taken from a real milk collection problem from County Meath, Ireland. The problem, introduced in this thesis as the 2-period Travelling Salesman Problem, is a relaxation of the practical milk collection problem in that the capacity restrictions are removed. In the absence of the truck capacity restriction, the solution will be one large tour for Day 1 and a second large tour for Day 2.

1.2 Additional Applications of the 2-Period TSP.

While the 2-period TSP has its origins in the milk collection arena, relaxations of many other routing problems can be formulated as the 2-period TSP. These include applications with Airport Coach Services and Post Collection. In addition, the formulation has application to the distribution of goods to retail outlets where the size of the outlet dictates the frequency and level of service the outlet is to receive. In this latter case the 2-period model can be generalised to an m-period model. The m-period problem is discussed in Chapter 8.

1.2.1 Airport Coach Service Application.

An airport coach servicing down-town hotels calls to the larger hotels every half-hour, and to the smaller hotels every hour. The tours that start on the hour call to all of the larger hotels and to a subset of the smaller hotels. The tours that start on the half-hour also call to all of the large hotels, and the remainder of the small hotels not serviced by the tours that start on the hour. The situation can be depicted as follows :

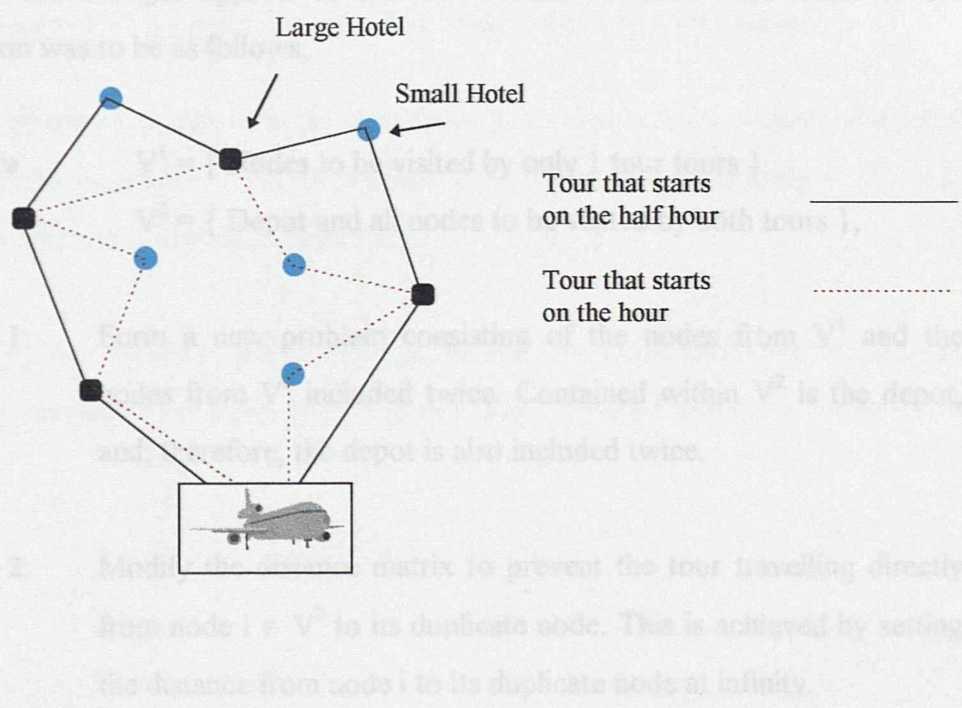


Figure 1.2 : Routes for the Airport Coach Service

Ignoring capacity constraints, the coach operator seeks to partition optimally the small hotels across the two tours, while generating optimal tours for the coaches.

1.2.2 Post Collection Application

To facilitate customers wishing to post letters, the Post Office locates, at various places in a town, post-boxes. Depending on the location of the post-boxes, the boxes are emptied by a Post Office service van every hour or every two hours. Ignoring capacity and maximum length of tour constraints, the allocation of post-boxes to tours, and the optimal design of the tours is an application of the 2-Period TSP.

1.3 Comparison with the Standard Travelling Salesman Problem.

The standard travelling salesman problem, TSP, is concerned with the tour of minimum length required to pass through every node in a graph. A considerable effort has been used by academics to investigate the TSP. When the author first encountered the 2-period TSP, he assumed that this problem could be transformed into a TSP, and, thus, all of the thought applied to the TSP could be used. The basis of the transformation was to be as follows,

where V^1 = { Nodes to be visited by only 1 tour tours }

V^2 = { Depot and all nodes to be visited by both tours },

Step 1: Form a new problem consisting of the nodes from V^1 and the nodes from V^2 included twice. Contained within V^2 is the depot, and, therefore, the depot is also included twice.

Step 2: Modify the distance matrix to prevent the tour travelling directly from node $i \in V^2$ to its duplicate node. This is achieved by setting the distance from node i to its duplicate node at infinity.

Figure 1.3 shows the nodes of an 11 node version of the 2-period TSP. For this problem $V^2 = \{ 1, 2, 3, 4, 5 \}$ and $V^1 = \{ 6, 7, 8, 9, 10, 11\}$. Figure 1.4 shows the transformed problem with all nodes in V^2 duplicated.

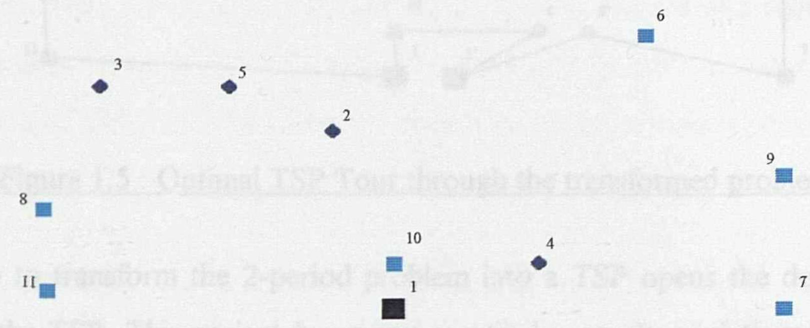


Figure 1.3 : 11 node 2-Period TSP.

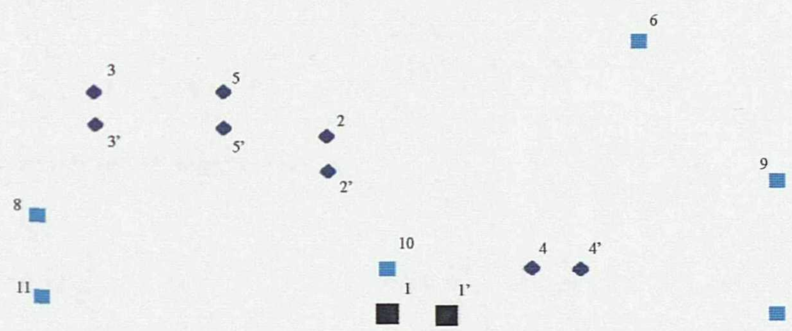


Figure 1.4 : 11 node 2-Period TSP with nodes 1 to 5 duplicated.

A travelling salesman solution to the transformed problem would consist of a large tour which visits the depot twice, each node $\in V^2$ twice, and each node $\in V^1$ only once. Such a solution is shown in Figure 1.5.

Unfortunately, the solution to the 2-period problem is two tours, and it is impossible to untangle the required two tours from the one tour obtained from the transformed problem.

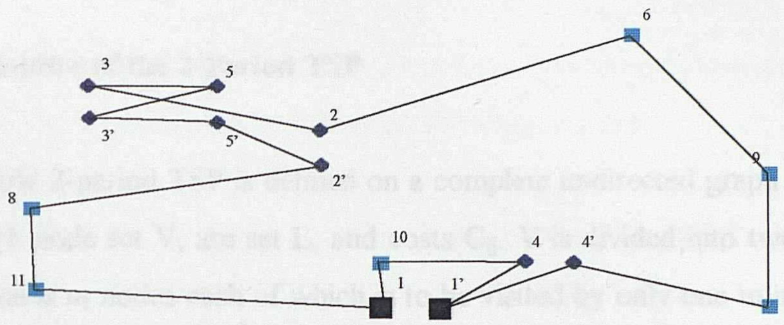


Figure 1.5 : Optimal TSP Tour through the transformed problem.

The failure to transform the 2-period problem into a TSP opens the door on a new variant of the TSP. This variant has many practical examples and therefore warrants the full analysis detailed in this thesis.

1.4 Definition of the 2-Period TSP

The symmetric 2-period TSP is defined on a complete undirected graph G = (V,E) on n nodes, with node set V, arc set E, and costs C_{ij}. V is divided into two sets, V^1 and V^2. V^1 contains n_1 nodes each of which is to be visited by only one tour. V^2 contains n_2 nodes each of which is to be visited by both tours. The problem is stated as

$$Minimise \sum_{i \in V} \sum_{j>i} \sum_{k=1}^{2} C_{ij} X_{ijk} \tag{1.1}$$

$$subject\ to$$

$$\sum_{j:j<i} X_{jik} + \sum_{j:j>i} X_{ijk} = 2, \qquad i \in V^2,\ k = 1,\ 2 \tag{1.2}$$

$$\sum_{j:j<i} X_{jik} + \sum_{j:j>i} X_{ijk} - 2Y_{ik} = 0, \qquad i \in V^1,\ k = 1,\ 2 \tag{1.3}$$

Subtour Elimination Constraints (1.4)

$$Y_{i1} + Y_{i2} = 1, \qquad i \in V^1 \tag{1.5}$$

$$X_{ijk} = 0 \text{ or } 1 \qquad i,j \in V, j > i, and\, k = 1\ or\ 2 \tag{1.6}$$

$$Y_{ik} = 0 \text{ or } 1 \qquad i \in V^1,\ k = \ 1\ or\ 2 \tag{1.7}$$

X_{ijk} = 1 if arc i to j is used by tour k; otherwise X_{ijk} =0. The Y variables are associated with the nodes to be visited by only one tour. Y_{ik} = 1 if node i, $i \in V^1$, is on tour k; otherwise Y_{ik} = 0. Constraints (1.5) require that each node i, where $i \in V^1$, is only on one of the two tours.

Constraints (1.2) and (1.3) together form the 2-matching constraints. Constraints (1.2) require that 2 arcs from each tour are attached to all nodes in V^2. While constraints (1.3) force each node in V^1 to be connected by two arcs to one of the tours. The tour to which a node i in V^1 is connected depends on whether Y_{i1} or Y_{i2} equals 1.

The subtour elimination constraints, constraints (1.4), require that the solution to the 2-period TSP be two complete tours. The constraints are derived from the subtour elimination constraints for the Standard Symmetric Travelling Salesman Problem. For that problem the sub-tour elimination constraint suggests that

$$\sum_{i,j \in S} X_{ij} \le n(S) - 1, \qquad \text{where S is a subset of the set of all nodes.}$$

To extend this sub-tour elimination concept to the 2-Period TSP requires a partition of S, a subset of all nodes, into S^1 and S^2 , where S^1 contains nodes visited by only one tour, and S^2 contains nodes visited by both tours. The form of the generalised sub-tour elimination constraint depends on whether either S^1 or S^2 are empty. The three possible cases are:

Case 1 : $S^1 = \phi$, S contains only nodes visited by both tours.

$$\sum_{i,j \in S} X_{ijk} \le n(S^2) - 1, \qquad \text{for k = 1 or 2}$$

Case 2 : S^1 and $S^2 \neq \phi$, S contains both types of nodes.

$$\sum_{i,j \in S} X_{ijk} \le \sum_{i \in S^1} Y_{ik} + n(S^2) - 1, \text{ for k = 1 or 2}$$

Case 3 : $S^2 = \phi$, S contains only nodes visited by one of the tours.

$$\sum_{i,j \in S} X_{ijk} \le \sum_{i \in S^1} Y_{ik} - \underset{i \in S^1}{Max}\left[Y_{ik}\right], \text{ for k = 1 or 2}$$

The solution to the standard TSP is a tour of minimum length. The 2-period TSP is more complex in that the solution contains both an allocation of the nodes in V^1 to one of the tours, and also a set of two tours of minimum total length.

When the 2-period TSP problem arises in a practical application, there is always the hope that an optimal solution can be found. Industry would like to believe that when Management Science is brought to a problem, then an optimal answer is available for implementation. However, like the TSP, if the search for an optimal solution proves impractical, then, the only resort is heuristic procedures.

1.5 Thesis Overview.

This thesis is concerned with the 2-period TSP. Chapter 2 reviews the academic literature on the TSP, and uses the experience of other researchers to indicate how solutions can be effectively obtained for this new variant of the TSP.

Chapter 3 optimally solves 3 examples of the 2-period TSP. These examples, termed the 11 node, 21 node, and 42 node problems, represent small and medium sized versions of the problem. The data within the 42 node problem relates to a problem of milk collection from Ireland. Chapter 3 shows that, for small problems, the $\{0,1\}$ mathematical formulation can be solved in reasonable time with sub-tour elimination constraints being added on an "as needed" basis. This direct solution of the $\{0,1\}$ formulation fails in Chapter 3 for the 42 node problem. An approach based on an increasingly constrained LP relaxation is required to solve the 42 node problem.

Chapter 4 builds on the successful solutions obtained in Chapter 3, and attempts to use the same solution methodology on a 100 node version of the problem. The explosive combinatorial nature of the problem is exposed in Chapter 4, and only after a heroic struggle did the 100 node problem finally yield an optimal solution.

Historically two solution methodologies have evolved for the TSP. One is based on LP relaxations and $\{0,1\}$ solution. This procedure is used in Chapters 3 and 4. The other methodology is based on branch and bound. Fundamental to this approach is the ability to find good lower bounds on the optimal solution. Chapter 5 discusses various approaches for obtaining lower bounds for the 2-period TSP. In addition to their use in obtaining optimal solutions, bounds are invaluable when analysing heuristic solutions in the absence of an optimal solution.

The problems, within Chapter 4, of finding an optimal solution to a 100 node version of the 2-period TSP, suggest that 100 nodes is very close to the upper limit on the problem size that can be optimally solved. Chapter 6 introduces heuristic procedures as an alternative option when a good, but not necessary optimal , solution is required. Chapter 7 empirically analyses a variety of heuristic procedures.

Chapter 8 summarises the thesis, and identifies areas of potential future research.

1.6 Day_1 and Day_2 Tours.

The 2-period TSP finds its origins in the mundane art of milk collection in County Meath, Ireland. In this application the scheduler attempts to find a solution for day 1 and a solution for day 2. To reflect the humble origins of the problem, the two tours created by the 2-period TSP are termed the Day_1 and Day_2 tours in this thesis.

Chapter 2 Literature Review

2.1 Introduction

The travelling salesman problem is the problem of finding the shortest Hamiltonian tour in a graph. This problem appears to have been formulated some 70 years ago and has been the subject of intensive investigation in combinatorial optimisation during the past 40 years. The interest in the problem arises both from the many practical problems that can be formulated as a TSP, and also because of its pivotal position in the mathematics of combinatorial optimisation.

The 2-period TSP, introduced in Chapter 1, is a variant of the TSP. This suggests that the history of the attempts to find optimal solutions to the TSP is of interest, and hopefully the experiences from the successes and failures will guide us in our attempts to find solutions to the 2-period TSP.

The original attempts at solving the TSP are based on Mathematical Formulations of the problem.

2.2 Mathematical Formulation

In their survey paper, Bellmore and Nemhauser [1966], described two mathematical formulations for the Travelling Salesman Problem. The first is based on the work by Dantzig, Fulkerson and Johnson [1954], the second is from work by Miller, Tucker and Zemlin [1960]. The main difference in these formulations is in how they handle the requirement that the solution is a tour. The two formulations for an asymmetric problem on n nodes are :

Formulation I (Dantzig et al)

$$\text{Minimise} \quad \sum_{i,j} C_{ij} X_{ij}$$

$$\text{Subject to} \quad \sum_{j} X_{ij} = 1 \qquad (i = 1,\ldots, n),$$

$$\sum_{i} X_{ij} = 1 \qquad (j = 1,\ldots,n),$$

$$\sum_{i \in S, j \in S} X_{ij} \le |S| - 1 \qquad \text{S is a subset of n nodes}$$

$$X_{ij} = 0,1$$

Formulation II (Miller et al)

$$\text{Minimise} \quad \sum_{i,j} C_{ij} X_{ij}$$

$$\text{Subject to} \quad \sum_{j} X_{ij} = 1 \qquad (1 \le i \le n),$$

$$\sum_{i} X_{ij} = 1 \qquad (1 \le j \le n),$$

$$u_i - u_j + (n-1)X_{ij} \le n - 2 \qquad (2 \le i \le n), \; (2 \le j \le n) \; and \; i \ne j$$

$$X_{ij} = 0,1$$

$$u_i \ge 0 \qquad (2 \le i \le n)$$

Formulation II requires significantly fewer constraints than Formulation I. However, as Bellmore and Nemhauser, show this does not necessarily mean that Formulation II is easier to solve. Consider a simple example of n nodes, where a solution consists of subtours (2, 3, 4, 2) and (5, ..., n, 5). Formulation I would block these subtours by the constraint :

$$X_{23} + X_{34} + X_{42} \le 2$$

Using Formulation II, three constraints are required

$$u_2 - u_3 + (n-1)X_{23} \leq n - 2$$

$$u_3 - u_4 + (n-1)X_{34} \leq n - 2$$

$$u_4 - u_2 + (n-1)X_{42} \leq n - 2$$

When these constraints are added we get

$$X_{23} + X_{34} + X_{42} \leq 3 - 3/(n-1)$$

Although this constraint is sufficient to block the subtour, it is weaker than

$$X_{23} + X_{34} + X_{42} \leq 2$$

and admits more feasible fractional subtours.

The solution of Formulation I with the integer restriction replaced by $0 \leq X_{ij} \leq 1$ will not generally provide a {0,1} solution. In 1954, Dantzig, Fulkerson, and Johnson found an optimal solution to a 42 city problem starting with the LP relaxation of Formulation I. They overcame the large number of subtour elimination constraints by beginning with only a few, and then adding new ones only as they were needed to block subtours. They used the cutting constraints, a forerunner of Gomory's cutting planes, to rule out fractional subtours while preserving integer solutions.

Miller, Tucker, and Zemlin [1960] experimented using a similar approach with Formulation II. Unfortunately, the results were disappointing. Since the work done by Miller et al, very little attention has been given to Formulation II although the LP relaxation is clearly weaker than that of Formulation I. It is suggested in Chapter 8, that possibly Formulation II justifies further attention.

Claus [1984] outlines a further formulation for the TSP. This formulation replaces the exponential number subtour elimination constraints by a number of constraints that is proportional to the number of nodes times the number of finite cost arcs in the graph. In addition, the new formulation introduces a new set of variables. Claus argues that the resulting polytope is smaller that the subtour elimination polytope. Few experimental results exist for this formulation.

2.3 Optimal Solution Based on Mathematical Programming

Miliotis [1976], using the pioneering FORTRAN Code of Land-Powell [1973], reports the successful solution of medium sized, < 50 nodes, TSP's using a combination of LP relaxation, inequalities added on an "as needed" basis, and branching to eliminate fractional values and to restore integer solutions. Miliotis notes that only a small number of omitted constraints are added during the solution process.

The use of pure Linear Programming to solve a TSP requires a set of inequalities that will linearly describe the convex polytope. Grotschel and Padberg [1977] suggest that the number of linear inequalities required to describe the polytope is astronomically large, and that an algorithmic approach to the TSP based on linear inequalities must fail.

In their paper, Grotschel and Padberg [1977] prove that :

- The 2-matching constraints,
- The subtour elimination constraints, and
- The comb constraints, suggested by Chvátal [1973],

define facets of the polytope for the symmetric TSP. However, they also state that these inequalities to not completely characterise the polytope of tours. The intersection of the above inequalities has fractional as well as tour vertices.

Grotschel and Padberg [1977] show that, while the number of the above constraints is so large that they cannot be explicitly included in the formulation, a linear programming approach which adds constraints on an "as needed" basis performs very well. They show that, since no complete linear description of the polytope is known, the optimal solution to the LP model is a lower bound on the optimum tour length. Finally, they suggest, that integer programming can be used to bridge the gap between the lower bound and the optimal solution.

Padberg and Hong [1980] support earlier work that inequalities defining facets of the convex hull of tours are of substantial computational value in the solution of symmetric TSPs. Their approach is to use a heuristic tour as a starting solution for a linear programme, and then to gradually add more inequalities that define facets in order to prove optimality.

Johnson and McGeoch [1995] report that the TSP is one of the major success stories of optimisation. Years of research into optimisation techniques, combined with the continuing rapid growth in computer technology have led from one new record to another. Over the past 15 years, the record for the largest nontrivial TSP solved to optimality has grown considerably from the famous 318 cities problem of Crowder and Padberg [1980].

An adaptation of the approach used by Grotschel and Padberg [1977] is the methodology used in Chapter 3 and Chapter 4 to find optimal solutions to the 42 node and 100 node problems.

In parallel with the search for optimal procedures based on Mathematical Programming, Branch and Bound approaches have been developed.

2.4 Branch and Bound

Garfinkel [1979] suggests that Branch and Bound approaches are appropriate for virtually all combinatorial problems. The term Branch and Bound appears in two different contexts in the search for an optimal solution to the TSP.

Firstly, it is a means of moving forward when an LP formulation, on the type discussed in Section 2.3, yields a non { 0,1} integer solution. In this case the general approach is to take a variable x, that should be {0, 1}, but whose current value is fractional, and branch on x=0 and x=1. This leads to a search tree which grows until a bounding approach indicates that a branch need not be explored beyond some node. This approach is used in Dantzig, Fulkerson & Johnson [1954].

Branch and bound is also used when an optimal solution of an easily solvable relaxation of the TSP is found. If this solution is a tour then the process terminates. If a solution is not a tour, then the solution is used as a lower bound on the tour, and the problem is branched into a set of sub-problems. The process continues until one of the sub-problems yields a feasible tour, and the bounding process suggests that the remaining nodes offer no potential for a better solution. Little, Murty, Sweeney and Karel [1963] describe such an approach for the TSP. Their bounding process is based on matrix reduction, while the branching is done by forcing one arc to be in the solution of one of the sub-problems, and prohibiting that same arc from the other sub-problems.

For Branch and Bound the quality of the computed bounds is of greater importance for the effectiveness of the algorithm that the branching rules used. The most celebrated bound for the symmetric TSP is one developed by Held and Karp [1970] and is based on the concept of a 1-tree. A 1-tree is derived from the minimum spanning tree plus an additional arc. The quality of the bound from the 1-tree is not usually very good. However, the quality of the bound can be increased considerably by the use of

Lagrangean relaxation. Empirical results suggests that this bound is normally within 1% of the optimal tour length.

Other bounds for the TSP have been derived from,

- The Assignment Problem, Balas and Christofides [1976], and
- The Shortest n-Paths, Houck, Picard, Queyranne and Vemuganti, [1977].

2.5 Need for Heuristics

Network problems can be classified according to a theoretical scheme based on the notion of "polynomially-bounded" and "NP-hard" classes. The class P is composed of those problems for which polynomially-bounded algorithms are known to exist. A polynomially-bounded algorithm is a procedure whose worst case computational effort increases only polynomially with problem size. Problems belonging to the class P can generally be solved to optimality quite efficiently.

In contrast to the class P, there is a large class of combinatorial problems for which no polynomially-bounded algorithm has yet been found. This is the class of "NP-hard" problems. For a more precise definition see Garey and Johnson [1979].

The class of NP-hard problems may be viewed as forming a hard core of problems that polynomial algorithms have not been able to penetrate so far. The practical implication of this classification is that for NP-hard problems, even a modest increase in problem size will result in a prohibitive growth in the computational time required to find the optimal solution. The TSP is a well known NP-hard problem.

To overcome our inability to find optimal solutions to large examples of NP-hard problems, one frequently resorts to heuristic or approximate procedures.

2.6 Heuristic Procedures

Notwithstanding the impressive gains in obtaining optimal solutions from increasingly larger TSPs, the reality is that heuristic procedures are still required to find "good" solutions in practical applications.

Johnson and McGeoch [1995] suggest that the world of heuristic approaches to the TSP can be divided into two classes - local search approaches and tour construction heuristics. They report that tour construction heuristics do surprisingly well in practice, and that local search heuristics typically get within 3-4% of the optimal. They remark that the success of the traditional approaches leaves less room for the new approaches like tabu search and simulated annealing to make a contribution.

2.7 Analysis of Heuristics

The development of alternate heuristics for the TSP requires a methodology to compare their relative performance. Three different methods of comparison are available. These are empirical analysis, worst-case analysis and probabilistic analysis.

Empirical analysis is the traditional method of comparison. The heuristic, under consideration, is applied to a set of test problems and the solution values obtained are compared to the optimal solution, if it is known, or to a lower bound, if the optimal solution is not known, or to other heuristic results. Over time a series of test problems have evolved in the academic literature, and all new heuristics are compared against these test problems. The problem with empirical analysis is that it gives no performance guarantees. What use is it in a practical situation that the methodology you have adopted normally performs well, but you have just been unlucky?

Fisher [1980] defines worst-case analysis as a way of establishing the maximum deviation from optimality that can occur when a specified heuristic is applied within a given problem class. Worst-case analysis has the advantage of providing a guarantee on "how bad" the heuristic result could be. The main disadvantage is that the worst-case performance is usually not predictive of average performance. The best worst case polynomially bounded heuristic for the TSP is one suggested by Christofides [1977]. This bound is marginally improved by Cornuejols and Nemhauser [1978].

To overcome some of the negativity associated with worst-case analysis, probabilistic analysis is introduced. Probabilistic analysis predicts how the heuristic will perform for a "typical" problem instance. What defines a "typical problem" is a major drawback with probabilistic analysis.

In this thesis, various test problems are defined for the 2-period TSP, and the developed heuristics are empirically tested against these test problems.

2.8 Variants of the Travelling Salesman Problem

The 2-period TSP is only one of many variants of the basic TSP. Other variants include :

Prize Collecting Travelling Salesman, Balas [1989]

M-Tour Travelling Salesman, Russell [1976]

Bottleneck Travelling Salesman, Garfinkel and Gilbert [1978]

Time-Dependent Travelling Salesman, Fox, Gavish and Graves [1980].

Examinations of the above variants are important because they add insight into the variant, and then by reflection back to the basic TSP. For example, the analysis by Balas [1989] of the inequalities that define the polytope of the Price Collecting Travelling Salesman Problem is of help when one analyses the inequalities of the 2-period TSP.

2.9 The Period Routing Problem

Both the vehicle routing problem and the travelling salesman problem are traditionally concerned with minimising total distance or total cost on the assumption that any tour derived or routes generated will be followed on a particular day. In formulating these models no regard is had for such practical considerations as driver familiarity with a route, customer requirements for a fixed call, and varying customer service levels. Practical considerations suggest that a routing model with a time horizon longer than one day, The Period Routing Model, may have many applications.

Beasley [1984] distinguished three different types of vehicle routing problems - The Daily Routing, The Period Routing, and the Fixed Routes.

The Daily Routing is concerned with developing a set of vehicle routes for a single day's deliveries. The vast bulk of routing heuristics such as Clarke and Wright [1964] have been developed for this type of problem.

The Period Routing is where a set of vehicle routes is developed for a certain period to meet customer service levels requirements. The typical period here is seven days, and customers have such requirements as only deliveries on Mondays and Thursdays, or Fridays only, etc. Christofides and Beasley [1984] developed a heuristic for the problem of designing vehicle routes to meet service levels for customers. Their heuristic is based on an initial allocation of customers to days, followed by an interchange procedure. Russell and Igo [1979] examines a routing problem in which the objective is to assign customer demand points to days of the week in order to optimise the week's routing effort.

The Fixed Routes problem arises when a set of vehicle routes has to be developed that can operate unchanged for a given period of time. The model has application in the area of milk collection because dairy farmers require the milk tanker to call at fixed times so as to synchronise with milking time. Beasley [1984] suggests a heuristic for the Fixed Routes problem.

The 2-Period TSP is both a variant of the TSP, and an example of a Period Routing Model.

Chapter 3 Optimal Solution

This chapter describes the solution of three examples of the 2-period TSP. These examples, termed the 11 node, 21 node and 42 node problems, represent small and medium sized versions of the problem. This chapter demonstrates that, while small problems can be solved directly using direct {0,1} programming, increasing the problem size considerably complicates the search for an optimal solution.

3.1 11 Node Problem

The 11 node problem consists of a depot, node number 1, 4 nodes to be visited by both tours, and 6 nodes to be visited by only one tour. The data for this problem is contained in Appendix 1.

Section 1.4 of chapter 1 details a {0,1} formulation for a general 2-period TSP. The formulation for the 11 node problem is as follows :

Minimise $Z = 41X_{01,02,1} + 58X_{01,03,1} + 14X_{01,04,1} + 54X_{01,05,1} + 73X_{01,06,1} +$
$40X_{01,07,1} + 45X_{01,08,1} + 50X_{01,09,1} + 10X_{01,10,1} + 40X_{01,11,1} + 22X_{02,03,1} +$
$36X_{02,04,1} + 14X_{02,05,1} + 42X_{02,06,1} + 64X_{02,07,1} + 36X_{02,08,1} + 51X_{02,09,1} +$
$32X_{02,10,1} + 50X_{02,11,1} + 57X_{03,04,1} + 10X_{03,05,1} + 54X_{03,06,1} + 86X_{03,07,1} +$
$32X_{03,08,1} + 73X_{03,09,1} + 50X_{03,10,1} + 51X_{03,11,1} + 50X_{04,05,1} + 61X_{04,06,1} +$
$32X_{04,07,1} + 51X_{04,08,1} + 36X_{04,09,1} + 10X_{04,10,1} + 51X_{04,11,1} + 45X_{05,06,1} +$
$78X_{05,07,1} + 36X_{05,08,1} + 63X_{05,09,1} + 45X_{05,10,1} + 54X_{05,11,1} + 73X_{06,07,1} +$
$78X_{06,08,1} + 45X_{06,09,1} + 63X_{06,10,1} + 92X_{06,11,1} + 82X_{07,08,1} + 30X_{07,09,1} +$
$41X_{07,10,1} + 80X_{07,11,1} + 81X_{08,09,1} + 41X_{08,10,1} + 20X_{08,11,1} + 45X_{09,10,1} +$
$85X_{09,11,1} + 41X_{10,11,1} + 41X_{01,02,2} + 58X_{01,03,2} + 14X_{01,04,2} + 54X_{01,05,2} +$
$73X_{01,06,2} + 40X_{01,07,2} + 45X_{01,08,2} + 50X_{01,09,2} + 10X_{01,10,2} + 40X_{01,11,2} +$
$22X_{02,03,2} + 36X_{02,04,2} + 14X_{02,05,2} + 42X_{02,06,2} + 64X_{02,07,2} + 36X_{02,08,2} +$
$51X_{02,09,2} + 32X_{02,10,2} + 50X_{02,11,2} + 57X_{03,04,2} + 10X_{03,05,2} + 54X_{03,06,2} +$
$86X_{03,07,2} + 32X_{03,08,2} + 73X_{03,09,2} + 50X_{03,10,2} + 51X_{03,11,2} + 50X_{04,05,2} +$
$61X_{04,06,2} + 32X_{04,07,2} + 51X_{04,08,2} + 36X_{04,09,2} + 10X_{04,10,2} + 51X_{04,11,2} +$
$45X_{05,06,2} + 78X_{05,07,2} + 36X_{05,08,2} + 63X_{05,09,2} + 45X_{05,10,2} + 54X_{05,11,2} +$
$73X_{06,07,2} + 78X_{06,08,2} + 45X_{06,09,2} + 63X_{06,10,2} + 92X_{06,11,2} + 82X_{07,08,2} +$

$30X_{07,09,2} + 41X_{07,10,2} + 80X_{07,11,2} + 81X_{08,09,2} + 41X_{08,10,2} + 20X_{08,11,2} +$
$45X_{09,10,2} + 85X_{09,11,2} + 41X_{10,11,2}$

Subject to:

1. $X_{01,02,1} + X_{01,03,1} + X_{01,04,1} + X_{01,05,1} + X_{01,06,1} + X_{01,07,1} + X_{01,08,1} + X_{01,09,1} + X_{01,10,1} + X_{01,11,1} = 2$
2. $X_{01,02,1} + X_{02,03,1} + X_{02,04,1} + X_{02,05,1} + X_{02,06,1} + X_{02,07,1} + X_{02,08,1} + X_{02,09,1} + X_{02,10,1} + X_{02,11,1} = 2$
3. $X_{01,03,1} + X_{02,03,1} + X_{03,04,1} + X_{03,05,1} + X_{03,06,1} + X_{03,07,1} + X_{03,08,1} + X_{03,09,1} + X_{03,10,1} + X_{03,11,1} = 2$
4. $X_{01,04,1} + X_{02,04,1} + X_{03,04,1} + X_{04,05,1} + X_{04,06,1} + X_{04,07,1} + X_{04,08,1} + X_{04,09,1} + X_{04,10,1} + X_{04,11,1} = 2$
5. $X_{01,05,1} + X_{02,05,1} + X_{03,05,1} + X_{04,05,1} + X_{05,06,1} + X_{05,07,1} + X_{05,08,1} + X_{05,09,1} + X_{05,10,1} + X_{05,11,1} = 2$
6. $X_{01,02,2} + X_{01,03,2} + X_{01,04,2} + X_{01,05,2} + X_{01,06,2} + X_{01,07,2} + X_{01,08,2} + X_{01,09,2} + X_{01,10,2} + X_{01,11,2} = 2$
7. $X_{01,02,2} + X_{02,03,2} + X_{02,04,2} + X_{02,05,2} + X_{02,06,2} + X_{02,07,2} + X_{02,08,2} + X_{02,09,2} + X_{02,10,2} + X_{02,11,2} = 2$
8. $X_{01,03,2} + X_{02,03,2} + X_{03,04,2} + X_{03,05,2} + X_{03,06,2} + X_{03,07,2} + X_{03,08,2} + X_{03,09,2} + X_{03,10,2} + X_{03,11,2} = 2$
9. $X_{01,04,2} + X_{02,04,2} + X_{03,04,2} + X_{04,05,2} + X_{04,06,2} + X_{04,07,2} + X_{04,08,2} + X_{04,09,2} + X_{04,10,2} + X_{04,11,2} = 2$
10. $X_{01,05,2} + X_{02,05,2} + X_{03,05,2} + X_{04,05,2} + X_{05,06,2} + X_{05,07,2} + X_{05,08,2} + X_{05,09,2} + X_{05,10,2} + X_{05,11,2} = 2$
11. $X_{01,06,1} + X_{02,06,1} + X_{03,06,1} + X_{04,06,1} + X_{05,06,1} + X_{06,07,1} + X_{06,08,1} + X_{06,09,1} + X_{06,10,1} + X_{06,11,1} - 2Y_{06,1} = 0$
12. $X_{01,07,1} + X_{02,07,1} + X_{03,07,1} + X_{04,07,1} + X_{05,07,1} + X_{06,07,1} + X_{07,08,1} + X_{07,09,1} + X_{07,10,1} + X_{07,11,1} - 2Y_{07,1} = 0$
13. $X_{01,08,1} + X_{02,08,1} + X_{03,08,1} + X_{04,08,1} + X_{05,08,1} + X_{06,08,1} + X_{07,08,1} + X_{08,09,1} + X_{08,10,1} + X_{08,11,1} - 2Y_{08,1} = 0$
14. $X_{01,09,1} + X_{02,09,1} + X_{03,09,1} + X_{04,09,1} + X_{05,09,1} + X_{06,09,1} + X_{07,09,1} + X_{08,09,1} + X_{09,10,1} + X_{09,11,1} - 2Y_{09,1} = 0$
15. $X_{01,10,1} + X_{02,10,1} + X_{03,10,1} + X_{04,10,1} + X_{05,10,1} + X_{06,10,1} + X_{07,10,1} + X_{08,10,1} + X_{09,10,1} + X_{10,11,1} - 2Y_{10,1} = 0$
16. $X_{01,11,1} + X_{02,11,1} + X_{03,11,1} + X_{04,11,1} + X_{05,11,1} + X_{06,11,1} + X_{07,11,1} + X_{08,11,1} + X_{09,11,1} + X_{10,11,1} - 2Y_{11,1} = 0$
17. $X_{01,06,2} + X_{02,06,2} + X_{03,06,2} + X_{04,06,2} + X_{05,06,2} + X_{06,07,2} + X_{06,08,2} + X_{06,09,2} + X_{06,10,2} + X_{06,11,2} - 2Y_{06,2} = 0$
18. $X_{01,07,2} + X_{02,07,2} + X_{03,07,2} + X_{04,07,2} + X_{05,07,2} + X_{06,07,2} + X_{07,08,2} + X_{07,09,2} + X_{07,10,2} + X_{07,11,2} - 2Y_{07,2} = 0$
19. $X_{01,08,2} + X_{02,08,2} + X_{03,08,2} + X_{04,08,2} + X_{05,08,2} + X_{06,08,2} + X_{07,08,2} + X_{08,09,2} + X_{08,10,2} + X_{08,11,2} - 2Y_{08,2} = 0$
20. $X_{01,09,2} + X_{02,09,2} + X_{03,09,2} + X_{04,09,2} + X_{05,09,2} + X_{06,09,2} + X_{07,09,2} + X_{08,09,2} + X_{09,10,2} + X_{09,11,2} - 2Y_{09,2} = 0$
21. $X_{01,10,2} + X_{02,10,2} + X_{03,10,2} + X_{04,10,2} + X_{05,10,2} + X_{06,10,2} + X_{07,10,2} + X_{08,10,2} + X_{09,10,2} + X_{10,11,2} - 2Y_{10,2} = 0$
22. $X_{01,11,2} + X_{02,11,2} + X_{03,11,2} + X_{04,11,2} + X_{05,11,2} + X_{06,11,2} + X_{07,11,2} + X_{08,11,2} + X_{09,11,2} + X_{10,11,2} - 2Y_{11,2} = 0$

23. $Y_{06,1} + Y_{06,2} = 1$
24. $Y_{07,1} + Y_{07,2} = 1$
25. $Y_{08,1} + Y_{08,2} = 1$
26. $Y_{09,1} + Y_{09,2} = 1$
27. $Y_{10,1} + Y_{10,2} = 1$
28. $Y_{11,1} + Y_{11,2} = 1$

29. X_{ijk} and $Y_{ik} \in \{ 0,1 \}$

30. The solution is two tours, with nodes 1 to 5 on both tours, and nodes 6 to 11 on only 1 tour.

Constraints 1 to 10 require that nodes 1 to 5 are connected to each of the two tours by two arcs. Constraints 11 to 22 force nodes 6 to 11 to be connected by two arcs to only one tour. The tour to which these nodes are connected depends on which of the Y's have a value of 1.

Constraint 30 requires that any solution is consistent with the tour requirements. The number of constraints required to explicitly express constraint 30 is large. Thus, the solution methodology adopted is to ignore constraint 30, and then, on an " as needed " basis to add constraints to prevent violations.

The above problem, with constraint 30 ignored, is solved using the {0,1} algorithm of CPLEX. The solution is shown in Figures 3.1 and 3.2.

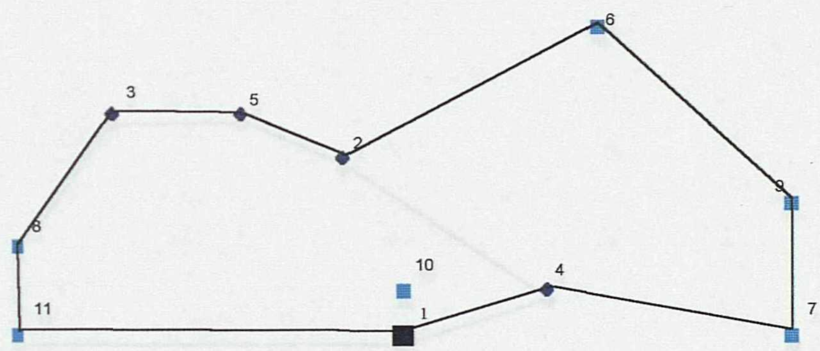


Figure 3.1 : 2-Matching Solution for Day_1

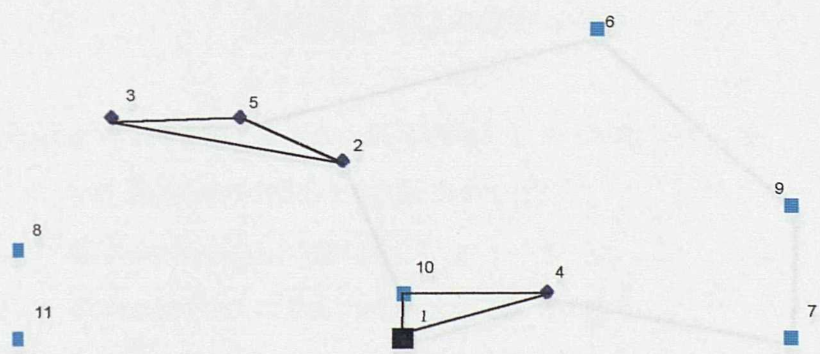


Figure 3.2 : 2-Matching Solution for Day_2

The solutions in Figures 3.1 and 3.2 have a total length of 359. The solution for Day_2 contains subtours. To prevent these subtours occurring in later solutions the following constraints are added to the model.

$$X_{01,04,2} + X_{01,10,2} + X_{04,10,2} \leq 2$$

$$X_{02,03,2} + X_{02,05,2} + X_{03,05,2} \leq 2$$

Because of the symmetry in the problem, the above constraints are duplicated to apply also to Day_1

$$X_{01,04,1} + X_{01,10,1} + X_{04,10,1} \leq 2$$

$$X_{02,03,1} + X_{02,05,1} + X_{03,05,1} \leq 2$$

Adding the 4 additional constraints to the model gives a new solution shown in Figures 3.3 and 3.4

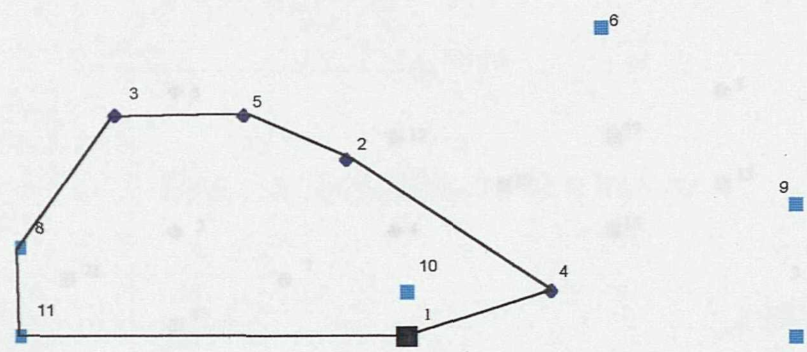


Figure 3.3 : Optimal Solution for Day_1

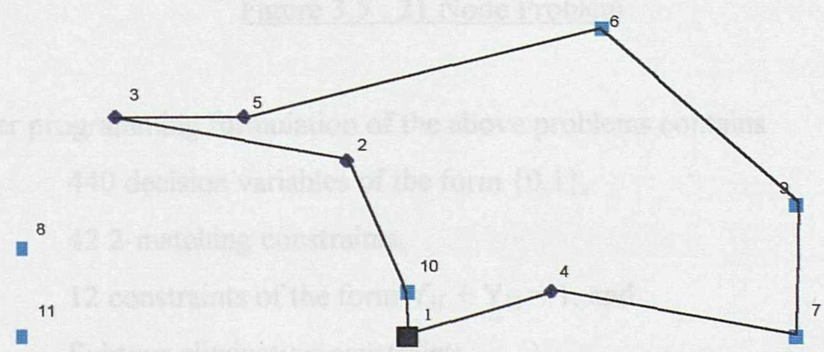


Figure 3.4 : Optimal Solution for Day_2

The solutions, shown in Figures 3.3 and 3.4, have a total tour length of 406. The solution satisfies the tour requirements, and is, therefore, the optimal solution.

The {0,1} programming facility within the CPLEX package solves the above problem is less than 1 second. For problems of equivalent size to the 11 node problem, the above solution methodology provides an optimal solution in reasonable time.

3.2 21 Node Problem

The 21 node problem consists of a depot, node number 1, 8 nodes to be visited by both tours, and 12 nodes to be visited by only one tour. The data for this problem is contained in Appendix 2, and the nodes are displayed in Figure 3.5.

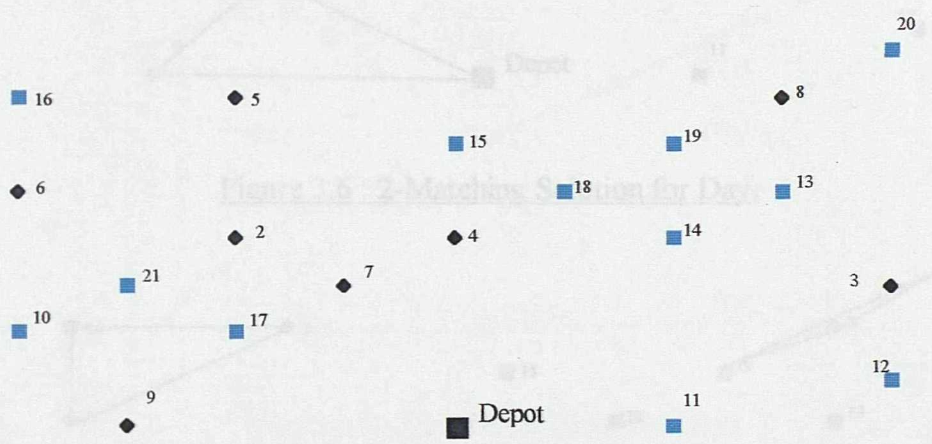


Figure 3.5 : 21 Node Problem

The integer programming formulation of the above problems contains

440 decision variables of the form {0,1},

42 2-matching constraints,

12 constraints of the form $Y_{i1} + Y_{i2} = 1$, and

Subtour elimination constraints.

The solution procedure adopted is to initially solve the {0,1} formulation with the subtour elimination constraints ignored. A loop is then entered during which subtour elimination constraints are added on an "as needed" basis, and the model resolved. The loop terminates when the solution meets all of the tour requirements.

The initial solution, the 2-matching solution, with all subtour elimination constraints ignored is shown in Figures 3.6 and 3.7. The objective value of this solution is 627.

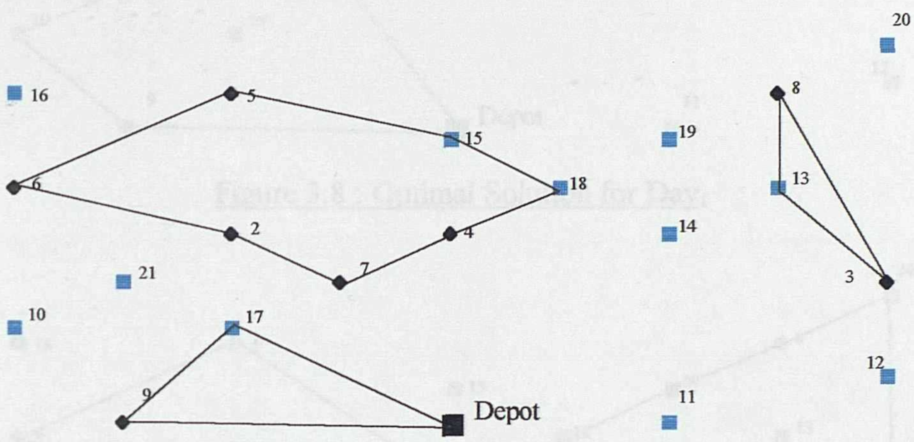


Figure 3.6 : 2-Matching Solution for Day_1

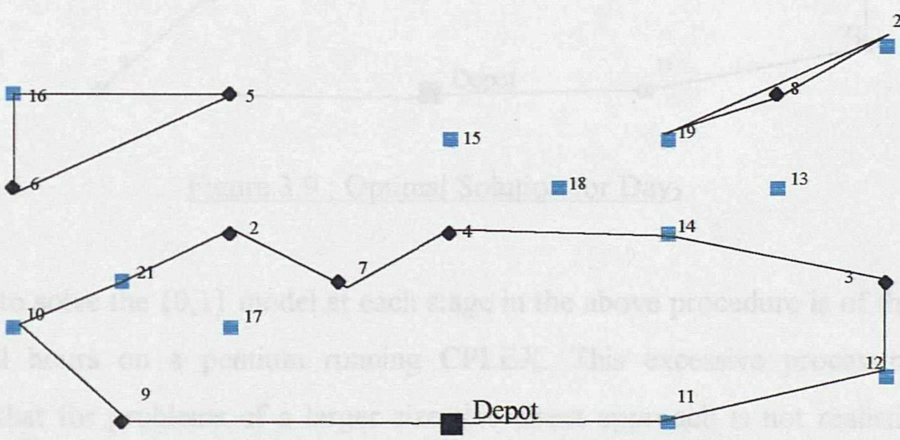


Figure 3.7 : 2-Matching Solution for Day_2

As Figures 3.6 and 3.7 show, the 2-matching solution contains subtours. Several iterations are required in which subtour elimination constraints are added on an "as needed" basis. Finally, a solution is obtained that satisfies the tour requirements. This solution with an objective value of 660 is shown in Figures 3.8 and 3.9.

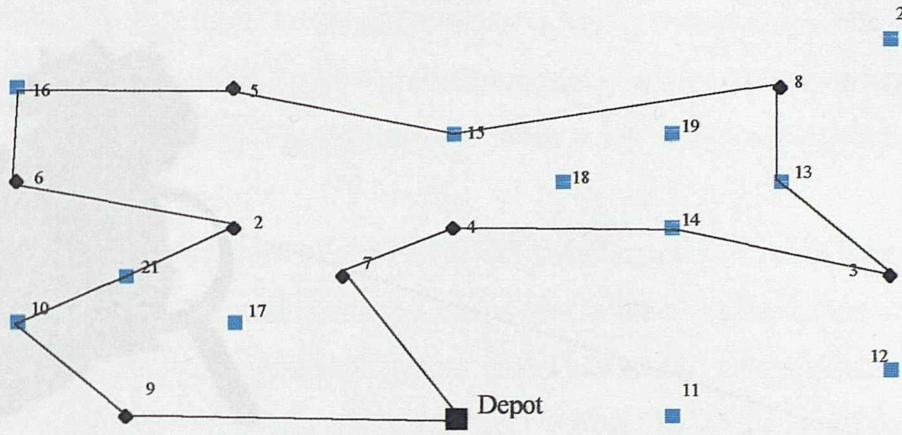


Figure 3.8 : Optimal Solution for Day_1

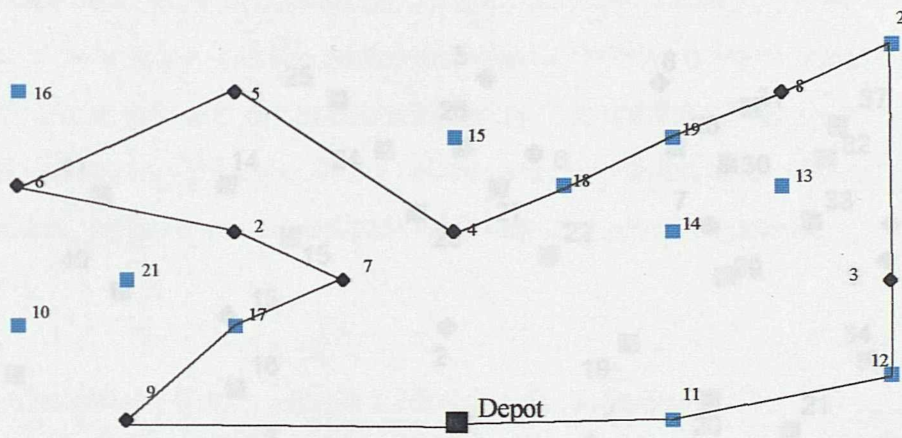
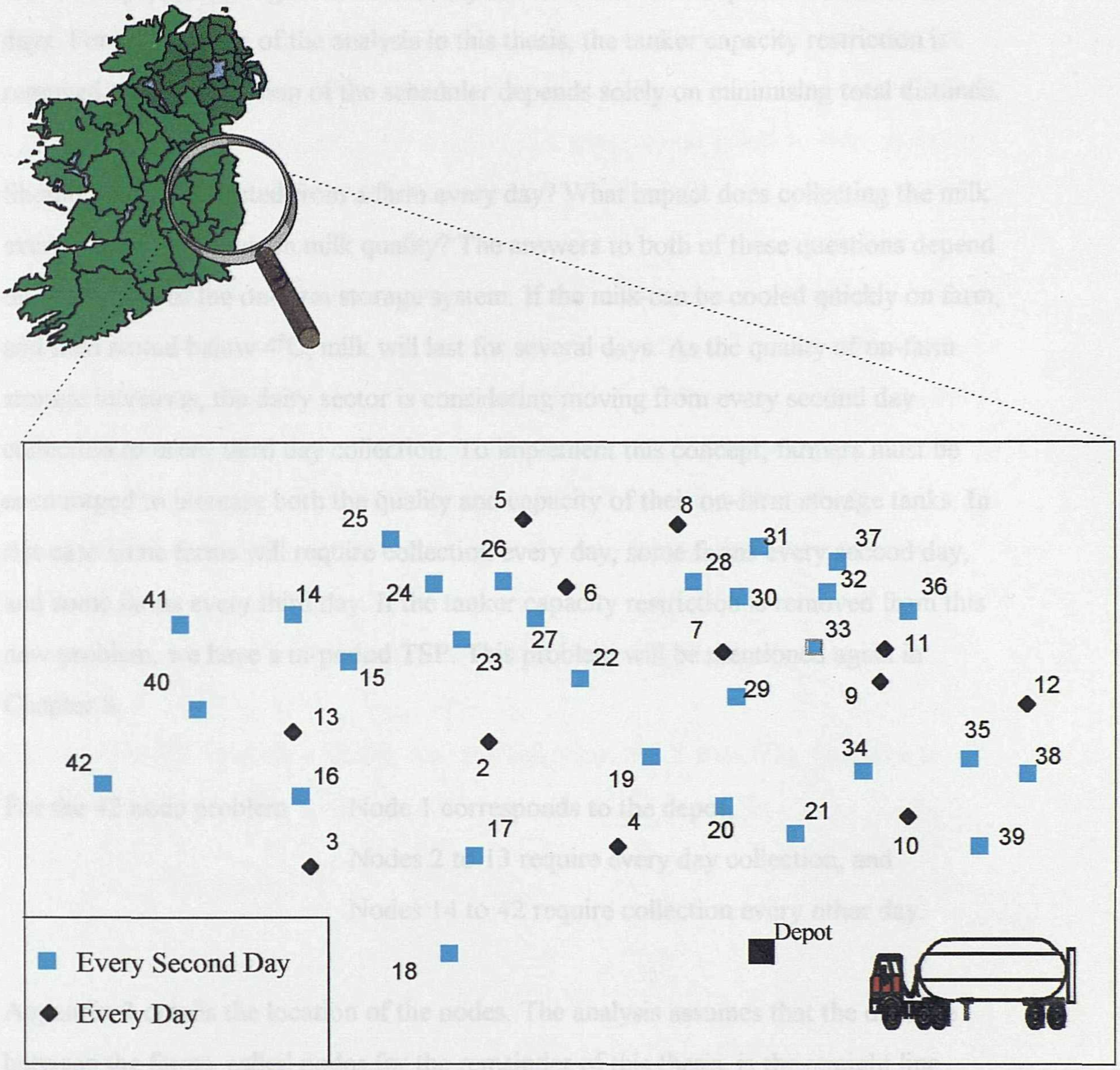


Figure 3.9 : Optimal Solution for Day_2

The time to solve the {0,1} model at each stage in the above procedure is of the order of several hours on a pentium running CPLEX. This excessive processing time suggests that for problems of a larger size this direct approach is not realistic. This indirect approach using LP relaxation is introduced in the solution of the 42 node problem.

3.3 42 Node Problem

The 42 node problem originates from the green fields surrounding Dublin, Ireland.



Here a cluster of 41 dairy farms are allocated to a milk tanker. Depending on the size of on-farm storage capacity, the milk from some farms must be collected every day, while other farms with sufficient storage to hold the milk output from two days, can be collected every other day. The scheduler creates two tours, and each tour is driven on alternate days. It is the scheduler's responsibility to allocate the every other day farms to one of the tours.

In practice the capacity of the tanker is of importance, and the scheduler attempts to allocate each every other day farm to a tour so as to minimise the total distance over the two days, having regard for the ability of the tanker to transport the milk on both days. For the purpose of the analysis in this thesis, the tanker capacity restriction is removed, and the decision of the scheduler depends solely on minimising total distance.

Should milk be collected from a farm every day? What impact does collecting the milk every second day have on milk quality? The answers to both of these questions depend on the quality of the on-farm storage system. If the milk can be cooled quickly on farm, and then stored below 4°C, milk will last for several days. As the quality of on-farm storage increases, the dairy sector is considering moving from every second day collection to every third day collection. To implement this concept, farmers must be encouraged to increase both the quality and capacity of their on-farm storage tanks. In this case some farms will require collection every day, some farms every second day, and some farms every third day. If the tanker capacity restriction is removed from this new problem, we have a m-period TSP. This problem will be mentioned again in Chapter 8.

For the 42 node problem Node 1 corresponds to the depot,
Nodes 2 to 13 require every day collection, and
Nodes 14 to 42 require collection every other day.

Appendix 3 details the location of the nodes. The analysis assumes that the distance between the farms, called nodes for the remainder of this thesis, is the straight line distance.

3.3.1 Solution of the 42 Node Problem

In Section 3.1 and Section 3.2 small versions of the 2-period TSP are solved directly using {0,1} programming. This approach requires excessive processing time for the 21 node problem. An attempt to directly solve the 2-matching {0,1} formulation of the 42 node problem was terminated after the CPLEX programme failed to find an optimal answer after several days continuous running on a Pentium PC.

This failure of the direct approach prompted the development of an more sophisticated procedure. This new approach solves increasingly constrained LP relaxations of the model. It is only when the LP relaxation satisfies all sub-tour and comb constraints that an attempt is made to solve the problem using integer programming. All timings quoted in this Section are obtained using the CPLEX package on a Pentium Pro PC running Windows NT. Figure 3.14 contains a flow diagram of the solution process.

3.3.2 Stage 1. 2-Matching Linear Programming Relaxation.

The starting LP relaxation is the one derived from the 2 matching constraints. The formulation of which is :

$$Minimise \sum_{i=1}^{41} \sum_{j=i+1}^{42} \sum_{k=1}^{2} C_{ij} X_{ijk}$$

Subject to

$$\sum_{j=1}^{i-1} X_{jik} + \sum_{j=i+1}^{42} X_{ijk} = 2, \qquad i = 1 \ldots 13, \quad k = 1, 2$$

$$\sum_{j=1}^{i-1} X_{jik} + \sum_{j=i+1}^{42} X_{ijk} - 2Y_{ik} = 0, \quad i = 14 \ldots 42, \; k = 1, 2$$

$$Y_{i1} + Y_{i2} = 1, \quad i = 14 \ldots 42$$

$$0 \leq X_{ijk} \; and \; Y_{ik} \leq 1$$

The above formulation consists of 1,722 X variables, 58 Y variables, and 113 constraints. The solution of this gives a value of 1,554 in a time of 0.27 seconds.

3.3.3 Stage 2. Constraint to remove the symmetry between Day_1 and Day_2.

The solution to the above LP relaxation is the same for the two days. As additional constraints are added to the formulation it is found that adding a constraint to prevent some condition for one of the days, simply causes that condition to occur on the other day. To break the symmetry in the problem, a constraint is added which forces Node 14 to be on the Day_1 tour. The form of this constraint is : $Y_{14,1} = 1$

The solution of this model gives a value of 1,568 in a time of 0.27 seconds.

3.3.4 Stage 3. Constraints of the form $X_{ijk} \leq Y_{ik}$ added iteratively.

The logic of the problem suggests that if node j, one of the every other day nodes, is not to be visited by tour k, then all of the X_{ijk} must equal zero. These requirements, named the VUB, variable upper bound, constraints by a friend and colleague Professor Paul Williams, are incorporated into the relaxation by the constraints :

$$X_{jik} \leq Y_{jk}, \quad i > j \, , \; j = 14 \ldots 42, \; k = 1,2$$
$$X_{ijk} \leq Y_{jk}, \quad i < j \, , \; j = 14 \ldots 42, \; k = 1,2$$

The power and importance of the above constraints are shown by the increase in the objective value solution that is obtained when the VUB constraints are added to the formulation. It will also be demonstrated later in this section that they are the difference between success and failure when one attempts to solve the problem using integer programming.

The number of VUB inequalities is quite large, and thus the approach is adopted whereby these constraints were added to the LP-relaxation on an "as needed" basis. In all 71 VUB constraints are violated and explicit constraints are added, and at the conclusion of this Stage a solution of value 1,694.5 was obtained in a time of 0.44 seconds.

Later in the process, as additional constraints were added to the LP-relaxation, the solutions are checked for VUB violations, and additional constraints are added as required.

3.3.5 Generalised Sub-tour Elimination Constraints

For the Standard Symmetric Travelling Salesman Problem, the sub-tour elimination constraint demands that

$$\sum_{i,j \in S} X_{ij} \leq n(S) - 1, \qquad \text{where S is a subset of the set of all nodes.}$$

To extend this sub-tour elimination concept to the 2-Period TSP requires a partition of S, a subset of all nodes, into S^1 and S^2, where S^2 contains nodes visited on both days, and S^1 contains nodes visited on only one day. The form of the generalised sub-tour elimination constraint depends on whether either S^1 or S^2 are empty. The three possible cases are:

Case 1 : $S^1 = \phi$, S contains only nodes visited on both days.

$$\sum_{i,j \in S} X_{ijk} \leq n(S^2) - 1, \qquad \text{for k = 1 or 2}$$

Case 2 : S^1 and $S^2 \neq \phi$, S contains both types of nodes.

$$\sum_{i,j \in S} X_{ijk} \leq \sum_{i \in S^1} Y_{ik} - n(S^2) - 1, \quad \text{for k = 1 or 2}$$

Case 3 : $S^2 = \phi$, S contains only nodes visited on one day.

$$\sum_{i,j \in S} X_{ijk} \leq \sum_{i \in S^1} Y_{ik} - \underset{i \in S^1}{Max}[Y_{ik}], \text{ for k = 1 or 2}$$

Consider the following examples that occur during the solution exercise.

Example 1 : $X_{5,8,1} = 1$, $X_{5,6,1} = 1$, and $X_{6,8,1} = 1$

This is a violation of a Case 1 type constraint, and the following inequalities are added to the problem :

$$X_{5,8,1} + X_{5,6,1} + X_{6,8,1} \leq 2$$

$$X_{5,8,2} + X_{5,6,2} + X_{6,8,2} \leq 2$$

Even though the violation only occurred on Day_1, a constraint is added for both days to avoid this portion of the solution jumping to the other day.

Example 2 : $X_{10,39,1} = 0.25$, $X_{10,34,1} = 0.25$, and $X_{34,39,1} = 0.25$, with $Y_{34,1} = 0.25$ and $Y_{39,1} = 0.25$.

The Case 2 inequalities added are :

$$X_{10,39,1} + X_{10,34,1} + X_{34,39,1} \leq Y_{34,1} + Y_{39,1}$$

$$X_{10,39,2} + X_{10,34,2} + X_{34,39,2} \leq Y_{34,2} + Y_{39,2}$$

A programme was written to automatically identify all circuits in the solution. For each circuit, the programme then sums the X values for all arcs in the circuit. Next, the programme checks whether the sum of the X's violates a subtour elimination constraint. Details of this programme that automatically detects violations of circuit based sub-tour constraints are given in Appendix 7. This programme is imbedded in a loop that consisted of solving the Linear Programme, and then using the automatic procedure both to detect violations and then to add the appropriate constraints. This process continues until no circuit based sub-tour violations exist. In addition to checking for sub-tour violations the programme adds, as required to prevent violations, VUB constraints.

In total the above loop is performed 7 times and 54 sub-tour elimination constraints together with an additional 6 VUB constraints are added.

The solution of the final loop gives a value of 1,711.8 in a time of 0.55 seconds.

This solution is shown in Figure 3.10. and the values are given in Appendix 4. The value on each arc is the sum of X_{ij1} and X_{ij2} for that arc. If no figure is associated with an arc, then $X_{ij1} + X_{ij2} = 1$ for that arc.

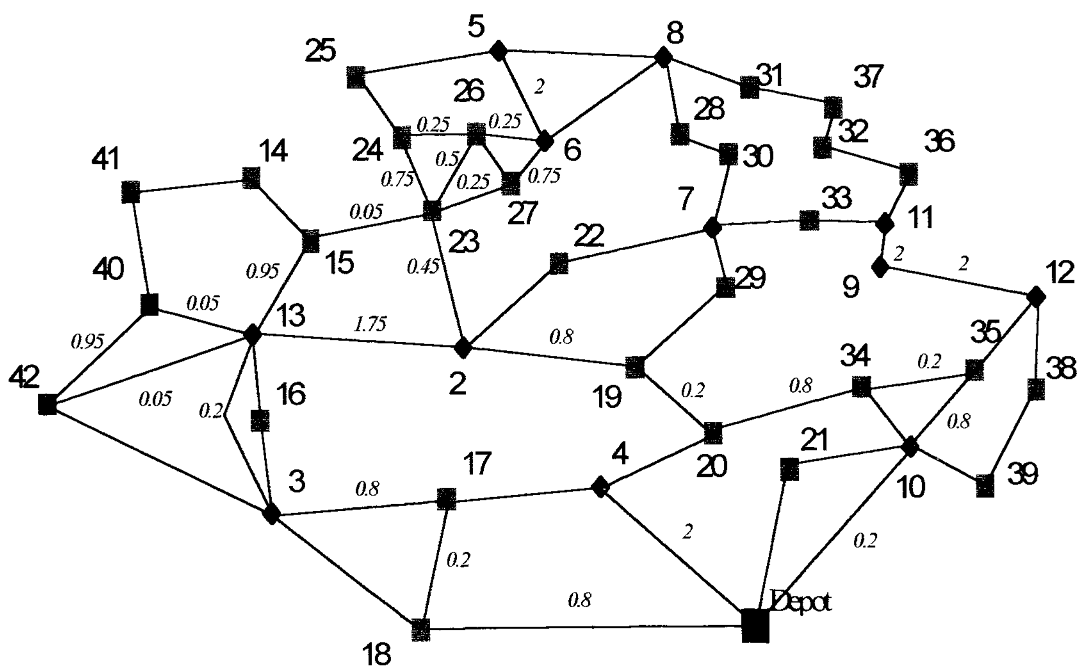
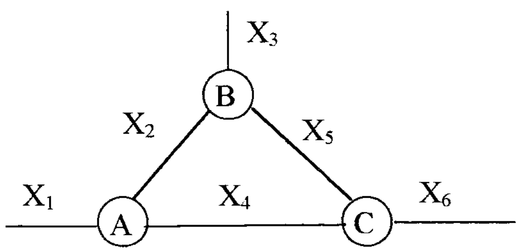


Figure 3.10 : LP Relaxation Solution after Stage 4.

3.3.6 Stage 5. Simple Comb Constraints.

At the end of Stage 4, the LP solution satisfies all generalised sub-tour inequalities. The next stage is to identify comb constraints that might be violated. For the STSP, the simplest of the combs will consist of a handle of 3 nodes, with a tooth of a single arc attached to each node. The constraint for this comb is derived as follows :



Eqn 1 :	Node A :	$X_1 + X_2 + X_4 \leq 2$
Eqn 2 :	Node B :	$X_2 + X_3 + X_5 \leq 2$
Eqn 3 :	Node C :	$X_4 + X_5 + X_6 \leq 2$
Eqn 4 :		$X_1 \leq 1$
Eqn 5 :		$X_3 \leq 1$
Eqn 6 :		$X_6 \leq 1$

Summing the above 6 equations gives :

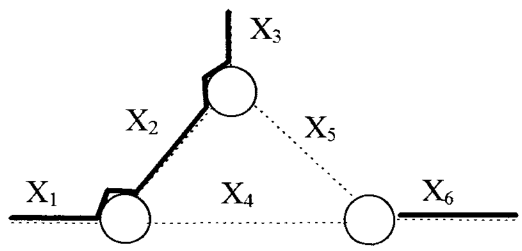
$$2\{ X_1 + X_2 + X_3 + X_4 + X_5 + X_6 \} \leq 9$$

Which implies, because of the integer requirement,

$$X_1 + X_2 + X_3 + X_4 + X_5 + X_6 \leq 4$$

The above proof was first suggested by Chvátal, and forms the simplest of the Chvátal Cuts.

One can also argue on intuitive grounds that any feasible solution to the above can use at most 4 of the above 6 arcs. In the diagram below the tour travels along X_1, X_2, and X_3. In addition, the tour travels along arc X_6. The use of any other arc is impossible in a feasible solution.



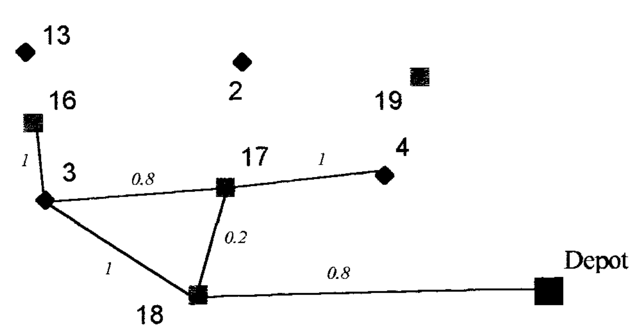
Extending the above to a handle of n nodes suggests :

$$\sum_{e \in Comb} X_e \leq n + \left\lfloor \frac{n}{2} \right\rfloor$$

where e ∈ comb refers to an arc in the comb.

The above inequalities applies to simple combs for which the tooth at each node consists of a single arc. Obviously more complicated structures exist. The comb concept is now extended to the to the 2-Period TSP, and the solutions are analysed on a case by case basis. Later in this section, a more general approached is adopted.

Consider the following portion of the solution shown in Figure 3.10.



The above figure shows a comb, in which the handle, H, consists of nodes 3, 17 and 18. The teeth consist of nodes 1, 4, and 16. The sum of arc values for this comb equals 4.8. The limit on the arc values for this comb can be derived as follows:

Day 1, Node 3 : $X_{3,16,1} + X_{3,17,1} + X_{3,18,1} \leq 2$

Day 1, Node 17: $X_{3,17,1} + X_{4,17,1} + X_{17,18,1} \leq 2Y_{17,1}$

Day 1, Node 18: $X_{1,18,1} + X_{3,18,1} + X_{17,18,1} \leq 2Y_{18,1}$

Day 2, Node 3 : $X_{3,16,2} + X_{3,17,2} + X_{3,18,2} \leq 2$

Day 2, Node 17: $X_{3,17,2} + X_{4,17,2} + X_{17,18,2} \leq 2Y_{17,2}$

Day 2, Node 18: $X_{1,18,2} + X_{3,18,2} + X_{17,18,2} \leq 2Y_{18,2}$

Arc 3 - 16: $X_{3,16,1} + X_{3,16,2} \leq 1$

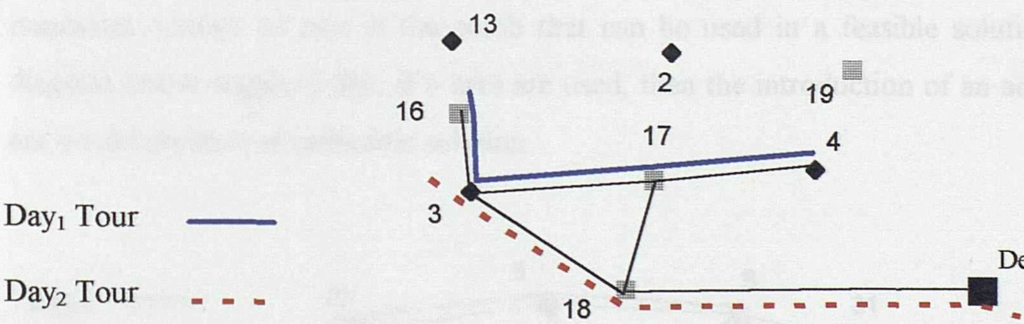
Arc 4 - 17: $X_{4,17,1} + X_{4,17,2} \leq 1$

Arc 1 - 18: $X_{1,18,1} + X_{1,18,2} \leq 1$

Adding the above equations suggests $$2 \sum_{e \in Comb} \sum_{k=1}^{2} X_{ek} \leq 11$$

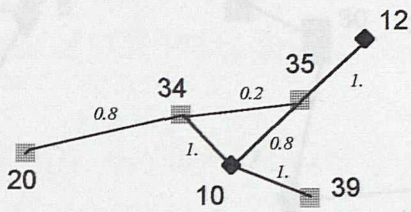
which implies $$\sum_{e \in Comb} \sum_{k=1}^{2} X_{ek} \leq 5$$

An examination of the comb confirms the result that at most 5 arcs in the comb can be used in a feasible solution. In the diagram below, the solid lines are arcs from a Day_1 tour, and the dashed lines are arc from a Day_2 tour. The use of any other arcs yields an infeasible solution.

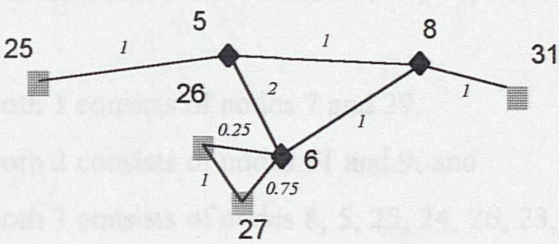


For the above comb, the actual sum of the X values equals 4.8. This value is less that the inequality limit of 5, and thus, in this instance, the comb constraint is not violated.

Examples of other Combs, contained in Figure 3.10., which are investigated are shown below :

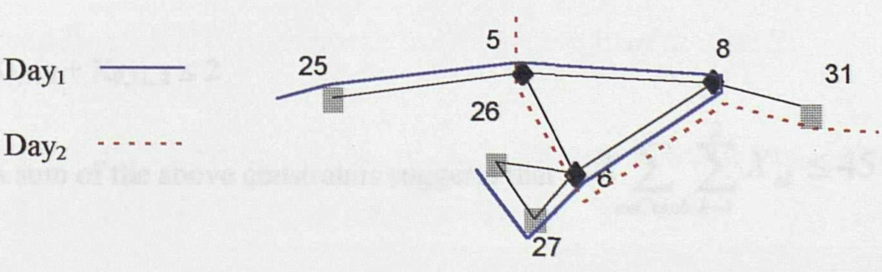


For the above comb $\sum_{e \in Comb} \sum_{k=1}^{2} X_{ek} = 4.8$ which satifies the limit of 5

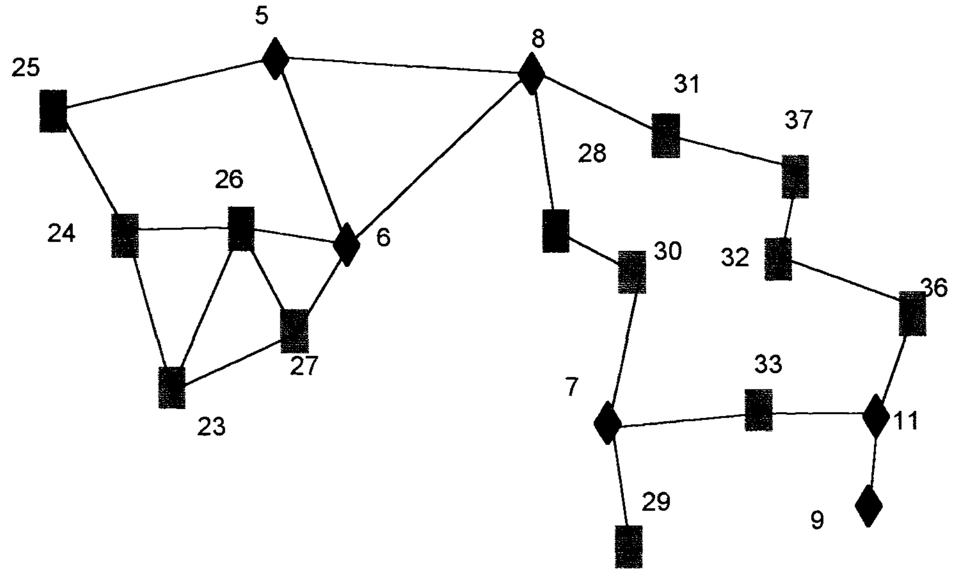


For the above comb, the sum of X's contained in the comb must be ≤ 8. In fact, the values add to 8, and thus the comb inequality is satisfied.

The limit of at most 8 for the above comb can be proved either by summing inequalities or by an intuitive argument. The intuitive argument is based on the maximum number of arcs in the comb that can be used in a feasible solution. The diagram below suggests that, if 8 arcs are used, then the introduction of an additional arc would produce an infeasible solution.



Consider the following portion of the solution from Figure 3.10.



In the above comb, the handle consists of nodes 8, 31, 37, 32, 36, 11, 33, 7, 30, and 28,

tooth 1 consists of nodes 7 and 29,

tooth 2 consists of nodes 11 and 9, and

tooth 3 consists of nodes 8, 5, 25, 24, 26, 23, 27, and 6.

The constraints for the comb are calculated as follows :

Sub tour elimination constraint for tooth 3 : $\Sigma X \leq 9$

Sub tour elimination constraint for tooth 3 less node 8 $\Sigma X \leq 7$

Node degree constraints for all nodes in the handle $\Sigma X \leq 26$

$X_{7,29,1} + X_{7,29,2} \leq 1$

$X_{9,11,1} + X_{9,11,,2} \leq 2$

A sum of the above constraints suggests that $2 \sum_{e \in Comb} \sum_{k=1}^{2} X_{ek} \leq 45$

thus, $\sum_{e \in Comb} \sum_{k=1}^{2} X_{ek} \leq 22$

The actual values for the above comb $\sum_{e \in Comb} \sum_{k=1}^{2} X_{ek} = 22.75$.

A constraint limiting the sum to 22 was added to the formulation and the model resolved. The solution gave a value of 1713.19 in a time of 0.55 seconds.

This solution is shown in Figure 3.11. and the values are given in Appendix 5. The value on each arc is the sum of X_{ij1} and X_{ij2} for that arc. If no figure is associated with an arc, then $X_{ij1} + X_{ij2} = 1$ for that arc

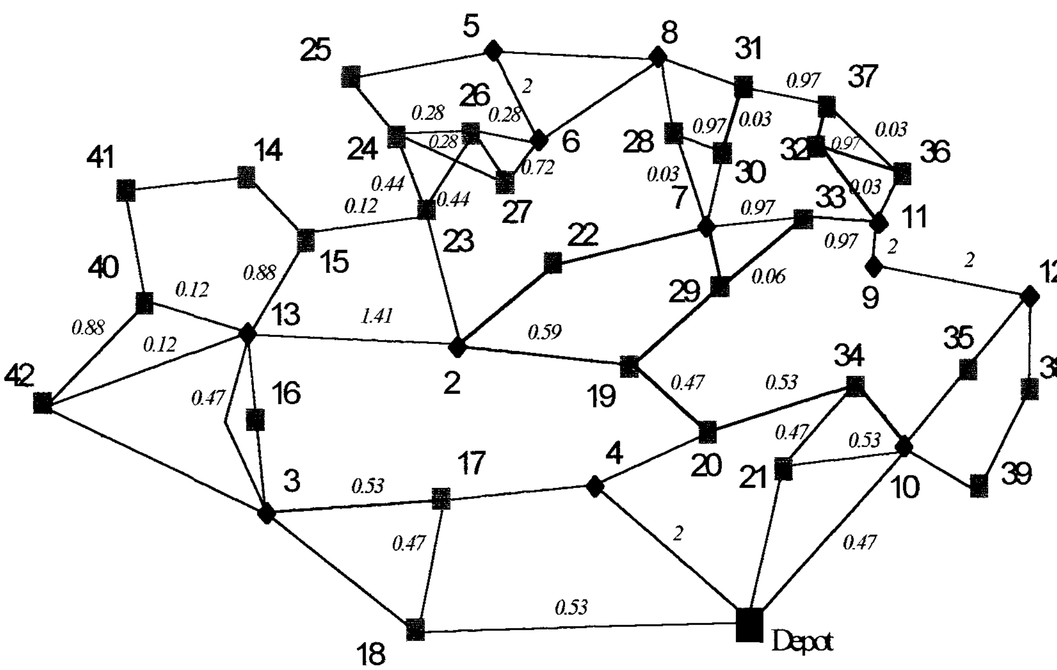


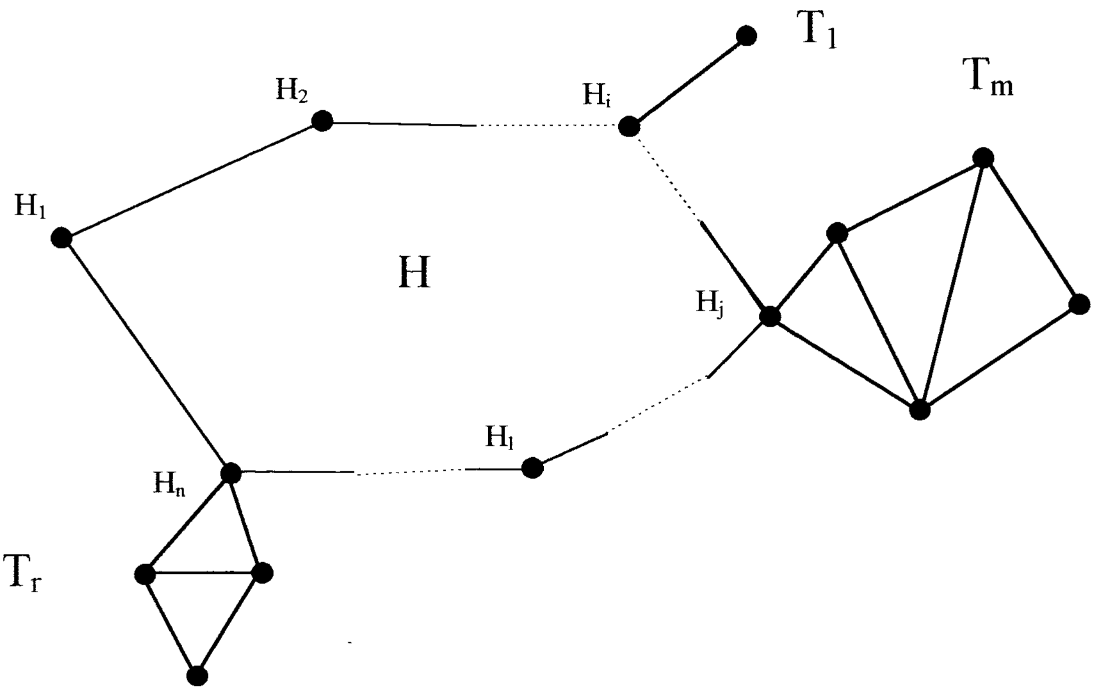
Figure 3.11 : LP Relaxation Solution after 1st set of Comb Constraints.

3.3.7 Derivation of the General Comb Constraints

Consider a comb consisting of a handle, H, and a set of teeth $T_1, T_2, \ldots, T_r$.

H consists of n nodes $\{H_1, H_2, \ldots H_n\}$. Of these n nodes some are every day nodes, and the remainder are every other day nodes.

Teeth are attached to the handle at various nodes, as shown below:



For each tooth in the comb, 2 constraints are derived. Constraint 1 is a subtour elimination constraint for the nodes in the tooth. Constraint 2 is a subtour elimination constraint for the nodes in the tooth with the node connecting the tooth to the handle excluded. The form of these constraints depends on whether the comb contains any every day nodes.

Consider the tooth, T_m, connected to the handle at node H_j, :

Constraint 1 : $\sum_{ij \in T_m} \sum_{k=1}^{2} X_{ijk} \leq$ 2 * (Number of Every Day Nodes in T_m) + (Number of Every Other Day Nodes in T_m) - 2 ; if T_m contains some Every Day Nodes.

$\sum_{ij \in T_m} \sum_{k=1}^{2} X_{ijk} \leq$ (Number of Every Other Day Nodes in T_m) - 1 ; if T_m contains no Every Day Nodes.

Constraint 2 : $$\sum_{ij \in T_m - H_j} \sum_{k=1}^{2} X_{ijk} \leq 2 * (\text{Number of Every Day Nodes in } T_m - H_j) + (\text{Number of Every Other Day Nodes in } T_m - H_j) - 2 \text{ ; if } T_m - H_j \text{ contains some Every Day Nodes.}$$

$$\sum_{ij \in T_m - H_j} \sum_{k=1}^{2} X_{ijk} \leq (\text{Number of Every Other Day Nodes in } T_m - H_j) - 1 \text{ ; if } T_m - H_j \text{ contains no Every Day Nodes.}$$

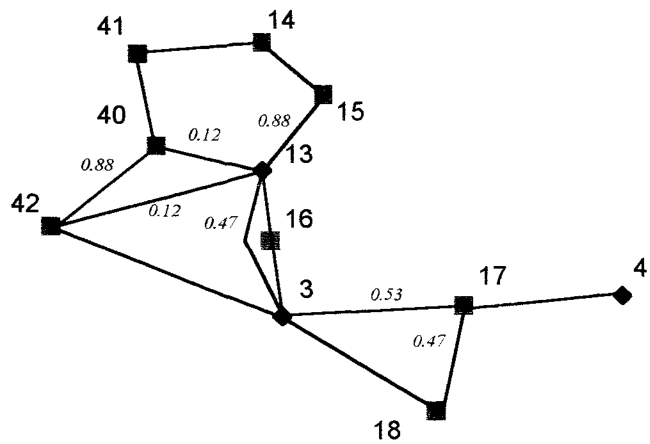
For every node in the handle a node degree constraint is generated. These constraints suggest that :

$$\sum_{\substack{\text{All arcs connect} \\ \text{to the node}}} \sum_{k=1}^{2} X_{ijk} \leq 4 \text{ ; if the node is an Every Day Node.}$$

$$\sum_{\substack{\text{All arcs connect} \\ \text{to the node}}} \sum_{k=1}^{2} X_{ijk} \leq 2 \text{ ; if the node is an Every Other Day Node.}$$

The complete comb constraint is achieved by adding the constraints of types 1 and 2 for all the teeth with the node degree constraints for all of the nodes in the handle. The left-hand side of this addition is twice the sum of all the arcs in the comb, while the right-hand side is a real number. The strength of the comb constraint is achieved if the right-hand side total is an odd number. The comb inequality is then achieved by dividing the sum by 2 and rounding down.

The following portion of the solution in Figure 3.11. is analysed using the above approach as follows :



Handle = { 3, 17, 18 }

T_1 = { 17,4 }

T_2 = { 3, 13, 14, 15, 16, 40, 41, 42 }

For tooth 1 :

Constraint 1 has right-hand side = 8

Constraint 2 has right-hand side = 6

For tooth 2:

Constraint 1 has right-hand side = 1

Constraint 2 has right-hand side = 0

For the Handle

Sum of node degrees ≤ 8

The addition of the above right-hand sides comes to 23. This divided by 2 and rounded down provides 11 as a limit of the sum of the arcs in the comb. The actual solution values with a sum of 11.47 violate this condition. A comb constraint preventing this comb occurring in a later solution was added and the formulation resolved. The solution gave a value of 1715.21 in a time of 0.55 seconds.

Additional effort could be spent identifying further comb constraints. However, an attempt at solving the problem using integer programming proved successful and the search for comb constraints was terminated.

3.3.8 Integer Programming Solution of the LP Model

On completion of Stages 1 to 5, the Linear Programme consisted of 1780 variables and 237 constraints. A complete listing of the constraints added in Stages 2, 3, 4 and 5 is shown in Appendix 8.

This LP is a relaxation of the original problem in which both the X and Y variables must take values from {0, 1}. Introducing this restriction on the X and Y variables and solving the IP by CPLEX gave an optimal solution, with no subtours, of 1,725 in a time of 25.54 seconds.

This solution is shown in Figures 3.12 and 3.13 and the values are given in Appendix 6.

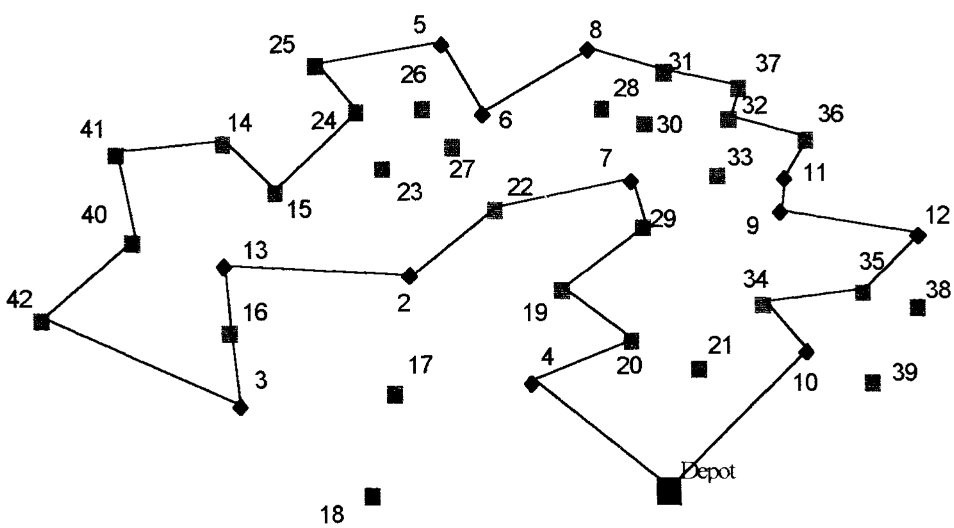


Figure 3.12 : Optimal Tour for Day$_1$.

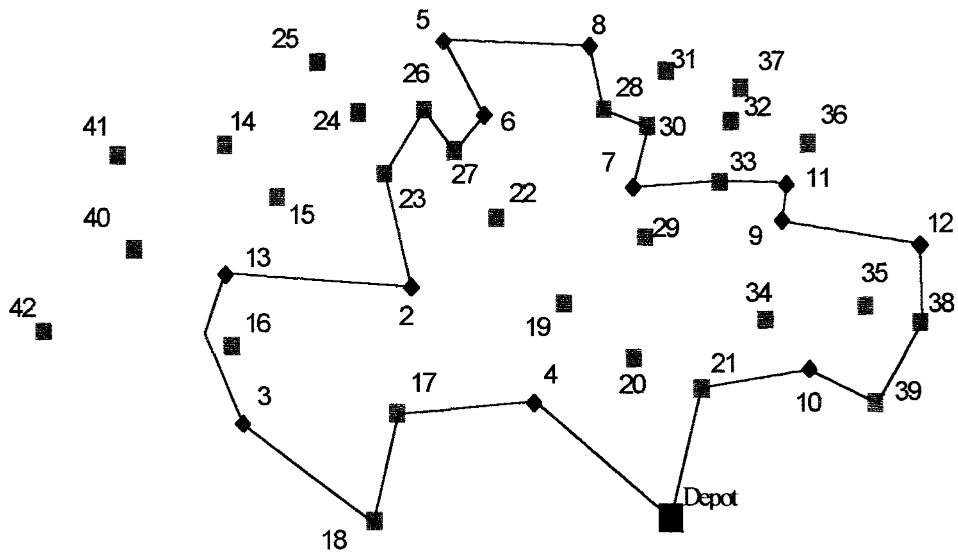


Figure 3.13 : Optimal Tour for Day_2.

The success of the IP solution at this stage is in contrast with an attempt to solve the original integer version of the 2 matching formulation using CPLEX. Table 3.1 shows the results of attempts at solving the IP at various points in the solution process.

Table 3.1 shows the impact of

- the additional VUB constraints,
- subtour elimination constraints, and
- comb constraints

on the ability of CPLEX to solve the IP problem.

Model Stage	Processing time (Secs)	Solution	Nodes	Iterations
2-Matching	-	Failed	-	-
2-Matching plus VUB	56.03	1,709	3,773	17,630
2-Matching plus VUB and Subtour Elimination	43.12	1,725	1,995	25,881
2-Matching plus VUB, Subtour Elimination and Comb Constraints.	25.54	1,725	1,106	15,256

Table 3.1 : Integer Programming Solutions using CPLEX.

The above table verifies the major importance of the VUB constraints in any attempt to solve this problem using Integer Programming.

Figure 3.14 represents the flow through the solution approach used to solve the 42-node problem.

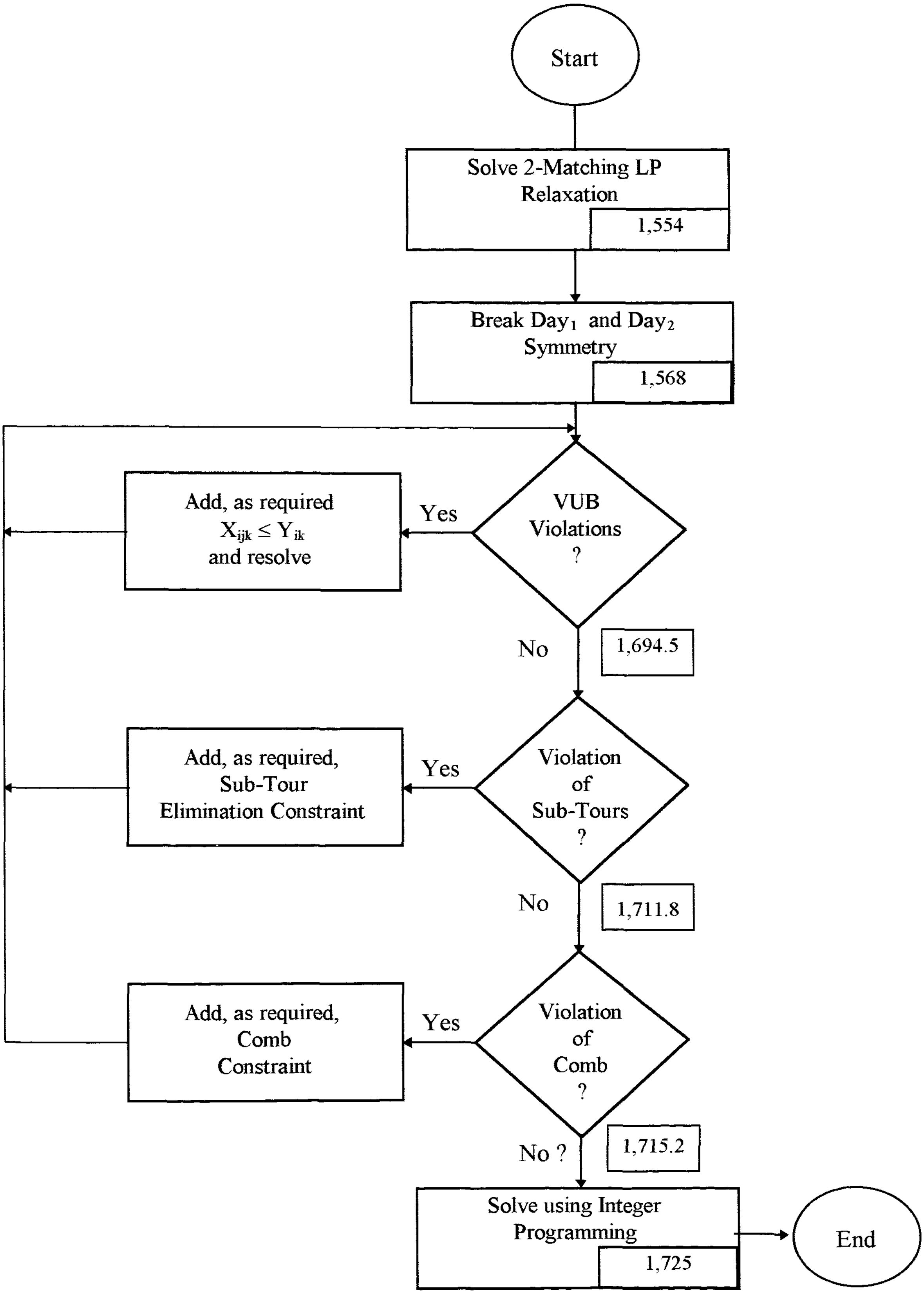


Figure 3.14 Solution Process for the 42 node problem

3.4 Conclusion

The 11 node 2-period TSP can be realistically solved using $\{0,1\}$ integer programming directly, with the subtour elimination constraints added on an "as needed" basis. This direct approached falters, but with perseverance can be made to succeed, for the 21 node problem.

The direct approach is unrealistic for the 42 node problem, and an approach based on successive LP relaxations provides an optimal solution. Within this approach, the VUB constraints are the difference between success and failure. Their addition to the 2-matching LP relaxation moves the problem forward to a stage where direct $\{0,1\}$ programming will optimally solve the problem. Generalised subtour and comb constraints also help in moving the LP relaxation closer to the optimal answer, but not to the same extent as the VUB constraints.

A version of the solution methodology for the 42-node problem is published in Butler, Williams and Yarrow [1997]. The approach in that paper differs marginally from the approach used in this thesis. In the paper the VUB constraints are added at each stage on an "as needed" basis. While, in this thesis, a VUB constraint, once added, remains in the model.

Chapter 4. Optimal Solution of the 100 Node Problem.

This chapter describes the procedure to find an optimal solution to an 100 node example of the 2-Period TSP. The 100 nodes consists of 40 nodes that are to be serviced by both tours, and 60 nodes that are to be serviced by only one tour. The data for this problem is contained in Appendix 9

4.1 Solution Approach

Chapter 3 describes a successful attempt to solve the 42-node problem. There, the solution procedure consists of a staged approached in which an increasingly constrained LP relaxation is solved. When the LP satisfies all VUB, subtour and comb constraints, then {0,1} programming is used to provide the optimal solution. It was initially hoped that this philosophy could be applied to the 100 node problem. However, as is described below, and as should have been expected with any combinatorial problem, the final solution proved extremely elusive, and as soon as one felt that the optimal solution was in sight, then the problem seemed to regain control, and in the space of one step the solution seemed as far away as ever. Professor Paul Williams described the solution seeking exercise as *similar to nailing jelly to a wall - just as it seemed that it was finally stable, a new area would ooze out, and .....*

4.2 2-Matching and VUB Constraints

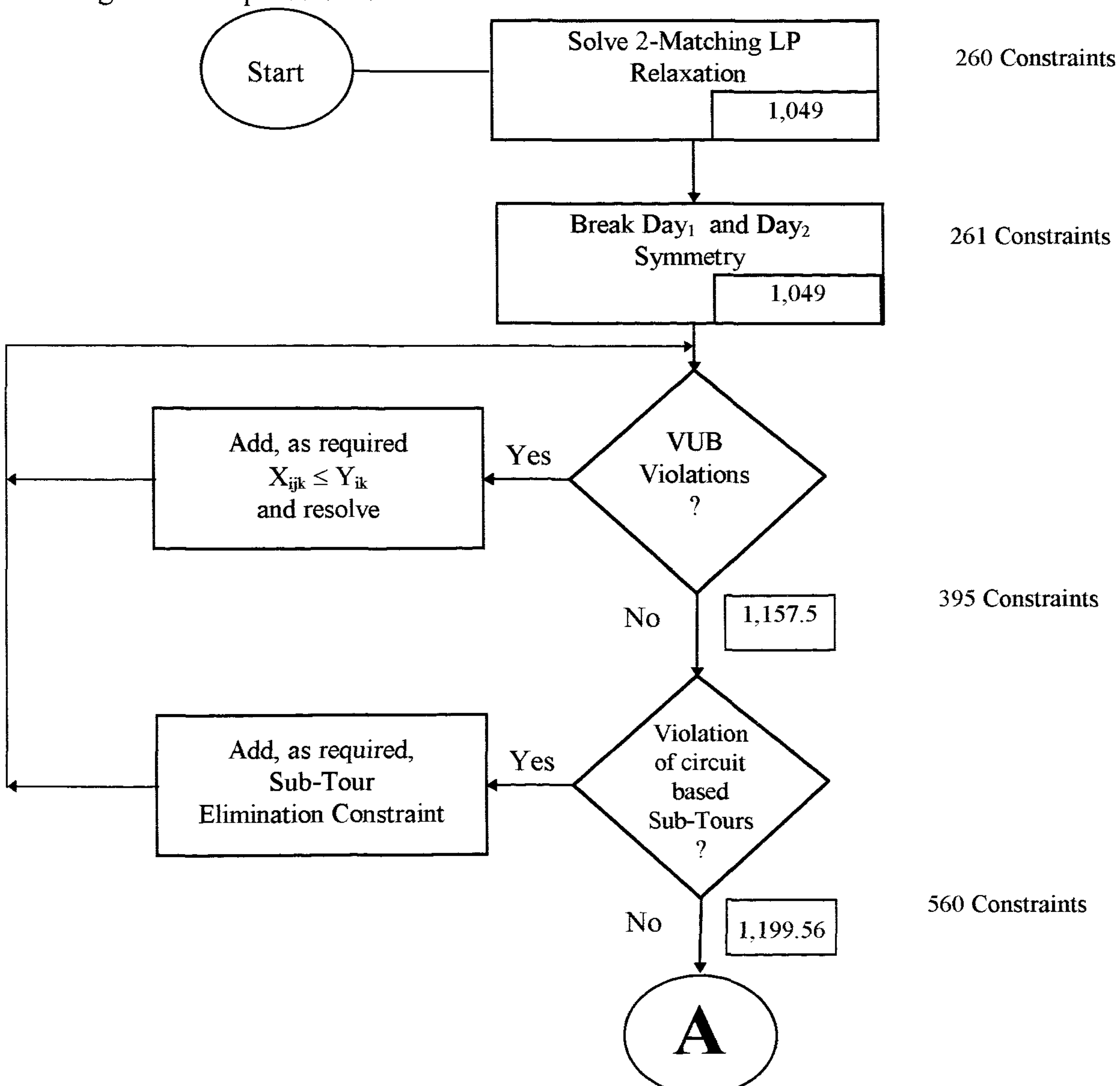
The 2 matching LP relaxation of the 100 node problem consists of 9,900 X variables, 120 Y variables and 260 constraints. This LP gave an objective value solution of 1,049 in a time of 1.48 seconds.

Following the methodology developed for the 42 node problem, a constraint is next added to break the symmetry of the problem. The LP solution at this stage contains VUB violations. Through a process of resolving the model and adding VUB constraints as needed, a solution is found that violates no VUB constraint. The resulting LP contains 395 constraints with a solution of 1,157.5. The above increase in the objective function from 1,049 to 1,157.5 once again shows the power and importance of the VUB constraints. Unfortunately, it was later found that no other class of constraint yields such an increase in the objective function.

4.3 Subtour Elimination Constraints

As described in Chapter 3, a computer program was written to identify and introduce the necessary preventative constraints for circuit based subtour violations. In all 165 subtour elimination constraints are added to the model. At the end of this stage, the LP model consists of 560 constraints, and solves with a value of 1,199.56.

A diagram of the process is :



At point **A**, above, the LP model contains 10,020 variables and 560 constraints. The objective value solution of this LP is 1,199.56. This solution contains fractional values for some of the arcs. The best that can be said is that the value of 1,199.56 is a lower bound on the optimal solution.

Three possible options were considered for the way forward from point **A**. These are :

1. Add the restrictions X_{ijk} and $Y_{ik} \in \{0,1\}$ and attempt to solve.

2. Add the restriction $Y_{ik} \in \{0,1\}$, while leaving $0 \leq X_{ijk} \leq 1$, and continue.

3. Increase the complexity of the LP relaxation by identifying firstly, non-circuit based sub-tour violations and then comb constraint violations.

Obviously, option 1 is the simplest. But realistically, option 1 is doomed to failure and so it proved. The complexity of the problem totally defeated CPLEX on a Pentium Pro. The CPLEX package aborted after 90,000 iterations when the 32 megabytes of core memory became inadequate to hold the solution tree.

4.4 Heuristic Answer I

Before failure, CPLEX found an integer solution, with subtours, of 1,213. This solution is not proven to be optimal. Analysis of this solution gave a partition of the Y's across the two tours. This might not be the optimal partition, but maybe it could be used as part of a heuristic solution to find a good upper bound. This detour into a heuristic solution proceeds as follows:

1. The LP model at point **A** is restricted by assigning the Y variables to the values suggested by the above partition.

2. The X_{ijk} are restricted to $\{0,1\}$.

3. The IP formulation is solved using CPLEX. In the solution sub-tours are identified and constraints are added to prevent their re-occurrence. The model is continually resolved until two feasible tours are found.

In the above heuristic, because the Y variables are pinned, CPLEX is solving 2 independent TSP problems. The final run produces an objective value solution of 1,225 in a time of 5.33 seconds. These tours are shown in Figures 4.1 and 4.2. This heuristic solution is considerably better that the previous best heuristic of 1,278.

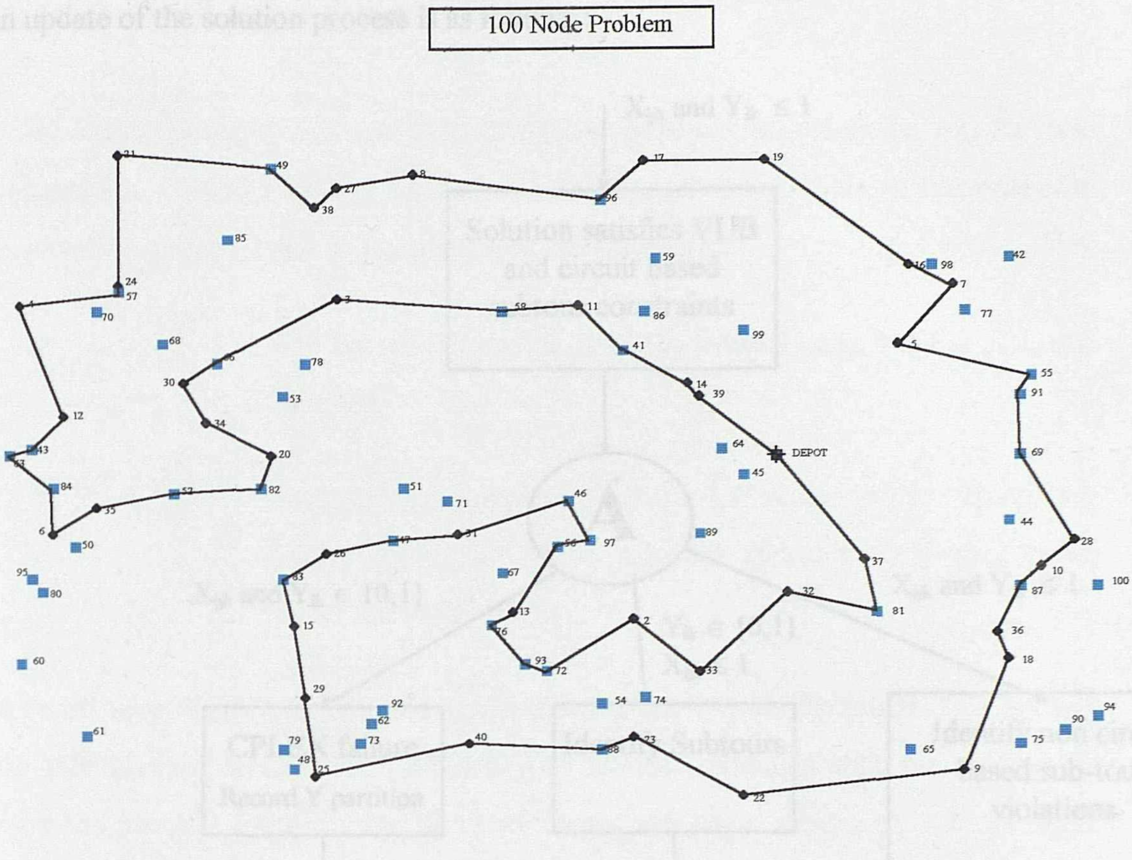


Figure 4.1. Day_1 tour of heuristic solution of total length 1,225

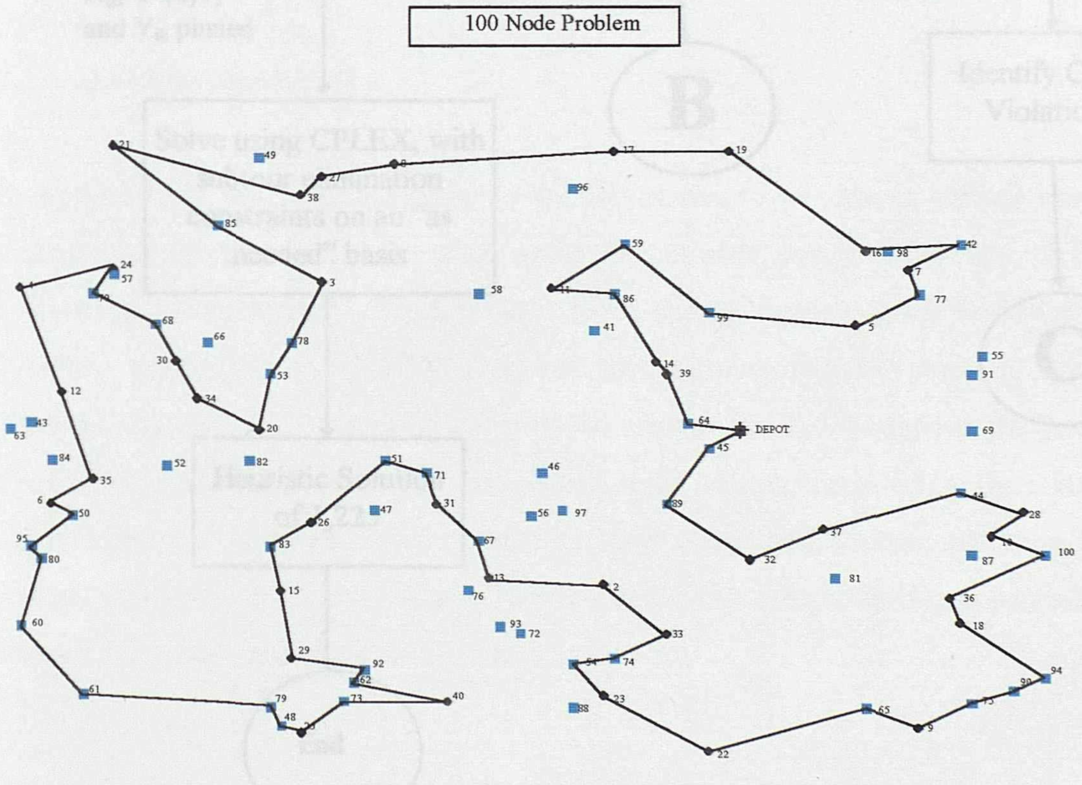
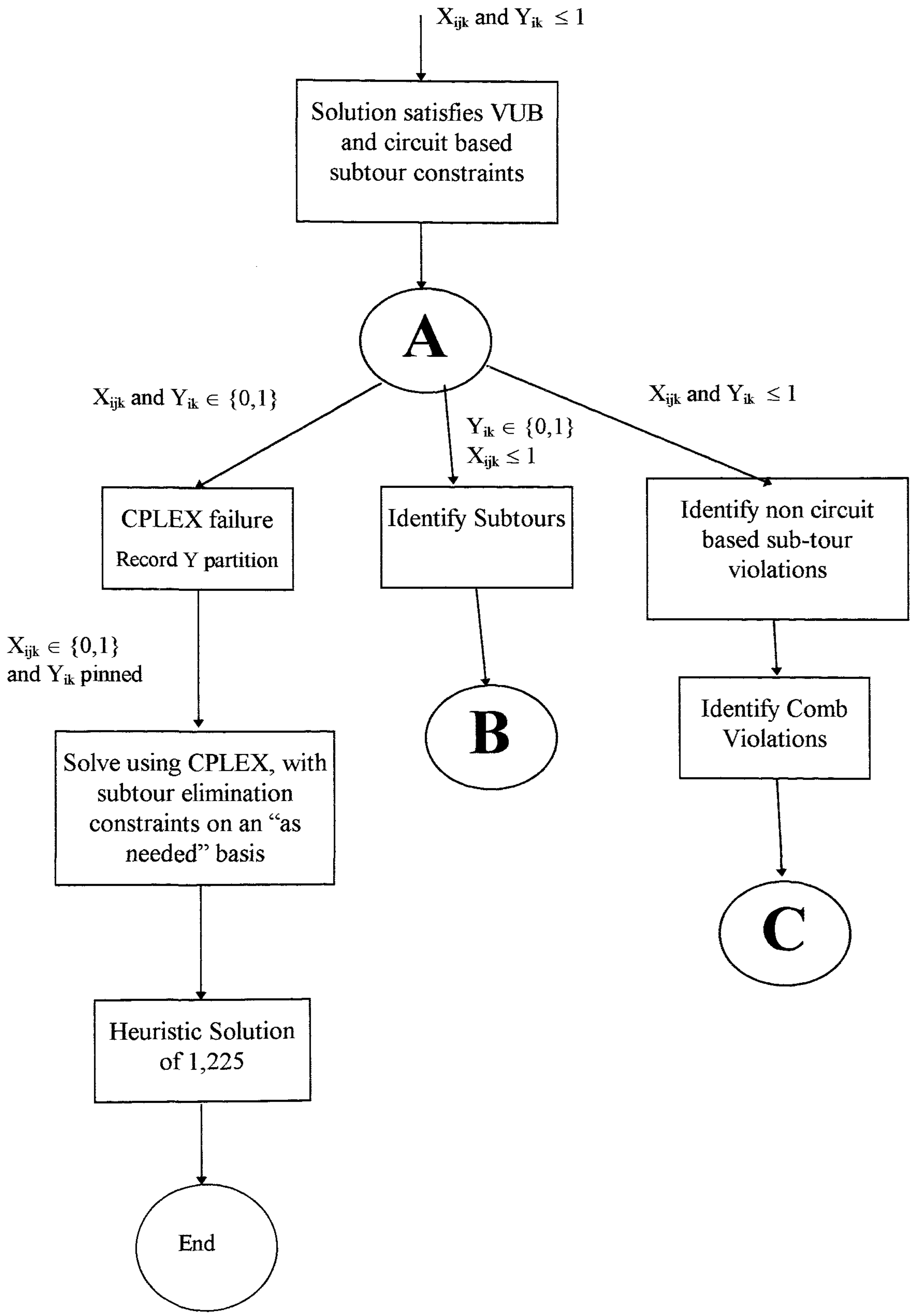


Figure 4.2. Day_2 tour of heuristic solution of total length 1,225

An update of the solution process is as follows :



4.5 Y's restricted to {0,1}

Earlier in this section it was stated that 3 options were considered as the way forward from point **A**. Option 1 led to a heuristic solution. Points **B** and **C** above correspond to the previous options 2 and 3. These options are restated as :

2. Add the restriction $Y_{ik} \in \{0,1\}$, while leaving $0 \leq X_{ijk} \leq 1$, and continue.

3. Increase the complexity of the LP relaxation by identifying firstly, non-circuit based sub-tour violations and then comb constraint violations.

The road from **A** to **B,** above, involves restricting the Y variables to {0, 1 }. Solving this MIP gives a solution of 1,202.67. This solution contains subtours. Using the automatic procedure for eliminating circuit based sub-tours subtours yields the same objective value solution of 1,202.67.

4.6 Heuristic Answer II

The options from this point are either to use IP, or else to attempt to identify non-circuit based subtour violations. Once again the simplest option is to use {0,1} programming. Unfortunately, CPLEX again failed due to memory space limitations. However, before failure, CPLEX produced an unproven integer solution, with subtours, of 1,204. As is described earlier in this section, this 1,204 becomes the basis of a heuristic solution. The Y values are pinned at the values suggested by the 1,204 solution, and the X variables are optimised. After eliminating several sub-tours, a heuristic solution of 1,224 is obtained. This is 1 better that the previous best heuristic solution. This solution is shown in Figures 4.3 and 4.4.

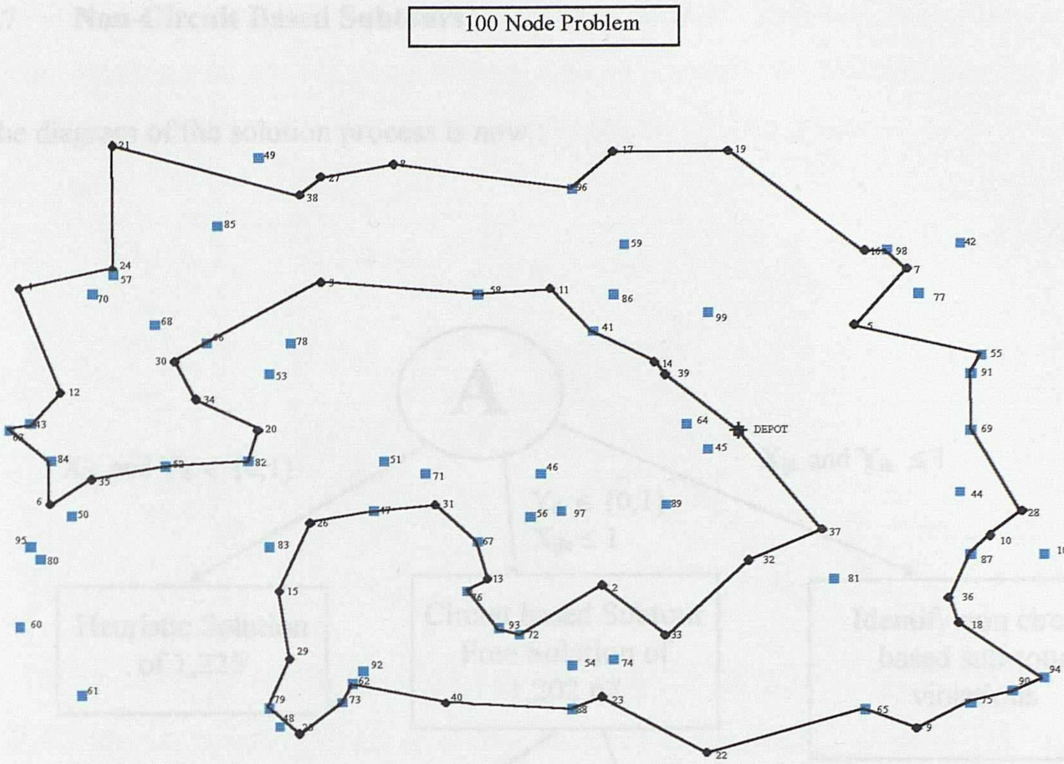


Figure 4.3 : Day_1 tour of heuristic solution of total length 1,224

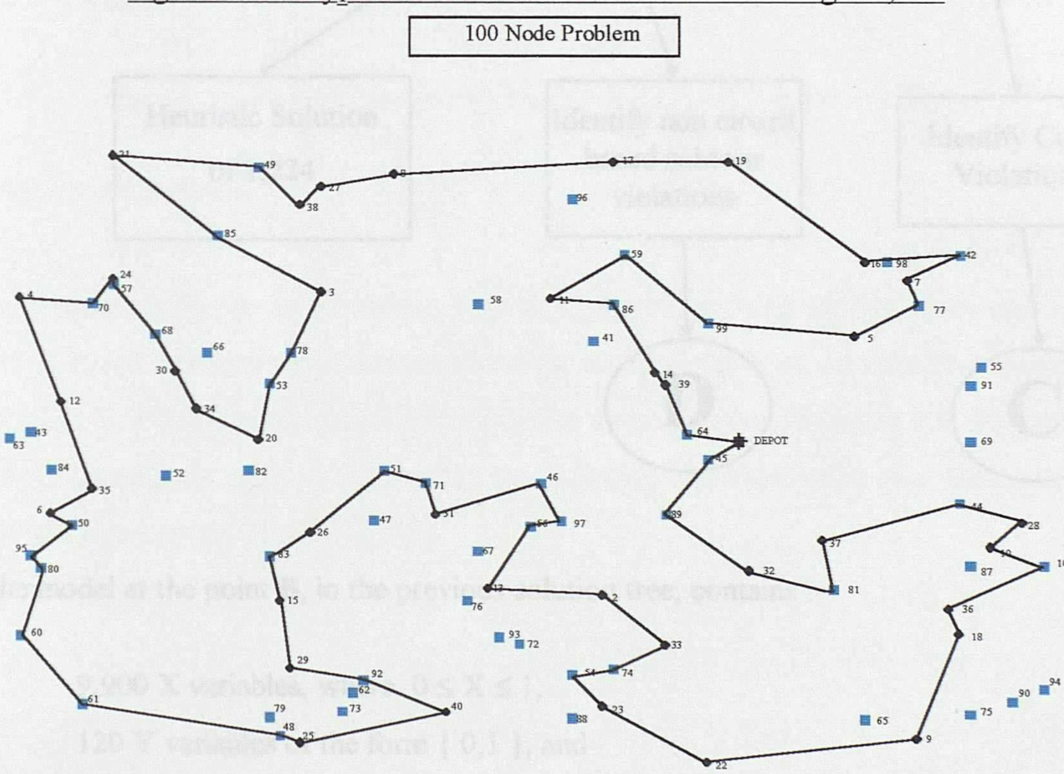
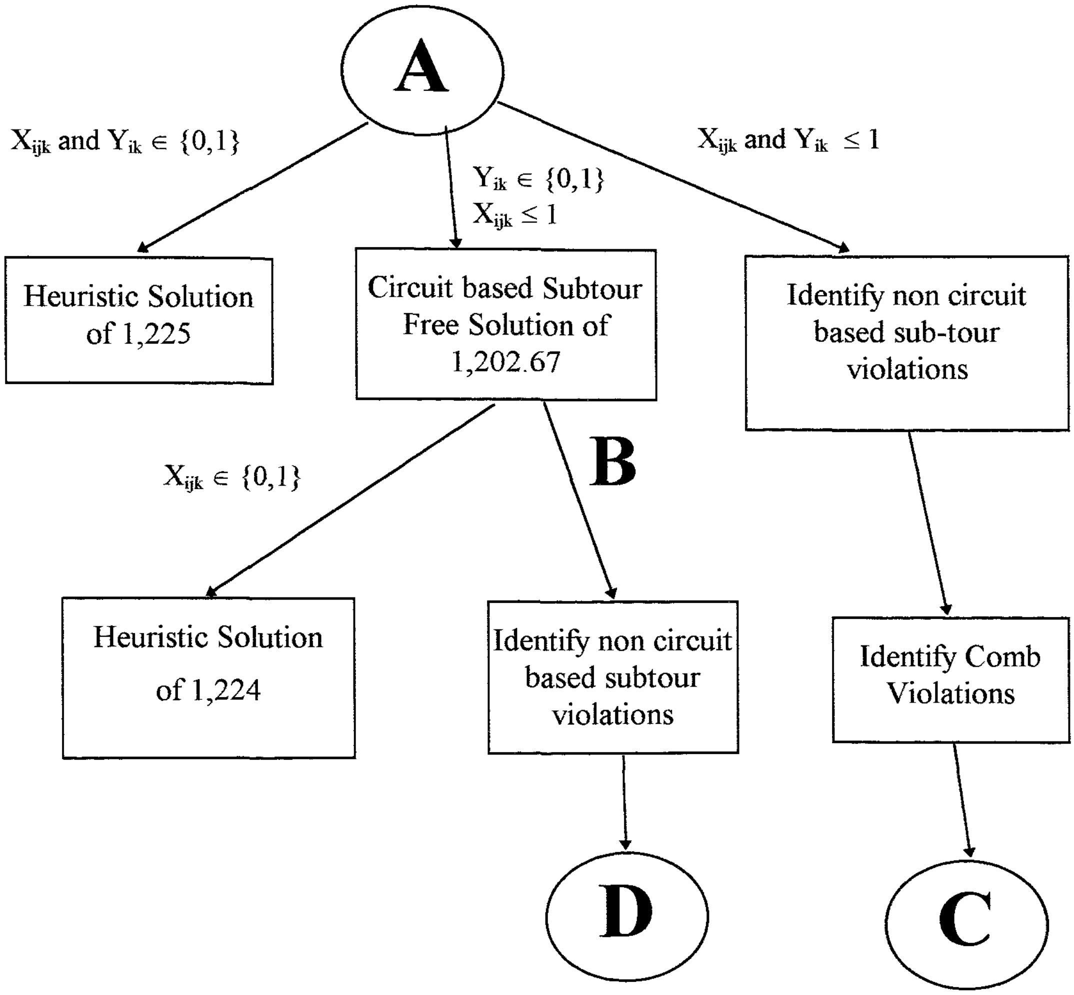


Figure 4.4 : Day_2 tour of heuristic solution of total length 1,224

4.7 Non-Circuit Based Subtours

The diagram of the solution process is now :



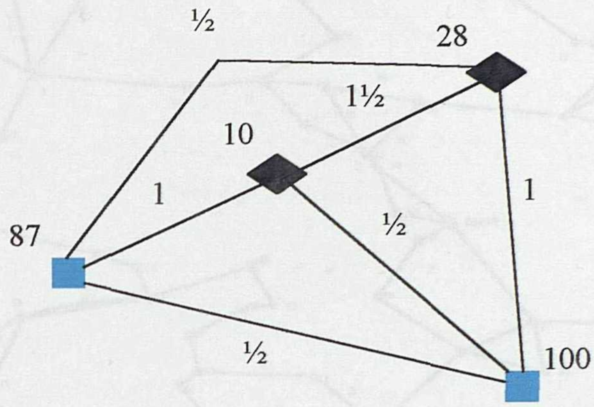
The model at the point **B**, in the previous solution tree, contains :

9,900 X variables, where $0 \leq X \leq 1$,

120 Y variables of the form { 0,1 }, and

562 constraints.

The objective value solution to this model is 1,202.667. This solution contains no circuit based sub-tours. However, having manually drawn the solution, several non circuit based subtours were found. One such example is shown below :



In the above diagram the value shown on each arc is $X_{ij1} + X_{ij2}$ for that arc. Individually, no circuit violates a subtour elimination constraint. However, the total value of the arcs connecting the 4 nodes must be ≤ 4. The above values sum to 5. Thus the constraint

$$X_{10,28,1} + X_{10,28,2} + X_{10,87,1} + X_{10,87,2} + X_{10,100,1} + X_{10,100,2} + X_{28,87,1} + X_{28,87,2} + \leq 4$$

is added to the model.

An iterative process of identifying and then preventing both circuit based and non circuit based sub-tour violations increased the solution value of the objective function to 1,213.5. The aggregate solution for the two days is shown in Figure 4.5. The value shown on each arc is $X_{ij1} + X_{ij2}$ for that arc. It is only when this value does not equal 1, that the appropriate value is shown.

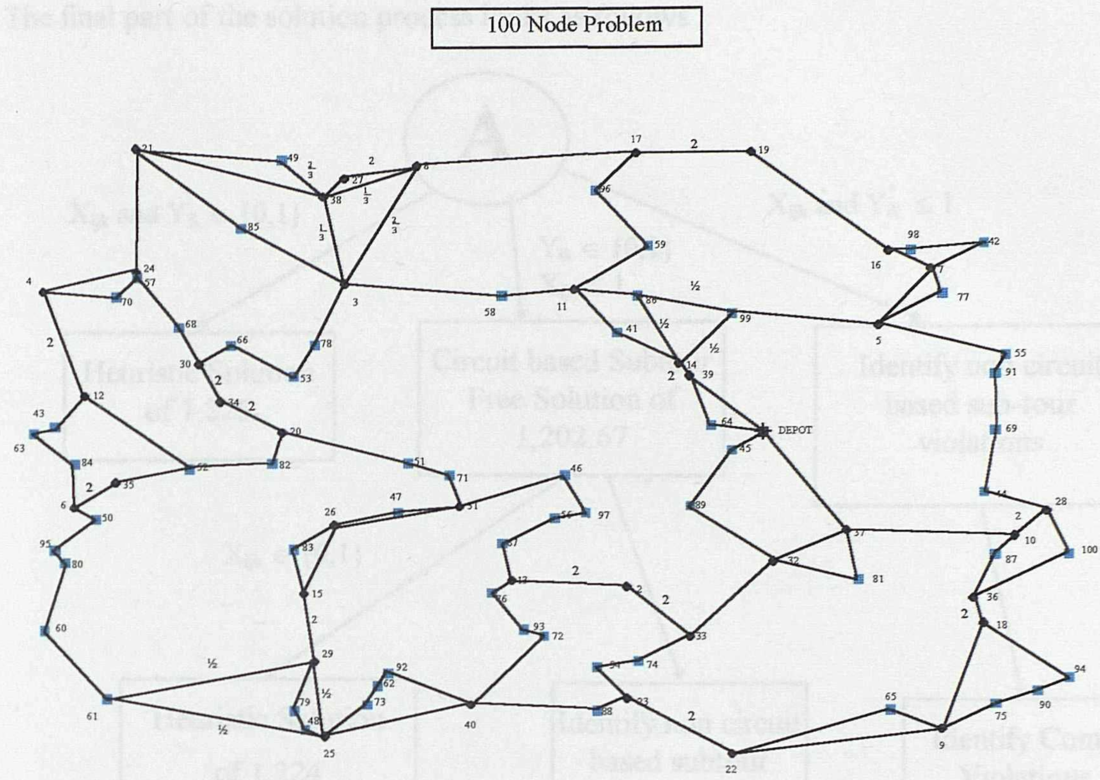


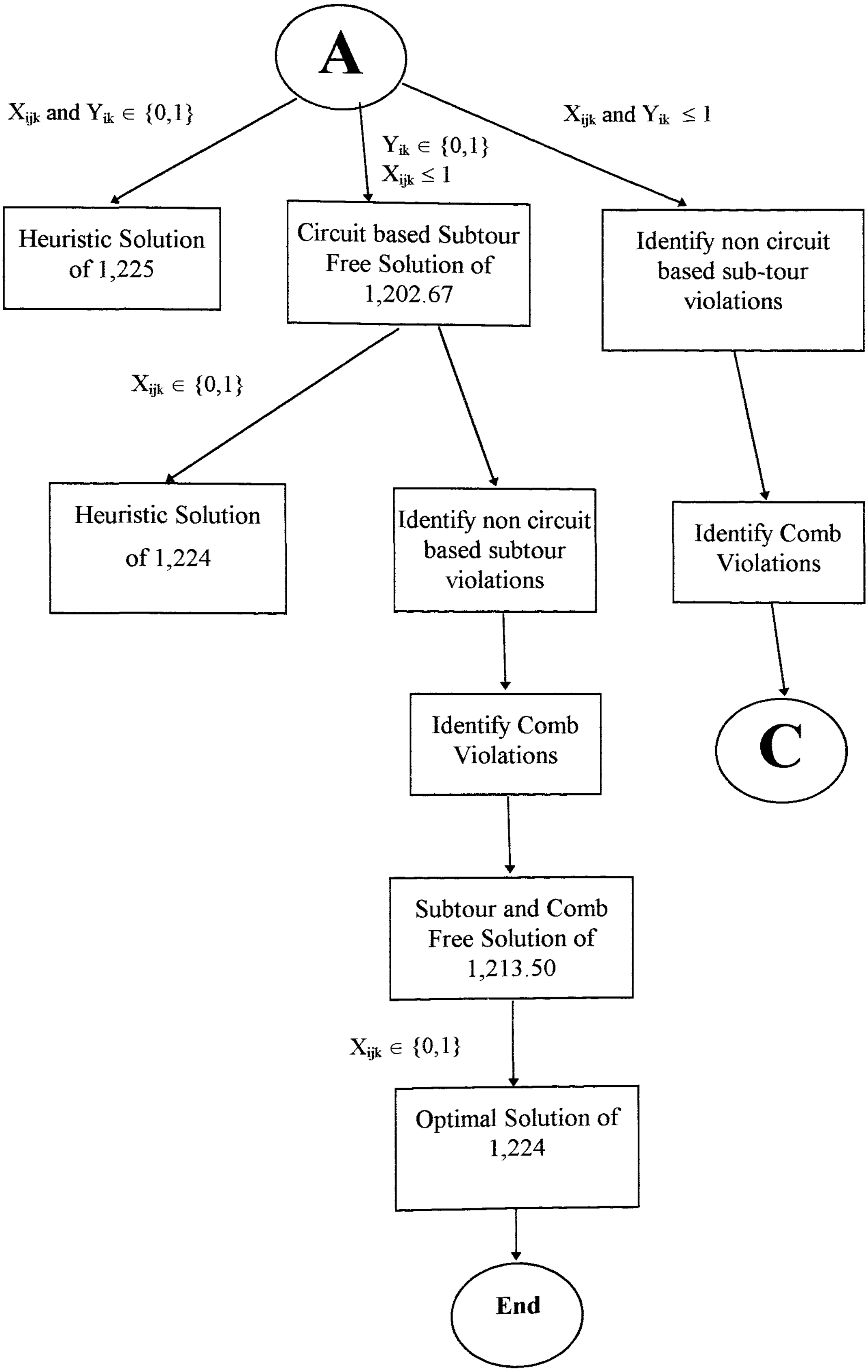
Figure 4.5 : MIP Solution at point D in the solution tree.

A detailed examination of Figure 4.5. suggests that this solution contains neither subtour or comb violations. At this stage a gap of 10.5 exists between the best heuristic objective value solution of 1,224 and the best lower bound solution of 1,213.5. Integer programming using CPLEX is now used to bridge this gap.

4.8 Solution by {0,1} Programming

The ability of CPLEX to solve the IP model at this stage proved difficult, and only after interesting experimentation between the CPLEX parameters that governs the solution strategy between depth-first search and best-bound search did CPLEX finally provide an optimal integer answer. Several of the initial answers contained subtours, but the final, sub-tour free, answer with an objective value solution of 1,224 is obtained. This solution is shown in Figures 4.6 and 4.7. The existence of the optimal solution made it unnecessary to explore further node C.

The final part of the solution process looks as follows :



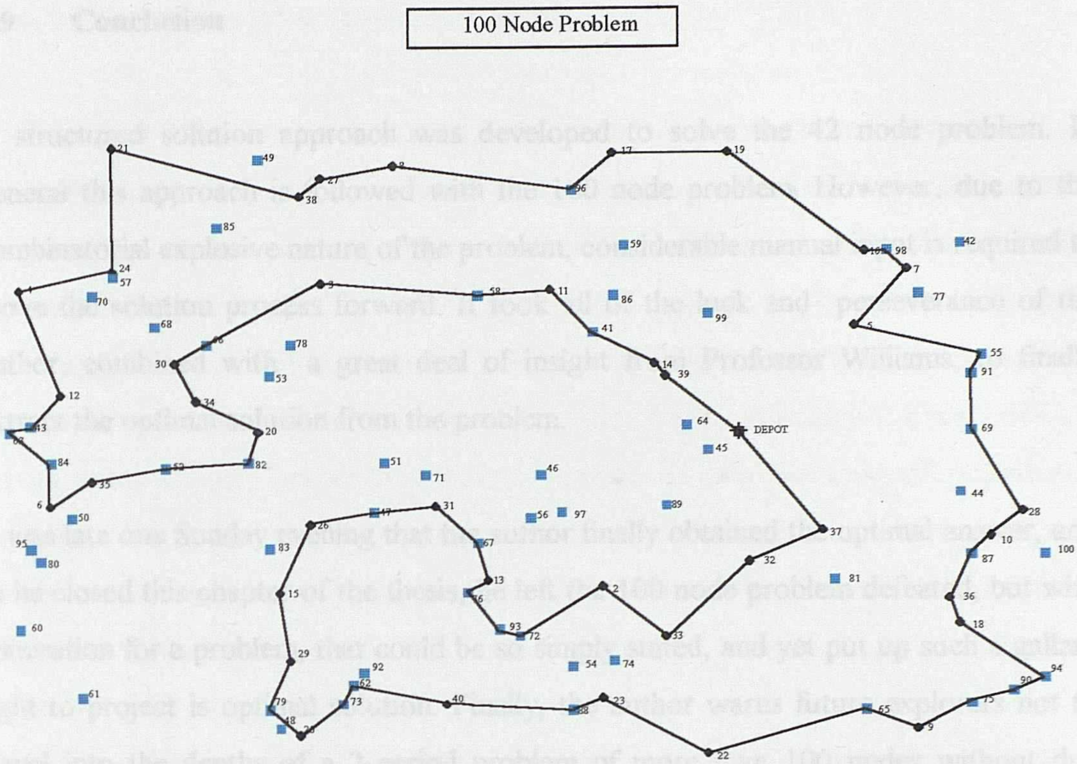


Figure 4.6 : Day_1 tour of optimal solution of total length 1,224

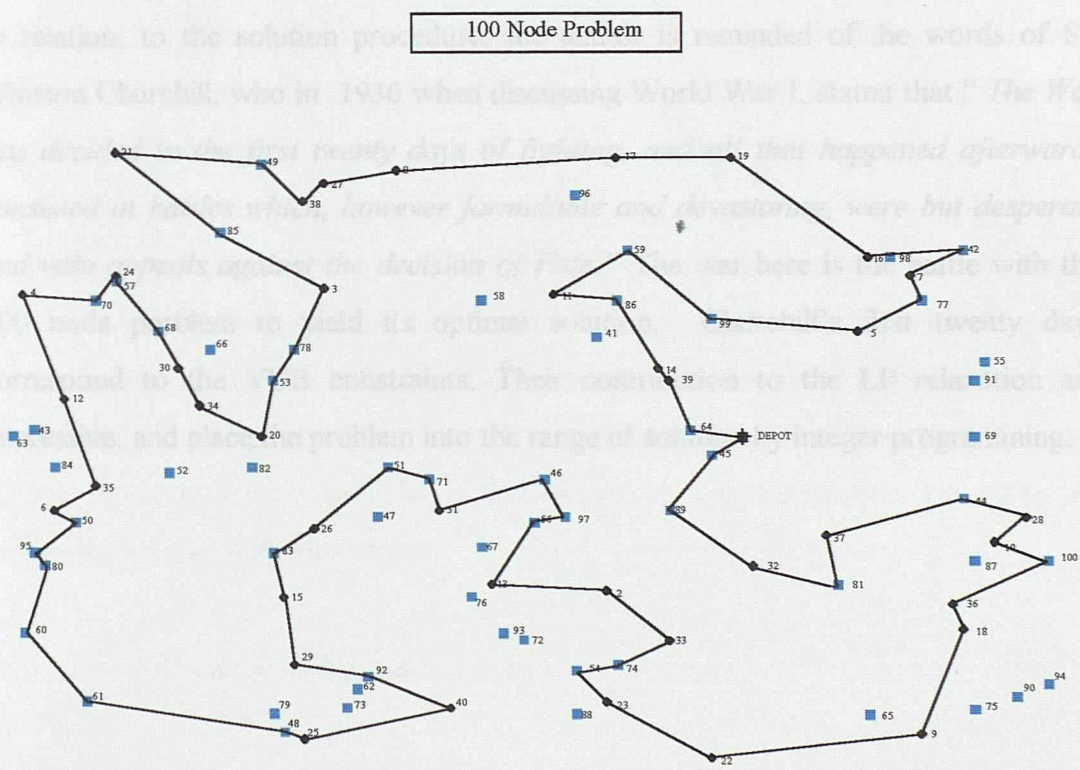


Figure 4.7 : Day_2 tour of optimal solution of total length 1,224

4.9 Conclusion

A structured solution approach was developed to solve the 42 node problem. In general this approach is followed with the 100 node problem. However, due to the combinatorial explosive nature of the problem, considerable manual input is required to move the solution process forward. It took all of the luck and perseverance of the author, combined with a great deal of insight from Professor Williams, to finally extract the optimal solution from the problem.

It was late one Sunday evening that the author finally obtained the optimal answer, and as he closed this chapter of the thesis, he left the 100 node problem defeated, but with admiration for a problem, that could be so simply stated, and yet put up such a gallant fight to project is optimal solution. Finally, the author warns future explorers not to travel into the depths of a 2-period problem of more than 100 nodes without due caution.

In relation, to the solution procedure, the author is reminded of the words of Sir Winston Churchill, who in 1930 when discussing World War I, stated that " *The War was decided in the first twenty days of fighting, and all that happened afterwards consisted in battles which, however formidable and devastating, were but desperate and vain appeals against the decision of Fate.*" The war here is the battle with the 100 node problem to yield its optimal solution. Churchill's first twenty days correspond to the VUB constraints. Their contribution to the LP relaxation are impressive, and place the problem into the range of solution by integer programming.

Chapter 5 : Bounds for the 2-Period TSP.

In Chapters 3 and 4, both the 42 node and 100 node versions of the 2-period TSP are optimally solved using a combination of constrained LP relaxations and integer programming. Another general purpose methodology that is adapted to the TSP is the technique of Branch and Bound. The concept goes back to the work of Dantzig, Fulkerson & Johnson [1954] on the TSP, although the term *branch and bound* was first used - and in the context of the TSP - by Little, Murty, Sweeney & Karel [1963]. As its name implies, branch and bound consists of two fundamental procedures. Branching is the process of partitioning a large problem into two or more sub-problems, and bounding is the process of calculating a lower bound on the optimal solution of a given sub-problem.

The ability of the branch and bound approach to optimally solve a discrete optimisation problem depends on the quality of the bounds produced. This chapter investigates three classes of bounds for the 2-period TSP. The first class is based on increasingly constrained LP relaxations. The second class is based on an extension of the work done by Held & Karp [1970, 1971] on the 1-tree concept. The third is based on Lagrangian relaxation.

In addition to their importance within optimal solution procedures, bounds play a major role in the evaluation of heuristic procedures. A key aspect in the empirical evaluation of heuristics is the comparison of the heuristic result with the optimal tour length. Unfortunately, especially for large problems, the optimal tour length is not known. In these cases it has become the practice to compare heuristic results to the best bound on the optimal solution.

5.1 Bounds based on LP Relaxations

The symmetric 2-period TSP is defined on a complete undirected graph G = (V,E) on n nodes, with node set V and arc set E and costs c_{ij}. V is divided into two sets, V^1 and

V^2. V^1 contains n_1 nodes each of which is to be visited by only one tour. V^2 contains n_2 nodes each of which is to be visited by both tours. The problem is stated as

$$Minimise \sum_{i \in V} \sum_{j>i} \sum_{k=1}^{2} C_{ij} X_{ijk} \qquad (5.1)$$

subject to

$$\sum_{j:j<i} X_{jik} + \sum_{j:j>i} X_{ijk} = 2, \qquad i \in V_2, \ k = 1, 2 \qquad (5.2)$$

$$\sum_{j:j<i} X_{jik} + \sum_{j:j>i} X_{ijk} - 2Y_{ik} = 0, \qquad i \in V_1, \ k = 1, 2 \qquad (5.3)$$

Subtour Elimination Constraints (5.4)

$$Y_{i1} + Y_{i2} = 1, \qquad i \in V_1 \qquad (5.5)$$

$$X_{ijk} = 0 \text{ or } 1 \qquad i, j \in V, j > i, and\, k = 1 \text{ } or \text{ } 2 \qquad (5.6)$$

$$Y_{ik} = 0 \text{ or } 1 \qquad i \in V_1, \ k = 1 \text{ } or \text{ } 2 \qquad (5.7)$$

A detailed explanation of subtour elimination constraints (5.4) is contained in Section 1.4 of Chapter 1.

As part of the solution procedure to the 42 node and 100 node examples of the 2-period TSP, detailed in Chapter 3 and Chapter 4, various LP relaxations are solved. All of these LP relaxations are solved in a few seconds using CPLEX on a Pentium Pro. These solutions provide bounds on the optimal solution.

5.1.1 2-Matching LP Relaxation

The 2-matching LP relaxation is obtained by removing constraints (5.4) and relaxing the {0,1} requirement in (5.6) and (5.7) to $0 \leq X_{ijk}$ and $Y_{ik} \leq 1$. Experience from Chapter 3 and Chapter 4 suggests that this relaxation provides a poor bound. The bound being 90% for the 42 node problem, and only 86% for the 100 node problem of the optimal value.

The solution to the 2-matching LP relaxation violates, what are termed in Chapter 3, the VUB constraints. By explicitly adding VUB constraints on an "as needed basis" provides a solution that satisfies both the 2-matching constraints and the VUB constraints. The quality of the bound provided by this solution is better, being 98% and 95% on the optimal solution for the 42 node and 100 node problems respectively.

5.1.2 *Held-Karp* Bound

One of the best lower bounds for the symmetric TSP, called the *Held-Karp* lower bound, is the solution to the linear programming relaxation of the standard integer programming formulation. Johnson, McGeoch and Rothberg [1996] show that, for randomly generated problems, the optimal tour length averages less than 0.8% over the *Held-Karp* bound. This LP relaxation satisfies all of the 2-matching and subtour elimination constraints. For the TSP the *Held-Karp* bound can be obtained either by solving the LP relaxation and then adding generalised subtour elimination constraints, or by Lagrangian relaxation applied to the 1-tree concept.

The 2-period problem is defined by the mathematical formulation (5.1) to (5.7). An LP relaxation of this formulation is obtained by replacing the integer restriction in (5.6) and (5.7) by the constraints :

$$0 \leq X_{ijk} \leq 1$$

$$0 \leq \mathrm{Y}_{\mathrm{ik}} \leq 1$$

To reflect the pioneering work done by Held and Karp, the bound obtained from the solution of the above relaxation is termed the *Held-Karp* bound for the 2-period TSP.

The *Held-Karp* bound for the 2-period TSP is obtained by first solving the 2-matching LP relaxation. Generalised subtour elimination constraints are next added on an "as needed" basis. The solution obtained is the required bound. This process is performed in Chapter 3 for the 42 node problem and Chapter 4 for the 100 node problem. The results provide a lower bound on the optimal answer of 99.2% and 99% for the 42 an 100 node problems respectively.

Table 5.1 summarises the quality of the bounds obtained from LP relaxations for the 42 node and 100 node problems.

Model	Bound			
	42 Node		100 Node	
2-matching LP relaxation.	1,554	90%	1,049	86%
2-matching LP relaxation plus VUB constraints	1,694.5	98%	1,157	95%
Held-Karp Bound	1,711.8	99.2%	1,211	99%
Optimal	1,725		1,224	

Table 5.1 : Bounds from LP Relaxation

5.2 Bounds from the Shortest Spanning 1-Tree.

A relaxation of the 2-period TSP is obtained by removing the restriction that the solution consists of two tours. In the one tour relaxation, each every day node is duplicated, and each every second day node is included only once. Therefore, the single tour passes through each every day node twice, and once through each every second day node. A value of ∞ is allocated to the distance between the two occurrences of each every day node. This prevents the tour travelling directly between these duplicated nodes.

The figure 5.1 shows the 11 node 2-period TSP with each every day node duplicated. In this problem, nodes 1 to 5 are every day nodes, and are duplicated with nodes 1' to 5' respectively.

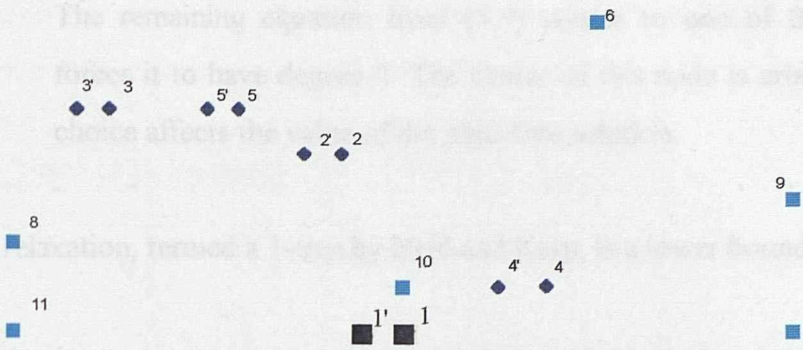


Figure 5.1 : 11 node 2-Period TSP with nodes 1 to 5 duplicated.

The TSP relaxation of the 11 node 2-period TSP contains 16 nodes. In general, the TSP relaxation of the 2-period TSP is defined on a complete undirected graph G = (V,E) on $2n_2 + n_1$ nodes. The node set V contains the nodes from V^1 together with the nodes from V^2 included twice. The symmetric cost matrix, C_{ij}, is derived in an obvious fashion from the original 2-period cost matrix, with the addition that $C_{ii'} = \infty$ for all $i \in V^2$. The problem is stated as

$$Minimise \sum_{i \in V}\sum_{j>i} C_{ij} X_{ij} \qquad i \in V, \qquad (5.8)$$

subject to

$$\sum_{j:j<i} X_{ji} + \sum_{j:j>i} X_{ij} = 2, \qquad i \in V, \qquad (5.9)$$

Subtour Elimination Constraints (5.10)

$$X_{ij} = 0 \text{ or } 1 \qquad i,j \in V, j > i, \qquad (5.11)$$

A relaxation of the TSP can be obtained through the following amendments to the above formulation:

- All the equations in (5.9), except one, are replaced by their sum. This sum forces the total number of arcs in the solution to equal $2n_2 + n_1 - 2$.

- The remaining equation from (5.9) relates to one of the nodes and forces it to have degree 2. The choice of this node is arbitrary, but the choice affects the value of the objective solution.

The above relaxation, termed a 1-tree by Held and Karp, is a lower bound on the TSP.

The 1-tree can be solved by omitting a nominated node, and then solving the MST through the remaining nodes. The omitted node is now connected to the spanning tree with the two shortest arcs.

Let Z_i be the value of the solution to the 1-tree when node $i \in V$ is nominated as the node to be omitted during the solution procedure. Z_i is a lower bound for the TSP, and is also a lower bound on the original 2-period TSP from which the TSP is derived.

The best bound, Z^*, is obtained by omitting each node in turn, and then selecting the largest of the 1-tree solutions, that is

$$Z^* = \underset{i \in V}{Max} \ (Z_i)$$

Figure 5.2 shows the 1-tree solution for the revised 11 node TSP with node 1' as the omitted node. The solution to this 1-tree problem has an objective value solution of 300.

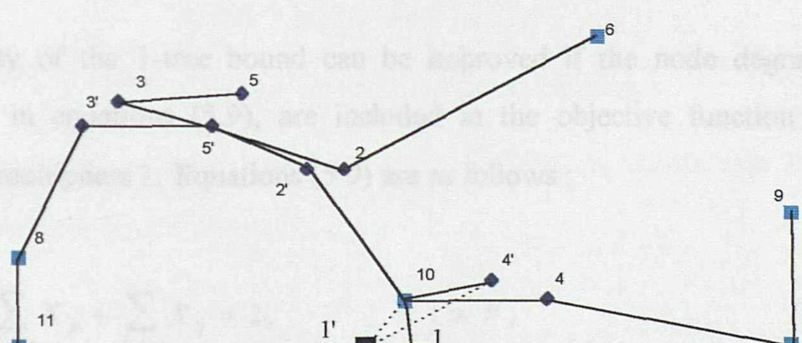


Figure 5.2 : 1-Tree for the revised 11 node TSP with node 1' as the omitted node.

The impact of the choice of node to be omitted during the 1-tree algorithm for the 11 node 2-period TSP is shown in the following Table 5.2.

Node to be Excluded	Value of the 1-tree.
1	300
2	300
3	296
4	300
5	296
6	328
7	322
8	326
9	322
10	300
11	326

Table 5.2 : Length of the 1-tree as a function of the excluded node.

The highest value in the above table, and thus the best bound from the 1-tree relaxation is 328. This is 81% of the optimal solution of 406.

The quality of the bound for the 2-period TSP from the 1-tree relaxation is consistently poor, the bound averages only 80% of the optimal value.

5.3 Lagrangean Relaxation

The quality of the 1-tree bound can be improved if the node degree constraints, expressed in equations (5.9), are included in the objective function by means of Lagrange multipliers λ. Equations (5.9) are as follows :

$$\sum_{j:j<i} X_{ji} + \sum_{j:j>i} X_{ij} = 2, \qquad i \in V, \qquad (5.9)$$

The Lagrangean problem,

$$P(\lambda) = Min\left\{ \sum_{i\in V}\sum_{j>i} C_{ij}X_{ij} + \sum_{i\in V}\lambda_i\left(\sum_{j<i}X_{ji} + \sum_{j>i}X_{ij} = 2\right) \right\}$$

where λ is any n vector,

is a relaxation of the TSP. P(λ) is a lower bound on the 2-period TSP for all λ. The best bound is obtained by finding λ^* such that

$$P(\lambda^*) = \text{Max } P(\lambda)$$

The problem of finding the best value for λ is complex, and involves, what is called, subgradient optimisation. This process starts with an initial estimate for λ and then λ is updated as P(λ) hopefully converges to its optimal value.

Despite its simplicity, the subgradient method gives rise to a number of problems regarding the rate of convergence of P(λ). Held, Wolfe, and Crowder [1974] provides references for the impact of the subgradient parameters on the rate of convergence.

The formula used in this section is :

$$\lambda_i^{k+1} = \lambda_i^k + t^k(d_i^k - 2), \qquad i \in V$$

where t^k is the "step-length", and

d_i^k is the degree of node i at iteration k..

Various studies exist for the best value for t^k. It can be shown that the subgradient optimisation process converges if

$$\sum_{k=1}^{\infty} t^k = \infty \text{ and } \text{Lim}_{k\to\infty} t^k = 0.$$

Series of the form $1 + \frac{1}{2} + \frac{1}{3} + \frac{1}{4} + \ldots\ldots$ satisfy the above conditions.

5.3.1 Best choice for t^k.

The empirical study was performed using the 42 node problem. In each case 100 iterations were performed and the value of P(λ) is recorded.

Case 1 :

$$\lambda_i^{k+1} = \lambda_i^k + t^k(d_i^k - 2), \qquad i \in V$$

$$\lambda_i^0 = 0 \;,\; t^0 = t, \text{ and } t^{k+1} = \alpha * t^k$$

Using various values for t and α, Table 5.3 is produced.

t	α	Bound after 100 iterations
1	0.9	1,581.47
5	0.9	1611.16
1	0.99	1606.21
5	0.99	1578.09

Table 5.3 : Convergence Value as a Function of t and α.

Note : the above series for t does not conform to the conditions in the literature for convergence. However, surprisingly good rates of convergence are obtained.

Case 2 :

$$\lambda_i^{k+1} = \lambda_i^k + t^k(d_i^k - 2), \qquad i \in V$$

$$\lambda_i^0 = 0 \;,\; t^k = \frac{1}{k}$$

Unfortunately, the above series does not converge to the "best" value within 100 iterations. Other series of a similar form were tested, and none of these performed as well as the best of Case 1.

The bound obtained from the 1-tree relaxation depends on the choice of node to be omitted during the procedure. However, as is to be expected from Lagrangean theory, empirical analysis shows that the choice of node to be omitted during the 1-tree procedure has no effect on either the convergence or the final value obtained by the Lagrangean procedure.

Table 5.4 show the bounds obtained from both the "best" 1-tree and Lagrangean relaxation for the various 2-period problems.

Bound from	Problem			
	11 Node		21 Node	
"Best" 1-tree	328	81%	528	80%
Lagrangean Relaxation	359	88%	578	88%
Optimal	406		660	

Bound from	Problem			
	42 Node		100 Node	
"Best" 1-tree	1,404	81%	904	80%
Lagrangean Relaxation	1,611	93%	1,127	92%
Optimal	1,725		1,224	

Table 5.4 : 1-Tree and Lagrangean Relaxation Bounds

5.4 Lagrangean Relaxation II.

The 2-matching relaxation of the 2-period TSP can be stated as :

$$Z = Minimise \sum_{i \in V} \sum_{j>i} \sum_{k=1}^{2} C_{ij} X_{ijk} \tag{5.12}$$

subject to

$$\sum_{j:j<i} X_{jik} + \sum_{j:j>i} X_{ijk} = 2, \qquad i \in V_2,\ k = 1,\ 2 \tag{5.13}$$

$$\sum_{j:j<i} X_{jik} + \sum_{j:j>i} X_{ijk} - 2Y_{ik} = 0, \qquad i \in V_1,\ k = 1,\ 2 \tag{5.14}$$

$$Y_{i1} + Y_{i2} = 1, \qquad i \in V_1 \tag{5.15}$$

$$X_{ijk} = 0 \text{ or } 1 \qquad i, j \in V, j > i, and\, k = 1 \text{ or } 2 \tag{5.16}$$

$$Y_{ik} = 0 \text{ or } 1 \qquad i \in V_1,\ k = 1 \text{ or } 2 \tag{5.17}$$

Z, the optimal solution to the 2-matching, is a lower bound on the optimal solution to the 2-period TSP.

Z, in turn, has a lower bound Z_{LP} which is the solution of the 2-matching problem with the integer constraints, (5.16) and (5.17), replaced by

$$0 \le X_{ijk} \le 1$$

$$0 \le Y_{ik} \le 1$$

Consider the problem, created from the 2-matching by adding the constraints (5.13) and (5.14) into the objective function in a Lagrangean fashion. This problem

$$Z(\mathrm{u,v}) = Minimise \left\{ \sum_{i \in V} \sum_{j>i} \sum_{k=1}^{2} C_{ij} X_{ijk} + \sum_{\mathrm{i} \in \mathrm{V}_2} \mathrm{u_i} \left(\sum_{j<i} X_{jik} + \sum_{j>i} X_{ijk} - 2 \right) \right.$$

$$\left. + \sum_{\mathrm{i} \in \mathrm{V}_1} \mathrm{v_i} \left(\sum_{j<1} X_{jik} + \sum_{j>i} X_{ijk} - 2Y_{ik} \right) \right\}$$

subject to

$Y_{i1} + Y_{i2} = 1,$	$i \in V_1$
$X_{ijk} = 0 \text{ or } 1$	$i,j \in V, j > i, and\, k = 1 \; or \; 2$
$Y_{ik} = 0 \text{ or } 1$	$i \in V_1,\; k = 1 \; or \; 2$

where u, and v are vectors, provides a lower bound on Z.

It is easy to solve this relaxation for given values of u and v. The optimal solution is

X_{ijk} = 0; if the objective function coefficient of $X_{ijk} \geq 0$.

= 1; otherwise.

In addition, either Y_{i1} or Y_{i2} is set to 1 depending on which has the smaller objective function coefficient.

Subgradient optimisation is used to find the values for u and v that maximises Z(u,v). For the 42 node problem, Z(u,v) converges to 1,554. While the 100 node problem converges to 1,049.

These bounds are the same as Z_{LP}. This is to be expected, since Fisher (1981) shows that a sufficient condition for Max (Z(u,v)) = Z_{LP} is that the Lagrangian problem is unaffected by removing the integrality restriction on the variables. Geoffrion (1974) calls this the integrality property.

Since the bound produced by Lagrangean Relaxation II is only as good as the LP relaxation of the 2-matching problem, has this bound any importance? The answer is yes, and mainly from a computational perspective. For large problems, the size of the LP might prove difficult to solve. Whereas the solution of the Lagrangian problem is much simpler.

5.5 Comparison of Bounds.

This chapter derives bounds from both LP relaxations and shortest spanning 1-trees. The bounds, in increasing order of magnitude, for both the 42 node and 100 node are shown in Table 5.5.

Bound from	Problem			
	42 Node		100 Node	
"Best" 1-tree	1,404	81%	522	80%
Lagrangian Relaxation II	1,554	90%	1,049	86%
2-Matching LP Relaxation	1,554	90%	1,049	86%
1-tree + Lagrangean Relaxation	1,611	93%	1,127	92%
2-Matching Relaxation plus VUB constraints.	1,694.5	98%	1,157	95%
Held-Karp Bound	1,711.8	99.2%	1,211	99%
Optimal	1,725		1,224	

Table 5.5 : Bounds for the 2-Period TSP.

Aside from the *Held*-Karp bound, the quality of the other bounds are sufficiently poor to suggest that branch and bound is not a viable option for solving very large 2-period TSPs. Further research is required to create bounds that will make branch and bound at least as competitive with LP based models for solving medium sized 2-period TSPs.

Chapter 6 Heuristic Procedures

Chapter 3 and Chapter 4 investigate procedures for finding the optimal solution to examples of the 2-period TSP. Experience from these chapters suggest that for problems with over 100 nodes the search time for an optimal solution becomes excessive. This chapter focuses on heuristic solutions, and the design of procedures that while not guaranteed to find optimal tours, do find what one hopes are 'good' solutions.

6.1 Heuristic Procedures

Three classes of heuristics are introduced in this section. These are

- a tour construction procedure,
- a cluster first route second procedure, and finally
- a tour improvement procedure.

Tour construction procedures start with an initial tour, and then using selection rules and insertion rules add new nodes to the solution.

Cluster first route second procedures use a clustering rule to group nodes and then tour generating procedures are used to sequence the nodes in a cluster in a route.

Tour improvement procedures start with a feasible solution and then seek to improve on the answer via a sequence of interchanges. The starting Node for the tour improvement procedure can be either a random sequence of nodes or else a solution generated by another heuristic.

The examination of heuristics is concerned with predicting the quality of the solution to be provided by the heuristic. Johnson and Papadimitriou [1985] identify three approaches for the comparison of heuristics. These are : worst case analysis, probabilistic analysis, and empirical testing. Each of these approaches has its advantages and its drawbacks. The results reported in this chapter are concerned with the performance of the three classes of heuristics against the 11 node, 21 node, 42 node, 100 node and 200 node problems. Optimal solutions are available for the first four of these test problems. For the 200 node problem a lower bound is used for comparison with the heuristic solution. The data for the 200 node problem is contained in Appendix 10.

The results reported in this chapter are not exhaustive. A more detailed empirical analysis of the heuristics is contained in Chapter 7

6.2 Cheapest Insertion Heuristic

This heuristic is from the class of tour construction procedures. In creating tour construction procedures, decisions must be made as to

- the choice of the initial subtour.
- the selection criteria that will be used to identify who next enters the tour.
- the insertion criterion that will dictate where the new entrant is placed in the tour.

With some algorithms, such as the cheapest insertion algorithm, the decisions as to what node is to be inserted and where are made at the same time, other algorithms can be created which employ different criteria. Variants such as nearest addition, nearest insertion, farthest insertion, and greatest angle of insertion can be identified. For the purpose of analysis in this chapter a variant of the cheapest insertion is used as representative of the tour construction class.

The cheapest insertion heuristic for the 2-period TSP first generates a 2-opt tour through all of the every day nodes, including the depot. This tour, T, becomes the starting tour for the day 1 tour, T_1, and the day 2 tour, T_2. For the purpose of generating T, the 2-opt procedure uses the every day nodes and depot in a random order as the starting tour.

Having created the initial tours, nodes are selected, from the set of un-allocated every other day nodes, for inclusion into one of the above tours. The selection criterion and insertion criterion are combined and the selected node is chosen as the node whose cost of insertion into one of the two tours is the cheapest. The process ends when all of the every other day nodes have been allocated to a tour.

The steps in the cheapest heuristic are as follows :

Step 1 : Generate a 2-opt tour, T, through all of the every day nodes, including the depot using a random sequence of nodes as the initial tour.

Step 2 : Initialise

$T_1 = T$,

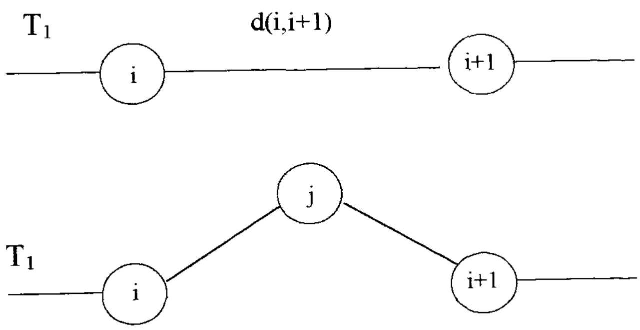
$T_2 = T$, and

$V^1 = \{$ Set of Every other day nodes $\}$.

Step 3 : For all nodes $j \in V^1$ calculate

d_{1j} = cheapest way to insert node j into T_1

$= \text{Min} \{ d(i,j) + d(j,i+1) - d(i,i+1)\}$ for all $i \in T_1$



d_{2j} = cheapest way to insert node j into T_2

$= \text{Min} \{ d(i,j) + d(j,i+1) - d(i,i+1)\}$ for all $i \in T_2$

Step 4 : Find the node $k \in V^1$ such that

$d_k = \text{Min} \{ d_{1k} , d_{2k} \}$

if $d_{1k} = \text{Min} \{ d_{1k} , d_{2k} \}$ then

insert node k into T_1 after node i^*

else

insert node k into T_2 after node i^*

delete node k from V^1

Step 5: If $V^1 = \{\emptyset\}$ then end

else goto Step 3

6.2.1 Application of the Cheapest Insertion Heuristic to the 21 node problem.

Step 1 : Generate a 2-opt tour, T, through all of the every day nodes, including the depot. This tour is shown in Figure 6.1

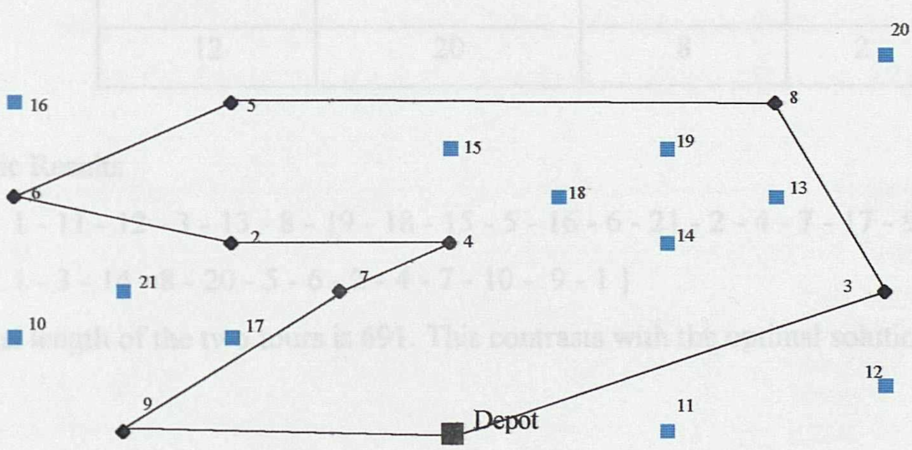


Figure 6.1 : 2-opt Tour Through All of the Every Day Nodes

Step 2 : $T_1 = \{ 1 - 3 - 8 - 5 - 6 - 2 - 4 - 7 - 9 - 1 \}$

$T_2 = \{ 1 - 3 - 8 - 5 - 6 - 2 - 4 - 7 - 9 - 1 \}$

$V^1 = \{ 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20 \}$

Step 3, 4 & 5 Repeated selection from V^1 of the cheapest node to insert. The results of these steps are included in the following table.

Iteration	**Node to Insert**	**After Node**	**Into Day 1 or 2**
1	17	7	1
2	13	3	1
3	15	8	1
4	19	8	1
5	11	1	1
6	12	11	1
7	18	19	1
8	16	5	1
9	14	3	2
10	21	6	1
11	10	7	2
12	20	8	2

Heuristic Results :

T_1 = { 1 - 11 - 12 - 3 - 13 - 8 - 19 - 18 - 15 - 5 - 16 - 6 - 21 - 2 - 4 - 7 - 17 - 9 - 1 }

T_2 = { 1 - 3 - 14 - 8 - 20 - 5 - 6 - 2 - 4 - 7 - 10 - 9 - 1 }

The total length of the two tours is 691. This contrasts with the optimal solution of 660.

6.3 Inside/Outside Heuristic

The second class of heuristics discussed in this paper is a cluster first, route second procedure. The concept of cluster first, route second was first introduced by Gillet and Miller [1974] with their Sweep heuristic for the vehicle routing problem. The 2-period TSP can be reduced to two TSP's if a decision is made as to how the every second day nodes are to be allocated over the two days. Thus, once a decision is made as to how the every second day nodes are allocated to the two days, one is then left with two tour creation problems. Adequate solutions can be found to these two problems by using any of the TSP tour generating heuristics.

Various criterion can be suggested that could cluster the every second day nodes into two groups. The clustering criterion used in this section is based on whether a node falls inside or outside a 2-opt tour through all of the every day nodes, including the depot.

The rationale behind this heuristic is that on both tours all of the every day nodes must be visited, and that the tour through these nodes underlies the final solution for both days. On one day the tour will veer inwards and pick up all of the every other day nodes inside the tour. On the other day the tour will veer outwards and pick up all of the every other day nodes outside the tour.

Figure 6.2 shows how the nodes inside the 2-opt tour through all of the every day nodes are allocated to Day 1, and the nodes outside the tour are allocated to Day 2.

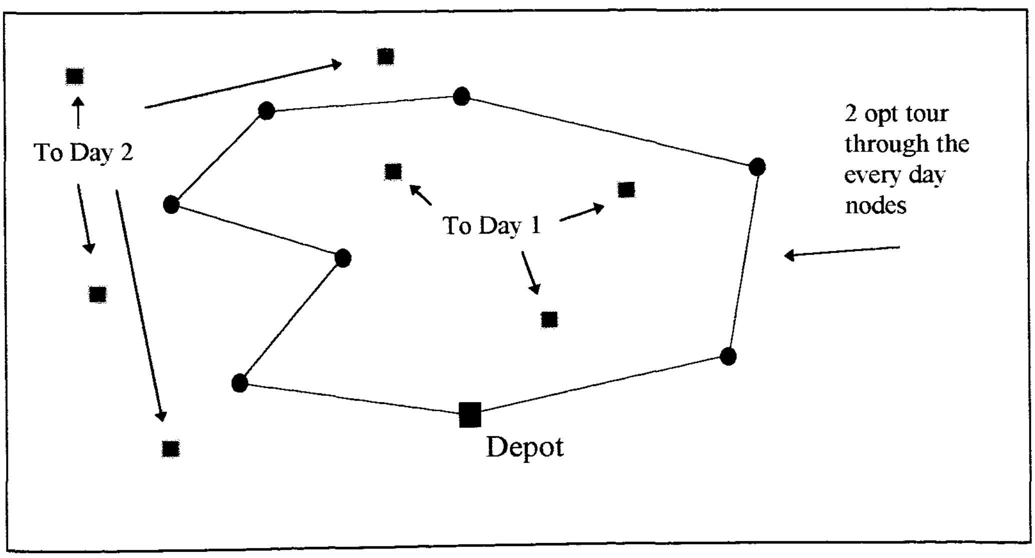


Figure 6.2 : Inside/Outside Partitioning of Every Second Day Nodes

The steps in the heuristic are as follows :

Step 1 : Generate a 2-opt tour, T, through all of the every day nodes, including the depot. The initial tour for the 2-opt procedure is a random ordering of the nodes.

Step 2 : Initialise

V^2 = { Nodes Requiring Every Day Collection }

V^1 = { Nodes Requiring Every Other Collection }

Step 3 Cluster V^1 into $V^1{}_1$ and $V^1{}_2$, where:

$V^1{}_1$ = { Set of Every Other Day nodes allocated to Day 1 }

$V^1{}_2$ = { Set of Every Other Day nodes allocated to Day 2 }

on the basis of whether a Node is inside or outside tour T.

Step 4 : Allocate

$V^2 + V^1{}_1$ = { Set of nodes to be collected on Day 1 }

$V^2 + V^1{}_2$ = { Set of nodes to be collected on Day 2 }

Step 5: Use the 2-opt procedure to find tours, T_1 and T_2 , through the sets $V^2 + V^1{}_1$ and $V^2 + V^1{}_2$. Random sequence of nodes are used as the starting tours for the 2-opt procedure.

6.3.1 Application of the Inside/Outside Heuristic to the 21 node problem.

Step 1 : Generate a 2-opt tour, T, through all of the every day nodes, including the depot. This tour is shown in Figure 6.3.

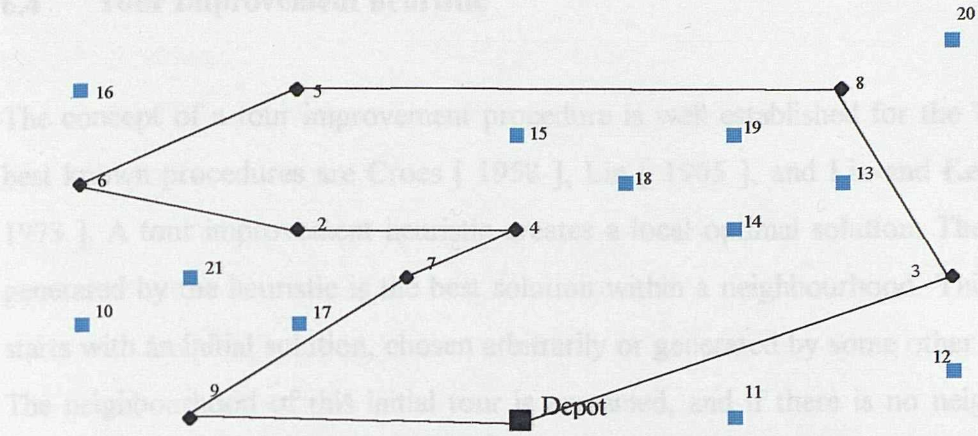


Figure 6.3 : 2-opt Tour Through All of the Every Day Nodes

Step 2 : $V^1 = \{ 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20 \}$

Step 3 : Cluster V^1 into

Inside, $V^1_1 = \{ 13, 14, 15, 18, 19 \}$

Outside, $V^1_2 = \{ 10, 11, 12, 16, 17, 20, 21 \}$

Step 4: Allocate

{ 1, 2, 3, 4, 5, 6, 7, 8, 9, 13, 14, 15, 18, 19 } to Day 1

{ 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 16, 17, 20, 21 } to Day 2

Step 5: Use the 2-opt procedure with random sequences as the initial tours to find tours through the Day_1 and Day_2 allocations.

The result of the Inside/Outside heuristic is :

T_1 = { 1 - 9 - 7 - 2 - 6 - 5 - 15 - 19 - 8 - 13 - 3 - 14 - 18 - 4 - 1 }

T_2 = { 1 - 11 - 12 - 3 - 20 - 8 - 5 - 16 - 6 - 2 - 21 - 10 - 9 - 17 - 7 - 4 - 1 }

The total length is 696. This contrasts with the optimal solution of 660.

6.4 Tour Improvement heuristic

The concept of a tour improvement procedure is well established for the TSP. The best known procedures are Croes [1958], Lin [1965], and Lin and Kernighan [1973]. A tour improvement heuristic creates a local optimal solution. The solution generated by the heuristic is the best solution within a neighbourhood. This process starts with an initial solution, chosen arbitrarily or generated by some other heuristic. The neighbourhood of this initial tour is examined, and if there is no neighbouring solution which is shorter that the initial solution, then this solution is at least a 'local optimum'. If a shorter solution is found, then this new solution becomes the basis of a new neighbourhood search, and the process is repeated until no better solution can be found within a neighbourhood. The heuristic by Lin and Kernighan [1973] defines the 'neighbours' of a tour to be those tours which can be generated from it by a limited number of interchanges of tour edges.

The tour improvement heuristic detailed in this section is a variant of the 2-opt procedure for the TSP. The procedure starts with an initial tour, T_1, for Day 1 and an initial tour, T_2, for Day 2. These tours can either be a random sequence and random allocation of the every second day nodes or else the tours can be a heuristic solution. The two initial tours T_1 and T_2 are combined into a composite tour as follows :

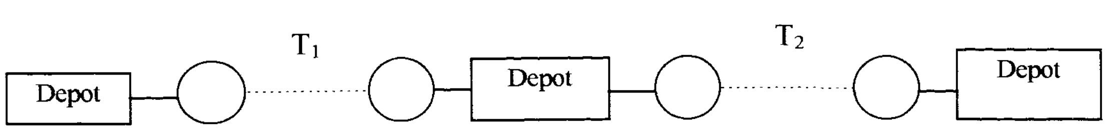


Figure 6.4 : Composite Tour for the Tour Improvement Heuristic

The neighbourhood of this initial tour is created by considering all feasible combinations of pairs of nodes. Arcs connecting these nodes in the current tour are deleted and a possible reconnection is considered. The arcs to be deleted and reconnected for each feasible combination of i and j are as follows :

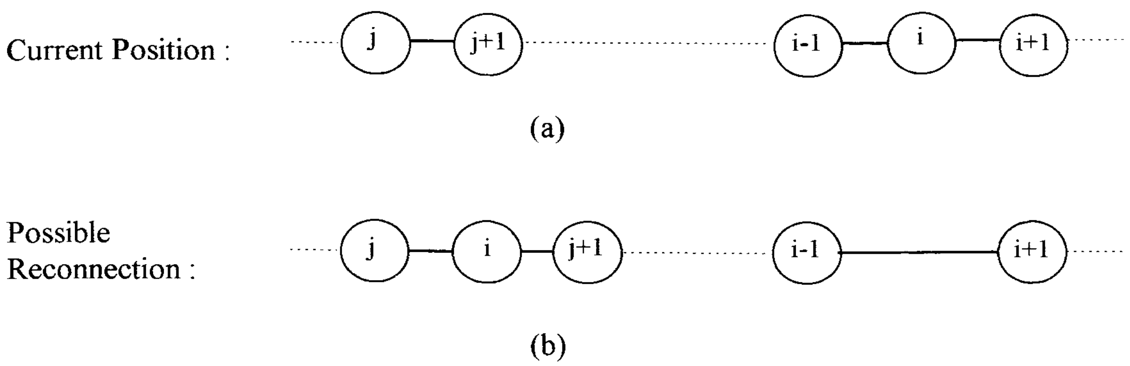


Figure 6.5 : Tour Improvement Example. (a) Current (b) Reconnected

If any of the feasible reconnections yield a reduction in the length of the composite tour then the deletions and reconnections are made and the neighbourhood search restarts. This neighbourhood search continues until there is no feasible deletion and reconnection that improves the current solution.

To test the performance of the Tour Improvement procedure, the heuristic was applied three times to each of the 5 test problems. Firstly, a random sequence of nodes together with a random allocation of the every second day nodes over the two days was used as the initial tour. In the second and third applications the initial tour was provided by both the cheapest insertion and inside/outside heuristics.

The performances of the tour improvement heuristic are detailed later in this chapter.

6.5 Heuristic Results

This section shows the results from applying the heuristic procedures to the 11 node, 21 node and 42 node problems.

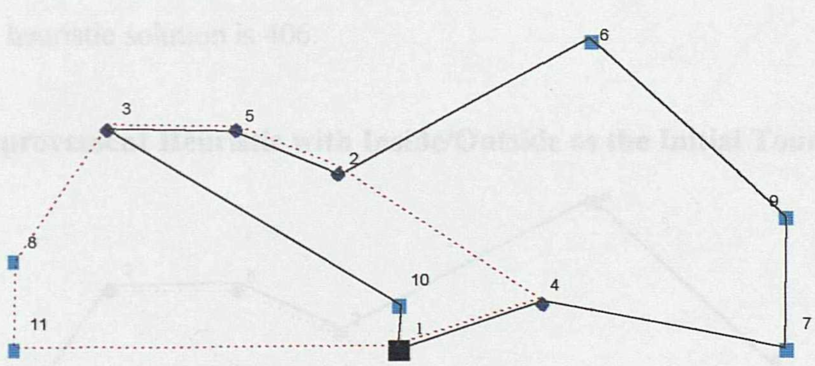
The heuristics examined in this section are :

- Cheapest Insertion.
- Inside/Outside.
- Tour Improvement with Cheapest Insertion as the initial tour.
- Tour Improvement with Inside/Outside as the initial tour.
- Tour Improvement with random initial tour.

6.6 Heuristic Results for the 11 Node Problem

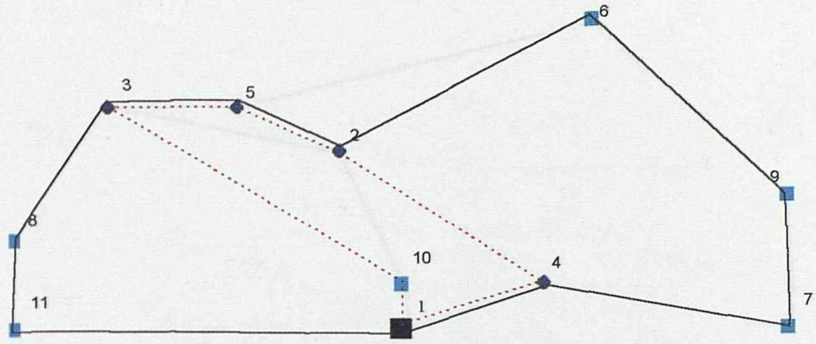
The results achieved when each of the heuristics is applied to the 11 node problem are detailed below. The results are given in Appendix 11.

6.6.1 Cheapest Insertion Heuristic



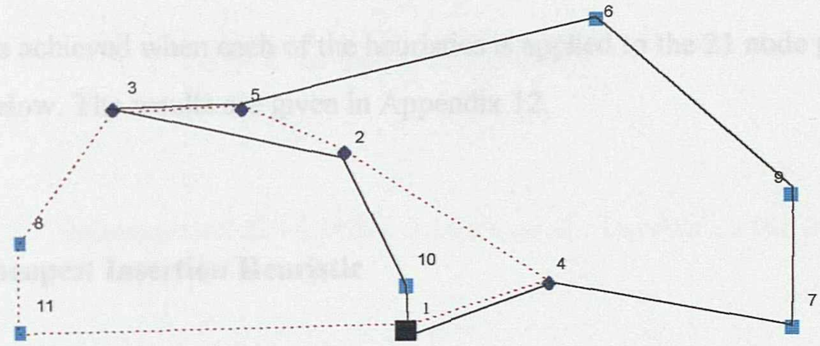
Total Length of Cheapest Insertion Heuristic solution is 413.

6.6.2 Inside/Outside Heuristic



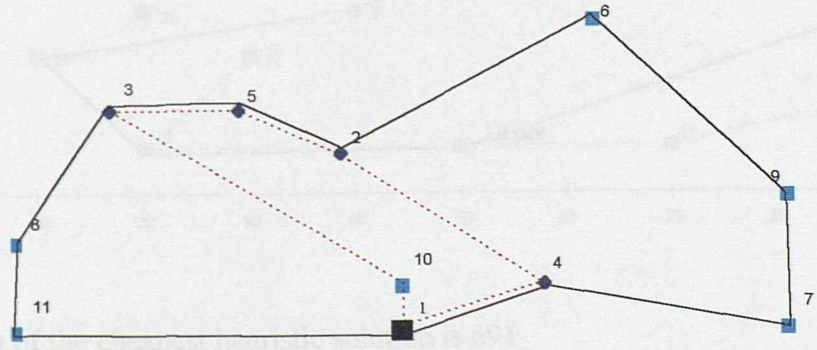
Total Length of Inside/Outside Heuristic solution is 413.

6.6.3 Tour Improvement Heuristic with Cheapest Insertion as the Initial Tour



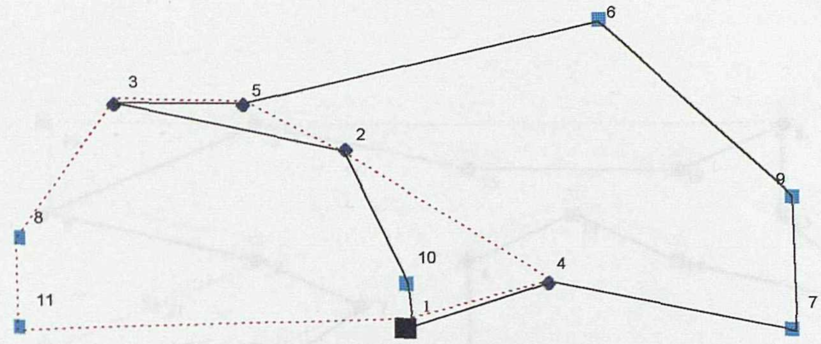
Length of heuristic solution is 406.

6.6.4 Improvement Heuristic with Inside/Outside as the Initial Tour



Length of heuristic solution is 413.

6.6.5 Improvement Heuristic with Random Initial Tour

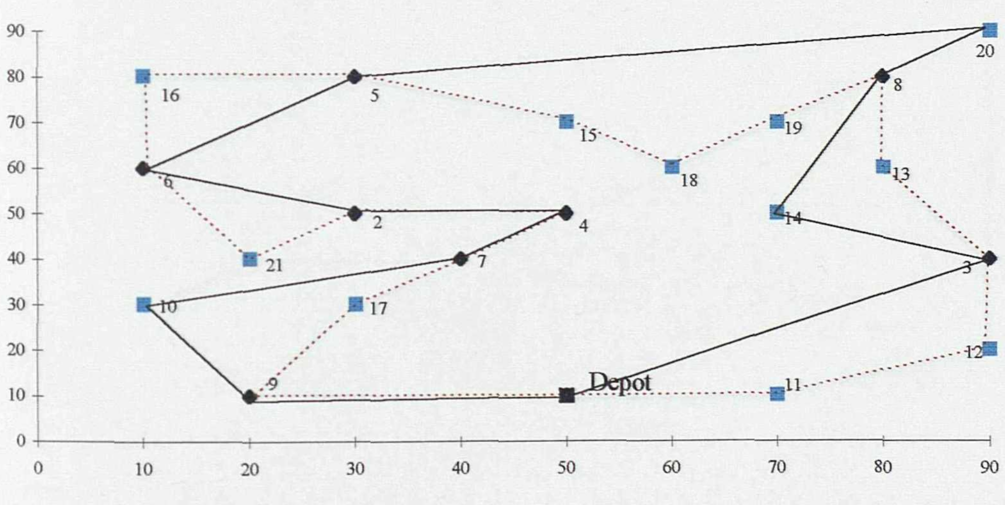


Length of Tour Improvement Heuristic solution is 406.

6.7 Heuristic Results for the 21 Node Problem

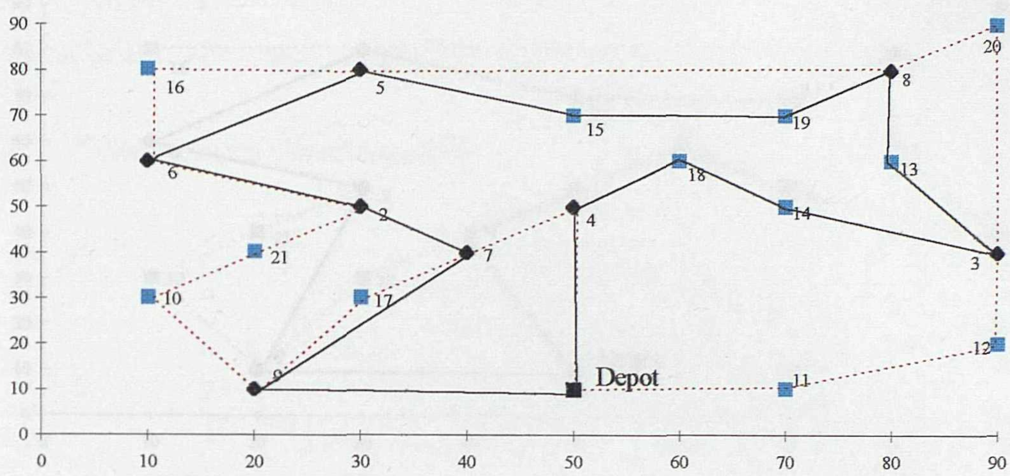
The results achieved when each of the heuristics is applied to the 21 node problem are detailed below. The results are given in Appendix 12.

6.7.1 Cheapest Insertion Heuristic



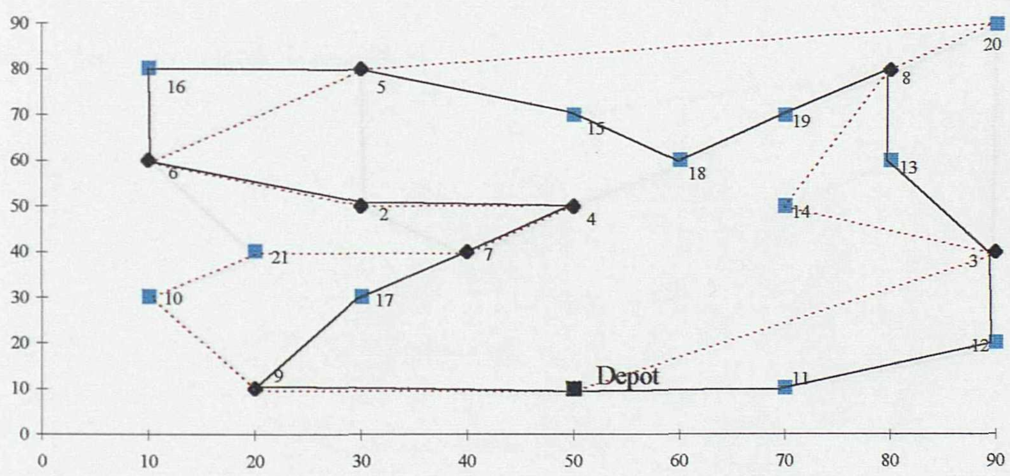
The length of the cheapest heuristic solution is 691

6.7.2 Inside/Outside Heuristic



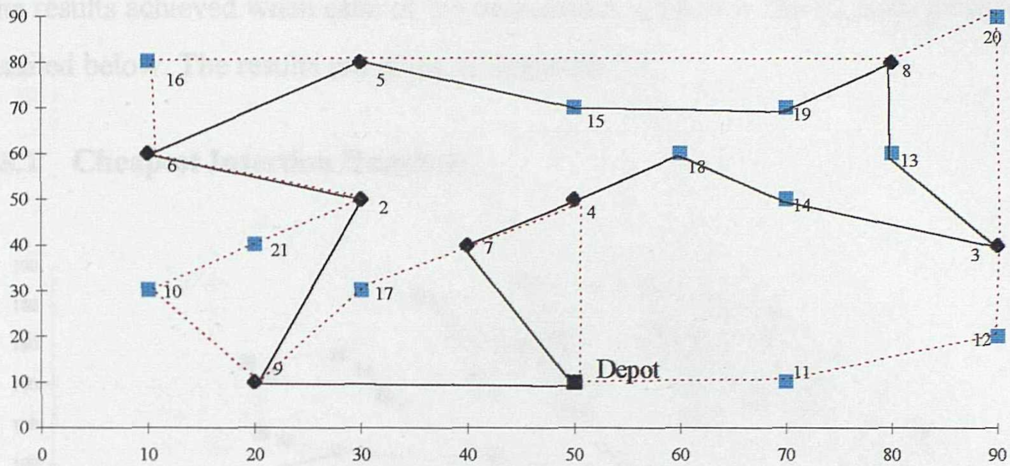
The length of the inside/outside heuristic solution is 696.

6.7.3 Tour Improvement Heuristic with Cheapest Insertion as the Initial Tour



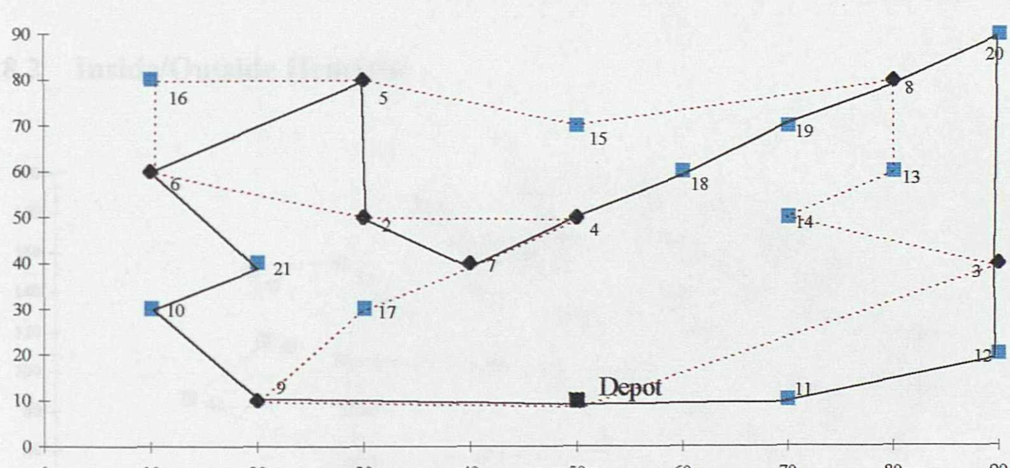
The length of the heuristic solution is 679.

6.7.4 Tour Improvement Heuristic with Inside/Outside as the Initial Tour



The length of the heuristic solution is 693.

6.7.5 Tour Improvement Heuristic with Random Initial Tour

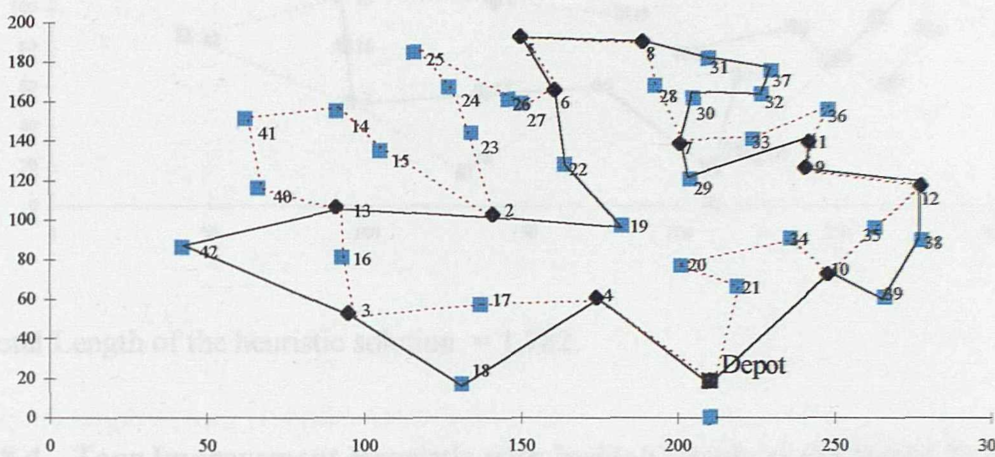


The length of the heuristic solution is 664.

6.8 Heuristic Results for the 42 Node Problem

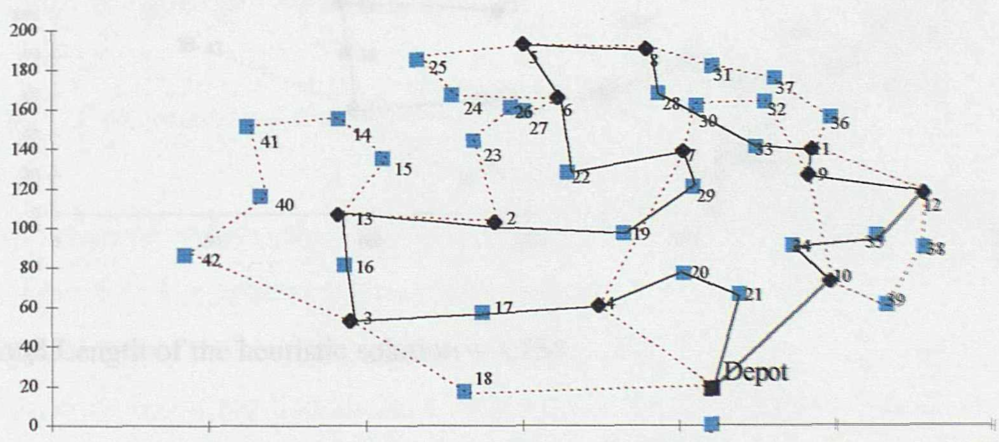
The results achieved when each of the heuristics is applied to the 42 node problem are detailed below. The results are given in Appendix 13.

6.8.1 Cheapest Insertion Heuristic



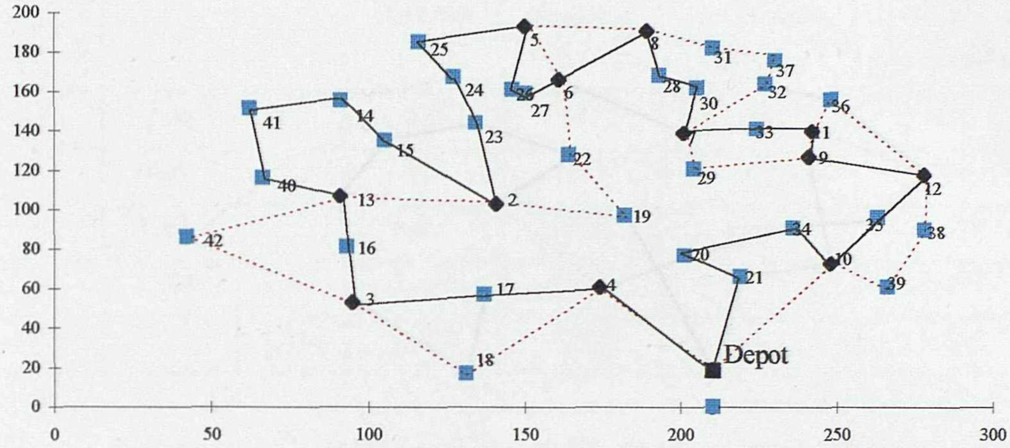
Total Length of Cheapest Insertion Heuristic = 1,791.

6.8.2 Inside/Outside Heuristic



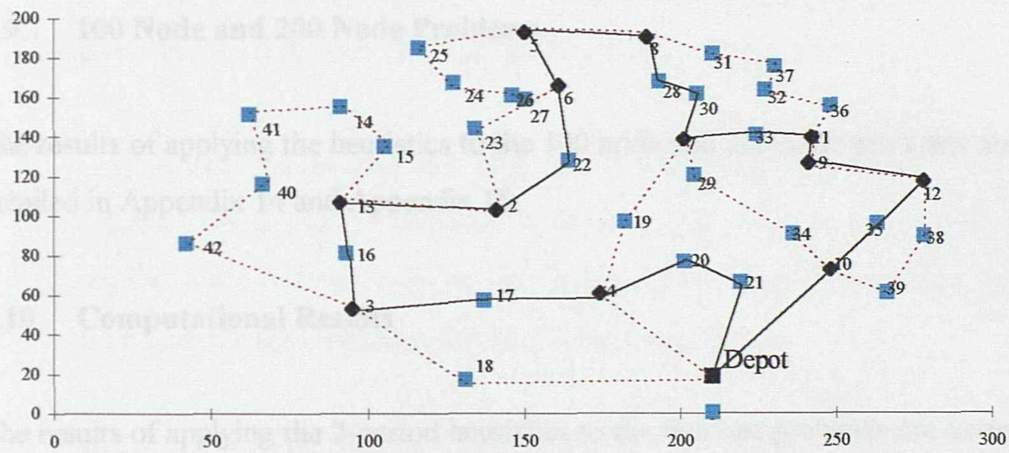
Total Length of Inside/Outside Heuristic = 1,891.

6.8.3 Tour Improvement Heuristic with Cheapest Insertion as the Initial Tour



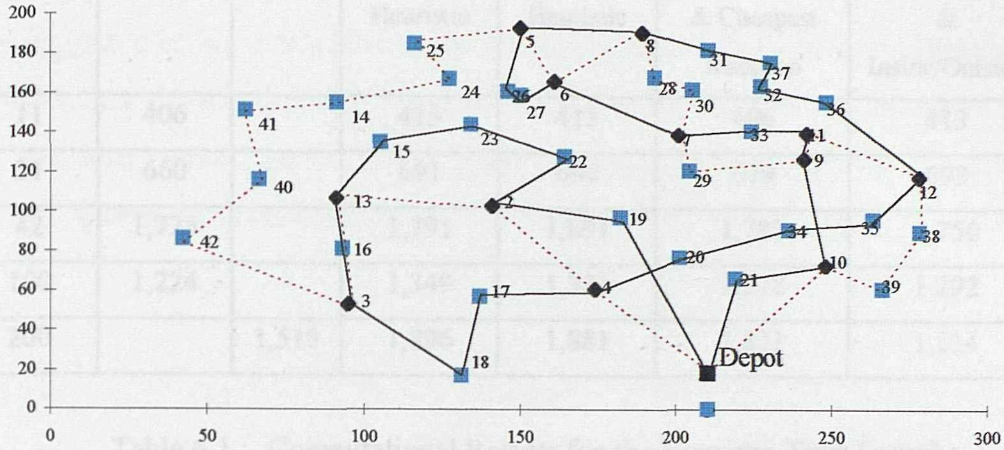
Total Length of the heuristic solution = 1,782.

6.8.4 Tour Improvement Heuristic with Inside/Outside as the Initial Tour



Total Length of the heuristic solution = 1,750.

6.8.5 Tour Improvement Heuristic with Random Initial Tour



Total Length of heuristic solution = 1,859.

6.9 100 Node and 200 Node Problems

The results of applying the heuristics to the 100 node and 200 node problems are detailed in Appendix 14 and Appendix 15.

6.10 Computational Results

The results of applying the 2-period heuristics to the five test problems are summarised in Table 6.1. The optimal solution is available for the test problems of size 11, 21, 42, and 100 nodes. In the absence of an optimal solution for the test problems of size 200, the current best lower bounds, derived from an LP relaxation solution, is shown.

Number of Nodes	Optimal Solution	Lower Bound	Cheapest Insertion Heuristic	Inside/ Outside Heuristic	Tour Improvement & Cheapest Insertion	Tour Improvement & Inside/Outside	Tour Improvement & Random Start
11	406		413	413	406	413	406
21	660		691	696	679	693	664
42	1,725		1,791	1,891	1,782	1,750	1,859
100	1,224		1,349	1,331	1,278	1,292	1,458
200		1,518	1,896	1,881	1,821	1,814	2,298

Table 6.1 : Computational Results for the Heuristic Tour Lengths

Analysis of the Table 6.1 suggests that :

1. The two-step composite procedure of the tour improvement process applied to an original heuristic solution gives the best answer to each of the test problems, with the exception of the 21 node problem.

2. On the basis of the five test problems, there is little to choose between the solutions offered by the cheapest insertion and the inside/outside heuristics. For the five test problems each heuristic is better for two of the problems , and they both give the same solution for the 11 node problem.

3. For the test problems with known optimal solutions, the best heuristic answer is inside 4.5% of optimality .

4. The poor solutions to the larger test problems generated by the tour improvement heuristic over a random start suggests that the use of random initial tours will not lead to good solutions with this version of the tour improvement procedure. This contrasts with the high quality of solution given

by the tour improvement procedure when applied to initial tours provided by another heuristic.

As suggested earlier in this chapter, no attempt is made here to perform a detailed analysis of 2-period heuristics. The objective behind this chapter is to introduce a range of heuristics for the 2-period TSP. Chapter 7 will perform a more comprehensive empirical analysis of the heuristics.

Chapter 7 Empirical Study of 2-Period Heuristics

7.1 Introduction

Chapter 6 introduces three classes of heuristics for the 2-period TSP. These are :

a tour construction class,

a cluster first route second class, and

a tour improvement class.

In this chapter, examples of the above classes of are analysed under the methodology proposed by Ball and Magazine [1981]. In their paper several criteria are suggested for the comparison of heuristics. These include :

Quality of Solution,

Running Time,

Ease of Implementation,

Flexibility, and

Simplicity.

Of the above criteria, Running Time and Quality of Solution can be objectively measured, the remaining criteria, Flexibility, Ease of Implementation, and Simplicity are more subjective. Later in this chapter scores are allocated, on the scale of 1 to 10, for each heuristic for these subjective criteria.

7.2 Running Time and Quality of Solution

To test the performance of the heuristics under the headings of Running Time and Quality of Solution a set of test problems were randomly generated. The test problems contained 10 examples of problems of size 50, 100, 150, 200, 250, 500, and 1,000 nodes. For each test problem the number of every day nodes is randomly generated in the range 25% to 75% of the problem size. For each test problem a set of X,Y co-ordinates was randomly generated with both X and Y in the range 0 to 100. The d(i,j) matrix for each problem is calculated using the rounded integer Euclidean distance between the points. Full details of the Test Problems can be obtained from the author.

Implementations of the various heuristics are applied to each test problem. The average solution lengths and the average running times for each heuristic over the 10 test problems for each of the 7 sizes of problems are calculated. The results are reported later in the chapter. The implementation by the author of the various heuristics might not be optimal. However, the comparative nature of the analysis minimises this negative aspect.

7.3 Tour Construction Heuristics

In creating tour construction heuristics, three decisions must be made. These are :

1. The choice of the initial subtours.

   With the 2-Period TSP, the final solution is two tours, thus, two initial tours will be initialised which will then grow into the final solution.

   For the heuristics analysed in this section, the initial tours consist of a simple loop from the depot back to itself.

2. The selection criteria that will be used to identify which node (not already in the solution) should next enter the solution.

3. The insertion criterion that will dictate into which of the two subtours and where the new entrant is placed.

With some algorithms, such as the cheapest insertion algorithm, decisions 2 and 3, that is the decisions as to which node is to be inserted and where, are made at the same time.

Once the Tour Construction algorithm creates the initial tour, the procedure loops through decisions 2 and 3 until all unallocated nodes have been allocated to a tour.

The literature on Tour Construction heuristics for the TSP suggest many variants. The variants adapted for the 2-period TSP and analysed in the Chapter are :

- Nearest Insertion,
- Cheapest Insertion,
- Arbitrary Insertion,
- Farthest Insertion,
- Nearest Addition, and
- Cheapest Addition.

In the above titles, the word "Insertion" means that when a new node is identified for inclusion in a tour, then all possible positions for that new node in the tour are quantified, and the smallest selected. On the other-hand, the word "Addition" implies that the process of selecting a node for inclusion, also identifies the position in the tour where the new node is to be placed.

All of the heuristics are implemented by the author, and the timings are based on the performance of the heuristics on a Pentium Pro PC.

7.3.1 Nearest Insertion Heuristic

Let Depot = Node number 1

V^1 = { Set of Nodes to be visited by only one tour }

V^2 = { Set of Nodes to be visited by both tours }

<u>Step 1</u> : <u>Initialisation.</u>

Initial Tour for Day_1 = { 1 - 1 }

Initial Tour for Day_2 = { 1 - 1 }

U = { Set of unallocated Nodes }

= { [V^1] + [V^2 - 1] + [V^2 - 1] }

The nodes in set V^1 are included once, and the nodes in V^2, minus the depot are included twice. The reason that the every day nodes are included twice is that each of them must be allocated twice, once to Day_1 and once to Day_2.

<u>Step 2</u> : <u>Selection.</u>

Find the node k from the set of unallocated nodes that is closest to any node in the subtours.

<u>Step 3</u> : <u>Insertion</u>

Insert k into one of the subtours between two nodes i and j so that d(i,k) + d(k,j) - d(i,j) is minimised. In this step it must be insured that each every day node only appears once in each tour.

Delete node k from the set of unallocated nodes.

<u>Loop</u> Steps 2 and Step 3 are repeated until the set of unallocated nodes is empty.

7.3.2 Cheapest Insertion

Step 1 Same as for the Nearest Insertion heuristic.

Steps 2 and 3 are combined into one step as follows :

For each node in the set of unallocated nodes, calculate the least cost way of inserting the node into one of the tours. The node with the smallest of the least insertion costs is selected. This node is inserted in the position that gave its least insertion cost.

Loop Same as for the Nearest Insertion heuristic.

7.3.3 Arbitrary Insertion

Step 1 Same as for the Nearest Insertion.

Step 2 Select a node k at random from the set of unallocated nodes.

Step 3 Same as for the Nearest Insertion.

Loop Same as for the Nearest Insertion.

7.3.4 Farthest Insertion,

Step 1 Same as for the Nearest Insertion.

Step 2 Find the node k from the set of unallocated nodes that is the farthest from any node in the subtours.

Step 3 Same as for the Nearest Insertion.

Loop Same as for the Nearest Insertion.

7.3.5 Nearest Addition

Step 1 Same as for the Nearest Insertion.

Step 2 Find the node k from the set of unallocated nodes that is the closest to any node in the subtours.

Identify the node j in the subtour that is the closest to the node to be inserted.

Step 3 Add the node k to the subtour either immediately before or after the node j, which ever is the cheapest.

Loop Same as for the Nearest Insertion.

7.3.6 Cheapest Addition

Step 1 Same as for the Nearest Insertion.

Step 2 and Step 3 are combined :

Find the node k from the set of unallocated nodes, and the node j from one of the subtours, so that the cost of inserting k immediately before or after j is a minimum.

Loop Same as for the Nearest Insertion.

7.4 Computational Results for the Tour Construction Heuristics.

Table 7.1 shows the average tour length, averaged over the 10 examples, for the various problem sizes using the different variants of the Tour Construction Heuristics.

	Problem Size, Number of Nodes						
Variant	50	100	150	200	250	500	1,000
Nearest Insertion	1,103	1,625	1,895	2,436	2,529	3,720	
Cheapest Insertion	1,090	1,588	1,824	2,317	2,419		
Arbitrary Insertion	1,014	1,481	1,737	2,237	2,343	3,682	4,551
Farthest Insertion	998	1,455	1,736	2,289	2,389	3,683	
Nearest Addition	1,158	1,709	1,987	2,565	2,615	3,806	
Cheapest Addition	1,090	1,588	1,824	2,317	2,356		

Table 7.1 : Average Heuristic Tour Length

Blank cells in Table 7.1 suggest that the heuristic was unable to solve the test problem is a reasonable processing time.

Values in a cell coloured RED in Table 7.1 are the smallest average distance for that problem size. This honour is shared among 2 heuristics - Arbitrary Insertion and Farthest Insertion. It can be argued that the Farthest Insertion heuristic brings the extreme points into the solution at an early stage, and can then accommodate later arrivals around these points. It is this approach that generates the good heuristic solution.

Table 7.1 is based on average tour lengths over 10 problems. This averaging process might be marginally distorting the results. However, the author is confident that the recommendations arising from this section have validity.

Values coloured in BLUE have the largest average distance for a given problem size. This distinction of being the worst heuristic is attached to the Nearest Addition Heuristic.

The results from Table 7.1 are shown in graphical form in Figure 7.1.

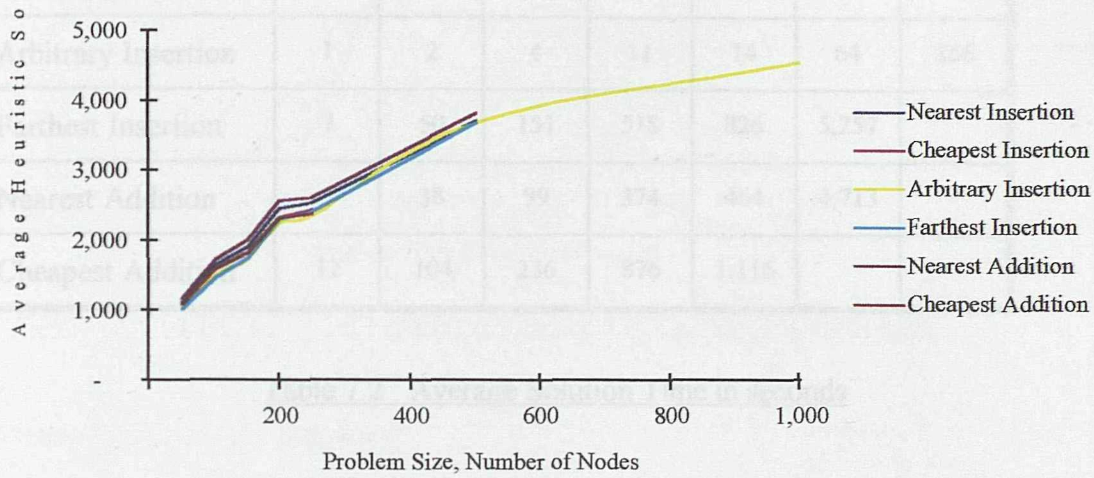


Figure 7.1 : Average Heuristic Solution versus Problem Size

Figure 7.1 shows the following :

- Cheapest Insertion and Cheapest Addition fail to solve problems over 250 nodes.
- Only Arbitrary Insertion can solve a problem of size 1,000 nodes.
- Consistently the better solutions are given by the Farthest Insertion and Arbitrary Insertion.

Typically, heuristic procedures are a trade off between quality of solution and processing time. Table 7.2 shows the average processing time, in seconds, required by each heuristic variant for each size of problem.

Variant	Problem Size, Number of Nodes						
	50	100	150	200	250	500	1,000
Nearest Insertion	6	39	101	382	550	4.995	
Cheapest Insertion	11	105	236	879	1.120		
Arbitrary Insertion	1	2	4	11	14	64	166
Farthest Insertion	7	50	151	518	826	5.257	
Nearest Addition	5	38	99	374	464	4.713	
Cheapest Addition	12	104	236	876	1.116		

Table 7.2 : Average Solution Time in seconds

The first observation from Table 7.2 is the large difference between the time taken by the Arbitrary Insertion heuristic, and all the other heuristics. Within the tour construction procedure, the selection step decides which unallocated node is next to join the solution. The cheapest, the nearest, the farthest, etc. make a large number of comparisons before the node is selected. The majority of processing time used by the heuristic is taken up by this selection step. With the Arbitrary heuristic no such time is spent. Here a node is selected at random .

Both the Cheapest Insertion and Cheapest Addition heuristics make a complex decision at the selection step. They decide at this step, not only who is to be selected from the set of unallocated nodes, but also where the selected node is to be positioned in the tour. The processing time required by the selection step dictates the overall processing time of the heuristic. Thus, the processing times taken by the Cheapest Insertion and Cheapest Addition heuristics are excessive, and under the test conditions are unable to solve a problem in excess of 250 nodes.

A graph of the processing time results is shown in Figure 7.2.

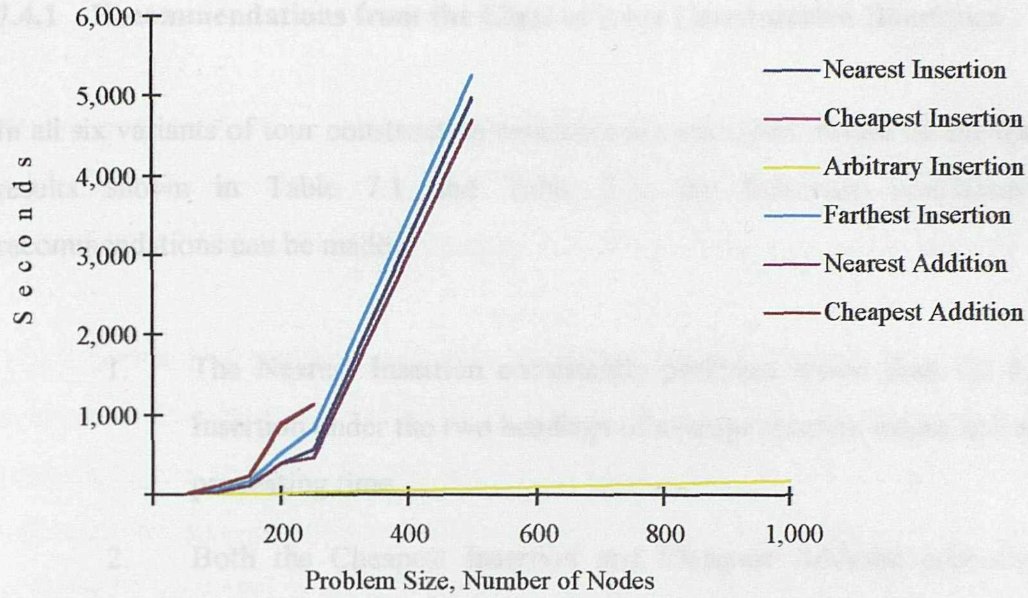


Figure 7.2 : Average Processing Time versus Problem Size

Figure 7.2 shows the following:

- Cheapest Insertion and Cheapest Addition fail to solve problems over 250 nodes.
- All heuristics, with the exception of Arbitrary Insertion, have a processing time that grows polynomially with node size.
- Arbitrary Insertion is the only realistic option for large problem sizes.

7.4.1 Recommendations from the Class of Tour Construction Heuristics

In all six variants of tour construction heuristics are examined. Based on the empirical results shown in Table 7.1 and Table 7.2, the following conclusions and recommendations can be made :

1. The Nearest Insertion consistently performs worse than the Farthest Insertion under the two headings of average solution length and average processing time.
2. Both the Cheapest Insertion and Cheapest Addition take excessive processing time, and the resulting solutions are inferior to other heuristics.
3. Arbitrary Insertion and Farthest Insertion consistently provide the best average heuristic solutions.
4. Nearest Addition provides for every test problem the worst solution.
5. Arbitrary Insertion offers the only option for solving large problems ($\geq$ 1,000 nodes).
6. Arbitrary Insertion offers, in every case, the least cost, or the 2nd least cost, heuristic solution.

In conclusion, the author recommends Arbitrary Insertion as the "best" of the tour construction heuristics.

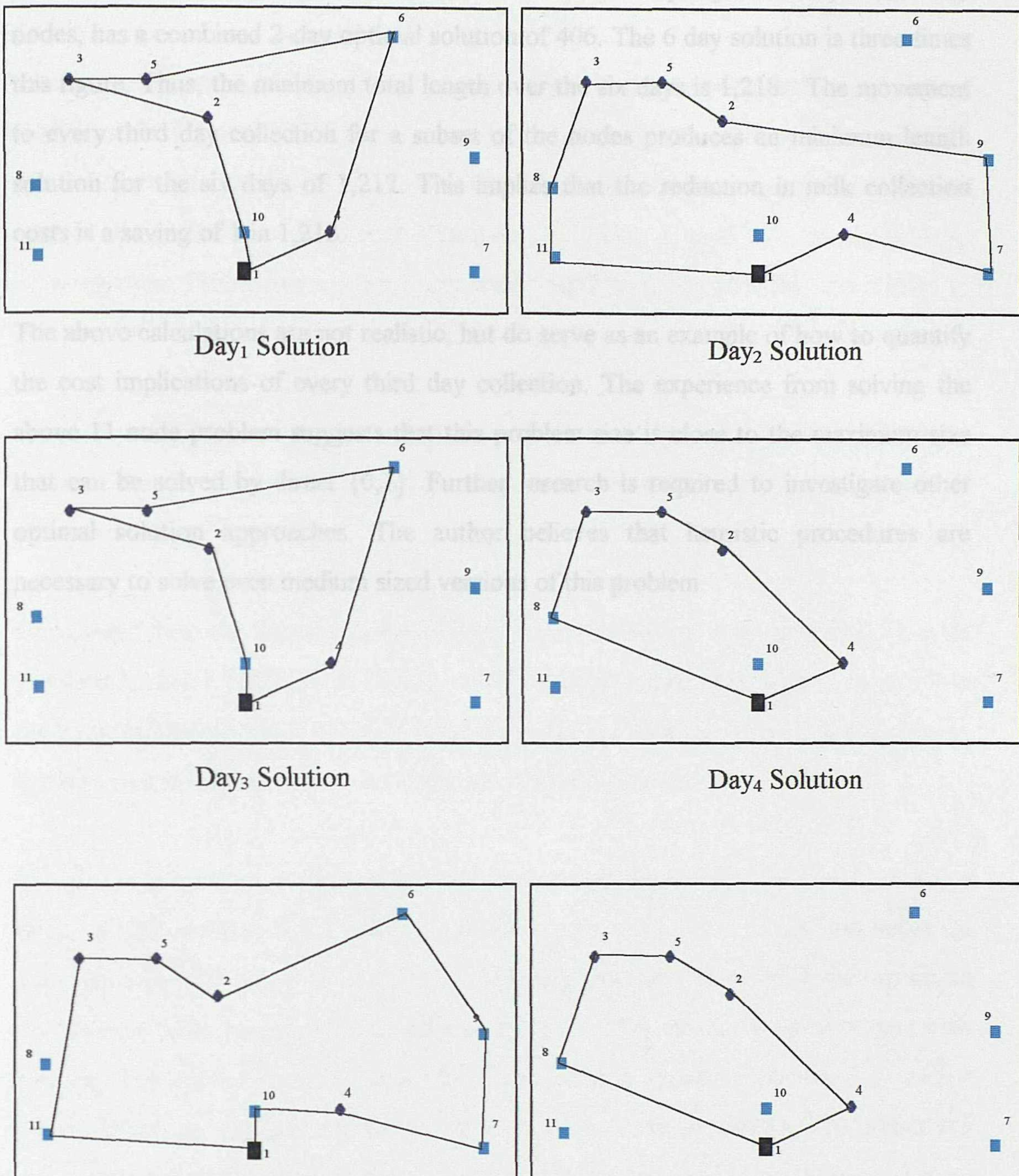


Figure 8.2 : Optimal Solution of Total Length 1,217

The optimal solution to the revised 11 node problem has a total length of 1,217.

Cluster Rule : The clustering criterion is based on whether a point falls inside or outside a 2-opt tour through all of the every day points, including the depot.

Tour Generation 2-opt tours through all of the nodes allocated to Day_1 and Day_2.

The results of this heuristic against the set of test problems is shown in Table 7.3 and Table 7.4.

7.5.2 Left Right Heuristic

In a conversation that the author had with an old milk scheduler, the scheduler suggested " that on one day he concentrates on the East of the county, while on the other day he concentrates on the West". From this conversation developed the Left-Right heuristic.

In essence a vertical line is drawn on the plane containing the nodes. All every second day nodes to the right of this line are allocated to Day_1 and all to the left are allocated to Day_2. Where the vertical line is drawn can be the subject of further analysis. For the purpose of this study, the vertical line is drawn trough the depot.

Cluster Rule : A vertical line is drawn through the depot. All every other day points to the right of this line are allocated to Day_1, the remainder are allocated to Day_2.

Tour Generation 2-opt tours through all of the nodes allocated to Day_1 and Day_2.

7.6 Computational Results for Cluster First, Route Second Heuristic

As is described earlier in the chapter, the heuristics are applied to 10 test problems from 7 different problem sizes. Table 7.3 shows the average tour length. While Table 7.4 shows the average processing time.

	Problem Size, Number of Nodes						
Variant	50	100	150	200	250	500	1,000
Inside Outside	1,007	1,450	1,693	2,218	2,301		
Left Right	997	1,421	1,641	2,125	2,205		
Tour Construction	998	1,455	1,736	2,237	2,343	3,682	4,551

Table 7.3 : Average Heuristic Tour Length

Included in Table 7.3 are the "best" results from the Tour Construction Heuristics. The conclusions from the above table are :

1. "Left Right" gives consistently better results than the "Inside Outside". There is no obvious explanation for this, except possibly due to the random nature of the test problems. The "Left Right" clustering rule might provide a more equitable partition of the every other day nodes.

2. "Left Right" gives better results that the best of the Tour Construction Heuristic.

3. The excessive processing time required by the 2-opt routing step in the "Left Right" heuristic prohibits solutions for problems in excess of 250 nodes. The explosive increase in processing time for the "Left Right" heuristic is confirmed in Table 7.4.

Variant	Problem Size, Number of Nodes						
	50	100	150	200	250	500	1,000
Inside/Outside	20	223	789	2,123	3,300		
Left Right	16	172	618	1,866	3,027		
Arbitrary Insertion	1	2	4	11	14	64	166

Table 7.4 : Average Solution Time in seconds

Included in Table 7.4 are the processing time for the Arbitrary Insertion heuristic.

It is sometimes quoted " that what you put into a heuristic, that is what you get back". Unfortunately, this does not seem to apply to the "Left Right" heuristic.

There is no real comparison between the processing time required by "Left Right" and the Arbitrary Insertion heuristics. For problems in excess of 100 nodes, the "Left Right" becomes expensive of processing time. While, the Arbitrary Insertion happily solves in reasonable time problems of 1,000 nodes.

7.7 Tour Improvement heuristic

The concept and details of a Tour Improvement procedure for the 2-period TSP is outlined in Chapter 6.

In this section 2 examples of Tour Improvement heuristics are analysed. These are :

- Tour Improvement after random initial tours, and
- Tour Improvement after Arbitrary Insertion initial tour.

7.8 Computational Results for Tour Improvement Heuristics

The computational results are shown in Table 7.5 and Table 7.6

Variant	Problem Size, Number of Nodes						
	50	100	150	200	250	500	1,000
Tour Improvement (Random Tour)	1,052	1,615	1,870				
Tour Improvement (Arbitrary Insertion)	954	1,372	1,555	2,072	2,062		
Arbitrary Insertion	1,014	1,481	1,737	2,237	2,343	3,682	4,551

Table 7.5 : Average Heuristic Tour Length

Variant	Problem Size, Number of Nodes						
	50	100	150	200	250	500	1,000
Tour Improvement (Random Tour)	150	1,525	4,725				
Tour Improvement (Arbitrary Insertion)	17	115	417	956	1,817		
Arbitrary Insertion	1	2	4	11	14	64	166

Table 7.6 : Average Solution Time in seconds

Observations on the above tables suggest :

1. Tour Improvement after a random start takes an excessive processing time to converge. Based on the quality of solution and processing time, the Tour Construction after a random start is excluded from further consideration.

2. Tour Improvement, using Arbitrary Insertion heuristic as the initial tour, provides, on average, a 9% improvement in tour length.
3. Tour Improvement heuristics use excessive processing time.

7.9 Conclusions

Table 7.7 and Table 7.8 detail the average solution length, and average processing time for the best heuristic in each class.

Variant	Problem Size, Number of Nodes						
	50	100	150	200	250	500	1,000
Tour Improvement (Arbitrary Insertion)	954	1,372	1,555	2,072	2,062		
Left Right	997	1,421	1,641	2,125	2,205		
Arbitrary Insertion	1,014	1,481	1,737	2,237	2,343	3,682	4,551

Table 7.7 : Average Heuristic Tour Length

Variant	Problem Size, Number of Nodes						
	50	100	150	200	250	500	1,000
Tour Improvement (Arbitrary Insertion)	17	115	417	956	1,817		
Left Right	16	172	618	1,866	3,027		
Arbitrary Insertion	1	2	4	11	14	64	166

Table 7.8 : Average Solution Time in seconds

Analysis of the above tables suggest :

1. Tour Improvement, with Arbitrary Insertion as the starting tour, provides a better solution in shorter processing time than the Left Right heuristic.

2. The recommendation for practical problems is to obtain a solution using the Arbitrary Insertion heuristic. Then, depending on processing power available, and the size of problem consider using the Tour Improvement approach to improve the solution quality. On average a 9% reduction can be expected from the Tour Improvement process.

7.10 Subjective Comparative Measures.

The previous sections compare heuristics based on the quality of solution produced and the computer time required to obtain these solutions. Both of these measures are quantifiable, and thus an objective comparison can be made.

A comparison of heuristics under such headings as Ease of Implementation, Flexibility and Simplicity is highly subjective. The best that can be said about the following comparison is that they reflect the views of the author, who's only qualification is that he has agonised over the computer implementation, discussed their meaning with Consultancy clients who must implement the answer, and has generally been associated with the heuristics over a period of many years.

An example of the above is a discussion that the author had with a consulting client. The author was attempting to explain to the client the basis of the heuristic that could be used to solve the client's routing problem. The client fully appreciated the rationale behind the "Inside/Outside" clustering methodology , but the client steadfastly refused to consider the "Largest Insertion" heuristic. No matter what the empirical evidence is, the client was not impressed by this approach.

In the following sections, the author allocates a score, on a scale of 1 to 10, to the various heuristics.

7.10.1 Ease of Implementation,

This criteria can be interpreted either as the ease with which the steps in the heuristic can be implemented in a computer programme, or else as the ease with which the heuristic can be used in a practical implementation.

Using the first interpretation, the author suggests that the Tour Construction Heuristics are the easiest to programme, while the Tour Improvement would be the hardest. The author's allocation of scores is :

Heuristic	**Ease of Programming the Heuristic Algorithm** Score 1 = Easy Score 10 = Hard
Tour Construction	**4**
Cluster First, Route Second	**6**
Tour Improvement	**9**

Table 7.9 : Ease of Programming Score

The alternate interpretation of the "Ease of Implementation" criteria, that is the ability of the heuristic to adapt to a practical application, yields different scores. A client will readily accept the "Inside Outside" concept. Tour Improvement procedures find favour in the field. However, "Farthest Insertion" and "Left Right" do not inspire confidence with clients. The author's scores are :

Heuristic	**Acceptability to a Client** Score 1 = Little Score 10 = Very Good
Farthest Insertion	**2**
"Left Right"	**2**
Cheapest Insertion	**6**
Nearest Insertion	**7**
"Inside/Outside"	**8**
Tour Improvement	**9**

Table 7.10 : Acceptability to Client Score

7.10.2 Flexibility

Heuristic procedures by their nature find a "good" solution within a certain neighbourhood. On occasions, the heuristic can be unlucky and the solution it finds is far from optimal. Because of this, it is desirable to be able to offer the heuristic a different "seed" and hopefully, bad luck will not strike twice, and a more acceptable solution is obtained.

Tour construction procedures offered this possibility. For instance an answer can be obtained from the Nearest Insertion heuristic. A marginal change in the programme will allow the same programme produce the Farthest Insertion solution. The flexibility that comes from a range of implementations is available with Tour Construction Heuristics. By their nature, Cluster First Route Second heuristics do not offer the same flexibility.

The ultimate in flexibility comes from the Tour Improvement class. A heuristic solution stuck at a poor solution can be shaken to a more acceptable answer using the Tour Improvement approach. The author's scores are :

Heuristic	**Flexibility** Score 1 = Rigid Score 10 = Flexible.
Cluster First, Route Second	**2**
Tour Construction	**7**
Tour Improvement	**9**

Table 7.11 : Flexibility Score

7.10.3 Simplicity

Basic to the 2-period TSP is a partitioning of the every other day nodes into two groups. Recalling how the milk scheduler solves his 2-period problem. His approach is to divide and then route. This simple concept is inherent in the Cluster First Route Second approach. At the far extreme is the "2opt" procedure. To explain the intricacies of this to a client is not easy. From experience with clients, the author's scale is as follows :

Heuristic	**Simplicity** Score 1 = Simple Score 10 = Complex
Cluster First, Route Second	**2**
Tour Construction	**4**
Tour Improvement	**9**

Table 7.12 : Simplicity to Client Score

7.11 Conclusion

Heuristics are a poor substitute for the optimal answer. Their existence derives from our inability to find an optimal answer. Heuristics should be judged on their ability to find a "good" answer in reasonable time. That task has been achieved for the 2-period TSP.

The advice for practitioners is to use the Arbitrary Insertion heuristic. As a final answer it is a "good" answer obtained with little processing time. The quality of this answer can be considerably improved, by an estimate average reduction of 9%, by post processing the solution through the Tour Improvement procedure. The limited analysis from Chapter 6 suggests that the solution from the Tour Improvement will be within 4% of the optimal.

In section 7.10, the heuristics are compared under subjective headings. The utility of this analysis is questionable, and it simply reflects the view of the author.

Chapter 8 Conclusion and Further Work

8.1 First Encounter with the 2-Period TSP

The author first encountered the 2-Period TSP in 1978 as part of a rationalisation study into the collection routes of an Irish dairy company. Advances with on-farm refrigeration allowed the dairy to collect milk from farms every second day if the farm had sufficient on-farm capacity to hold the output from two days milk production. The author was asked to investigate the implication of every other day collection. Based on very simple partitioning algorithms the routing economics seemed to suggest that savings could be made if some farms were moved from every day to every other day collection. Thus, in the late 1970's this dairy company introduced an incentive scheme to encourage farms to update their on-farm milk storage.

Consultancy assignments tend to favour a very pragmatic and often cursory approach. The client normally wants a good answer in reasonable time and not the optimal answer weeks later, and thus, the first encounter between the author and the 2-period TSP was very short. In the years that followed the author often questioned the respect given to the 2-period TSP on that first encounter. Questions such as :

- Can the 2-period TSP be formulated as a standard TSP?
- Is the 2-period TSP as difficult to solve as the standard TSP?
- Has the 2-period TSP applications outside the milk industry?
- Can optimal solution procedures be developed for the 2-Period TSP?
- Do heuristic procedures give good answers in reasonable time?
- Can efficient bounds be developed so that heuristics can be evaluated?

The above questions are all reasonable, and the author embarked on a voyage of discovery with Professor Paul Williams to answer the above questions. Luckily, the 2-period TSP rose to the challenge, and offered the author an insight into the full arena of combinatorial optimisation.

8.2 Realised Objectives

The investigation of the 2-period TSP yields a rich harvest. This is due in a large part to the number of practical routing applications that can be modelled by either the 2-period TSP or by the more general m-period TSP. Examples can be found in :

- transfer of passengers from hotels to airports,
- collection of mail from post boxes, and
- the distribution of goods to shops where different shops have different call frequencies.

Early in the investigation it became obvious that the 2-period TSP can not be transformed into the standard TSP. This means that ideas can be borrowed from the TSP, but must be re-modelled to suit the needs of the 2-period TSP. The practical applications for the formulation required information on the availability of optimal solutions for the 2-period TSP. We have shown in this thesis that examples of up to 100 nodes can be solved optimally. While for larger problems good heuristic procedures are available. As with any combinatorial type problem, the choice of best solution methodology depends on problem size. Table 8.1 lists, for the 2-period TSP, the best solution approach for various problem size.

An interesting by product of the optimal solution approach is the emergence of the VUB constraints. Some of these constraints are violated by the 2-Matching LP relaxation. Including these constraints on an "as needed" basis dramatically improves the basic LP solution. It is obviously an interesting area for further study to investigate the possibility that such variables might exist for the Standard Travelling Salesman Problem. The only way to increase the lower bound for the TSP, obtained from the LP relaxation of the 2-matching constraints, is to add generalised subtours. If VUB type constraints, with the ease with which they can be found, exist, then a better lower bound can be obtained.

Problem Size (Number of Nodes)	Recommended Solution Approach
≤ 15	Direct approach using { 0,1 } programming with subtour constraints added on an "as needed" basis.
10 to 25	At the upper end of this range a direct approach using {0,1} is very expensive of processing time. Perseverance, and good technology, will yield an optimal answer.
20 to 50	Step 1 LP relaxation including 2-matching, VUB, Subtour, and comb constraints. Step 2. {0,1} programming of the model from Step 1.
50 to 100	We are now at the limit of our ability to find an optimal solution. A man-machine interaction is the only way that an optimal answer can be obtained.
≥ 50	Advise Heuristic Procedures. However, the use of a tour improvement heuristic with a tour construction solution as a seed, will produce a "good" solution

Table 8.1 : Problem Size and "Best" Solution Approach

The author bids farewell to the 2-period TSP, but does acknowledge that considerable potential exists for further work.

8.3 Area for Further Study

Areas requiring further study include :

- every third day collection within the milk industry.
- application of the m-period TSP to distributing goods to shops.
- Re-introduction of the route capacity restriction.
- the status of the VUB, generalised subtour and comb constraints. Are they facets of the polytope?
- Alternate mathematical formulations.
- Prim's algorithm provides a greedy algorithm for the TSP. Does such a greedy algorithm exist for the 2-period TSP? Can it be extended to a 2-tree concept? Can Lagrangian relaxation be used to improve the bound?
- the use of Simulated Annealing to produce better heuristics.

Each of the above items are discussed in the following sections. The author wishes well all future researchers of the 2-period TSP.

8.3.1 Every 3rd Day Collection within the Milk Industry

In the late 1970's improved on-farm refrigeration allowed milk to be collected from farms every other day. Further improvements with both on-farm refrigeration and dairy hygiene, now, permit milk to be stored on-farm for up to three days. What implications has this for milk collection costs?

With the possibility of every 3rd day collection, a collection route will now have three types of farms. These are :

- farms requiring collection every day,
- farms requiring collection every second day, and
- farms requiring collection every third day.

The above problem can be formulated as a 6-period TSP, with

Farms to be collected every day will be collected on	Day_1, Day_2, Day_3, Day_4, Day_5, and Day_6.
Farms to be collected every second day will be collected on	Day_1, Day_3, and Day_5 or Day_2, Day_4, and Day_6.
Farms to be collected every third day will be collected on	Day_1 and Day_4 , Day_2 and Day_5 , or Day_3 and Day_6 .

To introduce a mathematical formulation of the above problem, assume that for the 11 node problem nodes 7, 9 and 11 have improved their on-farm tank capacity, and have moved to every third day collection. The revised call frequency is as shown in Table 8.2

Node	Call Type	Day_1	Day_2	Day_3	Day_4	Day_5	Day_6
1	Every	1	1	1	1	1	1
2	Every	1	1	1	1	1	1
3	Every	1	1	1	1	1	1
4	Every	1	1	1	1	1	1
5	Every	1	1	1	1	1	1
6	Second	1	1	1	1	1	1
7	Third	1	1	1	1	1	1
8	Second	1	1	1	1	1	1
9	Third	1	1	1	1	1	1
10	Second	1	1	1	1	1	1
11	Third	1	1	1	1	1	1

Table 8.2 : Call Frequency for the Revised 11 Node Problem

The solution to the above model is six tours, one for each of the six days. The objective function of the model is to minimise the total length of the 6 tours.

The decision variables are :

$X_{ijk} = 1$, if the tour on Day_k uses the link from i to j.

$= 0$, otherwise.

In addition, Y variables are used to indicate which tour the every second and every third day nodes are on. Table 8.3 shows the Y variables.

Node	Call Type	Day_1	Day_2	Day_3	Day_4	Day_5	Day_6
6	Second	$Y_{6,1}$	$Y_{6,2}$	$Y_{6,3}$	$Y_{6,4}$	$Y_{6,5}$	$Y_{6,6}$
7	Third	$Y_{7,1}$	$Y_{7,2}$	$Y_{7,3}$	$Y_{7,4}$	$Y_{7,5}$	$Y_{7,6}$
8	Second	$Y_{8,1}$	$Y_{8,2}$	$Y_{8,3}$	$Y_{8,4}$	$Y_{8,5}$	$Y_{8,6}$
9,	Third	$Y_{9,1}$	$Y_{9,2}$	$Y_{9,3}$	$Y_{9,4}$	$Y_{9,5}$	$Y_{9,6}$
10	Second	$Y_{10,1}$	$Y_{10,2}$	$Y_{10,3}$	$Y_{10,4}$	$Y_{10,5}$	$Y_{10,6}$
11	Third	$Y_{11,1}$	$Y_{11,2}$	$Y_{11,3}$	$Y_{11,4}$	$Y_{11,5}$	$Y_{11,6}$

Table 8.3 : Y Variables for the Revised 11 Node Problem

For the Every Second Day Nodes :

$Y_{i,1} + Y_{i,2} = 1$

$Y_{i,3} + Y_{i,4} = 1$

$Y_{i,5} + Y_{i,6} = 1$, and

either $Y_{i,1} + Y_{i,3} + Y_{i,5} = 0$ and $Y_{i,2} + Y_{i,4} + Y_{i,6} = 3$

or $Y_{i,2} + Y_{i,4} + Y_{i,6} = 0$ and $Y_{i,1} + Y_{i,3} + Y_{i,5} = 3$.

The above is written as

$$Y_{i,1} + Y_{i,2} = 1$$
$$Y_{i,3} + Y_{i,4} = 1$$
$$Y_{i,5} + Y_{i,6} = 1$$
$$Y_{i,1} + Y_{i,3} + Y_{i,5} = 3W_i$$
$$Y_{i,2} + Y_{i,4} + Y_{i,6} = 3Q_i$$
$$W_i + Q_i = 1$$
$$Y, W \text{ and } Q \in \{ 0,1 \}$$

Equivalent constraints for the every third day nodes are :

$$Y_{i,1} + Y_{i,2} + Y_{i,3} = 1$$
$$Y_{i,4} + Y_{i,5} + Y_{i,6} = 1$$
$$Y_{i,1} + Y_{i,4} = 2W_i$$
$$Y_{i,2} + Y_{i,5} = 2Q_i$$
$$Y_{i,3} + Y_{i,6} = 2R_i$$

$$W_i + Q_i + R_i = 1$$
$$Y, W, Q \text{ and } R \in \{ 0,1 \}$$

The model will contain a 2-matching constraint for every node for every tour. For nodes 1 to 5, the every day nodes, the right-hand-side will always be 2. For the other nodes the right-hand-side will be 2Y. This forces these nodes to be connected by two arcs to only the tours that visit these nodes.

The model formulation is given in Appendix 16.

The solution approach is to first solve the 2-matching problem The solution is shown Figure 8.1.

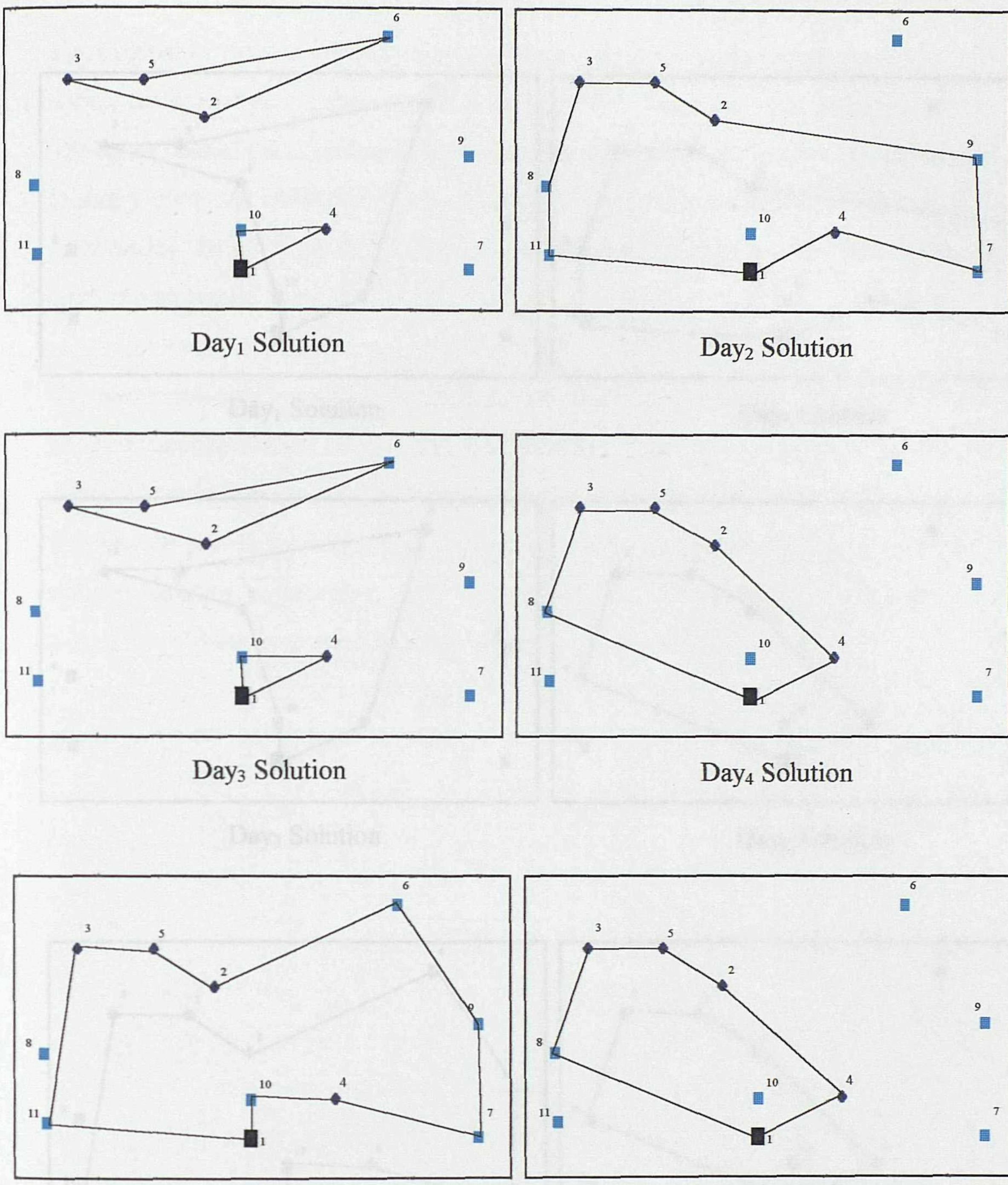
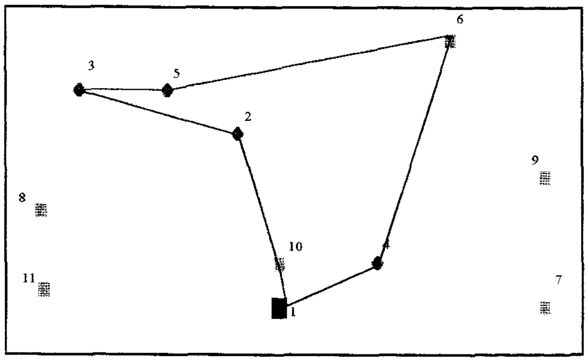
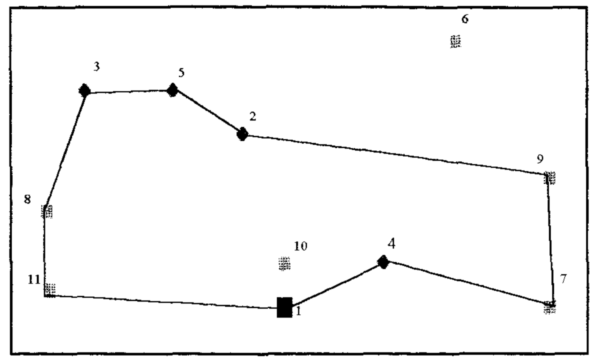


Figure 8.1 : 2-Matching Solution

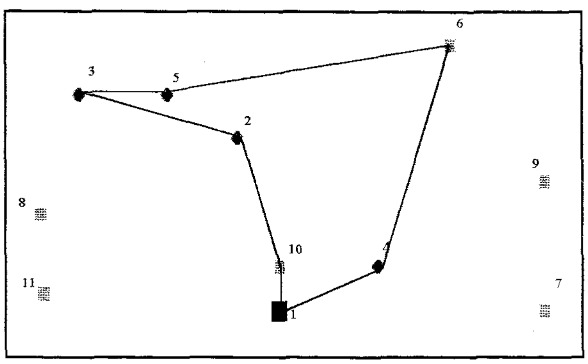
The solutions for Day_1 and Day_3 contain subtours. Subtour elimination constraints are added on an "as needed" basis, and further solutions are obtained. Unfortunately, the solution time to solve the above model is excessive. The solution shown in Figure 8.2 took several hours to obtain using CPLEX on a Pentium Pro Pc.



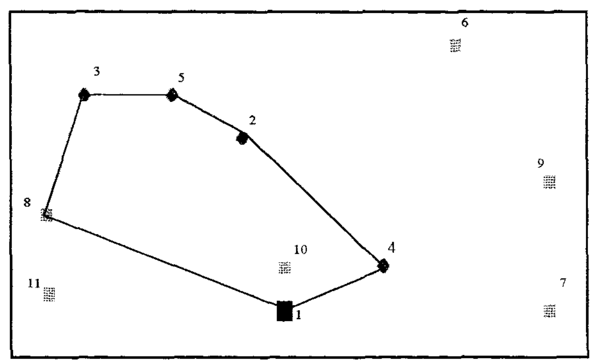
Day$_1$ Solution



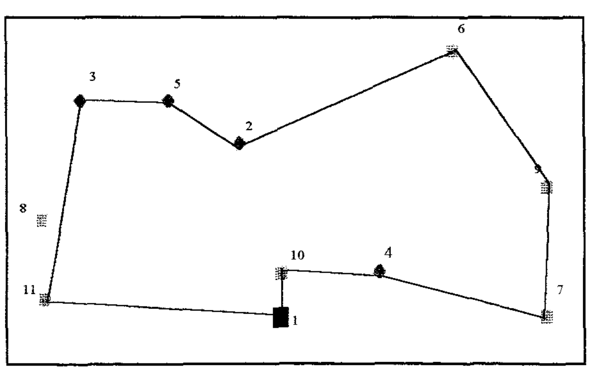
Day$_2$ Solution



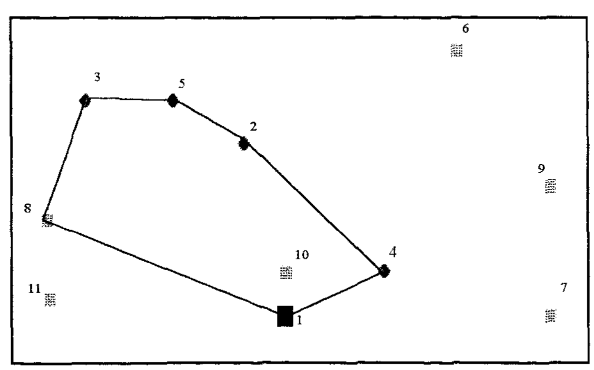
Day$_3$ Solution



Day$_4$ Solution



Day$_5$ Solution



Day$_6$ Solution

<u>Figure 8.2 : Optimal Solution of Total Length 1,217</u>

The optimal solution to the revised 11 node problem has a total length of 1,217.

The original 11 node problem, with its combination of every day and every second day nodes, has a combined 2-day optimal solution of 406. The 6 day solution is three times this figure. Thus, the minimum total length over the six days is 1,218. The movement to every third day collection for a subset of the nodes produces an minimum length solution for the six days of 1,217. This implies that the reduction in milk collection costs is a saving of 1 in 1,218.

The above calculations are not realistic, but do serve as an example of how to quantify the cost implications of every third day collection. The experience from solving the above 11 node problem suggests that this problem size is close to the maximum size that can be solved by direct {0,1}. Further research is required to investigate other optimal solution approaches. The author believes that heuristic procedures are necessary to solve even medium sized versions of this problem.

8.3.2 Delivery of Product to Customers with Different Call Frequencies

The frequency with which a distributor calls on its customers depends on the size of the customer. For example, large supermarkets might require a visit every day, medium sized shops might require a visit twice a week, while for smaller shops one visit a week might suffice. Hidden within such statements as twice a week are an implied gap between calls. Thus twice a week might really mean a call on Monday and Friday, or Tuesday and Thursday, etc.

Once options exist as to the required day on which a customer is to be visited, then the problem can be formulated using the m-period TSP model with the necessary Y variables.

Experience from the attempt to solve the 6-period, every 3rd day collection, 11 node problem suggests that optimal solutions will be hard to find. Research is required on the availability of optimal solution procedures. In the absence of an optimal answer, further research is required to develop good heuristic procedures.

The author is currently investigating a distribution fleet for Dublin County which uses a fleet of 26 refrigerated trucks to deliver milk and other dairy products to approximately 400 shops of varying size. Management are interested in the impact on distribution costs of varying the call frequency that the various shop type currently receive. The model formulation in this example is a typical capacitated m-period model. Where m will have a value of six, no calls are made on Sunday. The author has no prospect of finding the optimal solution to this problem, and is currently relying on a heuristic procedure to obtain good solutions.

8.3.3 Impact of the Route Capacity Restriction

When a milk scheduler is deciding which of the every other day farms to be allocated to a particular day, he must have regard for milk tanker capacity. Within the original problem definition for the 2-period TSP no capacity or tour length restrictions exist. For all of the practical applications noted for the 2-period TSP, most would have some capacity restriction. In discussions with academic colleagues, opinion is divided on whether the presence of the capacity restriction would make the problem easier or harder to solve. The author leaves to future researchers the specification and analysis of the capacitated 2-period TSP.

8.4 Facets of the Polytope

In Chapters 3 and 4 medium sized versions of the 2-period TSP are optimally solved using an increasingly constrained LP relaxation. The initial LP relaxation consists only of the 2-matching constraints. Constraints of three types are then added on an "as needed" basis. The classes of constraints added are :

- VUB constraints,
- Subtour elimination constraints, and
- Comb constraints.

When the LP relaxation contains no violations of the above constraint classes, the optimal solution of the model is obtained by $\{0,1\}$ programming. No attempt is made to prove that the above constraint classes are facets of the polytope. In this thesis, the classes of constraints are simply used to move the LP relaxation into the range of $\{0,1\}$ programming. The question of their status as facets is left to a future researcher.

8.5 Alternate Mathematical Formulation

The mathematical formulation for the 2-period TSP used in this thesis is based on the formulation for the TSP quoted in Miliotis [1976]. In that paper the formulation for the asymmetric TSP on a set of n nodes is :

$$\text{Minimise} \quad \sum_{i,j} C_{ij} X_{ij}$$

$$\text{Subject to} \quad \sum_{j} X_{ij} = 1 \qquad (i = 1,\ldots, n),$$

$$\sum_{i} X_{ij} = 1 \qquad (j = 1,\ldots,n),$$

$$\sum_{i \in S, j \in S} X_{ij} \leq |S| - 1 \qquad \text{S is a subset of n nodes}$$

$$X_{ij} = 0,1$$

The above formulation contains n^2 - 2 variables and 2^n + 2n - 2 constraints.

An alternate, and more compact, formulation is proposed by Miller, Tucker & Zemlin [1960]. In their formulation, in addition to the X_{ij} variables, extra indicator variables, u_i, are defined. These u_i variables, together with the constraints

$$u_i - u_j + (n-1)X_{ij} \leq n-2 \qquad (2 \leq i \leq n), \; (2 \leq j \leq n) \; and \; i \neq j$$

insure that a solution represents a feasible tour.

The complete Miller, Tucker & Zemlin formulation is :

$$\text{Minimise } \sum_{i,j} C_{ij} X_{ij}$$

$$\text{Subject to } \sum_{j} X_{ij} = 1 \qquad (1 \le i \le n),$$

$$\sum_{i} X_{ij} = 1 \qquad (1 \le j \le n),$$

$$u_i - u_j + (n-1)X_{ij} \le n-2 \qquad (2 \le i \le n),$$

$$(2 \le j \le n) \; and \; i \ne j$$

$$X_{ij} = 0,1$$

$$u_i \ge 0 \qquad (2 \le i \le n)$$

The above formulation has $n^2 + n$ constraints and n^2 variables Thus, this formulation has the advantage in that it contains significantly less constraints that the Miliotis formulation. The importance of this advantage is questionable. In practice, using Miliotis formulation, if subtour elimination constraints are added on an "as needed" basis, then only a small number of constraints are required.

Versions of the 2-period TSP are solved in this thesis using the Miliotis type formulation. Research is required to investigate how the Miller formulation can be adapted to the 2-period TSP.

Claus [1984] outlines a further formulation for the TSP. This formulation replaces the exponential number subtour elimination constrains by a number of constraints that is proportional to the number of nodes times the number of finite cost arcs in the graph. In addition, the new formulation introduces a new set of variables. Claus argues that the resulting polytope is smaller than the subtour elimination polytope. Research is required to investigate the importance of this formulation to the 2-period TSP.

8.6 Greedy Algorithm

The minimum spanning tree, MST, is a well known lower bound for the TSP. A 1-tree is formed by adding a selected arc to the MST. The 1-tree contains as many arcs as does a TSP tour, and provides a lower bound on the length of the optimal tour. The quality of this bound can improved considerably through the use of Lagrangian relaxation. The MST and 1-tree concept is of significant importance largely due to the fact that a greedy algorithm exists that finds an optimal solution.

It is easy to adapt Prim's algorithm to the 2-period problem - an algorithm used by the author grows two trees, one for Day_1 and one for Day_2 , in a manner similar to Prim. If an additional arc is added to both of these trees, then a graph exists, possibly termed a 2-tree, that has the same number of arcs as the optimal solution to the 2-period TSP.

Has the 2-tree graph the same property as the 1-tree for the TSP, that is, is the length of the 2-tree graph a lower bound on the optimal solution to the 2-period TSP? While investigating the possible bounds for the 2-period TSP, the author was initially convinced that the 2-tree was in fact a lower bound. This confidence was shaken when the 2-tree bound was increased through Lagrangian relaxation to a value above a known optimal solution. Obviously in this latter example a flaw exists. However, where is the flaw - is this flaw in the implementation of the greedy algorithm, the choice of the two arcs to be added, or the Lagrangian relaxation.

The questions left to a future researcher include :

- Can Prim's greedy algorithm be adapted to provide a guaranteed lower bound for the 2-period TSP?
- Can two additional arcs be added to improve this bound?
- Can the bound be improved through Lagrangian relaxation?

The author is concerned that the partitioning component of the 2-period TSP makes the greedy concept inappropriate.

8.7 Simulated Annealing

Chapter 6 analyses various heuristic procedures for the 2-period TSP. Table 6.1 summarises the findings, and suggests that tour improvement procedures are a useful solution technique for the 2-Period TSP. Kirkpatrick, Gelatt and Vecchi [1982] introduced the concept of adding a probabilistic dimension to either accepting or rejecting tour modifications in a tour improvement procedure. This process, termed Simulated Annealing, has shown potential with other combinatorial problems. Future research is required to investigate the utility of Simulated Annealing to the 2-period TSP.

Appendices

Appendix 1

11 Node Problem

Node Number	X Coord.	Y Coordinate		
		Every Day	Every Second Day	Depot
1	50			10
2	40	50		
3	20	60		
4	60	20		
5	30	60		
6	70		80	
7	90		10	
8	10		30	
9	90		40	
10	50		20	
11	10		10	

Distance Matrix

	1	2	3	4	5	6	7	8	9	10	11
1		41	58	14	54	73	40	45	50	10	40
2			22	36	14	42	64	36	51	32	50
3				57	10	54	86	32	73	50	51
4					50	61	32	51	36	10	51
5						45	78	36	63	45	54
6							73	78	45	63	92
7								82	30	41	80
8									81	41	20
9										45	85
10											41

Appendix 2

21 Node Problem

X, Y Coordinates

Node Number	X Coord.	Y Coordinate		
		Every Day	Every Second Day	Depot
1	50			10
2	30	50		
3	90	40		
4	50	50		
5	30	80		
6	10	60		
7	40	40		
8	80	80		
9	20	10		
10	10		30	
11	70		10	
12	90		20	
13	80		60	
14	70		50	
15	50		70	
16	10		80	
17	30		30	
18	60		60	
19	70		70	
20	90		90	
21	20		40	

21 Node Problem - Distance Matrix.

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21
1		45	50	40	73	64	32	76	30	45	20	41	58	45	60	81	28	51	63	89	42
2			61	20	30	22	14	58	41	28	57	67	51	40	28	36	20	32	45	72	14
3				41	72	82	50	41	76	81	36	20	22	22	50	89	61	36	36	50	70
4					36	41	14	42	50	45	45	50	32	20	20	50	28	14	28	57	32
5						28	41	50	71	54	81	85	54	50	22	20	50	36	41	61	41
6							36	73	51	30	78	89	70	61	41	20	36	50	61	85	22
7								57	36	32	42	54	45	32	32	50	14	28	42	71	20
8									92	86	71	61	20	32	32	70	71	28	14	14	72
9										22	50	71	78	64	67	71	22	64	78	106	30
10											63	81	76	63	57	50	20	58	72	100	14
11												22	51	40	63	92	45	51	60	82	58
12													41	36	64	100	61	50	54	70	73
13														14	32	73	58	20	14	32	63
14															28	67	45	14	20	45	51
15																41	45	14	20	45	42
16																	54	54	61	81	41
17																		42	57	85	14
18																			14	42	45
19																				28	58
20																					86

Appendix 3

42 Node Problem

Node Number	X Coord.	Y Coordinate		
		Every Day	Every Second Day	Depot
1	210			19
2	141	103		
3	95	53		
4	174	61		
5	150	193		
6	161	166		
7	201	139		
8	189	191		
9	241	127		
10	248	73		
11	242	140		
12	278	118		
13	91	107		
14	91		155	
15	105		135	
16	93		81	
17	137		57	
18	131		17	
19	182		97	
20	201		77	
21	219		66	
22	164		128	
23	134		144	
24	127		167	
25	116		185	
26	146		161	
27	150		159	
28	193		168	
29	204		121	
30	205		162	
31	210		182	
32	227		164	
33	224		141	
34	236		91	
35	263		96	
36	248		156	
37	230		176	
38	278		90	
39	266		61	
40	66		116	
41	62		151	
42	42		86	

Appendix 4

Fractional Solution satisfying all sub-tour elimination constraints

i	j	X_{ij1}	X_{ij2}	$X_{ij1} + X_{ij2}$
1	4	1.	1.	2.
1	10		0.2	0.2
1	18		0.8	0.8
1	21	1.		1.
2	13	0.75	1.	1.75
2	19	0.2	0.6	0.8
2	22	0.8	0.2	1.
2	23	0.25	0.2	0.45
3	13	0.1	0.1	0.2
3	16	0.1	0.9	1.
3	17	0.6	0.2	0.8
3	18	0.2	0.8	1.
3	42	1.		1.
4	17	0.8	.2	1.
4	20	0.2	.8	1.
5	6	1.	1	2.
5	8	0.35	.65	1.
5	25	0.65	.35	1.
6	8	0.65	.35	1.
6	26	0.12	.133333	0.25
6	27	0.23	.516667	0.75
7	22	0.8	.2	1.
7	29	0	1	1.
7	30	0.6	.4	1.
7	33	0.6	.4	1.
8	28	0.6	.4	1.
8	31	0.4	.6	1.
9	11	1.	1.	2.
9	12	1.	1.	2.
10	21	1.		1.
10	34	0.2	0.8	1.
10	35	0.8		0.8
10	39		1.	1.
11	33	0.6	0.4	1.
11	36	0.4	0.6	1.
12	35	1.		1.
12	38		1.	1.
13	15	0.95		0.95
13	16	0.1	0.9	1.
13	40	0.05		0.05
13	42	0.05		0.05
14	15	1.		1
14	41	1.		1
15	23	0.05		0.05
17	18	0.2		0.2
19	20	0.2		0.2

19	29		1.	1.
20	34		0.8	0.8
23	24	0.53	0.22	0.75
23	26	0.12	0.38	0.5
23	27	0.12	0.13	0.25
24	25	0.65	0.35	1.
24	26	0.12	0.13	0.25
26	27	0.35	0.65	1.
28	30	0.6	0.4	1.
31	37	0.4	0.6	1.
32	36	0.4	0.6	1.
32	37	0.4	0.6	1.
34	35	0.2		0.2
38	39		1.	1.
40	41	1.		1.
40	42	0.95		0.95

Appendix 5

Fractional Solution satisfying the 1st Set of Comb Constraints

i	j	X_{ij1}	X_{ij2}	$X_{ij1} + X_{ij2}$
1	4	1	1	2
1	10	.473118		.473118
1	21	.526882	.473118	1
2	13	.408602	1	1.408602
2	19	.032258	.55914	.591398
2	22	1		1
2	23	.55914	.44086	1
3	13	.236559	.236559	.473118
3	16	.236559	.763441	1
3	17	.526882		.526882
3	42	1		1
4	17	.526882	.473118	1
4	20	.473118	.526882	1
5	6	1	1	2
5	8	.27957	.72043	1
5	25	.72043	.27957	1
6	8	.72043	.27957	1
6	26	.139785	.139785	.27957
6	27	.139785	.580645	.72043
7	22	1		1
7	29	.064516	.935484	1
7	30	.5	.5	1
7	33	.435484	.532258	.967742
8	28	.467742	.532258	1
8	31	.532258	.467742	1
9	11	1	1	2
9	12	1	1	2
10	21	.053763	.473118	.526881
10	34	.473118	.526882	1
10	35	1		1
11	32	.032258		.032258
11	33	.435484	.532258	.967742
11	36	.532258	.467742	1
12	35	1		1
13	15	.88172		.88172
13	16	.236559	.763441	1
13	40	.11828		.11828
13	42	.11828		.11828
14	15	1		1
14	41	1		1
15	23	.11828		.11828
19	20	.473118		.473118
21	34	.473118		.473118
23	24	.44086		.44086
24	25	.72043	.27957	1
24	26	.139785	.139785	.27957

24	27	.139785	.139785	.27957
26	27	.27957	.72043	1
28	30	.467742	.5	.967742
29	33	.064516		.064516
30	31	.032258		.032258
31	37	.5	.467742	.967742
32	36	.5	.467742	.967742
32	37	.532258	.467742	1
36	37	.032258		.032258
40	41	1		1
40	42	.88172		.88172
1	18		.526882	.526882
3	18		1	1
7	28		.032258	.032258
10	39		1	1
12	38		1	1
17	18		.473118	.473118
19	29		.935484	.935484
20	34		.526882	.526882
23	26		.44086	.44086
38	39		1	1

Appendix 6

Optimal Solution to the 42 node problem

i	j	X_{ij1}	X_{ij2}	$X_{ij1} + X_{ij2}$
1	4	1	1	2
1	10	1		1
1	21		1	1
2	13	1	1	2
2	22	1		1
2	23		1	1
3	13		1	1
3	16	1		1
3	18		1	1
3	42	1		1
4	17		1	1
4	20	1		1
5	6	1	1	2
5	8		1	1
5	25	1		1
6	8	1		1
6	27		1	1
7	22	1		1
7	29	1		1
7	30		1	1
7	33		1	1
8	28		1	1
8	31	1		1
9	11	1	1	2
9	12	1	1	2
10	21		1	1
10	34	1		1
10	39		1	1
11	33		1	1
11	36	1		1
12	35	1		1
12	38		1	1
13	16	1		1
14	15	1		1
14	41	1		1
15	24	1		1
17	18		1	1
19	20	1		1
19	29	1		1
23	26		1	1
24	25	1		1
26	27		1	1
28	30		1	1
31	37	1		1
32	36	1		1
32	37	1		1

34	35	1		1
38	39		1	1
40	41	1		1
40	42	1		1

Appendix 7

Program to automatically detect violated sub-tour elimination constraints.

The complete IP formulation of the 2-Period TSP contains all possible sub-tour elimination constraints. The solution methodology adopted in this thesis is to add constraints on an "as needed" basis to the LP relaxation. A programme was written to identify within an LP solution all violations of the sub-tour inequalities. The programme was based on an algorithm described by Dr Derek O'Connor, a friend and departmental colleague at University College Dublin.

The algorithm first identifies all subtours within the LP solution. For each sub-tour the programme then classifies the sub-tour into 1 of 3 categories. The categories depend on whether S, the set of nodes in the sub-tour, contains all, some, or none of both day nodes. The inequalities that the sum of the X and, in some cases, the Y values for the sub-tour must satisfies depends on the categorisation of the sub-tour. In the event that a particular sub-tour violates an inequality, the programme writes to a computer file a constraint to prevent the sub-tour occurring in a later solution. The file containing all constraints is appended to the LP Relaxation, and the CPLEX package solves the new model.

The loop consisting of:

- CPLEX solution of the LP Model
- Identification of sub-tour violations.
- Generation of constraints into a computer file.
- LP Model extended by the addition of the new constraints.

was programmed in a DOS Batch File, and the process terminated when the LP solution contained no sub-tour violations.

The computer programme was written in Basic, and designed to run on a standard PC. A copy of the computer code is available from the author.

Appendix 8

Constraints introduced by Stages 2, 3, 4, and 5.

Constraint to remove symmetry between Day 1 and Day 2.

.c114: Y0141 = 1

VUB Constraints.

c115: X0140151 - Y0151 <= 0
c116: X0030161 - Y0161 <= 0
c117: X0030181 - Y0181 <= 0
c118: X0020191 - Y0191 <= 0
c119: X0040201 - Y0201 <= 0
c120: X0010211 - Y0211 <= 0
c121: X0020221 - Y0221 <= 0
c122: X0230241 - Y0231 <= 0
c123: X0230241 - Y0241 <= 0
c124: X0050251 - Y0251 <= 0
c125: X0260271 - Y0261 <= 0
c126: X0080281 - Y0281 <= 0
c127: X0070291 - Y0291 <= 0
c128: X0070301 - Y0301 <= 0
c129: X0080311 - Y0311 <= 0
c130: X0320371 - Y0321 <= 0
c131: X0090331 - Y0331 <= 0
c132: X0100341 - Y0341 <= 0
c133: X0120351 - Y0351 <= 0
c134: X0110361 - Y0361 <= 0
c135: X0320371 - Y0371 <= 0
c136: X0120381 - Y0381 <= 0
c137: X0100391 - Y0391 <= 0
c138: X0130421 - Y0421 <= 0
c139: X0130152 - Y0152 <= 0
c140: X0030162 - Y0162 <= 0
c141: X0170182 - Y0182 <= 0
c142: X0020192 - Y0192 <= 0
c143: X0040202 - Y0202 <= 0
c144: X0010212 - Y0212 <= 0
c145: X0020222 - Y0222 <= 0
c146: X0230242 - Y0232 <= 0
c147: X0230242 - Y0242 <= 0
c148: X0050252 - Y0252 <= 0
c149: X0060262 - Y0262 <= 0
c150: X0080282 - Y0282 <= 0

c151: X0070292 - Y0292 <= 0
c152: X0070302 - Y0302 <= 0
c153: X0080312 - Y0312 <= 0
c154: X0320372 - Y0322 <= 0
c155: X0090332 - Y0332 <= 0
c156: X0100342 - Y0342 <= 0
c157: X0120352 - Y0352 <= 0
c158: X0110362 - Y0362 <= 0
c159: X0320372 - Y0372 <= 0
c160: X0120382 - Y0382 <= 0
c161: X0100392 - Y0392 <= 0
c162: X0130422 - Y0422 <= 0
c163: X0030171 - Y0171 <= 0
c164: X0020231 - Y0231 <= 0
c165: X0240251 - Y0241 <= 0
c166: X0240251 - Y0251 <= 0
c167: X0060261 - Y0261 <= 0
c168: X0110331 - Y0331 <= 0
c169: X0140411 - Y0411 <= 0
c170: X0030172 - Y0172 <= 0
c171: X0030182 - Y0182 <= 0
c172: X0020232 - Y0232 <= 0
c173: X0240252 - Y0242 <= 0
c174: X0240252 - Y0252 <= 0
c175: X0260272 - Y0262 <= 0
c176: X0070332 - Y0332 <= 0
c177: X0400412 - Y0412 <= 0
c178: X0060272 - Y0272 <= 0
c179: X0110332 - Y0332 <= 0
c180: X0260271 - Y0271 <= 0
c181: X0070331 - Y0331 <= 0
c182: X0060271 - Y0271 <= 0
c183: X0400411 - Y0401 <= 0
c184: X0260272 - Y0272 <= 0
c185: X0130402 - Y0402 <= 0

Generalised Sub-tour Elimination Constraints

c186: X0030171 + X0030181 + X0170181 - Y0171 - Y0181 <= 0

c187: X0030172 + X0030182 + X0170182 - Y0172 - Y0182 <= 0

c188: X0060261 + X0060271 + X0260271 - Y0261 - Y0271 <= 0

c189: X0060262 + X0060272 + X0260272 - Y0262 - Y0272 <= 0

c190: X0130151 + X0130401 + X0140151 + X0140411 + X0400411 - Y0141 - Y0151 - Y0401 - Y0411 <= 0

c191: X0130152 + X0130402 + X0140152 + X0140412 + X0400412 - Y0142 - Y0152 - Y0402 - Y0412 <= 0

c192: X0400421 - Y0421 <= 0

c193: X0030422 - Y0422 <= 0

c194: X0050061 + X0050081 + X0060081 <= 2

c195: X0050062 + X0050082 + X0060082 <= 2

c196: X0100341 + X0100391 + X0340391 - Y0341 - Y0391 <= 0

c197: X0100342 + X0100392 + X0340392 - Y0342 - Y0392 <= 0

c198: X0120351 + X0120381 + X0350381 - Y0351 - Y0381 <= 0

c199: X0120352 + X0120382 + X0350382 - Y0352 - Y0382 <= 0

c200: X0010041 + X0010101 + X0020131 + X0020191 + X0030181 + X0030421 + X0040171 + X0070291 + X0070331 + X0090111 + X0090121 + X0100351 + X0110331 + X0120351 + X0130151 + X0140151 + X0140411 + X0170181 + X0190291 + X0400411 + X0400421 - Y0141 - Y0151 - Y0171 - Y0181 - Y0191 - Y0291 - Y0331 - Y0351 - Y0401 - Y0411 - Y0421 <= 9

c201: X0010042 + X0010102 + X0020132 + X0020192 + X0030182 + X0030422 + X0040172 + X0070292 + X0070332 + X0090112 + X0090122 + X0100352 + X0110332 + X0120352 + X0130152 + X0140152 + X0140412 + X0170182 + X0190292 + X0400412 + X0400422 - Y0142 - Y0152 - Y0172 - Y0182 - Y0192 - Y0292 - Y0332 - Y0352 - Y0402 - Y0412 - Y0422 <= 9

c202: X0020191 + X0020221 + X0070221 + X0070291 + X0190291 - Y0191 - Y0221 - Y0291 <= 1

c203: X0020192 + X0020222 + X0070222 + X0070292 + X0190292 - Y0192 - Y0222 - Y0292 <= 1

c204: X0010041 + X0010211 + X0040201 + X0200211 - Y0201 - Y0211 <= 1

c205: X0010042 + X0010212 + X0040202 + X0200212 - Y0202 - Y0212 <= 1

c206: X0050061 + X0050251 + X0060271 + X0240251 + X0240261 + X0260271 - Y0241 - Y0251 - Y0261 - Y0271 <= 1

c207: X0050062 + X0050252 + X0060272 + X0240252 + X0240262 + X0260272 - Y0242 - Y0252 - Y0262 - Y0272 <= 1

c208: X0030161 + X0030421 + X0130151 + X0130161 + X0140151 + X0140411 + X0400411 + X0400421 - Y0141 - Y0151 - Y0161 - Y0401 - Y0411 - Y0421 <= 1

c209: X0030162 + X0030422 + X0130152 + X0130162 + X0140152 + X0140412 + X0400412 + X0400422 - Y0142 - Y0152 - Y0162 - Y0402 - Y0412 - Y0422 <= 1

c210: X0010041 + X0010201 + X0040201 - Y0201 <= 1 c211: X0010042 + X0010202 + X0040202 - Y0202 <= 1

c212: X0050061 + X0050251 + X0060271 + X0230241 + X0230261 + X0240251 + X0260271 - Y0231 - Y0241 - Y0251 - Y0261 - Y0271 <= 1

c213: X0050062 + X0050252 + X0060272 + X0230242 + X0230262 + X0240252 + X0260272 - Y0232 - Y0242 - Y0252 - Y0262 - Y0272 <= 1

c214: X0030131 + X0030421 + X0130151 + X0140151 + X0140411 + X0400411 + X0400421 - Y0141 - Y0151 - Y0401 - Y0411 - Y0421 <= 1

c215: X0030132 + X0030422 + X0130152 + X0140152 + X0140412 + X0400412 + X0400422 - Y0142 - Y0152 - Y0402 - Y0412 - Y0422 <= 1

c216: X0230261 - Y0231 <= 0

c217: X0030131 + X0030161 + X0130161 - Y0161 <= 1

c218: X0030132 + X0030162 + X0130162 - Y0162 <= 1

c219: X0060261 + X0060271 + X0260271 - Y0261 - Y0271 <= 0

c220: X0060262 + X0060272 + X0260272 - Y0262 - Y0272 <= 0

c221: X0030421 - Y0421 <= 0

c222: X0400422 - Y0402 <= 0

c223: X0140151 + X0140411 + X0150411 - Y0141 - Y0151 <= 0

c224: X0140152 + X0140412 + X0150412 - Y0142 - Y0152 <= 0

c225: X0140151 + X0140411 + X0150411 - Y0141 - Y0411 <= 0

c226: X0140152 + X0140412 + X0150412 - Y0142 - Y0412 <= 0

c227: X0140151 + X0140411 + X0150411 - Y0151 - Y0411 <= 0

c228: X0140152 + X0140412 + X0150412 - Y0152 - Y0412 <= 0

c229: X0130151 + X0130401 + X0140151 + X0140411 + X0400411 - Y0141 - Y0151 - Y0401 - Y0411 <= 0

c230: X0130152 + X0130402 + X0140152 + X0140412 + X0400412 - Y0142 - Y0152 - Y0402 - Y0412 <= 0

c231: X0190202 - Y0192 <= 0

c232: X0020191 + X0020221 + X0070221 + X0070291 + X0190291 - Y0191 - Y0221 - Y0291 <= 1

c233: X0020192 + X0020222 + X0070222 + X0070292 + X0190292 - Y0192 - Y0222 - Y0292 <= 1

c234: X0010041 + X0010101 + X0040201 + X0100341 + X0200341 - Y0201 - Y0341 <= 2

c235: X0010042 + X0010102 + X0040202 + X0100342 + X0200342 - Y0202 - Y0342 <= 2

Comb Constraints

c236: X0050061 + X0050081 + X0050251 + X0060081 + X0060261 + X0060271
+ X0070291 + X0070301 + X0070331 + X0080281 + X0080311 + X0090111
+ X0110331 + X0110361 + X0230241 + X0230261 + X0230271 + X0240251
+ X0240261 + X0260271 + X0280301 + X0310371 + X0320361 + X0320371
+ X0050062 + X0050082 + X0050252 + X0060082 + X0060262 + X0060272
+ X0070292 + X0070302 + X0070332 + X0080282 + X0080312 + X0090112
+ X0110332 + X0110362 + X0230242 + X0230262 + X0230272 + X0240252
+ X0240262 + X0260272 + X0280302 + X0310372 + X0320362 + X0320372
<= 22

c237:X0030131 + X0030161 + X0030171 + X0030181 + X0040171 + X0030421
+ X0130161 + X0130151 + X0170181 + X0140151 + X0140411 + X0400411
+ X0130401 + X0400421 + X0130421
+ X0030132 + X0030162 + X0030172 + X0030182 + X0040172 + X0030422
+ X0130162 + X0130152 + X0170182 + X0140152 + X0140412 + X0400412
+ X0130402 + X0400422 + X0130422 <= 11

Appendix 9

100 Node Problem

Node Number	X Coord.	Y Coordinate		
		Every Day	Every Second Day	Depot
1	70			50
2	57	28		
3	30	77		
4	1	76		
5	81	70		
6	4	41		
7	86	79		
8	37	96		
9	87	5		
10	94	36		
11	52	76		
12	5	59		
13	46	29		
14	62	64		
15	26	27		
16	82	82		
17	58	98		
18	91	22		
19	69	98		
20	24	53		
21	10	99		
22	67	1		
23	57	10		
24	10	79		
25	28	4		
26	29	38		
27	30	94		
28	97	40		
29	27	16		
30	16	64		
31	41	41		
32	71	32		
33	63	20		
34	18	58		
35	8	45		
36	90	26		
37	78	37		
38	28	91		
39	63	62		
40	42	9		
41	56		69	
42	91		83	

Node Number	X Coord.	Y Coordinate		
		Every Day	Every Second Day	Depot
43	2		54	
44	91		43	
45	67		50	
46	51		46	
47	35		40	
48	26		5	
49	24		97	
50	6		39	
51	36		48	
52	15		47	
53	25		62	
54	54		15	
55	93		65	
56	50		39	
57	10		78	
58	45		75	
59	59		83	
60	1		21	
61	7		10	
62	33		12	
63	0		53	
64	65		54	
65	82		8	
66	19		67	
67	45		35	
8	14		70	
69	92		53	
70	8		75	
71	40		46	
72	49		20	
73	32		9	
74	58		16	
75	92		9	
76	44		27	
77	87		75	
78	27		67	
79	25		8	
80	3		32	
81	79		29	
82	23		48	
83	25		34	
84	4		48	
85	20		86	
86	58		75	
87	92		33	
88	54		8	

Node Number	X Coord.	Y Coordinate		
		Every Day	Every Second Day	Depot
89	63		41	
90	96		11	
91	92		62	
92	34		14	
93	47		21	
94	99		13	
95	2		34	
96	54		92	
97	53		40	
98	84		82	
99	67		72	
100	99		33	

Appendix 10

200 Node Problem

Node Number	X Coord.	Y Coordinate		
		Every Day	Every Second Day	Depot
1	55			23
2	73	6		
3	21	27		
4	76	74		
5	11	7		
6	21	16		
7	85	7		
8	76	4		
9	86	33		
10	37	34		
11	83	9		
12	30	6		
13	40	73		
14	55	10		
15	14	89		
16	23	86		
17	72	83		
18	59	41		
19	39	13		
20	3	42		
21	68	51		
22	37	17		
23	34	4		
24	52	79		
25	55	47		
26	25	76		
27	39	93		
28	28	74		
29	49	75		
30	62	51		
31	6	68		
32	14	27		
33	4	21		
34	40	17		
35	56	26		
36	72	5		
37	64	5		
38	51	48		
39	18	22		
40	87	99		
41	11	28		
42	39	62		
43	89	18		

Node Number	X Coord.	Y Coordinate		
		Every Day	Every Second Day	Depot
44	24	68		
45	17	18		
46	50	73		
47	79	33		
48	73	90		
49	93	14		
50	81	38		
51	62	95		
52	0	17		
53	39	45		
54	30	12		
55	26	91		
56	98	19		
57	94	52		
58	47	95		
59	24	12		
60	38	36		
61	85	1		
62	85	80		
63	22	12		
64	15	30		
65	80	40		
66	47	8		
67	49	48		
68	3	31		
69	34	51		
70	37	13		
71	51	37		
72	0	91		
73	44	99		
74	13	77		
75	36	4		
76	83	33		
77	93	81		
78	32	75		
79	25	33		
80	55	98		
81	4	77		
82	4	28		
83	7	12		
84	95	99		
85	90	0		
86	41	54		
87	82	89		
88	8	69		
89	3	46		

Node Number	X Coord.	Y Coordinate		
		Every Day	Every Second Day	Depot
90	3	12		
91	13		22	
92	77		11	
93	82		44	
94	58		61	
95	81		24	
96	10		49	
97	7		4	
98	19		4	
99	27		15	
100	93		76	
101	47		71	
102	4		37	
103	9		76	
104	39		2	
105	10		72	
106	83		3	
107	48		94	
108	34		24	
109	92		67	
110	26		42	
111	78		13	
112	95		35	
113	29		98	
114	56		36	
115	99		87	
116	13		13	
117	46		41	
118	94		65	
119	24		26	
120	7		60	
121	7		54	
122	18		53	
123	30		48	
124	70		24	
125	65		28	
126	8		20	
127	9		29	
128	25		9	
129	11		37	
130	80		30	
131	15		55	
132	70		65	
133	28		88	
134	77		43	
135	18		65	

Node Number	X Coord.	Y Coordinate		
		Every Day	Every Second Day	Depot
136	28		68	
137	34		71	
138	75		66	
139	51		11	
140	1		66	
141	42		10	
142	57		98	
143	63		0	
144	17		25	
145	93		1	
146	59		87	
147	64		46	
148	79		84	
149	32		92	
150	17		73	
151	80		24	
152	18		99	
153	32		0	
154	17		44	
155	1		13	
156	42		1	
157	59		80	
158	50		80	
159	31		53	
160	85		35	
161	17		66	
162	57		11	
163	37		90	
164	56		23	
165	0		29	
166	80		58	
167	1		29	
168	93		10	
169	41		21	
170	12		90	
171	70		82	
172	10		83	
173	61		26	
174	50		69	
175	3		18	
176	45		94	
177	18		11	
178	40		5	
179	25		15	
180	17		36	
181	72		79	

Node Number	X Coord.	Y Coordinate		
		Every Day	Every Second Day	Depot
182	32		2	
183	73		9	
184	68		29	
185	2		10	
186	25		3	
187	50		44	
188	18		60	
189	89		23	
190	21		68	
191	71		21	
192	14		53	
193	49		15	
194	74		73	
195	18		48	
196	78		75	
197	90		18	
198	38		7	
199	85		93	
200	59		34	

Appendix 11

Heuristic Solutions for the 11 Node Problem

Cheapest Insertion	Day 1 Tour :	Depot - 4 - 7 - 9 - 6 - 2 - 5 - 3 - 10 - Depot
	Day 2 Tour :	Depot - 4 - 2 - 5 - 3 -8 - 11 - Depot

Inside/Outside	Day 1 Tour :	Depot - 4 - 7 - 9 - 6 - 2 - 5 - 3 - 8 - 11 - Depot
	Day 2 Tour :	Depot - 4 - 2 - 5 - 3 -10 - Depot

Tour Improvement : (Cheapest Insertion as Initial Tour)	Day 1 Tour :	Depot - 4 - 7 - 9 - 6 - 5 - 3 - 2 - 10 - Depot
	Day 2 Tour :	Depot - 4 - 2 - 5 - 3 - 8 - 11 - Depot

Tour Improvement : (Inside/Outside as Initial Tour)	Day 1 Tour :	Depot - 4 - 7 - 9 - 6 - 2 - 5 - 3 - 8 - 11 - Depot
	Day 2 Tour :	Depot - 4 - 2 - 5 - 3 - 10 - 1

Tour Improvement : (Random Initial Tour)	Day 1 Tour :	Depot - 4 - 7 - 9 - 6 - 5 - 3 - 2 - 10 - Depot
	Day 2 Tour :	Depot - 4 - 2 - 5 - 3 - 8 - 11 - Depot

Appendix 12

Heuristic Solutions for the 21 Node Problem

Cheapest Insertion	Day 1 Tour :	Depot - 3 - 14 - 8 - 20 - 5 - 6 - 2 - 4 - 7 - 10 - 9 - Depot
	Day 2 Tour :	Depot - 11 - 12 - 3 - 13 - 8 - 19 - 18 - 15 - 5 - 16 - 6 - 21 - 2 - 4 - 7 - 17 - 9 - Depot
Inside/Outside	Day 1 Tour :	Depot - 9 -7 - 2 - 6 - 5 - 15 - 19 - 8 - 13 - 3 - 14 - 18 - 4 - Depot
	Day 2 Tour :	Depot - 11 - 12 - 3 - 20 - 8 - 5 - 16 - 6 - 2 - 21 - 10 - 9 - 17 - 7 - 4 - Depot
Tour Improvement : (Cheapest Insertion as Initial Tour)	Day 1 Tour :	Depot - 3 - 14 - 8 - 20 - 5 - 6 - 2 - 4 - 7 - 21 - 10 - 9 - Depot
	Day 2 Tour :	Depot - 11 - 12 - 3 - 13 - 8 - 19 - 18 - 15 - 5 - 16 - 6 - 2 - 4 - 7 - 17 - 9 - Depot
Tour Improvement : (Inside/Outside as Initial Tour)	Day 1 Tour :	Depot - 9 - 2 - 6 - 5 - 15 - 19 - 8 - 13 - 3 - 14 - 18 - 4 - 7 Depot
	Day 2 Tour :	Depot - 11 - 12 - 3 - 20 - 8 - 5 - 16 - 6 - 2 - 21 - 10 - 9 - 17 - 7 - 4 -Depot
Tour Improvement : (Random Initial Tour)	Day 1 Tour :	Depot - 3 - 14 - 13 - 8 - 15 - 5 - 16 - 6 - 2 - 4 - 7 - - 17 - 9 - Depot
	Day 2 Tour :	Depot - 9 - 10 - 21 - 6 - 5 - 2 - 7 - 4 - 18 - 19 - 8 - 20 - 3 - 12 - 11 -Depot

Appendix 13

Heuristic Solutions for the 42 Node Problem

Cheapest Insertion	Day 1 Tour :	Depot - 10 - 39 - 38 - 12 - 9 - 11 - 29 - 7 - 30 - 32 - 37 - 31 - 8 - 5 - 6 - 22 - 19 - 2 - 13 - 42 - 3 - 18 - 4 - Depot
	Day 2 Tour :	Depot - 21 - 20 - 34 - 10 - 35 - 12 - 9 - 11 - 36 - 33 - 7 - 28 - 8 - 5 - 6 - 27 - 26 - 25 - 24 - 23 - 2 - 15 - 14 - 41 - 40 - 13 - 16 - 3 - 17 - 4 - Depot

Inside/Outside	Day 1 Tour :	1 - 10 - 34 - 35 - 12 - 9 - 11 - 33 - 28 - 8 - 5 - 6 - 22 - 7 - 29 - 19 - 2 - 13 - 16 - 3 - 17 - 4 - 20 - 21 - Depot
	Day 2 Tour :	Depot - 4 - 7 - 30 - 32 - 9 - 10 - 39 - 38 - 12 - 11 - 36 - 37 - 31 - 8 - 5 - 25 - 24 - 6 - 27 - 26 - 23 - 2 - 13 - 15 - 14 - 41 - 40 - 42 - 3 - 18 - Depot

Tour Improvement : (Cheapest Insertion as Initial Tour)	Day 1 Tour :	Depot - 10 - 39 - 38 - 12 - 36 - 11 - 9 - 29 - 7 - 32 - 37 - 31 - 8 - 5 - 6 - 22 - 19 - 2 - 13 - 42 - 3 - 18 - 4 - Depot
	Day 2 Tour :	Depot - 21 - 20 - 34 - 10 - 35 - 12 - 9 - 11 - 33 - 7 - 30 - 28 - 8 - 6 - 27 - 26 - 5 - 25 - 24 - 23 - 2 - 15 - 14 - 41 - 40 - 13 - 16 - 3 - 17 - 4 - Depot

Tour Improvement : (Inside/Outside as Initial Tour)	Day 1 Tour :	Depot - 10 - 35 - 12 - 9 - 11 - 33 - 7 - 30 - 28 - 8 - 5 - 6 - 22 - 2 - 13 - 16 - 3 - 17 - 4 - 20 - 21 - Depot
	Day 2 Tour :	Depot - 4 - 19 - 7 - 29 - 34 - 10 - 39 - 38 - 12 - 9 - 11 - 36 - 32 - 37 - 31 - 8 - 5 - 25 - 24 - 26 - 6 - 27 - 23 - 2 - 13 - 15 - 14 - 41 - 40 - 42 - 3 - 18 - Depot

Tour Improvement : (Random Initial Tour)	Day 1 Tour :	Depot - 10 - 39 - 38 - 12 - 11 - 9 - 29 - 7 - 30 - 28 - 8 - 6 - 5 - 25 - 24 - 14 - 41 - 40 - 42 - 3 - 16 - 13 - 2 - 4 - Depot
	Day 2 Tour :	Depot - 21 - 10 - 9 - 11 - 33 - 7 - 6 - 27 - 26 - 5 - 8 - 31 - 37 - 32 - 36 - 12 - 35 - 34 - 20 - 4 - 17 - 18 - 3 - 13 - 15 - 23 - 22 - 2 - 19 - Depot

Appendix 14

Heuristic Solutions for the 100 Node Problem

Cheapest Insertion	Day 1 Tour :	Depot - 39 - 14 - 5 - 7 - 16 - 19 - 17 - 59 - 86 - 41 - 11 - 8 - 27 - 38 - 85 - 21 - 24 - 4 - 12 - 34 - 30 - 68 - 3 - 20 - 35 - 6 - 95 - 80 - 60 - 61 - 26 - 15 - 29 - 25 - 40 - 93 - 72 - 54 - 74 - 23 - 22 - 9 - 18 - 36 - 100 - 10 - 28 - 69 - 37 - 32 - 33 - 2 - 13 - 76 - 31 - 56 - 97 - 89 - 64 - Depot
	Day 2 Tour :	Depot - 39 - 14 - 99 - 5 - 91 - 55 - 77 - 7 - 42 - 98 - 16 - 19 - 17 - 96 - 11 - 58 - 8 - 27 - 38 - 49 - 21 - 24 - 57 - 70 - 4 - 12 - 63 - 43 - 34 - 30 - 66 - 3 - 78 - 53 - 20 - 82 - 52 - 35 - 84 - 6 - 50 - 26 - 83 - 15 - 29 - 79 - 48 - 25 - 73 - 62 - 92 - 40 - 88 - 23 - 22 - 65 - 9 - 75 - 90 - 94 - 18 - 36 - 87 - 10 - 28 - 44 - 37 - 81 - 32 - 33 - 2 - 13 - 67 - 31 - 47 - 51 - 71 - 46 - 45 - Depot

Inside/Outside	Day 1 Tour :	Depot - 64 - 39 - 14 - 99 - 5 - 7 - 16 - 19 - 17 - 59 - 86 - 41 - 11 - 58 - 8 - 27 - 38 - 3 - 85 - 21 - 24 - 57 - 4 - 70 - 68 - 66 - 30 - 12 - 34 - 20 - 82 - 52 - 35 - 6 - 26 - 47 - 51 - 71 - 31 - 46 - 2 - 32 - 33 - 74 - 72 - 93 - 13 - 76 - 15 - 29 - 25 - 73 - 62 - 92 - 40 - 54 - 23 - 22 - 65 - 9 - 18 - 36 - 87 - 10 - 28 - 81 - 37 - Depot
	Day 2 Tour :	Depot - 37 - 32 - 33 - 2 - 40 - 88 - 23 - 22 - 9 - 75 - 90 - 94 - 18 - 36 - 100 - 10 - 28 - 44 - 69 - 91 - 55 - 5 - 77 - 7 - 42 - 98 - 16 - 19 - 17 - 96 - 39 - 14 - 11 - 8 - 27 - 38 - 49 - 21 - 4 - 24 - 3 - 78 - 53 - 20 - 34 - 30 - 12 - 43 - 63 - 84 - 35 - 6 - 50 - 95 - 80 - 60 - 61 - 79 - 48 - 25 - 29 - 15 - 83 - 26 - 31 - 67 - 13 - 56 - 97 - 89 - 45 - Depot

Tour Improvement : (Cheapest Insertion as Initial Tour)	Day 1 Tour :	Depot - 64 - 39 - 14 - 5 - 7 - 16 - 19 - 17 - 96 - 59 - 11 - 58 - 3 - 8 - 27 - 38 - 85 - 21 - 24 - 4 - 70 - 68 - 30 - 34 - 20 - 12 - 43 - 63 - 84 - 35 - 6 - 95 - 80 - 60 - 61 - 29 - 25 - 40 - 93 - 72 - 54 - 74 - 23 - 22 - 9 - 18 - 36 - 100 - 28 - 10 - 37 - 32 - 33 - 2 - 13 - 76 - 15 - 26 - 47 - 31 - 56 - 97 - 89 - Depot
	Day 2 Tour :	Depot - 39 - 14 - 41 - 11 - 86 - 99 - 5 - 69 - 91 - 55 - 77 - 7 - 42 - 98 - 16 - 19 - 17 - 8 - 27 - 38 - 49 - 21 - 24 - 57 - 4 - 12 - 34 - 30 - 66 - 3 - 78 - 53 - 20 - 82 - 52 - 35 - 6 - 50 - 26 - 83 - 15 - 29 - 79 - 48 - 25 - 73 - 62 - 92 - 40 - 88 - 23 - 22 - 65 - 9 - 75 - 90 - 94 - 18 - 36 - 87 - 10 - 28 - 44 - 37 - 81 - 32 - 33 - 2 - 13 - 67 - 31 - 51 - 71 - 46 - 45 - Depot

Tour Improvement : (Inside/Outside as Initial Tour)	Day 1 Tour :	Depot - 64 - 39 - 14 - 99 - 5 - 7 - 16 - 19 - 17 - 96 - 11 - 58 - 3 - 8 - 27 - 38 - 85 - 21 - 24 - 57 - 4 - 70 - 68 - 66 - 30 - 12 - 34 - 20 - 82 - 52 - 35 - 6 - 50 - 26 - 51 - 71 - 31 - 46 - 2 - 33 - 74 - 54 - 72 - 93 - 13 - 76 - 15 - 29 - 25 - 73 - 62 - 92 - 40 - 23 - 22 - 65 - 9 - 18 - 36 - 87 - 10 - 28 - 81 - 32 - 37 - Depot
	Day 2 Tour :	Depot - 37 - 32 - 33 - 2 - 13 - 40 - 88 - 23 - 22 - 9 - 75 - 90 - 94 - 18 - 36 - 100 - 10 - 28 - 44 - 69 - 91 - 55 - 5 - 77 - 7 - 42 - 98 - 16 - 19 - 17 - 59 - 86 - 39 - 14 - 41 - 11 - 8 - 27 - 38 - 49 - 21 - 4 - 24 - 3 - 78 - 53 - 20 - 34 - 30 - 12 - 43 - 63 - 84 - 35 - 6 - 95 - 80 - 60 - 61 - 79 - 48 - 25 - 29 - 15 - 83 - 26 - 47 - 31 - 67 - 56 - 97 - 89 - 45 - Depot

Tour Improvement : (Random Initial Tour)	Day 1 Tour :	Depot - 37 - 36 - 18 - 94 - 90 - 75 - 9 - 22 - 23 - 88 - 40 - 25 - 29 - 15 - 50 - 6 - 35 - 84 - 63 - 43 - 12 - 4 - 70 - 24 - 21 - 38 - 27 - 8 - 17 - 19 - 16 - 7 - 5 - 55 - 91 - 69 - 28 - 10 - 44 - 39 - 14 - 41 - 11 - 58 - 3 - 66 - 30 - 34 - 20 - 26 - 47 - 31 - 67 - 13 - 2 - 33 - 32 - 89 - Depot
	Day 2 Tour :	Depot - 46 - 31 - 61 - 60 - 80 - 95 - 6 - 35 - 52 - 20 - 11 - 59 - 86 - 99 - 5 - 77 - 7 - 42 - 98 - 16 - 19 - 17 - 96 - 8 - 27 - 38 - 49 - 21 - 85 - 3 - 78 - 53 - 51 - 71 - 56 - 97 - 2 - 65 - 9 - 22 - 23 - 54 - 72 - 93 - 76 - 13 - 26 - 12 - 4 - 24 - 57 - 68 - 30 - 34 - 82 - 83 - 15 - 29 - 79 - 48 - 25 - 73 - 62 - 92 - 40 - 74 - 33 - 32 - 81 - 87 - 100 - 28 - 10 - 18 - 36 - 37 - 45 - 64 - 14 - 39 - Depot

Appendix 15

Heuristic Solutions for the 200 Node Problem

Cheapest Insertion	Day 1 Tour :	Depot - 164 - 191 - 124 - 184 - 125 - 173 - 35 - 71 - 114 - 200 - 18 - 25 - 30 - 147 - 21 - 94 - 132 - 138 - 166 - 109 - 118 - 57 - 93 - 134 - 65 - 50 - 47 - 130 - 76 - 160 - 9 - 112 - 189 - 43 - 197 - 56 - 49 - 168 - 145 - 85 - 61 - 106 - 7 - 11 - 111 - 92 - 183 - 8 - 2 - 36 - 143 - 37 - 162 - 14 - 139 - 66 - 178 - 156 - 104 - 75 - 23 - 153 - 182 - 12 - 186 - 98 - 5 - 97 - 83 - 185 - 90 - 155 - 52 - 175 - 126 - 33 - 82 - 165 - 167 - 68 - 102 - 20 - 89 - 96 - 192 - 154 - 195 - 122 - 188 - 131 - 121 - 120 - 31 - 88 - 105 - 103 - 81 - 72 - 170 - 15 - 172 - 74 - 150 - 26 - 16 - 133 - 55 - 149 - 163 - 27 - 73 - 176 - 58 - 107 - 80 - 142 - 51 - 48 - 199 - 40 - 84 - 115 - 77 - 100 - 62 - 87 - 148 - 196 - 4 - 194 - 181 - 17 - 171 - 157 - 24 - 158 - 29 - 46 - 174 - 101 - 13 - 137 - 78 - 28 - 135 - 161 - 190 - 44 - 136 - 42 - 86 - 67 - 38 - 187 - 117 - 110 - 123 - 159 - 69 - 53 - 60 - 10 - 108 - 79 - 180 - 129 - 64 - 127 - 41 - 32 - 91 - 144 - 3 - 119 - 39 - 45 - 6 - 63 - 59 - 179 - 99 - 54 - 70 - 19 - 22 - 34 - 169 - Depot
	Day 2 Tour :	Depot - 35 - 71 - 18 - 25 - 30 - 21 - 57 - 65 - 50 - 47 - 76 - 9 - 151 - 95 - 43 - 56 - 49 - 85 - 61 - 7 - 11 - 8 - 2 - 36 - 37 - 14 - 66 - 141 - 198 - 75 - 23 - 12 - 128 - 177 - 116 - 5 - 83 - 90 - 52 - 33 - 82 - 68 - 20 - 89 - 140 - 31 - 88 - 81 - 72 - 15 - 74 - 26 - 16 - 55 - 152 - 113 - 27 - 73 - 58 - 80 - 51 - 48 - 40 - 84 - 77 - 62 - 87 - 4 - 17 - 146 - 24 - 29 - 46 - 13 - 78 - 28 - 44 - 42 - 86 - 67 - 38 - 69 - 53 - 60 - 10 - 79 - 64 - 41 - 32 - 3 - 39 - 45 - 6 - 63 - 59 - 54 - 70 - 19 - 22 - 34 - 193 - Depot

200 Node Problem

Inside/Outside	Day 1 Tour :	Depot - 164 - 173 - 125 - 184 - 124 - 191 - 37 - 36 - 2 - 8 - 183 - 92 - 111 - 11 - 7 - 61 - 85 - 49 - 56 - 197 - 43 - 95 - 151 - 130 - 47 - 76 - 9 - 50 - 65 - 134 - 93 - 57 - 4 - 62 - 77 - 84 - 40 - 199 - 87 - 148 - 17 - 48 - 146 - 51 - 80 - 107 - 58 - 176 - 73 - 27 - 163 - 149 - 55 - 133 - 16 - 15 - 172 - 72 - 81 - 103 - 105 - 88 - 31 - 120 - 121 - 131 - 192 - 96 - 89 - 20 - 102 - 68 - 82 - 127 - 41 - 129 - 154 - 195 - 122 - 188 - 135 - 161 - 190 - 44 - 150 - 74 - 26 - 28 - 78 - 13 - 158 - 24 - 29 - 46 - 42 - 86 - 69 - 159 - 123 - 110 - 180 - 64 - 32 - 91 - 126 - 33 - 175 - 52 - 90 - 83 - 5 - 116 - 177 - 128 - 12 - 23 - 75 - 198 - 141 - 66 - 139 - 14 - 162 - 193 - 34 - 22 - 19 - 70 - 54 - 59 - 63 - 6 - 45 - 39 - 144 - 3 - 79 - 10 - 60 - 53 - 67 - 38 - 30 - 21 - 147 - 18 - 25 - 71 - 114 - 200 - 35 - Depot
	Day 2 Tour :	Depot - 35 - 71 - 18 - 25 - 38 - 67 - 187 - 117 - 60 - 10 - 53 - 69 - 86 - 42 - 136 - 44 - 26 - 28 - 78 - 137 - 13 - 101 - 29 - 46 - 174 - 94 - 30 - 21 - 65 - 50 - 47 - 76 - 160 - 9 - 112 - 57 - 118 - 109 - 166 - 138 - 132 - 194 - 4 - 196 - 181 - 87 - 62 - 100 - 77 - 115 - 84 - 40 - 48 - 17 - 171 - 157 - 24 - 51 - 142 - 80 - 58 - 73 - 27 - 113 - 55 - 16 - 152 - 15 - 170 - 72 - 81 - 74 - 88 - 31 - 140 - 89 - 20 - 68 - 167 - 165 - 82 - 33 - 52 - 155 - 90 - 185 - 83 - 97 - 5 - 98 - 186 - 153 - 182 - 75 - 23 - 12 - 59 - 63 - 45 - 32 - 41 - 64 - 79 - 119 - 3 - 39 - 6 - 179 - 99 - 54 - 108 - 169 - 34 - 22 - 70 - 19 - 178 - 104 - 156 - 66 - 143 - 61 - 85 - 145 - 168 - 49 - 56 - 189 - 43 - 11 - 7 - 106 - 8 - 2 - 36 - 37 - 14 - Depot

200 Node Problem

Tour Improvement : (Cheapest Insertion as Initial Tour)	Day 1 Tour :	Depot - 164 - 191 - 124 - 184 - 125 - 173 - 35 - 71 - 114 - 200 - 18 - 147 - 21 - 30 - 94 - 132 - 138 - 109 - 118 - 57 - 93 - 134 - 65 - 50 - 47 - 76 - 9 - 112 - 189 - 43 - 197 - 56 - 49 - 168 - 145 - 85 - 61 - 106 - 7 - 11 - 111 - 92 - 183 - 8 - 2 - 36 - 143 - 37 - 162 - 14 - 139 - 66 - 178 - 156 - 104 - 75 - 23 - 153 - 182 - 12 - 186 - 98 - 5 - 97 - 83 - 185 - 90 - 155 - 52 - 175 - 126 - 33 - 82 - 165 - 167 - 68 - 102 - 20 - 89 - 96 - 154 - 195 - 122 - 192 - 131 - 188 - 31 - 88 - 105 - 74 - 103 - 81 - 72 - 170 - 15 - 16 - 133 - 55 - 149 - 163 - 27 - 73 - 176 - 58 - 107 - 80 - 142 - 51 - 48 - 148 - 87 - 199 - 40 - 84 - 115 - 77 - 100 - 62 - 196 - 4 - 194 - 181 - 17 - 171 - 157 - 24 - 158 - 29 - 46 - 174 - 101 - 13 - 137 - 78 - 28 - 26 - 135 - 161 - 190 - 44 - 136 - 42 - 86 - 67 - 38 - 25 - 187 - 117 - 110 - 123 - 159 - 69 - 53 - 60 - 10 - 108 - 79 - 180 - 129 - 127 - 41 - 64 - 32 - 144 - 3 - 39 - 45 - 6 - 63 - 59 - 179 - 99 - 54 - 70 - 19 - 22 - 34 - 169 - Depot
	Day 2 Tour :	Depot - 35 - 71 - 18 - 25 - 30 - 21 - 166 - 57 - 65 - 50 - 76 - 9 - 160 - 47 - 130 - 151 - 95 - 43 - 56 - 49 - 85 - 61 - 7 - 11 - 8 - 2 - 36 - 37 - 14 - 66 - 141 - 198 - 75 - 23 - 12 - 128 - 177 - 116 - 5 - 83 - 90 - 52 - 33 - 82 - 68 - 20 - 89 - 121 - 120 - 140 - 31 - 88 - 81 - 72 - 15 - 172 - 74 - 150 - 26 - 16 - 55 - 152 - 113 - 27 - 73 - 58 - 80 - 51 - 48 - 87 - 40 - 84 - 77 - 62 - 4 - 17 - 146 - 24 - 29 - 46 - 13 - 78 - 28 - 44 - 42 - 69 - 86 - 67 - 38 - 53 - 60 - 10 - 79 - 119 - 3 - 64 - 41 - 32 - 91 - 39 - 45 - 6 - 63 - 59 - 54 - 70 - 19 - 22 - 34 - 193 - Depot

200 Node Problem

Tour Improvement : (Inside/Outside as Initial Tour)	Day 1 Tour :	Depot - 164 - 173 - 125 - 184 - 124 - 191 - 37 - 36 - 2 - 8 - 183 - 92 - 111 - 11 - 7 - 106 - 61 - 85 - 49 - 56 - 197 - 43 - 189 - 95 - 151 - 130 - 47 - 76 - 9 - 160 - 50 - 65 - 93 - 57 - 166 - 4 - 196 - 62 - 77 - 84 - 40 - 199 - 87 - 148 - 181 - 17 - 48 - 51 - 80 - 107 - 58 - 176 - 73 - 27 - 163 - 149 - 55 - 133 - 16 - 15 - 172 - 72 - 81 - 88 - 31 - 120 - 131 - 192 - 96 - 89 - 20 - 102 - 68 - 82 - 127 - 41 - 129 - 154 - 195 - 122 - 188 - 135 - 161 - 190 - 44 - 150 - 74 - 26 - 28 - 78 - 13 - 158 - 24 - 29 - 46 - 101 - 42 - 86 - 69 - 159 - 123 - 110 - 180 - 64 - 32 - 91 - 126 - 33 - 175 - 52 - 90 - 83 - 5 - 116 - 177 - 128 - 12 - 23 - 75 - 198 - 141 - 66 - 14 - 139 - 193 - 34 - 22 - 19 - 70 - 54 - 59 - 63 - 6 - 45 - 39 - 144 - 3 - 79 - 10 - 60 - 53 - 67 - 38 - 25 - 30 - 21 - 147 - 18 - 71 - 114 - 200 - 35 - Depot
	Day 2 Tour :	Depot - 35 - 71 - 18 - 25 - 38 - 67 - 187 - 117 - 60 - 10 - 53 - 69 - 86 - 42 - 136 - 44 - 26 - 28 - 78 - 137 - 13 - 24 - 29 - 46 - 174 - 94 - 30 - 21 - 134 - 65 - 50 - 47 - 76 - 9 - 112 - 57 - 118 - 109 - 138 - 132 - 194 - 4 - 62 - 100 - 77 - 115 - 84 - 40 - 87 - 48 - 17 - 171 - 157 - 146 - 51 - 142 - 80 - 58 - 73 - 27 - 113 - 152 - 55 - 16 - 15 - 170 - 72 - 81 - 103 - 74 - 105 - 88 - 31 - 140 - 121 - 89 - 20 - 68 - 167 - 165 - 82 - 33 - 52 - 155 - 90 - 185 - 83 - 97 - 5 - 98 - 186 - 153 - 182 - 75 - 23 - 12 - 54 - 59 - 63 - 6 - 45 - 39 - 32 - 41 - 64 - 79 - 3 - 119 - 179 - 99 - 108 - 169 - 34 - 22 - 70 - 19 - 178 - 104 - 156 - 66 - 143 - 61 - 85 - 145 - 168 - 49 - 56 - 43 - 11 - 7 - 8 - 2 - 36 - 37 - 162 - 14 - Depot

200 Node Problem

Tour Improvement : (Random Initial Tour)	Day 1 Tour :	Depot - 35 - 71 - 187 - 86 - 42 - 13 - 78 - 28 - 26 - 74 - 103 - 81 - 72 - 170 - 15 - 16 - 55 - 149 - 163 - 27 - 73 - 58 - 107 - 146 - 48 - 87 - 199 - 40 - 84 - 115 - 77 - 62 - 166 - 93 - 47 - 130 - 95 - 151 - 92 - 183 - 8 - 61 - 85 - 7 - 11 - 2 - 36 - 37 - 14 - 66 - 198 - 75 - 23 - 12 - 186 - 98 - 5 - 97 - 185 - 90 - 126 - 91 - 39 - 3 - 53 - 174 - 46 - 29 - 24 - 80 - 142 - 51 - 17 - 181 - 4 - 57 - 112 - 56 - 49 - 197 - 43 - 189 - 9 - 76 - 50 - 65 - 21 - 30 - 147 - 18 - 25 - 38 - 67 - 69 - 44 - 190 - 161 - 88 - 31 - 89 - 20 - 102 - 68 - 82 - 45 - 6 - 179 - 99 - 70 - 19 - 34 - 22 - 54 - 59 - 63 - 116 - 83 - 52 - 33 - 127 - 41 - 32 - 64 - 79 - 10 - 60 - Depot
	Day 2 Tour :	Depot - 164 - 35 - 173 - 124 - 191 - 111 - 11 - 7 - 43 - 9 - 160 - 76 - 47 - 50 - 65 - 134 - 21 - 137 - 78 - 16 - 15 - 72 - 172 - 81 - 140 - 121 - 96 - 129 - 64 - 144 - 39 - 59 - 128 - 153 - 182 - 108 - 10 - 60 - 123 - 159 - 136 - 44 - 28 - 26 - 133 - 55 - 152 - 113 - 27 - 73 - 176 - 58 - 80 - 51 - 48 - 40 - 84 - 87 - 148 - 17 - 171 - 157 - 29 - 101 - 86 - 53 - 110 - 180 - 82 - 167 - 165 - 68 - 20 - 89 - 192 - 188 - 135 - 150 - 74 - 105 - 88 - 31 - 120 - 131 - 122 - 195 - 154 - 119 - 178 - 156 - 104 - 75 - 23 - 12 - 6 - 45 - 90 - 155 - 52 - 175 - 33 - 3 - 79 - 69 - 42 - 13 - 158 - 24 - 46 - 94 - 18 - 184 - 125 - 200 - 114 - 71 - 117 - 67 - 38 - 25 - 30 - 132 - 138 - 194 - 4 - 196 - 62 - 77 - 100 - 109 - 118 - 57 - 56 - 49 - 168 - 145 - 85 - 61 - 106 - 8 - 2 - 36 - 143 - 37 - 162 - 14 - 139 - 193 - 169 - 34 - 22 - 32 - 41 - 83 - 5 - 177 - 63 - 54 - 70 - 19 - 141 - 66 - Depot

Appendix 16

IP Formulation of Every Third Day Milk Collection.

Minimise $Z = 41X_{01,02,1} + 58X_{01,03,1} + 14X_{01,04,1} + 54X_{01,05,1} + 73X_{01,06,1} +$

$40X_{01,07,1} + 45X_{01,08,1} + 50X_{01,09,1} + 10X_{01,10,1} + 40X_{01,11,1} + 22X_{02,03,1} +$

$36X_{02,04,1} + 14X_{02,05,1} + 42X_{02,06,1} + 64X_{02,07,1} + 36X_{02,08,1} + 51X_{02,09,1} +$

$32X_{02,10,1} + 50X_{02,11,1} + 57X_{03,04,1} + 10X_{03,05,1} + 54X_{03,06,1} + 86X_{03,07,1} +$

$32X_{03,08,1} + 73X_{03,09,1} + 50X_{03,10,1} + 51X_{03,11,1} + 50X_{04,05,1} + 61X_{04,06,1} +$

$32X_{04,07,1} + 51X_{04,08,1} + 36X_{04,09,1} + 10X_{04,10,1} + 51X_{04,11,1} + 45X_{05,06,1} +$

$78X_{05,07,1} + 36X_{05,08,1} + 63X_{05,09,1} + 45X_{05,10,1} + 54X_{05,11,1} + 73X_{06,07,1} +$

$78X_{06,08,1} + 45X_{06,09,1} + 63X_{06,10,1} + 92X_{06,11,1} + 82X_{07,08,1} + 30X_{07,09,1} +$

$41X_{07,10,1} + 80X_{07,11,1} + 81X_{08,09,1} + 41X_{08,10,1} + 20X_{08,11,1} + 45X_{09,10,1} +$

$85X_{09,11,1} + 41X_{10,11,1} +$

$41X_{01,02,2} + 58X_{01,03,2} + 14X_{01,04,2} + 54X_{01,05,2} + 73X_{01,06,2} + 40X_{01,07,2} +$

$45X_{01,08,2} + 50X_{01,09,2} + 10X_{01,10,2} + 40X_{01,11,2} + 22X_{02,03,2} + 36X_{02,04,2} +$

$14X_{02,05,2} + 42X_{02,06,2} + 64X_{02,07,2} + 36X_{02,08,2} + 51X_{02,09,2} + 32X_{02,10,2} +$

$50X_{02,11,2} + 57X_{03,04,2} + 10X_{03,05,2} + 54X_{03,06,2} + 86X_{03,07,2} + 32X_{03,08,2} +$

$73X_{03,09,2} + 50X_{03,10,2} + 51X_{03,11,2} + 50X_{04,05,2} + 61X_{04,06,2} + 32X_{04,07,2} +$

$51X_{04,08,2} + 36X_{04,09,2} + 10X_{04,10,2} + 51X_{04,11,2} + 45X_{05,06,2} + 78X_{05,07,2} +$

$36X_{05,08,2} + 63X_{05,09,2} + 45X_{05,10,2} + 54X_{05,11,2} + 73X_{06,07,2} + 78X_{06,08,2} +$

$45X_{06,09,2} + 63X_{06,10,2} + 92X_{06,11,2} + 82X_{07,08,2} + 30X_{07,09,2} + 41X_{07,10,2} +$

$80X_{07,11,2} + 81X_{08,09,2} + 41X_{08,10,2} + 20X_{08,11,2} + 45X_{09,10,2} + 85X_{09,11,2} +$

$41X_{10,11,2} +$

$41X_{01,02,3} + 58X_{01,03,3} + 14X_{01,04,3} + 54X_{01,05,3} + 73X_{01,06,3} + 40X_{01,07,3} +$

$45X_{01,08,3} + 50X_{01,09,3} + 10X_{01,10,3} + 40X_{01,11,3} + 22X_{02,03,3} + 36X_{02,04,3} +$

$14X_{02,05,3} + 42X_{02,06,3} + 64X_{02,07,3} + 36X_{02,08,3} + 51X_{02,09,3} + 32X_{02,10,3} +$

$50X_{02,11,3} + 57X_{03,04,3} + 10X_{03,05,3} + 54X_{03,06,3} + 86X_{03,07,3} + 32X_{03,08,3} +$

$73X_{03,09,3} + 50X_{03,10,3} + 51X_{03,11,3} + 50X_{04,05,3} + 61X_{04,06,3} + 32X_{04,07,3} +$

$51X_{04,08,3} + 36X_{04,09,3} + 10X_{04,10,3} + 51X_{04,11,3} + 45X_{05,06,3} + 78X_{05,07,3} +$

$36X_{05,08,3} + 63X_{05,09,3} + 45X_{05,10,3} + 54X_{05,11,3} + 73X_{06,07,3} + 78X_{06,08,3} +$

$45X_{06,09,3} + 63X_{06,10,3} + 92X_{06,11,3} + 82X_{07,08,3} + 30X_{07,09,3} + 41X_{07,10,3} +$

$80X_{07,11,3} + 81X_{08,09,3} + 41X_{08,10,3} + 20X_{08,11,3} + 45X_{09,10,3} + 85X_{09,11,3} +$

$41X_{10,11,3} +$

$41X_{01,02,4} + 58X_{01,03,4} + 14X_{01,04,4} + 54X_{01,05,4} + 73X_{01,06,4} + 40X_{01,07,4} +$

$45X_{01,08,4} + 50X_{01,09,4} + 10X_{01,10,4} + 40X_{01,11,4} + 22X_{02,03,4} + 36X_{02,04,4} +$

$14X_{02,05,4} + 42X_{02,06,4} + 64X_{02,07,4} + 36X_{02,08,4} + 51X_{02,09,4} + 32X_{02,10,4} +$

$50X_{02,11,4} + 57X_{03,04,4} + 10X_{03,05,4} + 54X_{03,06,4} + 86X_{03,07,4} + 32X_{03,08,4} +$
$73X_{03,09,4} + 50X_{03,10,4} + 51X_{03,11,4} + 50X_{04,05,4} + 61X_{04,06,4} + 32X_{04,07,4} +$
$51X_{04,08,4} + 36X_{04,09,4} + 10X_{04,10,4} + 51X_{04,11,4} + 45X_{05,06,4} + 78X_{05,07,4} +$
$36X_{05,08,4} + 63X_{05,09,4} + 45X_{05,10,4} + 54X_{05,11,4} + 73X_{06,07,4} + 78X_{06,08,4} +$
$45X_{06,09,4} + 63X_{06,10,4} + 92X_{06,11,4} + 82X_{07,08,4} + 30X_{07,09,4} + 41X_{07,10,4} +$
$80X_{07,11,4} + 81X_{08,09,4} + 41X_{08,10,4} + 20X_{08,11,4} + 45X_{09,10,4} + 85X_{09,11,4} +$
$41X_{10,11,4} +$
$41X_{01,02,5} + 58X_{01,03,5} + 14X_{01,04,5} + 54X_{01,05,5} + 73X_{01,06,5} + 40X_{01,07,5} +$
$45X_{01,08,5} + 50X_{01,09,5} + 10X_{01,10,5} + 40X_{01,11,5} + 22X_{02,03,5} + 36X_{02,04,5} +$
$14X_{02,05,5} + 42X_{02,06,5} + 64X_{02,07,5} + 36X_{02,08,5} + 51X_{02,09,5} + 32X_{02,10,5} +$
$50X_{02,11,5} + 57X_{03,04,5} + 10X_{03,05,5} + 54X_{03,06,5} + 86X_{03,07,5} + 32X_{03,08,5} +$
$73X_{03,09,5} + 50X_{03,10,5} + 51X_{03,11,5} + 50X_{04,05,5} + 61X_{04,06,5} + 32X_{04,07,5} +$
$51X_{04,08,5} + 36X_{04,09,5} + 10X_{04,10,5} + 51X_{04,11,5} + 45X_{05,06,5} + 78X_{05,07,5} +$
$36X_{05,08,5} + 63X_{05,09,5} + 45X_{05,10,5} + 54X_{05,11,5} + 73X_{06,07,5} + 78X_{06,08,5} +$
$45X_{06,09,5} + 63X_{06,10,5} + 92X_{06,11,5} + 82X_{07,08,5} + 30X_{07,09,5} + 41X_{07,10,5} +$
$80X_{07,11,5} + 81X_{08,09,5} + 41X_{08,10,5} + 20X_{08,11,5} + 45X_{09,10,5} + 85X_{09,11,5} +$
$41X_{10,11,5} +$
$41X_{01,02,6} + 58X_{01,03,6} + 14X_{01,04,6} + 54X_{01,05,6} + 73X_{01,06,6} + 40X_{01,07,6} +$
$45X_{01,08,6} + 50X_{01,09,6} + 10X_{01,10,6} + 40X_{01,11,6} + 22X_{02,03,6} + 36X_{02,04,6} +$
$14X_{02,05,6} + 42X_{02,06,6} + 64X_{02,07,6} + 36X_{02,08,6} + 51X_{02,09,6} + 32X_{02,10,6} +$
$50X_{02,11,6} + 57X_{03,04,6} + 10X_{03,05,6} + 54X_{03,06,6} + 86X_{03,07,6} + 32X_{03,08,6} +$
$73X_{03,09,6} + 50X_{03,10,6} + 51X_{03,11,6} + 50X_{04,05,6} + 61X_{04,06,6} + 32X_{04,07,6} +$
$51X_{04,08,6} + 36X_{04,09,6} + 10X_{04,10,6} + 51X_{04,11,6} + 45X_{05,06,6} + 78X_{05,07,6} +$
$36X_{05,08,6} + 63X_{05,09,6} + 45X_{05,10,6} + 54X_{05,11,6} + 73X_{06,07,6} + 78X_{06,08,6} +$
$45X_{06,09,6} + 63X_{06,10,6} + 92X_{06,11,6} + 82X_{07,08,6} + 30X_{07,09,6} + 41X_{07,10,6} +$
$80X_{07,11,6} + 81X_{08,09,6} + 41X_{08,10,6} + 20X_{08,11,6} + 45X_{09,10,6} + 85X_{09,11,6} +$
$41X_{10,11,6}$

Subject to:

$X_{01,02,1} + X_{01,03,1} + X_{01,04,1} + X_{01,05,1} + X_{01,06,1} + X_{01,07,1} + X_{01,08,1} + X_{01,09,1} + X_{01,10,1} + X_{01,11,1} = 2$
$X_{01,02,1} + X_{02,03,1} + X_{02,04,1} + X_{02,05,1} + X_{02,06,1} + X_{02,07,1} + X_{02,08,1} + X_{02,09,1} + X_{02,10,1} + X_{02,11,1} = 2$
$X_{01.03,1} + X_{02,03,1} + X_{03,04,1} + X_{03,05,1} + X_{03,06,1} + X_{03,07,1} + X_{03,08,1} + X_{03,09,1} + X_{03,10,1} + X_{03,11,1} = 2$
$X_{01,04,1} + X_{02,04,1} + X_{03,04,1} + X_{04,05,1} + X_{04,06,1} + X_{04,07,1} + X_{04,08,1} + X_{04,09,1} + X_{04,10,1} + X_{04,11,1} = 2$
$X_{01,05,1} + X_{02,05,1} + X_{03,05,1} + X_{04,05,1} + X_{05,06,1} + X_{05,07,1} + X_{05,08,1} + X_{05,09,1} + X_{05,10,1} + X_{05,11,1} = 2$

$X_{01,02,2} + X_{01,03,2} + X_{01,04,2} + X_{01,05,2} + X_{01,06,2} + X_{01,07,2} + X_{01,08,2} + X_{01,09,2} + X_{01,10,2} + X_{01,11,2} = 2$
$X_{01.02,2} + X_{02,03,2} + X_{02,04,2} + X_{02,05,2} + X_{02,06,2} + X_{02,07,2} + X_{02,08,2} + X_{02,09,2} + X_{02,10,2} + X_{02,11,2} = 2$

$$X_{01,03,2} + X_{02,03,2} + X_{03,04,2} + X_{03,05,2} + X_{03,06,2} + X_{03,07,2} + X_{03,08,2} + X_{03,09,2} + X_{03,10,2} + X_{03,11,2} = 2$$
$$X_{01,04,2} + X_{02,04,2} + X_{03,04,2} + X_{04,05,2} + X_{04,06,2} + X_{04,07,2} + X_{04,08,2} + X_{04,09,2} + X_{04,10,2} + X_{04,11,2} = 2$$
$$X_{01,05,2} + X_{02,05,2} + X_{03,05,2} + X_{04,05,2} + X_{05,06,2} + X_{05,07,2} + X_{05,08,2} + X_{05,09,2} + X_{05,10,2} + X_{05,11,2} = 2$$

$$X_{01,02,3} + X_{01,03,3} + X_{01,04,3} + X_{01,05,3} + X_{01,06,3} + X_{01,07,3} + X_{01,08,3} + X_{01,09,3} + X_{01,10,3} + X_{01,11,3} = 2$$
$$X_{01,02,3} + X_{02,03,3} + X_{02,04,3} + X_{02,05,3} + X_{02,06,3} + X_{02,07,3} + X_{02,08,3} + X_{02,09,3} + X_{02,10,3} + X_{02,11,3} = 2$$
$$X_{01,03,3} + X_{02,03,3} + X_{03,04,3} + X_{03,05,3} + X_{03,06,3} + X_{03,07,3} + X_{03,08,3} + X_{03,09,3} + X_{03,10,3} + X_{03,11,3} = 2$$
$$X_{01,04,3} + X_{02,04,3} + X_{03,04,3} + X_{04,05,3} + X_{04,06,3} + X_{04,07,3} + X_{04,08,3} + X_{04,09,3} + X_{04,10,3} + X_{04,11,3} = 2$$
$$X_{01,05,3} + X_{02,05,3} + X_{03,05,3} + X_{04,05,3} + X_{05,06,3} + X_{05,07,3} + X_{05,08,3} + X_{05,09,3} + X_{05,10,3} + X_{05,11,3} = 2$$

$$X_{01,02,4} + X_{01,03,4} + X_{01,04,4} + X_{01,05,4} + X_{01,06,4} + X_{01,07,4} + X_{01,08,4} + X_{01,09,4} + X_{01,10,4} + X_{01,11,4} = 2$$
$$X_{01,02,4} + X_{02,03,4} + X_{02,04,4} + X_{02,05,4} + X_{02,06,4} + X_{02,07,4} + X_{02,08,4} + X_{02,09,4} + X_{02,10,4} + X_{02,11,4} = 2$$
$$X_{01,03,4} + X_{02,03,4} + X_{03,04,4} + X_{03,05,4} + X_{03,06,4} + X_{03,07,4} + X_{03,08,4} + X_{03,09,4} + X_{03,10,4} + X_{03,11,4} = 2$$
$$X_{01,04,4} + X_{02,04,4} + X_{03,04,4} + X_{04,05,4} + X_{04,06,4} + X_{04,07,4} + X_{04,08,4} + X_{04,09,4} + X_{04,10,4} + X_{04,11,4} = 2$$
$$X_{01,05,4} + X_{02,05,4} + X_{03,05,4} + X_{04,05,4} + X_{05,06,4} + X_{05,07,4} + X_{05,08,4} + X_{05,09,4} + X_{05,10,4} + X_{05,11,4} = 2$$

$$X_{01,02,5} + X_{01,03,5} + X_{01,04,5} + X_{01,05,5} + X_{01,06,5} + X_{01,07,5} + X_{01,08,5} + X_{01,09,5} + X_{01,10,5} + X_{01,11,5} = 2$$
$$X_{01,02,5} + X_{02,03,5} + X_{02,04,5} + X_{02,05,5} + X_{02,06,5} + X_{02,07,5} + X_{02,08,5} + X_{02,09,5} + X_{02,10,5} + X_{02,11,5} = 2$$
$$X_{01,03,5} + X_{02,03,5} + X_{03,04,5} + X_{03,05,5} + X_{03,06,5} + X_{03,07,5} + X_{03,08,5} + X_{03,09,5} + X_{03,10,5} + X_{03,11,5} = 2$$
$$X_{01,04,5} + X_{02,04,5} + X_{03,04,5} + X_{04,05,5} + X_{04,06,5} + X_{04,07,5} + X_{04,08,5} + X_{04,09,5} + X_{04,10,5} + X_{04,11,5} = 2$$
$$X_{01,05,5} + X_{02,05,5} + X_{03,05,5} + X_{04,05,5} + X_{05,06,5} + X_{05,07,5} + X_{05,08,5} + X_{05,09,5} + X_{05,10,5} + X_{05,11,5} = 2$$

$$X_{01,02,6} + X_{01,03,6} + X_{01,04,6} + X_{01,05,6} + X_{01,06,6} + X_{01,07,6} + X_{01,08,6} + X_{01,09,6} + X_{01,10,6} + X_{01,11,6} = 2$$
$$X_{01,02,6} + X_{02,03,6} + X_{02,04,6} + X_{02,05,6} + X_{02,06,6} + X_{02,07,6} + X_{02,08,6} + X_{02,09,6} + X_{02,10,6} + X_{02,11,6} = 2$$
$$X_{01,03,6} + X_{02,03,6} + X_{03,04,6} + X_{03,05,6} + X_{03,06,6} + X_{03,07,6} + X_{03,08,6} + X_{03,09,6} + X_{03,10,6} + X_{03,11,6} = 2$$
$$X_{01,04,6} + X_{02,04,6} + X_{03,04,6} + X_{04,05,6} + X_{04,06,6} + X_{04,07,6} + X_{04,08,6} + X_{04,09,6} + X_{04,10,6} + X_{04,11,6} = 2$$
$$X_{01,05,6} + X_{02,05,6} + X_{03,05,6} + X_{04,05,6} + X_{05,06,6} + X_{05,07,6} + X_{05,08,6} + X_{05,09,6} + X_{05,60,6} + X_{05,61,6} = 2$$

$$X_{01,06,1} + X_{02,06,1} + X_{03,06,1} + X_{04,06,1} + X_{05,06,1} + X_{06,07,1} + X_{06,08,1} + X_{06,09,1} + X_{06,10,1} + X_{06,11,1} - 2Y_{06,1} = 0$$
$$X_{01,07,1} + X_{02,07,1} + X_{03,07,1} + X_{04,07,1} + X_{05,07,1} + X_{06,07,1} + X_{07,08,1} + X_{07,09,1} + X_{07,10,1} + X_{07,11,1} - 2Y_{07,1} = 0$$
$$X_{01,08,1} + X_{02,08,1} + X_{03,08,1} + X_{04,08,1} + X_{05,08,1} + X_{06,08,1} + X_{07,08,1} + X_{08,09,1} + X_{08,10,1} + X_{08,11,1} - 2Y_{08,1} = 0$$
$$X_{01,09,1} + X_{02,09,1} + X_{03,09,1} + X_{04,09,1} + X_{05,09,1} + X_{06,09,1} + X_{07,09,1} + X_{08,09,1} + X_{09,10,1} + X_{09,11,1} - 2Y_{09,1} = 0$$
$$X_{01,10,1} + X_{02,10,1} + X_{03,10,1} + X_{04,10,1} + X_{05,10,1} + X_{06,10,1} + X_{07,10,1} + X_{08,10,1} + X_{09,10,1} + X_{10,11,1} - 2Y_{10,1} = 0$$
$$X_{01,11,1} + X_{02,11,1} + X_{03,11,1} + X_{04,11,1} + X_{05,11,1} + X_{06,11,1} + X_{07,11,1} + X_{08,11,1} + X_{09,11,1} + X_{10,11,1} - 2Y_{11,1} = 0$$

$$X_{01,06,2} + X_{02,06,2} + X_{03,06,2} + X_{04,06,2} + X_{05,06,2} + X_{06,07,2} + X_{06,08,2} + X_{06,09,2} + X_{06,10,2} + X_{06,11,2} - 2Y_{06,2} = 0$$
$$X_{01,07,2} + X_{02,07,2} + X_{03,07,2} + X_{04,07,2} + X_{05,07,2} + X_{06,07,2} + X_{07,08,2} + X_{07,09,2} + X_{07,10,2} + X_{07,11,2} - 2Y_{07,2} = 0$$
$$X_{01,08,2} + X_{02,08,2} + X_{03,08,2} + X_{04,08,2} + X_{05,08,2} + X_{06,08,2} + X_{07,08,2} + X_{08,09,2} + X_{08,10,2} + X_{08,11,2} - 2Y_{08,2} = 0$$
$$X_{01,09,2} + X_{02,09,2} + X_{03,09,2} + X_{04,09,2} + X_{05,09,2} + X_{06,09,2} + X_{07,09,2} + X_{08,09,2} + X_{09,10,2} + X_{09,11,2} - 2Y_{09,2} = 0$$
$$X_{01,10,2} + X_{02,10,2} + X_{03,10,2} + X_{04,10,2} + X_{05,10,2} + X_{06,10,2} + X_{07,10,2} + X_{08,10,2} + X_{09,10,2} + X_{10,11,2} - 2Y_{10,2} = 0$$

$X_{01,11,2} + X_{02,11,2} + X_{03,11,2} + X_{04,11,2} + X_{05,11,2} + X_{06,11,2} + X_{07,11,2} + X_{08,11,2} + X_{09,11,2} + X_{10,11,2} - 2Y_{11,2} = 0$

$X_{01,06,3} + X_{02,06,3} + X_{03,06,3} + X_{04,06,3} + X_{05,06,3} + X_{06,07,3} + X_{06,08,3} + X_{06,09,3} + X_{06,10,3} + X_{06,11,3} - 2Y_{06,3} = 0$

$X_{01,07,3} + X_{02,07,3} + X_{03,07,3} + X_{04,07,3} + X_{05,07,3} + X_{06,07,3} + X_{07,08,3} + X_{07,09,3} + X_{07,10,3} + X_{07,11,3} - 2Y_{07,3} = 0$

$X_{01,08,3} + X_{02,08,3} + X_{03,08,3} + X_{04,08,3} + X_{05,08,3} + X_{06,08,3} + X_{07,08,3} + X_{08,09,3} + X_{08,10,3} + X_{08,11,3} - 2Y_{08,3} = 0$

$X_{01,09,3} + X_{02,09,3} + X_{03,09,3} + X_{04,09,3} + X_{05,09,3} + X_{06,09,3} + X_{07,09,3} + X_{08,09,3} + X_{09,10,3} + X_{09,11,3} - 2Y_{09,3} = 0$

$X_{01,10,3} + X_{02,10,3} + X_{03,10,3} + X_{04,10,3} + X_{05,10,3} + X_{06,10,3} + X_{07,10,3} + X_{08,10,3} + X_{09,10,3} + X_{10,11,3} - 2Y_{10,3} = 0$

$X_{01,11,3} + X_{02,11,3} + X_{03,11,3} + X_{04,11,3} + X_{05,11,3} + X_{06,11,3} + X_{07,11,3} + X_{08,11,3} + X_{09,11,3} + X_{10,11,3} - 2Y_{11,3} = 0$

$X_{01,06,4} + X_{02,06,4} + X_{03,06,4} + X_{04,06,4} + X_{05,06,4} + X_{06,07,4} + X_{06,08,4} + X_{06,09,4} + X_{06,10,4} + X_{06,11,4} - 2Y_{06,4} = 0$

$X_{01,07,4} + X_{02,07,4} + X_{03,07,4} + X_{04,07,4} + X_{05,07,4} + X_{06,07,4} + X_{07,08,4} + X_{07,09,4} + X_{07,10,4} + X_{07,11,4} - 2Y_{07,4} = 0$

$X_{01,08,4} + X_{02,08,4} + X_{03,08,4} + X_{04,08,4} + X_{05,08,4} + X_{06,08,4} + X_{07,08,4} + X_{08,09,4} + X_{08,10,4} + X_{08,11,4} - 2Y_{08,4} = 0$

$X_{01,09,4} + X_{02,09,4} + X_{03,09,4} + X_{04,09,4} + X_{05,09,4} + X_{06,09,4} + X_{07,09,4} + X_{08,09,4} + X_{09,10,4} + X_{09,11,4} - 2Y_{09,4} = 0$

$X_{01,10,4} + X_{02,10,4} + X_{03,10,4} + X_{04,10,4} + X_{05,10,4} + X_{06,10,4} + X_{07,10,4} + X_{08,10,4} + X_{09,10,4} + X_{10,11,4} - 2Y_{10,4} = 0$

$X_{01,11,4} + X_{02,11,4} + X_{03,11,4} + X_{04,11,4} + X_{05,11,4} + X_{06,11,4} + X_{07,11,4} + X_{08,11,4} + X_{09,11,4} + X_{10,11,4} - 2Y_{11,4} = 0$

$X_{01,06,5} + X_{02,06,5} + X_{03,06,5} + X_{04,06,5} + X_{05,06,5} + X_{06,07,5} + X_{06,08,5} + X_{06,09,5} + X_{06,10,5} + X_{06,11,5} - 2Y_{06,5} = 0$

$X_{01,07,5} + X_{02,07,5} + X_{03,07,5} + X_{04,07,5} + X_{05,07,5} + X_{06,07,5} + X_{07,08,5} + X_{07,09,5} + X_{07,10,5} + X_{07,11,5} - 2Y_{07,5} = 0$

$X_{01,08,5} + X_{02,08,5} + X_{03,08,5} + X_{04,08,5} + X_{05,08,5} + X_{06,08,5} + X_{07,08,5} + X_{08,09,5} + X_{08,10,5} + X_{08,11,5} - 2Y_{08,5} = 0$

$X_{01,09,5} + X_{02,09,5} + X_{03,09,5} + X_{04,09,5} + X_{05,09,5} + X_{06,09,5} + X_{07,09,5} + X_{08,09,5} + X_{09,10,5} + X_{09,11,5} - 2Y_{09,5} = 0$

$X_{01,10,5} + X_{02,10,5} + X_{03,10,5} + X_{04,10,5} + X_{05,10,5} + X_{06,10,5} + X_{07,10,5} + X_{08,10,5} + X_{09,10,5} + X_{10,11,5} - 2Y_{10,5} = 0$

$X_{01,11,5} + X_{02,11,5} + X_{03,11,5} + X_{04,11,5} + X_{05,11,5} + X_{06,11,5} + X_{07,11,5} + X_{08,11,5} + X_{09,11,5} + X_{10,11,5} - 2Y_{11,5} = 0$

$X_{01,06,6} + X_{02,06,6} + X_{03,06,6} + X_{04,06,6} + X_{05,06,6} + X_{06,07,6} + X_{06,08,6} + X_{06,09,6} + X_{06,10,6} + X_{06,11,6} - 2Y_{06,6} = 0$

$X_{01,07,6} + X_{02,07,6} + X_{03,07,6} + X_{04,07,6} + X_{05,07,6} + X_{06,07,6} + X_{07,08,6} + X_{07,09,6} + X_{07,10,6} + X_{07,11,6} - 2Y_{07,6} = 0$

$X_{01,08,6} + X_{02,08,6} + X_{03,08,6} + X_{04,08,6} + X_{05,08,6} + X_{06,08,6} + X_{07,08,6} + X_{08,09,6} + X_{08,10,6} + X_{08,11,6} - 2Y_{08,6} = 0$

$X_{01,09,6} + X_{02,09,6} + X_{03,09,6} + X_{04,09,6} + X_{05,09,6} + X_{06,09,6} + X_{07,09,6} + X_{08,09,6} + X_{09,10,6} + X_{09,11,6} - 2Y_{09,6} = 0$

$X_{01,10,6} + X_{02,10,6} + X_{03,10,6} + X_{04,10,6} + X_{05,10,6} + X_{06,10,6} + X_{07,10,6} + X_{08,10,6} + X_{09,10,6} + X_{10,11,6} - 2Y_{10,6} = 0$

$X_{01,11,6} + X_{02,11,6} + X_{03,11,6} + X_{04,11,6} + X_{05,11,6} + X_{06,11,6} + X_{07,11,6} + X_{08,11,6} + X_{09,11,6} + X_{10,11,6} - 2Y_{11,6} = 0$

$Y_{06,1} + Y_{06,2} = 1$

$Y_{06,3} + Y_{06,4} = 1$

$Y_{06,5} + Y_{06,6} = 1$

$Y_{06,1} + Y_{06,3} + Y_{06,5} = 3W_{06}$

$Y_{06,2} + Y_{06,4} + Y_{06,6} = 3Q_{06}$

$W_{06} + Q_{06} = 1$

$Y_{08,1} + Y_{08,2} = 1$

$Y_{08,3} + Y_{08,4} = 1$

$Y_{08,5} + Y_{08,6} = 1$

$Y_{08,1} + Y_{08,3} + Y_{08,5} = 3W_{08}$

$Y_{08,2} + Y_{08,4} + Y_{08,6} = 3Q_{08}$

$W_{08} + Q_{08} = 1$

$Y_{10,1} + Y_{10,2} = 1$

$Y_{10,3} + Y_{10,4} = 1$

$Y_{10,5} + Y_{10,6} = 1$

$Y_{10,1} + Y_{10,3} + Y_{10,5} = 3W_{10}$

$Y_{10,2} + Y_{10,4} + Y_{10,6} = 3Q_{10}$

$W_{10} + Q_{10} = 1$

$Y_{07,1} + Y_{07,2} + Y_{07,3} = 1$

$Y_{07,4} + Y_{07,5} + Y_{07,6} = 1$

$Y_{07,1} + Y_{07,4} = 2W_{07}$

$Y_{07,2} + Y_{07,5} = 2Q_{07}$

$Y_{07,3} + Y_{07,6} = 2R_{07}$

$W_{07} + Q_{07} + R_{07} = 1$

$Y_{09,1} + Y_{09,2} + Y_{09,3} = 1$

$Y_{09,4} + Y_{09,5} + Y_{09,6} = 1$

$Y_{09,1} + Y_{09,4} = 2W_{09}$

$Y_{09,2} + Y_{09,5} = 2Q_{09}$

$Y_{09,3} + Y_{09,6} = 2R_{09}$

$W_{09} + Q_{09} + R_{09} = 1$

$Y_{11,1} + Y_{11,2} + Y_{11,3} = 1$

$Y_{11,4} + Y_{11,5} + Y_{11,6} = 1$

$Y_{11,1} + Y_{11,4} = 2W_{11}$

$Y_{11,2} + Y_{11,5} = 2Q_{11}$

$Y_{11,3} + Y_{11,6} = 2R_{11}$

$W_{11} + Q_{11} + R_{11} = 1$

X_{ijk} Y_{ik} W_i Q_i and R_i $\in \{0, 1\}$

Solution is six tours, with nodes 1 to 5 on all tours,
nodes 6, 8, and 10 on 3 tours, and
nodes 7, 9, and 11 on two tour.

References

M. Ball and M. Magazine, 1981, "*The Design and Analysis of Heuristics*", Networks 11, pp 215-219.

E. Balas, 1989, "*The Prize Collecting Travelling Salesman Problem*", Networks 19, pp 621-636.

E. Balas and N. Christofides, 1976, "*A New Penalty Method for the Travelling Salesman Problem*", Presented at the 9th Math. Prog. Symposium, Budapest.

J. E. Beasley, 1984, "*Fixed Routes*", J. Opl Res. Soc. Vol 35, pp 49-55.

M. Bellmore and G. L. Nemhauser, 1968, "*The Travelling Salesman Problem : A Survey*", Operations Research 16, pp 538-558.

M Butler, H P Williams, and L A Yarrow, 1997, "*The Two-Period Travelling Salesman Problem Applied to Milk Collection in Ireland*", To appear : Computational Optimization and Applications 7, pp 1-16.

V. Chvátal, 1973, "*Edmond's Polytopes and Weakly Hamiltonian Graphs*", Maths. Prog 3.

N. Christofides and J. E. Beasley, 1984, "*The Period Routing Problem*", Networks 14, pp 237-256.

N. Christofides, A. Mingozzi, P. Toth, and C. Sandi, 1979, "*Combinatorial Optimisation*", John Wiley a Sons.

N. Christofides, 1977, "*Worst-Case Analysis of a New Heuristic for the Travelling Salesman Problem*", Management Science Research Report No 388, Carnegie-Mellon University.

A. Claus, 1984, "*A New Formulation For The Travelling Salesman Problem*", SIAM J. Alg. Disc. Meth., Vol 5, pp 21-25.

G. Clarke and J. W. Wright, 1964, "*Schedulling of Vehicles from a Cetral Depot to a Number of Delivery Points*", Operations Research 12, pp 568-581.

G. Cornuejols and G. L. Nemhauser, 1978, "*Tight Bounds for Christofides' Travelling Salesman Heuristic*", Mathematical Programming 14, pp 16-121.

G A Croes, 1958, "*A Method for Solving Travelling Salesman Problems*", Operations Research 6, pp 791-812.

H. Crowder and M. W. Padberg, 1980, "*Solving Large-Scale Travelling Salesman Problems toOptimality* ", Management Science 26, pp 495-509.

G. B. Dantzig, Fulkerson, D. R., and Johnson, S. M., 1954, "*Solution of a Large-Scale Travelling Salesman Problem*", Operations Research, Vol 2, pp 393-410.

C. Derman and M. Klein, 1966, "*Surveillance of Multi-Component Systems: A Stochastic Travelling Salesman Problem*", Naval Res. Logist. Quart. 13, pp 103-111.

M. L. Fisher, 1981, "*The Lagrangian Relaxation Method for Solving Integer Programming Problems*", Management Science, Vol 27, No. 1, pp 1-18.

M. L. Fisher, 1980, "*Worst-Case Analysis of Heuristic Algortihms*", Management Science 26, pp 1- 17.

M. L. Fisher, 1985, "*An Applications Oriented Guide to Lagrangian Relaxation*", Interfaces 15, No. 2, pp 10-21.

K. R. Fox, B. Gavish, and S. C. Graves, 1980, "*An n-Constraint Formulation of the Time Dependent Travelling Salesman Problem*", Operations Research 2, pp 191-194.

R. S. Garfinkel and K.C. Gilbert, 1978, "*The Bottleneck Travelling Salesman Problem: Algorithms and Probabilistic Analysis*", J. Assoc. Comput. Mach. 25, pp 435-448.

M. R. Garey and D. S. Johnson, 1979, "*Computers and Intractability: A guide to the Theory of NP-Completeness*", Freeman, San Francisco.

B E Gillet and L R Miller, 1974, "*A Heuristic Algorithm for the Vehicle Dispatch Problem*", Operations Research, 22, 340-349

M Grotschel, and M W. Padberg, 1977, "*On the Symmetric Travelling Salesman Problem: Theory and Computation*". Optimisation and Operations Research, Lecture Notes in Economics and Mathematical Systems 157, Springer, Berlin, pp 105-115.

M. Held and Karp, R. M. 1970, "*The Tavelling Salesman Problem and Minimum Spanning Trees*", Operations Research, Vol 18, pp 1138-1162.

M. Held and Karp, R. M. 1971, "*The Travelling Salesman Problem and Minimum Spanning Trees: Part II*", Math. Programming I, pp 6-25.

M. Held, Wolfe, P., and Crowder, H. P., 1974, "*Validation of Subgradient Optimization*", Math. Programming 6, pp 62-88.

D. Houck, J. Picard, M. Queyranne, and R. Vemuganti, 1977, "*The Travelling Salesman Problem and Shortest n-Paths*", University of Maryland.

D. S. Johnson and L. A. McGeoch, 1995, "*The Travelling Salesman Problem: A Case Study in Local Optimization*", Department of Mathematics and Computer Science, Amherst College, Amherst, MA01002.

D. S. Johnson, McGeoch, L. A., and Rothberg E., 1996, "*Asymptotic Experimental Analysis for the Held-Karp Travelling Salesman Bound*", Proceedings of the 7th Annual ACM-Siam Symposium on discrete algorithms.

D S Johnson and C H Papadimitriou, 1985, "*Performance Guarantees for Heuristics*", The Travelling Salesman Problem, Edited by E L Lawler, L K Lenstra, A H G Rinnooy Kan, and D B Shmoys, Chapter 5.

S Kirkpatrick, C D Gelatt, and M P Vecchi, 1982, "*Optimisation by Simulated Annealing*", IBM Computer Science/Engineering Technology Report, IBM Thomas J Watson Research Centre, Yorktown Heights, NY.

A. Land and S. Powell, 1973, "*Fortran Codes for Mathematical Programming*", Wiley, New York.

E. L. Lawler, Lenstra J. K., Rinnooy Kan, A .H. G., and Shmoys D. B., 1985, "*The Travelling Salesman Problem*", John Wiley & Sons.

S Lin, 1965, "*Computer Solutions of the Travelling Salesman Problem*", Bell System Tech J. 44, pp 2245-2269.

S Lin and B W Kernighan, 1973,"*An Effective Heuristic Algorithm for the Travelling Salesman Problem*", Operations Research 21, 498-516.

J. D. Little, Murty, K. G., Sweeney, D. W., and Karl, C., 1963, "*An Algorithm for the Traveling Salesman Problem*", Operations Research, Vol 11, pp 972-989.

P. Miliotis, 1976, "*Integer Programming Approaches to the Travelling Salesman Problem*", Math. Programming 10, pp 367-378.

C. E. Miller, A. W. Tucker, and R. A. Zemlin, 1960, "*Integer Programming Formulations and Traveling Salesman Problems* ", J. ACM 7, pp 3236-329.

M Padberg and S Hong, 1980, "*On the Symmetric Travelling Salesman Problem : A Computational Study*", Mathematical Programnming Study 12, pp 78-107.

R. Russell, 1977, "*An Effective Heuristic for the M-Tour Travelling Salesman Problem with Some Side Conditions*", Operations Research 25, pp 517 - 524.

R. Russell and W. Igo, 1979, "*An Assignment Routing Problem*", Networks 9, pp 1-17.