UNIVERSITY OF Southampton

University of Southampton Research Repository
ePrints Soton

Copyright © and Moral Rights for this thesis are retained by the author and/or other copyright owners. A copy can be downloaded for personal non-commercial research or study, without prior permission or charge. This thesis cannot be reproduced or quoted extensively from without first obtaining permission in writing from the copyright holder/s. The content must not be changed in any way or sold commercially in any format or medium without the formal permission of the copyright holders.

When referring to this work, full bibliographic details including the author, title, awarding institution and date of the thesis must be given e.g.

AUTHOR (year of submission) "Full thesis title", University of Southampton, name of the University School or Department, PhD Thesis, pagination

http://eprints.soton.ac.uk

UNIVERSITY of SOUTHAMPTON

FACULTY of BUSINESS and LAW

SCHOOL of MANAGEMENT

Models and Algorithms for the Pollution-Routing Problem and Its Variations

by

Emrah DEMİR

Thesis for the degree of Doctor of Philosophy

July 2012

UNIVERSITY of SOUTHAMPTON

Abstract

FACULTY of BUSINESS and LAW
SCHOOL of MANAGEMENT

Doctor of Philosophy

by Emrah DEMİR

This thesis is positioned within the field of green logistics with respect to CO_2 emissions in road freight transportation. In order to examine the different aspects of CO_2 emissions of freight transportation, three related, but different research questions are studied. Because CO_2 emissions are proportional to the amount of the fuel consumed by vehicles, the first goal of the thesis is to review and compare several available fuel emission models. The results of extensive computational experiments show that all emission models tested are sensitive to changes in load, speed and acceleration. Second, the dissertation studies the Pollution-Routing Problem (PRP), an extension of the classical Vehicle Routing Problem with Time Windows (VRPTW). The PRP consists of routing a number of vehicles to serve a set of customers within preset time windows, and determining their speed on each route segment, so as to minimise a function comprising fuel, emission and driver costs. A mathematical formulation of this problem cannot be solved to optimality for medium to large scale instances. For this reason, the thesis describes an adaptive large neighbourhood search (ALNS) based algorithm to solve the PRP. The algorithm iterates between a VRPTW and a speed optimisation problem, where the former is solved through an enhanced ALNS and the latter is solved using a polynomial time speed optimisation algorithm (SOA). The third question relates to the PRP and the two important objectives that should be taken into account, namely minimisation of fuel consumption and total driving time. Computational results on a large set of PRP instances show that the algorithm is both effective and efficient in solving instances of up to 200 nodes. The thesis therefore studies the bi-objective PRP where one of the objectives is related to the environment, namely fuel consumption (hence CO_2 emissions), and the other to driving time. An enhanced ALNS algorithm is described to solve the bi-objective PRP. The algorithm integrates the classical ALNS scheme with a specialized SOA. The results show that one need not compromise greatly in terms of driving time in order to achieve a significant reduction in fuel consumption and CO_2 emissions.

Contents

List of Figures

List of Tables

Declaration of Authorship

I, EMRAH DEMİR, declare that this thesis titled, 'Models and Algorithms for the Pollution-Routing Problem and Its Variations' and the work presented in it are my own. I confirm that:

- This work was done wholly or mainly while in candidature for a research degree at this University.
- Where any part of this thesis has previously been submitted for a degree or any other qualification at this University or any other institution, this has been clearly stated.
- Where I have consulted the published work of others, this is always clearly attributed.
- Where I have quoted from the work of others, the source is always given. With the exception of such quotations, this thesis is entirely my own work.
- I have acknowledged all main sources of help.
- Where the thesis is based on work done by myself jointly with others, I have made clear exactly what was done by others and what I have contributed myself.

Signed: **Emrah DEMİR**

Date: **July 2012**

Acknowledgements

This thesis arose in part out of years of research that has been done since I came to University of Southampton. By that time, I have worked with a great number of people whose contribution in assorted ways to the research and the making of the thesis deserve special mention. I would never have been able to finish my thesis without the guidance of those people. It is now a pleasure to convey my gratitude to them all in my humble acknowledgement.

First and foremost, I gratefully acknowledge my main supervisor Dr Tolga Bektaş, not only for his continuous feedback, guidance, and reviewing of every chapter in this thesis, but also for his unlimited support and patient encouragement, and for challenging me to always deliver the best throughout the whole period of my study, since day one to the day of the final submission. His involvement and his originality have triggered and nourished my intellectual maturity from which I will benefit, for a long time to come. Dr Bektaş, I am grateful in every possible way and hope to keep up our collaboration in the future.

I would like to thank my other supervisor, Prof. Gilbert Laporte, for guiding my research from the beginning and helping me to develop my background in operational research. His expertise in this area has helped improve my research skills and prepare me for future challenges. I have been extremely lucky to have these two supervisors who cared so much about my work, and who responded to my questions so promptly. Special thanks go to Prof. Julia Bennell, Prof. Chris Potts, Prof. Tom Van Woensel, Dr Güneş Erdoğan and Dr Maria Battarra, who were willing to participate in my presentations and giving me valuable feedbacks.

I would like to express my thanks to my friends and colleagues in the School of Management and the University of Southampton who made my whole stay such a great and enjoyable experience and for being there when I needed them.

I would also like to acknowledge LANCS Initiative for funding this project.

Last but not least, my deepest gratitude goes to my beloved parents' Mr. Celil Demir and Mrs. Münevver Demir and also to my brothers for their endless love, and encouragement. I would like to thank everybody who was important to the successful realisation of thesis, as well as expressing my apologies to those I could not mention personally.

This thesis is dedicated to my dearest family.

Bu tez sevgili aileme ithaf edilmiştir.

Abbreviations

Chapter I

GHG	**G**reen **H**ouse **G**as
CO_2	Carbon Dioxide
TSP	**T**ravelling **S**alesman **P**roblem
VRP	**V**ehicle **R**outing **P**roblem
CVRP	**C**apacitated **V**ehicle **R**outing **P**roblem
DVRP	**D**istance-Constrained **V**ehicle **R**outing **P**roblem
VRPB	**V**ehicle **R**outing **P**roblem with **B**ackhauls
VRPPD	**V**ehicle **R**outing **P**roblem with **P**ickup and **D**elivery
VRPTW	**V**ehicle **R**outing **P**roblem with **T**ime **W**indows
TDVRP	**T**ime **D**ependent **V**ehicle **R**outing **P**roblem
CVRSO	**C**omputerised **V**ehicle **R**outing and **S**cheduling **O**ptimisation
EVRP	**E**missions **V**ehicle **R**outing **P**roblem
EMVRP	**E**nergy-**M**inimising **V**ehicle **R**outing **P**roblem
FCVRP	**F**uel **C**onsumption **V**ehicle **R**outing **P**roblem
PRP	**P**ollution-**R**outing **P**roblem

Chapter II

Model 1	An instantaneous fuel consumption model
Model 2	A four-mode elemental fuel consumption model
Model 3	A running speed fuel consumption model
Model 4	A comprehensive modal emission model
Model 5	Methodology for calculating transportation emissions and energy consumption (MEET) model
Model 6	Computer programme to calculate emissions from road transportation (COPERT) model
GVWR	**G**ross **V**ehicle **W**eight **R**ating

LD	**L**ight **D**uty vehicle
MD	**M**edium **D**uty vehicle
HD	**H**eavy **D**uty vehicle

Chapter III

ALNS	**A**daptive **L**arge **N**eighbourhood **S**earch
SOP	**S**peed **O**ptimisation **P**roblem
SOA	**S**peed **O**ptimisation **A**lgorithm
NP-hard	**N**on-deterministic **P**olynomial-time **h**ard
CW	**C**larke and **W**right algorithm
RR	**R**andom **R**emoval operator
WDR	**W**orst-**D**istance **R**emoval operator
WTR	**W**orst-**T**ime **R**emoval operator
RoR	**R**oute **R**emoval operator
SR	**S**haw **R**emoval operator
PR	**P**roximity-based **R**emoval operator
TR	**T**ime-based **R**emoval operator
DR	**D**emand-based **R**emoval operator
HR	**H**istorical knowledge node **R**emoval operator
NR	**N**eighbourhood **R**emoval operator
ZR	**Z**one **R**emoval operator
NNR	**N**ode **N**eighbourhood **R**emoval operator
GI	**G**reedy **I**nsertion operator
RI	**R**egret **I**nsertion operator
GIN	**G**reedy **I**nsertion with **N**oise function operator
RIN	**R**egret **I**nsertion with **N**oise function operator
ZI	**Z**one **I**nsertion operator

Chapter IV

MOO	**M**ulti-**O**bjective **O**ptimisation
WM	**W**eighting **M**ethod
WMN	**W**eighting **M**ethod with **N**ormalisation
ECM	ϵ-**C**onstraint **M**ethod
HM	**H**ybrid **M**ethod
$I_{hv}(\mathcal{S})$	**H**ypervolume **I**ndicator
$I_{\epsilon}(\mathcal{S},\mathcal{R})$	**E**psilon **I**ndicator

Constants

Chapter I

A litre of diesel	C	=	2.67 kg of CO_2

Chapter II

Constant idle fuel rate	α	=	0.375–0.556 mL / s
Fuel consumption per unit of energy	β_1	=	0.05–0.16 mL / kJ
Fuel consumption per unit of energy-acceleration	β_2	=	0.01–0.05 mL / (kJ m / s^2)
Rolling drag force	b_1	=	0.1–0.7 kN
Rolling aerodynamics force	b_2	=	0.0003–0.0015 kN / (m / s^2)
Function parameter	A	=	21–100
Function parameter	B	=	0.0055–0.018
Air density	ρ	=	1.2041 kg / m^3
Vehicle drive train efficiency	n_{tf}	=	0.4
Efficiency parameter for diesel engines	η	=	0.9
Vehicle accessories fuel consumption	P_{acc}	=	0 hp
Engine displacement	V	=	5 litres
Coefficient of aerodynamic drag	C_d	=	0.7
Coefficient of rolling resistance	C_r	=	0.01
Gravitation constant	g	=	9.81 m / s^2
Engine friction factor	k	=	0.2 kJ / (rev / L)
Engine speed	N	=	33 rev / s
Conversion factor (g/s to L/s)	ψ	=	737
Frontal surface area	A	=	3.912 m^2
Fuel-to-air mass ratio	ξ	=	1
Heating value of a typical diesel fuel	κ	=	44 kJ / g
Idle fuel rate of a warm engine	α	=	0.1–1.0 mL / s
Road gradient	ω	=	0.00003–0.0015 deg

Curb weight	w	=	6350 kg
Driver wage per hour	f_d	=	£8
Fuel and CO_2 emissions cost per litre	f_c	=	£1.4

Chapter III

Total number of iterations	N_i	=	25000
Number of iterations for roulette wheel	N_w	=	450
Roulette wheel parameter	r_p	=	0.1
New global solution	σ_1	=	1
Better solution	σ_2	=	0
Worse solution	σ_3	=	5
Start-up temperature	P_{init}	=	100
Cooling rate	h	=	0.999
Lower limit of removable nodes	$\underline{s}$	=	5–20% of $\|\mathcal{N}\|$
Upper limit of removable nodes	$\overline{s}$	=	12–30% of $\|\mathcal{N}\|$
Zone parameter	z	=	11
First Shaw parameter	Φ_1	=	0.5
Second Shaw parameter	Φ_2	=	0.25
Third Shaw parameter	Φ_3	=	0.15
Fourth Shaw parameter	Φ_4	=	0.25
Noise parameter	μ	=	0.1

Chapter IV

Total number of loops	N_1	=	11
Total number of iterations	N_2	=	10000
Number of iterations for roulette wheel	N_3	=	200
Lower limit of removable nodes	$\underline{s}$	=	4
Upper limit of removable nodes	$\overline{s}$	=	16
The increase rate of the ECM	ς	=	300 s

Symbols

Chapter I

$\mathcal{N}$	Set of nodes	$\mathcal{N} = \{0, 1, 2, ..., n\}$
$\mathcal{N}_0$	Set of customers	$\mathcal{N}_0 = \mathcal{N} \backslash \{0\}$
$\mathcal{A}$	Set of ordered pairs of nodes, called arcs	$\mathcal{A} = \{(i, j) : i, j \in \mathcal{N}, i \neq j\}$
$\mathcal{G}$	A complete undirected graph	$\mathcal{G} = (\mathcal{N}, \mathcal{A})$
$\mathcal{K}$	A fixed-size fleet of vehicles	$\mathcal{K} = \{1, 2, ..., m\}$
$\mathcal{B}_1$	Set of linehaul customers	$\mathcal{B}_1 = \{1, 2, ..., n_b\}$
$\mathcal{B}_2$	Set of backhaul costumers	$\mathcal{B}_2 = \{n_b, n_{b+1}, ..., n\}$
$\mathcal{D}_i$	The origin of delivery of node i	$(i \in \mathcal{N}_0)$
$\mathcal{O}_i$	The destination of delivery of node i	$(i \in \mathcal{N}_0)$
Q	Vehicle capacity	kg
a_i	A lower bound on the time window of node i	$(i \in \mathcal{N})$
b_i	An upper bound on the time window of node i	$(i \in \mathcal{N})$
p_i	A non-negative supply for every i	$(i \in \mathcal{N}_0)$
q_i	A non-negative demand for every i	$(i \in \mathcal{N}_0)$
d_{ij}	Distance between node i and j	$(i, j \in \mathcal{N})$
t_i	Service time of customer i	$(i \in \mathcal{N}_0)$

Chapter II

f_t	Fuel consumption per unit time	mL / s
R_t	Total tractive force	kN
v	Vehicle speed	m / s
τ	Acceleration	m / s^2
M	Vehicle weight	kg
ω	Percent grade	%
F_t	Total fuel consumption	mL
t_0	Total journey time	s

F_a	Total fuel consumption during acceleration	mL
F_d	Total fuel consumption during deceleration	mL
F_c	Total cruise fuel consumption	mL
F_i	Total fuel consumption while idle	mL
F_s	Total fuel consumption	mL
v_i	Initial speed	m / s
v_f	Final speed	m / s
v_c	Cruise speed	m / s
v_r	Average running speed	km / h
t_a	Acceleration time	s
t_d	Deceleration time	s
t_c	Cruise time	s
t_i	Idle time	s
t_s	Travel time	s
x_a	Acceleration distance	km
x_c	Cruise distance	km
x_d	Deceleration distance	km
E_k	The change in kinetic energy per unit mass per unit distance during acceleration	J / kgm
E_{k+}	Sum of positive kinetic energy change	
Θ	Road gradient	$\deg$
P_{tract}	Total tractive power requirement	kW
P	Second-by-second engine power output	kW
FR	Fuel rate	g / km
ϵ	The rate of emissions	g / km
GC	Road gradient function	
LC	Load correction function	
F	Corrected CO_2 emissions	g

Chapter III

x_{ij}	A binary variable equal to 1 if arc (i, j) appears in a solution, and 0 otherwise	$(i, j \in \mathcal{N})$
f_{ij}	Total amount of flow on each arc (i, j)	$(i, j) \in \mathcal{A}$
y_j	The time at which service starts at node j	$(j \in \mathcal{N}_0)$
z^r_{ij}	A binary variable equal to 1 if arc (i, j)	

	$\in \mathcal{A}$ is traversed at a speed level r, and 0 otherwise	
$\bar{v}^r$	R non-decreasing speed levels $\bar{v}^r$	$(r = 1, 2, ..., R)$
s_j	The total time spent on a route that has a node $j \in \mathcal{N}_0$ as last visited before returning to the depot	s
e_i	The arrival time at node i	s
w_i	The waiting time at node i	s
v_i^l	Minimum allowable speed	m / s
v_i^u	Maximum allowable speed	m / s
$F(v)$	Fuel consumption on arc (i, j) when all parameters are fixed except for v	L
L	A large number	
$\mathcal{L}$	Removal list	
$\mathcal{S}$	Subset of nodes for removal	
X	A feasible solution	
X_p	A partly destroyed solution	
X_{init}	An initial feasible solution	
X_{best}	Best solution found during the search	
$X_{current}$	Current solution at any iteration	
X_{new}	Temporary solution at the end of an iteration	
P_d^t	Probability of removal operator d	$(d = 1, 2, ..., 12)$
P_i^t	Probability of insertion operator i	$(i = 1, 2, ..., 5)$
T	Temperature	

Chapter IV

$f_i(x)$	An objective function i	
k	The number of objective functions	
w_i	The weight of objective function i	

Chapter 1

Introduction

1.1 Overview

Logistics is the management of the flow of goods, information and other resources, including energy and people, between a point of origin and a point of consumption in order to meet the requirements of end-users. Logistics activities comprise freight transportation, storage, inventory management, materials handling and all the related information processing. The main objective of logistics is to co-ordinate these activities in a way to meet customer requirements at minimum cost to the service provider (Crainic and Laporte, 1997).

Green logistics is related to producing and dispatching goods in a sustainable way, while playing attention to environmental and social factors. In a "green" context, the objectives are not only based on economic considerations, but also aim at minimising other detrimental effects on society, such as pollution, on the environment (Sbihi and Eglese, 2007*a*). These effects include resource consumption, land use, acidification, toxic effects on ecosystems and humans and greenhouse gas (GHG) emissions (Knörr, 2009).

With an ever growing concern for the environment by governments and other private entities worldwide, several companies have started taking external costs of logistics into account, associated mainly with climate change, air pollution, noise, vibration and accidents (Forkenbrock, 2001; Rondinelli and Berry, 2000). Such factors should also be taken into account in the design of logistics activities,so as to yield environmental benefits, in particular the reduction of GHG emissions.

GHGs absorb and emit radiations within the thermal infra-red range in the atmosphere. The main GHGs are water vapour, CO_2, methane, nitrous dioxide and ozone. GHGs affect the temperature of the Earth significantly; without the Earth's surface temperature would be $33°C$ colder than at present on average. CO_2 levels were around 280 parts per million by volume (ppmv) before the industrial revolution started. Human activities since the start of the industrial era around 1750 have increased the levels of GHGs in the atmosphere. As of April 2012, CO_2 levels are estimated to be at 393 ppvm and are constantly increasing (Komhyr et al., 1989). Reducing CO_2 levels needs more attention from the scientific community.

Significant sources of GHG emissions are power stations, industrial processes and transportation. The transportation sector is responsible for 26% of the total emissions with an expected increase to 41% by 2050 (Bristow et al., 2004). The United States (US) transportation system is the world's largest, and is emitting more than 1800 million metric tons of CO_2 per year (Lattin and Utgikar, 2007). In California, the transportation sector is the largest contributor of GHG

emissions, making up over 40% of the state's total in 2006 (Yang et al., 2009). In United Kingdom (UK), the transportation sector is the third largest source of GHG emissions (Tight et al., 2005). Freight transport in the UK is responsible for 22% of the CO_2 emissions from the transportation sector, amounting to 33.7 million tonnes, or 6% of the CO_2 emissions in the country, of which road transport accounts for a proportion of 92% (McKinnon, 2007). As a result of the European Union (EU) enlargement, the European continental freight transportation demand has risen by 40%. In 2001, this traffic was responsible for around 29% of transportation-related CO_2 emissions or about 6% of total CO_2 emissions in Germany (Léonardi and Baumgartner, 2004).

The planning freight transportation activities has mainly focused on ways to save money or to increase profitability by considering internal transportation costs only, e.g., fuel cost, drivers' wages (see, e.g., Crainic, 2000; Forkenbrock, 1999, 2001). Green freight transportation considers external costs besides the traditional costs. GHGs also have an estimated cost; these costs are difficult, but not impossible, to quantify. Indeed, there exist several of published studies on the estimation of the social costs of CO_2 emissions (Forkenbrock, 2001).

The social cost of carbon is the reduction in quality of the environment and life as a result of carbon dioxide emissions (Price et al., 2007). Most of the published studies consider the estimate of social cost to be approximately £70 per ton carbon (tC) for emissions in 2000 (Clarkson and Deyes, 2002). This figure increases by approximately £1/tC per year in real terms for each subsequent year to account for the increasing social costs over time. A better approach could be to use the £70/tC as an illustrative point estimate of marginal damages, but to also use an upper value of £140/tC and a lower value of £35/tC for sensitivity analyses.

This chapter serves as an overview on the relevant body of literature to the thesis. Its structure is as follows. Section 1.2 presents a brief review of the VRP. Section 1.3 provides an overview of the existing green transportation planning literature. The overall structure of the thesis is given in Section 1.4.

1.2 The Vehicle Routing Problem

Freight transportation has many facets, particularly when viewed from the multiple levels of decision making. Arguably the most famous problem at the operational level is the Vehicle Routing Problem (VRP) (Cordeau et al., 2007). In this section, an overview of the VRP and its basic variants are presented.

The VRP consists of designing optimal delivery or collection routes for a set of vehicles from a central depot to a set of geographically scattered customers, subject to various constraints, such as vehicle capacity, route length, time windows, precedence relations between customers, etc. (Laporte, 2007). One of the first articles on the subject is due to Dantzig and Ramser (1959). Several variants of the problem are still used widely to solve real life problems (Golden et al., 2008). The VRP arises naturally as a central problem in the fields of transportation, distribution and logistics.

The most standard version of the VRP is the Capacitated Vehicle Routing Problem (CVRP), which can be described as follows. Let $\mathcal{G} = (\mathcal{N}, \mathcal{A})$ be a complete undirected graph with node set $\mathcal{N} = \{0, 1, 2, ..., n\}$, where each node $i \in \mathcal{N}\backslash\{0\}$ represents a customer having a non-negative demand q_i, while node 0 corresponds to the depot. Each arc $(i,j) \in \mathcal{A} = \{i,j : i,j \in \mathcal{N}, i \neq j\}$ is associated a distance d_{ij}. A fleet of m identical vehicles, each of capacity Q, is available at the depot (Cordeau et al., 2007). The CVRP calls for the determination of a set of m routes whose total distance is minimised and such that: (*i*) each customer belongs to exactly one route, (*ii*) each route starts and ends at the depot, (*iii*) the total demand of the customers served in a route does not exceed the vehicle capacity Q.

One variant of the VRP is the so-called Distance Constrained VRP (DVRP), where for each route, the capacity constraint is replaced by a constraint such that (iv) the total length of any vehicle route can not exceed a pre-determined limit (Laporte et al., 1984).

The VRP with backhauls (VRPB) is another extension of the CVRP in which the customer set is partitioned into two subsets. The first subset, B_1, contains n_b linehaul customers, each requiring a given quantity of product to be delivered. The second subset, B_2, contains $n - n_b$ backhaul customers, where a given quantity of product must be picked up. Customers are numbered so that $B_1 = (1, ..., n_b)$ and $B_2 = (n_{b+1}, ..., n)$. The VRPB includes constraints (*i*), (*ii*), and two extra constraints such that (*v*) the total demands of the linehaul and backhaul customers visited in each route do not exceed the vehicle capacity Q, and (*vi*) all linehaul customers must precede backhaul customers, if any, in every route (Toth and Vigo, 2001).

Another variant is the VRP with Pickup and Delivery (VRPPD), where a number of goods need to move from certain pickup locations to delivery locations. Each customer $i \in \mathcal{N}\backslash\{0\}$ is associated with two quantities q_i and p_i, representing the demand of a single commodity deliver to and pick-up from customer i, respectively. For each customer $i \in \mathcal{N}\backslash\{0\}$, $\mathcal{O}_i$ denotes the node corresponding to the origin of the delivery, and $\mathcal{D}_i$ denotes the node corresponding to the destination of the associated demand. The goal is to find optimal routes for a fleet of vehicles to visit the pick-up and drop-off locations. The VRPPD includes constraints (*i*), (*ii*),

(*iii*), and (*vii*) for each node i, the node $\mathcal{O}_i$, if different from the depot, must be served in the same route and before node i, and also (*viii*) for each customer i, the customer $\mathcal{D}_i$, if different from the depot, must be served in the same route and after customer i (Toth and Vigo, 2001).

An important extension of the VRP, and the one must relevant to the thesis, is the Vehicle Routing Problem with Time Windows (VRPTW), where each customer should be served within predefined time intervals. In addition to the above-mentioned features of the CVRP, this problem includes, for the depot and for each customer i ($i \in \mathcal{N}_0$), a time window $[a_i, b_i]$ during which this customer has to be served. Furthermore, a_0 denotes the earliest start time and b_0 denotes the latest return time to the depot for each vehicle. The additional constraints are that service should begin at node i ($i \in \mathcal{N}_0$) anytime after a_i but not later than b_i. If the arrival time at node i is earlier than a_i, the vehicle can wait until time a_i to start service. The VRPTW includes constraints (*i*), (*ii*), (*iii*), and (*ix*) for each customer *i*, the service starts within the time window, [a_i,b_i], and the vehicle stops for t_i time units (Cordeau et al., 2007; Toth and Vigo, 2001).

A graphical representation of a feasible VRPTW solution on 13 nodes and with four vehicles is shown in Figure 1.1.

In the Time Dependent VRP (TDVRP), the travel time between two customers or between a customer and the depot depends on the distance between the points and the time of day. The goal is to minimise the total time spent on all routes, which also must ensure deliveries to customers within prespecified time windows. The travelling time is calculated by using information on the departure time and the possible speeds profiles on each arc (i, j), which is assumed to be known at the beginning of the optimisation process (Malandraki and Daskin, 1992).

1.3 Review on Green Vehicle Route Planning

This section is concerned with studies on the VRP in which there is an explicit consideration of environmental concerns, and in particular CO_2 emissions. We use the term *green routing* to denote these type of problems. There exists an extensive amount of research on the VRP, both theoretical and practical. However, the literature on the green routing is still young (Sbihi and Eglese, 2007*b*). Similar to freight transportation, most of the published studies on the VRP are concerned with an overall objective of minimising internal economic costs. However, efforts are being made and research is beginning to emerge in this field (Sbihi and Eglese, 2007*a*).

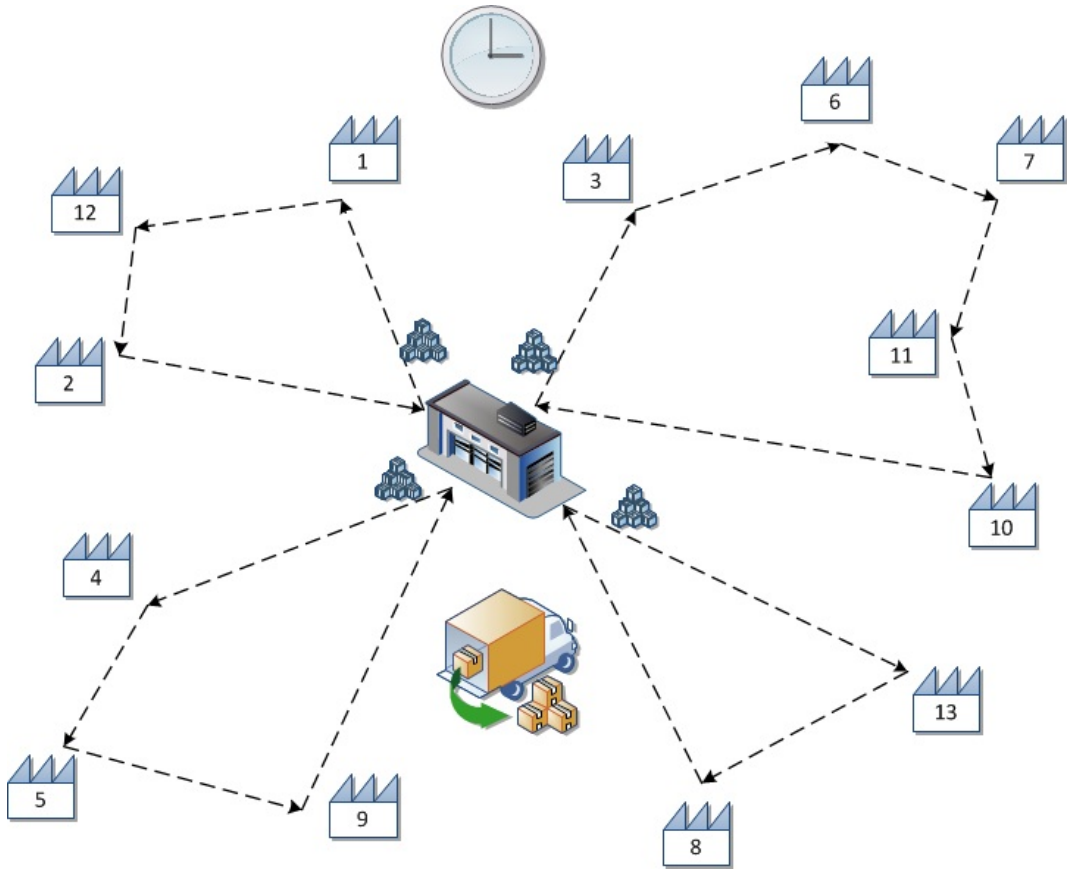


FIGURE 1.1: A graphical representation of the VRPTW

One important work by Eglese and Black (2010) studies the emissions arising in routing and lists some of the factors affecting fuel consumption. In contrast to the existing literature, the authors argue that speed is a more important factor than distance travelled in estimating emissions. Eglese and Black (2010) also mention other relevant factors such as load weight and distribution, vehicle engine, vehicle design, driving style, engine size and road gradient.

This section presents a three-part review of such studies, in a chronological order. First, methodological studies are reviewed. Then, studies using case studies are described. Finally, other means of freight transportation are briefly mentioned in the context of green freight transportation.

1.3.1 Methodological studies

This part of research reviews the methodological studies on the green VRP.

Christie and Satir (2006) focus on the estimation of emission reduction benefits and the potential energy savings that can be achieved through optimisation. The authors aim to quantify the benefits and potential efficiency gains in terms of emissions reduction using a computerised vehicle routing and scheduling optimisation (CVRSO) method. Fuel consumption is estimated using a function of distance, calculated in a simple manner. Their results suggest that reductions of up to 40% in energy consumption and GHG emissions can be achieved by implementing the CVRSO in the trucking industry, compared with any other manual solution techniques.

Palmer (2007) studies vehicle routing and CO_2 emissions models in his PhD thesis, with the VRPTW as the underlying problem. The objective is to find an effective method of identifying vehicle routes that minimise CO_2 emissions. The author considers two estimation models for calculating CO_2 emissions. These are Akçelik's elemental model (Akçelik, 1982) and Bowyer's model (Bowyer et al., 1985). These models are based mainly on the vehicle speed for estimating CO_2 emissions. Palmer (2007) bases his work on Bowyer's model after a comparison of two models. The proposed approach indicates that there exists a potential to reduce CO_2 emissions by around 5% when one moves from time minimised to CO_2 minimised solutions. As a consequence, the total time for traversing the routes rises by 4% and vehicle costs rise by about 0.5%.

In a later work by Kara et al. (2007), the so-called Energy-Minimising VRP (EMVRP) is introduced and formulated. The objective of the EMVRP is to minimise a weighted load function as a way of estimating fuel consumption. The load function is based on the physics rule stating work equals force times distance. The integer linear programming model proposed for the EMVRP is based on that of the CVRP. Since the model minimises the total work done on the road, the authors argue that this leads to minimising the total energy requirements, at least in terms of total fuel consumption. They study the differences between distance-minimising and energy-minimising solutions on benchmark CVRP instances from the literature and find that energy usage increases as total distance decreases. They conclude that there is a considerable difference between energy-minimising and distance-minimising solutions, and that the cost of routes minimising total distance may be up to 13% less than those minimising energy.

A study by Jabali et al. (2012a) investigates travel times and CO_2 emissions in the context of the TDVRP, where the effect of limiting vehicle speed is analysed. CO_2 emissions are modelled as a function of speed as introduced in INFRAS (1995) and in the MEET report (Hickman et al., 1999). This function estimates, in grams per kilometre, emissions for a vehicle travelling at an average speed. Jabali et al. (2012a) also address the issue of congestion, where the vehicle is forced to drive slower and therefore emits more CO_2. The authors describe a formulation of

the problem in which the costs of driving, fuel and CO_2 are included. Results of computational experimentation on benchmark problem sets from Augerat et al. (1998) including between 32 and 80 customers suggest that achieving a reduction of 11.2% in CO_2 emissions on average requires increasing the travel time by 14.8% under a 90 km/h speed profile. Furthermore, they report that an increase of 4.7% in travel time achieves a reduction of 3.7% in CO_2 emissions.

Yong and Xiaofeng (2009) study the VRP to minimise the fuel consumption due to service costs and the impact on the environment. The objective of the study is to minimise fuel consumption only, where fuel consumption per kilometre travelled is assumed to be known in advance. The authors do not consider the effect of vehicle load on fuel consumption. On a small size example, the authors compare two solution approaches, distance minimisation and fuel minimisation, with an enumerative method.

Another relevant study is by Maden et al. (2009), who look at the vehicle routing and scheduling problem to minimise the total travel time under congestion. The authors take into account regular congestion due to volume of traffic, and long-term road works, which can be predicted from historical data. They propose a heuristic algorithm to consider time-varying travel times. In order to reflect the characteristics of a real-life problem, their algorithm allows the time required to travel between locations to vary according to the time at which the journey starts (Eglese et al., 2006). The authors consider the current driving legislation such that: (*i*) there must be a driving break of 45 minutes every 4.5 hours, (*ii*) if the total working time is greater than six hours then a 30 minutes break must be taken and (*iii*) if the total working time is greater than nine hours then a 45 minute break must be taken. Their results suggest that the proposed approach may yield up to 7% savings in CO_2 emissions.

In the context of CO_2 minimisation, Urquhart, Scott and Hart (2010) study the Travelling Salesman Problem (TSP) to identify tours with low CO_2 emissions. The authors examine two different fuel emission models, which are a power based instantaneous fuel consumption model introduced by Bowyer et al. (1985) and a simpler spreadsheet based model from the UK National Atmospheric Emission Inventory. The authors use the latter model in their solution approach. Computational results on six randomly generated instances, each with between 10 and 30 delivery points, suggest that only a small improvement can be achieved using the fuel emission model because of inadequacy of the simple spreadsheet emission model. Another work by Urquhart, Hart and Scott (2010) studies the VRPTW to build low CO_2 solutions using evolutionary algorithms, using an instantaneous fuel consumption model (Bowyer et al., 1985). The authors look at the trade-off between CO_2 savings, distance and the required number of vehicles. Their results indicate that savings of up to 10% can be achieved, depending on the problem instance and the ranking criterion used in the evolutionary algorithm. In

a related study by Scott et al. (2010), the authors study the effect of topology and payload on CO_2 optimised vehicle routing, by using the COPERT estimation model which is based on the average speed of a vehicle and includes payload and correction factors for heavy goods vehicles (Ntziachristos and Samaras, 2000). They work on two different sets of problem data: delivery of groceries to households and delivery of paper from a central warehouse. They also work on several TSP instances randomly chosen from problem datasets, and assign average speeds for each road category. Their results suggest that the effect of gradient and payload are highly dependent on the mixture of the problems studied. The difference in CO_2 emissions between the solutions is found to be less than 2.1% for the COPERT model as in stated in the literature.

A TDVRP model to consider minimising fuel consumption is described by Kuo (2010), who proposes a simulated annealing algorithm for the TDVRP where speed and travel times are assumed to depend on the time of travel. The model not only takes loading weight into consideration, but also satisfies the non-passing property, which states that an early departure time results in an earlier arrival time, and vice versa. The fuel consumption is time-dependent, because travel speeds and travel times depend on the departure time. The author presents results which suggest that the proposed method provides a 24.61% improvement in fuel consumption over a method minimising transportation time and a 22.69% improvement over a method minimising transportation distances. Another work by Kuo and Wang (2011) looks at the VRP with CO_2 emissions. The main difference between these two works is that they use different meta-heuristics. The authors have proposed a tabu search algorithm to optimise the route plan. Their results suggest that fuel savings of up to 8.3% could be achieved by the proposed method.

An emissions VRP (EVRP), which is an extension of the TDVRP, is introduced by Figliozzi (2010), who also describes a formulation and solution approaches for the problem. The objective of the EVRP is the minimisation of emission costs, which are proportional to the amount of GHG emitted which, in turn, is a function of travel speed and distance travelled. In the solution approach, a partial EVRP is first solved to minimise the number of vehicles by using a TDVRP algorithm, and emissions are then optimised subject to a fleet size constraint. The departure times are also optimised using the proposed algorithm for any pair of customers. Computational results obtained using Solomon's test instances suggest that route characteristics and the type of the dominant constraint both play a significant role in the results.

Bektaş and Laporte (2011) introduced the Pollution-Routing Problem (PRP), which considers minimisation of fuel cost and driver costs. In estimating pollution, the authors investigate factors such as speed, load, and time windows, using the emissions functions proposed by Barth

and Boriboonsomsin (2009) and Barth et al. (2005). In these functions, the engine-out emission rate is directly related to the rate of fuel use. Bektaş and Laporte (2011) assume that in a vehicle trip all parameters will remain constant on a given arc, but load and speed may change from one arc to another. Their model approximates the total amount of energy consumed on the arc, which directly translates into fuel consumption and further into GHG emissions. The authors run experiments with data which consists of three classes of problems with 10, 15 and 20 nodes, and each class includes 10 instances where nodes represent UK cities. They analyse cases where customer demands are generated randomly based on a discrete uniform distribution, as well as the effects of variance in demand, of vehicle type and of time windows. Computational results reported by the authors suggest that, by using the proposed approach, energy savings can be up to 10% when time windows are in place, and up to 4% when the demand variation is high.

Suzuki (2011) studies the TSP with time windows (TSPTW) to minimise fuel consumption and pollutants emission considering time constraints and multiple stops for truck-routing. The author formulates three different mathematical models. These are distance-minimising, fuel-minimising and a new fuel-minimising formulations. The author uses the COmputer Programme to calculate Emissions from Road Transport (COPERT) model described by Ntziachristos and Samaras (2000) for the fuel-minimising formulation. In order to test the formulations, the author first applies enumeration for small-size TSPTW instances and then uses compressed annealing, which is designed specifically to solve the TSPTW. Computational results suggest that the proposed formulation may yield up to 6.9% savings in fuel consumption over distance-minimising and fuel-minimising formulations.

The PhD thesis by Qian (2012) studies fuel emission optimisation in VRP with time-varying speeds. The author aims to generate routes and schedules for a fleet of heavy goods vehicles so as to minimise the emissions in a road network where travel speeds depend on time. In order to calculate fuel consumption, the author uses a regression based model proposed by the Department of Transport, which considers speed as a variable. The author describes two route generation algorithms and applies one of them into a column generation based tabu search algorithm to solve the VRP. Computational results obtained by the algorithm on a London case study suggest that of up to 6–7% savings in fuel consumption may be achieved using the proposed approach.

The last work reviewed here is by Xiao et al. (2012), who studies the fuel consumption rate in the context of the CVRP (FCVRP). To estimate fuel consumption, the authors use a regression model based on statistical data, proposed by the Ministry of Land, Infrastructure, Transport and Tourism of Japan. They present a mathematical model and apply a simulated annealing

algorithm to solve the problem. Their results suggest that using FCVRP for route planning may result in up to 5% reduction in fuel consumption as compared with the standard CVRP model.

1.3.2 Case studies

In this section, publications presenting case studies on green road-based freight transportation are reviewed.

Tavares et al. (2008) look at optimisation of routing networks for waste transportation. The authors propose the use of geographic information systems (GIS) 3D route modelling to optimise the route with an aim to minimise fuel consumption in different municipalities of the island of Santo Antao of Cape Verde. Their model takes into account of both the road angle and the vehicle load. Their findings indicate that optimisation of fuel consumption yields up to 52% savings in fuel when compared to routes with the shortest distance, even if this implies increasing the travel distance by 34%. Another work by Tavares et al. (2009) is on the optimisation of municipal solid waste collection routes to minimise fuel consumption using 3D GIS modelling. The authors make use of the COPERT model described by Ntziachristos and Samaras (2000). For the case of the city of Praia, their approach reduces the travelled distance by 29% and fuel consumption by 16%. For the case of the Santiago island, savings in fuel consumption is found to be 12%.

Another real-life application was presented by Apaydin and Gonullu (2008). The authors attempt to control emissions in the context of route optimisation of solid waste in Trabzon, Turkey, with a constant emission factor to estimate fuel consumption. The authors aim to minimise the distance travelled by the trucks. Their results suggest that the route distance and time can be decreased by 24.6% using their approach, with implications of reducing CO_2 emission by 831.4 g on each route.

Maden et al. (2009) propose a heuristic algorithm to minimise the total travel time, where the proposed algorithm also considers the current driving legislation by inserting breaks for a driver when it is necessity in the context of the VRPTW; time-dependant travel times are taken into account. The approach is applied to schedule a fleet of delivery vehicles operating in the South West of the UK. Preliminary experiments are conducted on standard Solomon's benchmark instances. Their results suggest that the total savings in CO_2 may be up to 7%.

The last case study reviewed here is by Ubeda et al. (2011), who investigate the environmental effects of routing in Eroski, Spain. The authors compare four different approaches, namely

the current approach, rescheduling (CVRP), backhauling (VRPB) and green VRP. They use a matrix of emissions based on the estimation of CO_2 emitted between each link as described by Palmer (2007). Their results suggest that the implementation of the green routing approach has benefits from economic and ecological perspectives. In terms of the green aspect of routing, the results suggest that savings of 13.06% in distance and of 13.15% in emissions can be achieved using the green VRP approach.

1.3.3 Other modes of freight transportation

The previous section has reviewed studies on the VRP and its variants from a green perspective. There exist other studies looking at similar issues in other modes of transportation such as air, rail and water shipping. Below, a brief review of a couple of studies that explore similar issues in the broader context of freight transportation is presented.

A study by Bauer et al. (2010) investigates the minimisation of greenhouse gas emissions in inter-modal freight transportation. The primary objective is to minimise environmental-related costs (greenhouses gases) of freight transportation instead of travel or time related costs. The authors make use of the fuel consumption model by Barth et al. (2005), and assume that both fuel consumption and CO_2 emissions are linear functions of the vehicle load. The authors propose an integer linear programming formulation and present an application on a real-life rail service network design problem in Eastern Europe. Computational results on different scenarios indicate that the proposed approach is able to reduce CO_2 emissions by up to 30%.

In shipping, Fagerholt et al. (2010) propose an efficient algorithm for reducing CO_2 emissions from ships travelling on fixed routes. They use a model which calculates fuel consumption per time unit for a cargo ship by a cubic function of the ship speed. The algorithm is based on a discretisation of the time window of each node and solving shortest path problems on a directed acyclic graph. Their results suggest that it is possible to reduce fuel consumption by up to 25%.

A graphical representation of green road freight transportation and green VRP in the context of green logistics is shown in Figure 1.2.

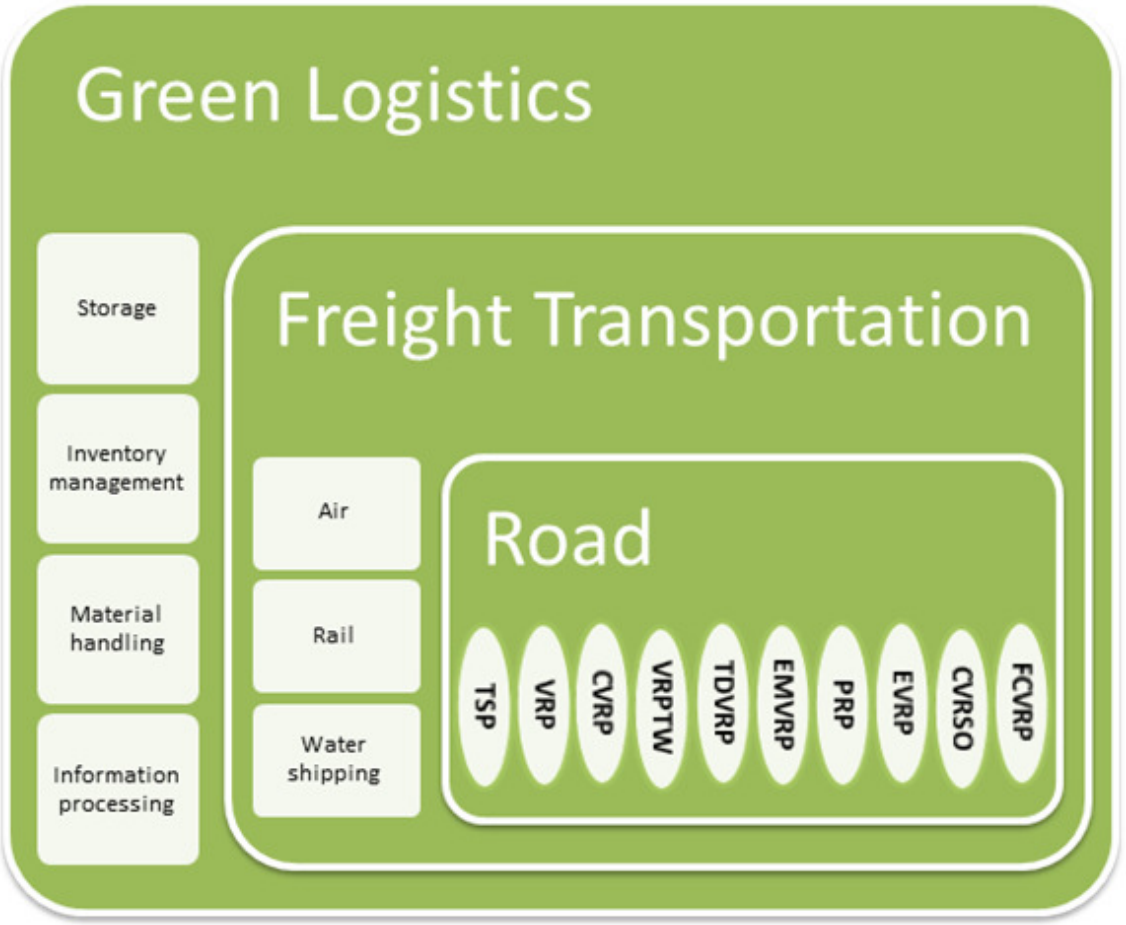


FIGURE 1.2: A graphical representation of green road freight transportation

1.4 Structure of the Thesis

The aim of this thesis is to analyse fuel consumption and CO_2 emissions in road-based freight transportation. The thesis follows a three-paper approach.

An overview of the three research papers, on which this thesis based, is presented in Table 1.1.

Chapter Two, entitled “A Comparative Analysis of Several Vehicle Emission Models for Freight Transportation”, reviews and numerically compares several available freight transportation vehicle emission models. Chapter Three, entitled “An Adaptive Large Neighbourhood Search Heuristic for the Pollution-Routing Problem”, introduces an adaptive large neighbourhood search (ALNS) algorithm and speed optimisation algorithm (SOA) to solve the Pollution-Routing Problem (PRP). Chapter Four, entitled “The Bi-Objective Pollution-Routing Problem”, looks at

TABLE 1.1: Overview of research papers

Research papers	Publication
Research paper I:	"A comparative analysis of several vehicle emission models for road freight transportation" Demir et al. (2011)
Research paper II:	"An adaptive large neighborhood search heuristic for the Pollution-Routing Problem" Demir et al. (2012*a*)
Research paper III:	"The bi-objective Pollution-Routing Problem" Demir et al. (2012*b*)

the bi-objective PRP in which one of the objectives is related to CO_2 emissions, and the other to driving time. Finally, Chapter Five entitled "Conclusions" summarises the contributions of each chapter as well as research limitations. It also provides directions for future research.

Chapter 2

A Comparative Analysis of Several Vehicle Emission Models for Freight Transportation

Abstract

Reducing greenhouse gas emissions in freight transportation requires using appropriate emission models within the planning process. This chapter reviews and numerically compares a number of vehicle emission models that appear in the existing literature. Numerical results in comparing and contrasting the models are presented.

2.1 Introduction

The transportation sector is one of the largest emitters of greenhouse gases (GHGs), especially carbon dioxide (CO_2). In recent years the use of freight transportation has grown as have the levels of carbon dioxide emitted by various modes of transportation. The US transportation system is the world's largest, which emitted more than 1,882 million metric tons of CO_2 ($mmtCO_2$) in 2009 (Conti et al., 2010). The transportation sector is the third largest source of GHGs in the UK, and emitted more than 150 $mmtCO_2$ (25% of the total CO_2 emissions) in 2009 (DECC, 2010), of which freight transportation accounts for 22% of the total. Road transportation represents a proportion of 92% of these emissions (McKinnon, 2007). As the environmental impacts of freight transportation increase, in particular in road-based transportation, reducing emissions becomes more important.

Unlike some other vehicle emitted GHGs, CO_2 is directly proportional to fuel consumption (Kirby et al., 2000). There are two ways to estimate fuel consumption for vehicles: (*i*) *on-road measurements* which are based on real-time collection of emissions data on a running vehicle, and (*ii*) *analytical fuel consumption* (or *emission*) models which estimate fuel consumption based on a variety of vehicle, environment and traffic-related parameters, such as vehicle speed, load and acceleration.

A new line of research, called green logistics, aiming at minimizing the harmful effects of transportation on the environment has started to emerge (Sbihi and Eglese, 2007*a*). In particular, an explicit consideration is typically given to reducing the levels of CO_2 through better operational level planning. Measuring and reducing emissions requires good estimations to be fed into the planning activities, which in turn require estimation models to be incorporated into the

planning methods. Examples of such work include Kara et al. (2007), Palmer (2007), Jabali et al. (2012a) and Bektaş and Laporte (2011) who investigate reducing CO_2 emissions in road-based freight transportation, Bauer et al. (2010) who consider emissions in rail transportation, and Fagerholt et al. (2010) who look at reducing emissions in shipping. The choice of the type and the nature of emission functions is important in order to provide accurate estimates in the planning of transportation activities.

There exists a variety of analytical emission models which differ in the ways they estimate fuel consumption or emissions, or in the parameters they take into account in the estimations. A review of emission models is presented by Ardekani et al. (1996), in which fuel consumption based on traffic management strategies is discussed. The authors group the existing models into two: urban (vehicle speed is less than 55 km/h) and highway (vehicle speed is at least 55 km/h) fuel consumption models. Another review article is due to Esteves-Booth et al. (2002), which presents a classification of some of the available fuel consumption models. The authors review three types of emission models, namely emission factor models, average speed models and modal models.

This chapter aims at presenting a comparative review and analysis of some of the available vehicle emission models. We identify ten such models and present results of a numerical assessment comparing and contrasting six of these, and investigating their behaviour on a variety of parameters. Our research differs from the prior reviews mentioned above in that we take a more analytical approach in comparing and contrasting the available models, and we supplement them with numerical comparison results. In particular, we perform several sensitivity analyses, as well as a comparison with on-road fuel consumption data reported in an empirical study by Erlandsson et al. (2008). The remainder of this chapter is organised as follows. Section 2.2 provides a review and an explicit description of ten vehicle emission models identified in the literature. Section 2.3 presents the results of numerical experiments along with a discussion. Conclusions are stated in Section 2.4.

2.2 Fuel Consumption Models

This section presents a review of emission models. An explicit description of each model is provided, along with a discussion pertaining to its development and applicability.

2.2.1 Model 1: An instantaneous fuel consumption model

An energy-related emissions estimation model, called the *instantaneous fuel consumption model*, or *instantaneous model* in short, is described by Bowyer et al. (1985). It is an extension of the power model published by Kent et al. (1982). The model uses vehicle characteristics such as mass, energy, efficiency parameters, drag force and fuel consumption components associated with aerodynamic drag and rolling resistance, and approximates the fuel consumption per second. The model assumes that changes in acceleration and deceleration levels occur within a one second time interval. The instantaneous model is

$$f_t = \begin{cases} \alpha + \beta_1 R_t v + (\beta_2 M \tau^2 v/1000) & \text{for } R_t > 0 \\ \alpha & \text{for } R_t \leq 0, \end{cases} \tag{2.1}$$

where f_t is the fuel consumption per unit time (mL/s), R_t is the total tractive force (kN = kilonewtons) required to move the vehicle and calculated as the sum of drag force, inertia force and grade force as $R_t = b_1 + b_2 v^2 + M\tau/1000 + gM\omega/100000$. Other parameters used in defining (2.1) are listed and further explained in Table 2.1, along with typical values for these parameters. These values are extracted from Bowyer et al. (1985), and Akçelik and Besley (1996, 2003).

TABLE 2.1: Notation used in Model 1

Notation	Description	Typical values
α	constant idle fuel rate (mL/s)	0.375–0.556
β_1	fuel consumption per unit of energy (mL/kJ)	0.09–0.08
β_2	fuel consumption per unit of energy-acceleration mL/(kJ · m/s^2)	0.03–0.02
b_1	rolling drag force (kN)	0.10–0.70
b_2	rolling aerodynamic force kN/(m/s^2)	0.00003–0.0015
ω	percent grade (%)	
τ	instantaneous acceleration (m/s^2)	
M	total weight (kg)	
v	speed (m/s)	

Using Model 1, the total amount of fuel consumption F_t (mL/s) for a journey of duration t_0 can be calculated as:

$$F_t = \int_0^{t_0} f_t^+ dt + \int_0^{t_0} f_t^- dt. \tag{2.2}$$

where

$$f_t^+ = \begin{cases} \alpha + \beta_1 R_t v + (\beta_2 M \tau^2 v/1000) & \text{for } R_t > 0 \\ 0 & \text{for } R_t \leq 0, \end{cases} \quad and f_t^- = \begin{cases} \alpha & \text{for } R_t < 0 \\ 0 & \text{for } R_t \geq 0, \end{cases}$$

The instantaneous model operates at a micro scale level and is better suited for short trip emission estimations. The model does not make use of macro-level (aggregated) data such as the number of stops. It is, however, able to take into account acceleration, deceleration, cruise and idle phases. Using data from special on-road experiments in Melbourne, Bowyer et al. (1985) showed that the model is able to approximate fuel consumption of individual vehicles within a 5% error margin for short trips. Further dynamometer tests suggested that its accuracy is within 10% for a variety of on-road experiments (Esteves-Booth et al., 2002).

2.2.2 Model 2: A four-mode elemental fuel consumption model

A four-mode *elemental model* is described by Bowyer et al. (1985). The model estimates fuel consumption for each of the four following modes of driving: idle, cruise, acceleration and deceleration. It is a refinement of a function reported by Akçelik (1982), which we do not cover here. The model includes the same parameters as Model 1 but also introduces new parameters, such as initial speed, final speed and energy-related parameters. The model requires data related to the total distance, cruise speed, idle time and average road grade as inputs. A vehicle is said to be in an idle mode when the engine is running but the speed is below 5 km/h. More accurate estimations can be made if the initial and final speeds for each acceleration and deceleration cycles are known. The model consists of four functions, F_a, F_d, F_c and F_i, which correspond to fuel consumption estimations (mL) for acceleration, deceleration, cruise and idle modes, respectively. These functions are described in more detail below.

1. Acceleration fuel consumption F_a
   The following function can be used to calculate the amount of fuel consumption over the acceleration phase of a vehicle from an initial speed v_i to a final speed v_f:

$$F_a = \max\{\alpha t_a + (A + k_1 B(v_i^2 + v_f^2) + \beta_1 M E_k + k_2 \beta_2 M E_k^2 + 0.0981 \beta_1 M \omega) x_a, \alpha t_a\}. \tag{2.3}$$

   Additional notation to that presented in Table 2.1 is given in Table 2.2. In (3), E_k denotes the change in kinetic energy per unit distance during acceleration and is calculated as $E_k = 0.3858\,10^{-4}(v_f^2 - v_i^2)/x_a$. Furthermore, $k_1 = 0.616 + 0.000544 v_f - 0.0171\sqrt{v_i}$ and $k_2 = 1.376 + 0.00205 v_f - 0.00538 v_i$. When the travel distance x_a and the travel time t_a are not known, they can be estimated as $x_a = m_a(v_i + v_f)t_a/3600$ where $m_a = 0.467 + 0.00200 v_f - 0.00210 v_i$ and $t_a = (v_f - v_i)/(2.08 + 0.127\sqrt{v_f - v_i} - 0.0182 v_i)$.

2. Deceleration fuel consumption F_d

   The following function can be used to calculate the amount of fuel consumption during the deceleration phase from an initial speed v_i to a final speed v_f:

$$F_d = \max\{\alpha t_d + (k_x A + k_y k_1 B(v_i^2 + v_f^2) + k_a \beta_1 M E_k + k_x \beta_1 M E_k^2 + 0.0981 \beta_1 M \omega) x_d, \alpha t_d\}, \tag{2.4}$$

   where $k_x = 0.046 + 100/M + 0.00421 v_i + 0.00260 v_f + 0.05444\omega$, $k_y = k_x^{0.75}$, $k_a = k_x^{3.81}(2 - k_x^{3.81})$ and $k_1 = 0.621 + 0.000777 v_i - 0.0189\sqrt{v_f}$. If the travel distance x_d and travel time t_d are not known, they can be estimated as above, although in this case the coefficients change slightly.

3. Cruise fuel consumption F_c

   The following function can be used to calculate the total amount of fuel consumption by a vehicle during a cruise phase allowing for speed fluctuations:

$$F_c = \max\{f_i/v_c + A + B v_c^2 + k_{E1} \beta_1 M E_{k+} + k_{E2} \beta_2 M E_{k+}^2 + 0.0981 k_G \beta_1 M \omega, f_i/v_c\} x_c, \tag{2.5}$$

   where f_i denotes the idle fuel rate (mL/h), v_c is the average cruise speed (km/h), and x_c denotes the travel distance (km). The change in total positive kinetic energy per unit distance during the cruise mode is calculated as $E_{k+} = \max\{0.258 - 0.0018 v_c, 0.10\}$ and the other parameters are set to $k_{E1} = \max\{12.5/v_c + 0.000013 v_c^2, 0.63\}$, $k_{E2} = 3.17$, and $k_G = 1 - 2.1 E_{k+}$ for $\omega < 0$, and $1 - 0.3 E_{k+}$ for $\omega > 0$.

4. Fuel consumption while idle F_i

   The following function can be used to calculate the total amount of fuel consumption when the vehicle is idle:

$$F_i = \alpha t_i, \tag{2.6}$$

   where t_i is the idle time (s), and α is the idle fuel rate (mL/s).

The total fuel consumption F_t (mL) using the elemental model can be calculated as follows:

$$F_t = \int_0^{t_a} F_a dt + \int_0^{t_d} F_d dt + \int_0^{t_c} F_c dt + \int_0^{t_i} F_i dt. \tag{2.7}$$

TABLE 2.2: Notation used in Model 2

Notation	Description	Typical values
A	function parameter (mL/km)	21–100
B	function parameter (mL/km)/(km/h)^2	0.0055–0.018
k_1	integration coefficient	
k_2	intergration coefficient	
k_{E1}	calibration parameter	
k_{E2}	calibration parameter	
k_G	calibration parameter	
k_x	an energy related parameter	
k_y	an energy related parameter	
k_a	an energy related parameter	

The elemental model assumes minimum loss of driving information and hence minimum loss of accuracy in fuel consumption estimates. The model is better suited for estimation of fuel consumption for short distance trips. However, its large number of parameters and the existence of four different functions may make it difficult to implement in practice, in comparison with the other available models. Bowyer et al. (1985) experimented with Model 2 and compared it against the instantaneous model. Their result suggests that the elemental model can predict fuel consumption within a 1% error margin. If the initial and final speeds are known, the model yields more accurate estimates for fuel consumption, and provides results very similar to those of the instantaneous model.

2.2.3 Model 3: A running speed fuel consumption model

The running speed fuel consumption model is an aggregated form of the elemental model and was introduced by Bowyer et al. (1985). The model calculates fuel consumption separately during periods when a vehicle is running and is in an idle mode. The model is as follows:

$$\begin{aligned} F_s = \max\{&\alpha t_i + (f_i/v_r + A + Bv_r^2 + k_{E1}\beta_1 ME_{k+} + k_{E2}\beta_2 ME_{k+}^2 + \\ &0.0981 k_G \beta_1 M\omega) x_s, \alpha t_s\}, \end{aligned} \tag{2.8}$$

where F_s is the total fuel consumption (mL), x_s is the total distance, v_r denotes the average running speed (km/h), t_s and t_i the travel and idle time, respectively. Average speed can be calculated as $v_r = 3600x_s/(t_s - t_i)$. Furthermore, $E_{k+} = \max\{0.35 - 0.0025v_r, 0.15\}$, $k_{E1} = \max\{0.675 - 1.22/v_r, 0.5\}$, $k_{E2} = 2.78 + 0.0178v_r$.

The running speed model is an extension of the instantaneous model and can be viewed as an aggregation of the elemental model. Acceleration, deceleration and cruise modes are considered together within a single function. However, this model does not take into account the idle mode of a vehicle. The running speed model can be used to estimate fuel consumption in a variety of traffic situations, ranging from short to long distance trips, although it is more useful in the latter case.

2.2.4 Model 4: A comprehensive modal emission model

A *comprehensive emissions model* for heavy-good vehicles was developed and presented by Barth et al. (2005); Scora and Barth (2006) and Barth and Boriboonsomsin (2008). It follows to some extent the model of Ross (1994) and is composed of three modules, namely engine power, engine speed and fuel rate, which are summarised as follows:

1. The engine power module:
   The power demand function for a vehicle is obtained from the total tractive power requirements P_{tract} (kW) placed on the vehicle at the wheels:

$$P_{tract} = (M\tau + Mg\sin\theta + 0.5C_d\rho Av^2 + MgC_r\cos\theta)v/1000. \tag{2.9}$$

   To translate the tractive requirement into engine power requirement, the following relationship is used:

$$P = P_{tract}/\eta_{tf} + P_{acc}, \tag{2.10}$$

   where P is the second-by-second engine power output (kW), η_{tf} is the vehicle drive train efficiency, and P_{acc} is the engine power demand associated with running losses of the engine and the operation of vehicle accessories such as usage of air conditioning. Additional notation used in Model 4 is shown in Table 2.3.

2. The engine speed module:
   Engine speed is approximated in terms of vehicle speed as

$$N = S(R(L)/R(L_g))v, \tag{2.11}$$

   where N = engine speed (rpm), S is the engine-speed/vehicle-speed ratio in top gear L_g, $R(L)$ is the gear ratio in gear $L = 1,\ldots,L_g$, and v is the vehicle speed (m/s).

TABLE 2.3: Notation used in Model 4

Notation	Description	Typical values
ρ	air density (kg/m^3)	1.2041
η_{tf}	vehicle drive train efficiency	0.4
η	efficiency parameter for diesel engines	0.45
P_{acc}	vehicle accessories fuel consumption (hp)	0
A	frontal surface area (m^2)	2.1–5.6
C_d	coefficient of aerodynamic drag	0.7
C_r	coefficient of rolling resistance	0.01
g	gravitational constant (m/s^2)	9.81
k	engine friction factor (kJ/rev/L)	0.9
N	engine speed (rev/s)	16–48
V	engine displacement (L)	2–8
θ	road grade angle (degrees)	
τ	instantaneous acceleration (m/s^2)	
M	total weight (kg)	
v	speed (m/s)	

3. The fuel rate module:
   The fuel rate (g/s) is given by the expression

$$FR = \xi(kNV + P/\eta)/44, \tag{2.12}$$

   where ξ is fuel-to-air mass ratio, k is the engine friction factor, and V is the engine displacement.

The comprehensive emissions model is similar to the instantaneous fuel consumption model. However, to produce accurate estimations, it requires detailed vehicle specific parameters for the estimations such as the engine friction coefficient, and the vehicle engine speed. Barth et al. (2005) have tested Model 4 under a variety of traffic scenarios for 23 different vehicle technology categories and different cycles. The same authors have also developed a computer software called the comprehensive modal emission model based on this particular emission model.

2.2.5 Model 5: Methodology for calculating transportation emissions and energy consumption (MEET) model

A publication of the European Commission by Hickman et al. (1999) on emission factors for road transportation (INFRAS, 1995) describes a methodology called MEET, used for calculating transportation emissions and energy consumption for heavy good vehicles. This methodology includes a variety of estimating functions, which are primarily dependent on speed and a number of fixed and predefined parameters for vehicles of weights ranging from 3.5 to 32 tonnes. For vehicles weighing less than 3.5 tonnes, the fuel consumption is estimated using a speed dependent regression function of the form $\epsilon = 0.0617v^2 - 7.8227v + 429.51$. For other classes of vehicles, MEET suggests the use a function the form:

$$\epsilon = K + av + bv^2 + cv^3 + d/v + e/v^2 + f/v^3, \tag{2.13}$$

where ϵ is the rate of emissions (g/km) for an unloaded goods vehicle on a road with a zero gradient, and v denotes the average speed of the vehicle (km/h). The parameter K and a–f are predefined coefficients whose values are given in Table 2.4 for different classes of vehicles.

TABLE 2.4: Emission parameters used in Model 5

Weight class	*K*	*a*	*b*	*c*	*d*	*e*	*f*
3.5 < Weight ≤ 7.5	110	0	0	0.000375	8702	0	0
7.5 < Weight ≤ 16	871	–16.0	0.143	0	0	32031	0
16 < Weight ≤ 32	765	–7.04	0	0.000632	8334	0	0
Weight > 32	1576	–17.6	0	0.00117	0	36067	0

Emission factors and functions suggested in the literature refer to standard testing conditions (i.e., zero road gradient, empty vehicle, etc.) and are typically calculated as a function of the average vehicle speed. Depending on the vehicle type, a number of corrections may be needed to allow for the effects of road gradient and vehicle load on the emissions, once a rough estimate has been produced. The following correction function is used to take the effect of road gradient into account:

$$GC = A_6v^6 + A_5v^5 + A_4v^3 + A_2v^2 + A_1v + A_0, \tag{2.14}$$

where *GC* is the road gradient correction factor. The coefficients A_0–A_6 used in calculating GC are provided in Table 2.5.

TABLE 2.5: Road gradient factor

Weight class	A_6	A_5	A_4	A_3	A_2	A_1	A_0	Slope
Weight ≤7.5	0	–3.01E–09	5.73E–07	–4.13E–05	1.13E–03	8.13E–03	9.14E–01	[0,4]
Weight ≤7.5	0	–1.39E–10	5.03E–08	–4.18E–06	1.95E–05	3.68E–03	9.69E–01	[–4,0]
7.5 < Weight ≤16	0	–9.78E–10	–2.01E–09	1.91E–05	–1.63E–03	5.91E–02	7.70E–01	[0,4]
7.5 < Weight ≤16	0	–6.04E–11	–2.36E–08	7.76E–06	–6.83E–04	1.79E–02	6.12E–01	[–4,0]
16 < Weight ≤32	0	–5.25E–09	9.93E–07	–6.74E–05	2.06E–03	–1.96E–02	1.45E+00	[0,4]
16 < Weight ≤32	0	–8.24E–11	2.91E–08	–2.58E–06	5.76E–05	–4.74E–03	8.55E–01	[–4,0]
Weight > 32	0	–2.04E–09	4.35E–07	–3.69E–05	1.69E–03	–3.16E–02	1.77E+00	[0,4]
Weight > 32	0	–1.10E–09	2.69E–07	–2.38E–05	9.51E–04	–2.24E–02	9.16E–01	[–4,0]

The following correction function is used to take the load factor into account:

$$LC = k + n\gamma + p\gamma^2 + q\gamma^3 + r/v + s/v^2 + t/v^3 + u/v, \tag{2.15}$$

where *LC* is the load correction factor, and k and n–u are coefficients whose values are presented in Table 2.6.

TABLE 2.6: Load correction factor

Weight class	k	n	p	q	r	s	t	u
Weight ≤7.5	1.27	0.0614	0	–0.00110	–0.00235	0	0	–1.33
7.5 < Weight ≤16	1.26	0.0790	0	–0.00109	0	0	–2.03E–7	–1.14
16 < Weight ≤32	1.27	0.0882	0	–0.00101	0	0	0	–0.483
Weight > 32	1.43	0.121	0	–0.00125	0	0	0	–0.916

MEET suggests estimating CO_2 emissions (g) as follows:

$$F = \epsilon \cdot GC \cdot LC \cdot Distance. \tag{2.16}$$

MEET is based on on-road measurements and all parameters are extracted from real-life experiments. The main deficiency of the model is its use of fixed vehicle-specific parameter settings for *any* vehicle in a given weight class.

2.2.6 Model 6: Computer programme to calculate emissions from road transportation (COPERT) model

The last model we review in this section is a *COmputer Programme to calculate Emissions from Road Transport* (*COPERT*) described by Ntziachristos and Samaras (2000). COPERT estimates emissions for all major air pollutants as well as greenhouse gases (i.e., CO_2) produced by different vehicle categories (e.g., passenger cars, light duty vehicles, heavy duty vehicles, mopeds and motorcycles). Similar to Model 5, COPERT uses a number of regression functions to estimate fuel consumption, which are specific to vehicles of different weights.

These regression functions are shown in Table 2.7.

TABLE 2.7: COPERT emission estimation functions

Weight class	Speed range (km/h)	Emission factor (g/km)
Weight ≤3.5	10–120	$0.0198v^2 - 2.506v + 137.42$
3.5 < Weight ≤7.5	0–47	$1425.2v^{-0.7593}$
3.5 < Weight ≤7.5	47–110	$0.0082v^2 - 0.0430v + 60.12$
7.5 < Weight ≤16	0–59	$1068.4v^{-0.4905}$
7.5 < Weight ≤16	59–110	$0.0126v^2 - 0.06589v + 141.18$
16 < Weight ≤32	0–59	$1595.1v^{-0.4744}$
16 < Weight ≤32	59–110	$0.0382v^2 - 5.1630v + 399.3$
Weight >32	0–58	$1855.7v^{-0.4367}$
Weight >32	58–110	$0.0765v^2 - 11.414v + 720.9$

COPERT is also based on on-road measurements, like Model 5. However, it does not take road gradient and acceleration into account. One interesting aspect of this model is its ability to differentiate between two different speed ranges for each vehicle class, as shown in Table 2.7.

2.2.7 A tabulated comparison

Fuel consumption depends on a number of factors which can be grouped into four categories: vehicle, driver, environmental conditions and traffic conditions, as identified by Ardekani et al. (1996). Using three of these four categories, we present in Table 2.8 a tabulated comparison of the six models reviewed so far. Driver related factors are difficult, if not impossible, to integrate into estimation models. The models listed and included in the comparisons are Models 1–3 by Bowyer et al. (1985), Model 4 by Scora and Barth (2006), Model 5 by Hickman et al. (1999), and Model 6 by Ntziachristos and Samaras (2000).

The tabulated comparison shows that all six models consider vehicle load, speed and acceleration, although the way in which they incorporate these factors in the approximation is highly varied, especially for vehicle load. Models 1–4 are similar in in their consideration of rather detailed and technical vehicle-specific parameters, such as vehicle shape (frontal area), and road conditions (e.g., gradient, surface resistance). This is not the case of Models 5 and 6 which present simpler estimations based on regressions through a predefined set of parameters for a number of vehicle classes. Model 5 is, to some extent, able to take into account factors of load and gradient through the correction factors, but this is not the case for Model 6.

TABLE 2.8: A comparison of Models 1–6 based on factors affecting fuel consumption

	Factors	Model 1	Model 2	Model 3	Model 4	Model 5	Model 6
Vehicle related	Total vehicle mass	×	×	×	×	×	×
	Engine size	×	×	×	×		
	Engine temperature					×	
	Oil viscosity				×		
	Gasoline type	×	×	×	×	×	×
	Vehicle shape	×	×	×	×		
	The degree of use of auxiliary electric devices				×		
Environment related	Roadway gradient	×	×	×	×	×	
	Wind conditions	×	×	×	×		
	Ambient temperature	×	×	×	×		
	Altitude	×	×	×	×		
	Pavement type	×	×	×	×		
	Surface conditions	×	×	×	×		
Traffic related	Speed	×	×	×	×	×	×
	Acceleration	×	×		×		

It is worth mentioning that none of the models listed in Table 2.8 explicitly considers driver-related factors or some vehicle related factors such as transmission type, tire pressure and so forth. However, quantifying such detailed factors is rather difficult and one should not expect any model to be able to fully incorporate these.

2.2.8 Other fuel consumption models

In this section, we review four other fuel consumption models identified in the literature. The reason why we separate these four models from the ones presented above is that the sources describing these models either do not provide enough details on the models themselves or on the parameters they require, hence they can not be included in the numerical comparisons due to lack of data availability.

One of these models was proposed by Everall (1968) and estimates emissions based on load and kinetic energy formulations, as shown below:

$$F = 0.0047(20E^{0.52})(1 + 40/v) + 0.0047M,$$

where F is the total fuel consumption (L), M denotes the load (kg) and E is the engine displacement (L). One other fuel consumption model which is yet an another extension of the

instantaneous model was proposed by Bowyer et al. (1985). This model relates fuel consumption per unit distance to average speed. It is only applicable to urban driving where the average travel speed is below 50 km/h. In this model

$$F_s = f_x x_s,$$

where the fuel consumption per unit distance is $f_x = f_i/v_s + cK$, and v_s denotes the average travel speed (km/h). Furthermore, $K = 1 - K_1(1 - M/1200) - K_2(1 - \beta_1/0.090) - K_3(1 - \beta_2/0.045 - K_4(1 - b_1/0.000278M) - K_5(1 - b_2/0.00108)$, K is the adjustment factor for different types of vehicles, c is the regression coefficient, K_1–K_5 are parameters based on the analysis of Sydney on-road data.

A physical emission rate estimator model introduced by Nam and Giannelli (2005) estimates emissions as

$$F = A + Bv + Cv^2,$$

where A, B, C are parameters calculated as $A = C_{R0}Mg$, $B = 0.0$, $C = (C_d A_r \rho)/2 + C_{R2}Mg$, and C_{R0} and C_{R2} are the zero and second order in speed rolling resistance force terms, respectively. The last model we review of this section was developed by Kirby (2006). It estimates F as

$$F = 3.6(k_1(1 + v_3/2v_m^3) + k_2 v)/v,$$

where v_m is the speed at which fuel consumption is optimal, v is the chosen constant speed (km/h). Furthermore, $k_1 = v_m^3(R_{90} - R_{120}/(v_m^3 - 113400))$, and $k_2 = (14580R_{120} - 25920R_{90} + 4v_m^3 R_{120} - 3v_m^3 R_{90})/36(v_m^3 - 11340)$. R_{90} and R_{120} are fuel consumption rates at speeds 90 km/h and 120 km/h, respectively.

2.3 Computational Experiments

We now present some experiments we have conducted to numerically compare and contrast Models 1–6 under different scenarios generated with varying values of some key parameters. We describe the settings of the four parameters used in the experiments in greater detail below.

2.3.1 Data generation and the experimental setting

In all the experiments, we assume a single vehicle driving on a road segment of 100 km, and vary a number of parameters including vehicle speed, load, acceleration and road gradient in the manner described below.

1. *Vehicle speed*: Different countries impose different restrictions on driving speed. In our experiments, the lower and upper speed limits are set to 20 km/h and 110 km/h, respectively.

2. *Vehicle load*: The gross vehicle weight rating (GVWR) is the maximum allowable total mass of a road vehicle or trailer when loaded, including the weight of the vehicle itself plus fuel, any passengers, cargo, and trailer weight. The commercial truck classification is usually based on the GVWR and uses eight classes. Classes 1 and 2 are referred to as "Light Duty", 3–5 as "Medium Duty", and 6–8 as "Heavy Duty". In this study, we consider a vehicle from each of these three groups. We define the *load factor* as the load carried by a vehicle expressed as a percentage of its empty weight. The load factors used for light duty vehicles are 0% (unloaded), 10% and 20%. The load factors used for medium duty vehicles are 0%, 15% and 30%. Finally, the load factors used for heavy duty vehicles are 0%, 30%, 60% and 90%.

3. *Acceleration*: The term acceleration is commonly used to express the rate at which speed increases. Conversely, the rate at which speed decreases is called deceleration. There are two types of acceleration: *(i) average acceleration* which denotes the change in velocity divided by the change in time, *(ii) instantaneous acceleration* which corresponds to the acceleration at a specific point in time. We consider the latter in our experiments.

4. *Road slope*: The gradient of a road has an effect of increasing or decreasing the resistance of a vehicle to traction, as the power employed during the driving operation determines the amount of fuel consumption. Road gradient factors are set to ± 0.57 and ± 1.15 degrees for the whole of the 100 km road segment.

Our experiments are based on a number of predefined scenarios generated by varying values of the four key parameters above. These scenarios are summarised in Table 2.9. For each of the scenarios 1–14, there are 10 possible speed values to choose from, ranging from 20 km/h to 110 km/h in increments of 10 km/h, as well as one from the three different types of vehicle (i.e., light, medium, heavy). This results in a total of $14 \times 10 \times 3 + 4 \times 3 = 432$ possible

scenarios. However, we only select representative samples from this set when presenting the results of analysis. In scenarios 15–18, vehicle speed is kept constant, either at 50 km/h or 70 km/h, but load is gradually changed from 0–10% in Scenarios 15 and 17 and from 0–30% in the other two.

TABLE 2.9: Setting of parameters in the 18 predefined scenarios

Scenario	Speed (km/h)	Load (kg)	Acceleration (km/h/s)	Road gradient (degrees)
1	20–110	0%	0	0
2	20–110	15%	0	0
3	20–110	30%	0	0
4	20–110	15%	0.01	0
5	20–110	15%	0.02	0
6	20–110	15%	–0.01	0
7	20–110	15%	–0.02	0
8	20–110	15%	0	0.57
9	20–110	15%	0	1.15
10	20–110	15%	0	–0.57
11	20–110	15%	0	–1.15
12	20–110	15%	0.01	0.57
13	20–110	15%	0.01	–0.57
14	20–110	15%	–0.01	0.57
15	50	0%–10%	0	0
16	50	0%–30%	0	0
17	70	0%–10%	0	0
18	70	0%–30%	0	0

Each model yields an estimation of fuel consumption measured in different units. Models 1–3 give fuel consumption in mL per time or distance. Model 4 estimates fuel consumption in gram fuel per time or distance, and Models 5 and 6 estimate CO_2 emissions in numbers of grams per distance. For comparison purposes, these outputs have all been converted to the estimated total fuel usage (in L) for the whole of the 100 km road segment. The parameters used in the experiments are those presented above, although at times, interpolation has been used to estimate the value of some parameters which were unavailable for certain load or speed profiles.

2.3.2 Results

We start by presenting results for the scenarios 1–14 in Table 2.9 for three different levels of speed: 50 km/h, 70 km/h and 100 km/h. The associated comparison results are presented in

Tables 2.10, 2.11 and Table 2.12, respectively. These tables provide, for each scenario, the total fuel consumption (L) estimated by each model.

TABLE 2.10: Fuel consumption with speed of 50 km/h for scenarios 1–14.

Scenario	Model 1	Model 2	Model 3	Model 4	Model 5	Model 6
1	19.79	16.52	31.90	15.44	10.35	10.65
2	22.08	18.24	35.53	18.14	13.15	10.65
3	25.08	20.34	39.88	21.74	17.39	21.28
4	37.80	11.58	40.33	19.62	16.42	14.49
5	47.62	17.04	43.37	21.52	19.98	16.81
6	19.40	17.44	34.89	18.77	13.68	11.30
7	21.74	17.80	35.53	17.79	13.15	10.65
8	28.09	23.94	40.94	24.27	13.78	10.65
9	34.09	29.64	46.34	30.41	14.37	10.65
10	16.08	14.35	32.32	12.00	12.52	10.65
11	10.07	10.46	29.11	5.87	11.92	10.65
12	43.81	15.24	45.73	25.76	17.21	14.49
13	31.79	8.20	37.12	13.48	15.64	14.49
14	25.41	23.21	40.30	24.90	14.34	11.30

TABLE 2.11: Fuel consumption with speed of 70 km/h for scenarios 1–14.

Scenario	Model 1	Model 2	Model 3	Model 4	Model 5	Model 6
1	28.64	17.06	34.63	15.95	11.35	13.20
2	32.23	18.86	38.61	18.40	14.42	13.20
3	36.96	21.09	43.38	21.63	18.06	21.28
4	48.21	15.98	43.60	21.33	19.98	17.00
5	51.26	19.68	44.51	22.04	21.03	17.69
6	25.93	16.41	36.73	17.88	13.44	11.67
7	26.06	17.36	36.81	17.63	13.38	11.76
8	38.23	24.63	44.02	24.54	15.11	13.20
9	44.24	30.40	49.42	30.67	15.76	13.20
10	26.22	14.52	35.40	12.26	13.73	13.20
11	20.22	10.18	32.19	6.13	13.07	13.20
12	54.22	20.37	49.00	27.47	20.93	17.00
13	42.21	11.44	40.39	15.20	19.02	17.00
14	31.94	22.29	42.14	24.01	14.08	11.67

From Tables 2.10–2.12, it can be seen that there is a considerable increase in fuel consumption with respect to the changes in vehicle speed. Model 1 is the most sensitive among all those tested. With this model, the difference in fuel requirements is approximately 146% when speed is increased from 50 km/h to 100 km/h. Models 2 and 4 show very similar results

TABLE 2.12: Fuel consumption with speed of 100 km/h for scenarios 1–14.

Scenario	Model 1	Model 2	Model 3	Model 4	Model 5	Model 6
1	48.78	21.72	41.14	19.92	17.89	18.70
2	55.16	24.04	45.85	22.51	22.72	18.70
3	63.65	26.91	51.50	25.86	27.75	27.31
4	59.92	24.54	47.29	23.67	24.87	19.79
5	58.73	24.54	46.86	23.55	24.22	19.47
6	47.06	19.97	43.34	20.76	19.45	16.81
7	41.94	19.71	41.78	19.70	17.68	15.62
8	61.17	29.86	51.25	28.65	23.81	18.70
9	67.17	35.69	56.66	34.78	24.85	18.70
10	49.16	19.29	42.64	16.38	21.63	18.70
11	43.15	14.55	39.43	10.24	20.60	18.70
12	65.92	30.10	52.70	29.80	26.06	19.79
13	53.91	19.30	44.08	17.53	23.67	19.79
14	53.06	25.98	48.75	26.89	20.38	16.81

for each of three speed levels. The models based on on-road measurements, Model 5 and 6 yield similar fuel consumption requirements in general.

Scenarios 1–3 show that fuel consumption depends on vehicle load. All models seem to be rather sensitive to changes in load and in acceleration. Models 1–4 are also very sensitive to changes in deceleration rates, but this is not so much the case for Model 5 and 6. Similar conclusions can be made for changes in road grade. In particular, all models, with the exception of Model 6, show an increase (decrease, respectively) in fuel consumption when there is an increase (decrease, respectively) in the road grade.

We present the results for the remaining scenarios 15–18 in Table 2.13.

TABLE 2.13: Fuel consumption for scenarios 15–18.

Scenario	Model 1	Model 2	Model 3	Model 4	Model 5	Model 6
15	20.49	17.05	33.03	16.26	12.87	10.65
16	22.05	18.19	35.42	18.11	14.57	14.90
17	29.75	16.87	35.88	16.70	14.12	13.21
18	32.20	18.07	38.50	18.37	15.57	16.44

In the following sections, we study the effects of vehicle type, weight, acceleration, deceleration, road gradient and resistance on the estimated fuel consumption. For this purpose, we only consider Model 2 (as representative of Models 1 and 3 which are therefore excluded) and Model 4. We also exclude Models 5 and 6 from the further analysis since the changes in these parameters affect the outputs of these models in the same way as they affect Models 2 and 4.

2.3.2.1 Effect of changes in vehicle type

This section presents results of experiments conducted for three types of vehicles, namely Light Duty (LD), Medium Duty (MD) and Heavy Duty (HD). For the experiment, we assume a 0% load factor, zero acceleration and a zero road gradient. The results are shown in Figures 2.1 and 2.2 for Models 2 and 4, respectively. Each figure shows, for the corresponding model, fuel consumption values (in L, on the *y*-axis) for varying speed values (on the *x*-axis) for the three types of vehicles.

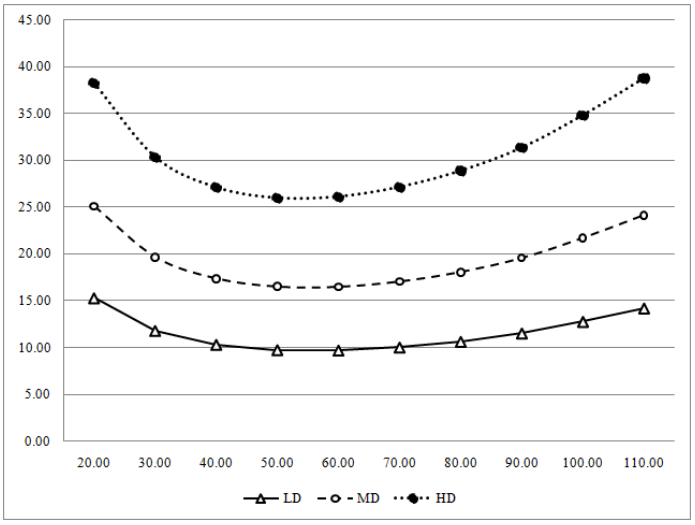


FIGURE 2.1: Total fuel consumption for three types of vehicles under different speed levels estimated by Model 2

Figures 2.1 and 2.2 show similar behaviors for different types of vehicles. For low speed values fuel consumption is very high because of the inefficiency in the usage of fuel. It decreases while speed goes up to a certain level, and then starts to increase because of the aerodynamic drag. Heavy vehicles consume significantly more fuel than the other two types, mainly due to their weight.

2.3.2.2 Effect of changes in vehicle weight

In this part of the analysis, we look at the effect of vehicle weight on fuel consumption for a medium duty vehicle. Light and heavy vehicles are not considered here since they exhibit similar patterns in terms of fuel consumption, with only the actual consumption values being

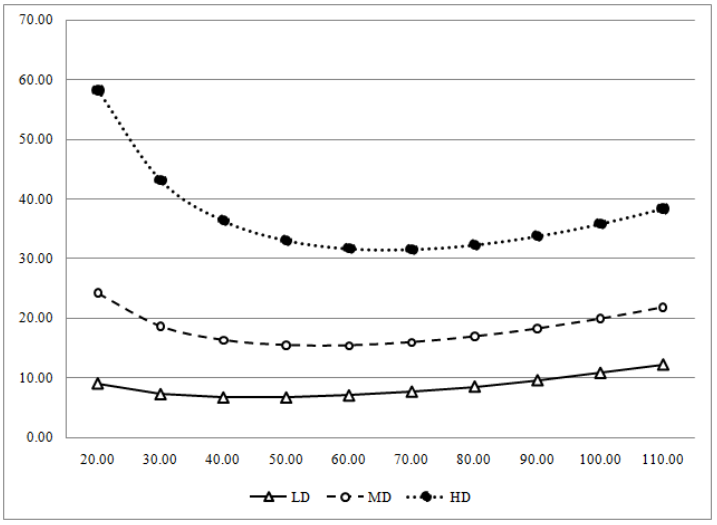


FIGURE 2.2: Total fuel consumption for three types of vehicles under different speed levels estimated by Model 4

different. Figures 2.3 and 2.4 show fuel consumption values for an unloaded, 15% and 30% loaded medium duty vehicle for models 1 and 4, respectively.

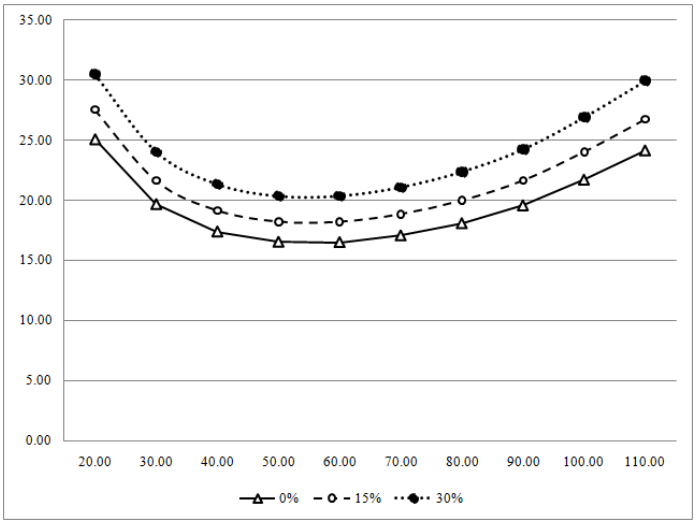


FIGURE 2.3: Total fuel consumption under various load profiles as estimated by Model 2

Figures 2.3 and 2.4 indicate that vehicle weight has a significant effect on fuel consumption and affects both models in similar ways. Model 4 is more sensitive respect to the changes

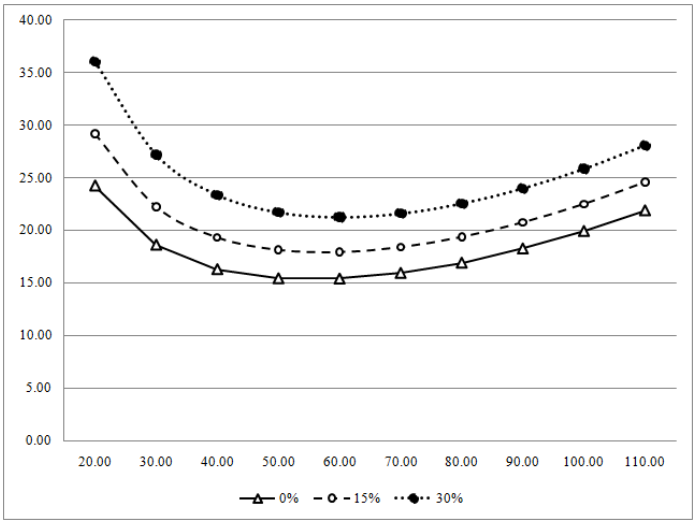


FIGURE 2.4: Total fuel consumption under various load profiles as estimated by Model 4

in load. Optimal vehicle speed turns out to be around 55 km/h for an unloaded medium duty vehicle using Model 2 and Model 4.

2.3.2.3 Effect of changes in acceleration and deceleration rates

In this part of the analysis, we investigate the effects of the changes in acceleration and deceleration on fuel requirements. For the former case, we allow the vehicle to accelerate at a rate of 0.01 m/s^2 over the 100 km road segment with a fixed initial speed. Initial speeds range from 20 km/h up to 100 km/h in increments of 10 km/h. For each initial speed, we assume that the vehicle accelerates up to 110 km/h using the specified rate. For deceleration, we consider a rate of –0.01 m/s^2. Initial speeds range from 30 km/h to 100 km/h, incremented in units of 10 km/h. For each initial speed, the vehicle is assumed to slow down to the speed of 20 km/h using the specified rate.

The results of this experiment are given in Figures 2.5 for acceleration, and in Figures 2.6 for deceleration, for a medium vehicle. The speed values on the x-axes of these figures are the starting speeds used in the experiments. The results shown in Figures 2.5 are unlike the ones presented earlier in that the fuel consumption does not exhibit a parabolic shape. This is partly explained by the fact that travel time decreases as speed increases. This model also shows that Model 2 is very sensitive to acceleration at relatively low levels of speed, whereas

Model 4 is not. Figure 2.6 shows that deceleration changes yield curves that are similar to those presented in the previous sections.

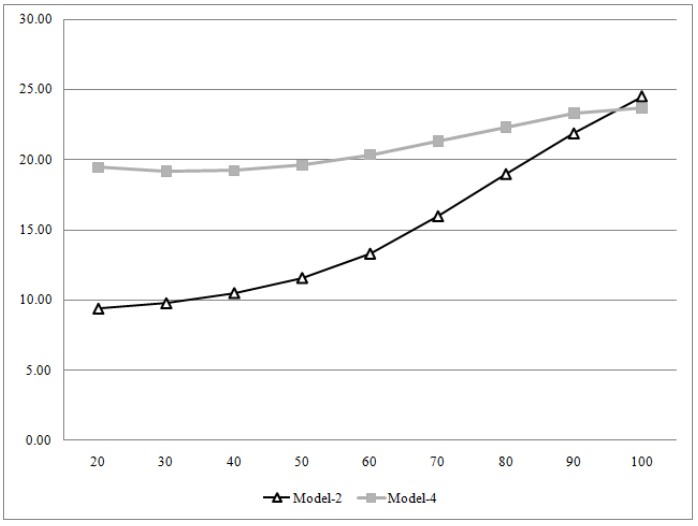


FIGURE 2.5: Total fuel consumption under a 0.01 m/s^2 acceleration as estimated by Model 2 and 4

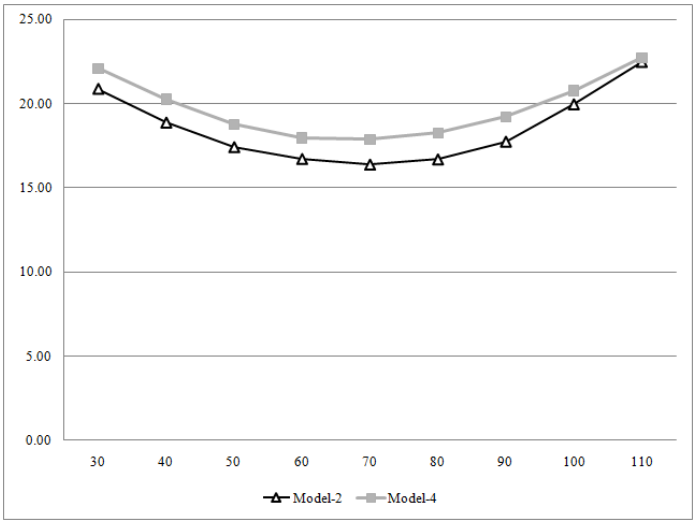


FIGURE 2.6: Total fuel consumption under a –0.01 m/s^2 deceleration as estimated by Model 2 and 4

2.3.2.4 Effect of changes in road gradient

Road gradient is another factor that affects fuel consumption. To test the significance of this effect, we consider two positive gradient values on the 100 km road segment as 0.57 and 1.15 degrees, and two negative values as –0.57 and –1.15 degrees. We assume that a medium duty vehicle travels on this road segment at a given average speed. Figures 2.7 and 2.8 show the results of this experiment obtained with Model 2. Figures 2.9 and 2.10 corresponds to the results obtained by Model 4.

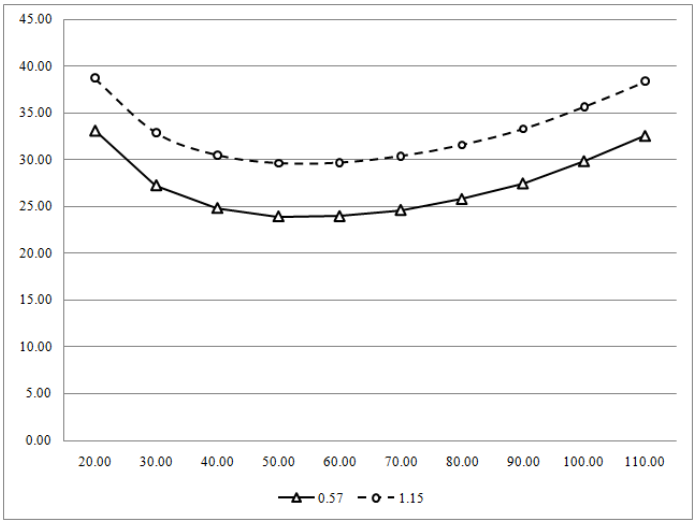


FIGURE 2.7: Effects of positive grades on total fuel consumption as estimated by Model 2

It can be seen from Figures 2.7 and 2.9 that a positive road gradient leads to an increased fuel consumption as compared to a negative road gradient. The figures also show that Model 4 is more sensitive to the changes in road gradient in the case of negative road gradients.

2.3.2.5 Effect of changes in resistance and drag

Rolling resistance, aerodynamic drag and road gradient resistance all influence the motion of the vehicle on the surface. The effects of the road gradient were analysed in the previous section. In this section, we look at how fuel consumption is affected by changes in resistance and drag coefficients.

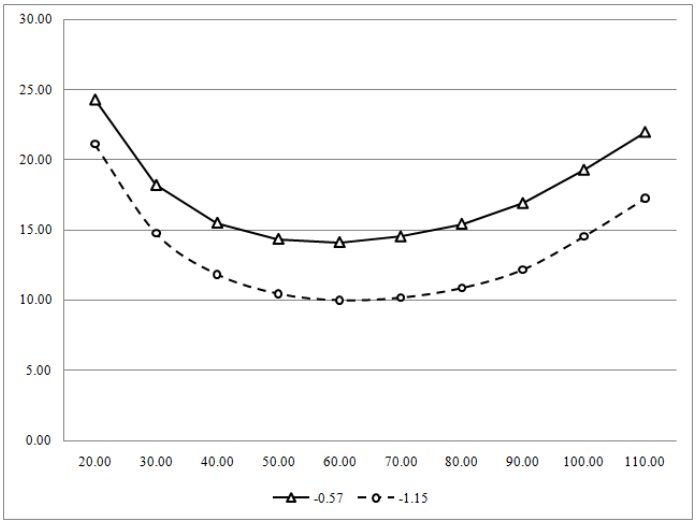


FIGURE 2.8: Effects of negative grades on total fuel consumption as estimated by Model 2

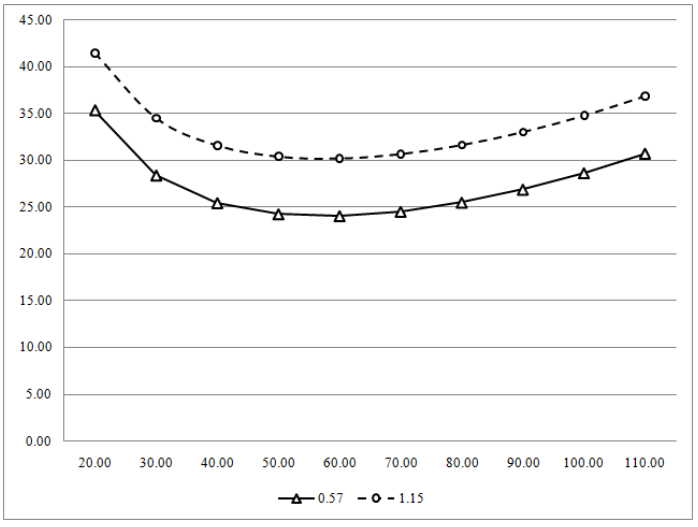


FIGURE 2.9: Effects of positive grades on total fuel consumption as estimated by Model 4

Rolling resistance occurs when a round object, such as a tire, rolls on a flat surface. It is responsible for over half of energy for the vehicle motion. The power required to overcome aerodynamic drag is higher at highway speeds. Aerodynamic drag is the force on an object that resists its motion through air. About one third of the energy produced by the engine of a good vehicle is used to overcome aerodynamic drag. The rest of the energy requirement is related to climbing. Rolling resistance ranges from 0.010 to 0.15, and aerodynamic drag

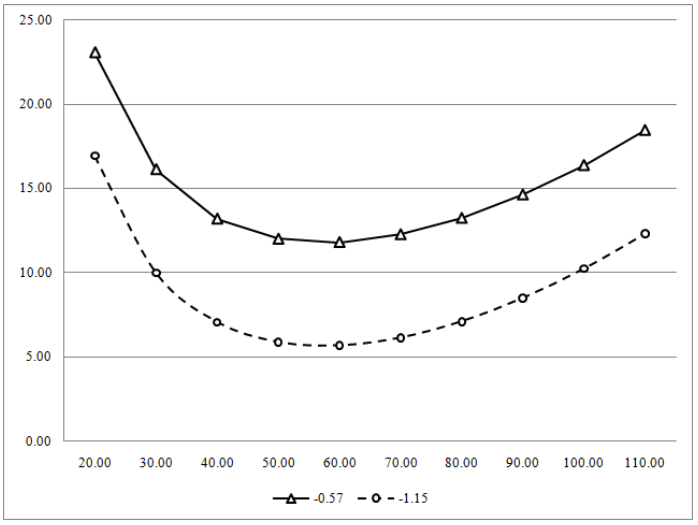


FIGURE 2.10: Effects of negative grades on total fuel consumption as estimated by Model 4

ranges from 0.6 to 0.8 (Genta, 1997). For the experiments of this section, the vehicle speed is set at 70 km/h and load is chosen as 15% of the empty weight of a medium duty vehicle. We present the results of the analysis for resistance in Table 2.14 obtained using Model 4. Model 2 does not allow a direct input of resistance and drag as parameters in the estimation and is therefore not included in our experiments.

TABLE 2.14: Effect of changes in rolling resistance and aerodynamic drag on fuel consumption (L).

Rolling resistance	Model 4	Aerodynamic drag	Model 4
0.010	18.40	0.60	17.57
0.011	19.02	0.65	17.99
0.012	19.63	0.70	18.40
0.013	20.24	0.75	18.82
0.014	20.86	0.80	19.23
0.015	21.47		

The results of the analysis indicate that resistance and drag both have significant effects on fuel consumption. In particular, if the rolling resistance goes down from 0.015 to 0.010 (i.e., by 33.3%), we can expect savings up to 14% in fuel consumption. Similarly, if the aerodynamic drag is reduced from 0.80 to 0.060 (25%), we can expect to achieve a saving around 8.6% in fuel consumption.

2.3.3 Comparison with on-road fuel consumption measurement data

Measurements of on-road fuel consumption of vehicles are typically carried out using various methods, such as engine and chassis dynamometer tests, tunnel studies, remote sensoring and on-board instrumentation readings. In this section of the analysis, we compare the results of Models 1–6 with measurements carried out by Erlandsson et al. (2008), who report results of on-road measurements of three heavy good vehicles weighing 15 t, 50 t and 60 t, representing three classes. These vehicles are equipped with the same basic engine, tested in their normal operation and driven by the same driver. The vehicles are certified according to emission requirements for Euro IV diesel. The results are given as the average of three test runs. In these experiments, average speeds are set to 38.8, 64.2 and 53.7 km/h for the three vehicles. The on-road experiment data, for each type of vehicle, are collected over 82 km/92 minutes, 55km/51 minutes, and 13km/20 minutes. Fuel consumption for each scenario is a projection on 100km.

Table 2.15 shows the results obtained by Models 1–6 in absolute terms (L) as well as the percentage difference from on-road fuel consumption measurements reported by Erlandsson et al. (2008).

TABLE 2.15: Comparison of the fuel consumption (L) measured by the six models with on-road fuel consumption: consumption values and percentage difference

On road	Vehicle weight (kg)	Average speed	Model 1	Model 2	Model 3	Model 4	Model 5	Model 6
30.3	15,000	38.8	37.73 (25%)	32.26 (6%)	51.12 (69%)	34.93 (15%)	19.18 (–37%)	24.10 (–21%)
43.6	50,000	64.2	76.58 (76%)	65.75 (51%)	85.61 (96%)	61.73 (42%)	33.85 (–22%)	41.17 (–6%)
53.0	60,000	53.7	61.42 (16%)	73.75 (39%)	96.27 (82%)	70.79 (34%)	36.44 (–31%)	44.21 (–17%)

The results presented in Table 2.15 are straightforward to interpret. There are rather large discrepancies between the results yielded by the models and those of the on-road experiments. Model 4 seems to provide the best estimation for a vehicle with weight of around 15000 kg. However, for heavier vehicles Model 6 yields better estimations. Models 5 and 6 underestimate emissions for this particular data set in all cases, whereas the remaining models overestimate them. A noteworthy case is Model 3 which provides results that are quite far off from the actual on-road measurements.

Schittler (2003) provides the average fuel consumption of a Class 8 vehicle (approximately 15 t) in Europe as 32.5 L/100 km. This figure is similar to the ones found by Model 2 and Model 4 as presented in Table 2.15.

2.4 Conclusions

This chapter has presented a review of ten emissions models found in the literature. It has also presented and analysed the results of extensive computational experiments conducted on six models by varying parameters such as vehicle load, speed, acceleration and road grade. The results showed that all models tested here are sensitive to changes in load, speed and acceleration, although the degree of the sensitivity changes from one model to another. Some models are not at all effected by changes in deceleration and road grade, whereas some others remain relatively insensitive.

Due to lack of availability of sufficient on-road measurement data, it was not possible to provide conclusive evidence to suggest a "one-fits-all" model to use for fuel estimations. However, benchmarks with limited on-road measurements taken from the literature show that most models tested in this chapter produce fuel consumption estimates that are far from those obtained through on-road experiments. Our results indicate that comprehensive modal emission model (Model 4) and a regression based model (Model 6) are able to provide relatively good estimations for a number of heavy-good vehicles. Further on-road measurement data are required to provide more conclusive evidence on which model is best to estimate fuel consumption.

Chapter 3

An Adaptive Large Neighbourhood Search Heuristic for the Pollution-Routing Problem

Abstract

The Pollution-Routing problem (PRP) is a recently introduced extension of the classical Vehicle Routing Problem with Time Windows which consists of routing a number of vehicles to serve a set of customers, and determining their speed on each route segment so as to minimise a function comprising fuel, emission and driver costs. This chapter presents an Adaptive Large Neighbourhood Search for the PRP. Results of extensive computational experimentation confirm the efficiency of the algorithm.

3.1 Introduction

The road transportation sector is a significant emitter of carbon dioxide (CO_2), the amount of which is directly proportional to fuel consumption (Kirby et al., 2000). Fuel consumption is dependent on a variety of parameters, such as vehicle speed, load and acceleration (Demir et al., 2011). The Pollution-Routing Problem (PRP) is an extension of the classical Vehicle Routing Problem with Time Windows (VRPTW). It consists of routing a number of vehicles to serve a set of customers within preset time windows, and determining their speed on each route segment, so as to minimise a function comprising fuel, emission and driver costs. The PRP was introduced by Bektaş and Laporte (2011) who proposed a non-linear mixed integer mathematical model for the problem, which could be linearised. However, solving even medium scale PRP instances to optimality using such a model remains a challenge.

In this chapter, we propose an extended Adaptive Large Neighbourhood Search (ALNS) algorithm for the PRP. The algorithm integrates the classical ALNS scheme (Pisinger and Ropke, 2005, 2007; Ropke and Pisinger, 2006*a*) with a specialised speed optimisation algorithm which computes optimal speeds on a given path so as to minimise fuel consumption, emissions and driver costs. The latter algorithm can also be used as a stand-alone routine to optimise speeds for the VRPTW. The remainder of this chapter is organised as follows. In Section 3.2, we present the Pollution-Routing Problem. Section 3.3 describes a new iterative heuristic algorithm for the PRP. Section 3.4 presents the results of extensive numerical experiments. Conclusions are stated in Section 3.5.

3.2 Mathematical Model for the Pollution-Routing Problem

We first formulate the PRP and discuss the extensions of the problem.

3.2.1 Description of the Pollution-Routing Problem

The PRP is defined on a complete directed graph $\mathcal{G} = (\mathcal{N}, \mathcal{A})$ where $\mathcal{N} = \{0, ..., n\}$ is the set of nodes, 0 is a depot and $\mathcal{A} = \{(i, j) : i, j \in \mathcal{N} \text{ and } i \neq j\}$ is the set of arcs. The distance from i to j is denoted by d_{ij}. A fixed-size fleet of vehicles denoted by the set $\mathcal{K} = \{1, ..., m\}$ is available, and each vehicle has capacity Q. The set $\mathcal{N}_0 = \mathcal{N} \backslash \{0\}$ is a customer set, and each customer $i \in \mathcal{N}_0$ has a non-negative demand q_i as well as a time interval $[a_i, b_i]$ for service. Early arrivals are permitted but the vehicle has to wait until time a_i before service can start. The service time of customer i is denoted by t_i.

3.2.2 Fuel and CO_2 emissions

The PRP is based on the *comprehensive emissions model* described by Barth et al. (2005), Scora and Barth (2006), and Barth and Boriboonsomsin (2008), which is an instantaneous model estimating fuel consumption for a given time instant. According to this model, the fuel rate is given by

$$FR = \xi(kNV + P/\eta)/\kappa, \tag{3.1}$$

where ξ is fuel-to-air mass ratio, k is the engine friction factor, N is the engine speed, V is the engine displacement, and η and κ are constants. The variable P is the second-by-second engine power output (in kW), and can be calculated as

$$P = P_{tract}/\eta_{tf} + P_{acc}, \tag{3.2}$$

where η_{tf} is the vehicle drive train efficiency, and P_{acc} is the engine power demand associated with running losses of the engine and the operation of vehicle accessories such as air conditioning. P_{acc} is assumed to be zero. The parameter P_{tract} is the total tractive power requirements (in kW) placed on the wheels:

$$P_{tract} = (M\tau + Mg\sin\theta + 0.5C_d\rho Av^2 + MgC_r\cos\theta)v/1000, \tag{3.3}$$

where M is the total vehicle weight (kg), v is the vehicle speed (m/s), τ is the acceleration (m/s^2), θ is the road angle, g is the gravitational constant, and C_d and C_r are the coefficient of

the aerodynamic drag and rolling resistance, respectively. Finally, ρ is the air density and A is the frontal surface area of the vehicle.

For a given arc (i, j) of length d, let v be the speed of a vehicle speed traversing this arc. If all variables in FR except for the vehicle speed v remain constant on arc (i, j), the fuel consumption (in L) on this arc can be calculated as

$$F(v) = kNV\lambda d/v \tag{3.4}$$

$$+ P\lambda\gamma d/v, \tag{3.5}$$

where $\lambda = \xi/\kappa\psi$ and $\gamma = 1/1000n_{tf}\eta$ are constants and ψ is the conversion factor of fuel from gram/second to liter/second. Furthermore, let M be the load carried between nodes i and j. More specifically, $M = w + f$, where w is the curb weight (i.e., the weight of an empty vehicle) and f is the vehicle load. Let $\alpha = \tau + g\sin\theta + gC_r\cos\theta$ be a vehicle-arc specific constant and $\beta = 0.5C_d\rho A$ be a vehicle-specific constant. We omit the indices (i, j) on the variables v, d, f, and α to simplify the presentation. Then, $F(v)$ can be rewritten as

$$F(v) = \lambda\left(kNV + w\gamma\alpha v + \gamma\alpha f v + \beta\gamma v^3\right)d/v. \tag{3.6}$$

All other parameters and values are given in Table 3.1. The cost of fuel and CO_2 emissions per second can be calculated as $f_c FR/\psi$, where f_c is the unit cost of fuel and CO_2 emissions. Applying equation (3.6) to a low-duty vehicle for speeds varying from 20 km/h to 110 km/h for a road segment of $d = 100$ km yields the fuel consumption curve shown in Figure 3.1. The function depicted in Figure 3.1 is the sum of two components, one induced by (3.4) and shown by the dashed line, and the other by (3.5) shown by the dotted line. One can see that the contribution of the first component of the function, namely kNV, will only be significant for low speed levels (less than 40 km/h), whereas that of P_{tract} is significant for higher speed levels. In the PRP model, Bektaş and Laporte (2011) consider speeds of 40 km/h and higher for which they only make use of P_{tract}. In this work, we will allow for lower speeds which yield higher fuel consumptions. This is accounted for by the kNV component of equation (3.1) which we will incorporate in our model.

One relevant study to the one considered here is by Jabali et al. (2012a), who describe a VRP that considers travel time, fuel, and CO_2 emissions costs in a time-dependent context, where the latter are estimated using emission functions provided in the MEET report (Hickman et al., 1999). The authors describe a Tabu Search algorithm to solve the problem and show, through computational experiments, that limiting vehicle speeds is effective in reducing emissions to a certain extent although costly in terms of total travel time.

TABLE 3.1: Parameters used in the PRP model

Notation	Description	Typical values
w	curb-weight (kg)	6350
ξ	fuel-to-air mass ratio	1
k	engine friction factor (kJ/rev/litre)	0.2
N	engine speed (rev/s)	33
V	engine displacement (litres)	5
g	gravitational constant (m/s^2)	9.81
C_d	coefficient of aerodynamic drag	0.7
ρ	air density (kg/m^3)	1.2041
A	frontal surface area (m^2)	3.912
C_r	coefficient of rolling resistance	0.01
n_{tf}	vehicle drive train efficiency	0.4
η	efficiency parameter for diesel engines	0.9
f_c	fuel and CO_2 emissions cost per litre (£)	1.4
f_d	driver wage per (£/s)	0.0022
κ	heating value of a typical diesel fuel (kJ/g)	44
ψ	conversion factor (g/s to L/s)	737
v^l	lower speed limit (m/s)	5.5 (or 20 km/h)
v^u	upper speed limit (m/s)	25 (or 90 km/h)

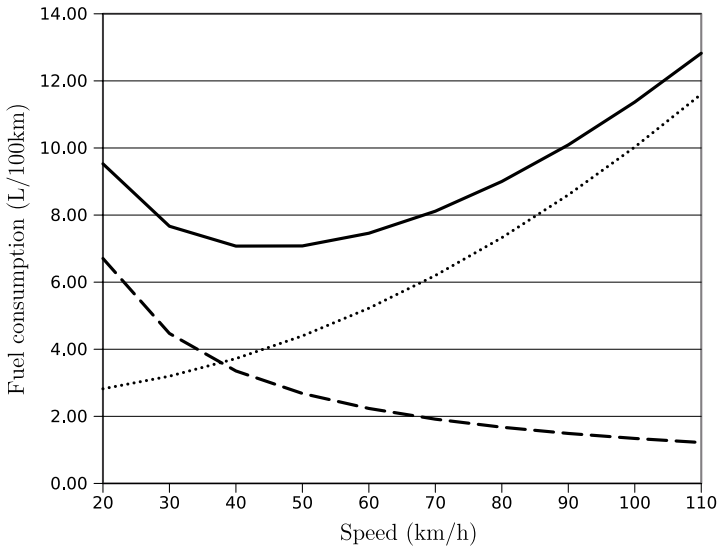


FIGURE 3.1: Fuel consumption as a function of speed (Bektaş and Laporte, 2011)

3.2.3 Integer programming formulation

We now present the integer programming formulation for the PRP. The model works with a discretised speed function defined by R non-decreasing speed levels $\bar{v}^r$ $(r = 1, ..., R)$. Binary variables x_{ij} are equal to 1 if and only if arc (i, j) appears in solution. Continuous variables f_{ij} represent the total amount of flow on each arc $(i, j) \in \mathcal{A}$. Continuous variables y_j represent the time at which service starts at node $j \in \mathcal{N}_0$. Moreover, s_j represents the total time spent on a route that has a node $j \in \mathcal{N}_0$ as last visited before returning to the depot. Finally, binary variables z_{ij}^r indicate whether or not arc $(i, j) \in \mathcal{A}$ is traversed at a speed level r. An integer linear programming formulation of the PRP is shown below:

$$\text{Minimise} \quad \sum_{(i,j)\in\mathcal{A}} kNV\lambda d_{ij} \sum_{r=1}^{R} z_{ij}^r / \bar{v}^r \tag{3.7}$$

$$+ \sum_{(i,j)\in\mathcal{A}} w\gamma\lambda\alpha_{ij} d_{ij} x_{ij} \tag{3.8}$$

$$+ \sum_{(i,j)\in\mathcal{A}} \gamma\lambda\alpha_{ij} d_{ij} f_{ij} \tag{3.9}$$

$$+ \sum_{(i,j)\in\mathcal{A}} \beta\gamma\lambda d_{ij} \sum_{r=1}^{R} z_{ij}^r (\bar{v}^r)^2 \tag{3.10}$$

$$+ \sum_{j\in\mathcal{N}_0} f_d s_j \tag{3.11}$$

subject to

$$\sum_{j \in \mathcal{N}} x_{0j} = m \tag{3.12}$$

$$\sum_{j \in \mathcal{N}} x_{ij} = 1 \qquad \forall i \in \mathcal{N}_0 \tag{3.13}$$

$$\sum_{i \in \mathcal{N}} x_{ij} = 1 \qquad \forall j \in \mathcal{N}_0 \tag{3.14}$$

$$\sum_{j \in \mathcal{N}} f_{ji} - \sum_{j \in \mathcal{N}} f_{ij} = q_i \qquad \forall i \in \mathcal{N}_0 \tag{3.15}$$

$$q_j x_{ij} \leq f_{ij} \leq (Q - q_i) x_{ij} \qquad \forall (i,j) \in \mathcal{A} \tag{3.16}$$

$$y_i - y_j + t_i + \sum_{r \in \mathcal{R}} d_{ij} z_{ij}^r / \bar{v}^r \leq K_{ij}(1 - x_{ij}) \qquad \forall i \in \mathcal{N}, j \in \mathcal{N}_0, i \neq j \tag{3.17}$$

$$a_i \leq y_i \leq b_i \qquad \forall i \in \mathcal{N}_0 \tag{3.18}$$

$$y_j + t_j - s_j + \sum_{r \in \mathcal{R}} d_{j0} z_{j0}^r / \bar{v}^r \leq L(1 - x_{j0}) \qquad \forall j \in \mathcal{N}_0 \tag{3.19}$$

$$\sum_{r=1}^{R} z_{ij}^r = x_{ij} \qquad \forall (i,j) \in \mathcal{A} \tag{3.20}$$

$$x_{ij} \in \{0,1\} \qquad \forall (i,j) \in \mathcal{A} \tag{3.21}$$

$$f_{ij} \geq 0 \qquad \forall (i,j) \in \mathcal{A} \tag{3.22}$$

$$y_i \geq 0 \qquad \forall i \in \mathcal{N}_0 \tag{3.23}$$

$$z_{ij}^r \in \{0,1\} \qquad \forall (i,j) \in \mathcal{A}, r = 1, ..., R. \tag{3.24}$$

This mathematical formulation of the PRP presented here is an extension of the one presented in Bektaş and Laporte (2011) to take into account for speeds 40 km/h or lower through the term (3.7) of the objective function. The objective function (3.7)–(3.10) is derived from (3.6). The terms (3.8) and (3.9) calculate the cost incurred by the vehicle curb weight and payload. Finally, the term (3.11) measures the total driver wages.

Constraints (3.12) state that each vehicle must leave the the depot. Constraints (3.13) and (3.14) are the degree constraints which ensure that each customer is visited exactly once. Constraints (3.15) and (3.16) define the arc flows. Constraints (3.17)–(3.19), where $K_{ij} = \max\{0, b_i + s_i + d_{ij}/l_{ij} - a_j\}$, and L is a large number, enforce the time window restrictions. Constraints (3.20) ensure that only one speed level is selected for each arc and $z_{ij}^r = 1$ if $x_{ij} = 1$.

The PRP is NP-hard since it is an extension of the classical Vehicle Routing Problem (VRP). Bektaş and Laporte (2011) have shown that a simplified version of this problem cannot be

solved to optimality for instances with more than 10 customers. To deal with larger-size instances, we have developed a heuristic to obtain good-quality solutions within short computational times, which is explained in the following section.

3.3 An Adaptive Large Neighbourhood Heuristic Algorithm for the PRP

The heuristic operates in two stages. In the first stage, it solves a VRPTW using ALNS. This metaheuristic is an extension of the Large Neighbourhood Search (LNS) heuristic first proposed by Shaw (1998), and based on the idea of gradually improving an initial solution by using both destroy and repair operators. In other words, LNS consists of a series of removal and insertion moves. If the new solution is better than the current best solution, it replaces it and use as an input to the next iteration. The LNS heuristic can be embedded within any local search heuristic such as simulated annealing or tabu search.

In the second stage, a speed optimisation algorithm (SOA) is run on the resulting VRPTW solution. Given a vehicle route, the SOA consists of finding the optimal speed on each arc of the route in order to minimise an objective function comprising fuel consumption costs and driver wages.

The proposed algorithm is designed as an iterative process whereby the ALNS uses fixed speeds as inputs to the VRPTW, following which the SOA is run on each route to improve the solution.

3.3.1 Adaptive large neighbourhood search

The ALNS heuristic framework was put forward by Pisinger and Ropke (2005, 2007); Ropke and Pisinger (2006*a*) to solve variants of the vehicle routing problem. Rather than using one large neighbourhood as in LNS, it applies several removal and insertion operators to a given solution. The neighbourhood of a solution is obtained by removing some customers from the solution and reinserting them as in Milthers (2009). The removal and insertion operators are selected dynamically according to their past performance. To this end, each operator is assigned a *score* which is increased whenever it improves the current solution. The new solution is accepted if it satisfies some criteria defined by the local search framework (e.g., simulated annealing) applied at the outer level. The graphical representation of the ALNS is

given in Figure 3.2. The main features of the ALNS algorithm will be described in detail below.

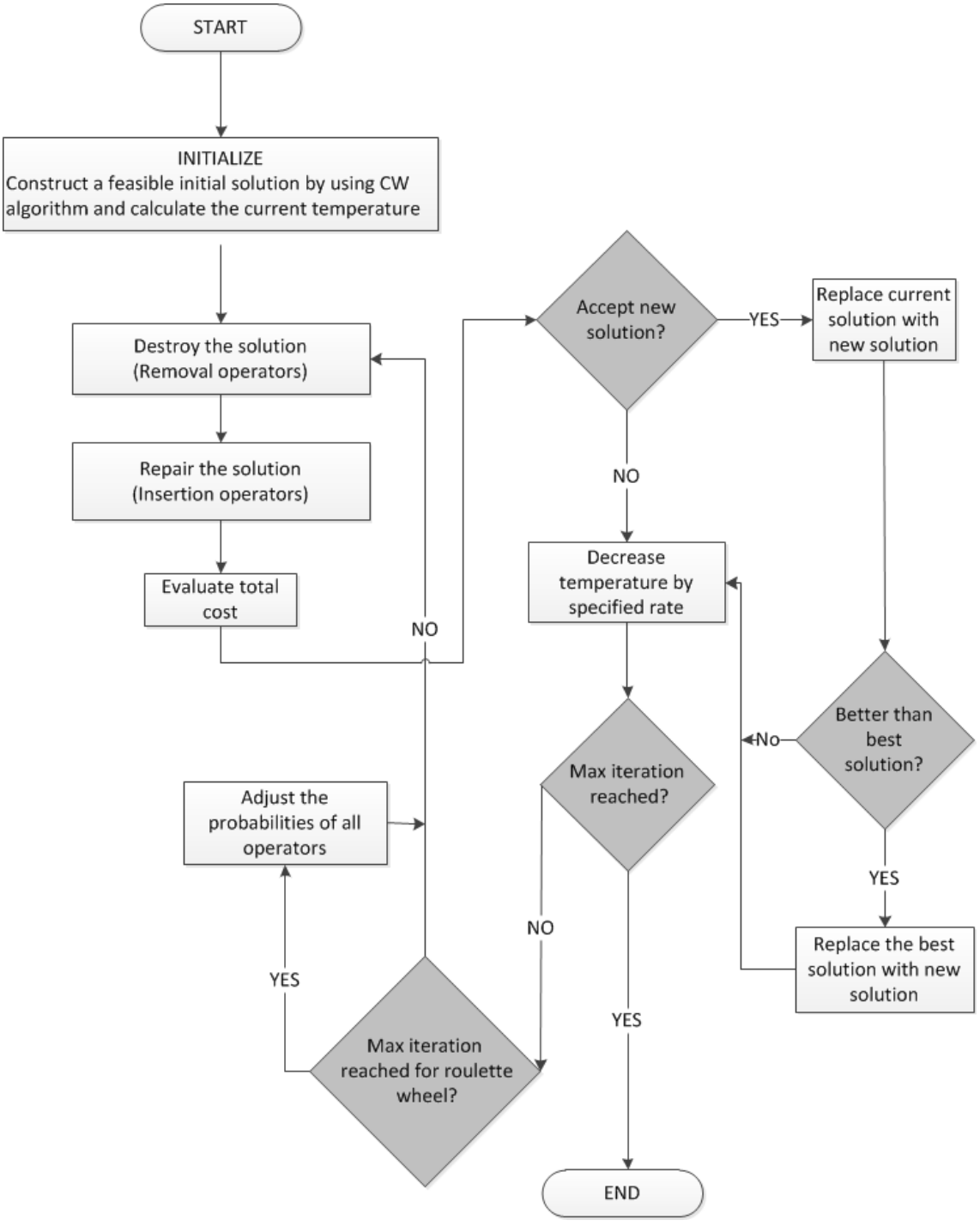


FIGURE 3.2: The framework of the ALNS

3.3.1.1 Initialisation

Several heuristic methods can be used to quickly obtain a feasible solution for the VRP. Cordeau et al. (2002) have analysed and reviewed some of the classical heuristics based on four different criteria: accuracy, speed, simplicity, and flexibility. This comparison shows that the Clarke and Wright (1964) heuristic has the distinct advantage of being very quick and simple to implement. We have therefore used it in our algorithm to obtain an initial solution. It is noteworthy that while additional constraints incorporated into the CW heuristic "usually results in a sharp deterioration" (Cordeau et al., 2002), the quality of the initial solution is not so important since as a rule ALNS can easily recover from a poor initial solution. This algorithm was implemented while maintaining the feasibility of capacity and time window constraints.

3.3.1.2 Adaptive weight adjustment procedure

The selection of the removal and insertion operators is regulated by a roulette-wheel mechanism. Initially, all removal or insertion operators are equally likely. Thus, for the twelve removal and five insertion operators, the probabilities are initially set to $1/12$ and $1/5$, respectively. During the algorithm, they are updated as $P_d^{t+1} = P_d^t(1-r_p) + r_p\pi_i/\omega_i$, where r_p is the roulette wheel parameter, π_i is the score of operator i and ω_i is the number of times it was used during the last N_w iterations. The score of each operator measures how well the operator has performed at each iteration. If a new best solution is found, the score of an operator is increased by σ_1. If the solution is better than the current solution, the score is increased by σ_2. If the solution is worse than the current solution but accepted, the score is increased by σ_3.

3.3.1.3 Removal operators

We now present the twelve removal operators used in our algorithm. The first nine are either adapted or inspired by Pisinger and Ropke (2005, 2007); Ropke and Pisinger (2006*a*,*b*) and Shaw (1998), whereas the last three are new. The destroy phase mainly consists of removing s customers from the current solution and adding them into a *removal list* $\mathcal{L}$ as illustrated in Figure 3.3.

A pseudo-code of the generic removal procedure is presented in Algorithm 1. The algorithm is initialised with a feasible solution X as input and returns a partially destroyed solution. The parameter ϕ defines the number of iterations of the search. In Algorithm 1, a chosen operator

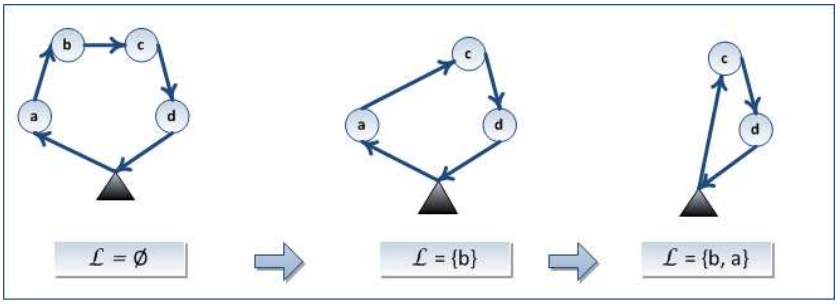


FIGURE 3.3

is used to remove a subset S of nodes from the solution. These nodes are inserted in a removal list $\mathcal{L}$. The algorithm iterates in a similar fashion for ϕ iterations.

Algorithm 1: The generic structure of the removal procedure

input : A feasible solution *X*, and maximal number of iterations ϕ
output: A partially destroyed solution X_p

1 *Initialise removal list ($\mathcal{L} = \emptyset$)*
2 **for** *ϕ iterations* **do**
3 *Apply remove operator to find the set S of nodes for removal*
4 $\mathcal{L} \leftarrow \mathcal{L} \cup S$
5 *Remove the subset S of nodes from X*

We now describe the removal operators used in our implementation:

1. **Random removal (RR)**: This operator starts with an empty removal list. It randomly removes s nodes from the solution, and runs for $\phi = s$ iterations. The idea of randomly selecting nodes helps diversify the search mechanism. The worst-case time complexity of the RR operator is $O(n)$.

2. **Worst-distance removal (WDR)**: This operator iteratively removes high cost customers, where the cost is defined as the sum of distances from the preceding and following customer on the tour, i.e., it removes node $j^* = \underset{j \in \mathcal{N}}{\operatorname{argmax}}\{|d_{ij} + d_{jk}|\}$. The worst-case time complexity of the WDR operator is $O(n^2)$.

3. **Worst-time removal (WTR)**: This operator calculates, for each node j, the deviation of service start time from time a_j, and then removes the node with the largest deviation. The idea is to prevent long waits or delayed service start times. The algorithm starts with an empty removal list, and runs for $\phi = s^2$ iterations (for $i = 1, ..., s$; $j = 1, ..., s$). The operator selects $j^* = \underset{j \in \mathcal{N}}{\operatorname{argmax}}\{|y_j - a_j|\}$ where y_j is the time at which service begins

at node j. The solution is updated after each node removal. The worst-case time complexity of the WTR operator is $O(n^2)$.

4. **Route removal (RoR)**: This operator removes a full route from the solution. It randomly selects a route from the set of routes in the solution. The remove operator then repeatedly selects a node j from this route until all nodes are removed. The RoR operator can be implemented in $O(1)$ worst-case time.

5. **Shaw removal (SR)**: The aim of this operator is to remove a set of customers that are related in a predefined way and therefore are easy to change. The logic behind the operator was introduced by Shaw (1998). The algorithm starts by randomly selecting a node i and adds it to the removal list. Let $l_{ij} = -1$ if $i \in \mathcal{N}$ and $j \in \mathcal{N}$ are in the same route, and 1 otherwise. The operator selects a node $j^* = \underset{j \in \mathcal{N}}{\operatorname{argmin}}\{\Phi_1 d_{ij} + \Phi_2 |a_i - a_j| + \Phi_3 l_{ij} + \Phi_4 |q_i - q_j|\}$, where Φ_1–Φ_4 are weights which are normalized to find the best candidate from solution. The operator is applied $\phi = s^2$ times by selecting a node not yet in the removal list which is most similar to the one last added to the list. The worst-case time complexity of the SR operator is $O(n^2)$.

6. **Proximity-based removal (PR)**: The operator removes a set of nodes that are related in terms of distance. This operator is a special case of the Shaw removal operator with $\Phi_1 = 1$, and $\Phi_2 = \Phi_3 = \Phi_4 = 0$. The way the operator works is graphically illustrated in Figure 3.4. The worst-case time complexity of the PR operator is $O(n^2)$.

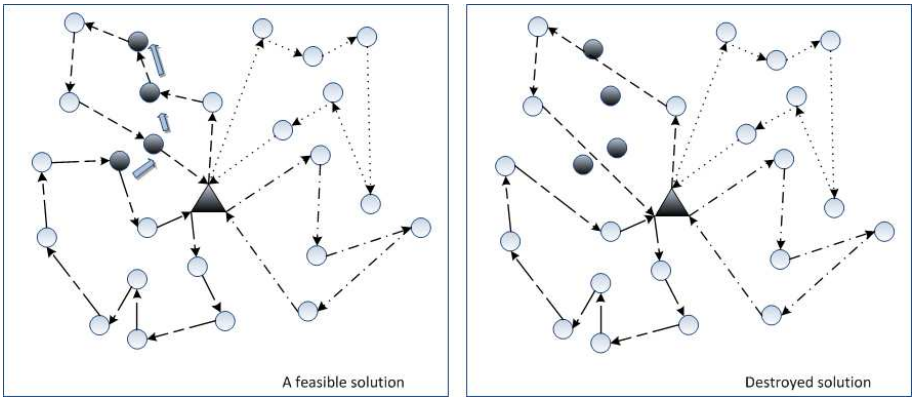


FIGURE 3.4: Proximity-based removal operator

7. **Time-based removal (TR)**: The operator is a special case of the Shaw removal with $\Phi_2 = 1$, and $\Phi_1 = \Phi_3 = \Phi_4 = 0$. The worst-case time complexity of the TR operator is $O(n^2)$.

8. **Demand-based removal (DR)**: This operator is a special case of the Shaw removal with $\Phi_4 = 1$, and $\Phi_1 = \Phi_2 = \Phi_3 = 0$. The worst-case time complexity of the DR operator is $O(n^2)$.

9. **Historical knowledge node removal (HR)**: The HR operator is similar to the neighbour graph removal operator used in Ropke and Pisinger (2006*b*). This operator keeps a record of the position cost of every node i, defined as the sum of the distances between its preceding and following nodes, and calculated as $d_i = d_{i-1,i} + d_{i,i+1}$ at every iteration of the algorithm. At any point in the algorithm, the best position cost d_i^* of node i is updated to be the minimum of all d_i values calculated until that point. The HR operator then picks a node j^* on a route with maximum deviation from its best position cost, i.e., $j^* = \underset{j \in \mathcal{N}_0}{\operatorname{argmax}}\{d_j - d_j^*\}$. Node j^* is then added to the removal list. The worst-case time complexity of the HR operator is $O(n)$.

10. **Neighbourhood removal (NR)**: This operator is based on the idea of removing nodes from routes which are extreme with respect to the average distance of a route. More specifically, in a given solution with a set of routes $\mathcal{B}$, the operator calculates, for each route $B = \{i_1, ..., i_{|B|}\}$ in $\mathcal{B}$ an average distance as $\bar{d}_B = \sum_{(i_1,i_2)\in B} d_{i_1 i_2}/|B|$ and selects a node $j^* = \underset{B\in\mathcal{B}; j\in B}{\operatorname{argmax}}\{\bar{d}_B - d_{B\setminus\{j\}}\}$. The worst-case time complexity of the NR operator is $O(n^2)$.

11. **Zone removal (ZR)**: The zone removal operator is based on removal of nodes in a predefined area in the Cartesian coordinate system in which nodes are located. The operator first compute the corner points of the area. The whole region is then split up into smaller areas. An area is randomly selected and all its node are removed. The removal operator selects $S = \{j^* | x(i^1) \leq x(j^*) \leq x(i^2)$ and $y(i^1) \leq y(j^*) \leq y(i^2)\}$, where $(x(i^1), x(i^2))$ are the *x*-coordinates of the selected zone i, and $(y(i^1), y(i^2))$ are the *y*-coordinates of the selected zone i. If the area does not contain any node, a new area is randomly selected and the process continues until s nodes are removed. The worst-case time complexity of the ZR operator is $O(n^2)$, although after an initial preprocessing of all areas, the worst-case time complexity can be reduced to $O(n)$.

12. **Node neighbourhood removal (NNR)**: This operator initially selects a random node and then removes $s - 1$ nodes around it encapsulated in a rectangular area around the selected node. The choice of the rectangular neighbourhood, as supposed to, say, a circular neighbourhood, is motivated by the presence of grid-shaped networks encountered in several real-world settings as illustrated in Figure 3.5. If the number of nodes

within the selected zone is less than $s-1$, then the area is enlarged by a given percentage. The worst-case time complexity of the NNR operator is $O(n^2)$.

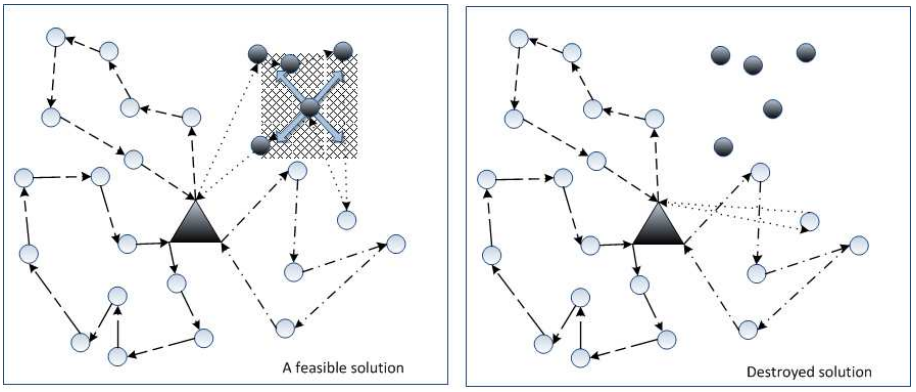


FIGURE 3.5: Node neighbourhood removal operator

3.3.1.4 Insertion operators

In this section, we present five insertion operators used in the ALNS algorithm. The first four of these operators are adapted from Ropke and Pisinger (2006*a*) whereas the last one is new. Insertion operators are used to repair a partially destroyed solution by inserting the nodes in the removal list back into the solution. These operators insert the removed nodes back into the existing routes when feasibility with respect to the capacity and time windows can be maintained, or they create new routes. We now briefly define the five insertion operators used in the main algorithm.

1. **Greedy insertion (GI)**: This operator repeatedly inserts a node in the best possible position of a route. The insertion cost is calculated as $d_i = d_{ji} + d_{ik} - d_{jk}$ for $j = 1, ..., s$ and $i = 1, ..., n$. The operator iterates $\phi = sn$ times. $S = \{j^*\}$ is selected such that $j^* = \underset{B \in \mathcal{B};\, j \in B}{\operatorname{argmin}}\{|d_i|\}$. The worst-case time complexity of the GI operator is $O(n^2)$.
2. **Regret insertion (RI)**: One problem with the greedy insertion operator is that it often postpones the insertion of the nodes until the later iterations when few feasible moves are possible. To counter this problem, this operator uses a 2-regret criterion. Let Δf_i denote the change in the objective value by inserting node i into its best and second best position for node i with respect to distance d_i as defined above. Let $i^* = \underset{i \in \mathcal{L}}{\operatorname{argmax}}$ $\{\Delta f_{i2} - \Delta f_{i1}\}$, where Δf_{i1} is the best feasible reinsertion and Δf_{i2} is the second best reinsertion of node i. For each node in the removal list, the operator is applied up to

$\phi = s^2 n$ times. This operator is quite time consuming, but unnecessary computations can be avoided when computing Δf_i. The worst-case time complexity of the RI operator is $O(n^3)$.

3. **Greedy insertion with noise function (GIN)**: This operator is an extension of the greedy algorithm but uses a degree of freedom in selecting the best place for a node. This degree of freedom is achieved by modifying the cost for node i: $New\,Cost = Actual\,Cost + \bar{d}\,\mu\,\epsilon$ where $\bar{d}$ is the maximum distance between nodes, μ is a noise parameter used for diversification and is set equal to 0.1, and ϵ is a random number between $[-1, 1]$. *New Cost* is calculated for each node in $\mathcal{L}$. The worst-case time complexity of the GIN operator is $O(n^2)$.

4. **Regret insertion with noise function (RIN)**: This operator is an extension of the 2-regret insertion algorithm but uses the same noise function as the GIN operator. The worst-case time complexity of the RIN operator is $O(n^3)$.

5. **Zone Insertion (ZI)**: This operator is similar to basic insertion but uses the time windows rather than distance to determine *best insertion* of each node. The zone algorithm determines the best position for each node and searches for another solution around it, feasible for the time window constraint. In other words, this operator tries to identify insertions that leave enough margin for future insertions. The worst-case time complexity of the ZI operator is $O(n^2)$.

3.3.1.5 Acceptance and stopping criteria

Simulated annealing was used as a local search framework for our ALNS algorithm. The ALNS algorithm with simulated annealing as a local search framework is presented in Algorithm 2. In the algorithm, X_{best} shows the best solution found during the search, $X_{current}$ is the current solution obtained at the beginning of an iteration, and X_{new} is a temporary solution found at the end of iteration that can be discarded or become the current solution. The cost of solution X is denoted by $c(X)$. A solution X_{new} is always accepted if $c(X_{new}) < c(X_{current})$, and accepted with probability $e^{-(c(X_{new})-c(X_{current}))/T}$ if $c(X_{new}) > c(X_{current})$, where T is the *temperature*. The temperature is initially set at $c(X_{init})P_{init}$ where $c(X_{init})$ is the objective function value of the initial solution X_{init} and P_{init} is an initialization constant. The current temperature is gradually

decreased during the course of the algorithm as hT, where $0 < h < 1$ is a fixed parameter. The algorithm returns the best found solution after a fixed number of iterations.

Algorithm 2: The general framework of the ALNS with simulated annealing

input : a set of removal operators D, a set of insertion operators I, initialization constant P_{init}, cooling rate h

output: X_{best}

Generate an initial solution by using the Clarke and Wright algorithm
Initialize probability P_d^t for each destroy operator $d \in D$ and probability P_i^t for each insertion operator $i \in I$
Let T be the temperature and j be the counter initialized as $j \leftarrow 1$
Let $X_{current} \leftarrow X_{best} \leftarrow X_{init}$
repeat
    Select a removal operator $d^* \in D$ with probability P_d^t
    Let X_{new}^* be the solution obtained by applying operator d^* to $X_{current}$
    Select an insertion operator $i^* \in I$ with probability P_i^t
    Let X_{new} be the new solution obtained by applying operator i^* to X_{new}^*
    if $c(X_{new}) < c(X_{current})$ **then**
        $X_{current} \leftarrow X_{new}$
    else
        Let $\nu \leftarrow e^{-(c(X_{new})-c(X_{current}))/T}$
    Generate a random number $\epsilon \in [0, 1]$
    if $\epsilon < \nu$ **then**
        $X_{current} \leftarrow X_{new}$
    if $c(X_{current}) < c(X_{best})$ **then**
        $X_{best} \leftarrow X_{new}$
    $T \leftarrow hT$
    Update probabilities using the adaptive weight adjustment procedure
    $j \leftarrow j + 1$
until *the maximum number of iterations is reached*

3.3.2 Speed optimisation

In this section, we introduce and analyse the speed optimisation problem (SOP). Given a vehicle route, the SOP consists of finding the optimal speed on each arc of the route between

successive nodes so as to minimise an objective function comprising fuel consumption costs and driver wages. The objective of SOP is non-linear due to the function used to estimate fuel consumption of a vehicle. Fixing all vehicle speeds at their optimal values with respect to fuel consumption may lead to violations of time window constraints. Furthermore, driver costs and fuel consumption do not always move in the same direction (see Bektaş and Laporte, 2011). This makes the SOP a non-trivial problem.

3.3.2.1 Mathematical model

The SOP is defined on a feasible path $(0, ..., n+1)$ of nodes all served by a single vehicle, where 0 and $n+1$ are two copies of the depot. The model uses the variable w_i to denote the waiting time at each node i, the variable v_i to represent the speed at which a vehicle travels between nodes i and $i+1$, and the variable e_i for the arrival time at node i. The vehicle has a minimum and maximum speed, represented by v_i^l and v_i^u, between nodes i and $i+1$. The formulation of SOP is as follows:

$$\text{Minimise} \quad \sum_{i=0}^{n} f_c F_i(v_i) \quad + \quad f_d e_{n+1} \tag{3.25}$$

subject to

$$e_{i+1} = e_i + w_i + t_i + d_i/v_i \qquad i = 0, ..., n \tag{3.26}$$

$$a_i \leq e_i + w_i \leq b_i \qquad i = 1, ..., n \tag{3.27}$$

$$v_i^l \leq v_i \leq v_i^u \qquad i = 0, ..., n \tag{3.28}$$

$$w_i \geq 0 \qquad i = 1, ..., n \tag{3.29}$$

$$e_i \geq 0 \qquad i = 1, ..., n+1 \tag{3.30}$$

$$v_i \geq 0 \qquad i = 1, ..., n \tag{3.31}$$

$$w_0 = e_0 = t_0 = 0, \tag{3.32}$$

where $F_i(v)$ is the total fuel consumption as derived in (3.6) but written using the load M_i, the acceleration τ_i and the road angle θ_i of arc $(i, i+1)$ for each $i = 0, ..., n$.

The objective function (3.25) minimises the total cost of fuel consumption and driver wages. We recall that f_c is the fuel and CO_2 emissions cost per liter and f_d is the driver wage per second. Other parameters are as defined in Section 3.2.2. Constraints (3.26) ensure that the arrival time at node $i+1$ is the sum of the arrival time at node i, the waiting time at node i, the service time at node i and the travel time to node i. Constraints (3.27) guarantee that service

at node i should start between a_i and b_i. Constraints (3.28) define upper and lower limits for speed. Constraints (3.29)–(3.31) impose the non-negativity restrictions on the variables.

We now describe a Speed Optimisation Algorithm (SOA) for the SOP. This algorithm is an adapted version of that of Norstad et al. (2010) and Hvattum et al. (2012), first proposed for ship routing. The algorithm is exact provided the cost function is convex. This is easily proved by adapting the arguments of Hvattum et al. (2012) to our case.

At the beginning of the algorithm, the speeds v_i are calculated for each link by considering the beginning of the next time window and total available time. These speed values are used to find violations if any. The violation is calculated if the departure time is less than sum of a_i and t_i or arrival time to the node is greater than b_i. Otherwise, the violation is set to zero. If the minimal possible speed limit is greater than the optimal speed, the optimal speed is increased to the minimal speed. This will not violate the time window since increasing speed means that less time is spent on the arc, and this speed is feasible if the lower speed does not violate the time window. The optimal speeds and current speeds are then compared; if the current speed is less than the optimal value, it is replaced with the optimal value. The algorithm selects at each stage the arc with the largest time window violation and eliminates the violation. In order to apply our SOP algorithm, it remains to show that (3.25) is convex.

Proposition 3.1. *The objective function (3.25) is convex.*

Proof. Using equations (3.26) and (3.32), the objective function (3.25) can be expanded as follows:

$$\sum_{i=0}^{n} f_c F_i(v) + f_d \left(\sum_{i=0}^{n} (w_i + t_i + d_i/v) \right)$$

$$= \sum_{i=0}^{n} \left[f_c F_i(v) + f_d (w_i + t_i + d_i/v) \right].$$

Let $g_i(v) = f_c F_i(v)$ and $h_i(v) = f_d(w_i + t_i + d_i/v)$ for each $i = 0, ..., n$. Then, since $dg_i(v)/dv = -kNV\lambda d_i/v^2 + 2\beta\lambda\gamma v d_i$ and $d^2 g_i(v)/dv = 2kNV\lambda d_i/v^3 + 2\beta\gamma\lambda d_i > 0$, $g_i(v)$ is a convex function. Similarly, since $dh_i(v)/dv = -f_d d_i/v^2$ and $d^2 h_i(v)/dv = 2f_d d_i/v^3 > 0$, $h_i(v)$ is a convex function. Since the sum of convex functions is convex, the proof follows. □

Proposition 3.2. *Given an arc $(i, i+1)$ and other parameters as described in the Section 3.4.1, the optimal speed which minimises fuel consumption costs and wage of driver is:*

$$v^* = \left(\frac{kNV}{2\beta\gamma} + \frac{f_d}{2\beta\lambda\gamma f_c} \right)^{1/3}, \tag{3.33}$$

and the optimal speed which minimises fuel consumption costs is:

$$v^* = \left(\frac{kNV}{2\beta\gamma} \right)^{1/3}. \tag{3.34}$$

Proof. (3.33) follows from $d(g_i(v) + h_i(v))/dv = 0$ for each $i = 0, ..., n$. (3.34) follows from $dh_i(v)/dv = 0$ for each $i = 0, ..., n$. □

A pseudo-code of the SOA is shown in Algorithm 3. The SOA runs in two stages. In the first the stage, optimum speeds are calculated to minimise fuel, emission and driver costs. The first stage also calculates the minimal required travel time time of the depot, which is then set equal to upper bound of the time windows. In the second stage, speeds are revised to optimise fuel consumption. In Algorithm 3, the only difference between two stages is the optimal speed calculation in line 6, where optimal speeds are calculated using (3.33) for the first stage and using (3.34) for the second stage. The algorithm starts with a feasible route; it takes input parameters *s* and *e*, and returns speed optimised routes.

3.4 Computational Results

This section presents results of extensive computational experiments performed to assess the performance of our ALNS heuristic. We first describe the generation of the test instances and of the parameters. We then present the results.

3.4.1 Data and experimental setting

For the computational experiments, three classes of PRP instances are generated, namely small, with up to 10 customers, medium, between 15 and 75 customers, and large, with more than 100 customers. Nine sets of 20 instances each were generated. The size of the instances ranges from 10 to 200 nodes. The instances represent randomly selected cities from the UK and therefore use real geographical distances. Time windows and service times are randomly

Algorithm 3: Speed Optimisation Algorithm(s, e)

initialise: $violation \leftarrow 0$, $p \leftarrow 0$, $D \leftarrow \sum_{i=s}^{e-1} d_i$, $T \leftarrow \sum_{i=s}^{e} t_i$

for $i = s + 1$ **to** e **do**
  if $\underline{e}_i \leq a_i$ **then**
    $v_{i-1} \leftarrow D\ /(\overline{e}_i - a_i - T)$
  else
    $v_{i-1} \leftarrow D\ /(\overline{e}_i - \underline{e}_i - T)$
  $v^*_{i-1} \leftarrow$ *Optimal Speed*
  if $\overline{e}_{i-1} + d_{i-1}/v_{i-1} < a_i$ *and* $\underline{e}_i \geq a_i + t_i$ *and* $i \neq |R| - 1$ **then**
    $v_{i-1} \leftarrow d_{i-1}/(a_i - \overline{e}_{i-1})$

  else if $\overline{e}_{i-1} + d_{i-1}/v_{i-1} < a_i$ *and* $\overline{e}_i \geq b_i + t_i$ *and* $i \neq |R| - 1$ **then**
    $v_{i-1} \leftarrow d_{i-1}/(b_i - \overline{e}_{i-1})$

  if $i = (N - 1)$ *and* $\overline{e}_i \neq \underline{e}_i$ **then**
    $v_{i-1} \leftarrow d_{i-1}/(a_i - \overline{e}_{i-1})$
  if $v^*_{i-1} < d_{i-1}/(b_{i+1} - a_i - t_i)$ **then**
    $v^*_{i-1} \leftarrow d_{i-1}/(b_{i+1} - a_i - t_i)$
  if $v^*_{i-1} > v_{i-1}$ **then**
    $v_{i-1} \leftarrow v^*_{i-1}$
  $\underline{e}_i \leftarrow \overline{e}_{i-1} + d_{i-1}/v_{i-1}$
  if $i \neq e$ **then**
    $\overline{e}_i \leftarrow \underline{e}_i + t_i$
  $g_i \leftarrow \max\{0, \underline{e}_i - b_i, a_i + t_i - \overline{e}_i\}$
  if $g_i > violation$ **then**
    $violation \leftarrow g_i$
    $p \leftarrow i$
if $violation > 0$ *and* $\underline{e}_p > b_p$ **then**
  $\overline{e}_p \leftarrow b_p + t_p$
  Speed Optimisation Algorithm(s, p)
  Speed Optimisation Algorithm(p, e)
if $violation > 0$ *and* $\overline{e}_p < a_p + t_p$ **then**
  $\overline{e}_p \leftarrow a_p + t_p$
  Speed Optimisation Algorithm(s, p)
  Speed Optimisation Algorithm(p, e)

generated. All instances are available for download from http://www.apollo.management.soton.ac.uk/prplib.htm.

The proposed algorithm was implemented in C. All experiments were conducted on a server with 3GHz speed and 1 GB RAM. A preliminary analysis was conducted to fine-tune the parameters. No claim is made that our choice of parameter values is the best possible. However, the settings used generally worked well on our test problems. Our algorithm contains fifteen user controlled parameters which are shown in the Table 3.2.

TABLE 3.2: Parameters used in the ALNS heuristic

Group	Description	Typical values
I	Total number of iterations (N_i)	25000
	Number of iterations for roulette wheel (N_w)	450
	Roulette wheel parameter (r_p)	0.1
	New global solution (σ_1)	1
	Better solution (σ_2)	0
	Worse solution (σ_3)	5
II	Startup temperature parameter (P_{init})	100
	Cooling rate (h)	0.999
III	Lower limit of removable nodes ($\underline{s}$)	5–20% of $\|\mathcal{N}\|$
	Upper limit of removable nodes ($\overline{s}$)	12–30% of $\|\mathcal{N}\|$
	Zone parameter (z)	11
	First Shaw parameter (Φ_1)	0.5
	Second Shaw parameter (Φ_2)	0.25
	Third Shaw parameter (Φ_3)	0.15
	Fourth Shaw parameter (Φ_4)	0.25
	Noise parameter (μ)	0.1

The parameters used in the ALNS algorithm are grouped into three categories as described below.

1. The first group defines the selection procedure with the roulette wheel mechanism. We note that our setting of the parameters σ_1, σ_2 and σ_3 is contrary to the expected setting $\sigma_1 \geq \sigma_2 \geq \sigma_3$, normally used to reward an operator for good performance. In our implementation and similar to Pisinger and Ropke (2005, 2007), we have chosen an unconventional setting of these parameters whereby the discovery of a worse solution is rewarded more than the discovery of a better solution. This is to help diversify the search in the algorithm.

   To show the number of times each removal and insertion operator was called within the heuristic algorithm, we provide some information in Tables 3.3 and 3.4, respectively.

These tables show, for each operator, the frequency of use in the algorithm as a percentage of the total number of iterations. The total time spent to run each operator is shown in parentheses. These results are obtained using one instance from each set.

The results shown in Table 3.3 indicate that the frequency of using the different removal operators do not significantly vary from one another. As the instances get larger in size, the frequencies of SR and NR increase compared to other operators. We note, however, that the time consumed by SR and NR operators is significantly higher than other operators for instances with more than 75 nodes.

As for the insertion operators, BGI is generally used slightly more than the other three as indicated by the results shown in Table 3.4. The times consumed by RI and RIN are significantly higher than those of the remaining operators.

2. The second group of parameters is used to calibrate the simulated annealing search framework and to define the initial temperature and cooling rate for the algorithm. Figure 3.6 shows the behaviour of the heuristic for a 100-node instance. The figure displays the way that best (X_{best}), current ($X_{current}$) and new (X_{new}) solutions change over 25000 iterations.

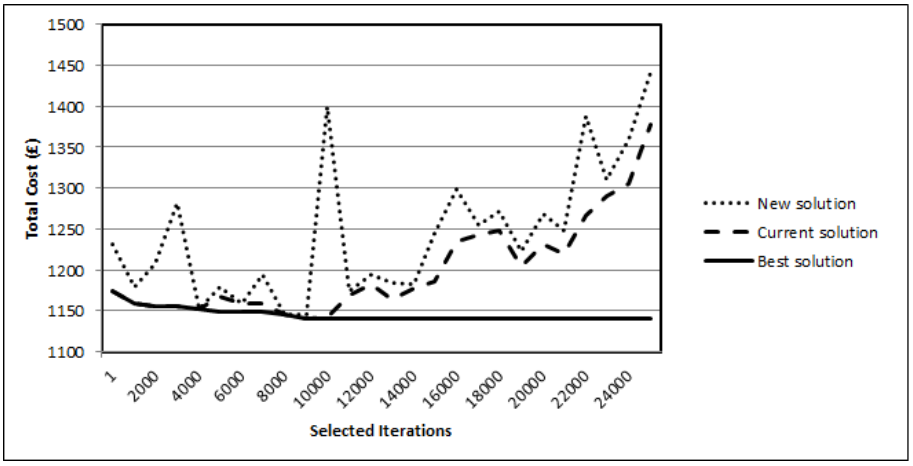


FIGURE 3.6: Solution values obtained by ALNS for a 100-node instance

3. The third and last group of the parameters are specific to the way in which the removal or insertion operators work. Here, the most important parameter that significantly affects the solution quality is the allowable number of removable nodes as defined by a lower $\underline{s}$ and an upper $\overline{s}$ bound, calculated as a percentage of the total number of nodes in an instance. To give the reader an idea of the effect of these two parameters, we provide in Table 3.5 some statistics for varying values of $\underline{s}$ and $\overline{s}$ on a 100-node instance. The

TABLE 3.3: Number of iterations as a percentage of 25000 and the CPU times required by the removal operators

Instance sets	Removal operators										
	RR	WDR	WTR	RoR	SR	PR	TR	DR	HR	NR	
10	10.9 (0.0)	10.7 (0.0)	11.0 (0.0)	7.8 (0.0)	10.7 (0.0)	9.8 (0.0)	9.7 (0.0)	10.7 (0.0)	8.9 (0.0)	10.3 (0.0)	
15	7.6 (0.0)	12.0 (0.0)	12.7 (0.0)	6.4 (0.0)	14.1 (0.0)	6.8 (0.0)	8.0 (0.0)	7.6 (0.0)	9.6 (0.0)	15.1 (0.0)	
20	7.4 (0.0)	10.3 (0.0)	12.1 (0.0)	3.2 (0.0)	12.9 (0.0)	8.8 (0.0)	9.2 (0.0)	9.6 (0.0)	11.7 (0.0)	17.3 (0.1)	
25	8.5 (0.0)	16.0 (0.0)	11.5 (0.0)	0.5 (0.0)	10.4 (0.1)	6.6 (0.0)	7.4 (0.0)	9.9 (0.0)	11.7 (0.0)	17.3 (0.1)	
50	7.1 (0.0)	12.3 (0.1)	13.2 (0.1)	7.3 (0.0)	15.3 (0.4)	6.3 (0.1)	8.4 (0.0)	7.9 (0.0)	9.7 (0.0)	12.5 (0.4)	
75	9.0 (0.0)	12.5 (0.1)	12.9 (0.2)	6.3 (0.0)	16.6 (1.0)	2.5 (0.0)	9.2 (0.0)	10.2 (0.0)	8.9 (0.0)	11.8 (1.3)	
100	6.3 (0.1)	13.5 (0.2)	10.8 (0.2)	7.0 (0.0)	14.2 (3.0)	7.2 (0.1)	8.6 (0.1)	7.2 (0.1)	10.9 (0.0)	14.3 (2.9)	
150	7.7 (0.0)	14.4 (0.3)	11.4 (0.4)	7.6 (0.0)	16.1 (6.2)	6.1 (0.1)	7.9 (0.2)	8.6 (0.2)	7.7 (0.1)	12.5 (6.7)	
200	9.1 (0.1)	16.1 (0.5)	9.1 (0.5)	4.8 (0.1)	12.4 (17.5)	8.7 (0.3)	7.7 (0.3)	7.9 (0.2)	8.7 (0.1)	15.5 (17.6)	

TABLE 3.4: Number of iterations as a percentage of 25000 and the CPU times required by the insertion operators

Instance sets	Insertion operators			
	GI	RI	GIN	RIN
10	32.1 (0.1)	28.4 (0.2)	16.8 (0.0)	22.7 (0.3)
15	22.6 (0.1)	27.0 (0.2)	21.2 (0.0)	29.2 (0.3)
20	27.0 (0.2)	28.9 (0.6)	19.0 (0.1)	25.0 (0.4)
25	31.0 (0.3)	33.5 (1.1)	12.9 (0.1)	22.6 (0.5)
50	27.6 (1.2)	29.5 (2.8)	20.1 (0.8)	22.8 (1.8)
75	30.3 (2.5)	30.9 (5.6)	18.3 (1.3)	20.5 (2.6)
100	28.5 (3.9)	28.4 (11.7)	19.2 (1.6)	24.0 (6.0)
150	27.8 (8.1)	25.6 (19.8)	26.4 (1.6)	20.1 (6.0)
200	26.2 (15.1)	23.4 (39.7)	25.0 (17.7)	25.3 (36.1)

table shows the percent deviation of the solutions found for each combination of $\underline{s}$ and $\overline{s}$ from the best known solution for this instance.

TABLE 3.5: Effect of removable nodes on the quality of solutions obtained and measured as percentage deviation from the best known solution

	$\underline{s}$	2	4	8	10	16
$\overline{s}$						
10		0.126	0.586	0.906	0.303	–
12		1.158	1.131	0.644	1.068	–
16		**0.000**	0.979	0.698	1.561	0.612
20		1.033	0.638	1.591	1.234	2.218
24		1.356	1.534	1.749	1.011	0.709
28		0.222	0.841	1.245	1.883	1.111
32		2.137	1.010	0.611	2.307	1.633
36		1.295	0.709	0.478	2.457	1.086
40		1.274	1.682	2.557	0.885	2.047
50		2.574	1.974	2.368	1.606	2.601
60		1.904	2.666	2.629	2.644	2.434
70		3.051	3.051	3.201	3.493	3.357

Other parameters in the third group include the following. The number of zones for zone removal and zone insertion operators is specified by z. Parameters Φ_1–Φ_4 are specific to the Shaw removal operator as explained in Section 3.3.1.3. Finally, the last parameter (μ) is used to diversify the solution.

3.4.2 Fine-tuning of the operators

In this section, we present results of computational testing on the selection of the operators within the algorithm.

3.4.2.1 Effect of the adaptive operator selection scheme

Adaptivity is assumed to be one of the advanced features of the ALNS algorithm. In this part of the section, we investigate whether the adaptive nature of the ALNS algorithm makes any difference in the results. The analysis of this section aims to compare the adaptive scheme with a static scheme, which assumes that the selection probabilities of the operators not change during the course of the algorithm. For this purpose, we test 10 instances from the Solomon set and the PRP set and present the results in Tables 3.6 and 3.7.

TABLE 3.6: Analysis of adaptivity of the operators on Solomon's benchmark instances

	$ALNS_{adaptive}$		$ALNS_{equal}$		
	Average value	CPU time s	Average value	CPU time s	Dev %
c101	827.3	82.4	827.3	81.64	0.00
c201	589.1	151.86	589.1	174.64	0.00
r101	1656.4	102	1651.2	104.88	0.31
r112	974.4	92.6	977.1	88.74	−0.28
r201	1158.6	156.72	1172.1	127.61	−1.15
r211	767.2	240.25	762.4	183.18	0.63
rc101	1634.6	99.7	1667.5	100.48	−1.97
rc108	1137.8	88.72	1175.9	91	−3.24
rc201	1286.7	150.27	1300.5	134	−1.06
rc208	786.9	193.49	795.6	163.66	−1.09
Average					−0.79

These tables show, for each instance, the average solution value and CPU time of 10 runs by using the adaptive scheme under column $ALNS_{adaptive}$ and by using the static scheme under column $ALNS_{equal}$. The last column presents the percentage deviation between $ALNS_{adaptive}$ and $ALNS_{equal}$. It can be seen from Tables 3.6 and 3.7 that an adaptive approach works better on both Solomon's and PRP instances. The improvement obtained by $ALNS_{adaptive}$ is around 0.79% on Solomon's benchmark instances and 1% on the PRP instances. These findings indicate that adaptivity helps improving solution quality.

TABLE 3.7: Analysis of adaptivity of the operators on PRP instances

	$ALNS_{adaptive}$		$ALNS_{equal}$		
	Average value	CPU time s	Average value	CPU time s	Dev %
UK100_01	2846.62	37.54	2865	44.46	−0.64
UK100_02	2695.59	38.67	2723.95	44.8	−1.04
UK100_03	2506.74	37.91	2554.72	43.03	−1.88
UK100_04	2440.21	36.52	2418.35	38.58	0.90
UK100_05	2333.23	36.14	2317.47	38.78	0.68
UK100_11	2728.11	37.2	2815.42	42.49	−3.10
UK100_12	2382.39	39.43	2453.71	43.68	−2.91
UK100_13	2660.16	37.77	2668.2	41.31	−0.30
UK100_14	2927.02	36.02	2971.84	40.24	−1.51
UK100_15	3059.44	34.28	3065.8	40.85	−0.21
Average					−1.00

3.4.2.2 Effect of penalising time of operators

In this part of the section, a different approach is tested for the changing the way in which operators are chosen. The classical approach only considers the improvement in the solution value, or the lack thereof, as a way of measuring an operator's performance. This section investigates the effect of the computational time required by the operators. The new cost function is based on penalising the operators with higher execution times in the hope of reducing the overall execution time of the ALNS algorithm.

Tables 3.8 and 3.9, for Solomon's benchmark and PRP instances, provide the average cost and CPU time of 10 runs by using the classical scheme under column $ALNS_{cost}$ and by using the time penalisation scheme under column $ALNS_{time}$. The last column presents the percentage deviation between the results found by $ALNS_{cost}$ and $ALNS_{time}$.

The results suggest that penalisation of CPU times improves the overall execution time of the algorithm but at the expense of solution quality. The percentage deviation for Solomon's benchmark instances and PRP instances are -1.13% and -0.94%, respectively.

3.4.2.3 Effect of changing the roulette wheel selection parameters

This section analyses the effect of using different sets of values for roulette wheel selection parameters (σ_1, σ_2, σ_3). Tables 3.10 and 3.11, for Solomon's benchmark and PRP instances,

TABLE 3.8: The time analysis of operators on Solomon's benchmark instances

	$ALNS_{cost}$		$ALNS_{time}$		
	Average value	CPU time s	Average value	CPU time s	Dev %
c101	827.3	82.4	827.3	63.67	0.00
c201	589.1	151.86	589.1	92.2	0.00
r101	1656.4	102	1652.7	92.25	0.22
r112	974.4	92.6	976.3	79.81	−0.19
r201	1158.6	156.72	1186.3	91.11	−2.33
r211	767.2	240.25	772.8	228.38	−0.72
rc101	1634.6	99.7	1692.1	61	−3.40
rc108	1137.8	88.72	1207.7	68.2	−5.79
rc201	1286.7	150.27	1281.2	111.28	0.43
rc208	786.9	193.49	782.9	125.54	0.51
Average					−1.13

TABLE 3.9: The time analysis of operators on PRP instances

	$ALNS_{cost}$		$ALNS_{time}$		
	Average value	CPU time s	Average value	CPU time s	Dev %
UK100_01	2846.62	37.54	2875.77	15.17	−1.01
UK100_02	2695.59	38.67	2712.29	41.02	−0.62
UK100_03	2506.74	37.91	2534.53	52.57	−1.10
UK100_04	2440.21	36.52	2432.76	43.53	0.31
UK100_05	2333.23	36.14	2358.75	44.59	−1.08
UK100_11	2728.11	37.2	2818.42	42.49	−3.20
UK100_12	2382.39	39.43	2379.79	40.93	0.11
UK100_13	2660.16	37.77	2696.54	55.7	−1.35
UK100_14	2927.02	36.02	2979.36	42.8	−1.76
UK100_15	3059.44	34.28	3050.55	35	0.29
Average					−0.94

provide the best values of 10 runs by using the parameter set shown in the second line of the table. The last three column present the percentage deviation between the use of the the initial values $(1,\ 0,\ 5)$ and the others tested.

The results suggest that the initially proposed setting $(1,\ 0,\ 5)$ performs better than other ones tested here.

3.4.3 Results of the ALNS heuristic on the VRPTW

To help assess the quality of the ALNS heuristic, we present computational results in Tables 6–8 on Solomon's benchmark VRPTW instances, which come in three sets r, c and rc classified with respect to the geographical locations of the nodes. These tables compare the results of the original ALNS as reported in (Pisinger and Ropke, 2005) and denoted ALNS$_O$, to our ALNS heuristic, denoted ALNS$_I$. The extended version of the algorithm using the new operators is denoted ALNS$_E$. The comparisons are made in terms of best solution values obtained through 10 runs of each algorithm. These tables present, for each instance, the value of the best known or optimal solution compiled from several sources (e.g., Milthers, 2009; Pisinger and Ropke, 2005) under column "Best known value". All figures presented in Tables 6–8 use a single decimal place (Milthers, 2009; Pisinger and Ropke, 2005) as opposed to two decimal places (e.g., Pisinger and Ropke (2007)). For each variant of the algorithm, we then present the value of best solution obtained in column "Best value" and the corresponding average CPU time required to run the algorithm. As for the last three columns, the column titled Dev$_{PR}$ (%) presents the percentage deviation of ALNS$_E$ from ALNS$_O$, the column titled Dev$_I$ (%) shows the percentage deviation of ALNS$_I$ from those reported under "Best known value", and and the column titled Dev$_E$ (%) shows the percentage deviation of ALNS$_E$ from those reported under "Best known value". In particular, let $\upsilon(A)$ be the solution value produced by algorithm A. Then, Dev$_{PR}$ (%) is calculated as $100\,(\upsilon(\text{ALNS}_O)-\upsilon(\text{ALNS}_E))\ /\upsilon(\text{ALNS}_O)$, Dev$_I$ (%) is calculated as $100\,(\upsilon(\text{Best})-\upsilon(\text{ALNS}_I))\ /\upsilon(\text{Best})$ and Dev$_E$ (%) is calculated as $100\,(\upsilon(\text{Best})-\upsilon(\text{ALNS}_E))\ /\upsilon(\text{Best})$, where $\upsilon(\text{Best})$ is the best known solution value for each instance.

As shown in Tables 3.12–3.14, the extended ALNS heuristic performs very well on the VRPTW instances considered in our tests. For a majority of the instances, the heuristic is able to discover the best known solution. For the rest of the instances, the percentage deviations are no greater than 0.72%. The tables also show that the operators work well to improve the solutions produced by ALNS$_I$, particularly on the r instances, and help discover solutions which are slightly better than those reported in Pisinger and Ropke (2005) on some instances. In

TABLE 3.10: Performance analysis of roulette wheel selection parameters on Solomon's benchmark instances

	$(\sigma_1, \sigma_2, \sigma_3)$								
	(1, 0, 5)	(5, 0, 1)	(5, 1, 0)	(5, 3, 0)					
	Best value	Best value	Best value	Best value	Dev_1 %	Dev_2 %	Dev_3 %		
c101	827.30	827.30	827.30	827.30	0.00	0.00	0.00		
c201	589.10	589.10	589.10	589.10	0.00	0.00	0.00		
r101	1656.40	1660.30	1659.00	1659.00	−0.24	−0.16	−0.16		
r112	974.40	984.20	988.20	988.20	−1.01	−1.42	−1.42		
r201	1158.60	1194.80	1169.30	1169.30	−3.12	−0.92	−0.92		
r211	767.20	759.80	755.00	755.80	0.96	1.59	1.49		
rc101	1634.60	1679.50	1664.10	1664.10	−2.75	−1.80	−1.80		
rc108	1137.80	1170.40	1152.00	1152.00	−2.87	−1.25	−1.25		
rc201	1286.70	1299.50	1307.10	1307.10	−0.99	−1.59	−1.59		
rc208	786.90	786.90	784.70	784.70	0.00	0.28	0.28		
Average					−1.00	−0.53	−0.54		

TABLE 3.11: Performance analysis of roulette wheel selection parameters on PRP instances

	$(\sigma_1, \sigma_2, \sigma_3)$								
	(1, 0, 5)	(5, 0, 1)	(5, 1, 0)	(5, 3, 0)					
	Best	Best	Best	Best	Dev_1	Dev_2	Dev_3		
	value	value	value	value	%	%	%		
UK100_01	2846.62	2830.79	2849.11	2864.31	0.56	−0.09	−0.62		
UK100_02	2695.59	2712.29	2715.23	2716.10	−0.62	−0.73	−0.76		
UK100_03	2506.74	2535.50	2530.91	2499.13	−1.15	−0.96	0.30		
UK100_04	2440.21	2411.31	2433.27	2499.21	1.18	0.28	−2.42		
UK100_05	2333.23	2316.61	2285.95	2307.23	0.71	2.03	1.11		
UK100_11	2728.11	2752.99	2743.60	2731.39	−0.91	−0.57	−0.12		
UK100_12	2382.39	2430.13	2465.39	2447.17	−2.00	−3.48	−2.72		
UK100_13	2660.16	2658.12	2650.00	2656.40	0.08	0.38	0.14		
UK100_14	2927.02	2978.02	2940.27	2965.57	−1.74	−0.45	−1.32		
UK100_15	3059.44	3056.55	3047.19	3000.50	0.09	0.40	1.93		
Average					−0.38	−0.32	−0.45		

TABLE 3.12: Results of the ALNS heuristic on benchmark VRPTW r instances

Solomon instances	Best known value	ALNS$_O$ Best value	ALNS$_O$ CPU time s	ALNS$_I$ Best value	ALNS$_I$ CPU time s	ALNS$_E$ Best value	ALNS$_E$ CPU time s	Dev$_{PR}$ %	Dev$_I$ %	Dev$_E$ %
r101	1637.7	1637.7	30	1652.8	53	1637.7	48	0.00	–0.92	0.00
r102	1466.6	1467.7	33	1479.4	52	1466.6	46	0.07	–0.87	0.00
r103	1208.7	1208.7	34	1222.1	52	1208.7	46	0.00	–1.11	0.00
r104	971.5	976	34	985.3	51	971.5	45	0.46	–1.42	0.00
r105	1355.3	1355.3	31	1369.8	53	1355.3	47	0.00	–1.07	0.00
r106	1234.6	1234.6	33	1242.9	50	1234.6	46	0.00	–0.67	0.00
r107	1064.6	1064.6	33	1073.7	48	1064.6	42	0.00	–0.85	0.00
r108	932.1	933.7	36	939.5	52	936.1	44	–0.26	–0.26	–0.43
r109	1146.9	1146.9	31	1151.3	53	1146.9	46	0.00	–0.38	0.00
r110	1068	1075.6	33	1081.6	55	1073.9	49	0.16	–1.27	–0.55
r111	1048.7	1048.7	33	1057.3	48	1049.9	42	–0.11	–0.82	–0.11
r112	948.6	948.6	33	954.2	45	948.6	40	0.00	–0.59	0.00
Average								0.03	–0.85	–0.09
r201	1143.2	1148.5	45	1160.1	77	1143.2	71	0.46	–1.48	0.00
r202	1029.6	1036.9	54	1051.2	79	1032.2	72	0.45	–2.10	–0.25
r203	870.8	872.4	60	881	83	873.3	76	–0.10	–1.17	–0.29
r204	731.3	731.3	67	754.9	80	731.3	75	0.00	–3.23	0.00
r205	949.8	949.8	58	951.8	76	950.4	71	–0.06	–0.21	–0.06
r206	875.9	880.6	61	887.6	82	881	76	–0.05	–1.34	–0.58
r207	794	794	72	803.5	89	794	85	0.00	–1.20	0.00
r208	701.2	701.2	86	714.7	93	702.9	88	–0.24	–1.93	–0.24
r209	854.8	855.8	60	863.1	78	854.8	74	0.12	–0.97	0.00
r210	900.5	908.4	59	920.6	75	906.3	70	0.23	–2.23	–0.64
r211	746.7	752.3	67	769.2	80	751.6	74	0.09	–3.28	–0.66
Average								0.08	–1.74	–0.25

particular, the average deviation is reduced from 0.85% to 0.09% for instances r101–r108 and from 1.74% to 0.25% for instances r201–r208 with the use of the new operators. Similar improvements are achieved on the instances rc for which the average deviation is reduced from 0.84% to 0.17% for instances rc101–rc108 and from 0.98% to 0.1% for instances rc201–rc208. These figures suggest that it is worthwhile to use the new operators to obtain good quality solutions. These results also confirm the effectiveness of our extended ALNS heuristic, given that it was designed to solve a problem more general than the classical VRPTW, whereas the best known solution values were obtained by means of specialized algorithms.

TABLE 3.13: Results of the ALNS heuristic on benchmark VRPTW c instances

Solomon instances	Best known value	ALNS$_O$ Best value	ALNS$_O$ CPU time s	ALNS$_I$ Best value	ALNS$_I$ CPU time s	ALNS$_E$ Best value	ALNS$_E$ CPU time s	Dev$_{PR}$ %	Dev$_I$ %	Dev$_E$ %
c101	827.3	827.3	29	827.3	44	827.3	41	0.00	0.00	0.00
c102	827.3	827.3	32	827.3	43	827.3	40	0.00	0.00	0.00
c103	826.3	826.3	34	826.3	44	826.3	41	0.00	0.00	0.00
c104	822.9	822.9	36	822.9	42	822.9	40	0.00	0.00	0.00
c105	827.3	827.3	30	827.3	41	827.3	39	0.00	0.00	0.00
c106	827.3	827.3	31	827.3	45	827.3	41	0.00	0.00	0.00
c107	827.3	827.3	31	827.3	44	827.3	41	0.00	0.00	0.00
c108	827.3	827.3	32	827.3	44	827.3	42	0.00	0.00	0.00
c109	827.3	827.3	34	827.3	43	827.3	40	0.00	0.00	0.00
Average								0.00	0.00	0.00
c201	589.1	589.1	69	589.1	85	589.1	82	0.00	0.00	0.00
c202	589.1	589.1	74	589.1	89	589.1	85	0.00	0.00	0.00
c203	588.7	588.7	80	588.7	96	588.7	92	0.00	0.00	0.00
c204	588.1	588.1	84	588.1	98	588.1	91	0.00	0.00	0.00
c205	586.4	586.4	76	586.4	91	586.4	86	0.00	0.00	0.00
c206	586	586	72	586	86	586	81	0.00	0.00	0.00
c207	585.8	585.8	74	585.8	88	585.8	86	0.00	0.00	0.00
c208	585.8	585.8	74	585.8	94	585.8	88	0.00	0.00	0.00
Average								0.00	0.00	0.00

3.4.4 The effect of speed optimisation

The SOA is extremely quick and able to improve upon the results produced by the ALNS. To give an example, Figure 3.7 shows the improvement by the SOA over routes found by the ALNS algorithm for a 100-node instance. In our implementation, the SOA terminates on a given route after three non-improving iterations. As Figure 3.7 shows, the improvement provided by the SOA on a given solution ranges between 0.41% and 3.01%.

A more detailed analysis of the effect of the SOA is provided in Table 3.15. This table presents, for one instance from each of the nine sets, the solutions obtained by using a distance-minimizing objective shown under column ALNS$_D$, and results with the SOA as applied on ALNS$_D$ shown under column ALNS$_D^+$. The ALNS$_D$ and ALNS$_D^+$ solutions are produced by initially using a distance minimizing objective, but the solution values reported in Table 3.15 are recalculated using the PRP objective. Table 3.15 also shows the results with ALNS using the PRP objective (4.19)–(4.23) shown under column ALNS$_P$ and results with SOA as applied on ALNS$_P$ shown under column ALNS$_P^+$. The last two columns present the percentage difference

TABLE 3.14: Results of the ALNS heuristic on benchmark VRPTW rc instances

Solomon instances	Best known value	ALNS$_O$ Best value	ALNS$_O$ CPU time s	ALNS$_I$ Best value	ALNS$_I$ CPU time s	ALNS$_E$ Best value	ALNS$_E$ CPU time s	Dev$_{PR}$ %	Dev$_I$ %	Dev$_E$ %
rc101	1619.8	1619.8	28	1623.7	48	1619.8	44	0.00	–0.24	0.00
rc102	1457.4	1463.5	30	1472.7	46	1463.5	42	0.00	–1.37	–0.42
rc103	1258	1267.0	31	1278.6	49	1267.1	43	–0.01	–1.64	–0.72
rc104	1132.3	1132.6	33	1139.8	48	1133.1	42	–0.04	–0.66	–0.07
rc105	1513.7	1513.7	30	1525.5	47	1513.7	43	0.01	–0.78	0.00
rc106	1372.7	1373.9	29	1381.4	44	1372.7	41	0.09	–0.63	0.00
rc107	1207.8	1209.3	30	1216.9	42	1209.3	40	0.00	–0.75	-0.12
rc108	1114.2	1114.2	31	1121.3	46	1114.2	43	0.00	–0.64	0.00
Average								0.01	–0.84	–0.17
rc201	1261.8	1262.6	42	1274.2	80	1262.7	74	–0.01	–0.98	–0.07
rc202	1092.3	1095.8	46	1102.3	76	1095.8	71	0.00	–0.92	–0.32
rc203	923.7	923.7	56	931.4	76	923.7	73	0.00	–0.83	0.00
rc204	783.5	785.8	68	790.8	79	783.8	76	0.29	–0.93	0.00
rc205	1154	1154	45	1163.1	68	1154	64	0.00	–0.79	0.00
rc206	1051.1	1051.1	52	1062.8	71	1051.1	68	0.00	–1.11	0.00
rc207	962.9	966.6	55	979.2	70	966.6	64	0.00	–1.69	–0.38
rc208	777.3	777.3	65	781.9	76	777.3	72	0.00	–0.59	0.00
Average								0.04	–0.98	–0.10

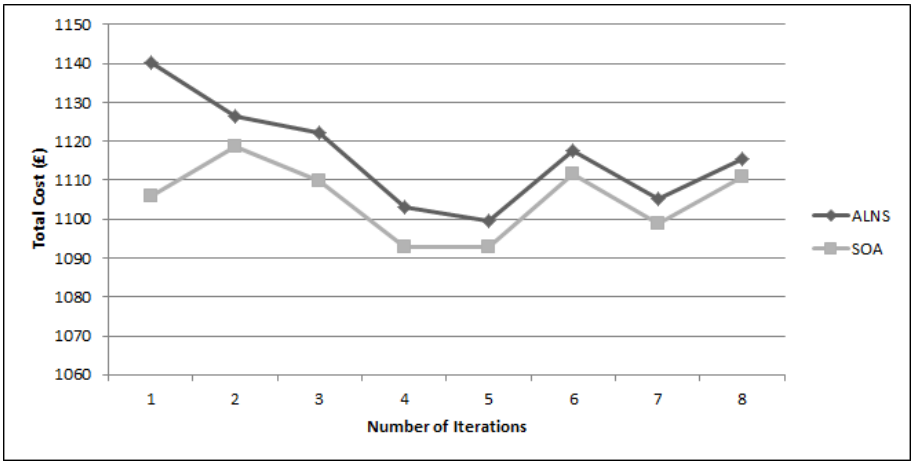


FIGURE 3.7: ALNS and SOP algorithm for a 100-node instance

between $ALNS_D$ and $ALNS_D^+$ under column D_D and the percentage difference between $ALNS_P$ and $ALNS_P^+$ under column D_P.

TABLE 3.15: Performance improvement using SOA on solutions obtained by ALNS

Instance sets	$ALNS_D$	$ALNS_D^+$	$ALNS_P$	$ALNS_P^+$	D_D %	D_P %
10	181.89	177.02	176.17	170.64	3.14	2.68
15	297.25	287.30	295.89	286.89	3.04	3.35
20	335.92	325.38	332.70	325.23	2.25	3.14
25	296.14	288.60	288.23	282.13	2.12	2.55
50	623.57	604.89	612.98	596.11	2.75	3.00
75	1016.26	985.99	1011.58	985.86	2.54	2.98
100	1304.37	1264.51	1262.33	1230.72	2.50	3.06
150	1524.99	1475.16	1512.60	1469.74	2.83	3.27
200	2241.34	2172.94	2212.31	2144.25	3.08	3.05
Average					2.69	3.01

The results shown in Table 3.15 indicate that the SOA is generally able to improve the solutions found by the ALNS by 2 to 4%. The average improvement in the case of minimizing the PRP function is 3.01%, which is slightly higher as compared to the average improvement of 2.65% seen in the case of minimizing distance.

3.4.5 PRP heuristic results

This section presents the results obtained by the proposed heuristic on the nine sets of PRP instances generated. Each instance was solved once with the proposed heuristic and once with the PRP model solved with a truncated execution of CPLEX 12.1 (IBM ILOG, 2009) with its default settings. A common time-limit of three hours was imposed on the solution time for all instances.

The detailed results of these experiments are presented in Tables 3.16 to 3.18. These tables give the results for 10-, 100- and 200-node instances, is given in Appendix A. Each table presents, for each instance, the solutions found by the heuristic and CPLEX. All columns in these tables are self-explanatory with the exception of the last, which shows the percentage deviation of the solution produced by the heuristic from the best solution obtained with CPLEX within the three-hour time limit.

TABLE 3.16: Computational results for 10-node instances

	CPLEX			Our heuristic								
Instances	Solution cost £	Total CPU time s	LP relaxation	Solution cost £	Distance km	# of vehicles	# of loop	ALNS time s	SOA time s	CPU time per loop s	Total CPU time s	Improvement %
UK10_01	170.66	163.4	79.3	170.64	409.0	2	4	0.5	0.0	0.5	2.1	0.01
UK10_02	204.87	113.9	95.1	204.88	529.8	2	4	0.6	0.0	0.6	2.3	0.00
UK10_03	200.33	926.0	78.5	200.42	507.3	2	4	0.5	0.0	0.5	2.0	−0.04
UK10_04	189.94	396.5	81.9	189.99	480.1	2	4	0.5	0.0	0.5	2.2	−0.03
UK10_05	175.61	1253.7	77.3	175.59	447.0	2	4	0.6	0.0	0.6	2.3	0.01
UK10_06	214.56	347.5	101.8	214.48	548.0	2	4	0.6	0.0	0.6	2.2	0.04
UK10_07	190.14	191.0	98.5	190.14	494.7	2	5	0.6	0.0	0.6	2.9	0.00
UK10_08	222.16	139.8	111.0	222.17	567.8	2	4	0.5	0.0	0.5	2.1	0.00
UK10_09	174.53	54.0	68.2	174.54	457.0	2	4	0.6	0.0	0.6	2.2	0.00
UK10_10	189.83	76.9	88.7	190.04	486.7	2	5	0.5	0.0	0.5	2.6	−0.11
UK10_11	262.07	50.5	129.3	262.08	697.2	2	4	0.5	0.0	0.5	2.2	0.00
UK10_12	183.18	1978.7	77.5	183.19	460.3	2	4	0.5	0.0	0.5	2.2	−0.01
UK10_13	195.97	1235.1	91.4	195.97	510.5	2	4	0.6	0.0	0.6	2.2	0.00
UK10_14	163.17	84.1	83.0	163.28	397.8	2	4	0.6	0.0	0.6	2.4	−0.07
UK10_15	127.15	433.3	59.6	127.24	291.4	2	4	0.6	0.0	0.6	2.4	−0.07
UK10_16	186.63	680.8	89.4	186.73	451.1	2	4	0.5	0.0	0.5	1.9	−0.06
UK10_17	159.07	27.0	83.4	159.03	387.5	2	4	0.6	0.0	0.6	2.3	0.03
UK10_18	162.09	522.1	70.3	162.09	401.5	2	4	0.5	0.0	0.5	2.2	0.00
UK10_19	169.46	130.5	84.2	169.59	414.5	2	7	0.6	0.0	0.6	4.1	−0.08
UK10_20	168.80	1365.5	73.1	168.80	412.8	2	4	0.5	0.0	0.5	2.0	0.00

TABLE 3.17: Computational results for 100-node instances

	CPLEX			Our heuristic								
Instances	Solution cost £	Total CPU time s	LP relaxation	Solution cost £	Distance km	# of vehicles	# of loops	ALNS time s	SOA time s	CPU time per loop s	Total CPU time s	Improvement %
UK100_01	1389.05	10800*	572.22	1240.79	2914.4	14	4	23.0	0.0	23.0	92.1	10.67
UK100_02	1302.16	10800*	548.13	1168.17	2690.7	13	4	24.5	0.0	24.5	98.2	10.29
UK100_03	1231.44	10800*	518.00	1092.73	2531.8	13	8	26.0	0.0	26.0	207.9	11.26
UK100_04	1174.75	10800*	487.57	1106.48	2438.5	14	7	21.4	0.0	21.4	149.7	5.81
UK100_05	1121.71	10800*	474.66	1043.41	2328.5	14	8	19.9	0.0	19.9	159.0	6.98
UK100_06	1320.40	10800*	565.78	1213.61	2782.4	14	7	19.1	0.0	19.1	133.8	8.09
UK100_07	1177.87	10800*	494.67	1060.08	2463.9	12	4	25.7	0.0	25.7	102.6	10.00
UK100_08	1230.92	10800*	516.07	1106.78	2597.4	13	9	23.3	0.0	23.3	209.5	10.09
UK100_09	1092.20	10800*	439.48	1015.46	2219.2	13	6	25.7	0.0	25.7	154.0	7.03
UK100_10	1163.95	10800*	499.34	1076.56	2510.1	12	8	24.9	0.0	24.9	199.0	7.51
UK100_11	1343.18	10800*	560.36	1210.25	2792.1	15	5	21.4	0.0	21.4	107.1	9.90
UK100_12	1227.01	10800*	483.91	1053.02	2427.3	12	9	22.9	0.0	22.9	206.4	14.18
UK100_13	1333.10	10800*	535.11	1154.83	2693.1	13	4	22.0	0.0	22.0	87.9	13.37
UK100_14	1410.18	10800*	599.61	1264.50	2975.3	14	4	23.0	0.0	23.0	91.8	10.33
UK100_15	1453.81	10800*	608.94	1315.50	3072.1	15	5	22.2	0.0	22.2	110.9	9.51
UK100_16	1105.58	10800*	448.32	1005.03	2219.7	12	10	25.5	0.0	25.5	254.7	9.09
UK100_17	1389.99	10800*	594.54	1284.81	2960.4	15	7	21.8	0.0	21.8	152.8	7.57
UK100_18	1219.45	10800*	501.69	1106.00	2525.2	13	4	23.1	0.0	23.1	92.6	9.30
UK100_19	1115.82	10800*	458.01	1044.71	2332.6	13	4	22.7	0.0	22.7	91.0	6.37
UK100_20	1396.97	10800*	594.06	1263.06	2957.8	14	9	22.7	0.0	22.7	204.4	9.59

*: Not solved to optimality in 10800 seconds.

TABLE 3.18: Computational results for 200-node instances

	CPLEX			Our heuristic								
Instances	Solution cost £	Total CPU time s	LP relaxation	Solution cost £	Distance km	# of vehicles	# of loops	ALNS time s	SOA time s	CPU time per loop s	Total CPU time s	Improvement %
UK200_01	2711.51	10800*	940.3	2111.70	4609.6	28	7	103.5	0.0	103.5	724.4	22.12
UK200_02	2726.99	10800*	897.2	1988.64	4444.4	24	9	113.4	0.0	113.4	1020.9	27.08
UK200_03	2621.57	10800*	903.7	2017.63	4439.9	27	4	101.0	0.0	101.0	404.1	23.04
UK200_04	3175.38	10800*	841.4	1934.13	4191.9	26	4	102.9	0.0	102.9	411.7	39.09
UK200_05	X	10800*	1008.6	2182.91	4861.9	27	9	102.9	0.0	102.9	926.4	X
UK200_06	X	10800*	811.1	1883.22	3980.4	27	4	112.6	0.0	112.6	450.3	X
UK200_07	3344.32	10800*	881.9	2021.95	4415.3	27	9	104.8	0.0	104.8	943.4	39.54
UK200_08	2980.33	10800*	951.8	2116.76	4664.4	27	4	107.6	0.0	107.6	430.6	28.98
UK200_09	2409.25	10800*	815.6	1894.18	4031.1	25	5	110.6	0.0	110.6	553.1	21.38
UK200_10	2871.22	10800*	1013.4	2199.95	4921.8	28	4	125.0	0.0	125.0	500.0	23.38
UK200_11	X	10800*	834.5	1941.19	4099.5	27	5	168.6	0.0	168.6	842.8	X
UK200_12	3388.33	10800*	994.0	2105.14	4808.5	25	11	64.6	0.0	64.6	711.0	37.87
UK200_13	X	10800*	986.9	2141.26	4760.3	25	4	111.1	0.0	111.1	444.6	X
UK200_14	X	10800*	894.4	2011.35	4369.9	27	4	112.7	0.0	112.7	450.6	X
UK200_15	3435.71	10800*	955.1	2110.86	4723.9	25	5	108.4	0.0	108.4	542.0	38.56
UK200_16	2691.36	10800*	935.3	2075.83	4545.9	27	4	113.9	0.0	113.9	455.6	22.87
UK200_17	3032.37	10800*	1028.4	2218.28	4972.8	26	4	102.3	0.0	102.3	409.2	26.85
UK200_18	3569.48	10800*	909.1	2004.68	4370.3	27	7	112.7	0.0	112.7	788.9	43.84
UK200_19	X	10800*	794.3	1844.90	3995.4	25	9	108.2	0.0	108.2	973.9	X
UK200_20	3561.79	10800*	984.1	2150.57	4805.4	27	5	106.2	0.0	106.2	531.1	39.62

X: No feasible solution found in 10800 seconds.

Table 3.16 shows that the heuristic finds the same solutions as those of CPLEX but in a substantially smaller amount of time. The fact that ALNS finds solutions which are better than the optimal solution provided by CPLEX for some instances is due to the discretisation used in the mathematical formulation. The average time required by CPLEX to solve the 10-node instances to optimality is 508.5 seconds where the same statistic for the ALNS to produce the reported solutions is 2.3 seconds.

A summary of the results of the 180 instances tested is presented in Table 3.19. In this table, we present, for each instance set, the average cost, the average deviation between the solution produced by the heuristic and the best solution obtained with the PRP model within the three-hour limit, the average CPU times for the heuristic and for CPLEX, and the average number of loops.

TABLE 3.19: Summary of comparisons between the proposed heuristic and CPLEX

Instance Sets	Average CPU time for CPLEX s	Average CPU time for the heuristic s	Average # of loops of the heuristic	Average time per loop s	Average improvement %
10	508.5	2.3	4.3	0.5	−0.02
15	10800*	3.9	4.4	0.9	0.07
20	10800*	6.4	4.6	1.4	0.53
25	10800*	10.0	5.3	1.9	0.88
50	10800*	35.4	6.5	5.5	2.38
75	10800*	70.5	5.9	12.1	5.97
100	10800*	145.3	6.3	23.0	9.35
150	10800*	348.0	6.0	62.2	18.06
200	10800*	625.7	5.9	109.7	31.01

As can be observed from Table 3.19, the results indicate that the proposed algorithm runs quickly even for large size instances. The minimum and maximum number of loops, for all instances, are four and 13, respectively. Instances of up to 100 nodes are solved in 145 seconds on average. The algorithm requires just over 10 minutes of computation time to solve instances with 200 nodes and is able to produce much better results than CPLEX does in three hours. The improvements can be as high as 10% for instances of up to 100 nodes, around 18% for instances with 150 nodes and around 30% for instances with 200 nodes. CPLEX was not able to find optimal solutions in three hours for 160 instances out of the total of 180 instances tested. The average CPU time per iteration of the algorithm is less than two minutes.

3.4.6 The analysis of speed values

This part of the analysis focuses on the speed values used in the solutions to some PRP instances with 100 nodes. Table 3.20 provides the percentage of three different speed categories: the optimal speed which minimises fuel consumption costs under column v_1^* (equation 3.34), the optimal speed which minimises fuel consumption and driver costs under column v_2^* (equation 3.33), and the speed values different v_1 and v_2 under column v_3.

TABLE 3.20: Obtained speed values on PRP instances

	v_1^* (%)	v_2^* (%)	v_3 (%)
UK100_01	4.4	91.2	4.4
UK100_02	5.3	93.0	4.4
UK100_03	4.4	87.7	7.0
UK100_04	11.4	82.5	7.0
UK100_05	1.8	90.4	6.1
UK100_11	5.3	86.0	2.6
UK100_12	7.0	89.5	1.8
UK100_13	4.4	90.4	0.0
UK100_14	0.9	91.2	7.0
UK100_15	3.5	92.1	4.4

The results shown in Table 3.20 indicate that the solutions found by the ALNS algorithm very often use the optimal speed v_2^*, showing that it is, in general, more important to minimise fuel and driver costs combined as opposed to only the former.

3.5 Conclusions

This chapter has described a heuristic algorithm to solve the PRP. The algorithm iterates between a VRPTW and a speed optimisation problem, the former solved through an enhanced ALNS and the latter solved using a polynomial time procedure. The enhanced ALNS uses new, as well as existing removal and insertion operators, which improve the solution quality. These operators can be used in ALNS for solving other types of problems. The SOA, on the other hand, improves the solution produced by the ALNS and minimises fuel consumption costs and driver wages by optimising vehicle speeds. The SOA has a negligible execution time, and is generic enough to be used as a stand-alone routine for other types of routing

problems in order to optimise speed. To fully evaluate the effectiveness of the heuristic algorithm, we have generated different sets of instances based on real geographic data and have compiled a library of PRP instances. We have presented results of extensive computational experimentation using the proposed heuristic and have compared it against the solutions produced using the integer linear programming formulation of the PRP. The results show that the proposed algorithm is highly effective in finding good-quality solutions on instances with up to 200 nodes.

Chapter 4

The Bi-Objective Pollution-Routing Problem

Abstract

The bi-objective Pollution-Routing Problem is an extension of the recently introduced Pollution-Routing Problem (PRP) which consists of routing a number of vehicles to serve a set of customers, and determining their speed on each route segment. The objective functions pertaining to minimisation of fuel consumption and driving time are conflicting and are thus considered separately. This chapter presents an adaptive large neighbourhood search algorithm (ALNS), combined with a speed optimisation algorithm, to solve the bi-objective PRP. Using the ALNS as the search engine, four *a posteriori* methods, namely the weighting method, the weighting method with normalisation, the epsilon-constraint method and a new hybrid method (HM), are tested using a scalarisation of the two objective functions. The HM combines adaptive weighting with the epsilon-constraint method. To evaluate the effectiveness of the algorithm, new sets of instances based on real geographic data are generated, and a library of bi-criteria PRP instances is compiled. Results of extensive computational experiments with the four methods are presented and compared with one another by means of the hypervolume and epsilon indicators. The results show that HM is highly effective in finding good-quality non-dominated solutions on PRP instances with 100 nodes.

4.1 Introduction

Until now, the planning of freight transportation activities has mainly focused on ways of saving money and increasing profitability by considering internal transportation costs only, e.g., fuel cost, drivers' wages (see, e.g., Crainic, 2000; Forkenbrock, 1999, 2001).

Freight transportation in the United Kingdom (UK) is responsible for 22% of the CO_2 emissions from the transportation sector, amounting to 33.7 million tonnes, or 6% of the CO_2 emissions in the country, of which road transport accounts for a proportion of 92% (McKinnon, 2007). The 2008 Climate Change Act commits the UK to an ambitious and legally binding 80% reduction in greenhouse gases (GHG) emissions by 2050, from a 1990 baseline. The transportation sector has an important role to play, as the third largest GHG contributor, in achieving reduction targets in the UK (Tight et al., 2005). The most prominent GHG is carbon dioxide (CO_2), the

emissions of which are directly proportional to the amount of fuel consumed by a vehicle. This amount is dependent on a variety of vehicle, environment and traffic-related parameters, such as vehicle speed, load and acceleration (Demir et al., 2011).

Freight transportation planning has many facets, particularly when viewed from the multiple levels of decision making. Arguably the most famous problem at this level is the well-known Vehicle Routing Problem (VRP), which consists of determining routes for a fleet of vehicles to satisfy the demands of a set of customers. The traditional objective in the standard VRP is to minimise a cost function which is traditionally considered to be the total distance traveled by all vehicles. Taking a more explicit look at externalities of freight transportation, and in particular vehicle routing, Bektaş and Laporte (2011) introduced the Pollution-Routing Problem (PRP) which aims at minimizing a total cost function comprising fuel and driving costs in the presence of time windows.

Most real-world problems involve multiple objectives. In the context of the PRP, two important objectives should be taken into account, namely minimisation of fuel consumption and the total driving time. The amount of fuel consumption depends on the energy required to move a vehicle from one point to another. As discussed in Demir et al. (2012*a*), for each vehicle there exists an optimal speed yielding a minimum fuel consumption. However, this speed is generally lower than the speed preferred by vehicle drivers in practice. Another important issue in road transportation is time management. In freight transport terminology, time is money and it is essential for firms to perform timely deliveries in order to establish and keep a good reputation. In practice, drivers' schedules tend to be flexible, with different numbers of hours worked each day, subject to driving time regulations. If a saving of one hour can be achieved on a given vehicle route, this would imply reducing the corresponding driver's costs by an hour (Fowkes and Whiteing, 2006). Reduction in time spent on a route can be achieved by travelling at higher speed, but this, in turn, increases fuel costs. Since the two objectives of minimizing fuel and time are conflicting, the problem requires the use of multi-objective optimisation to allow an evaluation of the possible trade-offs.

In this chapter, we investigate a bi-objective vehicle routing problem in which one of the objectives is related to fuel consumption and the other to driving time. This chapter also describes an enhanced adaptive large neighbourhood search (ALNS) algorithm for the bi-objective PRP, integrating the classical ALNS mechanism of Ropke and Pisinger (2006*a*) with a specialised speed optimisation algorithm described in Hvattum et al. (2012) and in Demir et al. (2012*a*). The scientific contribution of this study is four-fold: (*i*) to introduce of a bi-objective variant of the Pollution-Routing Problem, (*ii*) to apply and test existing multi-objective techniques to solve the bi-objective PRP, (*iii*) to describe a new hybrid heuristic for the bi-objective PRP, and

(*iv*) to perform extensive computational experiments using four *a posteriori* methods evaluated by means of two performance indicators.

The remainder of this chapter is organised as follows. In Section 4.2 we provide a general overview of multi-objective optimisation and we summarise the existing literature on multi-objective VRPs. Section 4.3 presents the bi-objective PRP along with a mathematical programming formulation. Section 4.4 includes the description of the heuristic algorithm. Section 4.5 presents the generation of the instances and the results of extensive computational experiments, together with managerial insights. Conclusions are stated in Section 4.6.

4.2 Multi-Objective Optimisation

Multi-objective optimisation (MOO), also known as multi-objective programming, multi-criteria or multi-attribute optimisation, is the process of simultaneously optimizing two or more conflicting objectives subject to a number of constraints. In this section, we consider a MOO problem of the form

$$\text{(MOO)} \quad \text{minimise} \quad \{f_1(x), f_2(x), ..., f_k(x)\} \tag{4.1}$$

$$\text{subject to} \quad x \in \mathcal{S}, \tag{4.2}$$

where f_k:$\mathfrak{R}^n \rightarrow \mathfrak{R}$ are $k \geq 2$ objective functions to be minimised simultaneously. The decision variables $x = (x_1, ..., x_n)^T$ belong to a non-empty feasible region (set) $\mathcal{S} \subseteq \mathcal{R}^n$. If there is no conflict between the objective functions, then a solution in which every objective attains its optimum values can be found. In this case, no special methods are needed. To avoid such trivial cases, we assume that no such solution exists. This means that the objective functions are at least partly conflicting. They may also be incommensurable, i.e., measured in different units (Miettinen, 1999), as is the case in this chapter.

For non-trivial multi-objective problems, one cannot identify a single solution that simultaneously optimises every objective. While searching for solutions, one reaches a point such that, when attempting to improve an objective, other objectives suffer as a result. A solution is called non-dominated, Pareto optimal, or Pareto efficient if it cannot be eliminated from consideration by replacing it with another solution which improves upon one of the objectives without worsening another. Finding such non-dominated solutions, and quantifying the trade-offs in satisfying the different objectives, is the goal of setting up and solving a MOO problem. The next section presents formal definitions of Pareto optimality.

4.2.1 Pareto optimality

In single objective optimisation problems, the main focus is on the space of the decision variables. In contrast, multi-objective problems are concerned with the objective function space. The concept of decision and objective space, and the correspondence between the two, are illustrated in Figure 4.1.

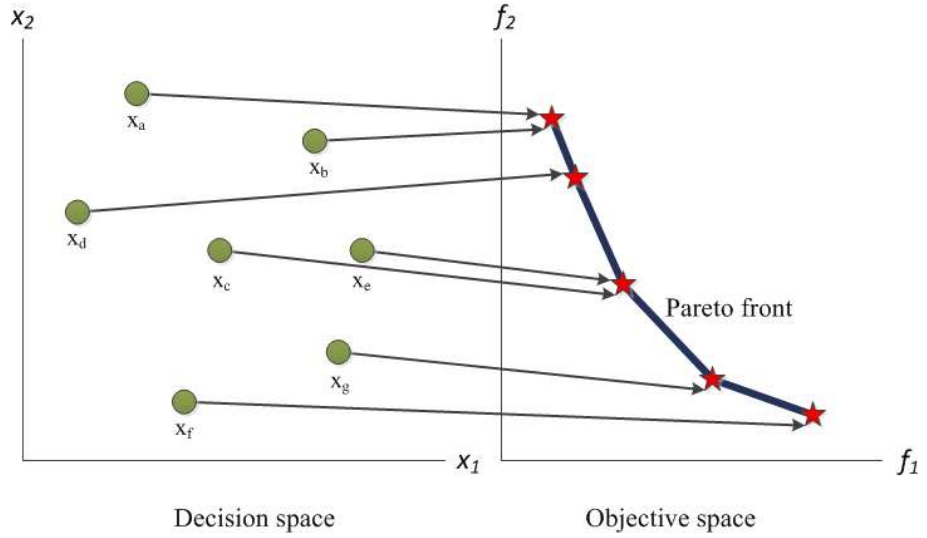


FIGURE 4.1: Decision and objective space

According to the concept of Pareto-optimality introduced by Vilfredo Pareto (see Pareto, 1971), every Pareto optimal point is an equally acceptable solution of the multi-objective optimisation problem. However, it is generally desirable to produce a single solution. Selecting one out of the set of Pareto optimal solutions calls for information that is not contained in the objective function, which differentiates MOO from single-objective optimisation. This requires a decision maker to select a solution. A decision maker is a person or a group of people with insight into the problem being solved and who can express preference relations between different solutions.

4.2.2 Multi-objective optimisation methods

In this section, we review several methods for solving MOO problems and for generating Pareto optimal solutions. General references on this topic can be found in Ehrgott and Gandibleux (2002) and Jozefowiez et al. (2008*a*).

Methods for MOO can be classified in various ways. One of them is based on whether many Pareto optimal solutions are generated or not, and on the role of the decision maker in solving the MOO problem (Rangaiah, 2009). Such a classification is shown in Figure 4.2, where the methods are initially grouped into two: (*i*) generating methods and (*ii*) preference-based methods. The former group of methods aims at generating one or more Pareto optimal points without any prior input from decision maker. In contrast, the latter uses extra information from a decision maker as part of the solution process. Generating methods are further divided into three: (*i*) no-preference methods, (*ii*) *a posteriori* methods using a scalarisation approach, and (*iii*) *a posteriori* method using a multi-objective approach.

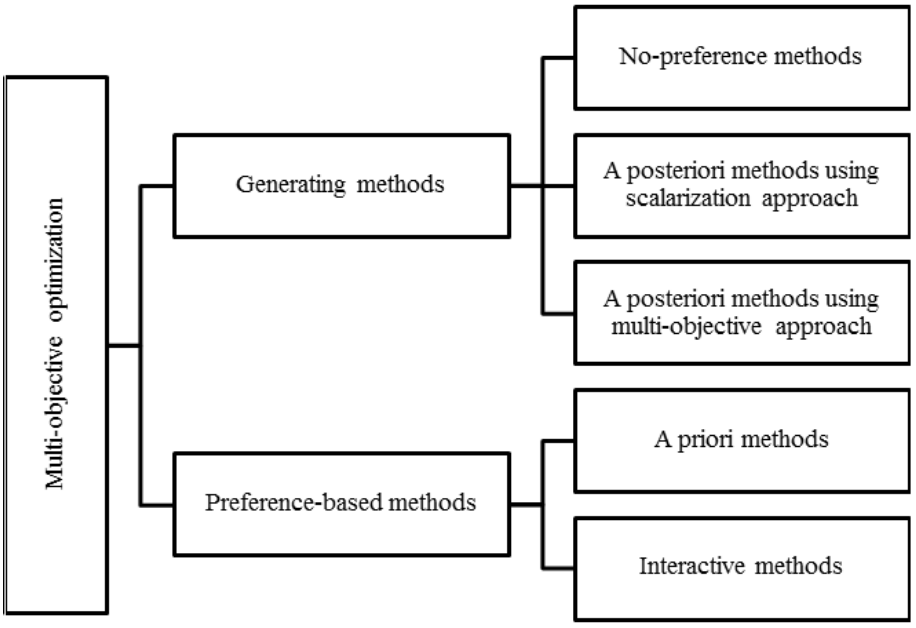


FIGURE 4.2: Classification of multi-objective optimisation methods

No-preference methods do not require any prior information and generally yield only one Pareto optimal point. Examples of such methods include global criterion and the multi-objective bundle method (Wierzbicki, 1999). The second type of *a posteriori* methods use a scalarisation approach. Scalarisation means converting the problem into a single or a family of single objective optimisation problems using a real-valued objective function, termed the scalarising function. The weighting and ϵ-constraint methods belong to *a posteriori* methods based on a scalarisation approach in which a series of scalarised single objective problems have to be solved to find the Pareto optimal points (Coello et al., 2002; Deb, 2001). The third type of *a posteriori* methods use multi-objective rank trial solutions based on the objective function values. Examples are the non-dominated sorting genetic algorithm and ant colony optimisation.

All *a posteriori* methods provide many Pareto optimal solutions to the decision maker who will then select their preferred one (Fonseca and Fleming, 1995).

Preference-based methods are further divided into two: (i) *a priori* and (ii) interactive methods. In the former, preferences of a decision maker are sought and are then included in the initial formulation of a single objective problem. Some of the *a priori* methods are value function methods, lexiographic ordering and goal programming. The latter requires interaction with the decision maker during the solution process. Examples are interactive surrogate worth trade-off method and the NIMBUS method (Deb and Chaudhuri, 2007).

This chapter focuses on the use of *a posteriori* methods using scalarisation of the objective functions. The methods used in this chapter are described below.

4.2.2.1 The weighting method (WM)

In the weighting method, the idea is to associate each objective function with a weighting coefficient and to minimise the weighted sum of the objectives. This way, multiple objective functions are transformed into a single objective function as in Miettinen (1999). We assume that the weighting coefficients w_i are non-negative for all $i = 1, ..., k$. Weights are normalised in such a way that $\sum_{i=1}^{k} w_i = 1$. The MOO is then transformed into the following problem:

$$\text{minimise} \quad \sum_{i=1}^{k} w_i f_i(x) \tag{4.3}$$

$$\text{subject to} \quad x \in \mathcal{S}. \tag{4.4}$$

Problem (4.3)–(4.4) is a single-objective optimisation problem which can be solved by existing methods, such as linear or integer programming.

4.2.2.2 The weighting method with normalisation (WMN)

This method is an extension of weighting method where the objective functions are normalised to take values between 0 and 1 (Grodzevich and Romanko, 2006). This is done by using the differences of optimal function values in the worst (called the Nadir) and the best (called the Utopia) points, which yield the length of the intervals in which objective functions vary within

the Pareto optimal set. The MOO problem is transformed into the following problem:

$$\text{minimise} \quad \sum_{i=1}^{k} w_i (f_i(x) - z_i^U)/(z_i^N - z_i^U) \tag{4.5}$$

$$\text{subject to} \quad x \in \mathcal{S}, \tag{4.6}$$

where z^U is the ideal objective vector, i.e., the Utopia point, and z^N is the Nadir point. The ideal point, which optimises all objective functions, is not normally feasible because of the conflicting objectives but provides lower bounds for the Pareto optimal set. The ideal point can be calculated as $z_i^U = f_i(x^{[i]})$ where $z_i = \arg\min_x \{f_i(x) : x \in \mathcal{S}\}$. The Nadir point, which corresponds to the worst objective value for each of the objectives, may be feasible and provides upper bounds for the Pareto optimal set. It is calculated as $z_i^N = \max_{1 \leq j \leq k} f_i(x^{[j]}), \forall i = 1, .., k$. In practice, these points can be found by calculating the best and the worst values of each objective function.

4.2.2.3 ϵ-constraint method (ECM)

In the ϵ-constraint method, only one of the objective functions is selected to be optimised, which all others are converted into constraints by imposing an upper bound. The problem to be solved then takes the form

$$\text{minimise} \quad f_l(x) \tag{4.7}$$

$$\text{subject to} \quad f_j(x) \leq \epsilon_j \quad \forall j = 1, ..., k; j \neq l \tag{4.8}$$

$$x \in \mathcal{S}. \tag{4.9}$$

In order to generate as many Pareto optimal solutions as possible, the right-hand side of constraint (4.8) is gradually increased by a small amount and the problem is solved again whenever ϵ_j is increased.

As indicated by Mavrotas (2009), this method offers several advantages over WM and WMN. The weighting method may lead to an extreme solution. In contrast, the ϵ-constraint method is able to produce non-extreme (interior) efficient solutions. Moreover, the computational effort is less than that of the weighting method. Furthermore, WMN requires the normalisation of the objective functions which may affect the results, whereas this is not required for the ϵ-constraint method. Other extensions of the ϵ-constraint method are discussed by Ehrgott and Ryan (2002) and Laumanns et al. (2006).

4.2.2.4 Hybrid method (HM)

This method combines the adaptive weighting and ϵ-constraint methods. The name "hybrid" was suggested by Chankong and Haimes (1983) and Vira and Haimes (1983). The proposed hybrid method takes the form

$$\text{minimise} \quad \sum_{i=1}^{k} w_i \, (f_i(x) - z_i^U) \tag{4.10}$$

$$\text{subject to} \quad f_j(x) \leq \epsilon_j \quad \forall j = 1, ..., k \tag{4.11}$$

$$x \in \mathcal{S}, \tag{4.12}$$

where $w_i > 0 \; \forall(i) \quad i = 1, ..., k$.

HM is inspired by the ECM because of its several advantages over other a posteriori methods. The main motivation of the HM is to improve the solution quality of non-dominated solutions with an adaptive implementation of weights while limiting one of the objective functions. HM therefore does not need to normalise the objective functions as in WMN.

4.2.3 Multi-objective route planning

In this section, we look at the existing studies on the multi-objective VRP. The VRP has been the subject of intensive research efforts both for heuristic and exact optimisation approaches. However, the multi-objective variant of the VRP has not been intensively investigated. A survey of multi-objective on VRPs can be found in Jozefowiez et al. (2008*b*), who present a classification of objectives related to different aspects of VRPs. These are: tour (cost, profit, makespan, balance and etc.), nodes/arcs (time windows, customer satisfaction and etc.), and resources (management of the fleet, characteristics of the product to collect/deliver, etc.).

Evolutionary algorithms constitute a widely popular approach in solving multi-objective VRPs. Due to their population-based nature, these algorithms are able to approximate the whole Pareto front (or surface) in a single run. An extensive survey on multi-objective evolutionary algorithms can be found in Zitzler et al. (2000), Veldhuizen and Lamont (2000) and Zhou et al. (2011). Among the existing work on multi-objective VRPs, we mention the ones below.

Lee and Ueng (1999) have studied a version of the vehicle routing problem in which the aim is to minimise the total distance and to balance the workload among employees. They have proposed a heuristic to determine a trade-off between the two objectives. A VRP model that considers three objectives and multiple periods is described by Ribeiro and Ramalhinho-Lourenço (2001). The first objective is cost minimisation, the second is the balance of work levels and the third is marketing research. The authors propose an iterated local search heuristic using weighting method to generate solutions. Zografos and Androutsopoulos (2004) consider a bi-objective Vehicle Routing Problem with Time Windows (VRPTW). These are minimisation of length and the risk of transporting hazardous materials. The authors propose a local search heuristic for this problem. In the context of passenger transportation, Pacheco and Martí (2005) study school bus routing with two objectives: to minimise the number of buses, and to minimise the longest time a student must stay in the bus. The authors present a number of constructive solution methods and a tabu search procedure to obtain non-dominated solutions. Chen et al. (2008) study a complex VRP in a multi-modal network, where besides arc capacities and time windows, additional constrains (i.e., mandatory and forbidden nodes) are considered. The authors attempt to optimise the travel time, the operative cost and a transportation mean sharing index for a real-life application in Italy. Another real-life application study was introduced by Caramia and Guerriero (2009). The authors worked on selecting an appropriate route for transporting nuclear waste in Taiwan. The objectives of this study are the minimisation of travel time, transportation risk and the exposed population. Another relevant study is that of Paquette et al. (2011), who proposed a tabu search algorithm incorporating a routing cost and quality of service in the dial-a-ride problem, where the reference point method is used to generate several non-dominated solutions. Finally, we mention the work by Qian et al. (2011), who seek to improve helicopter transportation safety by solving a routing problem with a risk objective expressed in terms of expected number of fatalities. Jabali et al. (2012a) consider a VRP model that takes into account of travel time, fuel, and CO_2 emissions costs in a time-dependent context, where the latter are estimated using emission functions provided in the MEET report (Hickman et al., 1999). The authors describe a tabu search algorithm for the problem and show, through computational experiments, that limiting vehicle speeds to a certain extent is effective in reducing emissions although costly in terms of total travel time.

4.3 The Bi-Objective Pollution-Routing Problem

We now describe the bi-objective PRP. This problem is defined on a complete directed graph $\mathcal{G} = (\mathcal{N}, \mathcal{A})$ where $\mathcal{N} = \{0, ..., n\}$ is the set of nodes, 0 is a depot and $\mathcal{A} = \{(i, j) : i, j \in \mathcal{N}$ and

$i \neq j\}$ is the set of arcs. The distance from node i to node j is denoted by d_{ij}. A fixed-size fleet of m vehicles, each of capacity Q, is available to serve the nodes. The set $\mathcal{N}_0 = \mathcal{N}\backslash\{0\}$ is a customer set, and each customer $i \in \mathcal{N}_0$ has a non-negative demand q_i as well as a time interval $[a_i, b_i]$ in which service must start; early arrivals to customer nodes are permitted but a vehicle, arriving early must wait until time a_i before service can start. The service time of customer i is denoted by t_i. The PRP, proposed by Bektaş and Laporte (2011) and further studied by Demir et al. (2012*a*), is a single-objective optimisation problem. Here, we look at the trade-off between the two conflicting objectives in the PRP, namely fuel consumption and total driving time. These objectives are described in greater detail below.

4.3.1 The fuel consumption objective

The fuel consumption objective is based on the *comprehensive emissions model* described by Barth et al. (2005), Scora and Barth (2006), and Barth and Boriboonsomsin (2008), which is an instantaneous model estimating fuel consumption for a given time instant. According to this model, the fuel rate is given by

$$FR = \xi(kNV + P/\eta)/\kappa, \tag{4.13}$$

where ξ is the fuel-to-air mass ratio, k is the engine friction factor, N is the engine speed, V is the engine displacement, and η and κ are constants. The parameter P is the second-by-second engine power output (in kW), and can be calculated as

$$P = P_{tract}/\eta_{tf} + P_{acc}, \tag{4.14}$$

where η_{tf} is the vehicle drive train efficiency, and P_{acc} is the engine power demand associated with running losses of the engine and the operation of vehicle accessories such as air conditioning. The parameter P_{tract} is the total tractive power requirements (in kW) placed on the wheels:

$$P_{tract} = (M\tau + Mg\sin\theta + 0.5C_d\rho Av^2 + MgC_r\cos\theta)v/1000, \tag{4.15}$$

where M is the total vehicle weight (kg), v is the vehicle speed (m/s), τ is the acceleration (m/s^2), θ is the road angle, g is the gravitational constant, and C_d and C_r are the coefficients of the aerodynamic drag and rolling resistance, respectively. Finally, ρ is the air density and A is the frontal surface area of the vehicle. For a given arc (i, j) of length d, let v be the speed of a vehicle speed traversing this arc. If all variables in FR except for the vehicle speed v remain

constant on arc (i, j), the fuel consumption (in L) on this arc can be calculated as

$$F(v) = \lambda kNVd/v \tag{4.16}$$

$$+ \lambda\gamma Pd/v, \tag{4.17}$$

where $\lambda = \xi/\kappa\psi$ and $\gamma = 1/1000n_{tf}\eta$ are constants and ψ is the conversion factor of fuel from gram/second to liter/second. Furthermore, let M be the load carried between nodes i and j. More specifically, $M = w + f$, where w is the curb weight (i.e., the weight of an empty vehicle) and f is the vehicle load. Let $\alpha = \tau + g\sin\theta + gC_r\cos\theta$ be a vehicle-arc specific constant and $\beta = 0.5C_d\rho A$ be a vehicle-specific constant. We omit the indices (i, j) on the variables v, d, f and α to simplify the presentation. Then, $F(v)$ can be rewritten as

$$F(v) = \lambda\left(kNV + w\gamma\alpha v + \gamma\alpha fv + \beta\gamma v^3\right)d/v. \tag{4.18}$$

In order to model (4.18) as an objective function, we use a discretised speed function defined by R non-decreasing speed levels $\bar{v}^r$ $(r = 1, ..., R)$. Binary variables z_{ij}^r indicate whether or not arc $(i, j) \in \mathcal{A}$ is traversed at a speed level r. We further define binary variables x_{ij} equal to 1 if and only if arc (i, j) appears in the solution, continuous variables f_{ij} representing the total amount of flow on each arc $(i, j) \in \mathcal{A}$, and continuous variables y_j representing the time at which service starts at node $j \in \mathcal{N}_0$. The mathematical representation of the fuel consumption objective is shown below:

$$\text{minimise} \quad \sum_{(i,j)\in\mathcal{A}} kNV\lambda d_{ij}\sum_{r=1}^{R} z_{ij}^r/\bar{v}^r \tag{4.19}$$

$$+ \sum_{(i,j)\in\mathcal{A}} w\gamma\lambda\alpha_{ij}d_{ij}x_{ij} \tag{4.20}$$

$$+ \sum_{(i,j)\in\mathcal{A}} \gamma\lambda\alpha_{ij}d_{ij}f_{ij} \tag{4.21}$$

$$+ \sum_{(i,j)\in\mathcal{A}} \beta\gamma\lambda d_{ij}\sum_{r=1}^{R} z_{ij}^r(\bar{v}^r)^2. \tag{4.22}$$

The objective function (4.19)–(4.22) is derived from (4.18). For further details, the reader is referred to Demir et al. (2012*a*).

4.3.2 The driving time objective

The driving time objective is the sum of the total journey time of all routes starting and ending at the depot. This time is equal to the arrival time to the depot under assumption that vehicles start their journey at time zero. The variable s_j represents the total time spent on a route that has a node $j \in \mathcal{N}_0$ as last visited before returning to the depot. The mathematical representation of the driving time objective is

$$\text{minimise} \quad \sum_{j \in \mathcal{N}_0} s_j. \tag{4.23}$$

This objective measures the total driving time. The total time spent on a route where customer $j \in \mathcal{N}_0$ is visited last before returning to the depot can be calculated as

$$s_j = c_j + t_j + d_{j0}/v^r, \tag{4.24}$$

where c_j is the waiting time at node j, t_j is the service time at node j, and (d_{j0}/v^r) is the journey time from last node of a route to the depot.

4.3.3 Constraints

The constraints of the integer programming formulation of the bi-objective PRP are similar to those given in Bektaş and Laporte (2011) and Demir et al. (2012*a*), and are shown below:

$$\sum_{j \in \mathcal{N}} x_{0j} = m \tag{4.25}$$

$$\sum_{j \in \mathcal{N}} x_{ij} = 1 \qquad \forall i \in \mathcal{N}_0 \tag{4.26}$$

$$\sum_{i \in \mathcal{N}} x_{ij} = 1 \qquad \forall j \in \mathcal{N}_0 \tag{4.27}$$

$$\sum_{j \in \mathcal{N}} f_{ji} - \sum_{j \in \mathcal{N}} f_{ij} = q_i \qquad \forall i \in \mathcal{N}_0 \tag{4.28}$$

$$q_j x_{ij} \leq f_{ij} \leq (Q - q_i) x_{ij} \qquad \forall (i,j) \in \mathcal{A} \tag{4.29}$$

$$y_i - y_j + t_i + \sum_{r \in \mathcal{R}} d_{ij} z_{ij}^r / \bar{v}^r \leq K_{ij}(1 - x_{ij}) \qquad \forall i \in \mathcal{N}, j \in \mathcal{N}_0, i \neq j \tag{4.30}$$

$$a_i \leq y_i \leq b_i \qquad \forall i \in \mathcal{N}_0 \tag{4.31}$$

$$y_j + t_j - s_j + \sum_{r \in \mathcal{R}} d_{j0} z_{j0}^r / \bar{v}^r \leq L(1 - x_{j0}) \qquad \forall j \in \mathcal{N}_0 \tag{4.32}$$

$$\sum_{r=1}^{R} z_{ij}^r = x_{ij} \qquad \forall (i,j) \in \mathcal{A} \tag{4.33}$$

$$x_{ij} \in \{0,1\} \qquad \forall (i,j) \in \mathcal{A} \tag{4.34}$$

$$f_{ij} \geq 0 \qquad \forall (i,j) \in \mathcal{A} \tag{4.35}$$

$$y_i \geq 0 \qquad \forall i \in \mathcal{N}_0 \tag{4.36}$$

$$z_{ij}^r \in \{0,1\} \qquad \forall (i,j) \in \mathcal{A}, r = 1, ..., R. \tag{4.37}$$

Constraints (4.25) state that each vehicle must leave the the depot. Constraints (4.26) and (4.27) are the degree constraints which ensure that each customer is visited exactly once. Constraints (4.28) and (4.29) define the arc flows. Constraints (4.30)–(4.32), where $K_{ij} = \max\{0, b_i + s_i + d_{ij}/l_{ij} - a_j\}$, and L is a large number, enforce the time window restrictions. Constraints (4.33) ensure that only one speed level is selected for each arc and $z_{ij}^r = 1$ if $x_{ij} = 1$.

The PRP is NP-hard since it is an extension of the classical VRP. Bektaş and Laporte (2011) have shown that even a simplified version of this problem cannot be solved optimally for mid-size instances. For this reason, heuristics are needed to obtain good-quality solutions within short computational times, one which we describe in the following section.

4.4 A Bi-Objective Adaptive Large Neighbourhood Search Algorithm with Speed Optimisation Algorithm

In this section, we present an enhanced version of the ALNS algorithm introduced in Demir et al. (2012*a*), to solve the bi-objective PRP. This metaheuristic is an extension of the large neighbourhood search (LNS) heuristic first proposed by Shaw (1998), and based on the idea of modifying an initial solution by means of destroy and repair operators. If the new solution is better than the current best solution, it replaces it and use as an input to the next iteration. The ALNS heuristic was put forward by Ropke and Pisinger (2006*a*) to solve variants of the VRP. Rather than using one large neighbourhood as in LNS, it applies several removal and insertion operators to a given solution. Insertion operators are used to repair a partially destroyed solution by inserting the nodes in the removal list back into the solution. These operators insert the removed nodes back into the existing routes when feasibility with respect to the capacity and time windows can be maintained, or they create new routes. The neighbourhood of a solution is obtained by removing some customers from the solution and reinserting them back. Our implementation uses the classical Clarke and Wright (1964) heuristic to construct an initial solution. Furthermore, we use 12 removal and five insertion operators in the ALNS algorithm, which are selected dynamically in the algorithm according to their past performance. To this end, each operator is assigned a *score* which is increased whenever it improves the current solution. The new solution is accepted if it satisfies a criterion defined by the simulated annealing (Kirkpatrick et al., 1983) used as local search framework applied at the outer level. A speed optimisation algorithm (SOA) is applied at each iteration of the algorithm. Given a vehicle route, the SOA consists of finding the optimal speed on each arc of the route in order to minimise an objective function comprising fuel consumption and driving time.

The pseudocode of the overall algorithm is given in Algorithm 4.

Algorithm 4: The bi-objective ALNS algorithm

input : a set of removal operators D, a set of insertion operators I, initialisation constant P_{init}, cooling rate h, maximum number of iterations j_{max}

output: X_{best}

Generate an initial solution by using the Clarke and Wright (1964) algorithm
Let T be the temperature and j be the counter initialised as $j = 1$
Let $\mathcal{S}$ be the non-dominated list and $\mathcal{S} = \varnothing$
Initialise probability $P_d = 1/|D|$ for each destroy operator $d \in D$ and probability $P_i = 1/|I|$ for each insertion operator $i \in I$
Set the all speed levels to at its maximum level Let $X_{current} = X_{best} = X_{init}$
repeat
    Select a removal operator $d^* \in D$ with probability P_d
    Let X_{new} be the solution obtained by applying operator d^* to $X_{current}$
    Select an insertion operator $i^* \in I$ with probability P_i
    Let X_{new} be the new solution obtained by applying operator i^* to $X_{current}$
    Apply Speed Optimisation Algorithm on X_{new}
    if $c(X_{new}) < c(X_{current})$ **then**
        $X_{current} = X_{new}$
    Let $\nu = e^{-(c(X_{new})-c(X_{current}))/T}$
    Generate a random number $\ell \in [0, 1]$
    if $\ell < \nu$ **then**
        $X_{current} = X_{new}$
    if $c(X_{current}) < c(X_{best})$ **then**
        $X_{best} = X_{new}$
    if X_{new} *is not dominated by any* $x \in \mathcal{S}$ **then**
        $\mathcal{S} \leftarrow \mathcal{S} \cup \{X_{new}\}$
    Compare X_{new} with non-dominated set and add to the set if X_{new} is non-dominated solution
    $\Delta \leftarrow h\,\Delta$
    Update probabilities using the adaptive weight adjustment procedure
    Adjust all speed levels to their maximum speed levels if a preset number of iterations is run
    $j \leftarrow j + 1$
until $j \leq j_{max}$

In Algorithm 4, X_{best} is the best solution found during the search, $X_{current}$ is the current solution obtained at the beginning of an iteration, and X_{new} is a temporary solution found at the end of iteration that can be discarded or become the current solution. The objective value of a solution X is denoted by $c(X)$ and is calculated depending on the bi-objective method in use. A solution X_{new} is always accepted if $c(X_{new}) < c(X_{current})$, and accepted with probability $e^{-(c(X_{new})-c(X_{current}))/T}$ if $c(X_{new}) > c(X_{current})$ where T denotes the *temperature*. The temperature is initially set at $c(X_{init})P_{init}$ where $c(X_{init})$ is the objective function value of the initial solution X_{init} and P_{init} is an initialisation constant. The current temperature is gradually decreased during the course of the algorithm as hT, where $0 < h < 1$ is a fixed parameter. The algorithm returns the set $\mathcal{S}$ of non-dominated solution found in the course of algorithm. Figure 4.3 depicts the steps of the ALNS algorithm.

4.5 Computational Results

This section presents the results of extensive computational experiments performed to assess the performance of the multi-objective methods using our ALNS algorithm with speed optimisation algorithm. We first describe the generation of the test instances, the parameters and the quality indicators used to assess the performance of the proposed methods. We then present the computational results. The parameters used in the experiments are given in Table 4.1.

4.5.1 Generation of the test instances

For the experiments, 13 sets of 10 instances each were generated, resulting in a total of 130 instances. Each instance has 100 nodes, which represent randomly selected cities from the UK, and uses real road distances. In Table 4.2 we give the design of each instance set. This table presents the number of vehicles, the lower and upper bounds of time windows, the service times and the load intervals for each instance set. Data pertaining to time windows, service times and load are randomly generated within these intervals. Each instance set is of a different nature, characterised by the average number of vehicles (minimum number required based on load), time windows (loose or tight) and load (homogeneous or heterogeneous). All instances are available for downloading from http://www.apollo.management.soton.ac.uk/prplib.htm.

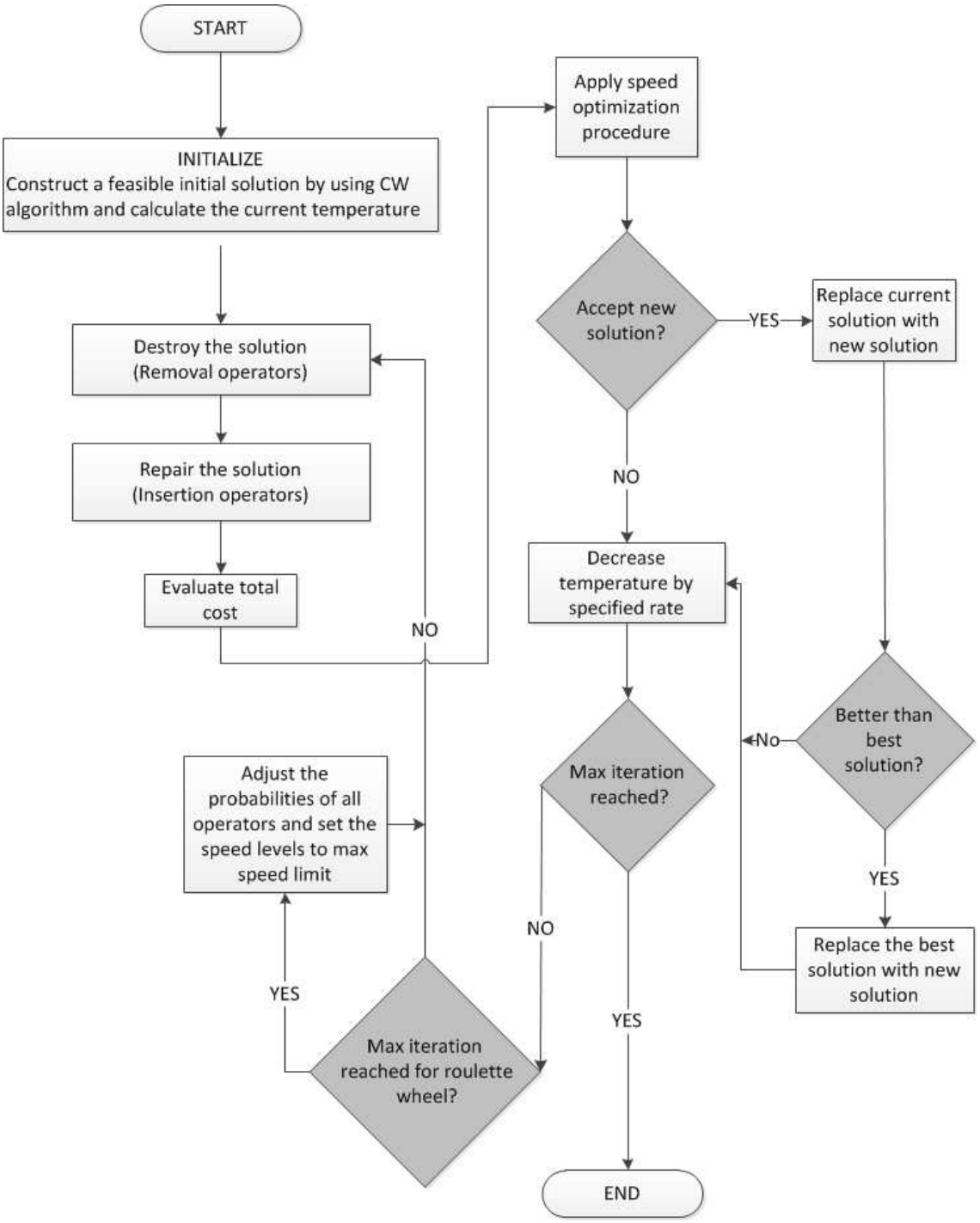


FIGURE 4.3: The framework of the ALNS with speed optimisation algorithm

TABLE 4.1: Parameters used in the computational tests

Notation	Description	Typical values
w	curb-weight (kg)	6350
ξ	fuel-to-air mass ratio	1
k	engine friction factor (kJ/rev/litre)	0.2
N	engine speed (rev/s)	33
V	engine displacement (litres)	5
g	gravitational constant (m/s^2)	9.81
C_d	coefficient of aerodynamic drag	0.7
p	air density (kg/m^3)	1.2041
A	frontal surface area (m^2)	3.912
C_r	coefficient of rolling resistance	0.01
n_{tf}	vehicle drive train efficiency	0.4
η	efficiency parameter for diesel engines	0.9
f_c	fuel and CO_2 emissions cost per litre (£)	1.4
f_d	driver wage per (£/s)	0.0022
κ	heating value of a typical diesel fuel (kJ/g)	44
ψ	conversion factor (g/s to L/s)	737
v^l	lower speed limit (m/s)	5.5 (or 20 km/h)
v^u	upper speed limit (m/s)	27.8 (or 100 km/h)

TABLE 4.2: The general structure of 100-node instances

		Time window			Load
Instance sets	Average # of vehicles	Lower bound s	Upper bound s	Service time s	kg
1	5	0	32400	300	180
2	5	600–2400	27000–32400	300	180
3	5	0	32400	300	130–230
4	5	600–2400	27000–32400	300	130–230
5	10	0	32400	600	360
6	10	600–2400	27000–32400	600	360
7	10	0	32400	600	310–410
8	10	600–2400	27000–32400	600	310–410
9	20	0	32400	900	720
10	20	600–2400	27000–32400	900	720
11	20	0	32400	900	670–770
12	20	600–2400	27000–32400	900	670–770
13	5–20	600–2400	27000–32400	300–900	180–760

The proposed algorithm was implemented in C. All experiments were conducted on a server

with 3GHz speed and 1 GB RAM. A preliminary analysis was conducted to fine-tune the parameters. No claim is made that our choice of parameter values is the best possible. However, the settings used generally worked well in our preliminary analysis. The detailed values of all ALNS parameters are presented in Demir et al. (2012*a*), with the exception of the new settings given in Table 4.3.

TABLE 4.3: Parameters used in the ALNS heuristic

Description	Typical values
Number of loops (N_1)	11
Total number of iterations (N_2)	10000
Number of iterations for roulette wheel (N_3)	200
Lower limit of removable nodes ($\underline{s}$)	4
Upper limit of removable nodes ($\overline{s}$)	16
The increase rate of the ECM (ς)	300 s

We ran the algorithm, N_1 times each with N_2 iterations for each instance. The scores of the operators were updated every N_3 iterations. The removable nodes are randomly chosen between $\underline{s} = 4$ and $\overline{s} = 16$ at each iteration.

4.5.2 Bi-objective solution methods

We have tested the following four methods. In each, the ALNS is used as the search engine to find and store the non-dominated solutions.

1. WM: Here the objective is to minimise the sum of a weighted bi-objective function. The weights are increased from zero to one in increments of 0.1. The aggregated objective function is calculated as $wf_1 + (1-w)f_2$ where, f_1 is the fuel consumption (in L) and f_2 is the driving time (in h).

2. WMN: The objective functions are normalised using Nadir (z_i^N) and Utopian (z_i^U) points ($i = 1, 2$). The aggregated objective function is calculated as $w(f_1 - z_1^U)/(z_1^N - z_1^U) + (1 - w)(f_2 - z_2^U)/(z_2^N - z_2^U)$, where f_1 and f_2 are as defined above.

3. ECM: The algorithm is first run to find the value ϵ_2 which is the minimum value f_2 attains at the end of N_2 iterations. The arrival time to the depot is then fixed to ϵ_2. The algorithm is then run to minimise f_1, and ϵ_2 is increased by ς at every iteration.

4. HM: The algorithm starts as in ECM. However, the objective function is calculated as the weighted of two functions as $w(f_1 - z_1^U) + (1 - w)(f_2 - z_2^U)$. The weight w is updated during the algorithm. To update the weights during the search, we have used the same procedure as in Paquette et al. (2011):

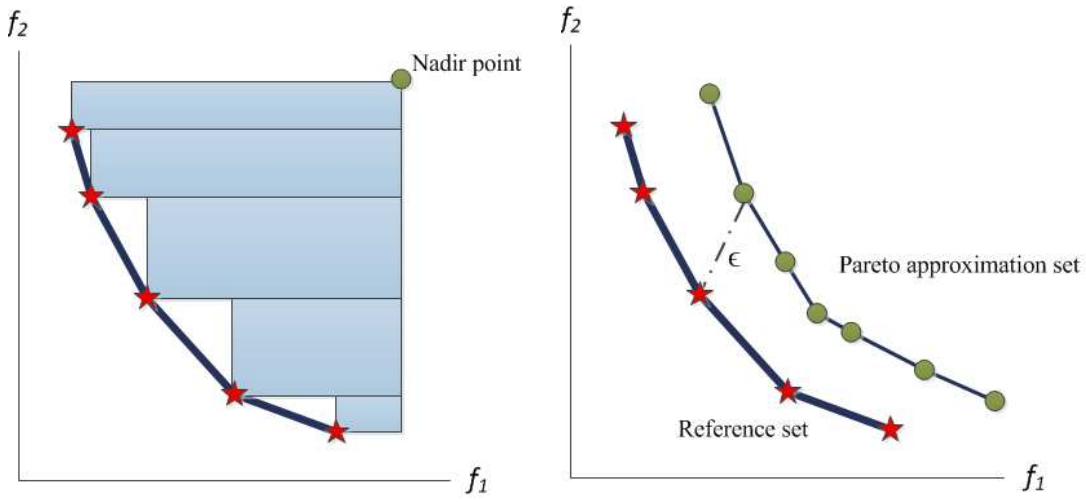
$$w_i = w_{i-1}(f_i(x) - z_i^U)\,(i = 1, ..., k) \tag{4.38}$$

$$w_h = w_h / \sum_{i=1}^{k} w_i\,(h = 1, ..., k), \tag{4.39}$$

where w_h is the normalised weight of the objective function h.

4.5.3 Solution quality indicators

The performance assessment of techniques in multi-objective optimisation is less straightforward than in single objective optimisation. Whereas the output of a single objective function can be compared directly with lower or upper bounds, the output of multi-objective optimisation is a set of solutions approximating the Pareto optimal front. Zitzler et al. (2003) present a review of the existing quality assessment indicators. We use two of these to compare our four methods, namely the hypervolume indicator and the ϵ-indicator (Figure 4.4). These are described in more detail below. We also use the number of Pareto solutions found as an additional way of assessing performance.



(a) Hypervolume indicator

(b) Epsilon indicator

FIGURE 4.4: Solution quality indicators

4.5.3.1 Hypervolume indicator

This metric was proposed by Zitzler and Thiele (1998*a*,*b*). The hypervolume indicator $I_{hv}(\mathcal{H})$ computes the volume of a given region H. It is depicted in Figure 4.4(a) for the bi-objective case. In this case, each point in the approximation set forms a rectangle shown by the shaded area with respect to a reference point (generally the Nadir point) that lies beyond the bounds of the approximation set. The hypervolume indicator is the area of the union of all rectangles (see Figure 4.4(a)). The larger the value of the indicator (area), the better is the set of solutions.

4.5.3.2 Epsilon indicator

The epsilon indicator $I_\epsilon(\mathcal{S},\mathcal{R})$ was proposed by Zitzler et al. (2003). An illustration of the epsilon indicator in two dimensions is given in Figure 4.4(b). The epsilon indicator relies on the concept of epsilon-dominance which is explained below:

Definition 4.1. Additive ϵ-dominance relation. Let $x_i, x_j \in \mathcal{S}$, x_i is said to additively ϵ-dominate x_j, and written as $x_j \succeq_{\epsilon+} x_i$ (or equivalently $f(x_i) \leq \epsilon + f(x_j)$).

Definition 4.2. Multiplicative ϵ-dominance relation. Let $x_i, x_j \in \mathcal{S}$, x_i is said to multiplicatively ϵ-dominate x_j, and written as $x_j \succeq_\epsilon x_i$ (or equivalently $f(x_i) \leq \epsilon f(x_j)$).

Of these two definitions, the multiplicative one is the most commonly used one in the literature. The ϵ-indicator is based on the weakly dominated dominance relation. For any method, the smaller is the ϵ value, the better is the performance of the method. The ϵ-indicator can be calculated as $I_\epsilon(\mathcal{S},\mathcal{R}) = \max_{r\in\mathcal{R}} \min_{s\in\mathcal{S}} \epsilon(s,r) = \max\{f_i(s) - r_i | 1 \leq i \leq k\}$, where r_i is the i^{th} component of the objective vector r, and $\mathcal{R}$ is the reference set defined here as the union of all known Pareto optimal solutions.

4.5.4 Results of the methods on PRP instances

In this section, we present the computational results obtained on PRP instances. Table 4.4 provides, for each method tested, the average CPU time required to solve all instances of each set. All values are averaged over the 10 instances of each set. In this table, the instance sets are grouped into five. Group I is characterised by homogeneous loads and loose time windows, whereas Group II has homogeneous loads but tight time windows. Group III instances have heterogeneous loads and loose time windows. Finally, Group IV and V instances have heterogeneous loads and tight time windows.

TABLE 4.4: Average CPU times of the four solution methods (in seconds)

Instance groups	Instance sets	WM	WMN	ECM	HM
I	1	198.4	199.5	196.6	198.7
	5	133.8	132.9	127.8	123.9
	9	120.8	119.4	107.6	93.9
Average		151.0	150.6	144.0	138.8
II	2	212.2	218.8	210.6	210.0
	6	148.2	147.2	135.5	131.6
	10	127.0	126.9	114.5	102.3
Average		162.5	164.3	153.5	148.0
III	3	237.0	240.9	235.8	232.7
	7	156.4	156.6	147.7	144.5
	11	132.3	132.0	127.1	119.7
Average		175.2	176.5	170.2	165.6
IV	4	246.4	248.8	248.1	245.3
	8	168.5	171.8	159.0	155.3
	12	137.9	138.9	125.9	115.2
Average		184.2	186.5	177.7	171.9
	13	152.6	153.7	147.1	141.9
Global average		167.0	168.3	160.2	155.0

As can be seen from Table 4.4, the methods are fast. Instances of 100 nodes are solved within 155 s on average for the HM method, based on around 110000 iterations in total. The data in Table 4.4 reveal that for any type of instance, the overall CPU times required by HM is smaller than that of the other methods. The ECM method has the second best performance in terms of CPU times. The average computation times of two weighting methods (WM and WMN) are almost the same. This is because these two methods require a similar computational effort to run. The average CPU time of Group I is around 138 s for HM whereas Group II is around 148 sec. The difference is due to the heterogeneous load distribution of customers. The effect of time windows can be seen from the performance of Group III instances, for which HM requires 165 s on average. The combined effect of time windows and load can be analysed by looking at the average CPU time of instances Group IV, which is about 171 s. Group IV, namely the instance set of thirteen, is solved in 142 sec.

Computational results of the performance measurements are summarised in the Table 4.5. This table presents, for each method, the number of Pareto solutions found, as well as the values of the hypervolume and the ϵ-indicators.

TABLE 4.5: Results of quality indicators on bi-objective PRP instances

	WM			WMN			ECM			HM		
Instance sets	# of Pareto solutions	$I_{hv}(\mathcal{S})$	$I_{\epsilon}(\mathcal{S},\mathcal{R})$	# of Pareto solutions	$I_{hv}(\mathcal{S})$	$I_{\epsilon}(\mathcal{S},\mathcal{R})$	# of Pareto solutions	$I_{hv}(\mathcal{S})$	$I_{\epsilon}(\mathcal{S},\mathcal{R})$	# of Pareto solutions	$I_{hv}(\mathcal{S})$	$I_{\epsilon}(\mathcal{S},\mathcal{R})$
1	3.5	21.44	1.1276	2.3	19.21	1.1264	17.4	117.22	1.1255	25.2	118.46	1.1271
5	5.7	67.52	1.1058	3.4	67.96	1.1056	33.8	302.98	1.1054	29.1	334.25	1.1071
9	6.6	440.08	1.1326	3.2	460.55	1.1334	48.9	3958.28	1.1330	60.5	4004.49	1.1340
2	12.3	44.47	1.1421	11.8	43.03	1.1484	19.7	61.97	1.1294	21.1	71.11	1.1227
6	18.1	359.63	1.1283	17.9	388.13	1.1267	28.5	711.99	1.1235	39.4	746.67	1.1081
10	25.0	1519.07	1.1324	19.7	1478.08	1.1407	49.4	4102.16	1.1413	62.7	4266.04	1.1225
3	3.7	40.52	1.1198	3.6	39.98	1.1196	26.4	156.81	1.1207	30.0	164.63	1.1250
7	4.4	70.92	1.1037	3.3	82.20	1.1031	26.0	331.50	1.1033	31.2	345.82	1.1019
11	4.7	441.61	1.1519	2.7	369.18	1.1532	42.2	3084.67	1.1551	65.7	3343.63	1.1541
4	11.5	94.72	1.1330	10.0	82.88	1.1412	23.2	194.40	1.1312	31.0	199.08	1.1285
8	18.6	165.01	1.1317	14.9	142.35	1.1314	28.5	208.28	1.1292	34.8	289.00	1.1203
12	20.2	1280.08	1.1483	14.5	1319.75	1.1499	48.6	3266.90	1.1481	58.6	3821.56	1.1299
13	16.6	176.57	1.1231	14.7	162.08	1.1304	25.7	293.10	1.1289	30.0	317.12	1.1131
Average	11.6	363.20	1.1293	9.4	358.11	1.1315	32.2	1291.56	1.1288	39.9	1386.30	1.1226

These results confirm that HM performs very well for all methods, with an average number of 40 Pareto solutions across all instances. The ECM yields the second best performance after HM. The number of Pareto solutions found are similar for WM and WMN, which are 11.6 and 9.4, respectively. We now compare the four methods using the two quality indicators. According to the hypervolume indicator, HM is superior to the other methods with an average value of 1386.3. In other words, the solutions found by HM represents a larger area than the other methods. The same indicator yields very poor results for WM and WMN. With the ϵ-indicator, HM once again exhibits the best performance, yielding the minimum value. The WM and ECM have a very similar performances based on ϵ-indicator, whereas WMN performs the worst.

We have also looked the effect of increasing a number of vehicles on the performance measure. For each method, Table 4.6 presents the number of Pareto optimal solutions found, the hypervolume and ϵ-indicators. In this table, the instance sets are grouped by the number of vehicles to see the effect on the resulting solutions.

The analysis of Table 4.6 shows that the average number of Pareto solutions increases with the number of vehicles. This is because the solution space is enlarged when more vehicles are needed. Finally, the number of successes of each method using the two indicators is reported in Table 4.7.

All results show that WM, WMN and ECM are clearly dominated by HM, as far as the success rate is concerned, indicating that a hybrid use of the existing methods is better suited to our problem. The weighting methods (WM and WMN) are better for finding extreme (corner) solutions, but they do not generate many Pareto solutions. On the other hand, the ϵ-constraint method finds more solutions but most of them are inferior to those provided by WM.

4.5.5 Details of the Pareto solutions

In this part of the analysis, we look at the nature of the Pareto solutions identified by the algorithm. For this purpose, we provide graphs of sample instances from each set for comparing the four methods in Figures 4.5–4.9, and vice versa in Figures 4.10–4.13. In these figures, the values on the x-axes represent the driving time objective (f_2) and the values on the y-axes show the fuel consumption (f_1).

Figures 4.5–4.9 exhibit similar behaviours for different type of instances. A recurring theme in the results presented in Figure 4.5 is that driving time can be decreased from about 52–56

TABLE 4.6: Results of quality indicators on benchmark instances grouped by the number of vehicles

	WM			WMN			ECM			HM		
Instance sets	# of Pareto solutions	$I_{hv}(\mathcal{S})$	$I_{\epsilon}(\mathcal{S},\mathcal{R})$	# of Pareto solutions	$I_{hv}(\mathcal{S})$	$I_{\epsilon}(\mathcal{S},\mathcal{R})$	# of Pareto solutions	$I_{hv}(\mathcal{S})$	$I_{\epsilon}(\mathcal{S},\mathcal{R})$	# of Pareto solutions	$I_{hv}(\mathcal{S})$	$I_{\epsilon}(\mathcal{S},\mathcal{R})$
1–4	7.8	50.29	1.1306	6.9	46.27	1.1339	21.7	132.60	1.1267	26.8	138.32	1.1258
5–8	11.7	165.77	1.1174	9.9	170.16	1.1167	29.2	388.69	1.1153	33.6	428.93	1.1094
9–12	14.1	920.21	1.1413	10.0	906.89	1.1443	47.3	3603.00	1.1444	61.9	3858.93	1.1351
13	16.6	176.57	1.1231	14.7	162.08	1.1304	25.7	293.10	1.1289	30.0	317.12	1.1131
Average	12.5	328.21	1.1281	10.4	321.35	1.1313	31.0	1104.35	1.1288	38.1	1185.83	1.1208

TABLE 4.7: Number of successes based on each instance set

Instance sets	WM		WMN		ECM		HM	
	$I_{hv}(\mathcal{S})$	$I_{\epsilon}(\mathcal{S},\mathcal{R})$	$I_{hv}(\mathcal{S})$	$I_{\epsilon}(\mathcal{S},\mathcal{R})$	$I_{hv}(\mathcal{S})$	$I_{\epsilon}(\mathcal{S},\mathcal{R})$	$I_{hv}(\mathcal{S})$	$I_{\epsilon}(\mathcal{S},\mathcal{R})$
1	0	2	0	4	5	2	5	2
2	0	2	0	0	3	3	7	5
3	0	2	1	4	3	3	6	1
4	0	1	0	2	3	3	7	4
5	1	2	0	4	4	3	5	1
6	0	0	0	1	4	2	6	7
7	0	1	0	2	3	4	7	3
8	1	2	0	1	3	2	6	5
9	0	2	0	3	7	4	3	1
10	0	2	0	1	6	0	4	7
11	0	5	0	3	2	1	8	1
12	0	1	0	0	3	0	7	9
13	0	1	0	2	1	1	9	6
Total	2	23	1	27	47	28	80	52

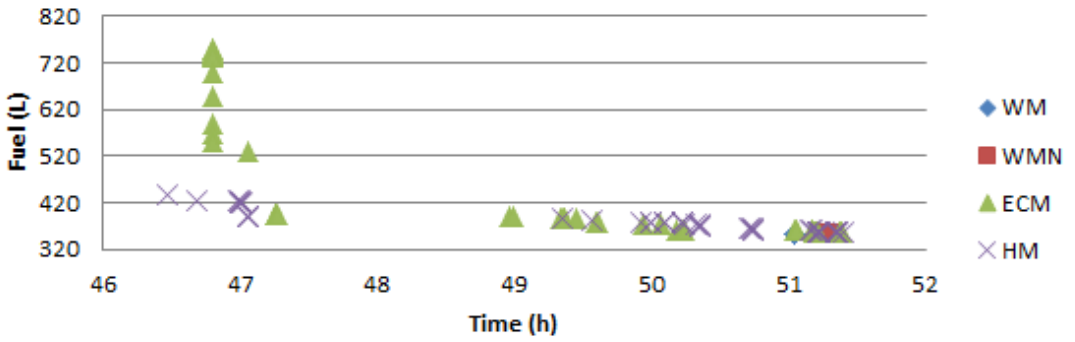


FIGURE 4.5: An instance from set 9

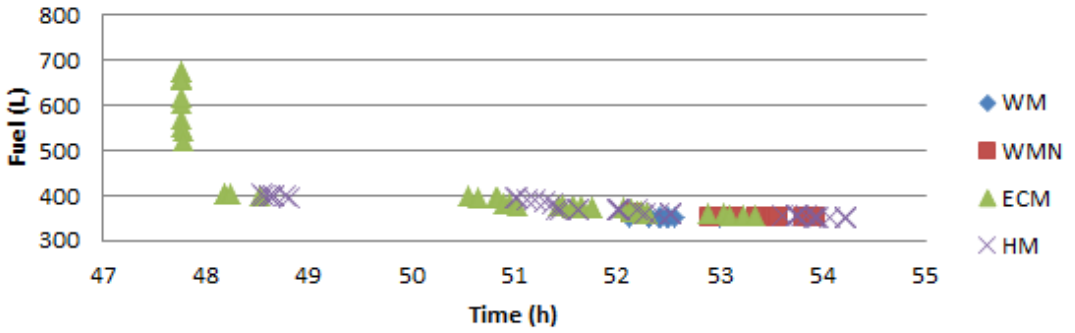


FIGURE 4.6: An instance from set 10

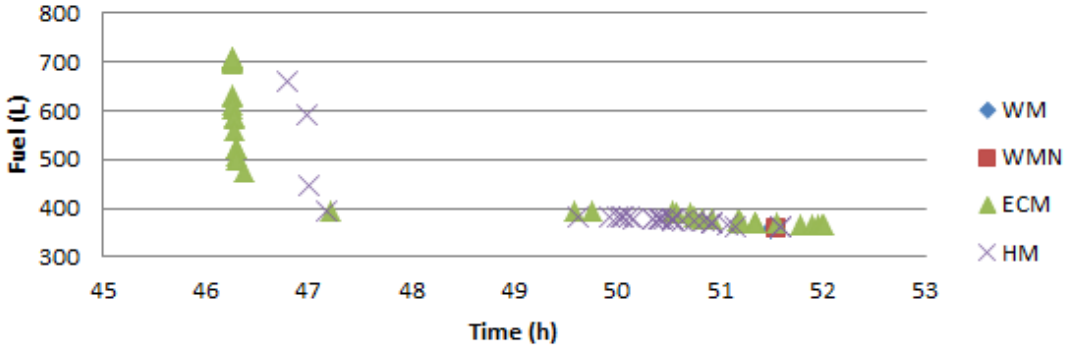


FIGURE 4.7: An instance from set 11

hours to 47–48 hours, depending on the instance tested, without much change in the fuel consumption. Conversely, fuel consumption can be brought down quite significantly, from around 700–800 litres to 400–500 litres with only a slight increase in driving time. This is particularly

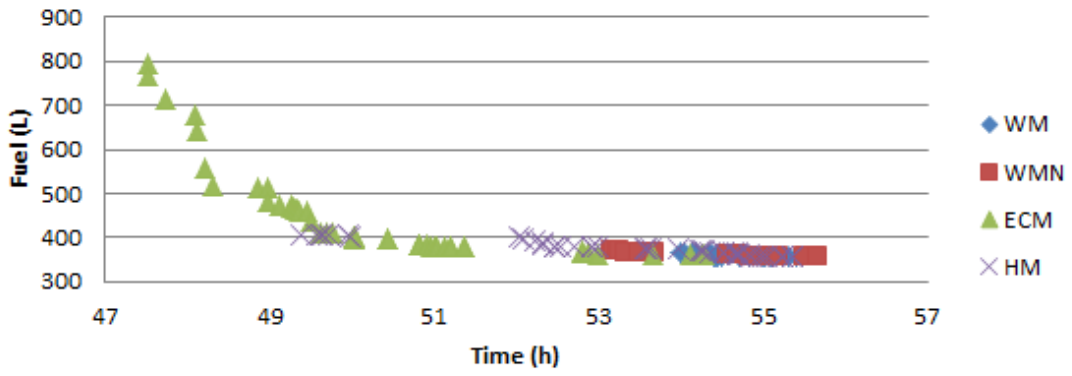


FIGURE 4.8: An instance from set 12

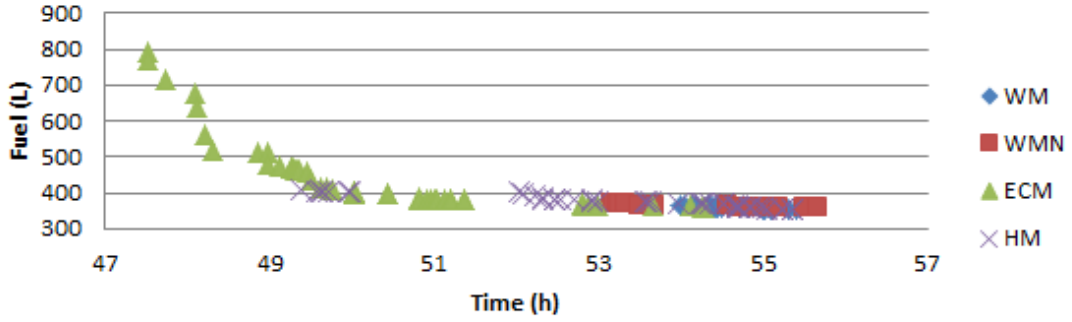


FIGURE 4.9: An instance from set 9

apparent in instances 9–11 where the number of vehicles is around 20. Furthermore, all four methods tested here are consistent with respect to the solutions generated and yield similar insights. Interestingly, these results on the trade-off between fuel consumption and driving time show that one does not necessarily need to sacrifice one objective heavily in order to improve the other. The tools presented in this chapter offer ways in which good quality solutions for the two objectives can be found.

Figures 4.10– 4.13 provide the results of this experiment performed with HM. The figures indicate that each type of instance has a different behaviour and yields different numbers of Pareto solutions. In Figure 4.10, the driving time can be decreased from about 24 to 21 hours, depending on the instance tested, with only a slight increase in the average fuel consumption for the five routes. As seen in Figures 4.11– 4.12, the total fuel consumption and driving time increase when more routes are needed. This means that more savings in fuel or driving time can be achieved. Figure 4.13 shows the behaviour of instance set 13 for which a reduction

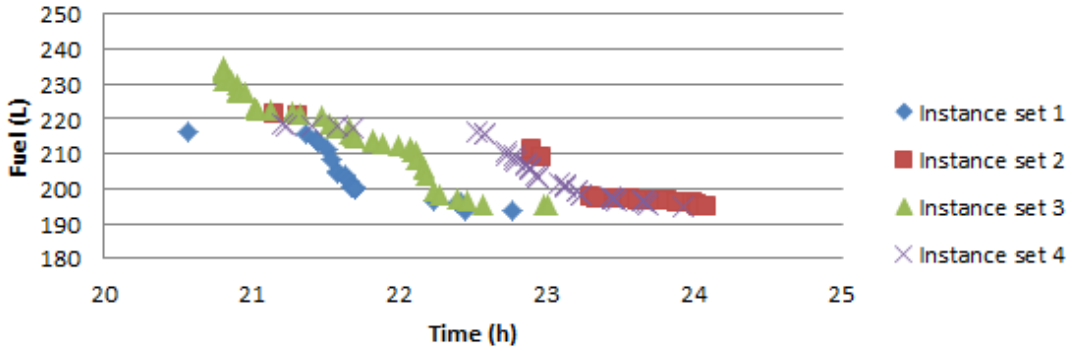


FIGURE 4.10: Instance sets 1–4

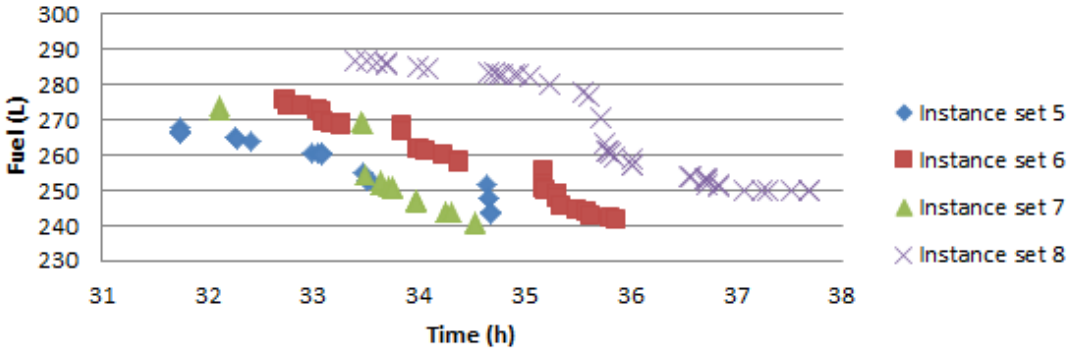


FIGURE 4.11: Instance sets 5–8

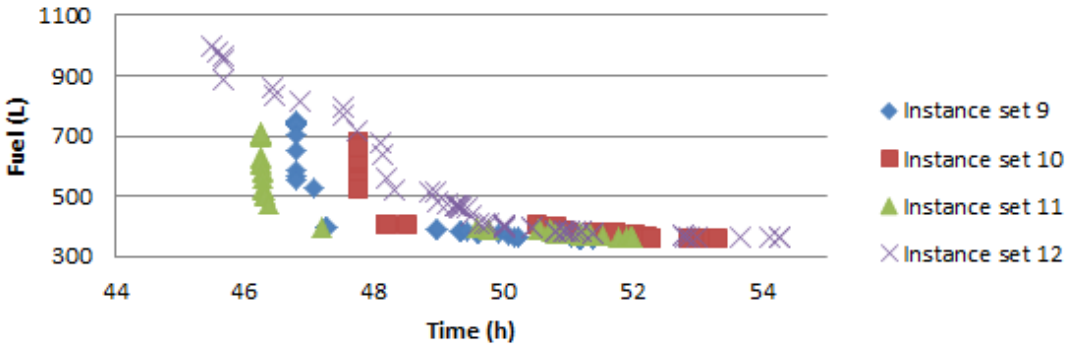


FIGURE 4.12: Instance sets 9–12

on fuel consumption from 500 to 280 litres (about 44%) may be achieved by increasing the driving time from 34.5 hours to 38.6 hours (about 12%).

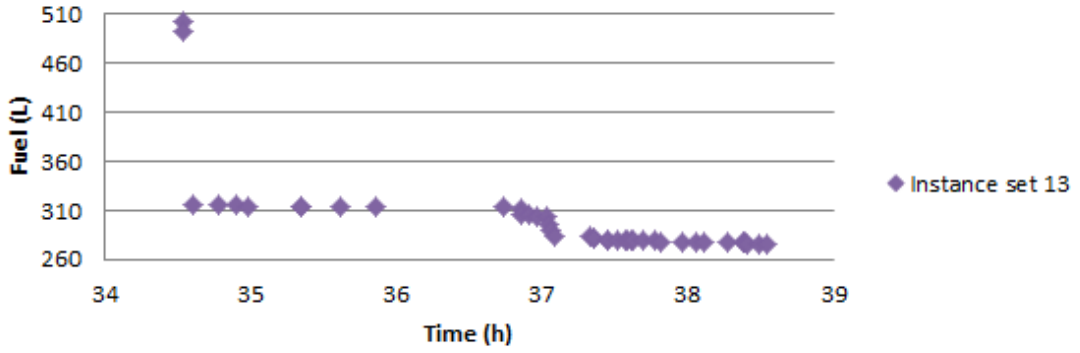


FIGURE 4.13: Instance set 13

4.5.6 Results for a sample 30-node instance

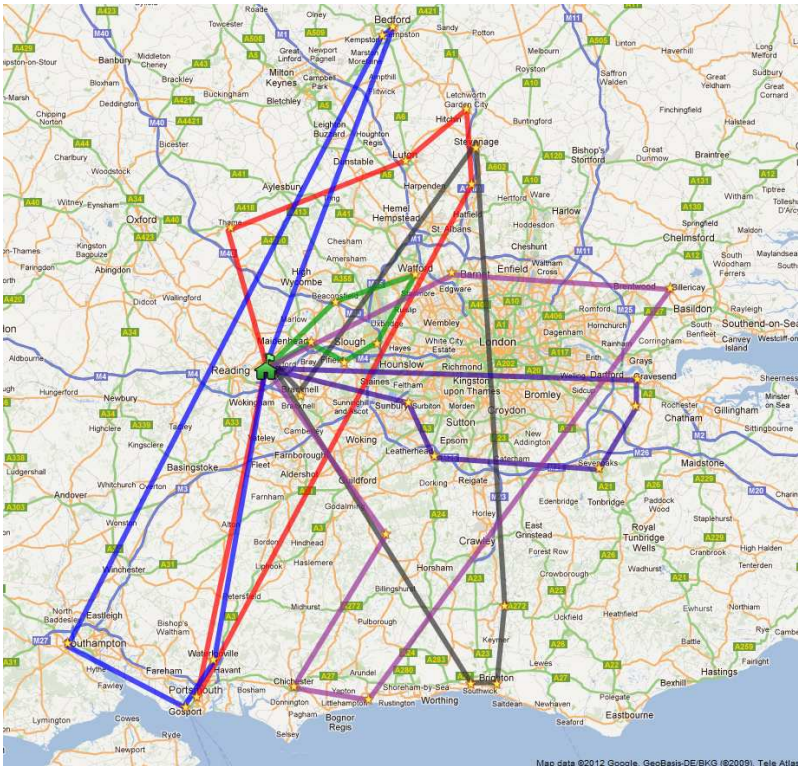
This section presents the results of a 30-node instance. Our aim is to take a closer look at the trade-off between the two objectives. Table 4.8 presents two non-dominated solutions obtained by HM on the total fuel consumption (in L), the total driving time (in h), the total fuel cost, the total driving time and the total amount of CO_2. In calculating the latter, we assume that a litre of diesel fuel produces 2.67 kg of CO_2 (Coe, 2005).

TABLE 4.8: Two non-dominated solutions of 30-node instance

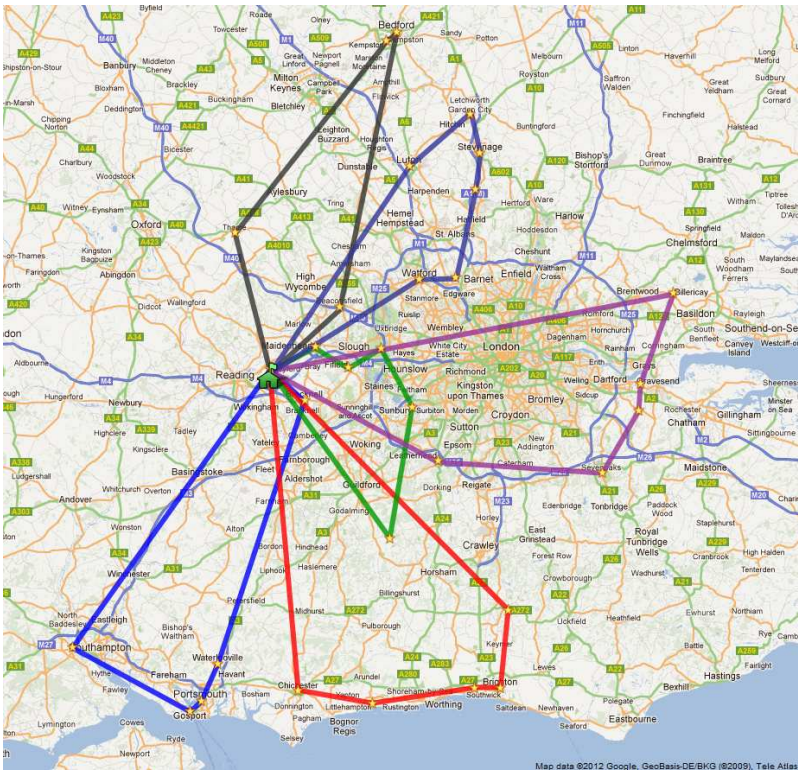
	# of routes	Total distance km	Fuel consumption L	Operational time h	CO_2 emissions kg	Fuel cost £	Driver cost £	Total cost £
Solution A	6	1621.7	321.57	21.16	858.59	450.20	169.28	619.48
Solution B	6	1270.1	233.54	23.21	623.55	326.96	185.68	512.64

Solutions A and B are depicted in Figure 4.14, with each route shown in a different shade. Solution A (Figure 4.14(a)) shows a time-minimizing tour of length 1621.7 km consuming approximately 321.57 L diesel fuel. The tour needs 21.16 hours to be traversed, and the total amount of CO_2 emitted is 858.59 kg. Solution B (Figure 4.14(b)) shows a fuel-minimizing tour of length 1270.1 km consuming around 233.54 L diesel fuel. The total amount of CO_2 for this tour is around 623.55 kg, with a driving time of 23.21 h and a total distance of 1270.1 km.

The trade-off between fuel consumption and time for this particular instance indicates that savings in energy can be achieved by increasing the total duration of routes. The 9.7% increase in driving time leads to a 27% saving in energy requirements. The reduction of CO_2 is around



(a) Solution A



(b) Solution B

FIGURE 4.14: Two non-dominated solutions for a 30-node instance

235 kg if Solution B is preferred to Solution A. In contrast, the reduction in driving time from 23.21 h to 21.16 h (about 8.8%) afforded by solution A implies an increase in CO_2 emissions of about 37.7%.

4.6 Conclusions

This chapter has studied the bi-criteria PRP in which one of the objectives is related to CO_2 emissions, and the other to driving time. An enhanced adaptive large neighbourhood search (ALNS) algorithm was proposed for the generation of non-dominated/Pareto optimal solutions. The algorithm integrates the classical ALNS with a specialised speed optimisation algorithm. The proposed algorithm first calls the ALNS using fixed speeds as inputs, then optimises speeds on each route.

Using the ALNS as the search engine, four *a posteriori* methods, namely the weighting method, the weighting method with normalisation, the epsilon-constraint method and a new hybrid method (HM), were tested using a scalarisation of the two objective functions. The HM combines an adaptive weighting with the epsilon-constraint method. To fully evaluate the effectiveness of the algorithm, new sets of instances based on real geographic data were generated, and a library of bi-criteria PRP instances was compiled. Results of extensive computational experimentation of the four methods were presented and compared with one another. Hypervolume and epsilon indicators were used to evaluate the performance of the four methods. Our results show that the HM is highly effective in finding good-quality non-dominated solutions on 100-node instances, both in terms of the hypervolume and of the epsilon indicators, as compared to the rest of the methods.

One interesting insight derived from the experiments is that one need not compromise greatly in terms of driving time in order to achieve a significant reduction in fuel consumption and CO_2 emissions. The converse of this insight also holds, i.e., considerable reductions in driving time can be gained if one is willing to increase fuel consumption only slightly. These results imply that the trade-offs between the conflicting objectives of fuel and time are not such that great sacrifices need to be made with respect to one objective in order to improve the other. The tools described in this chapter provide decision makers with a set of solutions from which they can choose. The multi-objective optimisation tools described for this particular problem are also general in their applicability in that they are independent of costs, such as those of fuel or driver wages, which may differ from one organisation to another.

Chapter 5

Conclusions

5.1 Overview

This thesis focused on environmental effects of road transportation and aimed to develop effective methodologies for planning vehicle routes that minimise CO_2 emitted by the vehicles. This final chapter of the thesis highlights the overall content of each of the three main chapters. It also presents the specific contributions of each chapter to the existing literature, limitations of the selected techniques and methodologies, and identifies various areas for further research.

5.2 Chapter II: A Comparative Analysis of Several Vehicle Emission Models for Freight Transportation

The second chapter of the thesis reviewed and compared several available vehicle emission models for freight transportation. The chapter presented six different fuel emission models and compared them with each other. A comparative study of fuel emission models does not seem to have done beforehand. The chapter can be seen as a first work to analyse and compare these models. Even though the chapter does not offer any new techniques, the comparison of the methods yielded important insights to understand the logic behind several vehicle fuel consumption models. Another important finding of the chapter was a list of all the factors affecting fuel consumption. The results showed that most suitable vehicle emission model is the one introduced by Barth et al. (2005); Scora and Barth (2006) and Barth and Boriboonsomsin (2008). Based on this insight, this model, namely the comprehensive modal emission model, was chosen as a tool for the fuel consumption estimator for the rest of the thesis.

Based on the extensive simulation results, vehicle speed and load are found to be the most prominent factors affecting fuel consumption. It is therefore important to travel at a speed that leads to minimum fuel consumption for a given routing plan. Vehicle load has also a significant effect on vehicle consumption, as fuel consumption increases with load. Vehicle type was found to be another important factor to be considered in fuel consumption. These findings suggest that light duty (LD) vehicles should be preferred over medium duty (MD) and heavy duty (HD) vehicles. MD should be also preferred to HD vehicle if possible. A positive road gradient leads to an increase in fuel consumption and it should be taken into account in route planning in future applications. Current GIS software can provide information of the road gradient. Resistance and drag should also be taken into account in the design of the vehicle and its accessories.

5.3 Chapter III: An Adaptive Large Neighbourhood Search Heuristic for the Pollution-Routing Problem

The third chapter of the thesis built upon the findings of chapter II and proposed a new methodology to generate more environmental-friendly vehicle routing plans. The insights obtained in chapter II have resulted in proposing a new methodology. The PRP is NP-hard and a simplified version of this problem cannot be solved to optimality for mid-size instances. This was the main motivation of this chapter. For this reason, we have developed a heuristic to obtain good-quality solutions within short computational times. In this chapter, we refer to two different problems. These are the pollution-routing problem which is a extension of a VRPTW and a speed optimization problem (SOP). An adaptive large neighbourhood search (ALNS) algorithm was proposed to solve the VRPTW and a speed optimization algorithm was applied.

The chapter proposed an adaptation of the ALNS to solve the PRP, for which the algorithm was enhanced with the introduction of new operators. These new operators were developed based mainly on speed and load to improve the solution quality. The ALNS uses 12 different removal and five insertion operators, which are selected dynamically in the algorithm according to their past performance. The design of selection of operators was also an important part of this research. The results suggested that the classical approach does not well perform well on the PRP due to limitations of the solution space. In this research, the selection of operators was based on penalising the best objective values to discover different parts of the solution space. This helped avoid local optima and to discover better results. The main algorithm proposed to solve the PRP was designed in an iterative way, where the ALNS uses fixed speeds as input to the VRPTW, following which the SOA is run on each route to improve the solution, and the process continues in an iterative way. In order to solve PRP as a VRPTW problem in the first stage of the solution process, the travelling time had to be assumed as fixed between pairs of nodes. The SOA was therefore applied to optimise the routes. The resulting algorithm is another contribution of research, which is fast and easy to implement for different types of routing problems.

The results of chapter suggest that this approach is fast and yields good-quality solutions in terms of the optimised objective values within a reasonable number of iterations. The proposed algorithm is able to solve instances with up to 200 customers which is a reasonable and practical figure for a standard routing plan.

5.4 Chapter IV: The Bi-Objective Pollution-Routing Problem

The fourth chapter of the thesis investigated a bi-objective PRP in which one of the objectives is related to fuel consumption and the other to driving time. The main motivation of this chapter was to look at the managerial implications of the PRP. Since fuel consumption and driving time are conflicting objectives, the trade-offs between these two were investigated in detail.

The research aims of this chapter were four-fold: to introduce of a bi-objective variant of the PRP, to apply and test existing multi-objective techniques for the solution of this variant, to describe and test a new hybrid heuristic for the problem, and to perform extensive computational experiments using four *a posteriori* methods evaluated by means of two well-known performance indicators and the number of Pareto solutions.

The objective function of the PRP in chapter III was set as an equally weighted sum of fuel consumption costs and wage of drivers. In the bi-objective PRP, two objective functions related to fuel consumption and driving time were treated separately. Four *a posteriori* methods were then used to generate solutions, three of which were existing methods but the fourth one is new.

The algorithm proposed in chapter IV is different from the one described in chapter III. In chapter III, the speed optimisation is run after 25K iterations and works in an iterative way. However, in chapter IV, it is applied at after each iteration. In order to use different sets of weights, the new approach ran for a preset number of iterations for each weight. The proposed algorithm returns a set of non-dominated Pareto solutions found in the course of algorithm. The number of Pareto solutions is also an important measure for assessing the quality of the solutions. To compare the multi-objective optimisation methods tested in the chapter, the hypervolume and ϵ-indicator measures were applied. Computational results indicated that best results were obtained by using the new method, namely the hybrid method (HM). The development of this technique to solve the bi-objective PRP could be regarded as an important contribution of the thesis.

The results of the chapter suggested that, the trade-offs between the conflicting objectives of fuel and time minimisation are such that no great sacrifices need to be made with respect to one objective in order to improve the other. An important contribution of the research is a technique that is able to produce different sets of solutions, from where a the decision maker may select from in order to reduce both fuel consumption (CO_2 emissions) and operational times. Our approach described for the bi-objective PRP is also general in the sense that it is independent of fuel or time-related costs which may differ from one organisation to another.

5.5 Limitations of the Thesis

It is acknowledged that there are some limitations and shortcomings of the research.

First, the main assumption of the PRP and the bi-objective PRP is that average speeds are used as an input without an explicit consideration of possible traffic congestion. It is known that CO_2 emissions increase with congestion. This research has offered, as a starting point, models and algorithms in which congestion is not considered, but the algorithms developed and described here could be modified to be able to take congestion into account.

Second, the actual fuel consumption depends on several factors, which have been discussed in chapter II. One of the factors that has not been considered here is the behaviour of drivers. Driving behaviour can influence the other factors investigated earlier. However, it is very difficult to quantify this parameter, to our knowledge, none of the models proposed in the literature takes this factor into account.

Fuel use during the cold start of an vehicle engine can be up to three times higher than the engine is warm. The effect of cold starts has not been considered in this research, which is the third limitation mentioned here. Some of the emission models introduced in chapter II consider the effect of cold starts but these models are inherently complex to be used in an algorithmic scheme for route planning.

Finally, the effect of stops on fuel consumption is not considered here. Energy requirements when service is being provided, either for collection or delivery, is assumed not to consume any fuel. In practice, the drivers tend to leave the engine on during service at customer sites, even though this is against the law in some countries.

5.6 Avenues for Further Research

To address the limitations introduced above, the following three areas were identified as further research directions.

First, the volume of emissions increases with travel time and lower speed levels. It is suggested that further research be conducted to minimise the effects of congestion in the context of the PRP. Research has already begun in this area, using some results of the present research. In particular, Franceschetti et al. (2012) use the PRP formulation and algorithms proposed in this research to minimise CO_2 emission with an explicit consideration of congestion.

Second, the current GIS software uses very simple regression models, based solely on distance travelled, to estimate fuel consumption. The present study could be integrated with such software to take into account fuel consumption for implementation in practice. This is an important development for at least two reasons: (*i*) the integration of the algorithms introduced here and GIS software could save both fuel and time, (*ii*) the interactive selection from a set of solutions generated by the proposed algorithm would allow for flexibility, productivity and support for route planners of freight companies. From this perspective, the last part of the research can be seen as a kind of decision support system application.

Third, the PRP research only focuses on the routing aspect of green logistics. Other problems which can be linked to routing may offer former reductions in emissions. For example, the facility location problem is concerned with physically locating a set of facilities (depots) so as to maximize the profit generated by providing service to a set of customers (Ghaddar and Naoum-Sawaya, 2011). Re-location of a depot may lead to reductions in CO_2 emissions. Ghaddar and Naoum-Sawaya (2011) show that the carbon emissions can be decreased significantly by incorporating a small penalty in the profits. A decrease of around 16% in carbon emissions results in less than 1% decrease in the profits. With about 8% decrease in profits, carbon emissions can be decreased by more than 40%. These figures are encouraging for further work on facility location problem.

Selection of the right vehicle from an available set of vehicles is another promising area to minimise CO_2 emissions. The effect of the characteristics of a vehicle on CO_2 emissions is partly studied in Chapter II but it is an area in need of more effort for comprehensive investigations. In the literature, the fleet size and mix vehicle routing problem consists of determining the type and the number of vehicles of each type with the minimisation of total costs (Jabali et al., 2012b). To our knowledge, there are no studies on this problem looking at fuel consumption and emissions in the way it was proposed in our work.

Fifth, the PRP assumes no pre-emption while processing any operation. This should be also considered within PRP formulation and methodology. As vehicle load is an important factor affecting fuel consumption, relaxing this assumption would offer an interesting line of research.

Finally, another promising application of the PRP is the field of vehicle scheduling, as opposed to vehicle routing. Driver working hours should be considered in the formulation of the PRP because of law requirements, as well as the acknowledged health hazards arising from intensive workload of routing plans. Although the proposed ALNS and SOA could be modified to reflect these type of real-life requirements, this remains a non-trivial task requiring further attention.

Appendix A

Detailed Computational Results

Table A.1: Computational results for 15-node instances

	CPLEX			Our heuristic								
Instances	Solution cost £	Total CPU time s	LP relaxation	Solution cost £	Distance km	# of vehicles	# of loops	ALNS time s	SOA time s	CPU time per loop s	Total CPU time s	Improvement %
UK15_01	287.72	10800*	124.51	286.89	713.2	2	4	1.3	0.0	1.3	5.1	0.29
UK15_02	209.93	10800*	99.69	209.12	517.8	2	4	1.0	0.0	1.0	4.0	0.39
UK15_03	280.896	10800*	136.90	280.89	714.1	3	6	0.9	0.0	0.9	5.4	0.00
UK15_04	296.55	10800*	128.27	296.56	746.1	3	4	0.8	0.0	0.8	3.3	0.00
UK15_05	284.94	10800*	132.84	284.94	726.5	2	4	0.9	0.0	0.9	3.8	0.00
UK15_06	233.68	10800*	110.15	234.40	551.9	3	5	0.8	0.0	0.8	4.1	−0.31
UK15_07	256.22	10800*	112.49	254.23	615.8	3	4	0.9	0.0	0.9	3.6	0.78
UK15_08	168.01	10800*	74.07	167.99	390.2	2	4	0.9	0.0	0.9	3.8	0.01
UK15_09	263.14	10800*	129.54	263.16	651.9	3	4	0.8	0.0	0.8	3.3	−0.01
UK15_10	216.85	10800*	109.91	215.66	536.0	2	4	0.9	0.0	0.9	3.7	0.55
UK15_11	258.95	10800*	126.84	258.95	645.0	2	4	0.9	0.0	0.9	3.6	0.00
UK15_12	310.90	10800*	152.72	311.02	781.2	3	5	0.9	0.0	0.9	4.4	−0.04
UK15_13	248.39	10800*	118.72	248.77	593.4	3	4	0.9	0.0	0.9	3.4	−0.15
UK15_14	332.27	10800*	165.06	332.26	848.1	3	5	0.9	0.0	0.9	4.5	0.00
UK15_15	222.26	10800*	99.31	222.27	549.4	2	5	0.9	0.0	0.9	4.7	0.00
UK15_16	205.73	10800*	94.99	205.73	490.3	2	5	0.9	0.0	0.9	4.5	0.00
UK15_17	282.66	10800*	142.49	282.65	684.5	3	4	0.8	0.0	0.8	3.4	0.00
UK15_18	315.15	10800*	129.54	315.75	786.7	3	4	0.9	0.0	0.9	3.4	−0.19
UK15_19	166.06	10800*	78.32	166.07	383.5	2	4	0.8	0.0	0.8	3.4	0.00
UK15_20	201.74	10800*	88.00	201.71	478.8	3	4	0.9	0.0	0.9	3.6	0.01

TABLE A.2: Computational results for 20-node instances

	CPLEX			Our heuristic								
Instances	Solution cost £	Total CPU time s	LP relaxation	Solution cost £	Distance km	# of vehicles	# of loops	ALNS time s	SOA time s	CPU time per loop s	Total CPU time s	Improvement %
UK20_01	323.18	10800*	146.83	323.65	784.8	3	5	1.5	0.0	1.5	7.4	−0.14
UK20_02	330.03	10800*	168.25	330.05	829.5	3	4	1.6	0.0	1.6	6.3	−0.01
UK20_03	208.09	10800*	92.02	207.65	459.2	3	4	1.4	0.0	1.4	5.8	0.21
UK20_04	324.88	10800*	161.30	324.74	798.5	3	4	1.3	0.0	1.3	5.4	0.04
UK20_05	296.24	10800*	130.38	295.54	717.6	3	4	1.4	0.0	1.4	5.7	0.23
UK20_06	359.19	10800*	169.24	349.16	840.4	3	4	1.3	0.0	1.3	5.1	2.79
UK20_07	228.83	10800*	101.00	228.05	524.0	3	4	1.5	0.0	1.5	6.0	0.34
UK20_08	277.30	10800*	129.93	277.23	657.5	3	5	1.3	0.0	1.3	6.4	0.03
UK20_09	321.72	10800*	156.87	321.68	798.6	3	4	1.6	0.0	1.6	6.5	0.01
UK20_10	294.59	10800*	128.65	292.90	704.1	3	8	1.4	0.0	1.4	11.5	0.57
UK20_11	369.38	10800*	188.71	363.80	885.1	3	4	1.3	0.0	1.3	5.2	1.51
UK20_12	315.85	10800*	138.62	315.80	777.0	3	4	1.4	0.0	1.4	5.8	0.02
UK20_13	314.38	10800*	151.06	308.94	770.2	3	4	1.6	0.0	1.6	6.4	1.73
UK20_14	409.40	10800*	183.52	405.50	1019.9	4	7	1.4	0.0	1.4	9.5	0.95
UK20_15	319.75	10800*	151.34	319.72	772.5	3	4	1.4	0.0	1.4	5.5	0.01
UK20_16	330.50	10800*	153.59	332.18	794.7	3	4	1.3	0.0	1.3	5.4	−0.51
UK20_17	360.85	10800*	186.45	358.41	886.0	3	4	1.4	0.0	1.4	5.7	0.68
UK20_18	353.10	10800*	168.56	345.28	840.2	3	5	1.3	0.0	1.3	6.7	2.22
UK20_19	321.65	10800*	150.25	321.66	790.3	3	5	1.4	0.0	1.4	6.8	0.00
UK20_20	325.46	10800*	161.07	325.44	806.0	3	4	1.5	0.0	1.5	6.1	0.01

TABLE A.3: Computational results for 25-node instances

	CPLEX			Our heuristic								
Instances	Solution cost £	Total CPU time s	LP relaxation	Solution cost £	Distance km	# of vehicles	# of loops	ALNS time s	SOA time s	CPU time per loop s	Total CPU time s	Improvement %
UK25_01	283.78	10800*	134.81	282.13	667.0	3	4	2.0	0.0	2.0	7.9	0.58
UK25_02	347.03	10800*	164.64	346.71	821.9	4	5	1.8	0.0	1.8	8.8	0.09
UK25_03	220.69	10800*	89.39	218.52	468.5	3	4	1.8	0.0	1.8	7.4	0.98
UK25_04	263.94	10800*	105.26	258.32	613.4	3	4	2.4	0.0	2.4	9.6	2.13
UK25_05	330.47	10800*	153.74	329.12	787.4	4	4	1.8	0.0	1.8	7.4	0.41
UK25_06	296.68	10800*	128.37	296.65	677.1	4	4	1.9	0.0	1.9	7.7	0.01
UK25_07	332.94	10800*	156.31	330.14	795.7	3	8	1.9	0.0	1.9	15.5	0.84
UK25_08	344.02	10800*	154.87	344.79	840.8	3	4	1.9	0.0	1.9	7.6	−0.22
UK25_09	306.17	10800*	131.89	302.81	679.9	4	4	2.0	0.0	2.0	8.0	1.10
UK25_10	362.80	10800*	174.40	358.66	856.4	4	4	1.8	0.0	1.8	7.2	1.14
UK25_11	376.01	10800*	175.09	370.91	884.9	4	6	1.6	0.0	1.6	9.3	1.36
UK25_12	402.00	10800*	189.61	400.68	965.9	4	5	1.9	0.0	1.9	9.4	0.33
UK25_13	248.32	10800*	102.99	241.53	527.2	4	10	2.1	0.0	2.1	20.5	2.73
UK25_14	375.73	10800*	181.78	375.58	907.3	4	4	2.2	0.0	2.2	8.6	0.04
UK25_15	367.10	10800*	172.48	362.65	903.4	3	6	2.2	0.0	2.2	13.0	1.21
UK25_16	344.80	10800*	162.46	340.70	822.3	4	5	2.2	0.0	2.2	11.0	1.19
UK25_17	451.18	10800*	222.65	451.20	1107.6	4	5	1.7	0.0	1.7	8.5	0.00
UK25_18	376.75	10800*	183.56	376.76	922.8	3	4	2.0	0.0	2.0	7.9	0.00
UK25_19	406.69	10800*	200.31	402.88	989.1	4	6	2.0	0.0	2.0	11.8	0.94
UK25_20	366.16	10800*	170.72	356.16	828.3	3	9	1.5	0.0	1.5	13.4	2.73

TABLE A.4: Computational results for 50-node instances

	CPLEX			Our heuristic								
Instances	Solution cost £	Total CPU time s	LP relaxation	Solution cost £	Distance km	# of vehicles	# of loops	ALNS time s	SOA time s	CPU time per loop s	Total CPU time s	Improvement %
UK50_01	600.47	10800*	266.26	593.77	1363.4	7	5	5.9	0.0	5.9	29.7	1.12
UK50_02	614.18	10800*	266.26	599.43	1403.9	7	10	6.0	0.0	6.0	60.3	2.40
UK50_03	640.28	10800*	290.75	626.21	1460.8	7	11	4.9	0.0	4.9	53.4	2.20
UK50_04	754.78	10800*	367.43	740.92	1774.5	8	7	5.1	0.0	5.1	35.8	1.84
UK50_05	644.53	10800*	296.40	636.00	1496.5	6	6	5.9	0.0	5.9	35.3	1.32
UK50_06	603.35	10800*	261.95	584.61	1294.5	8	10	5.5	0.0	5.5	54.6	3.11
UK50_07	552.08	10800*	246.02	541.07	1231.0	7	4	6.4	0.0	6.4	25.7	1.99
UK50_08	573.49	10800*	261.01	560.27	1286.3	7	7	5.6	0.0	5.6	39.5	2.30
UK50_09	697.26	10800*	324.57	687.79	1605.5	7	4	5.4	0.0	5.4	21.4	1.36
UK50_10	698.01	10800*	318.69	670.92	1572.1	7	5	5.1	0.0	5.1	25.6	3.88
UK50_11	638.25	10800*	290.98	618.94	1481.2	7	4	6.3	0.0	6.3	25.1	3.03
UK50_12	589.59	10800*	254.42	571.42	1344.0	7	7	5.8	0.0	5.8	40.7	3.08
UK50_13	596.75	10800*	277.72	589.11	1344.5	7	10	5.2	0.0	5.2	52.1	1.28
UK50_14	663.99	10800*	319.44	660.17	1580.6	7	6	5.8	0.0	5.8	35.0	0.57
UK50_15	618.80	10800*	275.20	584.13	1383.3	6	5	5.2	0.0	5.2	26.2	5.60
UK50_16	590.89	10800*	262.52	585.16	1365.9	7	9	5.7	0.0	5.7	51.0	0.97
UK50_17	480.85	10800*	197.44	456.56	973.9	7	4	5.0	0.0	5.0	20.0	5.05
UK50_18	707.33	10800*	313.23	681.72	1585.3	8	5	5.3	0.0	5.3	26.4	3.62
UK50_19	613.70	10800*	261.46	597.95	1370.1	7	4	5.3	0.0	5.3	21.2	2.57
UK50_20	680.44	10800*	316.72	678.56	1623.0	7	6	4.8	0.0	4.8	28.9	0.28

TABLE A.5: Computational results for 75-node instances

	CPLEX			Our heuristic								
Instances	Solution cost £	Total CPU time s	LP relaxation	Solution cost £	Distance km	# of vehicles	# of loops	ALNS time s	SOA time s	CPU time per loop s	Total CPU time s	Improvement %
UK75_01	1008.88	10800*	453.22	961.77	2245.4	11	8	11.8	0.0	11.8	94.1	4.67
UK75_02	891.36	10800*	372.58	836.71	1854.3	11	4	10.1	0.0	10.1	40.6	6.13
UK75_03	890.48	10800*	404.59	858.66	1978.0	10	4	14.4	0.0	14.4	57.6	3.57
UK75_04	829.58	10800*	344.20	792.87	1719.0	11	4	11.0	0.0	11.0	44.1	4.42
UK75_05	949.26	10800*	419.12	884.65	2095.7	10	5	12.7	0.0	12.7	63.5	6.81
UK75_06	962.05	10800*	421.01	911.45	2101.6	11	4	11.8	0.0	11.8	47.1	5.26
UK75_07	1005.74	10800*	446.19	955.55	2207.7	11	7	11.7	0.0	11.7	81.7	4.99
UK75_08	1008.46	10800*	439.76	821.54	1908.2	10	4	13.4	0.0	13.4	53.6	18.54
UK75_09	970.47	10800*	445.51	920.88	2167.8	10	6	13.6	0.0	13.6	81.6	5.11
UK75_10	999.37	10800*	454.72	969.63	2278.2	11	6	11.2	0.0	11.2	67.3	2.98
UK75_11	731.48	10800*	295.70	681.42	1452.6	10	6	10.9	0.0	10.9	65.4	6.84
UK75_12	876.92	10800*	390.43	846.17	1972.6	10	5	12.8	0.0	12.8	63.8	3.51
UK75_13	1014.85	10800*	458.02	969.13	2297.4	10	5	11.5	0.0	11.5	57.7	4.51
UK75_14	971.75	10800*	434.67	914.38	2153.1	10	4	12.5	0.0	12.5	50.1	5.90
UK75_15	1055.13	10800*	467.45	977.55	2292.3	10	4	11.8	0.0	11.8	47.1	7.35
UK75_16	971.15	10800*	429.56	927.29	2145.2	10	7	11.6	0.0	11.6	81.2	4.52
UK75_17	954.53	10800*	414.71	900.70	2089.2	11	7	11.0	0.0	11.0	76.8	5.64
UK75_18	911.00	10800*	391.29	839.85	1927.3	10	6	12.3	0.0	12.3	73.7	7.81
UK75_19	885.19	10800*	375.94	830.94	1906.1	10	10	12.3	0.0	12.3	123.0	6.13
UK75_20	947.70	10800*	409.88	902.09	2074.3	11	11	12.7	0.0	12.7	139.5	4.81

TABLE A.6: Computational results for 150-node instances

	CPLEX			Our heuristic								
Instances	Solution cost £	Total CPU time s	LP relaxation	Solution cost £	Distance km	# of vehicles	# of loops	ALNS time s	SOA time s	CPU time per loop s	Total CPU time s	Improvement %
UK150_01	1879.34	10800*	629.3	1437.79	3086.8	20	5	58.0	0.0	58.0	290.2	23.49
UK150_02	2069.98	10800*	773.4	1694.03	3821.9	20	6	64.9	0.0	64.9	389.7	18.16
UK150_03	1732.39	10800*	650.6	1488.93	3265.2	19	4	56.5	0.0	56.5	226.0	14.05
UK150_04	2009.32	10800*	757.7	1663.36	3790.0	21	8	30.7	0.0	30.7	245.6	17.22
UK150_05	1803.21	10800*	661.2	1500.62	3320.2	20	8	60.4	0.0	60.4	483.2	16.78
UK150_06	1840.38	10800*	662.7	1506.64	3310.1	21	13	56.1	0.0	56.1	729.8	18.13
UK150_07	2142.94	10800*	789.6	1731.06	3859.3	21	7	46.0	0.0	46.0	321.9	19.22
UK150_08	1840.84	10800*	706.7	1575.02	3460.0	20	5	82.6	0.0	82.6	413.2	14.44
UK150_09	2073.50	10800*	757.9	1673.56	3796.8	20	4	76.9	0.0	76.9	307.8	19.29
UK150_10	1971.42	10800*	744.5	1650.39	3706.5	20	5	47.5	0.0	47.5	237.3	16.28
UK150_11	2186.55	10800*	789.6	1715.91	3904.3	20	5	59.1	0.0	59.1	295.3	21.52
UK150_12	2086.94	10800*	815.7	1784.55	4047.3	21	4	72.9	0.0	72.9	291.5	14.49
UK150_13	1958.40	10800*	748.5	1642.59	3732.4	19	6	45.6	0.0	45.6	273.9	16.13
UK150_14	2000.61	10800*	753.3	1675.23	3792.0	20	4	96.9	0.0	96.9	387.8	16.26
UK150_15	1862.28	10800*	640.7	1444.44	3132.5	19	4	63.4	0.0	63.4	253.6	22.44
UK150_16	2160.75	10800*	771.1	1673.76	3756.1	20	11	21.3	0.0	21.3	233.9	22.54
UK150_17	2102.53	10800*	784.7	1656.78	3743.8	20	5	136.8	0.0	136.8	684.0	21.20
UK150_18	2082.04	10800*	770.7	1668.09	3823.1	20	6	49.7	0.0	49.7	298.1	19.88
UK150_19	2046.87	10800*	861.2	1809.18	4167.2	20	6	60.7	0.0	60.7	364.3	11.61
UK150_20	2139.95	10800*	818.5	1753.00	4035.6	20	4	58.5	0.0	58.5	233.9	18.08

Bibliography

Akçelik, R. (1982), *Progress in fuel consumption modelling for urban traffic management*, Australian Road Research Board Report 124.

Akçelik, R. and Besley, M. (1996), SIDRA 5 User Guide, Technical report, ARRB Transport Research Ltd., Vermont South, Australia.

Akçelik, R. and Besley, M. (2003), Operating cost, fuel consumption, and emission models in aaSIDRA and aaMOTION, *in* '25th Conference of Australian Institutes of Transport Research (CAITR 2003)', Adelaide, Australia.

Apaydin, O. and Gonullu, M. T. (2008), 'Emission control with route optimization in solid waste collection process: A case study', *Sadhana* **33**(2), 71–82.

Ardekani, S., Hauer, E. and Jamei, B. (1996), Traffic Impact Models, *in* 'Traffic Flow Theory', US Federal Highway Administration, Washington, D.C., pp. 1–7.

Augerat, P., Belenguer, J. M., Benavent, E., Corberán, A. and Naddef, D. (1998), 'Separating capacity constraints in the CVRP using tabu search', *European Journal of Operational Research* **106**(2-3), 546–557.

Barth, M. and Boriboonsomsin, K. (2008), 'Real-world CO_2 impacts of traffic congestion', *Transportation Research Record: Journal of the Transportation Research Board* **2058**(1), 163–171.

Barth, M. and Boriboonsomsin, K. (2009), 'Energy and emissions impacts of a freeway-based dynamic eco-driving system', *Transportation Research Part D* **14**(6), 400–410.

Barth, M., Younglove, T. and Scora, G. (2005), Development of a Heavy-Duty Diesel Modal Emissions and Fuel Consumption Model, Technical report, UC Berkeley: California Partners for Advanced Transit and Highways (PATH).

Bauer, J., Bektaş, T. and Crainic, T. G. (2010), 'Minimizing greenhouse gas emissions in intermodal freight transport: An application to rail service design', *Journal of the Operational Research Society* **61**(3), 530–542.

Bektaş, T. and Laporte, G. (2011), 'The pollution-routing problem', *Transportation Research Part B: Methodological* **45**(8), 1232–1250.

Bowyer, D. P., Biggs, D. C. and Akçelik, R. (1985), 'Guide to fuel consumption analysis for urban traffic management', *Australian Road Research Board Transport Research* **32**.

Bristow, A. L., Pridmore, A. M., Tight, M. R. and May, A. D. (2004), Low carbon transport futures: how acceptable are they?, *in* 'World Conference on Transportation Research', Istanbul, Turkey.

Caramia, M. and Guerriero, F. (2009), 'A heuristic approach to long-haul freight transportation with multiple objective functions', *Omega* **37**(3), 600–614.

Chankong, V. and Haimes, Y. Y. (1983), 'Optimization-based methods for multiobjective decision-making: An overview', *Large Scale Systems* **5**(1), 1–33.

Chen, Y. W., Wang, C. H. and Lin, S. J. (2008), 'A multi-objective geographic information system for route selection of nuclear waste transport', *Omega* **36**(3), 363–372.

Christie, J. S. and Satir, S. (2006), Saving our energy sources and meeting Kyoto emission reduction targets while minimizing costs with application of vehicle logistics optimization, *in* '2006 Annual Conference and Exhibition of the Transportation Association of Canada: Transportation without boundaries', Charlottetown, Prince Edward Island.

Clarke, G. and Wright, J. W. (1964), 'Scheduling of vehicles from a central depot to a number of delivery points', *Operations Research* **12**(4), 568–581.

Clarkson, R. and Deyes, K. (2002), Estimating the social cost of carbon emissions, Technical report. Available at: http://62.164.176.164/d/SCC.pdf (accessed on July 1, 2012).

Coe, E. (2005), Average carbon dioxide emissions resulting from gasoline and diesel fuel, Technical report, United States Environmental Protection Agency. Available at: http://www.etieco.com/content-files/EPA%20emissions%20calc%20420f05001.pdf (accessed on July 1, 2012).

Coello, C. A. C., Lamont, G. B. and Van Veldhuizen, D. A. (2002), *Evolutionary algorithms for solving multi-objective problems*, Kluwer, Dordrecht.

Conti, J., Chase, N. and Maples, J. (2010), US greenhouse gas emissions in the transportation sector, *in* D. Sperling and J. S. Cannon, eds, 'Climate and Transportation Solutions: Findings from the 2009 Conference on Transportation and Energy Policy', Institute of Transportation Studies, University of California, Davis, pp. 24–32.

Cordeau, J.-F., Gendreau, M., Laporte, G., Potvin, J.-Y. and Semet, F. (2002), 'A guide to vehicle routing heuristics', *Journal of the Operational Research Society* **53**(5), 512–522.

Cordeau, J.-F., Laporte, G., Savelsbergh, M. W. P. and Vigo, D. (2007), Vehicle routing, *in* C. Barnhart and G. Laporte, eds, 'Transportation: Vol.14 of Handbooks on Operations Research and Management Science', Elsevier, Amsterdam, pp. 367–428.

Crainic, T. G. (2000), 'Service network design in freight transportation', *European Journal of Operational Research* **122**(2), 272–288.

Crainic, T. G. and Laporte, G. (1997), 'Planning models for freight transportation', *European Journal of Operational Research* **97**(3), 409–438.

Dantzig, G. B. and Ramser, J. H. (1959), 'The truck dispatching problem', *Management Science* **6**(1), 80–91.

Deb, K. (2001), *Multi-Objective Optimization Using Evolutionary Algorithms*, Wiley, Chichester.

Deb, K. and Chaudhuri, S. (2007), 'I-mode: an interactive multi-objective optimization and decision-making using evolutionary methods', *Lecture Notes in Computer Science* **4403**, 788–802.

DECC (2010), '2009 Greenhouse gas emissions, final figures by end-user'. available at: http://decc.gov.uk/en/content/cms/statistics/climate_change/data/data.aspx (accessed on July 1, 2012).

Demir, E., Bektaş, T. and Laporte, G. (2011), 'A comparative analysis of several vehicle emission models for road freight transportation', *Transportation Research Part D: Transport and Environment* **6**(5), 347–357.

Demir, E., Bektaş, T. and Laporte, G. (2012*a*), 'An adaptive large neighborhood search heuristic for the Pollution-Routing Problem', Forthcoming in the *European Journal of Operational Research* .

Demir, E., Bektaş, T. and Laporte, G. (2012*b*), The bi-objective pollution-routing problem, Technical report, School of Management, University of Southampton, Southampton.

Eglese, R. W. and Black, D. (2010), Optimizing the routing of vehicles, *in* A. Mckinnon, S. Cullinane, M. Browne and A. Whiteing, eds, 'Green Logistics: Improving the environmental sustainability of logistics', Great Britain/KoganPage, pp. 215–228.

Eglese, R. W., Maden, W. and Slater, A. (2006), 'A road timetable^{TM} to aid vehicle routing and scheduling', *Computers & Operations Research* **33**(12), 3508–3519.

Ehrgott, M. and Gandibleux, X. (2002), *Multiple Criteria Optimization: State of the Art annotated Bibliographic Surveys*, Kluwer Academic, Dordrecht.

Ehrgott, M. and Ryan, D. M. (2002), 'Constructing robust crew schedules with bicriteria optimization', *Journal of Multi-Criteria Decision Analysis* **11**(3), 139–150.

Erlandsson, L., Almen, J. and Johansson, H. (2008), Measurement of emissions from heavy duty vehicles meeting Euro IV/V emission levels by using on-board measurement in real life operation, *in* '16th International Symposium Transport and Air Pollution, Graz, Austria'.

Esteves-Booth, A., Muneer, T., Kubie, J. and Kirby, H. (2002), 'A review of vehicular emission models and driving cycles', *Proceedings of the Institution of Mechanical Engineers, Part C: Journal of Mechanical Engineering Science* **216**(8), 777–797.

Everall, P. F. (1968), The effect of road and traffic conditions on fuel consumption, Technical report, Ministry of Transport, Crowthorne, United Kingdom.

Fagerholt, K., Laporte, G. and Norstad, I. (2010), 'Reducing fuel emissions by optimizing speed on shipping routes', *Journal of the Operational Research Society* **61**(3), 523–529.

Figliozzi, M. A. (2010), 'An iterative route construction and improvement algorithm for the vehicle routing problem with soft time windows', *Transportation Research Part C: Emerging Technologies* **18**(5), 668–679.

Fonseca, C. M. and Fleming, P. J. (1995), 'An overview of evolutionary algorithms in multiobjective optimization', *Evolutionary Computation* **3**(1), 1–16.

Forkenbrock, D. J. (1999), 'External costs of intercity truck freight transportation', *Transportation Research Part A: Policy and Practice* **33**(7–8), 505–526.

Forkenbrock, D. J. (2001), 'Comparison of external costs of rail and truck freight transportation', *Transportation Research Part A* **35**(4), 321–337.

Fowkes, T. and Whiteing, T. (2006), The value of freight travel time savings and reliability improvements – recent evidence from Great Britain, *in* 'European Transport Conference (ETC)', Association for European Transport, Strasbourg, France.

Franceschetti, A., Van Woensel, T., Honhon, D., Bektaş, T. and Laporte, G. (2012), The time-dependent pollution routing problem, Technical report, Industrial engineering and innovation Sciences, Technology University of Eindhoven, Eindhoven.

Genta, G. (1997), *Motor Vehicle Dynamics: Modelling and Simulation*, World Scientific Publishing, Singapore.

Ghaddar, B. and Naoum-Sawaya, J. (2011), 'Environmentally friendly facility location with market competition', *Journal of the Operational Research Society* .

Golden, B. L., Raghavan, S. and Wasil, E. (2008), *The vehicle routing problem: latest advances and new challenges*, Springer.

Grodzevich, O. and Romanko, O. (2006), Normalization and other topics in multi-objective optimization, In Proceedings of the Fields MITACS Industrial Problems Workshop, Toronto.

Hickman, J., Hassel, D., Joumard, R., Samaras, Z. and Sorenson, S. (1999), MEET-Methodology for calculating transport emissions and energy consumption. European Commission, DG VII, Technical report, ISBN 92-828-6785-4, Luxembourg, 362 p. Available at: http://www.transport-research.info/Upload/Documents/200310/meet.pdf (accessed on July 1, 2012).

Hvattum, L. M., Norstad, I., Fagerholt, K. and Laporte, G. (2012), 'Analysis of an exact algorithm for the vessel speed optimization problem', Forthcoming in the *Networks* .

IBM ILOG (2009), 'Copyright ©International Business Machines Corporation 1987'.

INFRAS (1995), Workbook on emission factors for road transport: explanatory notes, Technical report, Bern.

Jabali, O., Gendreau, M. and Laporte, G. (2012b), 'A continuous approximation model for the fleet composition problem', Forthcoming in the *Transportation Research Part B: Methodological* .

Jabali, O., Van Woensel, T. and de Kok, A. G. (2012a), 'Analysis of travel times and CO_2 emissions in time-dependent vehicle routing', Forthcoming in the *Production and Operations Management* .

Jozefowiez, N., Semet, F. and Talbi, E. (2008*a*), From single-objective to multi-objective vehicle routing problems: Motivations, case studies, and methods, *in* 'The Vehicle Routing Problem: Latest Advances and New Challenges', Vol. 43 of *Operations Research/Computer Science Interfaces Series*, Springer US, pp. 445–471.

Jozefowiez, N., Semet, F. and Talbi, E. G. (2008*b*), 'Multi-objective vehicle routing problems', *European Journal of Operational Research* **189**(2), 293–309.

Kara, I., Kara, B. Y. and Yetiş, M. K. (2007), Energy minimizing vehicle routing problem, *in* A. Dress, Y. Xu and B. Zhu, eds, 'Combinatorial Optimization and Applications, volume 4616 of Lecture Notes in Computer Science', Springer, Berlin/Heidelberg, pp. 62–71.

Kent, J. H., Post, K. and Tomlin, J. A. (1982), Fuel Consumption and Emission Modelling in Traffic Links, *in* 'Proceedings of SAE-A/ARRB 2nd Conference on Traffic Energy and Emissions', Melbourne, Australia.

Kirby, H. R. (2006), Modelling how changes in the driving cycle affect fuel consumption and carbon dioxide emission rates: the first stage, *in* 'World Sustainable Development Outlook 2006. Global and Local Resources in Achieving Sustainable Development. Conference of the World Association for Sustainable Development (4th)', Inderscience, Geneva, Switzerland, pp. 294–304.

Kirby, H. R., Hutton, B., McQuaid, R. W., Raeside, R. and Zhang, X. (2000), 'Modelling the effects of transport policy levers on fuel efficiency and national fuel consumption', *Transportation Research Part D: Transport and Environment* **5**(4), 265–282.

Kirkpatrick, S., Gelatt, C. D. and Vecchi, M. P. (1983), 'Optimization by simulated annealing', *Science* **220**(4598), 671–680.

Knörr, W. (2009), Ecological transport information tool- Environmental methodology and data, Technical report, Institut fur Energie und Umweldforschnag, Heidelberg.

Komhyr, W. D., Harris, T. B., Waterman, L. S., Chin, J. F. S. and Thoning, K. W. (1989), Atmospheric carbon dioxide at Mauna Loa Observatory. I- NOAA global monitoring for climatic change measurements with a nondispersive infrared analyzer, Technical report.

Kuo, Y. (2010), 'Using simulated annealing to minimize fuel consumption for the time-dependent vehicle routing problem', *Computers & Industrial Engineering* **59**(1), 157–165.

Kuo, Y. and Wang, C. (2011), 'Optimizing the VRP by minimizing fuel consumption', *Management of Environmental Quality: An International Journal* **22**(4), 440–450.

Laporte, G. (2007), 'What you should know about the vehicle routing problem', *Naval Research Logistics* **54**(8), 811–819.

Laporte, G., Desrochers, M. and Nobert, Y. (1984), 'Two exact algorithms for the distance-constrained vehicle routing problem', *Networks* **14**(1), 161–172.

Lattin, W. C. and Utgikar, V. P. (2007), 'Transition to hydrogen economy in the United States: A 2006 status report', *International Journal of Hydrogen Energy* **32**(15), 3230–3237.

Laumanns, M., Thiele, L. and Zitzler, E. (2006), 'An efficient, adaptive parameter variation scheme for metaheuristics based on the epsilon-constraint method', *European Journal of Operational Research* **169**(3), 932–942.

Lee, T. R. and Ueng, J. H. (1999), 'A study of vehicle routing problems with load-balancing', *International Journal of Physical Distribution & Logistics Management* **29**(10), 646–657.

Léonardi, J. and Baumgartner, M. (2004), 'CO_2 efficiency in road freight transportation: Status quo, measures and potential', *Transportation Research Part D* **9**(6), 451–464.

Maden, W., Eglese, R. W. and Black, D. (2009), 'Vehicle routing and scheduling with time-varying data: A case study', *Journal of the Operational Research Society* **61**(3), 515–522.

Malandraki, C. and Daskin, M. S. (1992), 'Time dependent vehicle routing problems: Formulations, properties and heuristic algorithms', *Transportation Science* **26**(3), 185–200.

Mavrotas, G. (2009), 'Effective implementation of the [epsilon]-constraint method in multi-objective mathematical programming problems', *Applied Mathematics and Computation* **213**(2), 455–465.

McKinnon, A. (2007), 'CO_2 Emissions from Freight Transport in the UK', *Report prepared for the Climate Change Working Group of the Commission for Integrated Transport* . Available at: http://resilientcities.gaiaspace.org/mushin/files/-1/7/2007climatechange-freight.pdf (accessed on July 1, 2012).

Miettinen, K. (1999), *Nonlinear Multiobjective Optimization*, Kluwer, Norwell.

Milthers, N. P. M. (2009), Solving VRP using Voronoi diagrams and adaptive large neighborhood search, Master's thesis, University of Copenhagen, Denmark.

Nam, E. K. and Giannelli, R. A. (2005), Fuel consumption modeling of conventional and advanced technology vehicles in the physical emission rate estimator (PERE), Technical report.

Norstad, I., Fagerholt, K. and Laporte, G. (2010), 'Tramp ship routing and scheduling with speed optimization', *Transportation Research Part C: Emerging Technologies* **19**(5), 853–865.

Ntziachristos, L. and Samaras, Z. (2000), COPERT III computer programme to calculate emissions from road transport: methodology and emission factors (version 2.1), Technical report, European Environment Agency, Copenhagen, Denmark. Available at: http://lat.eng.auth.gr/copert/files/tech50.pdf (accessed on July 1, 2012).

Pacheco, J. and Martí, R. (2005), 'Tabu search for a multi-objective routing problem', *Journal of the Operational Research Society* **57**(1), 29–37.

Palmer, A. (2007), The development of an integrated routing and carbon dioxide emissions model for goods vehicles, PhD thesis, Cranfield University, Bedford, United Kingdom.

Paquette, J., Cordeau, J.-F., Laporte, G. and Pascoal, M. M. B. (2011), Combining multicriteria analysis and tabu search for dial-a-ride problems, Working paper, CIRRELT, Montréal.

Pareto, V. (1971), *Manual of Political Economy*, A. M. Kelley. Macmillan, London. (First published 1909).

Pisinger, D. and Ropke, S. (2005), A general heuristic for vehicle routing problems, Technical report, DIKU - Department of Computer Science, University of Copenhagen. Available at: http://www.diku.dk/hjemmesider/ansatte/sropke/Papers/GeneralVRP_TechRep.pdf (accessed on July 1, 2012).

Pisinger, D. and Ropke, S. (2007), 'A general heuristic for vehicle routing problems', *Computers & Operations Research* **34**(8), 2403–2435.

Price, R., Thornton, S. and Nelson, S. (2007), The social cost of carbon and the shadow price of carbon: What they are, and how to use them in economic appraisal in the UK, Technical report. Available at: http://archive.defra.gov.uk/evidence/series/documents/shadowpriceofcarbondec-0712.pdf (accessed on July 1, 2012).

Qian, F., Gribkovskaia, I., Laporte, G. and Øyvind, H. (2011), 'Passenger and pilot risk minimization in offshore helicopter transportation', *Omega* **40**(5), 584–593.

Qian, J. (2012), Fuel emission optimization in vehicle routing problems with time-varying speeds, PhD thesis, Lancaster University, Lancaster, United Kingdom.

Rangaiah, G. P. (2009), *Multi-Objective Optimization: Techniques and Applications in Chemical Engineering*, Vol. 1, World Scientic: Singapore.

Ribeiro, R. and Ramalhinho-Lourenço, H. (2001), A multi-objective model for a multi-period distribution management problem, Technical report, Department of Economics and Business, Universitat Pompeu Fabra, Barcelona.

Rondinelli, D. and Berry, M. (2000), 'Multimodal transportation, logistics, and the environment: managing interactions in a global economy', *European Management Journal* **18**(4), 398–410.

Ropke, S. and Pisinger, D. (2006*a*), 'An adaptive large neighborhood search heuristic for the pickup and delivery problem with time windows', *Transportation Science* **40**(4), 455–472.

Ropke, S. and Pisinger, D. (2006*b*), 'A unified heuristic for a large class of vehicle routing problems with backhauls', *European Journal of Operational Research* **171**(3), 750–775.

Ross, M. (1994), 'Automobile fuel consumption and emissions: effects of vehicle and driving characteristics', *Annual Review of Energy and the Environment* **19**(1), 75–112.

Sbihi, A. and Eglese, R. W. (2007*a*), 'Combinatorial optimization and green logistics', *4OR: A Quarterly Journal of Operations Research* **5**(2), 99–116.

Sbihi, A. and Eglese, R. W. (2007*b*), 'The Relationship between Vehicle Routing & Scheduling and Green Logistics-A Literature Survey', *Lancaster University Management School* .

Schittler, M. (2003), State-of-the-art and emerging truck engine technologies for optimized performance, emissions and life cycle costs, Technical report, DaimlerChrysler AG (US).

Scora, G. and Barth, M. (2006), Comprehensive modal emission model (CMEM), version 3.01, user's guide, Technical report. Available at: http://www.cert.ucr.edu/cmem/docs/CMEM_User_Guide_v3.01d.pdf (accessed on July 1, 2012).

Scott, C., Urquhart, N. and Hart, E. (2010), 'Influence of Topology and Payload on CO_2 Optimised Vehicle Routing', *Applications of Evolutionary Computation* pp. 141–150.

Shaw, P. (1998), Using constraint programming and local search methods to solve vehicle routing problems, *in* 'Proceedings of the 4th International Conference on Principles and Practice of Constraint Programming', Springer, New York, pp. 417–431.

Suzuki, Y. (2011), 'A new truck-routing approach for reducing fuel consumption and pollutants emission', *Transportation Research Part D: Transport and Environment* **16**(1), 73–77.

Tavares, G., Zsigraiova, Z., Semiao, V. and Carvalho, M. (2009), 'Optimisation of MSW collection routes for minimum fuel consumption using 3D GIS modelling', *Waste Management* **29**(3), 1176–1185.

Tavares, G., Zsigraiova, Z., Semiao, V. and da Graça Carvalho, M. (2008), 'A case study of fuel savings through optimisation of MSW transportation routes', *Management of Environmental Quality: An International Journal* **19**(4), 444–454.

Tight, M. R., Bristow, A. L., Pridmore, A. and May, A. D. (2005), 'What is a sustainable level of CO_2 emissions from transport activity in the UK in 2050?', *Transport Policy* **12**(3), 235–244.

Toth, P. and Vigo, D. (2001), *The vehicle routing problem*, Society for Industrial and Applied Mathematics Philadelphia, PA, USA.

Ubeda, S., Arcelus, F. J. and Faulin, J. (2011), 'Green logistics at Eroski: A case study', *International Journal of Production Economics* **131**(1), 44–51.

Urquhart, N., Hart, E. and Scott, C. (2010), Building low CO_2 solutions to the vehicle routing problem with Time Windows using an evolutionary algorithm, *in* 'Evolutionary Computation (CEC), 2010 IEEE Congress', IEEE, pp. 1–6.

Urquhart, N., Scott, C. and Hart, E. (2010), 'Using an Evolutionary Algorithm to Discover Low CO_2 Tours within a Travelling Salesman Problem', *Applications of Evolutionary Computation* pp. 421–430.

Veldhuizen, D. A. V. and Lamont, G. B. (2000), 'Multiobjective evolutionary algorithms: Analyzing the state-of-the-art', *Evolutionary computation* **8**(2), 125–147.

Vira, C. and Haimes, Y. Y. (1983), *Multiobjective Decision Making: Theory and Methodology*, North-Holland, New York.

Wierzbicki, A. P. (1999), Reference point approach, *in* 'Multicriteria Decision Making: Advances in MCDM Models, Algorithms, Theory, and Applications', Kluwer, Dordrecht, pp. 9–39.

Xiao, Y., Zhao, Q., Kaku, I. and Xu, Y. (2012), 'Development of a fuel consumption optimization model for the capacitated vehicle routing problem', *Computers & Operations Research* **39**(7), 1419–1431.

Yang, C., McCollum, D., McCarthy, R. and Leighty, W. (2009), 'Meeting an 80% reduction in greenhouse gas emissions from transportation by 2050: A case study in California', *Transportation Research Part D* **14**(3), 147–156.

Yong, P. and Xiaofeng, W. (2009), Research on a vehicle routing schedule to reduce fuel consumption, *in* 'Measuring Technology and Mechatronics Automation', Vol. 3, pp. 825–827.

Zhou, A., Qu, B. Y., Li, H., Zhao, S. Z., Suganthan, P. N. and Zhang, Q. (2011), 'Multiobjective evolutionary algorithms: A survey of the state-of-the-art', *Swarm and Evolutionary Computation* **1**(1), 32–49.

Zitzler, E., Deb, K. and Thiele, L. (2000), 'Comparison of multiobjective evolutionary algorithms: Empirical results', *Evolutionary Computation* **8**(2), 173–195.

Zitzler, E. and Thiele, L. (1998*a*), An evolutionary algorithm for multiobjective optimization: The strength pareto approach, Technical report, TIK Report, Zurich, Switzerland.

Zitzler, E. and Thiele, L. (1998*b*), Multiobjective optimization using evolutionary algorithms: A comparative case study, *in* 'Parallel problem solving from nature-PPSN V', number 1498, Springer-Verlag, Amsterdam, pp. 292–301.

Zitzler, E., Thiele, L., Laumanns, M., Fonseca, C. M. and da Fonseca, V. G. (2003), 'Performance assessment of multiobjective optimizers: An analysis and review', *Evolutionary Computation, IEEE Transactions* **7**(2), 117–132.

Zografos, K. G. and Androutsopoulos, K. N. (2004), 'A heuristic algorithm for solving hazardous materials distribution problems', *European Journal of Operational Research* **152**(2), 507–519.