Stockholms universitet

Vehicle fleet optimization at a leading Swedish wholesaler and service supplier: Problem formulation and implementation

Optimering av fordonsflottan hos en ledande svensk grossist och tjänsteleverantör: problemformulering och implementering

Elin Forsberg

Handledare: Lars Arvestad

Examinator: Marc Hellmuth

Inlämningsdatum: 2025-01-05

Kandidatuppsats i datalogi
Bachelor Thesis in Computer Science

Abstract

In this report, a vehicle fleet optimization problem at a leading Swedish wholesaler and service supplier is formulated and analyzed with the purpose of reducing the fleet overcapacity. The relation to established routing and scheduling problems is discussed and three heuristic methods, Simulated Annealing, Tabu Search and Genetic Algorithms, are implemented. The methods are evaluated based on their ability to reduce the fleet size, while still adhering to the constraints of the problem. The results indicate that the Genetic Algorithm produces the best solutions and that it is possible to reduce the capacity violations of the original solution. Due to limitations in data, conclusions regarding the overcapacity cannot be stated and is a task recommended for further research with additional data.

Sammanfattning

I den här rapporten formuleras och analyseras ett fordonsoptimeringsproblem hos en ledande svensk grossist och tjänsteleverantör med målet att reducera flottans överkapacitet. Relationen till etablerade rutt- och schemaläggningsproblem diskuteras och tre heuristiska metoder, Simulerad Härdning, Tabu-sökning och Genetiska Algoritmer, implementeras. Metoderna utvärderas baserat på deras möjlighet att reducera fordonsflottans storlek, medan de fortfarande följer de krav problemet förutsätter. Resultatet visar att den Genetiska Algoritmen producerar de bästa lösningarna och att det är möjligt att reducera kapacitetsöverträdelserna i orginallösningen. På grund av begränsningar i tillgänglig data kan inte slutsatser kring överkapaciteten dras och lämnas istället till vidare undersökningar med ytterligare data.

Contents

1 Introduction

The company, denoted as WoS, together with its subsidiaries, operates in the food and logistics sectors, providing services to hotels, restaurants, stores and other establishments. Every day, from warehouses located in different parts of Sweden, thousands of deliveries are conducted to ensure both customer satisfaction and progress towards environmental goals. These tasks involve challenges such as smart route planning and effective operation.

1.1 Current approach at the company

The current way of operating the vehicle fleet at WoS, in regards to deliveries to their customers, consists of two parts. The first step includes construction of routes that fulfill the demands set by their customers, while still remaining within certain constraints selected by WoS. This is currently well executed at the company and results in suitable routes being created and implemented.

The second step consists of assigning a vehicle from their vehicle fleet to each specific route, with the constraint that each vehicle already is assigned to a specified depot. This is done in their route management system as well as manually, partially relying on the experience of their employees to create a satisfactory solution.

This way of working currently results in WoS having a larger fleet size than intended, since they report an overcapacity exceeding the desired level at ten percent. As a result the question of whether or not the overcapacity can be reduced arises.

The purpose of this thesis is therefore to analyze the current situation at WoS and to examine different optimization techniques applicable to this problem. Furthermore, the goal is to be able to produce an approximate optimal solution to propose solutions or strategies to lower the overcapacity.

1.2 Prerequisites

In order to customize the solution specifically to WoS, they provided a data set collected from their organization. The received data consists of two lists. The first includes pre-optimized routes executed during a week in October 2024, where each route has the following attributes: name, home-depot, date, duration, necessary vehicle capacity and registration number of the vehicle that executed the route. The second list consists of all available vehicles in WoS's vehicle fleet, including attributes: registration number, current home-depot, euro class, fuel-type and capacity.

The received data had not been formatted, and as a result, certain attributes of specific vehicles or routes were either missing or not representative of reality. The solution to this problem, agreed upon together with WoS, was to eliminate the routes and vehicles with extreme deviations and replace missing pieces of data with the mean of the respective column.

Another problem concerning the received data is that the starting times for each route were not available. In agreement and cooperation with WoS, the starting times were therefore constructed from knowledge of the general distribution of the route starting times. This was done by generating random numbers from a uniform distribution and for each number a corresponding random start time was generated within the represented percentage slot.

These solution methods allowed the majority of the dataset to be used and enabled a better representation of reality, compared to not having any knowledge regarding the starting times. In addition, these solutions also created a degree of uncertainty early on in the process. They may have introduced inaccuracies that must be considered when later making conclusions.

1.3 Problem formulation

In this section the problem at WoS, henceforth referred to as the WoS Problem (WSP), will be formulated and explained both in general and mathematically.

1.3.1 General description

The WSP optimization problem can be described as follows. One has a heterogeneous vehicle fleet, a fleet consisting of different types of vehicles, and a set of routes the fleet has to cover. Each route belongs to a specific depot and has to be executed at a specific time. In addition, all routes have their own capacity constraints needed to be fulfilled by the assigned vehicle. When determining where to place each vehicle, certain vehicles should be prioritized, such as if it is an electrical vehicle or by their euro class.

Given these conditions and constraints the desire is to assign routes to vehicles in order to obtain a solution that optimizes the fleet, primarily regarding size but also considering vehicle type. The result of this can then be used to reduce the overcapacity within the fleet by analyzing the solution and adapting the current fleet assignments.

1.3.2 Mathematical definition

The WSP is characterized by several constraints and trade-offs. In order to optimize the problem it is necessary to have a mathematical understanding and overview of it. The mathematical formulation of this problem is therefore specified in this section.

Starting with the following definitions:

Sets	Definition
V	The set of available vehicles, $\|V\| = n$
R	The set of routes, $\|R\| = m$
Constants	Definition
n	Number of available vehicles
m	Number of routes
I_j	Time interval for route j
e_i	Integer representing the type of vehicle i
c_i	Capacity of vehicle i
d_j	Demand of route j
depot_j	The depot route j originates from
Variables	Definition
y_i	Binary variable such that $y_i = \begin{cases} 1 & \text{if vehicle } i \text{ is used} \\ 0 & \text{otherwise} \end{cases}$
x_{ij}	Binary variable such that $x_{ij} = \begin{cases} 1 & \text{if vehicle } i \text{ is assigned route } j \\ 0 & \text{otherwise} \end{cases}$
t_{jk}	Binary variable such that $t_{jk} = \begin{cases} 0 & \text{if } I_k \cap I_j = \emptyset \\ 1 & \text{otherwise} \end{cases}$
h_{jk}	Binary variable such that $x_{ij} = \begin{cases} 0 & \text{if } \text{depot}_j = \text{depot}_k \\ 1 & \text{otherwise} \end{cases}$

Table 1: Table of definitions for the WSP

the aim is to minimize

$$\begin{aligned} F &= \sum_{i=1}^{n} y_i + \sum_{i=1}^{n} y_i \cdot e_i \\ &= \sum_{i=1}^{n} y_i \Big(1 + e_i \Big) \end{aligned} \tag{1}$$

such that (2)-(6) hold.

$$\sum_{i=1}^{n} x_{ij} = 1 \ , \quad \forall j \tag{2}$$

$$y_i = \min\left(1, \sum_{j=1}^{m} x_{ij}\right) \ , \quad \forall i \tag{3}$$

$$\sum_{j=1}^{m} d_j x_{ij} \leq c_i \ , \ \ \forall i \tag{4}$$

$$t_{jk} + x_{ij} + x_{ik} - 2 \leq 0 \ , \quad \forall i, \forall j, \forall k, j \neq k \tag{5}$$

$$h_{jk} + x_{ij} + x_{ik} - 2 \leq 0 \ , \quad \forall i, \forall j, \forall k \tag{6}$$

In this formulation the objective function F in (1) minimizes the number of vehicles and prioritizes certain vehicle types. Equation (2) ensures that each route is assigned exactly one vehicle, equation (3) that if a vehicle is assigned a route it must be used, equation (4) that there are no capacity constraints violations, equation (5) that there is no time overlap of two routes assigned to the same vehicle and equation (6) that all vehicles and its assigned routes originate from the same depot.

This mathematically captures the different aspects of the problem, enabling it to be optimized with all constraint being addressed.

1.4 Previous research and Related theoretical problems

This section discusses previous research regarding similar problems, such as the Vehicle routing problem and Vehicle scheduling problem, and their connection to the WSP.

1.4.1 The Vehicle Routing Problem

The Vehicle Routing Problem (VRP) was first introduced by G. B. Dantzig and J. H. Ramser in 1959 [1] and has since resulted in a substantial amount of research. The problem, although well-studied and implemented, has, because of the many constraints needed to fit an actual real life scenario, no one universally accepted definition. Instead the research has mainly focused on the standardized version of the VRP, recognizing that many of the implemented algorithms can be customized to fit the needs of several problem formulations with additional constraints. The standardized, also referred to as classical, VRP consists of constructing m vehicles routes that start and end at a depot such that each customer is served by exactly one vehicle, the total demand of each route does not exceed the total capacity and simultaneously minimizing the total routing cost [5]. The problem is defined mathematically in Definition 1.

Definition 1. The Vehicle Routing Problem [5]

Let $G = (V, E)$ be an undirected graph with vertex set $V = \{0, 1, ..., n\}$ and edge set $E = \{(i, j) \mid i, j \in V, i \neq j\}$. Vertex 0 represents the depot at which there are located at most m identical vehicles of capacity Q. With each customer $i \in V \setminus \{0\}$ is associated a non-negative demand $q_i \leq Q$. A cost matrix c_{ij} is defined on E. The problem is then:

Minimize

$$\sum_{[i,j] \in E} c_{ij} x_{ij} \tag{7}$$

subject to

$$\sum_{j \in V \setminus \{0\}} x_{0j} = 2m \ , \tag{8}$$

$$\sum_{i<k} x_{ik} + \sum_{k<j} x_{kj} = 2 \ \ (k \in V \setminus \{0\}) \ , \tag{9}$$

$$\sum_{i \in S, j \notin S \text{ or } i \notin S, j \in S} x_{ij} \geq 2b(S) \ \ (S \subset V \setminus \{0\}) \ , \tag{10}$$

$$x_{ij} = 0 \text{ or } 1 \ \ (i, j \in V \setminus \{0\}) \ , \tag{11}$$

$$x_{0j} = 0, 1 \text{ or } 2 \ \ (j \in V \setminus \{0\}) \ . \tag{12}$$

Where x_{ij} is an integer variable representing the number of times edge $[i, j]$ appears in the optimal solution and $b(S)$ is a lower bound on the number of vehicles required to serve all customers of S.

In Definition 1 equation (8) ensures that the same number of vehicles leaving the depot are also returning. Equation (9) ensures that exactly one vehicle visits each customer, since the constraint concludes that there are exactly two incident edges of each customer vertex. Equation (10) ensures that for each subset of customers there is both a trip to and from the subset for each vehicle assigned to a customer in the subset and it must be larger than the smallest amount of vehicles required to serve all customers in S. Lastly, equations (11) and (12) describe the integer variable x_{ij}.

A modification of the VRP is the Capacitated VRP (CVRP) and it is the most studied version of the problem. It introduces a capacity constraint where each customer has a demand and where the sum of the demands of each customer on a route cannot override the capacity of the assigned vehicle [7]. Other variants of the VRP include the VRP with Time Windows (VRPTW), introducing a time constraint for each customer [4], or the Green VRP (GVRP), which aims to address issues regarding sustainable transportation with a goal of balancing the trade-off between economic and environmental concerns while still producing effective routes [11]. A fourth variant is the Multi-depot VRP (MDVRP) where the set of depots is not equal to one and therefore vehicles can depart from different depots [7].

Another alternative variant of the VRP, that is constructed to consider several additional layers of complexity and therefore enabling the problem to be customized to fit certain real-life situations, is the Multi-Depot Heterogeneous Fleet Periodic Capacitated VRP with Time Windows (MDHFPCVRPTW). It has been found that the success of finding a solution to such a problem can depend on how one formulates it mathematically. The number of indices used on the variables effected the solutions [10].

When dealing with multi-objective VRPs, for example with both financial and environmental goals, one solution is to combine the two objectives and use weights to create a single-objective function. However, these approaches heavily depend on the assigned weights and to escape that problem one can instead chose a solution technique able to consider multi-objectives [7].

1.4.2 The Vehicle Scheduling Problem

The Vehicle Scheduling Problem (VSP) has been studied extensively during the last 50 years and therefore has many different formulation approaches. In general, the problem consists of the following. Given a set of routes with fixed departure time, arrival time, start location and end location, find an assignment of routes to vehicles such that each route is covered exactly once, each vehicle is assigned a feasible sequence of routes and the total cost is minimized. This can mathematically be represented as finding the minimum cost flow of a network described in Definition 2 [6].

Definition 2. The Vehicle Scheduling Problem [6]

Consider a network. Let N be the set of all nodes in the network, A the set of all arcs and AT the set of route arcs. The decision variable x_{ij} represent the flow on the arc (i,j) in the network and c_{ij} is the operational costs of taking route i directly after route j.

Minimize

$$\sum_{(i,j)\in A} c_{ij}x_{ij} \tag{13}$$

such that

$$\sum_{i:(i,j)\in A} x_{ij} - \sum_{i:(i,j)\in A} x_{ji} = 0 \ , \ \forall n \in N, \tag{14}$$

$$1 \leq x_{ij} \leq 1 \ , \ \forall (i,j) \in AT, \tag{15}$$

$$x_{ij} \geq 0. \tag{16}$$

In Definition 2 equation (14) ensures that the flow, which are vehicles in the VSP case, entering and exiting a node is equal, equation (15) that there is exactly one vehicle assigned to each route and equation (16) that the flow is positive within the network.

The VSP can be extended with additional constraints to consider more scenarios. Some of these include the multi-depot VSP (MDVSP), where there are more than one depot the routes can originate from. Others are the multiple vehicle types VSP (MVTVSP), VSP with time windows (VSPTW) and the vehicle type group VSP (VTGVSP) [6].

1.4.3 NP-hardness of the VRP and VSP

The VRP is an NP-hard problem which indicates that it cannot be solved in polynomial time [2].

The VSP has in contrary time complexity $O(n^3)$ [2], although modified versions of the problem are also NP-hard. For example the MDVSP is NP-hard [3] and even the (MVTVSP), which is an extended version of the VSP even without the multi-depot attribute, is NP-hard [6].

1.4.4 Relation to the WSP

One can think of the WSP in two different ways, where both solutions still convey the same findings.

The first one being a VRP where one treats each pre-optimized route as one single customer instead of a route consisting of several customers. The problem is then treated as a regular VRP with the additions of green thinking, a heterogeneous fleet, capacity constraints, time windows and multiple depots, with the multi-objective of minimizing the fleet size and total cost. It would therefore conclude in the Green multi-depot capacitated heterogeneous fleet VRP with time windows (GMDCHFVRPTW).

On the other hand one can think of the WSP as a scheduling problem and therefore as a VSP. In this formulation it is also necessary to include green thinking, multiple depots, multiple vehicle types (also referred to as heterogeneous fleet in the VRP case), vehicle type groups (in this case equivalent to a capacity constraint in the VRP case) and time windows and. The problem then becomes the Green multi-depot multi-vehicle type vehicle type group VSP with time windows (GMDMVTVTGVSPTW).

These formulations are equivalent, since they portray the same problem (the WSP), but they differ in the way one might experience and approach the problem.

1.5 Solution techniques for VRP and VSP

There are several different solution approaches to the VRP including exact algorithms, heuristics and meta-heuristics [7], the latter being solution methods that combine local improvement and higher level strategies to search the solution space in a robust way and being capable of escaping local optima [8]. Some of the best known exact algorithms can solve for up to 200 customers, but for real life scenarios that has to be solved in reasonable time and with large instances heuristics are required [7].

In the years following 2004 the majority of the evolution of heuristic techniques focused almost exclusively on meta-heuristics, which have been able to solve problems with up to 500 customers. In general these meta-heuristic approaches can be categorized into local search methods, such as simulated annealing, tabu search, iterated local search and variable neighborhood search, and population-based heuristics, such as ant colony optimization, genetic algorithms, scatter search and path relinking. Today, hybrids of these meta-heuristic approaches are becoming increasingly common, making it hard to determine any leading method [7].

For the VSP, and especially regarding the MDVSP, the most common solution methods are neighborhood search, tabu search, iterated local search and integer linear programming [9].

A computational comparison of meta-heuristics is, because of the uniqueness in functionality, in several ways more difficult compared to other algorithmic comparisons. For example, one can focus solely on runtime when comparing two exact solution techniques, but for meta-heuristics the solution quality also need to be considered. Another aspect is that meta-heuristics can be difficult to replicate for other researchers [8].

1.5.1 Branch-and-Cut-and-Price

The evolution of exact algorithms for the VRP has concluded in methods based on the idea of a Branch-and-Cut-and-Price algorithm (BCP). This algorithm combined the earlier Branch-and-Cut algorithms and column generation [7].

1.5.2 Tabu Search

Tabu Search (TS) is a local search method [7] that, in the large amount of research papers published regarding the VRP, include several implementations and are considered to be the top ranked ones [8][5]. The idea of TS is to avoid local optimum by occasionally accepting non-improving moves through the use of memory, namely a tabu list. This can avoid the circling back to previously visited solutions and instead exploring more of the search space [8].

1.5.3 Simulated Annealing

Simulated annealing (SA) is based on the idea of physical annealing of materials where a material is heated to the melting point and then cooled according to a specific temperature scheme. It has been applied to a variety of combinatorial problems, including both scheduling and routing problems and it is widely applied to real-life problems. The simplicity is the main benefit of SA [8].

1.5.4 Genetic Algorithm

Genetic algorithms (GA) are also a common approach to the VRP [5] and have also been shown to be particularly useful for scheduling problems, and more precisely resource scheduling applications [8].

The GA explores adaptiveness and diversity though the idea of natural evolution and the algorithm starts with a population of solutions and moves to the next generation through selected mutation and selection methods. Often, GA have been used as function optimizers even though some have argued that it instead is a system that adaptively finds competitive solutions in an environment where there might not be a clear static optimal solution [8].

1.5.5 Iterated Local Search

Iterated Local Search (ILS) has been a common and successful approach to scheduling problems both on its own and in combination with other meta-heuristics [8].

2 Preliminaries

In this section some notations used in the report will be explained.

Capacity violations are the number of times a vehicle is assigned a route with a demand that overrides the vehicle capacity. A depot violation or time violation occurs when a pair of routes with different depots or overlapping time interval, respectively, are assigned the to the same vehicle. The number of coverage violations indicate how many routes that are not covered by any vehicle and the assignment violations indicate when a vehicle is marked as used, but have not been assigned any route.

Lastly, type penalties indicate how good the vehicle selection is, considering the order of preference of vehicles. A type penalty of zero would indicate that only electrical vehicles have been used, since they do not result in any penalty. A penalty of 1, 2 and 3 are added for every use of a non-electrical vehicle with euro class 6, 5 and 4, respectively.

3 Method

The algorithms chosen to evaluate the WSP together with the method of comparison are presented in the following section.

3.1 Chosen methods

The methods chosen to attempt to solve the WSP are Simulated Annealing, Tabu Search and Genetic Algorithms. They have been chosen both arbitrarily and based on easy implementation, interesting concept and frequency of appearance when researching similar problems to the WSP.

3.2 Method of computation and comparison

To enable conclusion the route set is divided into seven subsets that contain all routes of a specific day of the week. The subset-size ranges from six to 151 routes, with an average of approximately 101 routes and median of 133 routes.

The methods are implemented in Python, where the SA algorithm uses the library "simanneal" [13], GA uses the library "pygad" [12], and TS is constructed without libraries.

The three heuristic methods will all be evaluated following the same structure. For every subset of routes, explained above, the algorithm will run 5 times. The best solution and an average solution will then be retrieved. The parameters of the algorithms were chosen such that the algorithm terminates within 20 minutes for the largest subset. The initial solution or solution population for

SA, TS and GA are all computed via a greedy algorithm (GRA), that prioritizes depot matching and time constraints, starting at a random route and random vehicle each time. The GRA is constructed without libraries.

The whole set will also be evaluated with each method, but the parameters will be altered so that the algorithm terminates within 25 minutes. This will be done twice and then choosing the best solution for comparison.

When comparing the solutions and solution methods all attributes will be compared, namely the number of vehicles, capacity violations, type penalties, depot violations, time violations, coverage violations and assignment violations. The attributes will be prioritized in the following order.

1. Assignment and Coverage violations
2. Depot and Time violations
3. Capacity violations
4. Number of vehicles
5. Type penalties

Furthermore, a comparison of the heuristics and the initial greedy solution will also be made in the same way. In addition a comparison of the original fleet size and the estimated fleet size, resulting from the computed solutions, will be made. This to conclude if the fleet can be reduced or not, and hence also the overcapacity.

4 Results

In this section the result of both the computational work and the way of operating at WoS will be presented. Neither one of GRA, SA, TS or GA produced any depot, time, coverage or assignment violations. Therefore these will not be presented since they do not differ between the methods.

4.1 Original solution at the company

The original numbers of vehicles, based on the solution retrieved from WoS, are presented in Table 2. The actual capacity violations from the original solution is also presented in the same table. From the original data there is a few capacity violations to be expected regardless of vehicle assignment, since the route demand of certain routes override the capacity of every vehicle in the fleet. The original assignment of routes also result in a type penalty for each subset. All of this is also presented in Table 2.

Day	Number vehicles	Actual capacity violations	Expected capacity violations	Type penalties
Day 1	97	4	1	85
Day 2	95	2	2	82
Day 3	98	4	3	86
Day 4	92	2	2	78
Day 5	96	3	1	81
Day 6	7	0	0	3
Day 7	5	1	1	4
Whole week	115	16	10	101

Table 2: Actual number of vehicles used, actual capacity violations, expected capacity violations and actual type penalties from the original data and solution

The results from the original solution in Table 2 show that even if there were only 10 expected capacity violations the original solution has 16 violations. Furthermore, the solution used 115 vehicles with a total of 101 type penalties.

4.2 Initial Greedy Algorithm

In Table 3 the result of the computations using only GRA is presented.

Number of:	Vehicles	Capacity violations	Type penalties
Day 1			
Best Solution	112	4	104
Average Solution	112	4.6	104.2
Day 2			
Best Solution	112	2	105
Average Solution	112.8	2	104
Day 3			
Best Solution	103	3	94
Average Solution	104	3.6	96.8
Day 4			
Best Solution	92	2	80
Average Solution	91.6	3	85.6
Day 5			
Best Solution	95	2	83
Average Solution	95	2.6	87.2
Day 6			
Best Solution	7	0	5
Average Solution	7	0.2	7.4
Day 7			
Best Solution	5	1	4
Average Solution	5	1	4.6
Whole week	128	20	118

Table 3: Result of GRA

The solutions for each subset are fairly consistent considering the number of vehicles, since the average solutions and best solutions vary at most one vehicle. The same is true for the capacity violations, that also have a small deviation on the average from the best solution. In contrast, the type penalties include more variation, up to 5.6 in difference, and therefore do not have the same level of consistency as for the number of vehicles or capacity violations.

For the whole time period, the whole week, the GRA would set the minimum fleet size to 128, including 20 capacity violations and 118 type penalties, to cover the given week of routes.

4.3 Simulated Annealing

In Table 4 the result of the computations using SA is presented.

Number of:	Vehicles	Capacity violations	Type penalties
Day 1			
Best Solution	112	4	106
Average Solution	112.2	5.2	104.6
Day 2			
Best Solution	112	2	101
Average Solution	112.2	2.2	103.8
Day 3			
Best Solution	105	3	97
Average Solution	104.6	3.6	97
Day 4			
Best Solution	91	2	79
Average Solution	91.6	2.8	83.2
Day 5			
Best Solution	95	1	88
Average Solution	95	2.2	88
Day 6			
Best Solution	7	0	3
Average Solution	7	0	4.4
Day 7			
Best Solution	5	1	0
Average Solution	5	1	1.8
Whole week	128	17	118

Table 4: Result of SA

The average number of vehicles does not differ more than one vehicle from the best solution making the deviance small and therefore the method fairly consistent regarding its solutions for each subset. The capacity violations differ at most 1.2, but is also rather consistent. The type penalty variation is larger, at most 4.2, compared to the number of vehicles and capacity violations.

Furthermore, SA determines the minimum fleet to consist of 128 vehicles with 17 capacity violations and 118 type penalties.

4.4 Tabu Search

In Table 5 the result of the computations using TS is presented.

Number of:	Vehicles	Capacity violations	Type penalties
Day 1			
Best Solution	112	3	104
Average Solution	112	5.2	101.4
Day 2			
Best Solution	112	2	97
Average Solution	112	2.4	102.4
Day 3			
Best Solution	104	3	97
Average Solution	104.6	3.8	96.4
Day 4			
Best Solution	92	2	87
Average Solution	92.2	2.8	87.4
Day 5			
Best Solution	96	1	89
Average Solution	95.2	2.4	86.8
Day 6			
Best Solution	7	0	1
Average Solution	7	0	1.8
Day 7			
Best Solution	5	1	0
Average Solution	5	1	0.6
Whole week	128	12	117

Table 5: Result of TS

The TS solutions for each subset have a small variation in vehicle numbers, at most 0.8, and a larger variation in capacity violations, at most 2.2, and type penalties, at most 5.4.

In total for the whole week TS sets the minimum fleet to 128 vehicles, containing 12 capacity violations and 117 vehicle penalties.

4.5 Genetic Algorithm

In Table 6 the result of the computations using GA is presented.

Number of:	Vehicles	Capacity violations	Type penalties
Day 1			
Best Solution	112	2	105
Average Solution	112	3	103.2
Day 2			
Best Solution	112	2	98
Average Solution	112.4	2	99
Day 3			
Best Solution	104	3	91
Average Solution	104.6	3	93
Day 4			
Best Solution	91	2	79
Average Solution	91.8	2	79.6
Day 5			
Best Solution	95	1	82
Average Solution	95	1	86.4
Day 6			
Best Solution	7	0	4
Average Solution	7	0	5
Day 7			
Best Solution	5	1	2
Average Solution	5	1	2.8
Whole week	128	13	118

Table 6: Result of GA

The deviance in the number of vehicles for each subset were small, 0.8, for the GA. In addition the capacity violations were very consistent, with only one subset including a deviation between the average and best solution with one violation. In contrast, the type penalties had a larger variation of at most 4.4.

In total, the GA algorithm sets the minimum fleet size to 128 vehicles with 13 capacity violations and 118 type penalties.

4.6 Method comparison

When comparing the computed solutions and the original solution it is important to note one significant limitation. The original vehicle numbers or assignment of routes does not allow for an accurate comparison with the generated solutions. This is due to the lack of original start-times, which would entail that a computation of time deviations on the original data with the created start-times could be very extreme only because the time intervals have shifted. In the same way the number of vehicles needed can differ greatly if the time intervals are shifted, making more or less routes overlap. Another aspect is that the random assignment did not consider which routes that might be more likely to be executed during the day or evening. This also makes the created solution deviate from reality and further making this comparison not be representative of how good a solution is. Although, since the start-times were created based on their general distribution the solutions generated from the algorithms should not deviate extremely from the actual solution. This would then enable some type of comparison between the methods and the original assignment of vehicles. Therefore, the number of vehicles, capacity violations and type penalties will be evaluated based on the original data but with this limitation in consideration.

When comparing the type penalties it is important to note that these are dependent on the number of vehicles. There is a limited number of electrical vehicles, which result in the need to incorporate other vehicle types. Then, if the solution requires more vehicles than there are electrical ones it is not possible to not have any addition in type penalty. Therefore, a comparison of the type penalty relative to the number of vehicles is more representative of the solution quality.

Starting by evaluating the number of vehicles, first only considering the subsets, all methods (GRA, SA, TS and GA) were consistent in how many vehicles that was used. It differs at most one vehicle in the average solutions compared to the best solution for all subsets, strengthening the conclusion that this is a good approximation of the minimum number of vehicles needed for each day. If the whole set is considered, all methods concluded the same number, 128, of vehicles.

Comparing the vehicle fleet size to the original solution, all solution methods resulted in a larger fleet size than the original solution. As mentioned above, it is difficult to make any conclusions regarding the performance of the method solely based on these results due to the lack of original start times. Therefore, a fleet size of 128, stated by all the methods, could be a good approximate solution for the given set of start times, or it could be so that all methods produced the same not well adapted solution (only regarding fleet size).

Continuing, considering type penalty relative to the fleet size, again only for the subsets, the GA performs better than the other methods. Its solutions include less type penalties relative the fleet size for the larger sets, day 1-5, but is only better than the GRA for the two smaller sets, day 6-7. SA and TS performs slightly better than the GRA on the larger sets, but both optimizes the smaller sets better than both the GRA and the GA, with TS performing the best. Considering the solutions for the whole set all methods perform approximately the same.

Comparing the type penalties for the methods with the original type penalties, the GA performs only slightly worse than the original solution on the subsets, but better than any of the other methods as stated above. Instead, looking at the whole set, all solution methods perform worse than the original solution, but not drastically.

Furthermore, regarding the subsets capacity penalties the GA is the most consistent in both finding the best solution among the methods, but also regarding the consistency in solutions. The average capacity penalty only deviates from the best solution on day 1, otherwise it is the same. Moreover, GRA, TS and SA also find good solutions, when comparing the methods, but they are not consistent since the average deviates more from the best solution than for the

GA. The GRA marginally performed worse than the other methods, which is to be expected since it is only a method of finding an initial solution and not an optimal solution.

Continuing by evaluating the methods compared to the original solution on the subsets, the capacity violations were improved by all methods, including the initial GRA. If, instead, the methods are evaluated by comparing their performance for the whole time period only TS and GA performed better than the original solution in regards to capacity penalties.

Taking all considerations into account the best performing and most consistent method was the GA. This is followed by TS, SA and lastly the GRA.

4.7 The company's way of operating compared to theory

The data received from WoS does not show any organized method or strategy in regards to the way of organizing their vehicle fleet. Therefore it is not possible to determine the similarities or differences compared to established methods. If this comparison is desired, new data must be received to enable conclusion.

5 Conclusion

In conclusion, the GA has the best performance. It was also shown that it is possible to reduce the existing capacity violations in the WoS solution, but no conclusion can be made regarding the fleet size due to the limitations in the dataset. In order to determine if the overcapacity can be reduced the original start times have to be implemented, and preferably more iterations have to be done to provide the best conditions for the chosen method.

In general, the methods were able to adapt well to the real life problem at WoS and show potential to produce a satisfactory solution for the WSP.

Further research should focus on implementing the original start times as well as examining the solution at depot-level. This could then determine if the overcapacity is rooted in one depot location or in several.

6 Bibliography

[1] G. B. Dantzig and J. H. Ramser, "The truck dispatching problem," *Management Science*, vol. 6, no. 1, pp. 80–91, 1959.

[2] J. K. Lenstra and A. H. G. R. Kan, "Complexity of vehicle routing and scheduling problems," *Networks*, vol. 11, no. 2, pp. 221–227, 1981.

[3] A. A. Bertossi, P. Carraresi, and G. Gallo, "On some matching problems arising in vehicle scheduling models," *Networks*, vol. 17, no. 3, pp. 271–281, 1987.

[4] J.-F. Cordeau, G. Laporte, and A. Mercier, "A unified tabu search heuristic for vehicle routing problems with time windows," *Journal of the Operational Research Society*, vol. 52, no. 8, pp. 928–936, 2001.

[5] G. Laporte, "What you should know about the vehicle routing problem," *Naval Research Logistics (NRL)*, vol. 54, no. 8, pp. 811–819, 2007.

[6] S. Bunte and N. Kliewer, "An overview on vehicle scheduling models," *Public transport*, vol. 1, no. 4, pp. 299–317, 2009.

[7] P. Toth and D. Vigo, *Vehicle routing: problems, methods, and applications.* SIAM, 2014.

[8] M. Gendreau, J.-Y. Potvin, *et al.*, *Handbook of metaheuristics.* Springer, 2018, vol. 3.

[9] E. F. Olariu and C. Frăsinaru, "Multiple-depot vehicle scheduling problem heuristics," *Procedia Computer Science*, vol. 176, pp. 241–250, 2020.

[10] A. Arenas-Vasco, J. C. Rivera, and M. G. Baldoquín, "Effect of formulations over a periodic capacitated vehicle routing problem with multiple depots, heterogeneous fleet, and hard time-windows," *PloS one*, vol. 19, no. 10, 2024.

[11] A. K. Garside, R. Ahmad, and M. N. B. Muhtazaruddin, "A recent review of solution approaches for green vehicle routing problem and its variants," *Operations Research Perspectives*, p. 100 303, 2024.

[12] A. F. Gad, *Pygad - python genetic algorithm*, `https://pygad.readthedocs.io/en/latest/`, Accessed: 2025-01-04.

[13] M. Perry *et al.*, *Python module for simulated annealing*, `https://github.com/perrygeo/simanneal`, GitHub repository, Accessed: 2025-01-04.

Datalogi
www.math.su.se

Beräkningsmatematik
Matematiska institutionen
Stockholms universitet
106 91 Stockholm