MASTERARBEIT | MASTER'S THESIS

Titel | Title

Fairness in Multiple Travelling Salesmen Problems

An Analysis and Computational Study

verfasst von | submitted by

Madita Freienberg B.A.

angestrebter akademischer Grad | in partial fulfilment of the requirements for the degree of

Master of Science (MSc)

Wien | Vienna, 2024

Studienkennzahl lt. Studienblatt | Degree programme code as it appears on the student record sheet:	UA 066 915
Studienrichtung lt. Studienblatt | Degree programme as it appears on the student record sheet:	Masterstudium Betriebswirtschaft
Betreut von | Supervisor:	Ass.-Prof. Dr. Christian Tilk

Abstract

The growing demand for trucks to deliver goods to customers has led to challenges in achiving optimal travel times and costs. Therefore, the efficient distribution of several trucks has become an important aspect of modern logistics, which cannot be guaranteed by the traditional TSP model. In contrast, this thesis focusses on the multiple TSP (mTSP), where more than one vehicle is used to meet the customers' demand. This leads to the advantage of being much more realistic and applicable to real-world applications. To solve the mTSP in a time-efficient way, the literature usually uses the MinSum objective function to minimise the total travel time. However, this does not lead to a balanced distribution among sellers, which has also become an important factor, especially in the area of collaborative transport and vehicle workload distribution. Therefore, this thesis analyses not only the MinSum objective but also the MinMax objective function, which minimises the maximum tour length. Moreover, a new constraint is introduced for the MinSum objective function to limit the maximum tour duration and thus achieve a more balanced distribution of routes. The objectives are considered in two different formulations, which are the node-flow and the arc-flow formulations. These models are tested on 48 instances of different sizes and solved the problem exactly by using the branch-and-bound algorithm in CPLEX. The solutions are then evaluated in terms of their general performance and compared in terms of their fairness in terms of route allocation and objective functions.

Zusammenfassung

Die zunehmende Nachfrage nach Lkws für die Auslieferung von Waren an Kunden, führt zu verstärkten Herausforderungen die Fahrzeiten und -kosten so effizient wie möglich zu halten. Daher ist die effiziente Verteilung mehrerer Lkw zu einem wichtigen Aspekt der modernen Logistik geworden, die mit dem traditionellen TSP-Modell nicht gewährleistet werden kann. Aufgrund dessen wird diese Arbeit sich mit dem multiple TSP befassen, bei dem mehr als ein Fahrzeug eingesetzt wird, um die Nachfrage zu befriedigen. Dies hat den Vorteil, dass es weitaus realistischer ist und sich auch auf Anwendungsbereiche im realen Leben anwenden lässt. Um das mTSP zeiteffizient zu lösen, wird in der Literatur oft die MinSum-Zielfunktion zur Minimierung der Gesamtreisezeit verwendet. Dies führt jedoch zu keiner fairen Verteilung zwischen den Fahrzeugen, was ebenfalls ein wichtiger Faktor geworden ist, insbesondere im Bereich des kollaborativen Transports und der Arbeitsauslastung unter den Fahrzeugen. Neben der MinSum-Zielfunktion wird in dieser Arbeit auch die MinMax-Zielfunktion analysiert, die die maximale Tourlänge minimiert. Außerdem wird eine neue Nebenbedingung für die MinSum-Zielfunktion eingeführt, um die maximale Tourdauer zu begrenzen und so eine ausgewogenere Verteilung der Routen zu erreichen. Die Zielfunktionen werden in zwei verschiedenen Formulierungen berücksichtigt, die sich einmal auf den Knoten- und einmal auf den Kantenfluss beziehen. Diese Modelle werden an 48 unterschiedlich großen Instanzen getestet und mit dem Branch-and-Bound Algorithmus in CPLEX exakt gelöst. Die Lösungen werden dann hinsichtlich ihrer Leistung im Allgemeinen sowie ihrer Fairness bei der Routenzuweisung und ihrer Zielfunktionen verglichen.

Table of Content

List of Figures I
List of Tables II
1 Introduction 1
2 Literature Review about Multiple Travelling Salesmen 3
2.1 The Multiple Travelling Salesmen Problem 3
2.2 Variants of Multiple Travelling Salesmen Problem 6
2.3 Solution Approaches for the mTSP 10
2.3.1 Exact Solution 11
2.3.2 Heuristic Solution 12
2.4 Fairness in Optimisation Problems 13
3 Programming Formulations 16
3.1 Implementation and Solution Approach 16
3.2 Models of Investigation 16
3.2.1 Node-Flow Formulation According to Sarin et al. 17
3.2.2 Arc-Flow Formulation According to Maffioli & Sciomachen 18
3.3 Test instances 19
4 Computational Results 20
4.1 Results for MinSum 21
4.2 Results for MinSum with added Constraint 24
4.2.1 Results According to Sarin et al. 25
4.2.2 Results According to Maffioli & Sciomachen 29
4.2.3 Comparative Results of Different Formulations Under New Constraints 33
4.3 Results for MinMax 36
5 Conclusion and Outlook 39
List of Literature III
Appendix VI

List of Figures

Figure 1: TSP with a single route. 4

Figure 2: mTSP with three routes.. 4

Figure 3: Overview of mTSP variations according to Cheikhrouhou & Khoufi (2021). 7

Figure 4: Overview of mTSP solution approaches... 10

List of Tables

Table 1: MinSum Solution solving Instance att532_5_21. 14

Table 2: Overview and comparison of the benchmark problems. 19

Table 3: Test instance combinations. 20

Table 4: Performance results of Model 1 and Model 3. 22

Table 5: Fairness results of Model 1 and Model 3. 23

Table 6: Route allocation of Model 1 and Model 3 using instance att532_3_21. 23

Table 7: Performance results of Model 1 with new constraint. 25

Table 8: Fairness results of Model 1 with new constraint. 27

Table 9: Route allocation of Model 1 with 5% and 15% constraint using instance att532_3_21. 28

Table 10: Performance results of Model 3 with new constraint. 29

Table 11: Fairness results of Model 3 with new constraint. 31

Table 12: Route allocation of Model 3 with 5% and 15% constraint using instance att532_3_21. 32

Table 13: Comparison of performance results under new constraints. 33

Table 14: Comparison of fairness results under new constraints. 35

Table 15: Performance results of Model 2 and Model 4. 36

Table 16: Fairness results of Model 2 and Model 4. 37

Table 17: Route allocation Model 2 and Model 4 using instance att532_3_21. 38

1 Introduction

The efficient allocation of trucks is an important issue for many companies in practice in order to increase time and cost efficiency by delivering goods to their customers (Wang et al., 2020). The classical Traveling Salesman Problem (TSP) involves a single vehicle visiting all customers. In contrast, the Multiple Travelling Salesmen Problem (mTSP) is gaining more attention from researchers because it involves multiple trucks, which increases the complexity of allocation efforts (Bektas, 2006). This makes mTSP a highly realistic problem, applicable in various fields such as transport and delivery, work or machine scheduling, disaster management, and agriculture (Cheikhrouhou & Khoufi, 2021). These sectors are highly time-critical, prompting companies to seek for the fastest possible delivery of goods (Wang et al., 2020). In the e-commerce sector, where customer expectation and the competition is particularly high, different types of delivery vehicles like drones, trucks, robots, and other unmanned vehicles are used (Kitjacharoenchai et al., 2019). These are considered to achieve a specific goal: to minimise the total travel time or costs, known as the MinSum objective, which is extensively used in the literature to solve the mTSP (Jiang et al., 2020; Kivelevitch et al., 2013).

However, another important factor in contemporary research is the fair distribution of the workload among the vehicles. It should be questioned what the efficient distribution of tour duration looks like and how the allocation of individual vehicles to customers should be organised. These questions were already discussed by Okonjo-Adigwe in 1988. To address fairness, various methods are introduced in the past years. One of the most studied methods in this field involves the MinMax objective function, which aims to balance the labour among salesmen by minimising the maximum tour duration (Matl et al., 2018).

The principal contributions of this work will therefore be to analyse the MinSum and MinMax objective functions in both an arc-flow and a node-flow formulation. Moreover, a new constraint is introduced for the MinSum objective function, which limits the maximum tour duration in the MinSum model. The objective of the analysis is the examination of two key aspects:

1. An evaluation of the overall efficiency of the various models, and
2. An investigation of the impact of changes to the objective function and models on the overall fairness in tour length and tour duration.

The information above is crucial as it provides an understanding of the relevance and motivation of the topic, as well as the key research goals to be addressed. The following section highlights the approach of the thesis and provides an overview of the individual steps and the structure of the thesis. The following second chapter creates a foundation of the topic by conducting a literature review from past and present academic resources, in order to create a better understanding for the concept of the mTSP. This includes defining the mTSP, with a closer look at two variations of problem formulations according to Sarin et al. (2014) and Maffioli and Sciomachen (1997), with their objective functions (MinSum and MinMax), the constraints as well as the decision variables.

Furthermore, it is necessary to know the importance in an historical context and significance in the TSP study area. The results and different opinions of the literature and research are contrasted, compared, and discussed. Additionally, Chapter 2 contains common variations and solution methods of the mTSP to showcase the complexity and diversity of the problem. Especially the branch-and-bound algorithm is defined, which is used in the further course of the thesis. Moreover, the topic of fairness is introduced, which is the basis of this work to evaluate the mTSP. Which discusses in particular, existing findings regarding fairness.

Chapter 3 then focuses on the implementation and the computational study of the problem. First, the relevant details to programme the mTSP are described, with a focus software C++, the hardware as well as maximum computational time. The different formulations of the mTSP are solved exactly, by applying the solver CPLEX which uses the branch-and-bound algorithm. The next section implements the two objectives (MinMax and MinSum), where each objective is further subdivided into two formulations, resulting in four models, two according to the formulation of Sarin et al (2014) and two according to the formulation of Maffioli and Sciomachen (1997). The study examines which objective functions are relevant and which restrictions and decision variables need to be added. The final step gives a detailed explanation of the data used in the study, including its source and selection of data with the specific characteristics, which are relevant to the problem.

The Chapter 4 compares and analyses then the results of the computational study. On the one hand, the overall performance of the models is examined in terms of solution time, optimality and the objective functions. On the other hand, the results of the objective functions are examined in terms of fairness. The aim is to analyse and compare how fair or equal the individual results are. In addition, a new constraint is added to the MinSum formulation to limit the maximum tour duration in order to improve the fairness of the MinSum objective function.

Finally, Chapter 5 summarises the results and observations of the previous analysis and highlights difficulties and further research fields.

2 Literature Review about Multiple Travelling Salesmen

As mentioned in the introduction, the first part of this paper helps the reader to gain a better understanding of the mTSP and fairness in optimisation problems. The following Sections 2.1 to 2.3 give a deeper insight into the topic, by providing a general literature overview of the definition, relevant variants and solution approaches of the mTSP. Section 2.4 contains a general definition of fairness in optimisation problems, as well as research from past and current literature particularly about fairness in the mTSP.

2.1 The Multiple Travelling Salesmen Problem

The mTSP is a generalisation of the classical TSP, whereby the TSP is extended so that multiple salesmen have to visit all customers (Al-Omeer & Ahmed, 2019; Bektas, 2006). Consequently, the TSP represents a fundamental element in the field of combinatorial optimisation and operations research. The TSP, which is the most investigated problem in this area, searches the shortest route that visits a set of cities and returns to the city from the start (Burkard et al., 1998). As industries and applications evolves, the need for more complex models, like the mTSP become apparent, reflecting real-world scenarios where multiple salesmen or vehicles operate simultaneously (Al-Omeer & Ahmed, 2019). In 2006, Bektas listed a large number of early applications, such as print press scheduling, workforce planning, transportation planning, mission planning, production planning, and settling satellite systems. This list continually expanded in the past years by adding allocation of computer resources, network resources to consumer, parts and processing machines in a factory and targets to unmanned vehicles (Kivelevitch et al., 2013). Further applications are school buses and mail delivery (Bolaños et al., 2015), as well as disaster management and agriculture (Cheikhrouhou & Khoufi, 2021). Additionally, the mTSP is a relaxation of the classical Vehicle Routing Problem (VRP), which also includes multiple vehicles but focusses especially on the vehicle capacity and the customers demand (Cheikhrouhou & Khoufi, 2021). However, the difference to classical TSP and VRP in academic study is that the mTSP research is much less extensive and does not require the same effort. But more and more researchers work on the mTSP, so that new efficient methods of solving this problem are constantly developed (Jiang et al., 2020). In particular, Cornejo-Acosta et al. (2023) investigated through an extensive literature review that the interest specifically for the single depot mTSP has increased in form of published papers from 1970 until 2023. Most authors define mTSP as an optimisation problem which takes into account several drivers to visit all customers (Cheikhrouhou & Khoufi, 2021). In contrast, the TSP considers only one salesman. This distinction becomes particularly clear in Figure 1 which represents the TSP, in which a single driver serves all 13 customers on a single route before returning to the depot. Figure 2, on the other hand, illustrates a scenario in which three drivers are responsible for distributing 13 customers across three distinct routes, this results in a division of labour.

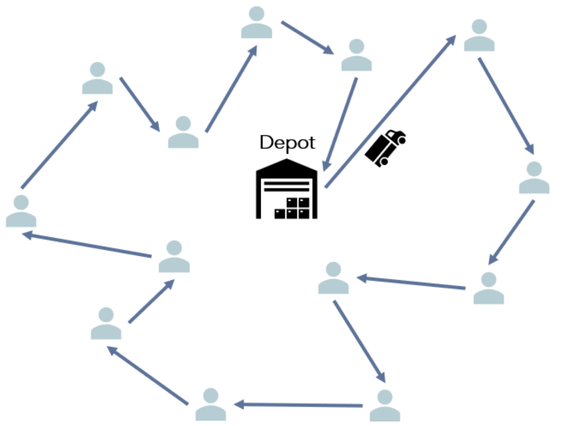

Figure 1: TSP with a single route.

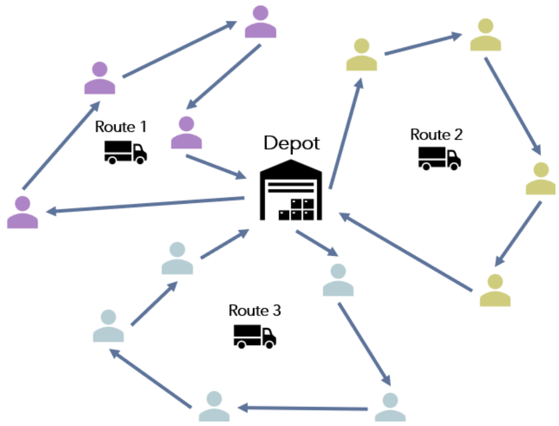

Figure 2: mTSP with three routes.

After a brief distinction is made between TSP and mTSP, the problem is now described in a more mathematical way. The number of salesmen is defined as m >= 2, who drive to a set of V = {1, …, n} customers one by one. Each route begins and ends at a single depot, although the start is labelled as “o”, as origin and the destination as “d” (Al-Omeer & Ahmed, 2019). The problem is considered as a complete graph *G* = (*V*, *A*), where a pair of different customers are connected through an arc (i, j) ∈ *A*. The primary objective in the classical mTSP is to minimise the total cost or distance of all routes, which is also called as MinSum objective (Jiang et al., 2020; Kivelevitch et al., 2013).

The following formulation according to Sarin et al. (2014), is mostly used in the literature and defines the extended Miller-Tucker-Zemlin (MTZ) formulation for the single asymmetrical TSP, which is explained in more detail later in this thesis. Regarding the formulation according to Sarin et al. (2014), following constraints need to be satisfied to get a feasible solution (Sarin et al., 2014). In order to facilitate understanding of the formulations, the notations are presented below:

- ***n*:** the number of customer nodes.
- ***V* = 1, …, n** is the set of customers in the graph.
- ***A*:** number of all edges from i to j in the graph.
- ***m*:** the number of salesmen.
- **t_{ij}:** travel time from node i to node j.
- **x_{ij}:** decision variable, which is either 1 if an arc is used or 0, if the arc is not used.
- **o:** Start vertex.
- **d:** End vertex.
- **M:** A large constant, called Big-M.

First of all, the objective function (1a) is explained, which is a MinSum function. The objective in this formulation aims to minimise the sum of travel time t_{ij} from 'i' to 'j', so of each arc that is used. Which is defined as x_{ij}, which is explained in more detail in constraint (1h).

$$z_{ATSP} = \min \sum_{(i,j)\in A} t_{ij}\, x_{ij} \tag{1a}$$

Constraints (1b) and (1c) represent the inflow and outflow restrictions, often referred to as the flow conservation of routes. These constraints ensure that exactly the same number of *m* vehicles must depart from origin depot (from o to j) and return to (from i to d) the destination depot.

$$\sum_{(o,j)\in\, \delta^{+}(o)} x_{oj} = m \tag{1b}$$

$$\sum_{(i,d)\in\, \delta^{-}(d)} x_{id} = m \tag{1c}$$

Whereas constraints (1d) and (1e) are classic allocation restrictions. These limitations ensure that each customer in the set *V* = 1, …, *n* has exactly one predecessor and exactly one successor. In other words, these constraints guarantee that exactly one tour enters each node, and one tour exits each node.

$$\sum_{(i,j)\in\, \delta^{+}(i)} x_{ij} = 1 \quad for\ all\ i\ \in V \tag{1d}$$

$$\sum_{(i,j)\in\, \delta^{-}(j)} x_{ij} = 1 \quad for\ all\ j\ \in V \tag{1e}$$

The next constraints (1f) and (1g) represent the classic MTZ subtour elimination constraints. In this particular context, the constraints prevent smaller subtours by enforcing a strict order and direct dependencies between the nodes. This helps to maintain the integrity of the entire route and ensure that all nodes are visited in a continuous, single tour without creating smaller, isolated subtours. In constraint (1g) the variable T_i represents a rank order index that is linked to the customer i. When i is one, which is the depot, the value of T_i is zero. The Big-M represents a very large number, which ensures that the constraint is active, when the arc between i and j is used.

$$T_i + t_{ij} - T_j - M\left(1 - x_{ij}\right) \le 0 \quad for\ all\ (i,j) \in A, i, j \neq o \tag{1f}$$

$$T_i \geq\ t_{oi} \quad for\ all\ i\ \in V \tag{1g}$$

The last constraint in this formulation is (1h), which represents the restriction on the decision variable, which is either 1, when the arc is used or 0, when the arc is not used.

$$x_{ij} \in \{0,1\}\ \ for\ all\ (i,j) \in A \tag{1h}$$

Although the formulation of Sarin et al. (2014) is used more frequently in the literature, it is nevertheless important to consider also other formulations in order to create a different perspective on the problem. Another approach to formulate the problem is the arc-flow formulation according to

Maffioli and Sciomachen (1997) which is introduced by Bianchessi et al. (2023). The formulation is less used in the literature, because more variables are needed, which makes it more complex compared to the node-flow formulation from Sarin et al. (2014). Further, the arc-flow formulation includes different time variables, defined as T_{ij} and T_{ji}, which lay on the arcs and not on the nodes as Sarin et al. (2014) is done.

$$z_{ATSP} = \min T_d \tag{2a}$$

$$T_{oj} = t_{oj} x_{oj} \quad for\ all\ j \in V \tag{2f}$$

$$\sum_{(i,j)\in \delta^+(i)} T_{ij} - \sum_{(j,i)\in \delta^-(i)} T_{ji} = \sum_{(i,j)\in \delta^+(i)} t_{ij} x_{ij} \quad for\ all\ i \in V \tag{2g}$$

$$T_{ij} \leq T_d \quad for\ all\ i \in V \tag{2h}$$

$$t_{ij}^0 x_{ij} \leq T_{ij} \leq M * x_{ij} \tag{2i}$$

The objective function (2a) is different compared to the model before. Whereas the flow conservation constraints (1b) to (1c) and assignment restrictions (1d) and (1e) are identical to the constraints in the Sarin et al. (2014) formulation. The next constraints (2f) and (2g) are the extended MTZ subtour elimination constraints, which include the time variables on the arcs. The next constraint (2h) ensures that the time spent at the depot is the maximum possible. Additionally, a further constraint (2i) is introduced, which represents the dependency of the time variable T_{ij} on x_{ij}. This implies that T_{ij} can only become positive if x_{ij} is used.

2.2 Variants of Multiple Travelling Salesmen Problem

With the initiation of new applications for mTSP, the number of variants has increased in the last years. To categorise these variants, Cheikhrouhou & Khoufi (2021) have defined classifications, such as salesmen characteristics, depot specifications, city specifications, objective function and problem constraints. The following variants which are shown in Figure 3 are organised based on these characteristics.

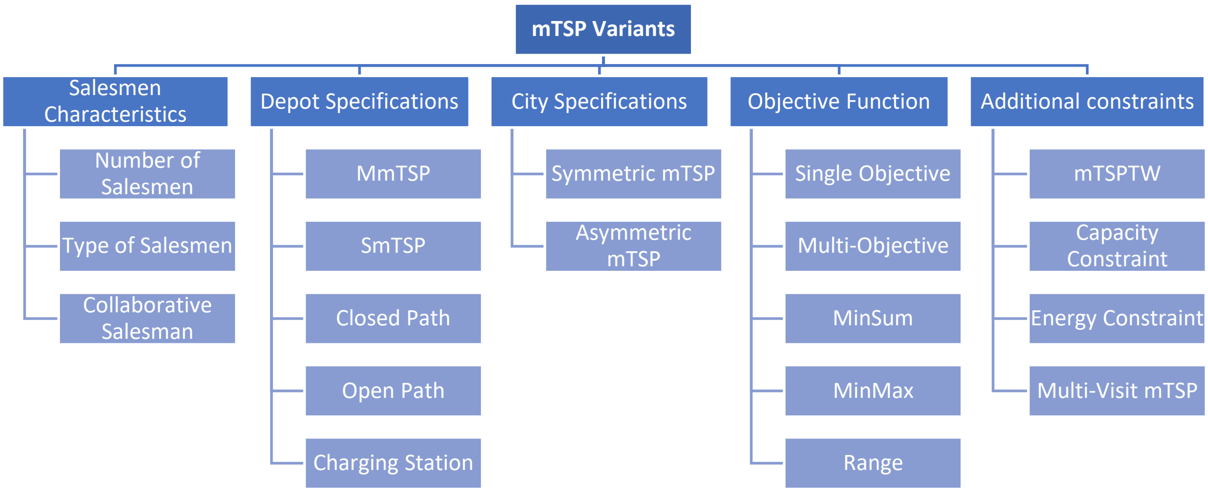

Figure 3: Overview of mTSP variations according to Cheikhrouhou & Khoufi (2021).

Variants with different salesman characteristics

Firstly, the category with the various salesman characteristics is examined. According to Bektas (2006) and Cheikhrouhou and Khoufi (2021), this includes the number of salespeople, which can either be limited to a fixed value or prioritised. That means on the one hand, if the problem specifies a fixed number of salesmen, it can be assumed that the number of vehicles included in the problem is predetermined and unchangeable. One the other hand, if the number of salesmen in the mTSP are variable, this means that the optimal number of salesmen is determined as part of the solution to the problem. This is particularly useful in scenarios where, for example, flexibility in resource allocation is desired (Bektas, 2006). Another distinguishing feature is the type of salesperson, which can vary. The type of salesman may be a person, a vehicle, a robot, or even a drone (Cheikhrouhou & Khoufi, 2021). A significant amount of state-of-the-art research in the field of mission planning indicates that the allocation of unmanned aerial vehicles (UAVs), more commonly known as drones, mobile robots and autonomous vehicles, appears as a primary area of investigation, (Alighanbari et al., 2003; Cheikhrouhou & Khoufi, 2021; Chen et al., 2017). This is largely driven by the increase in time-critical deliveries of goods in the context of e-commerce (Kitjacharoenchai et al., 2019). Another variant of the category is salesmen who cooperate or collaborate with other areas. In particular, different salesman types are used in collaborative roles to complete or enhance the efficiency of a route (Cheikhrouhou & Khoufi, 2021). A commonly cited example of this collaboration is the use of classic delivery vehicles and drones (Cheikhrouhou & Khoufi, 2021; Kitjacharoenchai et al., 2019; Murray & Raj, 2020). In this example, the trucks drive to a specific location with geographically close customers, after which the drones are released. Then, these drones carry out the last mile to the customer. This method is already tested in reality by e-commerce companies such as Google, DHL and Amazon. The aim of this testing is to reduce the time of the last mile delivery and the arrival time at the depot, which could also result in cost savings (Kitjacharoenchai et al., 2019; Murray & Raj, 2020).

Variants with different depot specifications

The next set of variants relates to the specifications of depots. A distinction is made between single or multiple depots, fixed or mobile depots, closed or open paths and charging points (Cheikhrouhou & Khoufi, 2021). The distinction between single and multiple depots can be identified through their start and finish locations. These two variants, which are also called SmTSP and MmTSP are the most investigated over the last decades (Cornejo-Acosta et al., 2023). In a single depot, which is used in the classical mTSP, all vehicles depart and finish their tours at the same point (Bektas, 2006; Kivelevitch et al, 2013), which is referred to as a closed path (Liu et al., 2013). This variant represents a centralised management approach, which is commonly utilised by delivery services for express shipment. This approach enables the delivery of goods in a time- and cost-efficient manner (Chen et al., 2017). In contrast, the mTSP with multiple depots starts and ends at different locations (Oberlin et al., 2009), which is also referred to as an open path (Liu et al., 2013). The driver is permitted to either return to the original location, which is called fixed depot or utilise another nearby depot as a non-fixed or mobile destination (Bektas, 2006). This variant is a decentralised system, which is employed in the context of complex real-life scenarios, such as hot rolling scheduling, which involves the planning and organisation of production processes in a rolling mill (Chen et al., 2017). Finally, there are the refuelling or charging stations that can be incorporated into the design if an mTSP with a petrol or electric tank option is to be considered. Refuelling or charging can be conducted at a depot or at additional locations (Cheikhrouhou & Khoufi, 2021). This group of variants also includes the mTSP with drone stations (mTSP-DS), which is used as soon as the drone battery runs low (Kloster et al., 2023).

Variants with different city specifications

The third section refers to city specifications. According to Oberlin et al. (2009) and Sarin et al. (2014), the variants can be differentiated as either symmetrical or asymmetrical mTSP. The difference between symmetric and asymmetric mTSP lies in the structure of the travel costs or distances between the cities or destinations. In the symmetric mTSP, the travel costs or distances between two cities are identical, regardless of the direction of travel. This implies that the cost of travelling from city A to city B is equivalent to the cost of travelling from city B to city A. In the asymmetric mTSP, however, the travelling costs or distances between two cities are different (Sarin et al., 2014). Cheikhrouhou and Khoufi (2021) state that a further distinction can be made in the city specifications, namely in the subdivision of the cities under the salesmen. The salesmen can either share all cities with each other or there are two groups of cities, where one group is shared by all salesmen and the second group of cities must be visited by a specific salesman.

Variants with different objective functions

The fourth group consists of variants in which different objective functions are enhanced. The existing literature distinguishes between single objective and multi-objective variants. For a single objective, the focus is on achieving one specific aim, such as the classical MinSum discussed in Chapter 2.1. Other objectives can include the reduction of the energy consumption, or mission time (Cheikhrouhou & Khoufi, 2021; Mahmoudinazlou & Kwon, 2023). The MinSum mTSP is subjected to a comprehensive review in the literature by Bektas (2006), Sarin (2014) and Svestka & Huckfeldt (1973), among others. Thus, it represents the most commonly used objective function (Soylu, 2015). Another single objective mentioned in Section 2.1 is the MinMax objective. The MinMax for mTSP, as described by França et al. (1995), is designed to minimise the maximum tour duration, and is therefore used in more realistic problems in order to balance the workload between salesmen (Liu et al., 2013; Mahmoudinazlou & Kwon, 2023; Somhom et al., 1999; Soylu, 2015). Similarly, another objective is known as the range objective. This objective is applied, for instance, when creating work schedules for machines where the goal is to minimise the time span between the minimum and maximum duration of machine usage (Soylu, 2015). In contrast, a multi-objective approach involves two or more objectives which need to be fulfilled simultaneously (Cheikhrouhou & Khoufi, 2021). Bolanos (2015) for example considers a multi-objective mTSP which is minimises the total costs on the one hand and improves the service level on the other hand. Another example is provided by Necula et al. (2015), who combine range and MinSum objectives which is similar to the objective investigated by Bolanos, Chen et al. (2018) and Trigui et al. (2017). This aims to optimise the overall costs and maximise the overall performance of robots.

Variants with additional constraints

The final group of mTSP variants includes additional problem constraints that can be applied to a problem formulation. One variant involves time constraints (mTSPTW), which means that customers may only be visited within a specified time window. This variant is frequently mentioned in the literature by several authors, such as Kivelevitch et al. (2013), Wang et al. (2020) or Al-Omeer and Ahmed (2019). An alternative example involves the allocation of the sequence of tasks for a fleet of vehicles, robots or unmanned vehicles (Chen et al., 2017). These tasks are typically subject to time constraints that must be satisfied (Alighanbari et al., 2003). Another additional restriction is the capacity constraint. This relates in particular to the capacity of packages, weight or volume of the vehicles. Accordingly, a vehicle may not transport more than the permitted limit, and this is taken into account when planning the route (Kivelevitch et al., 2013; Wang et al., 2020; Cheikhrouhou & Khoufi, 2021). The energy or tank constraint is similar to the capacity constraint. Each vehicle is restricted by a maximum amount of energy, which limits the driving range. As the energy is consumed during the tour, recharging occurs during the mission (Kivelevitch et al., 2013; Cheikhrouhou & Khoufi, 2021). Other constraints may include that customers allow multiple visits, e.g. for waste collection or food deliveries (Bérczi et al., 2023).

In summary, the mTSP's relevance grew with its application to various real-world problems such as vehicle routing, logistics, and manufacturing. Researchers extended the basic mTSP model to integrate practical constraints like time windows, capacity limits, and multi-depot scenarios. Notable contributions include the work of Laporte & Nobert (1987), who explored mTSP in the context of vehicle routing problems with time windows (VRPTW), providing comprehensive models and solutions, which are discussed in the next Chapter.

2.3 Solution Approaches for the mTSP

As previously stated, the topic of mTSP is thoroughly investigated in academic research over the years. Therefore, numerous approaches solving a diverse range of mTSPs are investigated. In this context, it is essential to distinguish between heuristic and exact solution methods, which are shown in Figure 4, as these approaches vary significantly in their processes and result qualities.

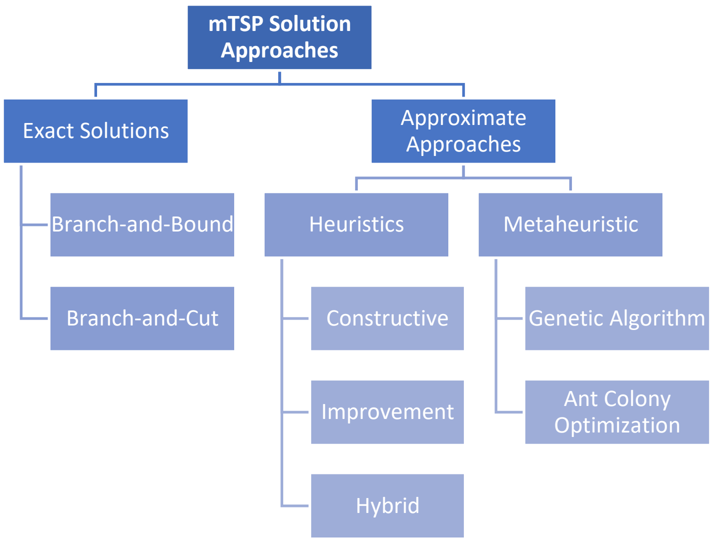

Figure 4: Overview of mTSP solution approaches

Exact methods identify the optimal solution, while heuristics produce solutions that are feasible and generally effective but not necessarily optimal. Due to the complex and time-intensive nature of finding optimal solutions, exact methods are more appropriate for smaller-scale problems (Jiang et al., 2020). In contrast, heuristics are preferred for larger and more complex problems, which provide near-optimal or feasible solutions more efficiently (Carter & Ragsdale, 2006; Jiang et al., 2020). This makes heuristics especially valuable for solving NP-hard problems like the mTSP (Chen et al., 2017; Lu & Yue, 2019), where the computational effort increases drastically for large problems. The most commonly used techniques to solve the mTSP are listed in Figure 4 and are outlined and explained in the following paragraphs.

2.3.1 Exact Solution

There are many different deterministic or exact algorithms which are proposed in the last years to solve the mTSP. Examples for that are the straight algorithm, reverse algorithm, branch-and-bound, and quasi assignment relaxation relaxing the sub-tour elimination constraints (SECs) and a lot more (Chen et al., 2017; Jiang et al., 2020). This paper is limited to the two approaches branch-and-bound (BnB) and branch-and-cut (BnC). Whereby the branch-and-cut algorithm is not relevant for the further course of this thesis.

Among these, the BnB algorithm is the most cited in the literature which is considered generally to solve the MinSum objective (Bektas, 2006; Chen et al., 2017; Cornejo-Acosta et al., 2023; He & Hao, 2023; Jiang et al., 2020). During the 1970s and 1980s, significant advances are made in developing exact algorithms for solving mTSP (Cornejo-Acosta et al., 2023). One of the earliest contributions are from Svestka and Huckfeldt (1973), who introduce an exact algorithm and formulation for the MinSum mTSP to find an optimal solution. In order to handle the added complexity of multiple salesmen, they use a BnB algorithm. The BnB algorithm is divided into two steps. The first step, branching, involves partitioning into smaller subproblems to simplify the overall problem. Each subproblem is then solved to the optimality and added as a new branch of the search tree. If the solution is infeasible or suboptimal the second step, bounding, is used to eliminate those solutions (Svestka & Huckfeldt, 1973; Cornejo-Acosta et al., 2023; Bektas, 2006). In 1986, Gavish and Srikanth tried to use the BnB algorithm for large symmetric problems, but this is generally not recommended because the computational effort is too high (Chen et al., 2017; Jiang et al., 2020). Whereas Ali and Kennington introduced in 1986 the BnB for asymmetric TSPs using a so-called Lagrangian relaxation that relaxes more complex constraints (Bektas, 2006). To get the most optimal results, it is proposed to include maximum 500 vertices and 10 salesmen to the problem (He & Hao, 2023).

Researchers like Laporte and Nobert (1980) enhanced the BnB technique by integrating cutting plane methods (Bolaños et al., 2015). The BnC algorithm is therefore an extension of the before mentioned BnB algorithm, which combines the efficiency of searching by using the BnB and the tightening of the solution area by using the cutting plane approach (Kazachkov et al., 2023). As with the BnB, the problem is divided into two subproblems, and both methods use bounds to decide which subproblem needs to be considered further and which needs to be eliminated. However, during the branch step, the cutting plane method is included to efficiently narrow down the range of solution values, thereby help with eliminating non-feasible solutions more rapidly (Bolaños et al., 2015; Kazachkov et al., 2023). Hence, BnC is relatively quick compared to BnB in solving instances with a maximum of 120 customers and four salesmen (Cheikhrouhou et al., 2021; He & Hao, 2023). Furthermore, BnC is more frequently used in mixed-integer linear programs, which are more complex, and thus often used for MinMax VRPs (He & Hao, 2023).

The problem becomes larger with the addition of multiple sellers to meet demand and even more complex for exact methods due to its NP-hardness (Chen et al., 2017). According to Liu et al. (2013), several researchers investigated that exact algorithms always deliver poorer results in comparison to heuristics, when solving mTSPs. This limitation led to a growing interest in heuristic and metaheuristic approaches in the 1990s. Consequently, it is also important to mention the most used ones.

2.3.2 Heuristic Solution

A heuristic is used, when the problem gets too big to solve optimal with an exact approach. Therefore, a significant number of authors utilise different heuristic and metaheuristic approaches (Cornejo-Acosta et al., 2023). Before outlining the most mentioned heuristics and metaheuristics in detail, it is necessary to explain the terms in more detail. These two terms “heuristic” and “metaheuristic” are often used interchangeably in academic literature. But the terms differentiate in some point. A heuristic for example is designed for a specific problem and is generally quicker in generating solutions (Chen et al., 2017). Heuristics are typically classified into three categories: constructive heuristics, which include nearest and farthest neighbour (Sze & Tiong, 2007), improvement heuristics, which contain local search, Lin-Kernighan (Vali & Salimifard, 2017) and the classical 2-opt exchange (Somhom et al., 1997), and hybrid heuristics, which is considering a combination of different approaches. Whereas metaheuristics do not necessarily refer to a specific problem but represent a generalised technique. The objective of metaheuristic algorithms is to identify robust and efficient solutions by escaping the local optima and exploring one or more other neighbourhoods for a better optimum (Gendreau & Potvin, 2010). In the following Section, the metaheuristics Genetic Algorithm (GA), and Ant Colony Optimisation (ACO) will be discussed in detail.

First, the GA will be discussed, which is adjusted in the 1990s for the mTSP (Cornejo-Acosta et al., 2023). A comprehensive review of the literature highlights that GA is the most used heuristic in both historical and contemporary studies (Carter & Ragsdale, 2006; Mahmoudinazlou & Kwon, 2023). This approach is particularly popular for UAVs such as robots (Cheikhrouhou et al., 2021) and consistently achieves satisfactory results within a reasonable timeframe and is relatively straightforward to implement (Al-Omeer & Ahmed, 2019). Moreover, the GA is an evolutionary method inspired by natural selection, which is starting with a randomly generated chromosome. Through iterative processes based on the fitness of the individual chromosomes, the algorithm develops this population towards better solutions until a termination condition is met (Carter & Ragsdale, 2006; Jiang et al., 2020). Further, modifications to enhance its application to the mTSP are proposed by Zhou et al. (2018), who introduced two partheno GAs which include new mutation operations. Moreover, recent academic papers have adopted this solution approach, and often combining it with other methods to achieve better results. Kitjacharoenchai et al. (2019), on the one hand, proposed a combination of the GA, K-means and Nearest Neighbour, and random cluster/tour approaches to optimise mTSP solutions involving drones. Mahmoudinazlou and Kwon (2023), on the other hand, proposed a hybrid genetic algorithm that utilises a divide-and-conquer strategy to improve solution efficiency.

Another commonly used technique, which especially considers trucks, is known as ACO (Cheikhrouhou et al., 2021). This model is initially developed by Doringo in 1992 and simulates the communication of ants searching for the shortest paths to food sources (Lu & Yue, 2019). By using artificial ants that leave virtual pheromones on paths between decision points, ACO algorithms guide subsequent ants to shorter routes based on the intensity of pheromone trails, iteratively refining the solution until optimal or near-optimal paths are discovered (Lu & Yue, 2019). In 2006, Junjie and Dingwei presented the effectiveness of ACO by addressing complex routing challenges, such as the mTSP (Mahmoudinazlou et al., 2023). Furthermore, Jiang et al. (2020) have described an innovative hybrid approach combining the partheno GA with ACO to tackle large-scale mTSPs.

In recent years, the focus is shifted towards integrating machine learning techniques with traditional optimisation methods to enhance solution accuracy and speed. The initiation of powerful computational tools and parallel processing also enables the solving of larger and more complex mTSP instances. Researchers increasingly explore stochastic and dynamic variants of mTSP, reflecting real-time decision-making environments.

2.4 Fairness in Optimisation Problems

This chapter introduces the second important term of the thesis, the fairness aspect in the mTSP. However, before discussing fairness in mTSP in more detail, it is essential to first explain the importance to consider fairness in optimisation problems and why the MinSum objective function presents a major issue in including fairness.

The MinSum objective, which is introduced by França et al. 1995, is the most used function in literature as mentioned before. Because it focusses on the minimisation of the total travelled time of all vehicles it is always looking for the most optimal and efficient way in terms of the overall time (Lu & Yue, 2019; Chen & Hooker, 2023). However, several authors investigate that the MinSum objective leads to an imbalance in the workload among salesmen (Bhadoriya et al., 2024; Lu & Yue, 2019). This is also illustrated in the example below. Instance att532_5_21, which contains 21 customers and five vehicles, is solved with a MinSum objective function. As shown in Table 1, vehicles one to four are responsible to serving one single customer each, while all other remaining customers are served via route five. This demonstrates the unbalanced distribution of the work, with vehicle five doing most of the work.

Table 1: MinSum Solution solving Instance att532_5_21.

Route	Route distribution using MinSum o.f.
1	0 – 1 – 21
2	0 – 4 – 21
3	0 – 5 – 21
4	0 – 6 – 21
5	0 – 7 – 8 – 10 – 11 – 9 – 12 – 13 – 17 – 19 – 16 – 14 – 15 – 18 – 20 – 2 – 3 – 21

There are many real and practical scenarios in which the fair allocation of resources must be ensured so that ethical concerns are included, as in disaster management, care or telecommunications (Chen & Hooker, 2023). A comprehensive definition of the term fairness is presented by the Cambridge Dictionary (n.d.). According to them, fairness is defined as "the quality of treating people equally or in a way that is right or reasonable" (Cambridge Dictionary, n.d.). In contrast to the general definition, Chen and Hooker (2023) claim that equality is not necessarily the same as fairness in optimisation problems. There are different types of fairness that need to be considered, for example the equally distributed resources throughout all vehicles or the fairness in profit. According to Bertsimas et al. (2011) and Bhadoriya et al. (2024), the fairness aspect of allocation problems such as the mTSP, VRP or collaborative transports are widely studied in many practical fields and theoretical studies, especially in social sciences, welfare economics and engineering. In this context Soriano et al. (2023) define the term workload, which is the total number of visited customers, the total time or distance duration or the total load delivered. Therefore, workload balancing may not directly result in monetary benefits such as increased profits or cost reductions. However, it significantly increases salesmen and customer satisfaction and provides greater flexibility in resource availability (Matl et al., 2018; Soriano et al., 2023).

Okonjo-Adigwe investigates in 1988 how the workload under salesmen can be balanced through an effective method, therefore this is not a new topic in the field of mTSP. Over time, as discussed in Chapter 2.2, the variations of the mTSP have increased and changed. Consequently, various methodologies for incorporating fairness considerations are proposed in numerous academic studies. Examples for that are fairness criteria, additional constraints, multi-objectives or adapted objectives. Starting with Chen and Hooker (2023), who introduce a guide to formulate fairness in an optimisation model by proposing ethic-related metrics like maximin, which maximises the minimum utility, leximax which maximises the welfare of the worst or the McLoone index, which compares the total utility. Furthermore, numerous authors (Bhadoriya et al., 2024; Sánchez, 2022; Xinying Chen & Hooker, 2023) including Matl et al. (2018), mention the Gini coefficient, which represents the statistical dispersion of the distribution of work. Further fairness metrics that Matl et al. (2018) discuss are rank function or standard deviation. Other authors such as Soriano et al. (2023) and Bhadoriya et al. (2024) include novel extra constraint to their formulations to consider workload balance such as ε- fair mTSP, that includes second-order cone constraint or the Δ-fair mTSP include a constraint that limits the workload among salesmen. Alternatively, multi-objective functions may be used to distribute shared profits or costs in a manner that is perceived to be fair and efficient at the same time (Soriano et al., 2023). It is also possible to modify existing objective functions in order to integrate workload balance (Soriano et al., 2023). The most common objectives are the range objective, which minimises the time range between minimum time and maximum time (Matl et al., 2023) and the MinMax objective which minimises the maximum distance between the routes (Bhadoriya et al., 2024; Lu and Yue, 2019). The previously mentioned Okonjo-Adigwe (1988) introduce for example the MinMax objective in their paper. This approach is also adopted by other researchers in recent years (Matl et al., 2023). Venkatesh and Singh (2015) compare in their paper two objectives, the MinSum and MinMax. They investigated as well that the MinMax objective tends to balance the labour among salespersons and prioritise the disadvantaged routes to ensure fairness (Chen & Hooker, 2023; Bhadoriya et al., 2024; He & Hao, 2023). The only problem that is investigated by using the MinMax objective is, due to the fair distribution of customers, the overall efficiency and performance in obtaining optimal solutions decrease (Bertsimas et al., 2011; Bhadoriya et al., 2024). Moreover, only small instances calculated with MinMax can be solved optimally within a reasonable time (Soylu et al., 2015; Bhadoriya et al., 2024).

In conclusion, while the MinSum objective dominates mTSP research, the MinMax and the range objectives provide a valuable alternative for ensuring fairness and ethical considerations into optimisation problems. For the further course of this thesis, it is interesting to compare these three objective functions in regard to fairness to examine how the distribution of work and travel time varies or correlates with the individual routes. However, due to its complexity, since it has to minimise both the total travel time and the maximum tour duration, the objective range is not considered in a single formulation set. Therefore, only MinSum and MinMax are taken into account in the calculation and comparison of fairness in the mTSP. Furthermore, in order to ensure the MinSum objective function is both fair and efficient, a new fairness constraint is introduced which limits the maximum tour duration of the routes.

3 Programming Formulations

This chapter intends to be a prerequisite for the following analysis. Section 3.1 provides information and details on implementation and processing in C++ which is important to analyse the results. Followed by Section 3.2 which defines the four models that focus on the subject of investigation. These models are divided into two objective functions which are MinSum and MinMax. Where each model is formulated as a node-flow and as an arc-flow formulation. This section also discusses the general conditions of the vehicle environment. Finally, in Section 3.3 the mTSP instances and the needed data will be introduced as well as explained.

3.1 Implementation and Solution Approach

For the implementation part the software C++ is used to integrate the previously mentioned models, where the resulting code is compiled into 64-bit single-threaded code which uses MS Visual Studio 2022. The CPLEX 22.1.1 library incorporates into the C++ code in order to solve the instances in the following chapter. The hardware, which is used to achieve all results, is a standard PC with an Intel® Core™ i5-8250U processor with 1.6 GHz and 8 GB RAM under Microsoft Windows 11 Home. In order to achieve optimal results, a maximum execution time of 1,800 seconds is set to solve each instance.

The use of CPLEX within C++ includes the solution method BnB, which is already defined in Chapter 2.3.1. There are several reasons why this algorithm inherits advantages. Firstly, according to several surveys by Bektas (2006), Chen and Jia (2017), Cornejo-Acosta et al. (2023), He and Hao (2022) and Jiang et al. (2020), the BnB leads to exact solutions for many types of mTSP. Secondly, the BnB is a powerful method for solving mTSPs as it offers both efficiency and accuracy through its structured search and the use of bounds. Thirdly, it is particularly useful when exact solutions are required and offers the flexibility to be customised to different variants of the mTSP. And lastly, BnB is a reasonable method for evaluating and comparing results.

3.2 Models of Investigation

As mentioned in the introduction the subject of investigation are four models, which help to answer the overall research goals of this thesis. The models are categorised into two flow formulations from Chapter 2.1, which are widely used to model and solve network flow problems, such as the classic TSP, mTSP and VRP. The first formulation is the node flow formulation according to Sarin et al. (2014). The formulation includes variables, the so-called time variables, that represent the flow on the individual nodes of the network. This implies that the variables represent the amount of flow entering and leaving the individual nodes. The second formulation is the arc-flow formulation according to Maffioli and Sciomachen (1997), which includes time variables that correspond to the flow on individual arcs and thus indicate whether an arc is used and, if so, how much flow passes through this arc. Therefore, the arc flow formulation has more time variables compared to the node flow formulation. In the computational study, these two formulations are to be calculated with two models each. These two models consist of different objective functions of the MinMax and MinSum objective functions, which are defined in Chapter 2.1 and presented previously as variants of the mTSP in Chapter 2.3. The reason for choosing these specific models is, is firstly, that different objectives are taken into account. Each model pursues a different objective, so that a comprehensive comparison and evaluation of the performance and quality of the models is possible. And secondly, these models help to assess fairness. This is because the result of each objective function leads to different distributions of the workload along the vehicles.

Moreover, it should be noted that this thesis is considering a road network with homogeneous vehicles, means all vehicles are the same, there is no difference in the type or the capabilities of the vehicle. Additionally, all customers are the same, that means there are no restrictions as mentioned in Chapter 2.2 such as capacity restrictions or time windows are considered. Only the distance between the customers, the number of customers and the number of vehicles vary in the calculations. This is because the customers are located at different locations within the road network for each instance. As the different models are to be analysed for their quality and performance in terms of fairness, different sizes of instances are used for the calculation, which is explained in Section 3.4.

By structuring the research around these formulations, the aim is to provide a clear and systematic approach to solve complex network flow problems while addressing the overarching goals of fairness, efficiency, and performance evaluation.

3.2.1 Node-Flow Formulation According to Sarin et al.

The first set of formulations are node-flow formulated and are known from Sarin et al. (2014) explained in Chapter 2.1. As already mentioned, the time variables, defined as T_i, T_j, T_{jd} or T_{id}, lie on the nodes. The first two models are explained below.

Model 1 – MinSum mTSP according to Sarin et al. (2014)

The constraint definitions are the same as in 2.1. Accordingly, the objective aims to minimise the maximum travel time. Constraints (1b) and (1c) represent the in- and outflow restrictions, which means, that every vehicle of each instance must be used. Constraints (1d) and (1e) ensure that each customer has exactly one predecessor and exactly one successor. Constraints (1f) and (1g) are the MTZ Subtour elimination restrictions including the new time variable T_i. And the last constraint in this formulation (1h) represents the restriction on the decision variables.

Model 2 – MinMax according to Sarin et al. (2014)

The next formulation is considering the MinMax objective function, where the constraints are exactly the same as the constraints for the MinSum objective. The only difference is that the objective function is different. The objective in this formulation is MinMax, which aims to minimise the maximum time at the end depot d and not to minimise the overall efficiency. This is defined as follows:

$$z_{ATSP} = \min T_d \tag{2a}$$

3.2.2 Arc-Flow Formulation According to Maffioli & Sciomachen

Another approach to formulate the mTSP problem is the arc-flow formulation according to Maffioli and Sciomachen (1997) which is defined by Bianchessi et al. (2023). The formulation is much less used in the literature, because more variables are needed, which makes it more complex compared to the node-flow formulation. This is because Models 3 and 4 concentrating on arc-flow formulation new time variables are added which are T_{ij} and T_{ji}.

Model 3 – MinSum mTSP according to Maffioli and Sciomachen (1997)

Model 3 is similar to the Models 1 and 2 in Chapter 3.2.1 in terms of constraints, but in this case, the focus is more on the arcs. Therefore, by modifying the original constraint (1f), the focus is on the flow of the time variables T_{ij} and T_{ji}, which can be seen in the new constraint (2g) in Chapter 2.1.

The objective function is the same as in Model 1, as well as the flow conservation constraints (1b) to (1c) and assignment restrictions (1d) and (1e) that are identical to the constraints in this formulation, which means that they do not need to be specified again. The next constraints (2f) and (2g) are the extended MTZ subtour elimination constraints, which include the time variables on the arcs. Additionally, a further constraint (2i) is introduced, which represents the dependency of the time variable T_{ij} on x_{ij}. This implies that T_{ij} can only become positive if x_{ij} is used and is therefore positive. However, in this model, the constraint (2h) is not included as stated in Chapter 2.1, as it focuses on the maximum tour duration, which is not important to fulfil in a formulation with a MinSum objective function.

Model 4 – MinMax mTSP according to Maffioli and Sciomachen (1997)

The Model 4 once again contains the MinMax objective function from Model 2 and contains all of the constraints previously described in Model 4. The only difference between this model and the previous Model 3 is that the constraint (2h) is introduced instead of the constraint (2i). This ensures that the time spent at the depot is the maximum possible. This is represented as follows:

$$T_{ij} \leq T_d \quad for\ all\ i \in V \tag{4i}$$

Now that the four models are defined again, the next step is to introduce the instances that are to be used for the calculation.

3.3 Test instances

The computational study is tested on an instance set which is originally proposed by He and Hao (2023). The whole benchmark set is taken from the TSP-LIB under the link: http://comopt.ifi.uni-heidelberg.de/software/TSPLIB95/tsp/. The instance set is divided into eight instance classes, providing a comprehensive basis for evaluating various algorithms with 578 instances. The instance classes are categorised and defined in detail in Table 2. Overall, the instances are divided into two sizes with different numbers of cities (V) and salesmen (m). The instances mtsp51, mtsp100 and mtsp150 belong to the small-scale problems and are introduced by Carter and Ragsdale (2006). These classes consider V = {51, 100, 150} cities and m = {2, 3, 4, 5, 6, 7, 8, 9, 10} salesmen as two-dimensional symmetric problems. This means that the location of each node is defined by a coordinate (x, y). In this context, symmetrical means that the distances or times between two nodes are identical in both directions (Cornejo-Acosta et al., 2023). Thus, the distance from node A to B is the same as the distance from node B to A. The second problem size includes large-scaled problems that involve city sizes of V = {200, 318, 532, 783, 1173}. These instance classes consist of kroa200, lin318, att532, rat783 and pcb1173, which are introduced by Wang et al. (2017), among others. However, in both problem sizes, values between two and ten are considered for m.

Table 2: Overview and comparison of the benchmark problems.

No.	Instance classes	*V*	*m*	Scale	No. of benchmark problems
1	mtsp51	51	2, 3, 4, 5, 6, 7, 8, 9, 10	small-scale	183
2	mtsp100	100	2, 3, 4, 5, 6, 7, 8, 9, 10		
3	mtsp150	150	2, 3, 4, 5, 6, 7, 8, 9, 10		
4	kroa200	200	2, 3, 4, 5, 6, 7, 8, 9, 10	large-scale	395
5	lin318	318	2, 3, 4, 5, 6, 7, 8, 9, 10		
6	att532	532	2, 3, 4, 5, 6, 7, 8, 9, 10		
7	rat783	783	2, 3, 4, 5, 6, 7, 8, 9, 10		
8	pcb1173	1173	2, 3, 4, 5, 6, 7, 8, 9, 10		

As the calculation of the total number of 578 problems would require an enormous computational effort and is therefore not practical, this thesis will only focus on 48 instances in more detail to ensure that the selection is representative and easy to handle. Initially, the instances listed in Table 2 were filtered with regards to vehicles and customers. As a first step, all instances were excluded that utilised more than five vehicles. This significantly reduced the number of instances as well as the complexity level to solve instances with higher numbers of vehicles. In a second step, n cities are selected from the size V to be visited by m. Therefore, the instances with fewer than n = 20 and more than n = 50 are excluded, as these would not cover all instance classes. For example, the instance class mtsp51 does not have 60 cities that can be visited. Another reason for selecting these specific numbers of cities and vehicles is that, as mentioned in Chapter 3.1, each instance runs for a maximum of 1800 seconds, making the computational time enormous if larger instances were included. Therefore, small to medium-size problems are chosen to ensure both complexity and computability. The original number of cities is reduced to ensure practical manageability. As shown in Table 3, three customer values, n = {20,30,50}, are considered for each of the instances, each of which needs to be served by two types of vehicle fleets, m = {3,5}. This results in six combinations per instance class. Multiplied by the eight classes, this gives a total of 48 instances that are tested.

Table 3: Test instance combinations.

n	***m***
20	3, 5
30	3, 5
50	3, 5

The graph in each instance is structured around a road network, meaning the travel time from point A to point B represents an Euclidean distance and is measured in units. Furthermore, for each level of n, all m start and finish their individual tours at the same depot, as in previous mTSP studies. Additional parameters, such as delivery times or the prioritization of customers, are not considered in these instances. This structured approach ensures that the selected instances are representative, manageable and suitable for a detailed analysis of the mTSP models with MinSum and MinMax objectives. Now that the instances are explained in detail and the calculation basis for the computational solutions are created, the results of the calculations are presented and analysed in the next chapter.

4 Computational Results

This chapter presents the results of the computational study outlined before. The following sections focus on the analysis and comparison of the four models. Section 4.1 initially tests and compares the solutions of the MinSum Models 1 and 3. The goal is to evaluate the performance and overall efficiency of the two models using different numbers of customers and different numbers of vehicles. Furthermore, this section compares the models regarding their fairness and introduces examples of the route distribution. The following Section 4.2 also compares the Models 1 and 3 with one another, using the same criteria as in Section 4.1. Since the MinSum objective function leads to less favourable results in terms of fairness, a new constraint, the "Upper Bound Tour Duration", is introduced in this thesis, whose parameters can be varied. Furthermore, these changes, which are defined as percentages, are tested. Furthermore, these changes, which are defined as percentages, are subjected to testing. Finally, Section 4.3 compares the MinMax models two and four under the same criteria as in Section 4.1.

4.1 Results for MinSum

In the first experiment, the solution values of the MinSum mTSP Models 1 and 3 are to be compared, whereby model one is node-flow formulated and model four is arc-flow formulated. For the calculation itself, the 48 instances explained in detail in Chapter 3.3 are used. However, since the table with the complete solution view is too extensive for the main part of this thesis, only aggregated tables are discussed in this and the following sections for illustrative purposes and for a better understanding of the interpretation. The results for all 48 instances can be found in the Appendix 1.

Performance Results

Firstly, the efficiency of models one and three is analysed and compared in general. Therefore, Table 4 compares on the one hand the best objective function values "Obj." calculated by the BnB algorithm, to determine which MinSum model calculates the shortest total travel time in order to be the most efficient. On the other hand, the optimality gaps as defined by He and Hao (2023), "Opt. Gap", are evaluated to compare the difference between the upper bound (UB) and the lower bound (LB) of the objective in percent, which is calculated as follows:

$$Optimality\ Gap\ (\%) = 100 * (UB - LB) * UB$$

Here, UB, also known as the best objective value within maximum computation time, and LB, the best initial objective value. A gap of 0.00% indicates that the upper bound (UB) is equal to the lower bound (LB). A negative gap signifies an improved result, while a positive gap means that the UB is higher than the LB. Furthermore, the solution times, designated as "Time (s)" also called CPU time, for each instance are compared to determine how quickly each model can solve the different sized instances. A lower time generally indicates that the model is more efficient, while higher times suggest the opposite. The final category of interest is the "#BnB" This represents the average number of branch-and-bound nodes that are solved with the algorithm. This overall approach aims to provide detailed information about the quality and performance of each model. The values for the first model can be found in the column "Model 1: Node-flow formulated," and the values for the fourth model can be found in the column "Model 4: Arc-flow formulated." It is also notable that if a value is superior in comparison, it is highlighted in grey.

Table 4: Performance results of Model 1 and Model 3.

	Model 1: Node-flow formulated				Model 3: Arc-flow formulated			
n	Obj.	Opt. Gap (%)	Time (s)	#BnB	Obj.	Opt. Gap (%)	Time (s)	#BnB
20	74,769	0.00%	21	199,670	74,769	0.00%	38	199,501
30	84,385	1.56%	228	616,727	84,385	1.56%	228	461,160
50	104,539	0.22%	571	730,411	104,539	0.18%	456	298,795
Avg.	87,897	0.59%	273	515,605	87,897	0.58%	241	319,819

As illustrated in Table 4, both models solve the individual instances almost optimally. The objective values for both models are identical with a total average (Avg.) objective of 87,897 units. This indicates that both models perform equally in terms of minimising the total travelling time of all vehicles. Additionally, the optimality gaps for both models are very close, with Model 1 having an average optimality gap of 0.59% and Model 3 having an average of 0.58%. Both models show a high degree of reaching near optimal solutions, and have a high effectiveness, with Model 3 being marginally better. Especially for the instances with 50 customers, the Model 3 shows a better performance with regard to the optimality gap, which is only 0.18% on average in Model 3, whereas it is 0.22% in Model 1. The same is observed for computing time. For instances with 50 customers, the CPU time is 456 seconds on average, whereas the Model 1 takes significantly longer at 571 seconds. The model 3 also requires less time on average to solve the instances in the overall result. However, it can be examined that Model 1, like Model 3, optimally solves the instances with 20 customers, but Model 1 takes less time and performs faster. Another observation regarding Model 3 is, that it is also performing significantly better in terms of the number of BnB nodes solved. On average, Model 3 solves 319,819 nodes compared to Model 1's 515,605 nodes. This indicates that Model 3 is more efficient in navigating the solution space, especially again in larger instances with 50 customers. Another remarkable observation is, that both models struggle to solve the instances with 30 customers. Firstly, the optimality gap is significantly higher with over 1% on average compared to the other instance sets, and the BnB nodes are also notably increased in comparison to the other sets.

Fairness Results

In the next step, the fairness aspect within the different models is analysed, examining how evenly customers and maximum travel times are distributed when the objective function changes and how this impacts the objective results. As illustrated in Table 5, the maximum tour length and the range time between the shortest and longest times are discussed first to achieve this analysis.

Table 5: Fairness results of Model 1 and Model 3.

	Model 1: Node-flow formulated		**Model 3: Arc-flow formulated**	
n	Max Tour	Range time	Max Tour	Range time
20	51,469	46,758	51,469	46,758
30	67,293	63,285	67,293	63,285
50	89,049	86,218	89,049	86,218
Avg.	69,270	65,420	69,270	65,420

As observed earlier in the upper section, the values for Model 1 and Model 3 are identical. Additionally, it can be clearly seen that the values for the maximum route also increase as the number of *n* increases. A comparison of the tour duration and the range time reveals that the values are very close to each other. This implies that the route or the tour durations are unevenly distributed and therefore not fair. To demonstrate this, a common example will be used for both models to show how the distribution of routes behaves. An instance from Appendix 1 is used to take a closer look at the routes. The instance att532_3_21 has three salesmen and must serve 21 cities. As the values of Model 1 and Model 3 are the same, there is no comparison between the models itself, the analysis applies to both models. Table 6 shows the allocation of individual routes for the selected instance and provides a comparative analysis of the MinSum target function, the maximum tour duration, and the range time.

Table 6: Route allocation of Model 1 and Model 3 using instance att532_3_21.

	Model 1: Node-flow formulated and Model 3: Arc-flow formulated
	Routes
Route 1	0 1 21
Route 2	0 4 21
Route 3	0 5 6 11 10 8 7 9 12 13 17 19 16 14 15 18 20 2 3 21
MinSum Obj.	54,415
Max. Tour Duration	35,335
Range Time	28,449

While solving this, it becomes clear that Route 1 that Route 2 only have to serve one customer each, whereas Route 3 includes all remaining *n*. Furthermore, the comparison of the MinSum objective function and the maximum route duration reveals an additional insight into this unbalanced distribution of work. The objective function value and the value of the maximum route duration are observed to be closely aligned, where the Route 3 represents 65% of the total travel time, with a value of 35.335. It can thus be concluded that the remaining two routes result in shorter travel times. This is particularly demonstrated by the range time, which also represents a considerable discrepancy between the shortest and longest travel times with a value of 28,449 units. This shows once again how unbalanced the routes are and how unfair the MinSum Models 1 and 3 are.

In conclusion, the results indicate that both models achieve near-optimal solutions with similar performance metrics, although Model 3 is slightly more efficient. The performance analysis shows that Model 3 has marginally better optimality gaps, faster computation times, and fewer BnB nodes explored, especially in larger instances. The fairness analysis reveals that both models distribute the routes unevenly, with significant discrepancies in maximum tour duration and range times, highlighting an imbalance in workload distribution and an overall lack of fairness.

4.2 Results for MinSum with added Constraint

A review of the MinSum mTSP results demonstrates that models 1 and 3 are not optimal in terms of fairness. Accordingly, the following study is an extension of the previous models by introducing a new fairness constraint to test whether the MinSum model can be improved. In particular with regards to the overall performance, but more importantly with regard to the objective value and the fairness of the route distribution. To enhance fairness in the distribution of routes among salesmen, the constraint aims to limiting the tour duration of the longest route, which may ensure a more balanced workload distribution. The new constraint, which is referred to as UB-TD in the following chapters, is defined as follows:

$$Upper\ Bound\ Tour\ Duration = max\ Tour\ Duration * 1 + \frac{p}{100}$$

This constraint sets an upper limit for the maximum route length, based on the minimised maximum route duration T_d of the MinMax mTSP Models 2 and 4. The results presented in Appendix 6 are used due to the fact that the MinMax formulation results in a more balanced distributed solution, as explained in Chapter 2.4. Subsequently, the maximum tour duration is increased once more by a percentage *p* (in %), which can impact the flexibility of the upper limit. Consequently, in further experiments, the upper limit can be either tightened or loosened by a small or large percentage, respectively, in order to observe the effects on the objective results. The following Sections 4.2.1 to 4.2.2 present a comparative analysis of the results obtained in Section 4.1 and the impact of the new constraint with different percentages. Hereby, Section 4.2.1 focuses on the formulation according to Sarin et al. (2014) and Section 4.2.2 on the formulation according to Maffioli and Sciomachen (1997). In the final Section 4.2.3, all results are compared again in order to identify any changes in terms of performance and fairness between the two mTSP formulations. To obtain different comparison values, the maximum tour duration is increased by the specified percentages: 5%, 10%, 15% and 20%. The reason for choosing these values is that percentages below 5% increase the computational effort enormously, and all percentages above 20% are ineffective as there is no improvement to the initial solutions.

4.2.1 Results According to Sarin et al.

As mentioned before, this section starts by changing the percentage values of the multiplier within the new constraint and compares these with the initial solutions of Chapter 4.1 according to the Sarin et al.'s (2014) formulation. Same as before, the aim is to compare these values according to the overall performance and their changes regarding fairness.

Performance Results

To compare the performance and the quality of the results, the same subjects as mentioned before are used. In order to ensure a better comparison of the initial MinSum results and the modified constraint results, the mentioned test criteria are shown in Table 7. The detailed solution values per individual instance can be found in Appendix 2 to 5. The following table compares the objective values (measured in units), the number of BnB nodes, the computational time and the optimality gap, for the different aggregated instances across varying percentages of the new UB-TD constraint.

Table 7: Performance results of Model 1 with new constraint.

	n	Initial	5%	10%	15%	20%
Obj.	20	74,769	93,899	89,883	83,716	83,084
	30	84,386	100,437	95,481	92,497	91,901
	50	104,539	107,999	107,026	106,736	106,054
	Avg.	**87,898**	**100,778**	**97,463**	**94,317**	**93,680**
#BnB	20	199,676	4,671,326	4,499,718	2,655,083	1,686,857
	30	616,727	3,856,098	3,245,856	3,093,584	2,793,733
	50	730,411	1,206,233	1,079,731	1,025,776	863,543
	Avg.	**515,605**	**3,244,552**	**2,941,769**	**2,258,148**	**1,781,378**
Time (s)	20	21	956	938	563	356
	30	228	1,473	1,282	1,315	1,115
	50	571	1,010	1,003	894	954
	Avg.	**273**	**1,145**	**1,074**	**924**	**808**
Opt. Gap	20	0.00%	8.26%	5.33%	1.74%	1.62%
	30	1.56%	11.63%	7.66%	5.55%	4.52%
	50	0.22%	5.65%	4.12%	3.58%	1.99%
	Avg.	**0.59%**	**8.51%**	**5.70%**	**3.63%**	**2.71%**

Firstly, it is notable that the initial solution consistently achieves the lowest MinSum value, which is used as a basis for comparison. This indicates that the original solution is the most effective in minimising the total travel time. Furthermore, an important observation is that as the percentages of the constraint increases, the MinSum objective values for different instance sizes decrease. In particular, the constraints with 15% and 20% lead to values that are almost as low as those of the initial solution, where the 15% constraint leads to 94,317 units and the 20% constraint to 93,680 units. This indicates that higher percentages do not have a significant impact in lowering the efficiency compared to the initial solution. In contrast, the 5% constraint increases the objective value significantly by 15%, indicating higher complexity in solving the problem. This shows that tighter constraints are more complex to solve the problem efficiently.

The second aspect to look at is the number of BnB nodes. Again, one can see that the initial solution requires the fewest nodes across all instances to achieve an optimal result, with the fewest nodes required for small instances and the most for large instances. Overall, it is observed that the fewest nodes are required for n = 50, in contrast to n = 20. For example, with a 10% constraint it is observed that 4,499,718 BnB nodes are required for n = 20, but only 1,079,731 nodes are required for n = 50, which is a quarter of the first result. However, once again the tighter the constraint, the fewer nodes are needed.

The third part of Table 7 compares the overall computational time in seconds, that represents the time which is required to solve a problem. It shows that the initial solution without additional restrictions has the lowest solution times. However, it should be noted that n = 30 has the highest solution time for all UB-TD constraints, followed by n = 50 and n = 20. A more detailed examination of the individual instances (Attachments 2 to 5) reveals that for n = 30, the instance classes mtsp150, kroa200 and rat783 in particular require a greater amount of CPU time compared to n = 50, regardless of the number of vehicles. This unexpected result suggests that the problem or the data set may present a different level of complexity, potentially due to the varying distances in the coordinate system, which is not evident from the analysis. The 5% constraint significantly increases the computational time compared to the initial solution, with n = 20 taking an average of 956 seconds instead of 21 seconds. In contrast, the 10%, 15% and 20% constraints have higher solution times than the initial solution but tend to move closer to the initial times as the percentage increases. This suggests that a higher percentage of constraints leads to a lower CPU time required. For n = 20, the solution times are lower across all constraints compared to n = 30 and n = 50, especially for the values 15% and 20%. For n = 50, the solution time decreases with higher percentages, but is generally lower than for n = 30 for the same constraints. This is unexpected and should be further investigated, because normally a larger problem size correlates with a longer total travel time due to the higher complexity of solving large instances and distances. Overall, the computational effort increases significantly with additional constraints, especially at the 5% level, indicating higher complexity.

The next part of compares the optimality gaps in percent of the initial solution and the varying constraint levels to investigate how close the solution is to the optimal value, with lower values indicating better performance. Initially, the optimality gaps are very close to optimal, with particularly low gaps for smaller instances with n = 20 and bigger instances with n = 50. Compared to the initial solutions, the overall optimality gaps increase by introducing the new constraint, indicating a deviation from the almost 0% gaps initially observed. For 15% and 20%, the small instances with n = 20 is the easiest to solve optimal, followed by n = 50 and n = 30 which is the most challenging. In contrast, for 5% and 10%, the big instances with n = 50 is solved most optimally instead of the smallest instances. Furthermore, n = 30 consistently shows a higher average optimality gap than the other sizes, reflecting increased complexity. Introducing a 5% constraint significantly increases the optimality gaps across all sizes, demonstrating that such constraints complicate achieving optimal solutions. However, as the percentage constraints increase, the optimality gaps improve, indicating better performance with higher percentages.

A more detailed examination of the three tables reveals that the introduction of the new UB-TD constraints increases the complexity and quality of the mTSP model proposed by Sarin et al. (2014), particularly with regard to the computational time and the optimality gaps. It is also noteworthy that the constraints with lower percentages such as 5% demonstrate a less effective reduction in the total travel time, which is due to the tightness of the upper bound. Nevertheless, this indicates that the distribution of routes has also changed in terms of fairness, which is examined in the following section.

Fairness Results

As previously stated, the quality of the results is reduced as a consequence of the introduction of the constraint in relation to the MinSum target function. However, it is now relevant to test how the maximum journey duration and the range time have changed in order to obtain information about fairness. As previously stated, the initial results presented in Chapter 4.1 are once again compared with the different constraint levels and are shown in Table 8.

Table 8: Fairness results of Model 1 with new constraint.

	n	Initial	5%	10%	15%	20%
Max. travel duration	20	51,469	30,846	32,065	33,935	34,381
	30	67,294	37,946	40,382	41,599	42,185
	50	89,049	71,043	73,110	73,595	76,000
	Avg.	**69,271**	**46,612**	**48,519**	**49,709**	**50,855**
Range time	20	46,758	16,384	21,708	28,297	29,216
	30	63,286	31,112	36,348	37,571	38,022
	50	86,218	68,212	70,279	70,764	73,169
	Avg.	**65,420**	**38,569**	**42,778**	**45,544**	**46,802**

Firstly, the initial solution shows the largest values across all problem sizes. Interestingly, the 5% constraint achieves the best values this time, with an average of 46,612 units, which is a decrease of approximately 32% compared to the initial average solution. As the constraint percentage increases, the maximum travel duration values also increase. Compared to 5%, the 10% to 20% values increase gradually to 48,519 units, 49,709 units and 50,855 units. In addition, larger problem sizes lead to higher values. These observations suggest that different constraint levels improve fairness compared to the initial solution, particularly when they get tighter. Therefore, lower constraint percentages lead to a better performance and fairer solutions across different instance sizes.

The next step is to examine how the range between the shortest and longest tour duration has changed. Table 8 demonstrates that the constraint with 5% provides the optimal results, while the initial solution presents the worst outcomes, which strengthens the thesis, that the MinSum Model gives generally poor fairness results. A reduction of approximately 41% is observed between the constraint with 5%, which has an average range value of 38,569 units, and the initial value of 65,420 units. With regard to the other percentages, it can be stated that these also getting closer to the initial value once more, the looser the UB-TD constraint becomes. This indicates that, in general, with the introduction of the constraint, the longest and shortest routes should have approached each other and become more balanced in terms of the amount of work involved.

After the range time and the maximum route duration are examined, an example is now used to show how the distribution of customers within the routes has changed. For this purpose, instance att532_3_21 is used as an example. As demonstrated in Chapter 4.1, the routes for the normal MinSum Model are not balanced at all. Where the objective value is 54,415 units, the maximum tour duration is 35,335 units and the range time is 28,449 units. Table 9 presents a comparison of the results in the route distribution for the two specified restriction levels, namely 5% and 10%.

Table 9: Route allocation of Model 1 with 5% and 15% constraint using instance att532_3_21.

	Model 1 with 5% constraint	**Model 1 with 15% constraint**
	Routes	Routes
Route 1	0 1 21	0 1 21
Route 2	0 3 2 20 18 15 4 21	0 4 15 18 20 2 3 21
Route 3	0 6 8 7 9 12 13 17 19 16 11 10 5 21	0 5 10 11 14 16 19 17 13 12 9 7 8 6 21
MinSum Obj.	56,108	56,108
Max. Tour Duration	26,405	26,405
Range Time	19,519	19,519

As is observed, the 5% constraint and the 15% constraint get identical results in the objective, in the maximum tour duration and in the range time. The only difference is that the order of the visited cities changes in the routes. Compared to the normal MinSum model yields this to a 3% increase in the objective function compared to the initial solution, which leads to a poorer performance and lower quality in the objective results. But in contrast, the maximum tour length and range time are simultaneously decreased by 25% and 31%, respectively, which results in a more balanced distribution of workload among the salesmen.

4.2.2 Results According to Maffioli & Sciomachen

Analogous to Chapter 4.2.1, this section compares the different constraint levels with the initial solution from Chapter 4.1, but this time according to the formulation by Maffioli and Sciomachen (1997). It should be noted at the start of this process that not all instances can be solved optimally within the given time, particularly as the constraints becomes smaller e.g. 5%-constraint. Therefore, to make sure, that the results are valid to compare the instances att532_5_51, kroa200_5_51, lin318_3_51, lin318_5_31, lin318_5_51, mtsp100_3_51 and mtsp150_5_51 are removed in following results. Thus, the instances decrease from 48 to 41. The exact instances can be found in Appendix 2 to 5.

Performance Results

Starting again with the comparison of the performance and the quality of the results. The following Table 10 compares the objective values, the number of BnB nodes, the CPU time and optimality gap for the different instances across varying percentages of the new UB-TD constraint.

Table 10: Performance results of Model 3 with new constraint.

	n	Initial	5%	10%	15%	20%
Obj.	20	74,769	94,522	90,297	84,137	83,638
	30	86,452	108,342	102,627	99,561	98,106
	50	78,007	90,960	88,442	85,653	85,136
	Avg.	**79,833**	**93,737**	**90,413**	**86,792**	**86,106**
#BnB	20	199,501	1,092,799	829,476	76,381	50,206
	30	491,902	1,612,780	1,052,427	1,101,254	908,070
	50	241,509	407,829	499,689	466,892	431,956
	Avg.	**310,971**	**1,037,803**	**793,864**	**548,176**	**463,410**
Time (s)	20	38	416	332	24	17
	30	242	1,378	935	830	675
	50	305	1,801	1,721	1,474	1,258
	Avg.	**178**	**1,106**	**892**	**673**	**561**
Opt. Gap	20	0.00%	3.88%	1.61%	0.01%	0.01%
	30	1.67%	7.76%	2.96%	0.96%	1.18%
	50	0.01%	13.16%	9.08%	5.92%	3.54%
	Avg.	**0.61%**	**7.56%**	**3.93%**	**1.80%**	**1.30%**

Firstly, the initial solution demonstrates consistent superiority in achieving the lowest MinSum value for all the instance sets and on average with 79,743 units. Nevertheless, it is important to note that the middle-size instances with n = 30 are always above the average value. Particularly the smaller instance classes mtsp100, mtsp150 and kroa200 lead to higher solution values over 200,000 units, which indicates a poor performance regarding the minimisation of the total travel time. For the instance set n = 50, the results are on average lower than the total average. For example, the result for the 5% constraint is the most effective at n = 50 compared to the other two instance sets, with 90,960 units. However, it can be observed that as the percentages increase, the MinSum objective values approach the initial solution, thereby demonstrating that the process becomes more efficient by loosen the constraint. For example, for n = 20, the initial solution calculates an average objective value of 74,769 units. For the 10% constraint, a value of 90,297 units is calculated, while for the 20% constraint, a value of 83,638 units is calculated, which thus approaches the initial solution once more.

The second aspect is the number of BnB nodes. This time, one can see that the initial solution requires only the fewest nodes across the instances with n = 30 and n = 50 to achieve an optimal result, with the fewest nodes required for n = 20 and n = 50 and the most for middle-large instances n = 30 with 491,902 nodes. Overall, it is observed that the fewest nodes are required again for n = 50, in contrast to n = 20. This time the 10% constraint requires 829,476 BnB nodes are required for n = 20, but only 499,689 nodes are required for n = 50. But overall, the tighter the constraint, the more nodes are needed. It is noteworthy that for the 20% constraints, only 50,206 nodes are needed to solve the instances with n = 20, which is less than the initial solution, even though the aggregated objective function is significantly higher than the initial solution. Which indicates that especially the looser constraints performing better than the initial formulation, which requires particularly more than 1,000,000 nodes to solve the instance rat783.

The third part of Table 10 again compares the total computing time in seconds, but the values have changed compared to the solutions of the optimal solution. The table shows that the initial solution without additional constraints gives the shortest solution times for n = 30 with 242 seconds, n = 50 with 305 seconds and on average with 195 seconds. Only for n = 20 does the 15% and 20% constraint provide a better solution, but the values here are close to each other. For n = 20, the 5% and 10% constraint are the highest, as with the objective function, this suggests that these two constraint levels are not suitable for smaller instances. For instances n = 50, the computation times are also high, which indicates additional effort in solving the instances. However, it can be said that the tighter the constraint, the more computing time it requires and is therefore more complicated to solve.

The last performance criterion is again the optimality gap. The initial solution has overall the lowest optimality gap with 0.61%. As illustrated in Table 10, it is evident that the constraints with 5% and 10% encounter difficulties to solve the different instances optimally. Particularly, the large instances with *n* = 50 result in an optimality gap of 13.16% for the 5% constraint, which is approx. 13% higher than the initial solution. The 20% constraint demonstrate the greatest performance among the other constraints in the context of large instances. In the case of smaller instances with *n* = 20 and *n* = 30, the 15% and 20% constraints demonstrate the greatest efficiency and are close to the values of the original solution.

The results demonstrate that the initial MinSum formulation represents the most efficient approach when compared to the different constraint levels. It is particularly evident that the performance per constraint level weakens with tighter upper bound. It is noteworthy that especially the large instances with *n* = 50 lead to the weaker results regarding the computational time and the optimality gap, as it takes more time to solve due to more complexity in allocating more customers to the vehicles. But compared to that n = 30 have weaker solutions regarding the overall objective function value and the BnB nodes.

Fairness Results

In regard to the fairness of routes, the maximum travel duration and the range time is compared first in Table 11. It reveals an important finding: the initial solution performs significantly worse across all three instances compared to the different constraint levels.

Table 11: Fairness results of Model 3 with new constraint.

	n	Initial	5%	10%	15%	20%
Max. travel duration	20	51,469	30,701	31,886	33,145	33,492
	30	69,117	34,904	36,326	37,068	38,359
	50	67,756	35,860	37,229	38,212	38,511
	Avg.	**61,898**	**33,497**	**34,813**	**35,816**	**36,497**
Range Time	20	46,758	15,731	21,529	25,316	25,663
	30	64,882	18,152	23,320	24,391	27,327
	50	65,445	22,364	25,655	28,226	28,527
	Avg.	**57,946**	**18,235**	**23,191**	**25,688**	**26,970**

It is noteworthy that the tightest constraint produces the optimal solutions, with an average of 33,497 units, representing a 46% reduction in comparison to the initial solution. As previously discussed in Chapter 4.1, the initial solution is identified to be not fair. However, the results obtained for the 5% constraint indicate that it is a more balanced approach, with route durations distributed more evenly. The 20% constraint also shows a notable reduction in the maximum route duration, with an average of 36,497 units. It is evident that as the constraint becomes loosen, the maximum tour duration increases.

The last step again is to check, how the range times have changed according to the formulation of Maffioli and Sciomachen, which is also shown in Table 11. As with the maximum route duration, the 5% constraint yields again the lowest results, with an average of 18,235 units. This indicates that the discrepancy between the longest and shortest routes is minimal, resulting in a more balanced distribution of routes. In addition, this value is reduced by 69% in comparison to the initial solution. The values of the remaining constraints are also lower, indicating that these should be considerably fairer and more balanced than the initial solution. However, it is also observed that as the constraint becomes loosen, the range times increase, and the individual routes become more unbalanced. It is also noteworthy that, the bigger the instances are the higher the range value gets. This may also have contributed to the increased computational effort previously described.

In order to conclude whether the goal of achieving fairness is met, the instance att532_3_21 from the 5% and 15%constraint is used as an illustrative example once again. The Table 12 demonstrates the following routes:

Table 12: Route allocation of Model 3 with 5% and 15% constraint using instance att532_3_21.

	Model 3 with 5% constraint	**Model 3 with 15% constraint**
	Routes	Routes
Route 1	0 2 3 21	0 1 21
Route 2	0 4 18 20 15 21	0 4 15 18 20 2 3 21
Route 3	0 5 10 11 14 16 19 17 13 12 9 7 8 6 1 21	0 5 10 11 14 16 19 17 13 12 9 7 8 6 21
MinSum Obj.	63,055	56,108
Max. Tour Duration	25,427	26,405
Range Time	10,628	19,519

It can be observed that the constraint with 15% has no changes to the Model 1 from the chapter before. The value of the objective increases by 3% compared to the initial solution and the maximum tour duration as well as the range time decreased by 25% and 31%. In contrast, the results of the 5% improved and show that the MinSum objective value of the 5% constraint is 16% higher than the initial solution value (54,415 units), but the maximum travel duration (25,427 units) is 28% lower and the range time is 63% lower. Thus, it can be stated that the introduction of the constraint results in an increase in the MinSum objective function and also in an increase in the computational effort, which in turn leads to a reduction in the quality of the results. However, the constraints can also ensure a better balance of the routes, as can be seen from the range time and maximum travel duration. It can also be concluded that Model 3 can achieve better results in terms of fairness with a tighter constraint.

4.2.3 Comparative Results of Different Formulations Under New Constraints

After testing and comparing the initial values of the formulations proposed by Sarin et al. (2014) and Maffioli and Sciomachen (1997) with the different constraint levels, this chapter will once again present a comparison of the results of both formulations in order to determine which formulation performs better and distributes the routes more fairly. Starting with the performance results, same as before, the Table 13 contains the results of the MinSum objective, the required BnB nodes, the solution time and the optimality gaps within the difference instances. But this time, the two models, Model 1 (Node-flow formulated) and Model 3 (Arc-flow formulated), are positioned directly one above each other. Moreover, as in Section 4.2.1, the following results include only 41 instances, as the BnB algorithm is not consistently able to solve all instances in an optimal manner. The exclusion of these instances ensures the validity of the comparative analysis of the models.

Table 13: Comparison of performance results under new constraints.

		MinSum Obj.			**#BnB**			**Time (s)**			**Opt. Gap**		
Instances		20	30	50	20	30	50	20	30	50	20	30	50
Model 1: Node-flow	Initial	74,769	86,453	78,007	199,676	657,277	299,320	21	243	198	0.00%	1.66%	0.01%
	5%	93,899	101,914	81,213	4,671,326	3,784,798	1,045,926	957	1,449	797	8.26%	10.42%	6.25%
	10%	89,883	97,610	80,467	4,499,718	3,173,620	913,796	938	1,246	859	5.33%	7.08%	4.80%
	15%	83,716	94,637	79,948	2,655,083	3,038,223	831,424	563	1,283	772	1.74%	5.22%	3.83%
	20%	83,084	94,001	79,099	1,686,857	2,690,722	626,258	356	1,069	756	1.62%	4.17%	1.65%
	Avg.	**85,070**	**94,923**	**79,747**	**2,742,532**	**2,668,928**	**743,345**	**567**	**1,058**	**678**	**3.39%**	**5.71%**	**3.31%**
Model 3: Arc-flow	Initial	74,769	86,452	78,007	199,501	491,902	241,509	38	243	305	0.00%	1.67%	0.01%
	5%	94,522	108,342	90,960	1,092,799	1,612,780	438,314	416	1,378	1,802	3.88%	7.76%	13.16%
	10%	90,297	102,627	88,442	829,476	1,052,427	499,689	332	935	1,721	1.61%	2.96%	9.08%
	15%	84,137	99,561	85,653	76,381	1,101,254	466,892	24	830	1,474	0.01%	0.96%	5.92%
	20%	83,638	98,106	85,136	50,206	908,070	431,956	17	675	1,258	0.01%	1.18%	3.54%
	Avg.	**85,473**	**99,018**	**85,640**	**449,672**	**1,033,287**	**415,672**	**166**	**812**	**1,312**	**1.10%**	**2.91%**	**6.34%**

A review of the data in the table reveals that Model 1 consistently reaches the optimal average objective value across all instances, including both the initial solution and the varying constraint solutions. Furthermore, an examination of Appendix 2 of the 5% constraint reveals that Model 1 is the superior option in 21 out of 41 instances, while Model 3 is the dominant model in only three instances. These instances are characterised by a high number of vehicles and customers, specifically five vehicles and 50 customers, within instance classes rat783 and pcb1173. The results demonstrate that Model 3 does not show improvements on average in comparison to Model 1. As the constraints become looser, the optimal objective functions tend to decrease and improves therefore the objective function.

However, when examining the average number of BnB nodes, it becomes evident that Model 3 performs better across all instances. The initial solutions consistently require the fewest nodes. Notably, for *n* = 20, the constraints 15% and 20% in Model 3 perform exceptionally well, with only 76,381 and 50,206 BnB nodes, respectively. Another interesting observation, which is already noted in the comparison of objective values, is that the solutions for *n* = 30 involve a significantly high number of nodes in both models. This is particularly visible in Model 3, where the average reaches 1,033,287 nodes. Additionally, both models are remarkably low for instances with *n* = 50, which was also visible in the objective function part.

The results of the CPU time indicate that Model 3 is the more efficient of the two models, particularly for small instances (*n* = 20 and *n* = 30), where it provides faster results. The average CPU time for Model 3 is 166 seconds and 812 seconds, compared to 567 seconds and 1,058 seconds for Model 1. This suggests that Model 3 is more efficient. In contrast, when examining the results for the larger instances (*n* = 50), it becomes evident that Model 3 requires a significantly longer solution time than Model 1, with an average of almost twice as long. This indicates that the larger instances are more challenging to calculate than the smaller ones in Model 3. Nevertheless, an examination of the original solutions reveals that Model 3 requires a greater amount of time overall than Model 1, which suggests a higher computational complexity.

Similar results can be observed with respect to the optimality gaps, which remain below 15% on average in both models. Nevertheless, it is notable that Model 3 also produces superior or even more optimal results for the smaller instances, with values of 1.10% and 2.91% in comparison to 3.39% and 5.71%. A comparison of the objective values of Model 3, which are higher than in Model 1, reveals that the higher lower bounds in Model 3 are the reason for the small optimality gap. Furthermore, it is evident that the optimality gaps of Model 3 have increased significantly in comparison to Model 1, which suggests that the computational effort has become more complex. It is also important to note that the application of more tight constraints will generally result in inferior results and a greater computational effort.

The Table 14 gives an overview of the fairness results again comparing both models with the initial solution values and the values of the added UP-TD constraint. This time the maximum tour duration and the range time are compared.

Table 14: Comparison of fairness results under new constraints.

Instances		Max Tour Duration			Range Time		
		20	30	50	20	30	50
Model 1: Node-flow	Initial	51,469	69,118	67,756	46,758	64,883	65,445
	5%	30,846	38,684	57,510	16,384	31,436	55,199
	10%	32,065	41,188	60,394	21,708	36,926	58,083
	15%	33,935	42,481	61,754	28,297	38,227	59,443
	20%	34,381	43,106	64,157	29,216	38,707	61,846
	Avg.	**36,539**	**46,915**	**62,314**	**28,473**	**42,036**	**60,003**
Model 3: Arc-flow	Initial	51,469	69,117	67,756	46,758	64,882	65,445
	5%	30,701	34,904	35,860	15,731	18,152	22,364
	10%	31,886	36,326	37,229	21,529	23,320	25,655
	15%	33,145	37,068	38,212	25,316	24,391	28,226
	20%	33,492	38,359	38,511	25,663	27,327	28,527
	Avg.	**36,139**	**43,155**	**43,513**	**27,000**	**31,615**	**34,043**

The data presented in this table demonstrates that Model 3 exhibits the optimal performance in both categories, namely maximum tour duration and range time. The introduction of the constraint in both models results in significantly better outcomes compared to the initial solution. A detailed comparison of the maximum tour duration reveals that the 5% constraint, as previously mentioned, is the least efficient in relation to the MinSum objective. However, it should be noted that this constraint calculates the lowest maximum value in both models. Model 3 calculates the lowest value for the instances with n = 20, yielding 30,701 units, and the highest value in the line for the instances with n = 50, yielding 35,860 units. As the constraint becomes weaker, the values of the other constraint levels increase. A similar trend is observed in the range time, where the values associated with the 5% constraint of Model 3 are notably lower than the other values. These results suggest that the introduction of the constraint may facilitate a greater degree of fairness. In the case of Model 3, it can be seen from the example in Section 4.2.1 and 4.2.2 that fairness is likely to be more optimal than in Model 1.

Overall, the experiments demonstrate that adding a fairness constraint to the MinSum models significantly improves the balance of the routes, ensuring a fairer distribution of workload among salesmen. While there is a slight increase in computation time, the enhanced fairness and overall performance of the models justify the additional constraint. This approach provides a more equitable solution to the mTSP, aligning with the goals of efficiency and fairness. Regarding the comparison of both formulations and also the change of the constraint level, one can say that the Model 3 with the formulation according to Maffioli and Sciomachen (1997) performing better with the additional constraint better than the Model 1 with the formulation according to Sarin et al. (2014) and leading therefore to a better overall performance quality. In addition, the Model 3 is performing well in terms of fairness, as it is effectively balancing the routes, with a maximum tour duration and range time that are both relatively low.

4.3 Results for MinMax

The third experiment is structured in a manner identical to that of the first experiment. However, in this scenario, the results of the MinMax Objective mTSP Models 2 and 4 are compared. Similarly, only three groups of instances are presented here for illustrative purposes, the complete calculation can be found in Appendix 6.

Performance Results

In Table 15, the values look significantly different compared to the first experiment. This time the values in column "Obj." determine which MinMax model minimises the maximum route time at the depot the most. The values for the second model can be found in the column "Model 2: Node-flow formulated," and the values for the fourth model can be found in the column "Model 4: Arc-flow formulated." And again, the values with better solutions are marked in dark grey.

Table 15: Performance results of Model 2 and Model 4.

	Model 2: Node-flow formulated				**Model 4: Arc-flow formulated**			
n	Obj.	Opt. Gap	Time (s)	#BnB	Obj.	Opt. Gap	Time (s)	#BnB
20	30,461	69.00%	1,800	7,313,381	29,933	42.85%	1,800	3,096,980
30	38,429	80.84%	1,800	4,142,755	33,265	61.09%	1,800	1,008,623
50	71,639	91.69%	1,800	2,393,958	42,672	83.28%	1,800	139,213
Avg.	46,843	80.51%	1,800	4,616,698	35,290	62.41%	1,800	1,414,939

As can be seen in Table 15, both models performing differently in regard to the four categories. Firstly, the MinMax objective function. A general examination of the table reveals that, in general, the Model 4 performs better than the Model 2. An examination of the objective function values in the "Avg." column reveals that Model 4 calculates a significantly lower average MinMax objective function value of approximately 35,290 units, in contrast to Model 2, which calculates an average value of 46,843 units, representing a difference of approximately 28%. This may indicate that Model 4 is more effective at minimising the longest tour. Furthermore, upon closer examination of the three different problem sizes, it can be observed that for small and middle problems n = 20 and n = 30, the objective values are in similar ranges. However, when n = 50, it can be seen that Model 2 has more difficulties in solving the MinMax possibly low, compared to the Model 4. This can also be observed in the optimality gaps. Overall, both models perform relatively poorly. Nevertheless, Model 4 performs better than Model 2, with an average gap of 62%. It is noteworthy that both models require a considerable amount of time to solve the individual instances. In the example instances, it can be observed that both models always require the full 1800 seconds to deliver the results, which are still far from optimal overall. This indicates that the computational effort is extremely high compared to the first experiment in Section 3.1. Another point is, Model 4 explores significantly fewer BnB nodes compared to Model 2, with an average of 1,414,939 nodes versus 4,616,698 for Model 2. This indicates that Model 4 is more efficient in navigating the solution space. This is particularly visible for larger instances (*n* = 50), where Model 4 requires significantly fewer nodes to solve the instances.

Overall, these findings indicate that Model 4 offers more efficient and optimal solutions. Despite both models reaching the computation time limit, Model 4 demonstrates superior performance and efficiency, making it a more effective choice for solving the mTSP with the MinMax objective.

Fairness Results

In terms of fairness, it can be said that both models perform in a highly fairly manner overall compared to the MinSum Model in Section 4.1. This implies that the routes are both fair and balanced for each instance. However, this should be verified first. Therefore, Table 16 first compares the total travel time and the range time of models 2 and 4.

Table 16: Fairness results of Model 2 and Model 4.

	Model 2: Node-flow formulated		**Model 4: Arc-flow formulated**	
n	Total travel time	Range time	Total travel time	Range time
20	115,700	1,950	111,448	3,517
30	145,133	1,990	127,145	2,752
50	266,165	3,561	160,752	3,780
Avg.	175,666	2,500	133,115	3,349

Model 4 consistently achieves lower total travel times across all problem sizes compared to Model 2, indicating greater efficiency in minimising the total distance travelled. However, Model 2 has lower values for range time, particularly for *n* = 20 and *n* = 30, which suggests more balanced workloads among the salesmen. In contrast, for *n* = 50, Model 4's range time is slightly higher, indicating greater variability in tour lengths.

Nevertheless, minor differences exist, to illustrate this point, the instance att532_3_21 from the sections before will be considered. It can be assumed that the routes of Model 4 are divided up more fairly in comparison to Model 2 due to the lower MinMax target value. As shown in Table 17, the individual routes of Models 2 and 4 hardly differ in length. However, the customers visited are arranged in a manner that results in notable discrepancies in the MinMax objective function.

Table 17: Route allocation Model 2 and Model 4 using instance att532_3_21.

	Model 2: Node-flow formulated	**Model 4: Arc-flow formulated**
	Routes	Routes
Route 1	0 6 10 12 5 3 2 4 21	0 1 5 6 8 9 7 10 11 14 16 19 17 13 12 21
Route 2	0 8 11 14 19 17 16 9 13 7 1 21	0 2 3 18 15 4 21
Route 3	0 18 20 15 21	0 20 21
MinMax Obj.	25,362	24,404
Total Travel Time	75,529	72,701
Range Time	**335**	**346**

A comparison of the two models reveals that the minimal maximum tour duration in Model 4 is approximately 958 units smaller than that of Model 2. In contrast, the total travel time for Model 4 has decreased 2,828 units in comparison to Model 2, which indicates that the travel times for Model 4 are less evenly distributed across the individual routes. Therefore, it can be assumed that the smaller the MinSum or the total travel time is, the more unbalanced the routes become and vice versa. This is also confirmed by the value of the range time. The discrepancy between the shortest and longest tours in Model 4 is 3% higher than that observed in Model 2. It can therefore be concluded that Model 4 performs better in general but has more difficulties in allocating the routes in terms of balancing workload and improving fairness.

5 Conclusion and Outlook

In this thesis the multiple Travelling Salesman Problem (mTSP) is addressed, which is a generalization of the classic TSP that integrates more than one salesman into the problem, making it not only more realistic but also more complex. As mentioned at the theoretical chapter, the mTSP is extensively studied in the literature, with various formulations proposed to solve it. Among these, the node-flow formulation according to Sarin et al. (2014) and the arc-flow formulation by Maffioli and Sciomachen (1997) are defined in more detail, as they play a fundamental role in the computational study presented in this thesis. The node-flow approach focuses on minimizing the total travel time, whereas the objective of the arc-flow formulation, which is proposed in Bianchessi et al. (2023), aims to reduce the maximum tour duration. Based on the objective function it became clear that the MinSum model is much simpler than the MinMax model, as it requires significantly less time to solve the individual instances, the optimality gaps were equal or close to 0% and the total travel time is also significantly lower. Whereas the MinMax objective function requires more effort and is therefore less easy to solve exactly and optimally, as evidenced by the optimality gaps of over 50 % and the requirement of the maximum CPU time for calculating the individual instances.

This paper analyses and compares four models: two MinSum models based on the formulation of Sarin et al. (2014) and Maffioli and Sciomachen (1997), and two MinMax models also based on both formulations. It is well-known from the literature that the MinSum objective does not ensure fairness, as it tends to produce one very long route and several shorter ones when multiple vehicles are used, which is supported by the results of the computational study. Therefore, a new "Upper Bound Travel Duration" -constraint is implemented and designed to limit the maximum route length to enhance fairness and promote a more balanced distribution of routes. To solve the models, the BnB algorithm is used, which offers significant advantages, especially for the MinSum objectives. This algorithm enables a detailed comparison of values and adapts well to the new constraint. Nevertheless, the study shows that larger instances require considerably more time to get a solution and the introduced constraint presents an additional challenge to solve all instances. This applies in particular to the MinSum formulation according to Maffioli and Sciomachen (1997), in which some instances are not solved.

The study also yielded to other important observations. Firstly, it is observed that a low MinSum objective implicates a high imbalance between the individual route lengths and times. Because the objective does not balance route lengths, as it focusses only on finding the most efficient solution compared to the MinMax objective. In contrast, a low maximum travel duration and a low range time show that the model is overall less efficient regarding the total travel time but the results regarding fairness are more optimal, as the routes are more balanced, and the travel time of each route is closer to one another. Especially the arc-flow MinMax model establishes more performance efficiency compared to the node-flow model by offering lower MinMax solutions, requiring less time to solve, exhibiting lower optimality gaps, and needing fewer BnB nodes. However, the node-flow MinMax balances the maximum tour length better, the range time as well as the MinMax objective is lower in this case. Regarding the MinSum objective, for both formulations the values are identical, therefore both show poor examples regarding the fairness aspect. Nevertheless, the arc-flow formulation performs more efficient in term of solution time for bigger instances and needs overall less BnB nodes than the node-flow formulation.

While the additional constraint increases computational effort, it generally improves the balancing of route lengths and the maximum travel time more. The MinSum model by Sarin et al. (2014) with the additional constraint provides superior solutions for the objective values and solves larger instances faster. The model according to Maffioli and Sciomachen (1997) performs more efficient for small instances in terms of less BnB nodes, less time, and lower optimality gaps. It is also notable that the constraints affect the balance of the routes in different ways, depending on whether the constraint is tighter or looser. In contrast to the classical MinSum formulation, the introduced UB-TD constraint generally leads to a more optimal distribution of the maximum travel duration, which is significantly reduced on average. Furthermore, the range between the shortest and longest travel times is reduced, resulting in a more balanced distribution of travel durations. However, the total travel time increases at the same time, as the model now attempts to balance the tour lengths on the one hand and fulfil the new constraint on the other hand. Consequently, the efficiency of the model decreases, particularly for tighter constraints (e.g., 5% or 10%), which are more challenging than looser constraints but more effective in achieving a fairer distribution in workload.

Due to its practical applicability in many fields, mTSP is an increasingly important topic which offers several interesting possibilities for future research. Firstly, this study focuses on mTSP formulations with MinSum and MinMax objectives. With respect to fairness, there are already many methods to consider fairness within mTSP. However, it could be interesting to investigate also the range objective, where the discrepancy between the shortest and the longest travel time is minimized. Secondly, as this thesis has only investigated 48 instances, the investigation and comparison of further instance sizes could be interesting, especially in terms of additional vehicles or additional cities to determine further effects on the models. And thirdly, given the challenges that the Maffioli and Sciomachen (1997) model has, in coping with the additional constraint and solving all instances, it would be interesting to investigate whether increasing CPU time or using metaheuristics could improve the results more. But also studying alternative mTSP variants, such as mTSP with time windows or collaborative mTSP with trucks and drones, could provide valuable insights into improving fairness.

List of Literature

Alighanbari, M., Kuwata, Y., & How, J. P. (2003). Coordination and control of multiple UAVs with timing constraints and loitering. *Proceedings of the 2003 American Control Conference, 2003.*, *6*, 5311–5316. https://doi.org/10.1109/ACC.2003.1242572

Al-Omeer, M. A., & Ahmed, Z. H. (2019). Comparative study of crossover operators for the MTSP. *2019 International Conference on Computer and Information Sciences (ICCIS)*, 1–6. https://doi.org/10.1109/ICCISci.2019.8716483

Bektas, T. (2006). The multiple traveling salesman problem: An overview of formulations and solution procedures. *Omega*, *34*(3), 209–219. https://doi.org/10.1016/j.omega.2004.10.004

Bérczi, K., Mnich, M., & Vincze, R. (2023). Approximations for many-visits multiple traveling salesman problems. *Omega*, *116*, 102816. https://doi.org/10.1016/j.omega.2022.102816

Bhadoriya, A. S., Deka, D., & Sundar, K. (2024). Equitable Routing—Rethinking the Multiple Traveling Salesman Problem (arXiv:2404.08157). arXiv. http://arxiv.org/abs/2404.08157

Bianchessi, N., Tilk, C., & Irnich, S. (2023, Mai 16). On solving the minmax multiple traveling salesman problem by column generation [Presentation]. *Talk at the Column Generation Workshop 2023*, Montréal, Canada.

Bolaños, R. I., Echeverry, M. G., & Escobar, J. W. (2015). A multiobjective non-dominated sorting genetic algorithm (NSGA-II) for the Multiple Traveling Salesman Problem. *Decision Science Letters*, *4*(4), 559–568. https://doi.org/10.5267/j.dsl.2015.5.003

Burkard, R. E., Deineko, V. G., Van Dal, R., Van Der Veen, J. A. A., & Woeginger, G. J. (1998). Well-Solvable Special Cases of the Traveling Salesman Problem: A Survey. *SIAM Review*, *40*(3), 496–546. https://doi.org/10.1137/S0036144596297514

Cambridge Dictionary. (n. d.). *Definition of fairness*. Cambridge Dictionary. Accessed 30 June 2024, from https://dictionary.cambridge.org/dictionary/english/fairness

Carter, A. E., & Ragsdale, C. T. (2006). A new approach to solving the multiple traveling salesperson problem using genetic algorithms. *European Journal of Operational Research*, *175*(1), 246–257. https://doi.org/10.1016/j.ejor.2005.04.027

Cheikhrouhou, O., & Khoufi, I. (2021). A comprehensive survey on the Multiple Traveling Salesman Problem: Applications, approaches and taxonomy. *Computer Science Review*, *40*, 100369. https://doi.org/10.1016/j.cosrev.2021.100369

Chen, Y., Jia, Z., Ai, X., Yang, D., & Yu, J. (2017). A modified two-part wolf pack search algorithm for the multiple traveling salesmen problem. *Applied Soft Computing*, *61*, 714–725. https://doi.org/10.1016/j.asoc.2017.08.041

Cornejo-Acosta, J. A., García-Díaz, J., Pérez-Sansalvador, J. C., & Segura, C. (2023). Compact Integer Programs for Depot-Free Multiple Traveling Salesperson Problems. *Mathematics*, *11*(13), 3014. https://doi.org/10.3390/math11133014

França, P. M., Gendreau, M., Laporte, G., & Müller, F. M. (1995). The m -Traveling Salesman Problem with Minmax Objective. *Transportation Science*, *29*(3), 267–275. https://doi.org/10.1287/trsc.29.3.267

Gendreau, M., & Potvin, J.-Y. (Eds.). (2010). *Handbook of Metaheuristics* (Vol. 146). Springer US. https://doi.org/10.1007/978-1-4419-1665-5

He, P., & Hao, J.-K. (2023). Memetic search for the minmax multiple traveling salesman problem with single and multiple depots. *European Journal of Operational Research*, *307*(3), 1055–1070. https://doi.org/10.1016/j.ejor.2022.11.010

Jiang, C., Wan, Z., & Peng, Z. (2020). A new efficient hybrid algorithm for large scale multiple traveling salesman problems. *Expert Systems with Applications*, *139*, 112867. https://doi.org/10.1016/j.eswa.2019.112867

Kazachkov, A. M., Bodic, P. L., & Sankaranarayanan, S. (2023). An Abstract Model for Branch and Cut. *Mathematical Programming*. https://doi.org/10.1007/s10107-023-01991-z

Kitjacharoenchai, P., Ventresca, M., Moshref-Javadi, M., Lee, S., Tanchoco, J. M. A., & Brunese, P. A. (2019). Multiple traveling salesman problem with drones: Mathematical model and heuristic approach. *Computers & Industrial Engineering*, *129*, 14–30. https://doi.org/10.1016/j.cie.2019.01.020

Kivelevitch, E., Cohen, K., & Kumar, M. (2013). A Market-based Solution to the Multiple Traveling Salesmen Problem. *Journal of Intelligent & Robotic Systems*, *72*(1), 21–40. https://doi.org/10.1007/s10846-012-9805-3

Kloster, K., Moeini, M., Vigo, D., & Wendt, O. (2023). The multiple traveling salesman problem in presence of drone- and robot-supported packet stations. *European Journal of Operational Research*, *305*(2), 630–643. https://doi.org/10.1016/j.ejor.2022.06.004

Laporte, G., & Nobert, Y. (1980). A Cutting Planes Algorithm for the m-Salesmen Problem. *Journal of the Operational Research Society*, *31*(11), 1017–1023. https://doi.org/10.1057/jors.1980.188

Lin, S., & Kernighan, B. W. (1973). An Effective Heuristic Algorithm for the Traveling-Salesman Problem. *Operations Research*, *21*(2), 498–516. https://doi.org/10.1287/opre.21.2.498

Liu, D., Hou, M., & Qu, H. (2013). A simple model for the multiple traveling salesmen problem with single depot and multiple sink. *COMPEL - The International Journal for Computation and Mathematics in Electrical and Electronic Engineering*, *32*(2), 556–574. https://doi.org/10.1108/03321641311296945

Lu, L.-C., & Yue, T.-W. (2019). Mission-oriented ant-team ACO for min–max MTSP. *Applied Soft Computing*, *76*, 436–444. https://doi.org/10.1016/j.asoc.2018.11.048

Mahmoudinazlou, S., & Kwon, C. (2023). *A Hybrid Genetic Algorithm for the min-max Multiple Traveling Salesman Problem* (arXiv:2307.07120). arXiv. http://arxiv.org/abs/2307.07120

Matl, P., Hartl, R. F., & Vidal, T. (2018). Workload Equity in Vehicle Routing Problems: A Survey and Analysis. *Transportation Science*, *52*(2), 239–260. https://doi.org/10.1287/trsc.2017.0744

Murray, C. C., & Raj, R. (2020). The multiple flying sidekicks traveling salesman problem: Parcel delivery with multiple drones. *Transportation Research Part C: Emerging Technologies*, *110*, 368–398. https://doi.org/10.1016/j.trc.2019.11.003

Necula, R., Breaban, M., & Raschip, M. (2015). Tackling the Bi-criteria Facet of Multiple Traveling Salesman Problem with Ant Colony Systems. *2015 IEEE 27th International Conference on Tools with Artificial Intelligence (ICTAI)*, 873–880. https://doi.org/10.1109/ICTAI.2015.127

Sánchez, A. L. (2022). On balancing fairness and efficiency in routing of cooperative vehicle fleets. *CEUR Workshop Proceedings*, *3173*.

Sarin, S. C., Sherali, H. D., Judd, J. D., & Tsai, P.-F. (2014). Multiple asymmetric traveling salesmen problem with and without precedence constraints: Performance comparison of alternative formulations. *Computers & Operations Research*, *51*, 64–89. https://doi.org/10.1016/j.cor.2014.05.014

Somhom, S., Modares, A., & Enkawa, T. (1999). Competition-based neural network for the multiple travelling salesmen problem with minmax objective. *Computers & Operations Research*, *26*(4), 395–407. https://doi.org/10.1016/S0305-0548(98)00069-0

Soriano, A., Gansterer, M., & Hartl, R. F. (2023). The multi-depot vehicle routing problem with profit fairness. *International Journal of Production Economics*, *255*, 108669. https://doi.org/10.1016/j.ijpe.2022.108669

Soylu, B. (2015). A general variable neighborhood search heuristic for multiple traveling salesmen problem. *Computers & Industrial Engineering*, *90*, 390–401. https://doi.org/10.1016/j.cie.2015.10.010

Svestka, J. A., & Huckfeldt, V. E. (1973). Computational Experience with an *M* -Salesman Traveling Salesman Algorithm. *Management Science*, *19*(7), 790–799. https://doi.org/10.1287/mnsc.19.7.790

Sze, S. N., & Tiong, W. K. (2007). A Comparison between Heuristic and Meta-Heuristic Methods for Solving the Multiple Traveling Salesman Problem. *International Journal of Mathematical and Computational Sciences*, *1*(1), 13–16.

Vali, M., & Salimifard, K. (2017). *A Constraint Programming Approach for Solving Multiple Traveling Salesman Problem*.

Wang, M., Ma, T., Li, G., Zhai, X., & Qiao, S. (2020). Ant Colony Optimization With an Improved Pheromone Model for Solving MTSP With Capacity and Time Window Constraint. *IEEE Access*, *8*, 106872–106879. https://doi.org/10.1109/ACCESS.2020.3000501

Xinying Chen, V., & Hooker, J. N. (2023). A guide to formulating fairness in an optimization model. *Annals of Operations Research*, *326*(1), 581–619. https://doi.org/10.1007/s10479-023-05264-y

Appendix

Attachment 1 – Full results of the MinSum Models 1 and 3

	Model 1: Node-flow formulated						**Model 3: Arc-flow formulated**					
Instance	Obj.	Opt. Gap	Time (s)	#BnB	Max. tour	Range time	Obj.	Opt. Gap	Time (s)	#BnB	Max. tour	Range time
att532_3_21.txt	54415	0,00%	0	0	35335	28449	54415	0,01%	0	0	35335	28449
att532_3_31.txt	65378	0,01%	2	8933	43693	36807	65378	0,01%	5	12798	43693	36807
att532_3_51.txt	112188	0,01%	975	1235999	90503	83617	112188	0,01%	157	93265	90503	83617
att532_5_21.txt	77155	0,00%	0	0	34535	27649	77155	0,00%	0	52	34535	27649
att532_5_31.txt	87797	0,01%	13	68512	42302	35416	87797	0,01%	3	8537	42302	35416
att532_5_51.txt	134160	0,13%	1800	2084824	88665	81779	134160	0,01%	292	148514	88665	81779
kroa200_3_21.txt	145882	0,00%	0	0	124804	114580	145882	0,00%	0	0	124804	114580
kroa200_3_31.txt	154715	0,00%	1	0	134617	125373	154715	0,00%	1	29	134617	125373
kroa200_3_51.txt	173427	0,01%	29	44582	156601	149019	173427	0,01%	88	53452	156601	149019
kroa200_5_21.txt	170990	0,00%	0	0	117484	107260	170990	0,00%	0	0	117484	107260
kroa200_5_31.txt	175003	0,00%	0	0	131509	122265	175003	0,00%	0	0	131509	122265
kroa200_5_51.txt	191972	0,01%	27	49825	152758	145176	191972	0,01%	79	47743	152758	145176
lin318_3_21.txt	39852	0,00%	0	0	36228	35606	39852	0,00%	0	17	36228	35606
lin318_3_31.txt	44003	0,00%	0	117	40379	39757	44003	0,00%	0	31	40379	39757
lin318_3_51.txt	75776	0,01%	1723	2457901	72152	71530	75776	1,89%	1800	1044753	72152	71530
lin318_5_21.txt	49224	0,00%	0	318	35792	35170	49224	0,00%	0	547	35792	35170
lin318_5_31.txt	53375	0,01%	3	8485	39943	39321	53375	0,00%	0	23	39943	39321
lin318_5_51.txt	84890	3,02%	1802	1925758	71458	70836	84890	0,89%	1800	929867	71458	70836
mtsp51_3_21.txt	2874	0,00%	0	0	2384	2142	2874	0,00%	0	0	2384	2142
mtsp51_3_31.txt	3288	0,00%	0	0	2898	2756	3288	0,00%	1	184	2898	2756
mtsp51_3_51.txt	4489	0,00%	13	15917	4185	4043	4489	0,00%	101	46146	4185	4043
mtsp51_5_21.txt	3311	0,00%	0	0	2129	1895	3311	0,00%	0	0	2129	1895
mtsp51_5_31.txt	3615	0,00%	1	37	2829	2687	3615	0,00%	1	111	2829	2687
mtsp51_5_51.txt	4746	0,00%	0	0	4072	3950	4746	0,00%	9	1750	4072	3950
mtsp100_3_21.txt	122505	0,00%	0	0	100628	93322	122505	0,00%	0	0	100628	93322
mtsp100_3_31.txt	146757	0,00%	0	16	132441	125431	146757	0,00%	0	0	132441	125431
mtsp100_3_51.txt	170643	0,26%	1800	2171438	161469	159601	170643	0,01%	186	95992	161469	159601
mtsp100_5_21.txt	146145	0,00%	0	0	100623	93317	146145	0,00%	0	0	100623	93317
mtsp100_5_31.txt	161840	0,00%	0	11	124813	117803	161840	0,00%	1	0	124813	117803
mtsp100_5_51.txt	182506	0,01%	417	655665	143311	141443	182506	0,01%	1202	804729	143311	141443
mtsp150_3_21.txt	151979	0,00%	0	0	127777	116315	151979	0,00%	0	0	127777	116315
mtsp150_3_31.txt	189700	0,01%	7	28310	169750	162540	189700	0,01%	8	19389	169750	162540
mtsp150_3_51.txt	217101	0,01%	8	5756	205291	200691	217101	0,01%	4	1576	205291	200691
mtsp150_5_21.txt	186549	0,00%	0	0	73072	61610	186549	0,00%	0	0	73072	61610
mtsp150_5_31.txt	209558	0,01%	12	59096	168304	161094	209558	0,01%	5	9577	168304	161094
mtsp150_5_51.txt	235110	0,01%	7	3626	200718	196118	235110	0,01%	96	98764	200718	196118
pcb1173_3_21.txt	13711	0,00%	0	0	11347	10545	13711	0,00%	0	16	11347	10545
pcb1173_3_31.txt	18835	0,01%	1	1856	16471	15669	18835	0,00%	1	494	16471	15669
pcb1173_3_51.txt	32641	0,01%	203	399481	30277	29475	32641	0,01%	76	72493	30277	29475
pcb1173_5_21.txt	18789	0,00%	0	0	11001	10199	18789	0,00%	0	0	11001	10199
pcb1173_5_31.txt	23913	0,01%	1	4837	16125	15323	23913	0,01%	1	431	16125	15323
pcb1173_5_51.txt	37719	0,01%	64	130312	29931	29129	37719	0,01%	190	140785	29931	29129
rat783_3_21.txt	5699	0,00%	242	2269402	5187	5033	5699	0,00%	277	1561711	5187	5033
rat783_3_31.txt	5738	14,67%	1809	5337300	5322	5168	5738	13,12%	1803	3379094	5322	5168
rat783_3_51.txt	7261	0,00%	175	329173	6845	6691	7261	0,00%	565	538414	6845	6691
rat783_5_21.txt	7218	0,00%	93	925100	5182	5028	7218	0,00%	325	1629675	5182	5028
rat783_5_31.txt	6660	10,14%	1802	4350127	5312	5158	6646	11,86%	1810	3947855	5298	5144
rat783_5_51.txt	7991	0,00%	95	176316	6545	6391	7991	0,00%	656	662483	6545	6391
Avg.	87898	0,59%	273	515605	69271	65420	87897	0,58%	241	319819	69270	65420

Attachment 2 – Full results of the MinSum Models 1 and 3 with 5% constraint

	Model 1 – 5%: Node-flow formulated						**Model 3 – 5%: Arc-flow formulated**					
Instance	Obj.	Opt. Gap	Time (s)	#BnB	Max. tour	Range time	Obj.	Opt. Gap	Time (s)	#BnB	Max. tour	Range time
att532_3_21.txt	56108	0,01%	5	36184	26405	19519	63055	0,01%	643	2207236	25427	10628
att532_3_31.txt	68363	3,21%	1800	3305492	32362	25476	82977	13,57%	1804	2248359	29572	4028
att532_3_51.txt	115404	3,52%	1805	1985398	73080	66194	126478	9,56%	1800	484913	60452	53566
att532_5_21.txt	80960	0,01%	388	2329282	25427	18541	80960	0,01%	61	250643	25427	18541
att532_5_31.txt	101465	13,62%	1820	5281581	29572	22686	101465	9,93%	1805	2239804	29572	22686
att532_5_51.txt	143119	6,99%	1805	2167378	61470	54584	-	-	-	-	-	-
kroa200_3_21.txt	166230	0,01%	29	208592	60444	8236	166230	0,01%	5	15923	60444	8236
kroa200_3_31.txt	174309	8,14%	1802	4668548	63344	9296	174309	0,01%	56	66659	63344	9296
kroa200_3_51.txt	175293	0,01%	221	275163	147760	140178	193695	5,50%	1800	352495	70157	13369
kroa200_5_21.txt	209786	7,09%	1802	8793047	55703	45479	209786	0,01%	192	434496	55703	45479
kroa200_5_31.txt	219112	16,58%	1813	5726098	56911	47667	219112	9,51%	1800	1758701	56911	47667
kroa200_5_51.txt	194225	0,01%	952	1145869	100425	92843	-	-	-	-	-	-
lin318_3_21.txt	70747	34,89%	1802	8530438	26869	7307	70747	30,49%	1802	4761700	26869	7307
lin318_3_31.txt	53785	13,66%	1805	4716469	28362	27740	74653	31,26%	1800	1926394	26869	3401
lin318_3_51.txt	79860	8,23%	1803	1869057	69472	68850	106537	27,85%	1808	102978	39269	9535
lin318_5_21.txt	74578	25,38%	1802	8110265	25744	25122	74578	19,64%	1800	4087604	25744	25122
lin318_5_31.txt	78275	29,66%	1804	4925598	26869	26247	-	-	-	-	-	-
lin318_5_51.txt	91305	11,62%	1802	2300915	51000	50378	-	-	-	-	-	-
mtsp51_3_21.txt	3171	0,00%	289	1698979	1197	328	3289	0,00%	52	151142	1101	13
mtsp51_3_31.txt	3399	0,00%	24	87565	1579	1331	3628	0,00%	685	863032	1292	231
mtsp51_3_51.txt	4520	0,00%	82	83392	3048	2906	4666	1,35%	1800	488793	2042	1117
mtsp51_5_21.txt	3942	8,28%	1802	8688376	940	453	3942	0,00%	64	151952	940	453
mtsp51_5_31.txt	3763	0,00%	73	246104	1374	1232	4452	8,77%	1817	1455871	984	235
mtsp51_5_51.txt	4796	0,00%	36	42471	2272	2150	5055	4,79%	1809	129649	1539	1397
mtsp100_3_21.txt	165163	20,22%	1811	9468965	67486	31112	165163	0,01%	49	144193	67486	31112
mtsp100_3_31.txt	169023	10,12%	1802	5091391	91750	84444	191743	5,98%	1800	1794695	72693	23265
mtsp100_3_51.txt	170643	1,05%	1800	1310728	161469	159601	-	-	-	-	-	-
mtsp100_5_21.txt	224092	30,55%	1802	8803030	60801	53495	224092	11,92%	1800	4586693	60801	53495
mtsp100_5_31.txt	244770	31,87%	1806	6144948	68211	61201	249090	19,30%	1800	1787349	66615	59605
mtsp100_5_51.txt	194418	9,62%	1808	2255001	98438	96570	232000	15,68%	1800	229600	74115	72247
mtsp150_3_21.txt	176131	0,01%	300	2013898	67765	20067	179030	0,01%	35	121132	66519	18821
mtsp150_3_31.txt	191472	0,01%	14	61069	95976	88766	214083	0,01%	593	860344	78188	13475
mtsp150_3_51.txt	220483	0,01%	32	30437	183763	179163	238662	3,61%	1800	411037	90178	23738
mtsp150_5_21.txt	221904	5,68%	1802	8384180	48066	7721	221904	0,01%	76	225051	48066	7721
mtsp150_5_31.txt	237872	8,05%	1806	4401486	71700	64490	245103	0,01%	805	973720	61052	53842
mtsp150_5_51.txt	236693	0,01%	31	46526	117751	113151	-	-	-	-	-	-
pcb1173_3_21.txt	16385	0,01%	619	3055438	8158	7356	16385	0,01%	3	6659	8158	7356
pcb1173_3_31.txt	20978	5,99%	1800	4078715	15563	14761	24125	10,73%	1800	2001853	12925	12123
pcb1173_3_51.txt	32641	0,01%	202	271258	30277	29475	42248	22,50%	1800	577976	24852	24050
pcb1173_5_21.txt	20267	0,01%	18	137157	8156	7354	20267	0,00%	1	2429	8156	7354
pcb1173_5_31.txt	28007	12,08%	1803	4714903	12923	12121	28007	7,34%	1800	1911970	12923	12121
pcb1173_5_51.txt	46210	20,24%	1809	2537514	24046	23244	46051	18,84%	1800	434339	23887	23085
rat783_3_21.txt	5699	0,00%	851	3449325	5187	5033	5699	0,00%	71	303622	5187	5033
rat783_3_31.txt	5753	18,28%	1800	4213806	5337	5183	5738	0,00%	1692	3136940	5322	5168
rat783_3_51.txt	7261	0,00%	169	227334	6845	6691	10462	28,73%	1800	651499	5568	5414
rat783_5_21.txt	7218	0,00%	182	1034059	5182	5028	7218	0,00%	7	34304	5182	5028
rat783_5_31.txt	6646	14,73%	1800	4033790	5298	5144	6646	0,00%	612	1166008	5298	5144
rat783_5_51.txt	11105	29,12%	1810	2751292	5575	5421	10280	20,99%	1800	622840	5810	5656
Avg.	100778	8,51%	1147	3244552	46612	38569	98709	8.05%	1123	1099109	33634	18028

Attachment 3 – Full results of the MinSum Models 1 and 3 with 10% constraint

	Model 1 – 10%: Node-flow formulated						**Model 3 – 10%: Arc-flow formulated**					
Instance	Obj.	Opt. Gap	Time (s)	#BnB	Max. tour	Range time	Obj.	Opt. Gap	Time (s)	#BnB	Max. tour	Range time
att532_3_21.txt	56108	0,01%	4	27849	26405	19519	56108	0,00%	0	405	26405	19519
att532_3_31.txt	68280	2,43%	1800	3633955	34989	28103	68363	0,01%	33	47140	32362	25476
att532_3_51.txt	115549	3,66%	1800	1474211	77651	70765	116260	0,57%	1800	615744	63015	56129
att532_5_21.txt	78955	0,01%	4	28544	26405	19519	78955	0,01%	1	1354	26405	19519
att532_5_31.txt	92941	5,35%	1806	4059107	31110	24224	92941	1,24%	1800	2166821	31110	24224
att532_5_51.txt	139107	4,42%	1806	2634298	63015	56129	139107	1,27%	1800	471195	63015	56129
kroa200_3_21.txt	166230	0,01%	68	551956	60444	8236	166230	0,01%	5	10009	60444	8236
kroa200_3_31.txt	165457	0,01%	432	1400861	78586	69342	174309	0,01%	115	151259	63344	9296
kroa200_3_51.txt	175293	0,01%	148	230920	147760	140178	193695	6,43%	1808	151283	70157	13369
kroa200_5_21.txt	209786	5,46%	1800	8804718	55703	45479	209786	0,01%	464	1004022	55703	45479
kroa200_5_31.txt	201142	8,26%	1802	4926302	62544	53300	216938	11,00%	1809	1142113	60354	51110
kroa200_5_51.txt	194225	0,01%	238	305851	100425	92843	-	-	-	-	-	-
lin318_3_21.txt	52771	9,89%	1800	7483018	28299	27677	52771	0,01%	489	1332573	28299	27677
lin318_3_31.txt	53785	14,21%	1802	4965299	28362	27740	56922	9,37%	1800	1890860	28299	27677
lin318_3_51.txt	75776	1,09%	1800	1633292	72152	71530	-	-	-	-	-	-
lin318_5_21.txt	72221	24,64%	1807	8295361	27556	26934	72221	19,76%	1802	4603775	27556	26934
lin318_5_31.txt	63544	16,24%	1821	4329400	28299	27677	78275	25,17%	1800	2153853	26869	26247
lin318_5_51.txt	91305	10,74%	1802	2195102	51000	50378	-	-	-	-	-	-
mtsp51_3_21.txt	3171	0,00%	175	1126294	1197	328	3171	0,00%	11	33940	1197	328
mtsp51_3_31.txt	3372	0,00%	8	28631	1673	1425	3476	0,00%	26	32231	1378	627
mtsp51_3_51.txt	4517	0,00%	70	75464	3393	3251	4625	0,46%	1800	630060	2122	1118
mtsp51_5_21.txt	3942	7,80%	1801	8750902	940	453	3942	0,00%	165	405869	940	453
mtsp51_5_31.txt	3716	0,00%	18	71603	1447	1305	4298	6,85%	1800	1638229	1025	628
mtsp51_5_51.txt	4796	0,00%	55	51337	2272	2150	4949	2,49%	1802	136186	1637	1515
mtsp100_3_21.txt	151535	12,30%	1802	7904093	75661	61090	158163	0,01%	95	301701	72798	58227
mtsp100_3_31.txt	168126	9,82%	1804	4819456	98057	90751	188408	5,64%	1800	1673037	75635	26207
mtsp100_3_51.txt	170643	1,66%	1800	1333990	161469	159601	-	-	-	-	-	-
mtsp100_5_21.txt	208112	25,15%	1803	9193300	64954	57648	208112	5,89%	1800	4218050	64954	57648
mtsp100_5_31.txt	222085	25,08%	1811	5978235	69967	62957	222085	10,30%	1800	1589058	69967	62957
mtsp100_5_51.txt	194418	7,92%	1802	1675663	98438	96570	230890	17,80%	1801	225500	77415	75547
mtsp150_3_21.txt	163819	0,01%	4	36748	70727	47959	163819	0,01%	1	808	70727	47959
mtsp150_3_31.txt	191472	0,01%	9	38232	95976	88766	209394	0,01%	205	274619	79434	28747
mtsp150_3_51.txt	217101	0,01%	5	1926	205291	200691	232009	0,01%	1000	382665	93798	46644
mtsp150_5_21.txt	221904	0,01%	1524	8708974	48066	7721	221904	0,01%	419	1138196	48066	7721
mtsp150_5_31.txt	235190	6,86%	1806	5164284	74662	67452	241442	0,01%	667	975573	63966	56756
mtsp150_5_51.txt	236693	0,01%	23	35208	117751	113151	260963	4,58%	1800	302502	69186	64586
pcb1173_3_21.txt	16385	0,01%	937	4327496	8158	7356	16385	0,00%	5	11291	8158	7356
pcb1173_3_31.txt	20031	0,01%	184	681542	16127	15325	22283	0,01%	343	575705	13697	12895
pcb1173_3_51.txt	32641	0,01%	994	933035	30277	29475	40381	18,21%	1800	640440	26046	25244
pcb1173_5_21.txt	20267	0,01%	9	64243	8156	7354	20267	0,00%	1	2571	8156	7354
pcb1173_5_31.txt	26165	4,70%	1800	4059610	13695	12893	26165	0,01%	250	476677	13695	12893
pcb1173_5_51.txt	42900	13,61%	1805	2254702	26171	25369	42900	11,05%	1800	677687	26171	25369
rat783_3_21.txt	5699	0,00%	1071	4745199	5187	5033	5699	0,00%	41	158685	5187	5033
rat783_3_31.txt	5738	16,98%	1803	3961851	5322	5168	5738	0,00%	1092	2150764	5322	5168
rat783_3_51.txt	7261	0,00%	100	141992	6845	6691	9516	22,26%	1800	687560	5829	5675
rat783_5_21.txt	7218	0,00%	394	1946799	5182	5028	7218	0,00%	10	48364	5182	5028
rat783_5_31.txt	6646	12,54%	1800	3815331	5298	5144	6646	0,00%	489	1002316	5298	5144
rat783_5_51.txt	10192	22,78%	1808	2298712	5842	5688	9197	11,54%	1800	849764	6095	5941
Avg.	97463	5,70%	1074	2941769	48519	42778	94356	3.93%	892	830607	34813	23191

Attachment 4 – Full results of the MinSum Models 1 and 3 with 15% constraint

	Model 1 – 15%: Node-flow formulated						**Model 3 – 15%: Arc-flow formulated**					
Instance	Obj.	Opt. Gap	Time (s)	#BnB	Max. tour	Range time	Obj.	Opt. Gap	Time (s)	#BnB	Max. tour	Range time
att532_3_21.txt	56108	0,01%	6	52265	26405	19519	56108	0,00%	1	800	26405	19519
att532_3_31.txt	68280	2,76%	1800	3244108	34989	28103	68363	0,01%	13	19277	32362	25476
att532_3_51.txt	114895	3,12%	1804	1823129	81110	74224	116260	0,76%	1800	582313	63015	56129
att532_5_21.txt	78955	0,01%	4	27330	26405	19519	78955	0,01%	1	1644	26405	19519
att532_5_31.txt	91210	3,51%	1800	3337540	32362	25476	91210	0,01%	62	83567	32362	25476
att532_5_51.txt	139493	4,79%	1807	2189700	63015	56129	139107	1,43%	1800	424637	63015	56129
kroa200_3_21.txt	159616	0,01%	32	273245	72914	55376	166230	0,01%	15	47919	60444	8236
kroa200_3_31.txt	165457	1,27%	1800	3885533	78586	69342	174309	0,01%	153	188901	63344	9296
kroa200_3_51.txt	173427	0,01%	45	73683	156601	149019	193695	6,55%	1800	471341	70157	13369
kroa200_5_21.txt	186847	0,01%	14	108134	60433	50209	186847	0,01%	2	3713	60433	50209
kroa200_5_31.txt	193760	4,46%	1800	4453246	63333	54089	197988	0,01%	1035	1259224	62640	53396
kroa200_5_51.txt	194225	0,01%	683	673795	100425	92843	217714	7,61%	1800	304163	67130	59548
lin318_3_21.txt	49879	2,55%	1800	7173745	28362	27740	49879	0,01%	167	506437	28362	27740
lin318_3_31.txt	52312	12,32%	1802	5296298	31637	31015	53785	1,11%	1800	2392801	28362	27740
lin318_3_51.txt	75776	0,01%	359	492998	72152	71530	84705	10,65%	1800	331948	44237	43615
lin318_5_21.txt	56501	0,01%	227	1405282	28362	27740	56501	0,01%	28	80926	28362	27740
lin318_5_31.txt	60407	10,57%	1806	3923999	28362	27740	60407	0,01%	300	488302	28362	27740
lin318_5_51.txt	91467	12,62%	1811	3193238	45160	44538	-	-	-	-	-	-
mtsp51_3_21.txt	3049	0,00%	2	19656	1365	878	3171	0,00%	11	28472	1197	328
mtsp51_3_31.txt	3345	0,00%	2	5809	1752	1610	3476	0,00%	15	18034	1378	627
mtsp51_3_51.txt	4517	0,00%	86	70187	3393	3251	4564	0,00%	250	59004	2230	2088
mtsp51_5_21.txt	3755	0,00%	905	4637936	1049	815	3755	0,00%	31	85337	1049	815
mtsp51_5_31.txt	3716	0,00%	11	40954	1447	1305	4097	1,78%	1800	1929360	1073	931
mtsp51_5_51.txt	4796	0,00%	26	25712	2272	2150	5043	4,67%	1812	133150	1667	1525
mtsp100_3_21.txt	151535	12,26%	1802	8666034	75661	61090	151535	0,01%	8	19628	75661	61090
mtsp100_3_31.txt	159491	3,99%	1811	4024626	102757	95451	188408	6,76%	1800	1816141	75635	26207
mtsp100_3_51.txt	170643	1,60%	1800	1495936	161469	159601	-	-	-	-	-	-
mtsp100_5_21.txt	180467	13,01%	1802	8816170	67486	60180	180467	0,01%	43	131947	67486	60180
mtsp100_5_31.txt	207342	19,66%	1828	6070157	74179	67169	205508	2,20%	1800	1601986	72142	65132
mtsp100_5_51.txt	194418	9,90%	1804	2018794	98438	96570	213945	12,28%	1800	226186	79716	77848
mtsp150_3_21.txt	155793	0,00%	0	408	74007	62545	155793	0,00%	0	0	74007	62545
mtsp150_3_31.txt	191472	0,01%	13	42247	95976	88766	208223	0,01%	159	202584	82771	36753
mtsp150_3_51.txt	217101	0,01%	13	15506	205291	200691	227783	0,01%	76	23566	98173	66053
mtsp150_5_21.txt	207389	0,01%	122	957386	53833	42371	207389	0,01%	5	13049	53833	42371
mtsp150_5_31.txt	226429	0,61%	1800	4789165	78902	71692	238543	0,01%	151	235621	64956	57746
mtsp150_5_51.txt	236693	0,01%	37	52503	117751	113151	258357	5,44%	1806	167569	72223	67623
pcb1173_3_21.txt	16385	0,01%	888	4353815	8158	7356	16385	0,01%	4	14909	8158	7356
pcb1173_3_31.txt	18835	0,00%	1	3393	16471	15669	21621	0,01%	39	58327	14192	13390
pcb1173_3_51.txt	32641	0,01%	160	191114	30277	29475	37182	11,92%	1800	627862	27348	26546
pcb1173_5_21.txt	20267	0,01%	16	108291	8156	7354	20267	0,01%	2	4606	8156	7354
pcb1173_5_31.txt	25503	0,01%	1172	3054505	14190	13388	25503	0,01%	30	55396	14190	13388
pcb1173_5_51.txt	41064	9,73%	1801	1948889	27348	26546	41064	7,48%	1800	696825	27348	26546
rat783_3_21.txt	5699	0,00%	1086	4297504	5187	5033	5699	0,00%	60	239202	5187	5033
rat783_3_31.txt	5752	16,86%	1800	3505483	5336	5182	5738	0,84%	1800	3444811	5322	5168
rat783_3_51.txt	7261	0,00%	260	276480	6845	6691	8574	13,45%	1800	866600	6098	5944
rat783_5_21.txt	7218	0,00%	301	1584134	5182	5028	7218	0,00%	11	43511	5182	5028
rat783_5_31.txt	6646	12,84%	1800	3820283	5298	5144	6646	1,68%	1800	3212775	5298	5144
rat783_5_51.txt	9359	15,52%	1807	1870744	5966	5812	8417	2,10%	1800	982072	6370	6216
Avg.	94317	3,63%	924	2258148	49709	45544	90150	1.80%	673	546581	35816	25688

Attachment 5 – Full results of the MinSum Models 1 and 3 with 20% constraint

	Model 1 – 20%: Node-flow formulated						**Model 3 – 20%: Arc-flow formulated**					
Instance	Obj.	Opt. Gap	Time (s)	#BnB	Max. tour	Range time	Obj.	Opt. Gap	Time (s)	#BnB	Max. tour	Range time
att532_3_21.txt	56108	0,01%	6	39323	26405	19519	56108	0,01%	1	658	26405	19519
att532_3_31.txt	68280	2,82%	1800	3469007	34989	28103	68280	0,01%	28	44296	34989	28103
att532_3_51.txt	114532	2,69%	1801	1522050	84829	77943	116260	1,14%	1800	530220	63015	56129
att532_5_21.txt	78955	0,01%	4	32424	26405	19519	78955	0,01%	2	3009	26405	19519
att532_5_31.txt	91210	3,40%	1800	3068934	32362	25476	91210	0,01%	177	325781	32362	25476
att532_5_51.txt	139810	5,02%	1802	2237300	69003	62117	139107	1,11%	1800	509416	63015	56129
kroa200_3_21.txt	157570	0,01%	10	88533	74443	64219	166230	0,01%	13	37503	60444	8236
kroa200_3_31.txt	165457	0,01%	1182	2898867	78586	69342	174309	0,01%	277	374342	63344	9296
kroa200_3_51.txt	173427	0,01%	31	44951	156601	149019	193695	6,92%	1800	484957	70157	13369
kroa200_5_21.txt	186847	0,01%	17	125053	60433	50209	186847	0,01%	3	6504	60433	50209
kroa200_5_31.txt	193760	5,86%	1804	4823243	63333	54089	193760	0,01%	54	58545	63333	54089
kroa200_5_51.txt	194225	0,01%	455	498585	100425	92843	211160	4,75%	1800	321558	70118	62536
lin318_3_21.txt	47769	0,01%	168	1047526	30421	29799	47769	0,01%	44	133402	30421	29799
lin318_3_31.txt	50156	0,01%	936	4709390	33059	32437	53785	2,64%	1800	1920404	28362	27740
lin318_3_51.txt	75776	0,84%	1800	1536806	72152	71530	84705	9,61%	1800	330069	44237	43615
lin318_5_21.txt	54391	0,01%	52	370129	30421	29799	54391	0,01%	9	31312	30421	29799
lin318_5_31.txt	60407	9,86%	1806	4338894	28362	27740	60407	0,01%	369	585809	28362	27740
lin318_5_51.txt	88730	7,45%	1800	1956128	53624	53002	104543	19,44%	1805	97100	41747	41125
mtsp51_3_21.txt	3022	0,00%	2	7306	1423	1189	3171	0,00%	31	119794	1197	328
mtsp51_3_31.txt	3345	0,00%	5	13438	1752	1610	3429	0,00%	5	3440	1451	820
mtsp51_3_51.txt	4517	0,00%	63	56113	3393	3251	4540	0,00%	69	22985	2300	2158
mtsp51_5_21.txt	3593	0,00%	87	488346	1101	867	3648	0,00%	7	18625	1099	865
mtsp51_5_31.txt	3689	0,00%	10	34504	1569	1427	4019	0,00%	520	746184	1099	957
mtsp51_5_51.txt	4796	0,00%	35	29615	2272	2150	4893	0,00%	1441	360659	1789	1667
mtsp100_3_21.txt	151535	12,42%	1802	8333775	75661	61090	151535	0,01%	10	30028	75661	61090
mtsp100_3_31.txt	158355	3,28%	1800	3894231	108324	98845	185154	4,92%	1800	1905701	81257	40705
mtsp100_3_51.txt	170643	2,04%	1800	1274270	161469	159601	-	-	-	-	-	-
mtsp100_5_21.txt	180467	13,47%	1802	8053247	67486	60180	180467	0,01%	126	302860	67486	60180
mtsp100_5_31.txt	202161	17,59%	1806	5869453	75635	68625	202161	0,01%	941	1220239	75635	68625
mtsp100_5_51.txt	189128	4,59%	1800	1190959	117176	115308	213945	12,87%	1800	387679	79716	77848
mtsp150_3_21.txt	155793	0,00%	0	825	74007	62545	155793	0,00%	0	0	74007	62545
mtsp150_3_31.txt	191472	0,01%	29	108558	95976	88766	198053	0,01%	4	3654	88286	57953
mtsp150_3_51.txt	217101	0,01%	7	4533	205291	200691	227783	0,01%	186	54171	98173	66053
mtsp150_5_21.txt	203727	0,01%	44	371803	55212	43750	203727	0,01%	2	4023	55212	43750
mtsp150_5_31.txt	225714	0,01%	1056	3508731	79073	71863	238543	0,01%	742	1032271	64956	57746
mtsp150_5_51.txt	236693	0,01%	45	51018	117751	113151	253590	2,64%	1800	450011	75564	70964
pcb1173_3_21.txt	16385	0,01%	847	4064030	8158	7356	16385	0,01%	6	16729	8158	7356
pcb1173_3_31.txt	18835	0,01%	1	869	16471	15669	21309	0,01%	159	279766	14846	14044
pcb1173_3_51.txt	32641	0,01%	69	88330	30277	29475	35284	6,84%	1800	665295	28521	27719
pcb1173_5_21.txt	20267	0,01%	18	131273	8156	7354	20267	0,00%	1	3152	8156	7354
pcb1173_5_31.txt	25191	0,01%	209	666681	14844	14042	25191	0,01%	20	40892	14844	14042
pcb1173_5_51.txt	39166	3,92%	1800	1749143	28519	27717	39166	3,39%	1800	792892	28519	27717
rat783_3_21.txt	5699	0,00%	455	2033036	5187	5033	5699	0,00%	13	54014	5187	5033
rat783_3_31.txt	5738	16,65%	1800	3681847	5322	5168	5738	6,48%	1800	2635900	5322	5168
rat783_3_51.txt	7261	0,00%	153	206078	6845	6691	7804	4,20%	1800	991404	6370	6216
rat783_5_21.txt	7218	0,00%	377	1803081	5182	5028	7218	0,00%	11	41676	5182	5028
rat783_5_31.txt	6646	12,85%	1800	3613080	5298	5144	6646	3,55%	1800	3029637	5298	5144
rat783_5_51.txt	8417	5,31%	1800	1370811	6370	6216	7991	0,00%	80	29294	6545	6391
Avg.	93680	2,71%	808	1781378	50855	46802	89297	1.30%	561	457168	36497	26970

Attachment 6 – Full results of the MinMax Models 2 and 4

	Model 2: Node-flow formulated						**Model 4: Arc-flow formulated**					
Instance	Obj.	Opt. Gap	Time (s)	#BnB	Total TT	Range time	Obj.	Opt. Gap	Time (s)	#BnB	Total TT	Range time
att532_3_21.txt	25362	52,98%	1800	8461869	75529	335	24404	30,58%	1806	5036516	72701	346
att532_3_31.txt	32303	63,09%	1800	3454857	96152	693	29443	55,90%	1822	582430	88043	204
att532_3_51.txt	70908	95,40%	1808	3003602	206352	4978	58588	79,19%	1800	253315	174353	842
att532_5_21.txt	24404	46,80%	1800	6953465	111432	4223	24404	46,73%	1800	3045302	103507	9605
att532_5_31.txt	28654	54,69%	1800	3176693	138704	1678	28654	54,54%	1843	619633	132617	8129
att532_5_51.txt	58588	77,84%	1800	2254632	282034	4523	58588	80,17%	1800	172915	253162	17220
kroa200_3_21.txt	63759	76,65%	1800	5250703	188786	2318	59930	8,25%	1800	5163117	179688	56
kroa200_3_31.txt	71594	84,84%	1800	3614486	213825	796	62555	18,98%	1800	1838777	187261	338
kroa200_3_51.txt	140758	93,43%	1801	1593202	420762	1167	68640	80,81%	1800	131486	203307	1356
kroa200_5_21.txt	54190	65,22%	1800	4693105	259753	4183	54190	50,83%	1800	3503060	237176	15870
kroa200_5_31.txt	56875	73,82%	1800	2779166	278194	3293	54904	69,23%	1800	639182	262849	9208
kroa200_5_51.txt	108850	90,03%	1819	1031753	518274	11351	58444	80,26%	1800	221017	288973	1376
lin318_3_21.txt	25730	85,92%	1800	5428674	74310	1895	25730	67,86%	1800	3380798	74510	2067
lin318_3_31.txt	28343	87,22%	1800	4718842	84685	294	25730	75,96%	1895	555909	76157	985
lin318_3_51.txt	66427	94,55%	1800	1785097	196323	2273	39602	84,89%	1800	217318	117708	571
lin318_5_21.txt	25504	75,74%	1800	5794609	118653	4424	25504	61,71%	1800	2287310	119545	5370
lin318_5_31.txt	25730	75,96%	1800	5642198	125041	1709	25504	71,73%	1814	924044	114826	8796
lin318_5_51.txt	48813	87,33%	1801	2187206	238282	3098	35566	82,61%	1800	197708	175889	675
mtsp51_3_21.txt	1203	79,38%	1800	6814422	3523	56	1101	25,97%	1800	3436095	3289	13
mtsp51_3_31.txt	1558	89,60%	1800	5228761	4596	42	1254	61,09%	1800	907281	3646	68
mtsp51_3_51.txt	3131	94,83%	1800	3086694	9342	45	1949	87,28%	1800	23446	5596	160
mtsp51_5_21.txt	923	59,48%	1800	11359715	4545	37	916	47,77%	1800	2213506	4478	40
mtsp51_5_31.txt	1344	81,55%	1800	5537643	6640	56	949	60,47%	1800	704205	4678	25
mtsp51_5_51.txt	2171	90,23%	1811	4526789	10449	216	1493	83,39%	1800	17621	7088	185
mtsp100_3_21.txt	68922	81,55%	1800	5463298	204014	1838	66221	24,47%	1800	3131266	190602	6338
mtsp100_3_31.txt	90282	90,98%	1800	2558698	266394	3529	70328	49,29%	1800	1975278	210390	361
mtsp100_3_51.txt	163485	95,59%	1800	1238833	486244	2667	81150	86,87%	1810	98799	240208	1901
mtsp100_5_21.txt	59422	65,58%	1800	5544575	292324	2655	59422	39,14%	1800	1658938	291617	2631
mtsp100_5_31.txt	65002	80,44%	1800	3425604	320447	3623	64189	66,52%	1801	1019698	313377	3137
mtsp100_5_51.txt	99098	91,79%	1800	1316414	475513	8988	71268	81,48%	1800	166623	332788	15570
mtsp150_3_21.txt	65218	66,38%	1800	5567645	195517	96	64380	4,48%	1800	4445989	189131	2849
mtsp150_3_31.txt	106879	90,79%	1815	1190846	313314	6069	75055	14,13%	1800	1882390	224610	403
mtsp150_3_51.txt	194632	96,30%	1804	934916	577621	5502	85892	74,83%	1806	99050	255520	1884
mtsp150_5_21.txt	46820	51,17%	1800	4850707	226398	3662	46820	13,43%	1800	2612937	231065	1417
mtsp150_5_31.txt	68622	76,77%	1800	2568375	329902	7335	58491	58,66%	1800	994913	285613	3594
mtsp150_5_51.txt	119233	89,32%	1800	1589416	578165	8090	63117	79,82%	1800	138491	306603	4840
pcb1173_3_21.txt	7814	70,28%	1800	9573899	23296	109	7814	53,18%	1800	4033405	22303	813
pcb1173_3_31.txt	14939	89,54%	1800	3804285	44130	590	12455	75,09%	1820	785180	34121	2720
pcb1173_3_51.txt	33888	95,39%	1805	3067146	100559	653	23803	87,05%	1800	223592	70869	401
pcb1173_5_21.txt	7814	60,30%	1800	8446479	36778	1920	7814	55,02%	1800	2128127	29876	4712
pcb1173_5_31.txt	12450	75,08%	1800	5854798	60991	836	12450	75,24%	1910	481336	58969	2604
pcb1173_5_51.txt	23808	86,97%	1800	1860094	111587	2751	23794	87,05%	1809	77499	98355	12337
rat783_3_21.txt	5142	85,96%	1800	11041713	14285	888	5142	78,62%	1800	2641878	15016	319
rat783_3_31.txt	5146	93,04%	1800	6697774	15341	79	5142	86,58%	1800	1732312	15251	106
rat783_3_51.txt	7116	97,61%	1808	4058662	20792	523	5311	89,84%	1800	149288	15697	179
rat783_5_21.txt	5142	80,67%	1800	11769221	22056	2554	5142	77,58%	1817	833440	18660	3819
rat783_5_31.txt	5142	85,96%	1800	6031057	23771	1218	5142	84,08%	1868	495404	21914	3354
rat783_5_51.txt	5312	90,36%	1808	4768878	26347	152	5542	86,97%	1800	39237	25917	980
Avg.	46843	80,51%	1802	4616698	175666	2500	35290	62,41%	1809	1414939	133115	3349