Universidad Autónoma de Nuevo León

Facultad de Ingeniería Mecánica y Eléctrica

Subdirección de Estudios de Posgrado

The Dial-a-Tour Problem: A study of Single and Bi-Objective Optimization Methods

por

Oscar Alejandro Hernández López

como requisito parcial para obtener el grado de

doctorado en ingeniería de sistemas

Diciembre 2024

Universidad Autónoma de Nuevo León

Facultad de Ingeniería Mecánica y Eléctrica

Subdirección de Estudios de Posgrado

The Dial-a-Tour Problem: A study of Single and Bi-Objective Optimization Methods

por

Oscar Alejandro Hernández López

como requisito parcial para obtener el grado de

DOCTORADO EN INGENIERÍA DE SISTEMAS

Diciembre 2024

UNIVERSIDAD AUTÓNOMA DE NUEVO LEÓN
Facultad de Ingeniería Mecánica y Eléctrica
Posgrado

Los miembros del Comité de Evaluación de Tesis recomendamos que la Tesis "The Dial-a-Tour Problem: A Study of Single and Bi-Objective Optimization Methods", realizada por el estudiante Oscar Alejandro Hernández López, con número de matrícula 1985273, sea aceptada para su defensa como requisito parcial para obtener el grado de Doctor en Ingeniería de Sistemas.

El Comité de Evaluación de Tesis

Dr. Vincent André Lionel Boyer
Director

Dra. María Angélica Salazar Aguilar
Revisor

Dra. Iris Abril Martínez Salazar
Revisor

Dr. Norberto Alejandro Hernández Leandro
Revisor

Dra. Yasmín Ríos Solís
Revisor

Dr. Jobish Vallikavungal Devassia
Revisor

Vo.Bo.

Dr. Simón Martínez Martínez
Subdirector de Estudios de Posgrado

Institución 190001

Programa 557620

Acta Núm. 361

Ciudad Universitaria, a 24 de enero de 2025.

To my family: for all their love and support.

Contents

List of Figures

List of Tables

Acknowledgements

To begin with, I express my gratitude to my parents, beloved wife, and family. You are my constant source of inspiration and strength. I wouldn't come this far without your support and sacrifices. Your love means the world to me.

I am especially grateful to my supervisor, Dr. Vincent Boyer, for sharing his extensive knowledge and expertise and his patience and thoughtful guidance since the beginning of my graduate studies. His support has greatly influenced my academic journey and inspired me to strive for the best in my work.

I sincerely thank the Committee members for their valuable insights and constructive feedback, which have significantly enhanced this work.

I also appreciate all the teachers in the systems engineering graduate program for their meaningful lessons and dedication, which inspire us to become better professionals.

I thank my classmates for being such a fantastic group to learn with during this academic experience, for collaborating on group projects, for the shared experiences, and a positive environment.

Finally, I am grateful to CONAHCYT for granting me the scholarship (grant 929768) that allowed me to complete my studies in the Graduate Program in Systems Engineering at FIME, UANL.

Abstract

Oscar Alejandro Hernández López.

As a candidate to obtain the degree of Doctor in Systems Engineering.

Universidad Autónoma de Nuevo León.

Facultad de Ingeniería Mecánica y Eléctrica.

Title of the study: The Dial-a-Tour Problem: A study of Single and Bi-Objective Optimization Methods.

Number of pages: 75.

Introduction With the rise in urbanization and the demand for flexible and efficient travel options, on-demand transportation systems present unique opportunities to improve accessibility, reduce waiting times, and optimize fleet utilization. Understanding and addressing the complexities of these systems is crucial for creating sustainable and responsive urban mobility solutions. This work focuses on a particular situation, which we refer to as the Dial-a-Tour Problem (DATP).

Background The studies on routing problems within operations research are crucial, especially in the tourism sector, which significantly impacts many countries' economies and development. For instance, Mexico, recognized as the most competitive Latin American travel destination, welcomed over 38 million international tourists in 2022, contributing more than eight percent to its gross domestic product. This work addresses a challenge within the tourism sector, aiming to enhance the efficiency and effectiveness of tour routing and scheduling in the tourism industry.

Hypothesis Applying optimization techniques to the DATP can significantly enhance the efficiency of tour routing and scheduling operations for tour operators, resulting in reduced operational costs. Analyzing the problem from a multi-objective perspective can also improve customer satisfaction compared to traditional single-objective optimization methods.

Research questions

- What are the key criteria for optimizing and modeling the DATP when treated as a single-objective problem?
- How can the DATP be modeled, and what algorithms are most likely to be effective for solving it?
- What additional factors make the DATP more realistic when approached as a bi-objective problem?
- How can the Pareto front be used to analyze the trade-offs between objectives?
- What are the key performance metrics for evaluating single and bi-objective solutions?

General objective The study focuses on designing a schedule for tourist reservations while adhering to operational constraints.

Specific objectives

- Formulate a single-objective optimization model for the DATP.
- Develop and implement algorithms and heuristics for solving the single-objective DATP
- Extend to Bi-Objective Optimization, formulating the optimization model and implementing multi-objective optimization techniques, such as Pareto front analysis.
- Evaluate the performance of these methods on benchmark instances to demonstrate the effectiveness of the proposed methods.

Methodology To develop this work, the next methodology will be followed:

1. Literature review related to the DATP.
2. Problem definition.
3. Mathematical Formulation of the Single and Bi-Objective Problems.
4. Solution Approaches: exact methods and heuristics to solve the problems.
5. Experimental Design and Computational Test
6. Analysis of Results

Contributions and conclusions This work contributes to optimizing route planning in the tourism industry, specifically in scheduling vehicle visits to tourism destinations. An efficient heuristic approach, based on a local search, has been proposed for the single objective problem, and a Constraint Programming (CP) approach as an alternative to solve it. In addition, regarded as the bi-objective variant of the problem, a Non-dominated Sorting Genetic Algorithms algorithm is adapted and applied, and also an augmented ϵ-Constraint procedure, combined with the CP model, is proposed to tackle the problem with this multi-criteria perspective.

As a research product, part of this work is already published in a JCR journal:

- Boyer, V., Cervantes–Mendieta E., Hernández-López, Oscar A. and Salazar–Aguilar, M. A. (2024). The Dial-a-Tour Problem, *Computers & Operations Research*, 173, 106832.

Future work Future research can be directed toward testing these methods with heterogeneous vehicle fleets for both single and bi-objective perspectives. For single-objective problems, studying the impact of total requests on vehicle usage can help tour operators estimate the number of vehicles needed through simulations. For bi-objective problems, an in-depth analysis of the trade-offs between objectives in real-world cases is necessary, along with evaluating the benefit of having a larger set of solution options, which can influence algorithm selection.

Thesis structure Chapter 1 introduces the problem situation, relevance, and statement of the DATP. In Chapter 2, a literature review on single and multi-objective related works and their solution methods are analyzed. Chapter 3 covers theories and concepts that support the work on the DATP. Chapter 4 details the mathematical models for both the single and bi-objective approaches, along with the algorithms and heuristics used. Chapter 5 presents the experimental work and results obtained with the proposed methods. Finally, Chapter 6 concludes by summarizing the findings, contributions to the field, and suggestions for future research.

Vincent André Lionel Boyer, PhD.
Advisor

Chapter 1

Introduction

Transportation systems have become increasingly important for modern society, influencing various aspects of daily life, such as where people choose to work, live, and vacation. As transportation shapes these decisions, its role in critical fields like operations research has expanded, especially in routing problems where optimization is essential. This is particularly true in tourism, a key driver of economic growth in many countries. Effective transportation enables tourists to reach their destinations, making it an essential component in the development and success of the tourism industry.

Tourism has been a sector of significant growth over the past decades. As shown in Figure 1.1, global travel and tourism data reflect a steady increase from 1950 to 2020. However, this long-term trend was interrupted by the COVID-19 pandemic, which led to a sharp decline in tourism as health measures were implemented to control the spreading of the virus. According to Statista's Mobility Market Insights [61], tourism revenues shrank by approximately 55 percent in 2019, and international tourist arrivals fell to 407 million during the first year of the pandemic. Despite this setback, tourism began to recover, reaching 969 million arrivals worldwide by 2022, showing resilience in the sector. This recovery is not just a rebound but a continuation of the initial growth trend as the industry adapts to new global challenges and strengthens its role in the worldwide economy.

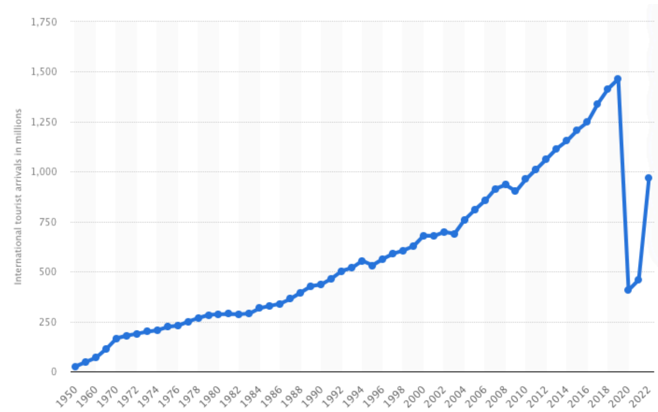

FIGURE 1.1: Number of tourist arrivals worldwide 1950-2022

Source: Statista's Mobility Market Insights [61]

According to the Travel and Tourism Development Index, Mexico is the top travel spot in Latin America. In 2022, it exceeded 38 million international tourist arrivals, solidifying its status as Latin America's top destination for visitors [62].

1.1 Relevance

This study introduces a problem devised in the tourism sector, the Bi-Objective Dial-a-Tour problem, which is approached from a multi-objective perspective. This work contributes to developing and implementing mathematical models that schedule visits to tourist attractions using Mixed Integer Programming (MIP) and Constraint Programming (CP) formulations. Additionally, a Non-dominated Sorted Genetic Algorithm (NSGA-II) has been adapted to the problem's characteristics for its solution.

1.2 Objective

The main objective of this work is to develop multi-objective models for the Dial-a-Tour Problem based on two modeling and solution paradigms, MIP and CP frameworks, and a metaheuristic approach. These paradigms aim to approximate the Pareto Fronts efficiently.

1.3 Problem Statement

The Mexican tourist sector faces a challenge in efficiently transporting numerous visitors from their lodging to the attractions included in their scheduled tours, especially during busy seasons such as spring and summer. Local travel agencies provide a variety of tours that consist of ground transportation, guided activities, and entry to different sites. When a group of tourists book a tour, they indicate the location for pickup, which is usually a hotel, and the travel agency establishes the time for pickup. Subsequently, all the visits to the attractions must be planned, with their respective visit times, to give continuity to the tours. This process results in a complex on-demand transportation problem, ensuring that each group has a structured and enjoyable experience in this sequential planning, maximizing their time at each attraction and minimizing waiting times and delays.

The problem addressed in this work shares characteristics with the above-described scenario. Figure 1.2 represents this situation, where each group of tourists is collected from and dropped off at their designated hotels following their trip, with specific locations and the order of visits known in advance. Each tourist group is represented by a different color and is located at their respective hotels (green nodes). Their routes follow the direction of the arrows to the attractions (grey nodes) and must adhere to the sequence predefined by the two available tours. In addition, each location is characterized by a staying length, representing the time at each attraction required to enjoy the experience.

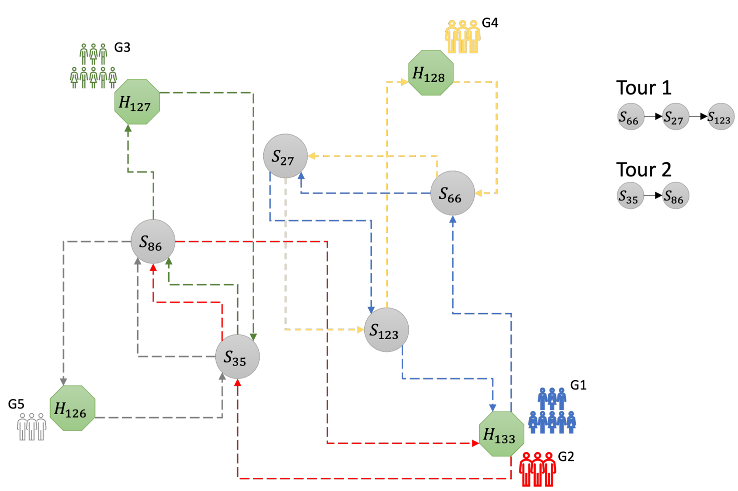

Figure 1.2: Instance of the Dial-a-Tour Problem

The problem requires matching multiple tourist groups with different vehicles for shared transportation and scheduling visits to ensure seamless transfers to local attractions. Reducing vehicle usage in this context is crucial for minimizing operational costs and environmental impact [16, 18]. Travel agencies can significantly reduce fuel consumption, maintenance, and transportation expenses by optimizing vehicle allocation and scheduling. Efficient vehicle usage improves cost-effectiveness and contributes to sustainability efforts, aligning with the growing demand for eco-friendly transportation solutions in the tourism industry. Based on those mentioned above, a primary objective of this problem, emphasizing efficient vehicle utilization, is to minimize the number of vehicles required to fulfill all requests while maintaining a high level of service.

1.4 Thesis Structure

This thesis is organized as follows. Chapter 2 is dedicated to the literature review, covering the applications closely related to the problem at hand, especially within the context of Vehicle Routing Problems: the Pick and Delivery Problem and one of its generalizations, the Dial-a-Ride Problem (DARP). The significant contributions include the development of heuristic algorithms and formulations, mainly based on MIP. The vast majority are related to the single-variant and focus on optimizing cost operation. Furthermore, it refers to multi-objective approaches, addressing aspects related to service quality from the user perspective. They highlight the significance and benefits of adopting such an approach for practical solutions in tourism management.

Chapter 3 provides essential definitions and concepts crucial to support this dissertation. Its purpose is to establish a comprehensive understanding of the algorithms and techniques. Specifically, it addresses the concepts of dominance and PFs for analyzing solutions within a multiobjective framework and examines the metrics used to compare these solutions in terms of performance. The computation of the overall nondominated vector generation, hypervolume, and k-distance metrics are selected to compare the PF. Furthermore, it delves into the fundamental details of CP as the paradigm used for effective problem-solving, describing the interval variables' characteristics and the used constraint structure.

Chapter 4 introduces a mathematical model and a heuristic approach for the single-objective variant of the problem. The proposed model is based on MIP, and the heuristic procedure is based on a pickup-delivery insertion pair as the main operator. In addition, a Local Search procedure is added to refine solutions. Then, the MIP model is extended to formalize the bi-objective variant, where two criteria must be optimized simultaneously. In this transition, other solution strategies for this version are presented. Besides the proposal of the MIP model, an NSGA-II is proposed to tackle the problem from an approximate point of view. Furthermore, two CP approaches are proposed to solve those problems using the augmented ϵ-Constraint method.

Chapter 5 presents the main insights from the investigation. It includes a description of the characteristics of the proposed benchmark used to test the solution methods and its generation process. Additionally, the experimental results are exposed, offering a comparison in terms of the performance of solvers for both single and bi-objective variants of the problem. The single-objective MIP model demonstrated its inability to find solutions for more than five groups within one hour. The Constructive Heuristic generated solutions for all tourism groups. Additionally, the LS demonstrated improvements in the initial solutions generated by the constructive algorithm, reducing the vehicle usage in cases up to 40% to fulfill the requests. The NSGA-II provided more solution points for the bi-objective variant than the other two CP methods. Regarding the hypervolume metric, the first CP model is more effective in exploring the solution space since it reaches higher average values.

Finally, Chapter 6 offers the conclusions arrived with this dissertation. The main contributions, findings, and future lines in which this work can be extended are mentioned. Then, Appendix A and B refer to the tables of results of all instances with each solution method.

Chapter 2

Related literature

2.1 Vehicle Routing Problem

This Dial-a-Tour problem can be considered in a generalized scope as one of the variants of Vehicle Routing Problem (VRP). The VRP is a combinatorial optimization problem in operations research. Solving it efficiently is always the focus of literature due to its significant industrial applications. The main objective is to design the optimal set of routes for a fleet of vehicles to deliver and collect loads from a set of customers [24]. The VRP first appeared in Dantzig and Ramser [17] in 1959 and has since become one of the most studied problems in the literature, and further expanded in many different directions by considering many realistic aspects. Its scope has been extended to include scenarios with time windows [20, 60], split deliveries [1, 22], multiple depots [36, 46], backhauls [27, 64], among others raised in the fields of logistics, transportation, and supply chain management.

Derived from the VRP, a particular problem is closely related to our Dial-a-Tour Problem: The Pick-up and Delivery Problem (PDP). This problem involves determining the best vehicle routes to pick up goods from specified locations and deliver them to designated destinations [28]. It is essential to ensure all the customers' orders on a vehicle are delivered to their corresponding delivery locations efficiently. In some applications, all stops are predetermined [35] and other real-time scenarios where requests are revealed, particularly relevant for ride-sharing and on-demand delivery services [37].

2.2 The Dial-a-Ride Problem

Within the existing literature, a well-known transportation optimization problem closely resembles the problem at hand: the Dial-a-Ride Problem (DARP). The DARP also can be seen as a generalized form of the PDP and is a type of application used by travel agencies for transporting people from an origin location to a place of destination Several variations can be found among the extensive related works to this problem.

The DARP was introduced by Cordeau and Laporte [15]. Figure 2.1 shows the related research originated from this seminal work discussed in this chapter. The figure visually represents the relationship between these contributions, showing how these works are linked to the original and how they have been extended. The interconnections and dependencies of the presented works demonstrate the wide range of extensions with the different features considered and detailed in this chapter.

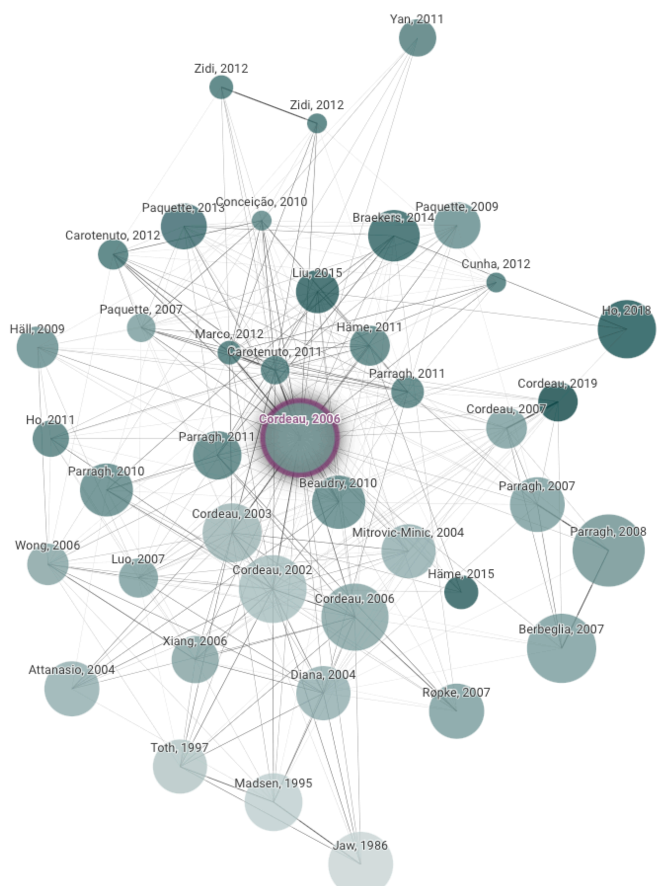

FIGURE 2.1: Related works to the DARP

A DARP refers to determining the most efficient service routes and pick-up and drop-off times for the customers. In the DARP, clients create transportation requests for moving between specific starting and ending points (i.e., from a pick-up to a drop-off location). The transportation is performed by vehicles offering shared services, meaning multiple clients may share the same vehicle simultaneously.

According to Cordeau and Laporte [13], three decisions relate to constructing a DARP solution. These are identifying clusters of customers that will be served by the same vehicle, arranging customers into a vehicle route, and determining the sequence of pick-up, driving, and drop-off activities for each route.

Besides all the commonly addressed objectives related to cost operation, another important one related to the client is the Quality of the Service (QoS). Dial-a-Ride services, involving tangible aspects like vehicles and time, allow for the inclusion of measurable quality aspects in service specifications, aligning with practices found in the literature by Paquette et al. [50]. In their work, they surveyed some related studies and mentioned some standards of quality utilized by operations researchers. to conduct the QoS. Among the main attributes we can cite the difference between real and intended delivery time, waiting times while being transported or before the pick-up, total waiting time, maximum ride time, average ride time, a ratio of actual ride time to direct ride time, and maximum number of stops while a passenger is on the vehicle.

The DARP can be studied under two distinct modes. In the static variant, all transportation requests are pre-determined, allowing route planning in advance [8]. In the dynamic mode, transportation requests unfold during the day while the service is already in progress, resulting in real-time construction of vehicle routes due to the changes in the solution over time [4, 9, 42]. Parragh et al. [54] provides modeling aspects about both static and dynamic modes. Even when operating in the dynamic mode, it is common to initially address a static problem for a predefined set of requests to establish an initial solution, which can be adjusted as new requests arise. Further requirements to implement such a system can also be found in the work of Conceição [11], which focuses on the static version of DARP.

Cordeau [12] proposed a Mixed Integer Programming (MIP) with a 3-index problem formulation and a polynomial number of constraints. They formulate the problem on a complete directed graph, separating nodes in subsets of pick-up and drop-off nodes respectively, and each arc has a defined routing cost of traveling. Their objective is to minimize the total routing cost. To solve the problem they proposed a Branch-and-Cut (B&C) algorithm, solving a Linear Programming (LP) relaxation at first, where valid inequalities are generated by a separation heuristic process identifying violations in the sub-tour elimination constraints, capacity, precedence, and order constraints.

Ropke et al. [57] exposed two formulations related to the Pick-up and Delivery Problem with Time Windows (PDPTW) and the DARP, introducing new B&C algorithms for the defined problem. Based on the aforementioned work of Cordeau [12], their work leads to refined formulations of the problem with fewer variables and tighter bounds, incorporating valid inequalities that integrate the pickup and delivery framework with the constraints of vehicle capacity or the time window. Also, the authors present the results of computational experiments on various sets of test instances, demonstrating the capability to successfully solve cases involving up to 8 vehicles and 96 requests.

As stated by Ropke et al. [57], PDP, PDPTW, and DARP represent extensions of the classical VRP [63], which results in NP-hard complexity. Consequently, their solution methods have predominantly centered on heuristics. However, it is noteworthy that the imposition of specific constraints allows for a more tractable solution approach, facilitating the attainment of optimality within reasonable computational limits. Thus, methodologies such as branch-and-cut have successfully solved several related problems.

Parragh [52] modeled the problem based on a usually complete directed graph but introduced in her application different types of users. Comparing the two and three index formulations proposed in this work, the first one proves to be more efficient for the problem. The author implemented a B&C algorithm to solve those models, combining the branch-and-bound and the cutting-plane ideas. As it is known, the optimal solution derived from the LP relaxation serves as a lower bound for the original MIP problem. Moreover, all families of valid inequalities intended solely to strengthen the model are incorporated in a cutting-plane manner. Separation algorithms are responsible for checking the current solution to identify potential violations of the excluded constraints and the previously defined valid inequalities, which are referred to in the works of Cordeau [12] and Ropke et al. [57]. Additionally, to accelerate the optimization process an adjusted version of a Variable Neighborhood Search (VNS) [45] is applied to compute initial upper bounds.

Braekers et al. [6] extend the previous problem defined by Parragh [52], but with multiple depots, and also adapted the B&C algorithm by Ropke et al. [57]. A deterministic annealing metaheuristic has been developed to address larger instances of the problem. It can be described as a variant of the simulated annealing metaheuristic and is based on the work of Dueck and Scheuer [23]. The authors test their approaches on a benchmark for the heterogeneous DARP, especially to evaluate their B&C algorithm, and introduced new specific instances for their problem as well.

Hüll et al. [31] studied a variant where a portion of each journey can be conducted through a fixed route service. The paper proposes an arc-based formulation and demonstrates strategies to enhance its solvability, including arc removal, substitution of variables, and the inclusion of sub-tour elimination constraints. The study explores the application of an exact solution method on small instances of the problem. Additionally, the paper outlines procedures for generating and visualizing input and output data using a geographic information system.

Liu et al. [38] also formulated the problem using an MIP along with other specific constraints, like manpower planning. They introduced valid inequalities to strengthen their proposed models and developed a B&C algorithm, which can solve instances of up to 22 requests. Their results are tested on data from The Non-Emergency Ambulance Transfer Service in Hong Kong.

Cordeau and Laporte [14] proposed a Tabu Search (TS) algorithm. It begins with an initial solution and then, at a certain iteration, transitions from the current one to the best option within its defined neighborhood. To prevent repetitive patterns, solutions that share certain characteristics with the last explored solutions are designated as "forbidden" for a specific number of iterations, except in cases where they represent a new best solution.

Madsen et al. [41] described a system for solving a DARP variant including time windows, and focusing on minimizing vehicle utilization costs. While no mathematical model is presented, the study outlined a heuristic algorithm capable of weighting total driving time, vehicle usage, waiting time, and cost by incorporating user-specified parameters that influence tour characteristics. The heuristic is primarily based on an insertion algorithm. It begins by considering the next not-allocated customer in the vehicle schedules. For each vehicle, it generates all feasible insertions of the customer in the schedule and calculates the change in cost. If a feasible insertion exists, it selects the one with the minimum change in cost and inserts the job in the schedule. If no feasible insertion is possible, the customer is added to a list that cannot be served. Customers who cannot be allocated to any vehicle are usually assigned to taxis, which can then be added to the vehicle list.

Diana and Dessouky [21] also presented a heuristic algorithm based on insertions, which focuses on minimizing the weighing sum of three factors: the distance traveled by vehicles, the ratio of ride time over a direct time for the customers, and the idle times in their schedules. The approach maintains the concept of ranking requests by pickup time but introduces two key modifications. First, it avoids initializing a route with a request that could be simply added later. Second, it allows swaps in a ranking order to prioritize requests that might be challenging to insert later due to their location. The approach aims to find the best insertion for each unrouted request in each itinerary, minimizing the related cost. A cost matrix is constructed, with rows representing requests and their columns the routes. Requests with no feasible insertion are assigned a high cost. The regret for each request is then calculated as the total difference between each element in the row and the smallest. The request with the largest regret is next inserted. This process is repeated until all requests are added or all regret costs are zero, indicating that cannot be insertions into any existing route. The authors mentioned that although minimization of vehicle usage is not intended at first, the algorithm gradually decreases it in subsequent runs until some requests can no longer be accommodated.

Jaw et al. [33] propose another heuristic algorithm based on sequential insertions of all requests, identifying the best insertions. Feasibility in this context refers to the condition that the insertion of a customer into a vehicle route should not violate any service-quality constraints for both the new customer and existing customers to that particular vehicle. The optimization process focuses on reducing the extra "cost" associated with inserting the customer into the vehicle schedule. The cost function is defined as a weighted sum, encompassing the disutility experienced by all customers (attributed to excess ride times and deviations from optimal pick-up or delivery times) and system costs, quantified by a function that gauges the utilization of available vehicles.

Luo and Schonfeld [40] presented a heuristic called rejected-reinsertion for minimizing the number of vehicles required to fulfill the entire demand of requests. Whenever inserting a new request into the existing vehicle routes becomes infeasible the rejected-reinsertion operation is used. It functions by temporarily removing an assigned request, closely located to the new one in both timeframe and location, from its current vehicle. Subsequently, the new request is added to the vehicle route, followed by reinserting the removed request elsewhere in the schedule. Computational testing across two sets of problems demonstrates that this heuristic yields vehicle reductions of up to 17% compared to a parallel insertion heuristic, showcasing its computational efficiency.

Muelas et al. [47] proposed a VNS algorithm, its fundamental concept is to decompose the initial set of requests into independent subsets and optimize its specific set of routes independently. This decentralization allows for parallel processing. In this work, the authors aim to minimize the total cost of the routes with an evaluation function that considers violations in pick-up times, loads, and riding times. In the initial step in the algorithm, rather than opting for a random partitioning strategy, employ the k-medoids method [39], used when dealing with the clustering of large datasets. It creates a pattern of defined similar requests with the hypothesis that grouping in such a way enhances the ability to compute efficient routes, contrasting with an approach where each request has an equal probability of being assigned to any partition.

Parragh et al. [55] presented a VNS-based heuristic. This methodology starts from an initial incumbent solution and then, at each step, generates a random solution in the current neighborhood using a shaking process. Then, a Local Search (LS) is applied to replace the incumbent if it produces an improvement. Otherwise, the next larger neighborhood is utilized in the next iteration. Their approach also permits ascending moves, meaning that deteriorating solutions may become incumbent solutions with a certain probability. Also, graph pruning and time window tightening techniques, as described in Cordeau [12] before initiating the optimization procedure, are used to narrow down the search space.

Xiang et al. [66] considered a variant with a heterogeneous vehicle fleet and a team of drivers with various skills to fulfill customer requests. The paper describes a specific heuristic designed to address a large-scale static DARP, which employs LS and diversification strategies to improve initial solutions, followed by an intensification strategy for further refinement. Computational tests on generated instances are performed to test and compare the algorithm with the lower bounds obtained from the column generation method, particularly for instances with up to 200 requests. The authors concluded while the results are not highly sensitive to the initial solution, investing effort in constructing a good initial solution significantly reduces computational time. With a well-constructed initial solution, the heuristic solves instances with up to 2000 requests within 10 hours.

Wong and Bell [65] worked on a variant for the transportation of elderly and handicapped individuals, where specialized vehicles with facilities like lifts are necessary, hence a multi-dimensional nature of vehicle capacity is considered. They proposed an insertion heuristic where trips are ranked based on their insertion difficulty, with priority given to those farther from the center and other requests. In the insertion phase, feasible ways to insert each request into existing routes are identified for each vehicle, considering spatial and temporal factors. The best insertion position with the minimum additional cost to the objective function is determined for each request. If no feasible insertion is possible due to high fleet utilization, a taxi is dispatched. This process is iterated until all requests are accommodated or until no further insertions are possible. The algorithm aims to minimize total transportation costs.

Yan and Chen [67] addressed an application closely related to the DARP, the carpooling problem (the practice of multiple individuals sharing a private vehicle to travel along a partially common route between their respective starting and ending locations). The problem is formulated as a network flow model, which struggles to find optimal solutions within 5 hours. Therefore, a solution procedure based on

Lagrangian relaxation and subgradient methods [25] is developed. They first relax the capacity constraints to formulate a Lagrangian problem, which is subsequently solved to obtain the optimal solution lower bound. Next, a Lagrangian heuristic is devised to calculate the optimal solution upper bound. A sub-gradient method is employed to adjust the Lagrangian multipliers iteratively, refining both bounds. This process continues until an acceptable convergence result is achieved or until the preset number of iterations is exceeded.

Masson et al. [43] treated the problem with the particularity of including transfers, meaning that users may switch vehicles in their trips at specific points called transfer points. Their solution is based on an Adaptive Large Neighborhood Search (ALNS) metaheuristic. Following its principle of “destroying” and “repairing” a solution iteratively to improve it. The destroying operator is done by removing some of the solution requests, and the repair step tries to reinsert the destroyed requests. Then, an acceptance criterion is used to update the current solution. The operators are selected with a roulette wheel principle. The authors evaluated the algorithm through real-life and generated instances, comparing heuristic solutions of the DARP and the DARP with transfers, affirming that the achieved cost savings attributed to transfers can reach up to 8% in real-life instances.

For further reference, the reader may refer to the surveys of Ho et al. [30] and Berbeglia et al. [5], which cite other contributions about the DARP and its variants, motivated by a renewed interest in shared-ride applications for the general public in recent years.

2.3 The Multi-Objective Dial-a-Ride Problem

The DARP has also been studied from a multiobjective perspective since, in many real-life applications, it is often most intuitive and realistic to model this practical problem by considering multiple criteria simultaneously, allowing a more comprehensive representation of the complexities and nuances presented in real-world scenarios. A recent work of Zajac and Huber [68] surveyed a wide range of multi-objective routing problem variants and their corresponding trade-off, showing how objectives are handled. The main objectives treated in those applications are related to the minimization of routing and total costs, as well as total distances and the balance in the makespan of routes. In addition, the authors describe adopted solution approaches for multi-objective routing problems, which help understand the advantages and disadvantages of the applied strategies.

Paquette et al. [51] combine the multicriteria analysis and a TS procedure [14, 26]. The authors incorporated cost minimization and terms related to the QoS as objectives. Their approach is tested on generated random instances and data collected from a transporter in Montreal. Their algorithm uses a search mechanism where The weights for the criteria are adjusted dynamically, leading to the elimination of the resulting dominated solutions. The TS is used in conjunction with the multicriteria reference point approach introduced by Clímaco et al. [10] to identify a set of non-dominated feasible solutions that collectively form an approximation of the Pareto front, constituting their framework.

In their multiobjective approach, Baugh Jr et al. [3] propose tackling the problem with a simulated annealing technique. Their considered objectives are the maximization of customer satisfaction, measured through the degree of violations of time windows, and on the other hand, the cost minimization, characterized by the total travel time of all required vehicles. The algorithm initiates with an initial solution and a given temperature. At each step, a new solution is generated and accepted if it results in an improvement to the objective. and also can be accepted with a probability if it worsens. Once a specified number of steps has been completed, the annealing temperature is decreased, and the process is iteratively repeated until the system converges and stops with a solution. In their work, the authors prove the intractability of the DARP from this perspective and state the need for heuristic techniques to deal with the problem. Although the procedure was not validated, they tested their algorithm on a small dataset provided by the Winston-Salem Transit Authority, which serves over 300 customers daily.

Parragh et al. [53] faced a multi-objective DARP for a major ambulance dispatcher in the Austrian Red Cross, with a two-phase solution heuristic procedure where they try to minimize costs of the total travel distance of the vehicles and a client-centered objective measured by the average user ride time. First, they implemented an iterated VNS-based heuristic, generating approximate weighted sum solutions. A second phase uses ideas of path relinking [56] from the approximated set obtained by the first phase, aiming to generate all those solutions not found previously.

2.4 Summary

A summary of the relevant related work to single objective variants is provided in Table 2.1. The table shows the solution techniques applied by the authors in their works. The *Variants* column outlines the main features included. A capacity constraint for the vehicle is represented as (c), and other considerations as heterogeneous vehicles (hv), waiting time constraints (wt), time windows for customers (tw), multiple depots (md), manpower planning considering working hours on staff resources (mp), vehicles performing multiple trips per day (mt), quality of service constraints, given by deviations concerning maximum riding times (qos), customers can share rides while traveling (rs), and transfers, i.e. users may change vehicles during their trip (t). The *Objective* column specifies what authors attempt to optimize, and that includes minimization of total routing cost of vehicles (trc), travel times (tt), and vehicle usage (u).

Table 2.1: Literature Review Summary

Author(s)	Solution techniques			Variants	Objective
	Exact	Heuristic	Metaheuristic		
Hüll et al. [31]	MIP			c	trc
Ropke et al. [57]	MIP			c-tw	trc
Parragh [52]	MIP		VNS	hv-wt	trc
Braekers et al. [6]			SA	hv-wt-md	trc
Liu et al. [38]	MIP			c-mt-mp	tt
Cordeau and Laporte [14]			TS	c-tw	trc
Madsen et al. [41]		X		c-tw	trc
Jaw et al. [33]		X		c-qos	trc
Luo and Schonfeld [40]		X		qos-tw	u
Muelas et al. [47]			VNS	c-tw-wt	trc
Xiang et al. [66]		X		hv-tw	trc
Wong and Bell [65]		X		c-hv-wt	trc
Yan and Chen [67]	MIP			c-rs	trc
Masson et al. [43]			ALNS	c-t-qos	tt

From the bi-objective perspective, the problem of the related works reflects its practical nature and complexity in real-world scenarios. Thereby, heuristic techniques are predominant as solution strategies. Common across these studies is the emphasis on balancing cost efficiency with service quality, highlighting the need for specialized heuristic methods to manage the intricacies of the DARP.

Previous studies have explored multi-objective routing problems in various contexts, such as urban transportation, ambulance dispatch, and general logistics. However, this work addresses the complexities specific to tourism. This contribution considers traditional objectives like minimizing vehicle usage, which directly affects travel costs and distances, while also integrating tourism-specific criteria, such as optimizing schedules to reduce waiting times, which helps improve the overall tourist experience. Furthermore, by employing a wide range of optimization techniques, this work aims to provide practical solutions that contribute to the field of tourism management.

Chapter 3

Theoretical basis

3.1 Dominance and Pareto Fronts

The Pareto Front (PF), also known as the Pareto frontier, represents the set of optimal solutions in a multi-objective problem. In such problems, there are multiple conflicting objectives, and it's not possible to identify a single solution that optimizes all objectives simultaneously. The PF consists of a set of solutions where no other is superior in all objectives simultaneously in the search space, as well as the study of their corresponding trade-off [48]. In this context, a solution is said to dominate another if it is at least as good as the other solution in all respects and strictly better in at least one [49]. This concept of dominance forms the basis for comparison and evaluation, providing a clear framework for identifying superior solutions and informing decision-making processes. It is important to highlight that the PFs generated in this work are two-dimensional since two objectives must be optimized: minimizing vehicle usage and on the other hand, waiting times, which refers to the quality of service.

Solving a bi-objective optimization problem involves the process of building the PF, as the set of efficient solutions aforementioned. Figure 3.1 illustrates this concept; suppose that f_1 and f_2 are functions with minimizing objectives (smaller values are preferred), and points 1, 2, 3, 4, and 5 represent possible solutions or decisions. Based on these assumptions and the dominance concepts, solution 4 is dominated by solution 2, since the latter is preferred for both objectives. Similarly, solution 5 is dominated by solution 3. In conclusion, one can approximate the Pareto Frontier with solutions 1, 2, and 3, since no other solution is equally good or better than them to all objectives.

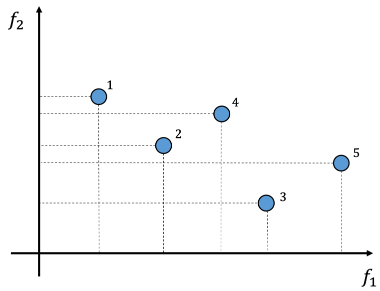

Figure 3.1: Dominance in a Pareto Front

3.2 Performance of multi-objective algorithms

Several methods have been suggested for the assessment and comparison of these PFs [2, 34]. A key consideration is how to conduct equitable comparisons between two or more front approximations. This task becomes particularly challenging, especially when the shape and size of the actual PF are generally unknown.

This work incorporates three metrics to evaluate the PFs obtained from competing algorithms. The quality of a PF approximation depends on two factors: (1) the closeness of the points on the estimated front to the true PF, and (2) the variety of points on the estimated front, where increased diversity typically signifies improved performance [7]. Based on this previous idea, it has been decided to include the Overall Nondominated Vector Generation ($ONVG$), the Hypervolume (H), and the $K-distance$ as metrics to ensure a more insightful comparison between the generated PFs. This way, it is aimed to result in better decision-making and compensate for the drawbacks of using the information from just an individual metric.

The $ONVG$ is a metric that is simply defined as $|PF|$. It measures the number of distinct nondominated points produced. This metric is independent, induces a complete ordering on the set of approximations, and represents a cardinal measure [34].

The Hypervolume (H), or extent of space occupied, is a metric proposed by Zitzler and Thiele [69]. It indicates the space limited by a reference solution and the set of non-dominated solutions S; returning the area which contains solutions dominated by at least one member of S. Therefore, the larger the H, the better the Pareto approximation.

The H of a PF depends on a reference point defined by the user, which influences the ranking of competing fronts. In instances where the true PF is unknown, Cao et al. [7] state the reference point $\mathbf{r}$ is typically selected as the nadir point, which is a vector containing the worst values of each criterion as its components, or a point marginally (slightly) worst to this nadir point. Then, the hypervolume, in a minimization context, can be formalized as:

$$H(PF, \mathbf{r}) = \bigcup_{\mathbf{s} \in PF} space(\mathbf{s}, \mathbf{r}) \tag{3.1}$$

where $space(\mathbf{s}, \mathbf{r}) = \{\mathbf{v} \in \mathbb{R}^2 | \mathbf{r} \succ \mathbf{v} \succeq s\}$, is the criterion space, formed by 2-dimension rectangles that contain all criterion vectors $\mathbf{v} \in \mathbb{R}^2$, dominated by the elements $s \in$ PF. Then, Figure 3.2 shows a hypothetical 2-dimensional PF which consists of 5 points, with the nadir point $(f_1(\chi)_{worst}, f_2(\chi)_{worst})$ where each component corresponds to the maximum value that we can expect in each criterion. Each point $(f_1(\chi)_i, f_1(\chi)_i)$ represents a non-dominated solution vector. The colored area represents graphically the hypervolume metric. It has been constructed taking the Nadir as the reference point, and the values have been normalized to $[0, 1]$ to transform the values of the two objective variables to a common scale.

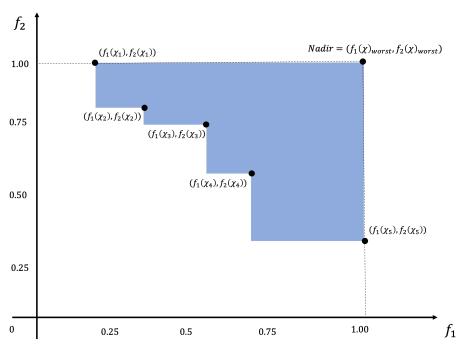

Figure 3.2: Hypervolume representation

When comparing PFs to a common reference point, they can be put together as shown in Figure 3.3. In this case, four PFs are presented, and with this metric one can state that PF_1 dominates PF_2, PF_3, and PF_4, since it has the maximum value in the hypervolume. $H(PF_2, \mathbf{r}) > H(PF_3, \mathbf{r})$ as solutions 4, 5, and 6 in PF_3 are jointly dominated by solutions 4 and 5 in PF_2, despite the first three solutions being common to both fronts. Moreover $H(PF_2, \mathbf{r}) > H(PF_4, \mathbf{r})$ since PF_2 has two more solutions than PF_4.

In addition to the hypervolume indicator, it is also considered to evaluate the PFs according to the $k - distance$, which evaluates a degree of dispersion of the generated solutions, originally employed by Zitzler et al. [70] to assess individuals

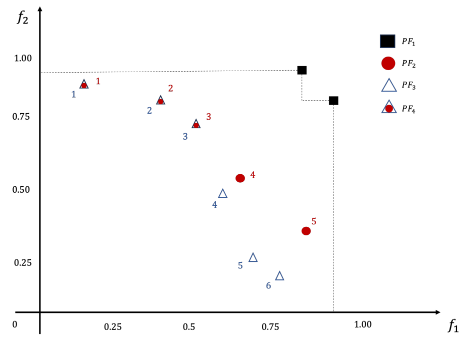

Figure 3.3: Hypervolume comparison

within their SPEA2 algorithm. Based on the k-nearest neighbor approach of Silverman [59], the k-distance for each point quantifies its Euclidean distance to the k-th nearest point. It is important to note that this metric is established for each point independently. Subsequently, the average k-distance is computed to evaluate the entire approximation set.

3.3 Constraint Programming Framework

Constraint Programming (CP) provides an alternative approach next to the MIP for problem modeling. However, there is currently no universally accepted CP language; the syntax and expressiveness depend on the solver. In this dissertation, only the constraints found in IBM's CP optimizer [32] are used.

Mixed Integer Programming models solve optimization problems using branch-and-bound, branch-and-cut, or cutting-plane algorithms. These methods explore possible solutions by relaxing the integer constraints, solving the continuous relaxation, and iteratively narrowing down the feasible region until an optimal integer solution is found. In contrast, CP focuses on constraint satisfaction rather than optimization, using techniques like backtracking and constraint propagation to systematically prune the search space and find feasible solutions that satisfy all constraints. This makes this approach more flexible for handling complex or highly combinatorial constraints [58].

Generally speaking, CP models use interval variables to represent an interval of time when an activity is performed, which allows us to refer to them as the operations in our models. An interval variable $\alpha = \{r, d, t, o\}$ is defined as this tuple, representing respectively its earliest time of the interval, its latest end time, and minimum duration. In addition, there is a characteristic specifying if it is optional (no obligation to schedule the operation) or mandatory. Interval variables include a start time $startOf(\alpha)$ and an $endOf(\alpha)$. They can also be set as absent and, in this case, should be ignored by any expression or constraint in the model. Additionally, Table 3.1 describes the type of constraints used in the formulation to construct our models.

TABLE 3.1: Description of Constraint Programming constraints

Constraint	Description
$presenceOf(\alpha)$	States if interval variable α is present in the solution.
$alternative(\alpha$,B)	If interval α is present, then exactly one of the intervals in set B is present.
$endBeforeStart(\alpha,\beta)$	Ensure the ending time of interval α must occur before or at the same time as the starting time of interval β.
$startBeforeEnd(\alpha,\beta,z)$	This constraint ensures that the starting time of interval α, in addition to a specific duration z must occur before or at the same time as the ending time of interval β.
$span(\alpha, B)$	If interval α is present spans over all present intervals from the set B. This constraint is satisfied automatically if either of the intervals is absent.
$noOverlapSequence(B, dist)$	Ensures that the intervals sequence in set B do not overlap. The distance matrix *dist* specifies a sequene dependent setup times for each pair of activites. Absent intervals are ignored as well.

Chapter 4

Methodology

To tackle the Bi-Objective DATP, it is important to derive efficient approaches for the mono-objective variant. This chapter presents the mathematical model and a heuristic approach for the single-objective DATP; then, the model is extended to the bi-objective problem.

The DATP involves assigning tourist groups from a set G to vehicles and creating efficient routes for picking up and dropping off these groups at the places where they are hosted or any designated location in their reserved tour T_g. For each tour, T_g precedence of the visits is known in advance and the staying duration that is required to carry out the related task on each site. All visits must be completed within an established planning horizon h. The objective function is to minimize the number of vehicles required to address all tourist requests. Each vehicle begins its route with the initial pickup and ends with its last delivery operation. After visiting one location, a group of tourists might be moved using a different vehicle to a different place. Transshipment and splitting groups are prohibited while traveling from a pickup point to a delivery point.

The DATP can be modeled on a directed, weighted graph $D(V, A)$, where each pickup and delivery operation is associated with a node i. The vertex set $V = \{T_g\} \cup \{o_g\} \cup \{d_g\}$, $\forall g \in G$. Each set T_g contains a duplicate element for the pick-up and delivery operations at each location to be visited by a group g necessary to complete its tour. T_g is ordered in such a way that if i is a pick-up operation, then $i + 1$ represents its corresponding delivery. The precedence relationship specifies the order of T_g. Vertices o_g and d_g are the origin and destination nodes for each group g. Then, o_g represents the initial pickup and d_g the final delivery at the hosting location of group g.

Arcs set A represents the connections of spots. Each arc $(i, j) \in A$ has a known cost of t_{ij}, denoting the travel time from the precedent pick-up operation in node i

to the delivery operation at node j. The summary of the notation used in this work is given in Table 4.1.

TABLE 4.1: Sets and parameters of the DATP.

	Description
Notation	
o	the virtual initial depot node representing the origin of the vehicle fleet.
d	the virtual final depot node representing the final destination of the vehicle fleet.
o_g	node associated with the first pickup of tourist group $g \in G$.
d_g	node associated with the last delivery of tourist group $g \in G$.
M, M'	big numbers to ensure values are tight enough to reflect actual bounds in the used constraints.
Sets	
K	set of vehicles.
G	set of tourist groups.
T_g	A sequenced collection of nodes linked to every pickup and delivery task to finalize the tour of a group $g \in G$., including o_g and d_g.
N	set of all nodes ($N = \bigcup_{g \in G} T_g \ \cup \{o, d\}$)
$N^* = N \setminus \{o, d\}$	set of all nodes without the initial and final virtual depot.
$N^+ \subset N^*$	nodes related to pickup requests.
E_g	set of 2-tuples of adjacent nodes $(i, j) \in T_g^2$, $g \in G$, such that j must be performed after i.
Parameters	
h	planning horizon.
Q	capacity of the vehicles.
p_g	size of tourist group $g \in G$.
$\underline{L}_g$	minimum time required to complete the tour scheduled by tourist group $g \in G$.
t_{ij}	traveling time from request i to request j (Note that, if $i \notin N^+$ and $(i, j) \in E_g$, for a group $g \in G$, l_{ij} is equal to the staying time at the spot of request i).
δ_i	equals to 1, if $i \in N^+$ (pickup node), and -1 otherwise.

4.1 Single objective Dial-a-Tour Problem

4.1.1 Mixed Integer Programming Model

The DATP is modeled with the following decision variables:

$$x_{kij} = \begin{cases} 1, & \text{if vehicle } k \text{ travels from node } i \text{ to node } j, \\ 0, & \text{otherwise.} \end{cases}$$

u_i : position value of node $i \in N$ within a route.
q_i : vehicle load at node $i \in N$.
s_i : arrival time at node $i \in N$.

$$\text{Min} \sum_{k \in K} \sum_{i \in N \setminus \{o,d\}} x_{koi} \tag{4.1}$$

subject to:

$$\sum_{j \in N \setminus \{o\}} x_{koj} = 1 \qquad \forall k \in K \tag{4.2}$$

$$\sum_{\substack{j \in N \setminus \{d\} \\ i \neq j}} x_{kji} = \sum_{\substack{j \in N \setminus \{o\} \\ i \neq j}} x_{kij} \qquad \forall i \in N \setminus \{o,d\}, \forall k \in K \tag{4.3}$$

$$\sum_{k \in K} \sum_{\substack{j \in N \setminus \{d\} \\ i \neq j}} x_{kji} = 1 \qquad \forall i \in N \setminus \{o,d\} \tag{4.4}$$

$$s_i + t_{ij} \leq s_j + (M + t_{ij}) \left(1 - \sum_{k \in K} x_{kij}\right) \qquad \forall i \in N \setminus \{d\}, \forall j \in N \setminus \{o,i\} \tag{4.5}$$

$$s_i + t_{ij} \leq s_j \qquad \forall (i,j) \in E_g,\ \forall g \in G \tag{4.6}$$

$$\sum_{\substack{m \in N \cup \{o\} \\ m \neq i}} x_{kmi} = \sum_{\substack{m \in N \cup \{o\} \\ m \neq j}} x_{kmj} \qquad \forall k \in K, \forall g \in G, \forall (i,j) \in E_g \ with\ i \in N^+ \tag{4.7}$$

$$q_i + \delta_j p_g \leq q_j + (M' + \delta_j p_g) \left(1 - \sum_{k \in K} x_{kij}\right) \qquad \forall i \in N \setminus \{d\}, \forall g \in G, \forall j \in T_g \tag{4.8}$$

$$u_i - u_j - |N| x_{kij} = |N| - 1 \qquad \forall k \in K \tag{4.9}$$

$$x_{ijk} \in \{0,1\} \qquad \forall (i,j) \in N^2 \tag{4.10}$$

$$s_i \in \mathbb{R}\ and\ q_i \in \mathbb{R} \qquad \forall i \in N \tag{4.11}$$

$$u_i \in \mathbb{Z} \qquad \forall i \in N\ with\ u_o = 0\ and\ u_d = |N| \tag{4.12}$$

The Objective Function (4.1) is to minimize the number of vehicles used. Constraints (4.2) guarantee that a maximum of k vehicles will depart from the initial depot. Constraints (4.3) ensure the flow preservation at each node. Constraints (4.4) indicate mandatory departures and arrivals. Constraints (4.5) ensure the vehicle's time consistency. In these constraints, $M = h$ has been set. Constraints (4.6) guarantee precedence in the tour operations of each group. Constraints (4.7) ensure subsequent pickup and delivery operations must be performed by the same vehicle. Constraints (4.8) assure the vehicle capacity is not exceeded, setting $M' = Q$. Constraints (4.9) represent sub-tour elimination for nodes whose travel time between them is zero, for example, in cases where a vehicle performs a delivery and then a pick-up at the same node. Constraints (4.10)–(4.12) define the domain of the decision variables.

4.1.2 Heuristic Procedure

The total number of available vehicles directly impacts the size of the MILP formulation. To mitigate this, we aim to reduce the number of variables and constraints associated with K by creating a feasible solution by iteratively inserting requests in a route while maintaining feasibility.

This approach is detailed in Algorithm 1. The algorithm starts by selecting requests that have not yet been assigned. Each request is paired as a pick-up and delivery and then searched for insertion candidates. Requests are ordered to their earliest release time. A best-insertion process is used to determine the insertion point for each request, considering route length and minimal delays. This process selects the best insertion among all available routes at that moment. If a request cannot be assigned to a route, a new route is created. In cases where a new route is opened for a request, all preceding operations of the corresponding group are assigned to the new route opened for the new vehicle to avoid idle time.

Algorithm 1: Constructive heuristic

Data: A partial Solution $\mathscr{V}$ ($\mathscr{V}$ may be empty)

Let $\mathscr{R}$ be the list of unassigned pickup-delivery pair (i, j) in $\mathscr{V}$;
while $\mathscr{R} \neq \emptyset$ **do**
    Let $(i, j) \in \mathscr{R}$ pickup-delivery pair with the earliest release time;
    $\mathscr{R} \leftarrow \mathscr{R} \setminus \{(i, j)\}$;
    Try to insert the pair (i, j) together in any route in $\mathscr{V}$ while minimizing the resulting delay in the overall solution;
    if *no feasible insertion for (i, j) is found* **then**
        Open a new route R and insert(i, j) in R;
        Remove from $\mathscr{V}$ all the group requests preceding (i, j) and insert them in R;
        $\mathscr{V} \leftarrow \mathscr{V} \cup \{R\}$;

4.1.3 Local Search

To refine solutions iteratively and seek better ones obtained by the previous Constructive Heuristic, a Local Search (LS) algorithm is proposed. Algorithm 2 details this method, which systematically explores the neighborhood of the current solution to find an improved one. The neighborhood structure is created by randomly selecting a vehicle and removing all its operations. Moreover, it eliminates all subsequent operations of groups served by this vehicle from the overall solution. Then, Algorithm 1 is used to reassign the operations. For example, if the chosen vehicle finishes a pick-up and delivery for a specific group, all subsequent operations after this pair are reassigned. This may include operations not executed by the selected vehicle, offering flexibility during the solution reconstruction phase.

The method aims to improve the order of requests in the corresponding routes of a solution. This may involve rearranging operations within routes to reduce total tour lengths and vehicle usage. If no improvement is identified, the algorithm resets from a perturbed solution. This perturbation randomly eliminates operations from routes in the current solution based on a specified probability, intending to introduce variety and potentially evade local optima, with Algorithm 1 used for repair.

Algorithm 2: Local Search

Data: An initial Solution $\mathscr{V}$
Result: $\mathscr{V}_{best}$

$\mathscr{V}_{best} \leftarrow \mathscr{V}$;
$Iter \leftarrow 0$ (Number of successive iterations without improvement);
while $Iter < MaxIter$ ***and*** *time limit is not reached* **do**
    Let $\mathscr{C}$ be the set of candidate routes in $\mathscr{V}$;
    while $Iter < MaxIter$ ***and*** *time limit is not reached* **do**
        Choose randomly a route $R \in \mathscr{C}$;
        $\mathscr{C} \leftarrow \mathscr{C} \setminus \{R\}$;
        $\mathscr{V}' \leftarrow \mathscr{V} \setminus \{R\}$;
        Remove from $\mathscr{V}'$ all requests $\{i, ..., d_g\} \subset T_g$ of group $g \in G$ if $i \in R$;
        Repair $\mathscr{V}'$ with Algorithm 1;
        if $\mathscr{V}'$ *uses less vehicles than* $\mathscr{V}$ **then**
            $\mathscr{V} \leftarrow \mathscr{V}'$;
            $\mathscr{C} \leftarrow \emptyset$ (we exit the while loop);
    if $\mathscr{V}$ *has not been improved* **then**
        $Iter \leftarrow Iter + 1$;
        Remove randomly 50% of the pairs (i, j) of pickup and deliveries from $\mathscr{V}$ and repair it with Algorithm 1;
        $\mathscr{V} \leftarrow \mathscr{V} \cup \{R\}$;
    if $\mathscr{V}$ *uses less vehicles than* $\mathscr{V}_{best}$ **then**
        $Iter \leftarrow 0$;
        $\mathscr{V}_{best} \leftarrow \mathscr{V}$;

4.2 Bi-Objective Dial-a-Tour Problem

In the context of the single objective DATP, which aims to minimize the number of vehicles used, it is crucial to consider a second objective related to service quality provided to tourists. Tourists also value the overall experience, including minimizing their waiting times. Long waiting periods can negatively influence how the service provided is perceived. Incorporating an additional objective that prioritizes reducing waiting times can improve customer satisfaction. This can lead to better service provision and take a more comprehensive approach to the issue, ultimately contributing to the sustainability and success of the tour-operating business. Hence, a bi-objective variant of the problem is detailed in this section.

4.2.1 Mixed Integer Programming Model

This bi-objective variant seeks to minimize the waiting times and vehicle usage necessary to complete the tours. Then, in addition to Constraints (4.2-4.12) the problem is modeled as follows:

$$\min\ z_1 = W_{max} \tag{4.13}$$

$$\min\ z_2 = \sum_{k \in K} \sum_{i \in N \setminus \{,d\}} x_{koi} \tag{4.14}$$

$$\text{s.t. } W_{max} \geq w_g \qquad \forall g \in G \tag{4.15}$$

$$w_g \leq 1 - \frac{s_{d_g} - s_{o_g} - \underline{L}_g}{\underline{L}_g} \qquad \forall g \in G \tag{4.16}$$

$$(4.2) - (4.12).$$

where w_g denotes the waiting time for each group $g \in G$ performing its tour. The Objective Function (4.1) is set as the second objective z_2. The value $W_{\max}$ in the first objective z_1 represents the maximum waiting time for all groups. Minimizing this value is expected to minimize waiting times for all groups in the problem. Constraints (4.15-4.16) are used to compute waiting times for vehicles.

4.2.2 Non-dominated Sorted Genetic Algorithm II

To tackle the bi-objective DATP, an NSGA-II algorithm [19] is proposed. Algorithm 3 illustrates this procedure. The algorithm uses a unique non-dominated sorting technique to categorize solutions into different fronts or layers, where each front represents a set of solutions that are not dominated by any other solution in the same front. This sorting is crucial for maintaining diversity and providing a trade-off range of solutions.

Algorithm 3: NSGA-II Pseudocode

Data: Instance of the DATP
Result: Set of non-dominated solutions $\mathscr{F}$;
Generate a random population P of size N;
Set $gen \leftarrow 1$;
while $gen \leq$ *maximum generations* **do**
    Let Q be the offspring population;
    while $|Q| < N$ **do**
        Randomly select two parent individuals I_1 and I_2 from P;
        Apply crossover to create two offspring $\mathscr{O}$ from I_1 and I_2;
        Apply mutation to $\mathscr{O}$;
        $Q \leftarrow Q \cup \mathscr{O}$;
    Let R be the combined population $P \cup Q$;
    Perform non-dominated sorting on P to assign solutions to fronts $\mathscr{T}_i$;
    Calculate crowding distance for each I in each $\mathscr{T}_i$;
    Let P' be the next generation of P;
    if $|\mathscr{T}_1| \geq N$ **then**
        Let $\mathscr{R} \in R$ be the first N individuals with the least crowding distance of $\mathscr{T}_1$;
        $P' \leftarrow \mathscr{R}$;
    else
        $P' \leftarrow \mathscr{T}_1$;
        while $|P'| < N$ **do**
            Let $\mathscr{M}$ be the remaining members with the least crowding distance of the next consecutive front $\mathscr{T}_{i+1}$;
            $P' \leftarrow P' \cup \mathscr{M}$;
    $P \leftarrow P'$;
    $gen \leftarrow gen + 1$;

In this genetic algorithm, the encoding process involves representing a solution as a sequence of genes. Each gene corresponds to a pick-up or delivery operation at a spot. When a new individual is created, its genes are initialized and will represent the order in which tourist groups are scheduled in the vehicle routes.

Decoding a solution involves interpreting the sequence of genes to construct a feasible solution for the problem. Here, operations are assigned to the solution based on the genes. The fitness is calculated based on the number of vehicles and the waiting time of the groups in the solution.

When the algorithm creates an initial population, it sorts solutions using a non-dominating sorting, assigning rankings in different PFs based on their dominance.

Then, the crowding distance helps to differentiate those in the same PF.

The crossover technique combines genetic information from two parent solutions to create new candidate solutions (offspring). This is done by selecting a segment of genes between randomly chosen minimum and maximum positions. The aim is to ensure the offspring inherit continuous segments from both parents to preserve feasible routes. For this operator, two positions are randomly selected, genes between these positions are directly copied from one parent, and the remaining positions are filled with genes from the other parent, ensuring no duplication and maintaining relative order.

Once the offspring are created for introducing variability and exploring new regions of the solution space, the mutation operator reverses segments of the gene sequence through a center inverse mutation strategy by reversing parts of the solution sequence, choosing a random point k with the sequence of genes. The sequence before k and the sequence after k are reversed separately.

After the crossover and mutation operators occur, the offspring generate a combined population. After sorting the solutions, the method reduces the population size by removing solutions from the back of the sorted population list until it reaches the set population size.

4.2.3 Constraint Programming Model 1

In this formulation, sp_g is an interval variable indicating the span over the sequence of operations of the group $g \in G$, sv_k is another interval variable indicating the span over the operations of vehicle $k \in K$. In addition, o_i is an interval variable that represents the operation $i \in N$, and a_{ik} is an optional interval variable that denotes the alternative of operating $i \in N$ with vehicle $k \in K$. Furthermore, c_g is a variable indicating the proportion of waiting time for operations of the group $g \in G$.

$$Min\ Z_1 = \max_{g \in G}\{c_g\} \tag{4.17}$$

$$Min\ Z_2 = \sum_{k \in K} PresenceOf(s_k) \tag{4.18}$$

subject to:

$$span(visitalt_{ik} \quad | \quad i \in N) = s_k \qquad \forall k \in K \tag{4.19}$$

$$alternative(visit_i, \{visitalt_{ik} \quad | \quad \forall k \in K\}) \qquad \forall i \in N \qquad (4.20)$$
$$noOverlap(\{visitalt_{ik} \quad | \quad \forall i \in N\}, M) \qquad \forall k \in K \qquad (4.21)$$
$$endBeforeStart(visit_i, visit_j, l_{ij}) \qquad \forall (i,j) \in N \times N \qquad (4.22)$$
$$PresenceOf(visitalt_{ik}) \Rightarrow PresenceOf(visitalt_{jk}) \qquad \forall (i,j) \in E_g, \forall g \in G, \ \forall k \in K \qquad (4.23)$$
$$\sum_{i \in N^+} step(visitalt_{ik}, q_i) - \sum_{j \in N^-} step(visitalt_{jk}, q_j) \leq Q \qquad \forall k \in K \qquad (4.24)$$
$$PresenceOf(visit_i) = true \qquad \forall i \in N \qquad (4.25)$$

The Objective Function (4.17) tries to minimize the waiting times for groups. An additional objective function (4.18) seeks to reduce the quantity of vehicles being utilized. Constraints (4.19) assign span for vehicles from the first to the last visited nodes for each vehicle. Constraints (4.20) assign operations to exactly one vehicle. Constraints (4.21) ensure no overlapping in the sequence of operations. Constraints (4.22) define the minimum time lag in precedence constraints to express staying times in the visits at locations before starting the subsequent operations. Constraints (4.23) Make sure that each pickup and its related delivery task is carried out by the same vehicle, and Constraints (4.24) avoid exceeding vehicle capacity. Constraints (4.25) state that every pick-up and delivery operation must be carried out. We will refer to this model as CP1.

4.2.4 Constraint Programming Model 2

In an additional refinement of modeling the DATP, a second CP model, known as CP2, is introduced. Unlike the CP1 model, which is based on the MILP Model and duplicates each node, representing both delivery and pick-up operations at the same location, handling them as instantaneous with a setting the length of 0, the CP2 adopts a distinct strategy. A set of requests $R = \bigcup_{g \in G} \{r = (i,j) \ : \ (i,j) \in E_g \ and \ i \in N^+\}$ is defined, and its elements are associated with 2-tuples of successive (pick-up, delivery) operations from all groups.

For a given request $r \in R$, the associated pick-up and delivery nodes are denoted as r_+ and r_-, respectively. Additionally, $R_g \subset R$ refers to the subset of requests from the group $g \in G$.

The CP model has the following variables:

z_r: index of the vehicle performing request $r \in R$
$request_r$: interval variable which represents request $r \in R$
$request_{rk}$: interval variable which represents request $r \in R$ carried out by vehicle $k \in K$

Then, in addition to the Objective Function (4.17) it can be modeled as follows:

$$Min\ Z_2 = \max_{r \in R}\{z_r\} \tag{4.26}$$

subject to:

$$z_r = \sum_{k \in K} k \times PresenceOf(requestalt_{rk}) \qquad \forall r \in R \tag{4.27}$$

$$\begin{aligned} z_r = z_{r'} \Rightarrow & |startOf(request_r) - endOf(request_{r'})| \geq l_{r_+ r'_-} \vee \\ & |endOf(request_r) - startOf(request_{r'})| \geq l_{r_- r'_+} \vee \\ & |startOf(request_i) - startOf(request_j)| \geq l_{r_+ r'_+} \vee \\ & |endOf(request_i) - endOf(request_j)| \geq l_{r_- r'_-} \end{aligned} \qquad \forall (r, r') \in R^2 \tag{4.28}$$

$$alternative(request_r, \{requestalt_{rk} \quad | \ \forall k \in K\}), \qquad \forall r \in R \tag{4.29}$$

$$endBeforeStart(request_r, request_{r'}, l_{r_- r'_+}) \qquad \forall g \in G, \forall (r, r') \in R_g^2,\ (r_-, r'_+) \in E_g \tag{4.30}$$

$$\sum_{g \in G} \sum_{r \in R_g} pulse(request_{rk}, w_g) \leq Q \qquad \forall k \in K \tag{4.31}$$

The objective function (4.26) corresponds to the minimization of vehicle usage. Constraints (4.27) indicate the assignment of vehicle requests. Constraints (4.28) stands for time consistency between consecutive requests, so the minimum travel time is respected. Constraints (4.29) indicate that precisely one vehicle carries out a request. Constraints (4.30) indicate the visit precedence considering the duration of stay at each spot. Constraints (4.31) make sure that the total loading by each vehicle stays within its capacity limit.

4.2.5 Augmented ϵ-Constraint Method

The augmented ϵ-Constraint method (AUGMECON) proposed by Mavrotas [44], as a variant of the ϵ-Constraint [29], designed to eliminate the generation of weakly Pareto optimal solutions is used to obtain our non-dominated solutions of the Pareto front. The ϵ-constraint is one of the most widespread methods in the literature for obtaining non-dominated solutions for multi-objective problems. This method proposed by Haimes [29], is based on optimizing one objective function, whereas the others are set as constraints of the model, and using parametrical variations in the right-hand-side of the now constrained objective functions, a Pareto front is obtained.

Generally, the ϵ-Constraint method involves a technique where the number of objective functions is reduced by converting them into constraints of the problem.

To find our non-dominated solutions through the AUGMECON method, our model is reformulated, maintaining Constraints 4.19-4.25 as follows:

$$MinZ_1 = \max_{g \in G}\{c_g\} - eps \times s_2/r_2 \tag{4.32}$$

subject to:

$$\sum_{k \in K} PresenceOf(s_k) + s_2 = \epsilon \tag{4.33}$$

$$(4.19) - (4.25). \tag{4.34}$$

The *eps* value represents a small number, set to 10^{-3} as suggested by Mavrotas [44], s_2 is a surplus variable, and r_2 is the range of the second objective function. The ϵ parameter is varied across the range r_2 to generate the PF points. Those ϵ values vary from 1 (minimum amount of used vehicles possible) to a maximum value computed by a simple greedy procedure where no groups have to wait, assigning the groups iteratively until no more can be added, then another vehicle is enabled. This way, the result acts like an upper bound, and the worst values in the number of vehicles used are not allowed. For each value of ϵ, the single objective problem of minimizing waiting times by the function (4.32) and subject to the vehicle usage constraint 4.33 is solved. This involves solving a series of constrained optimization problems, each with a different ϵ value. The resulting solutions will provide different trade-offs between waiting times and vehicle usage. Each solution generated corresponds to a specific point on the PF. The process continues until the entire feasible range of ϵ values is explored, ensuring that all potential trade-offs are considered.

4.3 Summary

In this chapter, the solution methods for each of the variants of the DATP are presented. For the single objective, a MIP model is detailed. Then, a constructive heuristic procedure is described as an alternative to deal with the model size. The procedure is based on the insertions of pickup and delivery pairs. In another step, a local search procedure is considered to help improve the previously generated solutions.

Additionally, the problem has been extended to a bi-objective variant, along with the proposal of several solution methods. Another MIP model is adapted, an NSGA-II adapted to this kind of application is presented, and two CP models are proposed, which are applied using an ϵ-Constraint method to generate the Pareto Fronts.

Chapter 5

Results

5.1 Instance Generator

To the best of our knowledge, no benchmark can be found in the literature that can be adapted to our problem; therefore, a new set of instances has been proposed. To create realistic instances, tours offered by various Mexican agencies of Cancún and some nearby zones across the Yucatan Peninsula have been analyzed, extracting full-day tours and randomly assigning groups to existing tours.

Additionally, the time spent at each attraction ranges from 60 to 210 minutes, and vehicles have a capacity of 20 passengers. Tourist groups range in size from 1 to 9 individuals. By varying the seed in the generation process, we ensured diversity in the layouts, affecting the duration of stays at attractions, travel times, group sizes, and starting locations. This benchmark has been tested for the single-objective variant of the problem.

In the extension for the bi-objective variant of this work, and using this information, a random generator has been developed to create five classes of instances: Class 0, Class 1, Class 2, Class 3, and Class 4. Each class consists of ten instances with groups of sizes 5, 10, 30, 50, 70, and 90. The primary distinction among the classes lies in the diversity of tours offered. In Class 0, only one tour is available; therefore, all tourists select it, concluding that there will be no diversity in the tours. In Class 1, the number of different available tours is approximately 25% of the total number of tourist groups; in Class 2, this increases to around 50%; in Class 3, to around 75%; and in Class 4, there are as many tours as tourist groups.

5.2 Experimental Results

The mathematical models and algorithms were implemented using C++ as the programming language. The computational experiments were conducted on an Apple iMac with a 2.3GHz Intel Core i5 processor and 8 GB of RAM. The optimization models were solved using IBM ILOG CPLEX Optimization Studio and CP Optimizer, version 22.11.

5.2.1 Single-Objective DATP

5.2.1.1 Solver Comparisons

Tests are conducted using the proposed benchmark to assess the performance of our proposed methods. Our analysis included the outcomes of both the MIP model and the constructive heuristic. Furthermore, the solution generated by the constructive heuristic has been utilized as an initial input for the MIP model and subsequently for initializing the local search approach. The CPLEX solver and the other algorithms were terminated after a one-hour runtime, with the best solutions identified based on fitness (i.e., the number of vehicles utilized) being reported.

Each solution generated by the methodologies comprises a sequence of pick-up and delivery operations to satisfy the tour requests made by tourists. This sequence is illustrated in Figure 5.1, which presents an example solution depicting how two vehicles manage the requests from five groups.

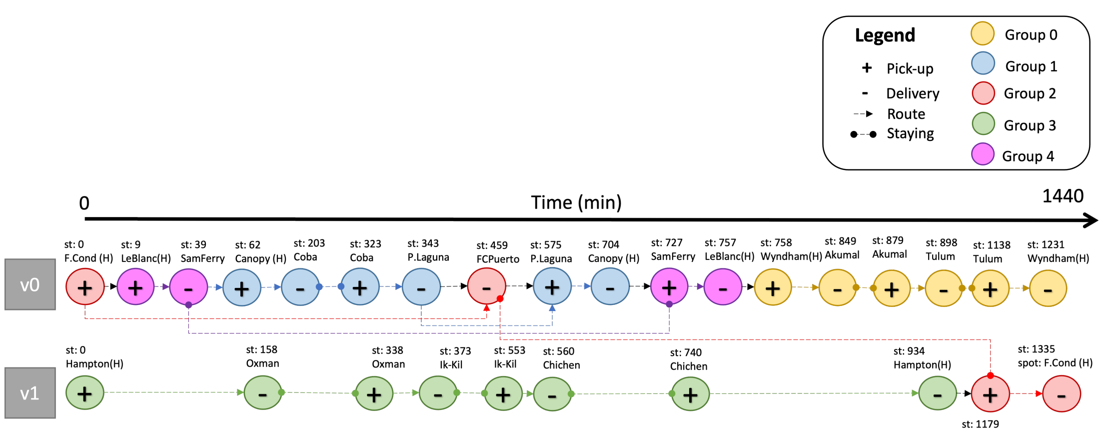

Figure 5.1: Schedule Example

General results are detailed in Table 5.1, comparing the exact and heuristic approaches. It can be noticed that the MIP model demonstrated its inability to find solutions for more than five groups within the specified one-hour timeframe, making it impractical for larger group sizes. Nevertheless, to enhance the initial solutions provided by the constructive method, we conducted experiments that extended to scenarios involving ten groups. We fed the MIP model with an initial solution and evaluated potential enhancements in terms of vehicle utilization. A decrease in the number of vehicles was only noted in a single instance, specifically for a scenario involving five groups.

Table 5.1: Results of the MILP method and the Local Search approach

	Feasible Solutions		Optimals		Avg. Gap		CPU Time	
Number of groups	**MILP**	**Local Search**	**MILP**	**Local Search**	**MILP**	**Local Search**	**MILP**	**Local Search**
5	10	10	2	3	0.400	0.350	2881.40	0.01
10	-	10	-	0	-	0.585	3600.00	0.01
20	-	10	-	0	-	0.715	3600.00	0.10
30	-	10	-	0	-	0.656	3600.00	0.11
40	-	10	-	0	-	0.604	3600.00	2.40
50	-	10	-	0	-	0.639	3600.00	3.80
60	-	10	-	0	-	0.632	3600.00	6.50
70	-	10	-	0	-	0.623	3600.00	10.80
80	-	10	-	0	-	0.618	3600.00	13.30
90	-	10	-	0	-	0.621	3600.00	22.50
100	-	10	-	0	-	0.616	3600.00	24.50

(-) The solver does not find solution

The constructive heuristic successfully generated solutions for all tourist groups, expanding the analysis to all benchmark instances. Regarding vehicle usage, the impact of incorporating the local search approach to the initial solution obtained by the constructive method has been examined. This influence is analyzed in Figure 5.2, illustrating a notable improvement when employing the local search strategy. The horizontal axis of the boxplot represents instances grouped by tourist group sizes (10 instances per group), while the vertical axis shows the number of vehicles obtained by the analyzed methods. It indicates that as the number of groups increases, the gap between the local search and the constructive method increases, with reductions in the vehicles required to fulfill all tourist requests reaching nearly 40%. This demonstrates the effectiveness of the local search in enhancing the initial solution.

Finally, it is found pertinent to investigate how the number of requests influences the corresponding number of vehicles required to complete all operations. Figure 5.3 illustrates the correlation between the total number of requests and the corresponding number of vehicles needed, categorized by groups. The figure shows a scatter plot. The horizontal axis represents the total number of requests present

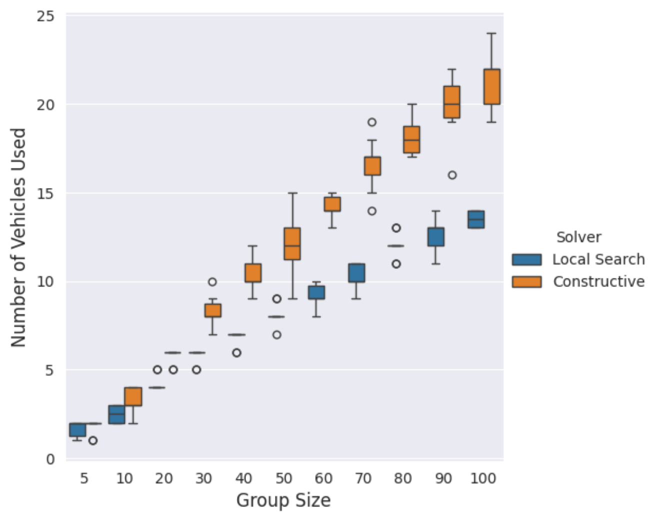

Figure 5.2: Effect of the LS on the Constructive Method to reduce vehicle usage.

in all instances of the benchmark, while the vertical axis represents the number of vehicles used to handle all the requests. Instances are grouped by color to differentiate between their different group sizes. This graph offers a visual reference for estimating the approximate number of vehicles necessary to handle a specific volume of tourism requests, helping in decision-making processes.

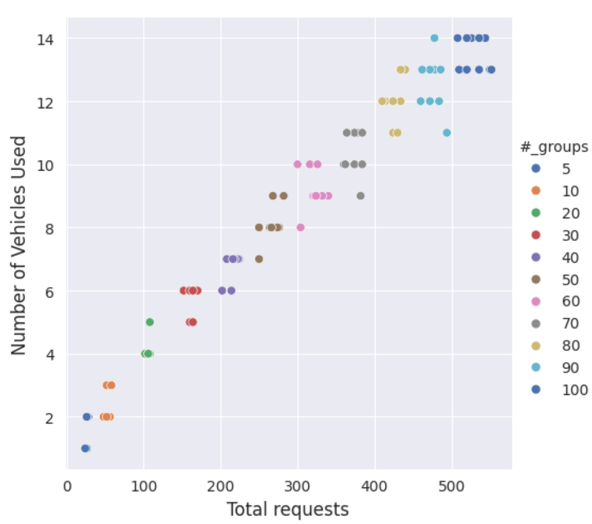

Figure 5.3: Total number of requests handled by vehicles

5.2.2 Bi-Objective DATP

5.2.2.1 Solver Comparisons

A comparison of different metrics for multi-objective algorithms is presented, aiming to provide a comprehensive analysis of their effectiveness in evaluating the performance of such algorithms. The metrics under consideration include Hypervolume and K-distance (spacing). Each metric is examined in terms of its ability to capture key aspects of solution quality, such as diversity and convergence, across a range of the tested instances. The CPU time taken by each solver to create the corresponding Pareto Front (PF) is also measured. The findings describe the strengths and limitations of each metric in the field of multi-objective optimization.

The Hypervolume is widely successful and popular because it considers both proximity and diversity while strictly adhering to Pareto principles [7]. This means that when one PF approximation dominates another, the hypervolume of the dominant one is always greater than the other. To assess the spread of solutions, the K-distance metric was used to measure the extent of the spread in a computed PF approximation. In addition, the Overall Non-dominated Vector Generation (ONVG) metric has been included. This metric is considered important for finding approximations of PFs with many points, as it provides the decision-maker with a wide range of options for making the final decision.

Three solution methods are evaluated in the solver comparisons for these metrics: The PFs obtained using NSGA-II and the AUGMECON procedure using both CP1 and CP2 models.

Overall Non-dominated Vector Generation Figure 5.4 shows the results of ONVG obtained by the different methods. This metric represents the cardinality (number of points) of the PF approximations generated by the different algorithms.

In this boxplot, the horizontal axis represents instances grouped by the different tourist group sizes. Each group has 50 cases (10 instances per each of the five classes). The vertical axis shows the number of points obtained by the analyzed methods. In this case, the AUGMECON-CP1 algorithm provides more points than the other AUGMECON-CP2 and the NSGA-II algorithms, finding up to 13 points as average for bigger instances. However, as this metric by itself sometimes does not refer to the quality of a non-dominated set, it might be useful to analyze algorithms from more perspectives.

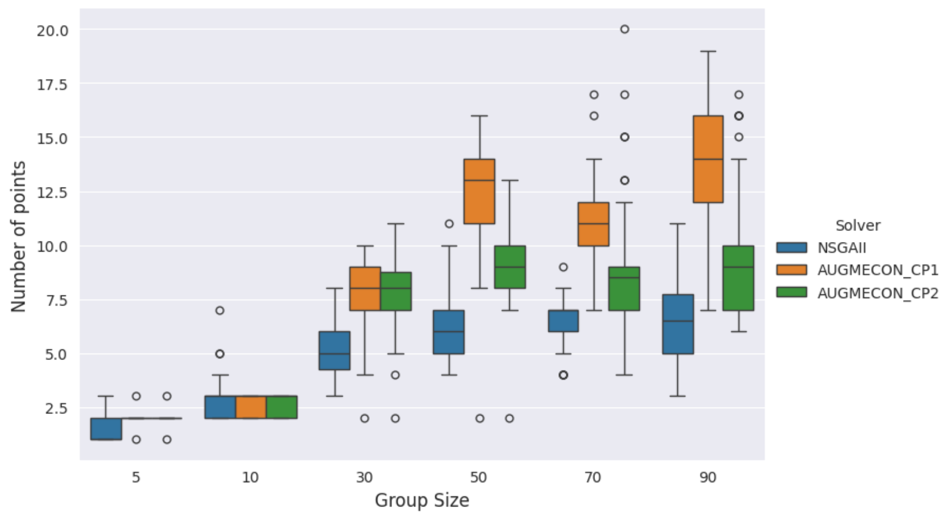

Figure 5.4: Comparison of the ONVG of the different solution methods

Hypervolume During the evaluation process, it has been encountered a notable number of missing (NA) values in the results for this metric. The reason is that to compute this metric, two or more elements in the solution vector must be present [7].

Analyzing the results (see Appendix B), one can say that for smaller instances, the NSGA-II method efficiently finds a very strong solution early in the process, which dominates all other potential solutions, preventing the formation of a diverse PF for other trade-offs between objectives in those instances.

To conduct a fair and reliable comparison of the methods based on the hypervolume metric, it is essential to focus on groups of instances where all methods provide sufficient data for analysis. Specifically, in Group 5, 70% of the instances (35 out of 50) produced NA values for the NSGA-II. Given this high percentage, instances for this group have been excluded from further analysis using this metric when comparing the generated PFs.

Figure 5.5 then, shows the comparison of the results of the hypervolume metric obtained by the approximations of the PFs generated by the different solvers across different numbers of groups. The horizontal axis of the boxplot represents instances grouped by tourist group sizes (each group with 50 instances), while the vertical axis shows the number of vehicles obtained by the analyzed methods. Overall, it appears that the PF generated by the CP1 method tends to achieve higher mean hypervolume values compared to the other two in most of the group sizes. This suggests that the PF of the AUGMECON-CP1 may be more effective in exploring the solution space and generating diverse solutions. The majority of fluctuations are devised with a smaller number of groups, but the algorithms seem relatively consistent across the bigger group sizes, with levels of performance between 0.7 and 0.9 regardless of the number of groups in the instances.

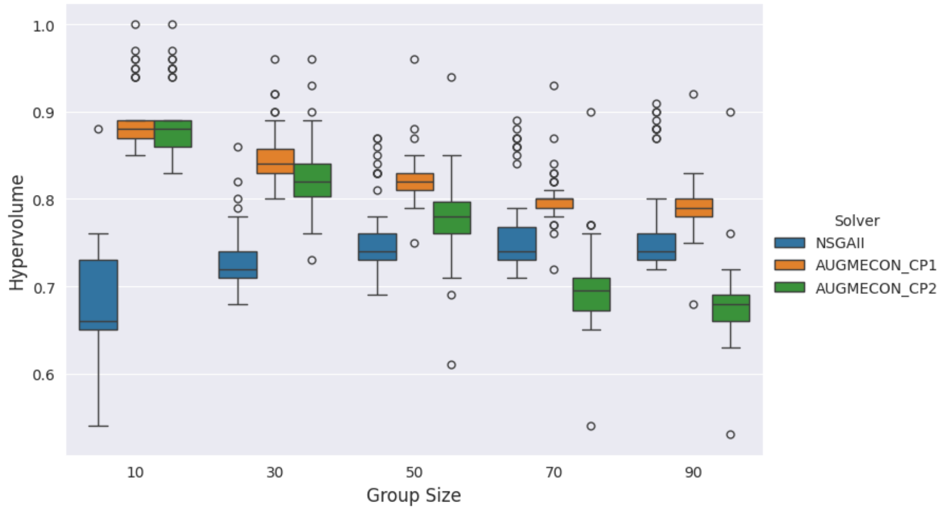

Figure 5.5: Comparison of Hypervolume of the different solution methods

K-distance Figure 5.6 shows results after applying the K-Distance with $k = 2$ metric. As this metric considers the distance between a point and its closest neighbor, it indicates the degree of spread in the solutions of the Pareto set approximations. In this boxplot, the horizontal axis represents instances grouped by the different tourist group sizes, each group with 50 instances. The vertical axis shows the values obtained by the analyzed methods for this metric. The PFs of the AUGMECON-CP2 method show more separated solution points, indicating wider ranges in their sets. The other two algorithms show similar results with nearer points in most solutions of the analyzed instances.

A comparison of the CPU Times of the different solution methods is shown in Figure 5.7. The NSGA-II method takes the set limit of 3600 seconds for generating the approximated PF. On the other hand, for the CP1 method, stability of around 3600 seconds can be observed when dealing with instances of more than 40 groups. In addition, with the CP2 method, it can be seen an increasing trend as the number of groups grows.

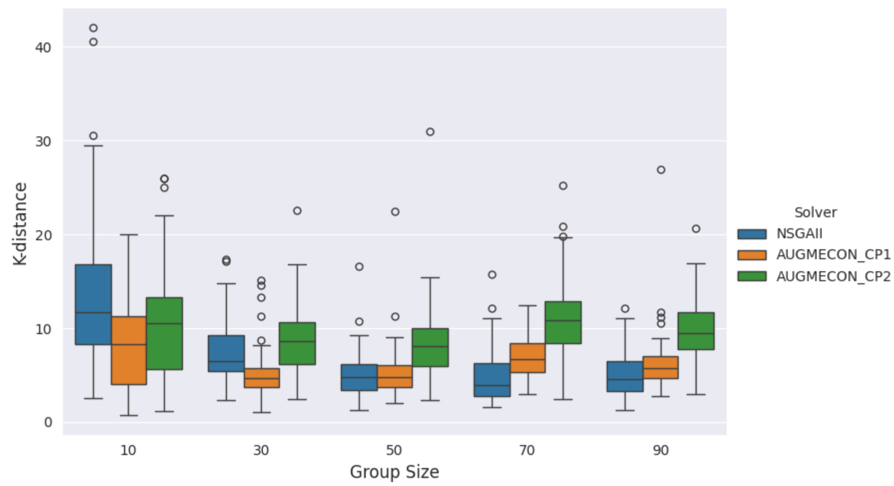

FIGURE 5.6: Comparison of K-Distance metric with different solution methods

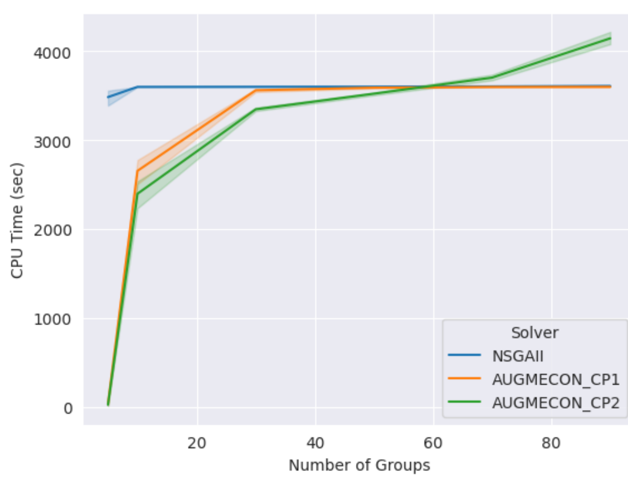

FIGURE 5.7: Comparison of CPU Time with different solution methods

Chapter 6

Conclusions

In this work, an optimization problem devised in the tourism industry has been introduced. In the literature review, it has been found that it strongly resembles a well-studied problem: *the Dial-a-Ride Problem*. Multiple solution approaches and variants have been proposed to solve it, highlighting heuristics and metaheuristics-based techniques due to the nature of the problem. Our objective with this work is to contribute to the optimization of the route planning process in the tourism industry, specifically in the scheduling of vehicle visits to tourist attractions.

This dissertation attempts to analyze the problem from two perspectives, with a single-objective framework and then as a multi-objective problem. For the single-objective problem, two solution methods have been proposed for this challenge: a MIP model and a heuristic. The MIP model struggles to find solutions for more than 5 groups within an hour timeframe. Hence, the heuristic approach is proposed to deal with this limitation, and an LS is implemented to further improve these solutions. Results demonstrate a better performance for the LS based on the tests on the proposed benchmark from a real-case scenario. As a result, it can be said that for real-world problems, the LS approach can efficiently handle up to 100 groups of customers within 25 seconds at most. In addition, it can reduce the number of vehicles used to perform all requests by up to 40% in some scenarios. Finally, a visual tool is included to assist tour operators in determining the number of vehicles required based on a specific volume of requests.

Our work also aims to improve the overall tourism experience. Therefore, it introduces a variant where waiting times are considered as a minimization objective, becoming a bi-objective problem. From this perspective, three solution methods have been proposed as other contributions. First, two approximate Pareto frontiers are constructed using the AUGMECON method based on CP models. Second, an NSGA-II adapted to this type of on-demand transportation problem. Several metrics have been computed to compare these methods. Regarding the hypervolume, on average, the first CP model (AUGMECON-CP1) tends to achieve higher values. In terms of the number of generated solutions, the AUGMECON-CP1 algorithm provides more solutions than the other methods. with values up to 6 points as average in the greater instances. This suggests that this PF offers higher-quality solutions than the other two methods, effectively balancing the objectives, dominating more of the objective space, and providing more diverse options for decision-makers. The better performance of AUGMECON-CP1 over AUGMECON-CP2 can be attributed to the efficiency of the constraints used in its formulation. In particular, AUGMECON-CP2 relies on more complex conditional constraints for time consistency and vehicle operations, which introduce additional computational effort. By using no overlap constraints to manage vehicle scheduling and precedence constraints for group sequences, the AUGMECON-CP1 model reduces the need for complex conditional logic.

6.1 Future work

Future research could involve testing these methods with heterogeneous vehicle fleets for both single and bi-objective perspectives. Additionally, for the single objective problem, studying the influence of the number of total requests for vehicle usage can be beneficial for tour operators, enabling them to estimate the number of vehicles needed for a given planning schedule through simulations. Finally, for the bi-objective perspective, the necessity of performing an in-depth analysis of the trade-off between the objectives in real-world cases. In addition, the study of the significance of preferring to have a larger set of solution options can influence the choice of an algorithm when addressing the problem.

Appendix A

Results of the Single-Objective Solution Methods

A.1 Mixed Integer Programming Model

Table A.1: Results of the MIP Model

Instance	Fitness	Gap	CPU_time (sec)	#_spots	#_nodes	#_groups	#_tours	VCap	LB	Peek_Load	Av_Load
Ca_5_0	2	0.5	3600	30	28	5	22	20	1	18	6.12
Ca_5_1	2	0.5	3600	29	26	5	22	20	1	16	5.58
Ca_5_2	2	0.5	3600	29	26	5	22	20	1	15	4.05
Ca_5_3	2	0.5	3600	30	28	5	22	20	1	13	6.69
Ca_5_4	1	0.0	1	30	26	5	22	20	1	20	13.59
Ca_5_8	2	0.5	3600	30	24	5	22	20	1	18	7.16
Ca_5_5	2	0.5	3600	30	26	5	22	20	1	18	7.57
Ca_5_6	2	0.5	3600	30	26	5	22	20	1	19	8.81
Ca_5_7	1	0.0	13	29	24	5	22	20	1	19	10.95
Ca_5_9	2	0.5	3600	30	28	5	22	20	1	14	4.18

A.2 Mixed Integer Programming Model + Greedy Method

Table A.2: Results of the MIP Model with the Greedy Constructive Method

Instance	Fitness	Gap	CPU_time (sec)	#_spots	#_nodes	#_groups	#_tours	VCap	LB	Peek_Load	Av_Load
Ca_5_0	2	0.50	3600	30	28	5	22	20	1	17	3.83
Ca_5_1	2	0.50	3600	29	26	5	22	20	1	13	3.76
Ca_5_2	2	0.50	3600	29	26	5	22	20	1	15	3.05
Ca_5_3	2	0.50	3600	30	28	5	22	20	1	16	4.83
Ca_5_4	1	0.01	0	30	26	5	22	20	1	20	5.27
Ca_5_5	2	0.50	3600	30	26	5	22	20	1	15	4.82
Ca_5_6	2	0.50	3600	30	26	5	22	20	1	14	5.25
Ca_5_7	1	0.01	0	29	24	5	22	20	1	15	6.20
Ca_5_8	1	0.01	216	30	24	5	22	20	1	19	9.22
Ca_5_9	2	0.50	3600	30	28	5	22	20	1	11	1.19
Ca_10_0	4	0.75	3600	32	52	10	22	20	1	20	4.08
Ca_10_1	4	0.75	3600	32	56	10	22	20	1	16	3.89
Ca_10_2	3	0.67	3600	33	52	10	22	20	1	17	5.28
Ca_10_3	3	0.67	3600	33	56	10	22	20	1	19	5.81
Ca_10_4	4	0.75	3600	33	58	10	22	20	1	20	3.37
Ca_10_5	3	0.67	3600	34	52	10	22	20	1	17	4.48
Ca_10_6	3	0.67	3600	34	54	10	22	20	1	20	5.85
Ca_10_7	3	0.67	3600	33	54	10	22	20	1	19	4.86
Ca_10_8	2	0.50	3600	32	48	10	22	20	1	20	7.59
Ca_10_9	4	0.75	3600	34	58	10	22	20	1	16	3.36

A.3 Greedy Method

Table A.3: Results of the Greedy Method

Instance	Fitness	Gap	CPU_time (sec)	#_spots	#_nodes	#_groups	#_tours	VCap	LB	Peek_Load	Av_Load
Ca_5_0	2	0.50	0	30	28	5	22	20	1	17	3.83
Ca_5_1	2	0.50	0	29	26	5	22	20	1	13	3.76
Ca_5_2	2	0.50	0	29	26	5	22	20	1	15	3.05
Ca_5_3	2	0.50	0	30	28	5	22	20	1	16	4.83
Ca_5_4	1	0.00	0	30	26	5	22	20	1	20	5.27
Ca_5_5	2	0.50	0	30	26	5	22	20	1	15	4.82
Ca_5_6	2	0.50	0	30	26	5	22	20	1	14	5.25
Ca_5_7	1	0.00	0	29	24	5	22	20	1	15	6.20
Ca_5_8	2	0.50	0	30	24	5	22	20	1	10	3.53
Ca_5_9	2	0.50	0	30	28	5	22	20	1	11	1.19
Ca_10_0	4	0.75	0	32	52	10	22	20	1	20	4.08
Ca_10_1	4	0.75	0	32	56	10	22	20	1	16	3.89
Ca_10_2	3	0.67	0	33	52	10	22	20	1	17	5.28
Ca_10_3	3	0.67	0	33	56	10	22	20	1	19	5.81
Ca_10_4	4	0.75	0	33	58	10	22	20	1	20	3.37
Ca_10_5	3	0.67	0	34	52	10	22	20	1	17	4.48
Ca_10_6	3	0.67	0	34	54	10	22	20	1	20	5.85
Ca_10_7	3	0.67	0	33	54	10	22	20	1	19	4.86
Ca_10_8	2	0.50	0	32	48	10	22	20	1	20	7.59
Ca_10_9	4	0.75	0	34	58	10	22	20	1	16	3.36
Ca_20_0	6	0.83	0	36	102	20	22	20	1	20	4.61
Ca_20_1	6	0.83	0	36	108	20	22	20	1	20	5.29
Ca_20_2	5	0.80	0	36	108	20	22	20	1	20	4.64
Ca_20_3	6	0.83	0	36	106	20	22	20	1	19	5.56
Ca_20_4	6	0.83	0	36	108	20	22	20	1	20	5.03
Ca_20_5	5	0.80	0	36	102	20	22	20	1	20	5.57
Ca_20_6	6	0.83	0	36	104	20	22	20	1	20	4.56
Ca_20_7	6	0.83	0	36	108	20	22	20	1	20	4.40
Ca_20_8	6	0.83	0	36	106	20	22	20	1	19	5.21
Ca_20_9	6	0.83	0	36	108	20	22	20	1	20	5.32
Ca_30_0	8	0.88	0	36	164	30	22	20	1	19	4.54
Ca_30_1	10	0.90	0	36	152	30	22	20	1	20	4.66
Ca_30_2	9	0.89	0	36	164	30	22	20	1	20	5.28
Ca_30_3	8	0.88	0	36	160	30	22	20	1	19	5.70
Ca_30_4	8	0.88	0	36	164	30	22	20	1	20	5.40
Ca_30_5	9	0.89	0	36	170	30	22	20	1	20	5.62
Ca_30_6	8	0.88	0	36	164	30	22	20	1	20	6.69
Ca_30_7	8	0.88	0	36	162	30	22	20	1	20	6.61
Ca_30_8	8	0.88	0	36	162	30	22	20	1	20	4.84
Ca_30_9	7	0.86	0	36	160	30	22	20	1	19	5.71
Ca_40_0	9	0.89	0	36	214	40	22	20	1	20	5.20

Table A.3 continued from previous page

Instance	Fitness	Gap	CPU_time (sec)	#_spots	#_nodes	#_groups	#_tours	VCap	LB	Peek_Load	Av_Load
Ca_40_1	11	0.91	0	36	202	40	22	20	1	20	5.55
Ca_40_2	10	0.90	0	36	208	40	22	20	1	20	6.37
Ca_40_3	12	0.92	0	36	216	40	22	20	1	20	5.22
Ca_40_4	9	0.89	0	36	216	40	22	20	1	20	6.48
Ca_40_5	12	0.92	0	36	224	40	22	20	1	20	5.90
Ca_40_6	10	0.90	0	36	216	40	22	20	1	20	5.85
Ca_40_7	11	0.91	0	36	210	40	22	20	1	20	5.95
Ca_40_8	10	0.90	0	36	218	40	22	20	1	20	6.37
Ca_40_9	10	0.90	0	36	222	40	22	20	1	20	5.90
Ca_50_0	9	0.89	0	36	250	50	22	20	1	20	5.69
Ca_50_1	11	0.91	0	36	266	50	22	20	1	20	6.03
Ca_50_2	13	0.92	0	36	276	50	22	20	1	20	5.61
Ca_50_3	11	0.91	0	36	266	50	22	20	1	20	6.24
Ca_50_4	12	0.92	0	36	282	50	22	20	1	20	6.55
Ca_50_5	15	0.93	0	36	250	50	22	20	1	20	4.97
Ca_50_6	12	0.92	0	36	268	50	22	20	1	20	6.85
Ca_50_7	13	0.92	0	36	274	50	22	20	1	20	5.95
Ca_50_8	13	0.92	0	36	270	50	22	20	1	20	5.50
Ca_50_9	12	0.92	0	36	264	50	22	20	1	20	6.42
Ca_60_0	14	0.93	0	36	304	60	22	20	1	20	5.45
Ca_60_1	13	0.92	0	36	324	60	22	20	1	20	5.66
Ca_60_2	14	0.93	0	36	340	60	22	20	1	20	6.79
Ca_60_3	13	0.92	0	36	320	60	22	20	1	20	7.05
Ca_60_4	14	0.93	0	36	332	60	22	20	1	20	6.78
Ca_60_5	15	0.93	0	36	300	60	22	20	1	20	5.95
Ca_60_6	15	0.93	0	36	316	60	22	20	1	20	6.52
Ca_60_7	15	0.93	0	36	322	60	22	20	1	20	5.44
Ca_60_8	14	0.93	0	36	326	60	22	20	1	20	6.47
Ca_60_9	14	0.93	0	36	322	60	22	20	1	20	6.29
Ca_70_0	17	0.94	0	36	384	70	22	20	1	20	4.97
Ca_70_1	17	0.94	0	36	384	70	22	20	1	20	6.47
Ca_70_2	14	0.93	0	36	378	70	22	20	1	20	7.80
Ca_70_3	16	0.94	0	36	364	70	22	20	1	20	6.63
Ca_70_4	15	0.93	0	36	360	70	22	20	1	20	7.17
Ca_70_5	19	0.95	0	36	362	70	22	20	1	20	5.89
Ca_70_6	17	0.94	0	36	374	70	22	20	1	20	6.16
Ca_70_7	17	0.94	0	36	374	70	22	20	1	20	5.47
Ca_70_8	16	0.94	0	36	382	70	22	20	1	20	5.68
Ca_70_9	18	0.94	0	36	374	70	22	20	1	20	5.59
Ca_80_0	17	0.94	0	36	430	80	22	20	1	20	5.95
Ca_80_1	17	0.94	0	36	440	80	22	20	1	20	7.16
Ca_80_2	18	0.94	0	36	434	80	22	20	1	20	6.11
Ca_80_3	18	0.94	0	36	410	80	22	20	1	20	6.28
Ca_80_4	19	0.95	0	36	410	80	22	20	1	20	5.87
Ca_80_5	18	0.94	0	36	414	80	22	20	1	20	6.28
Ca_80_6	20	0.95	0	36	424	80	22	20	1	20	6.66
Ca_80_7	18	0.94	0	36	424	80	22	20	1	20	5.79
Ca_80_8	17	0.94	0	36	434	80	22	20	1	20	6.59
Ca_80_9	20	0.95	0	36	434	80	22	20	1	20	5.55
Ca_90_0	19	0.95	0	36	472	90	22	20	1	20	6.72
Ca_90_1	20	0.95	0	36	486	90	22	20	1	20	6.57
Ca_90_2	16	0.94	0	36	460	90	22	20	1	20	7.17

Table A.3 continued from previous page

Instance	Fitness	Gap	CPU_time (sec)	#_spots	#_nodes	#_groups	#_tours	VCap	LB	Peek_Load	Av_Load
Ca_90_3	19	0.95	0	36	484	90	22	20	1	20	6.57
Ca_90_4	21	0.95	0	36	480	90	22	20	1	20	5.90
Ca_90_5	22	0.95	0	36	472	90	22	20	1	20	5.64
Ca_90_6	22	0.95	0	36	462	90	22	20	1	20	6.30
Ca_90_7	20	0.95	0	36	478	90	22	20	1	20	6.01
Ca_90_8	21	0.95	0	36	478	90	22	20	1	20	6.69
Ca_90_9	20	0.95	0	36	494	90	22	20	1	20	5.96
Ca_100_0	20	0.95	0	36	520	100	22	20	1	20	7.05
Ca_100_1	22	0.95	0	36	550	100	22	20	1	20	6.92
Ca_100_2	19	0.95	0	36	510	100	22	20	1	20	6.19
Ca_100_3	19	0.95	0	36	536	100	22	20	1	20	6.86
Ca_100_4	22	0.95	0	36	544	100	22	20	1	20	6.05
Ca_100_5	22	0.95	0	36	520	100	22	20	1	20	6.50
Ca_100_6	23	0.96	0	36	508	100	22	20	1	20	6.87
Ca_100_7	20	0.95	0	36	526	100	22	20	1	20	6.55
Ca_100_8	24	0.96	0	36	536	100	22	20	1	20	5.86
Ca_100_9	22	0.95	0	36	552	100	22	20	1	20	5.92

A.4 Local Search

Table A.4: Results of applying the Local Search

Instance	Fitness	Gap	CPU_time (sec)	#_spots	#_nodes	#_groups	#_tours	VCap	LB	Peek_Load	Av_Load
Ca_5_0	2	0.50	0	30	28	5	22	20	1	17	3.83
Ca_5_1	2	0.50	0	29	26	5	22	20	1	13	3.76
Ca_5_2	2	0.50	0	29	26	5	22	20	1	15	3.05
Ca_5_3	2	0.50	0	30	28	5	22	20	1	16	4.83
Ca_5_4	1	0.00	0	30	26	5	22	20	1	20	5.27
Ca_5_5	2	0.50	0	30	26	5	22	20	1	15	4.82
Ca_5_6	2	0.50	0	30	26	5	22	20	1	14	5.25
Ca_5_7	1	0.00	0	29	24	5	22	20	1	15	6.20
Ca_5_8	1	0.00	0	30	24	5	22	20	1	17	8.74
Ca_5_9	2	0.50	0	30	28	5	22	20	1	11	1.19
Ca_10_0	3	0.67	0	32	52	10	22	20	1	20	4.81
Ca_10_1	3	0.67	0	32	56	10	22	20	1	17	4.04
Ca_10_2	3	0.67	0	33	52	10	22	20	1	17	5.28
Ca_10_3	2	0.50	0	33	56	10	22	20	1	20	7.05
Ca_10_4	3	0.67	0	33	58	10	22	20	1	20	3.27
Ca_10_5	2	0.50	0	34	52	10	22	20	1	20	9.32
Ca_10_6	2	0.50	0	34	54	10	22	20	1	20	8.30
Ca_10_7	2	0.50	0	33	54	10	22	20	1	20	8.13
Ca_10_8	2	0.50	0	32	48	10	22	20	1	20	7.59
Ca_10_9	3	0.67	0	34	58	10	22	20	1	19	3.81
Ca_20_0	4	0.75	0	36	102	20	22	20	1	20	7.12
Ca_20_1	5	0.80	0	36	108	20	22	20	1	20	3.97
Ca_20_2	4	0.75	0	36	108	20	22	20	1	20	5.09
Ca_20_3	4	0.75	0	36	106	20	22	20	1	20	6.93
Ca_20_4	4	0.75	0	36	108	20	22	20	1	18	6.14
Ca_20_5	4	0.75	0	36	102	20	22	20	1	20	6.30
Ca_20_6	4	0.75	0	36	104	20	22	20	1	20	5.36
Ca_20_7	5	0.60	0	36	108	20	22	20	2	20	5.05
Ca_20_8	4	0.50	0	36	106	20	22	20	2	20	7.05
Ca_20_9	4	0.75	1	36	108	20	22	20	1	20	6.65
Ca_30_0	5	0.60	1	36	164	30	22	20	2	19	6.04
Ca_30_1	6	0.67	1	36	152	30	22	20	2	20	5.29
Ca_30_2	6	0.67	1	36	164	30	22	20	2	20	6.56
Ca_30_3	5	0.60	1	36	160	30	22	20	2	20	8.13
Ca_30_4	6	0.67	1	36	164	30	22	20	2	19	6.78
Ca_30_5	6	0.67	2	36	170	30	22	20	2	20	5.94
Ca_30_6	6	0.67	1	36	164	30	22	20	2	20	7.41
Ca_30_7	6	0.67	1	36	162	30	22	20	2	20	7.46
Ca_30_8	6	0.67	1	36	162	30	22	20	2	20	5.89
Ca_30_9	6	0.67	1	36	160	30	22	20	2	20	7.15
Ca_40_0	6	0.67	3	36	214	40	22	20	2	20	7.07

Table A.4 continued from previous page

Instance	Fitness	Gap	CPU_time (sec)	#_spots	#_nodes	#_groups	#_tours	VCap	LB	Peek_Load	Av_Load
Ca_40_1	6	0.67	3	36	202	40	22	20	2	20	9.29
Ca_40_2	7	0.57	1	36	208	40	22	20	3	20	8.15
Ca_40_3	7	0.71	2	36	216	40	22	20	2	20	6.67
Ca_40_4	7	0.57	2	36	216	40	22	20	3	20	8.46
Ca_40_5	7	0.57	2	36	224	40	22	20	3	20	7.38
Ca_40_6	7	0.57	3	36	216	40	22	20	3	20	7.78
Ca_40_7	7	0.57	3	36	210	40	22	20	3	20	7.62
Ca_40_8	7	0.57	3	36	218	40	22	20	3	20	7.75
Ca_40_9	7	0.57	2	36	222	40	22	20	3	20	8.11
Ca_50_0	7	0.71	4	36	250	50	22	20	2	20	6.35
Ca_50_1	8	0.62	4	36	266	50	22	20	3	20	6.24
Ca_50_2	8	0.62	5	36	276	50	22	20	3	20	8.17
Ca_50_3	8	0.62	3	36	266	50	22	20	3	20	8.61
Ca_50_4	9	0.67	4	36	282	50	22	20	3	20	7.75
Ca_50_5	8	0.62	3	36	250	50	22	20	3	20	8.20
Ca_50_6	9	0.67	4	36	268	50	22	20	3	20	7.91
Ca_50_7	8	0.62	4	36	274	50	22	20	3	20	8.57
Ca_50_8	8	0.62	4	36	270	50	22	20	3	20	6.60
Ca_50_9	8	0.62	3	36	264	50	22	20	3	20	6.87
Ca_60_0	8	0.62	5	36	304	60	22	20	3	20	7.59
Ca_60_1	9	0.67	8	36	324	60	22	20	3	20	7.17
Ca_60_2	9	0.67	7	36	340	60	22	20	3	20	7.59
Ca_60_3	9	0.67	6	36	320	60	22	20	3	20	7.52
Ca_60_4	9	0.56	8	36	332	60	22	20	4	20	7.57
Ca_60_5	10	0.60	4	36	300	60	22	20	4	20	8.02
Ca_60_6	10	0.60	10	36	316	60	22	20	4	20	8.07
Ca_60_7	9	0.56	6	36	322	60	22	20	4	20	9.23
Ca_60_8	10	0.70	6	36	326	60	22	20	3	20	7.39
Ca_60_9	9	0.67	5	36	322	60	22	20	3	20	9.56
Ca_70_0	10	0.70	12	36	384	70	22	20	3	20	7.85
Ca_70_1	11	0.64	10	36	384	70	22	20	4	20	7.85
Ca_70_2	11	0.55	15	36	378	70	22	20	5	20	7.65
Ca_70_3	11	0.64	6	36	364	70	22	20	4	20	8.37
Ca_70_4	10	0.60	11	36	360	70	22	20	4	20	8.89
Ca_70_5	10	0.60	9	36	362	70	22	20	4	20	7.68
Ca_70_6	11	0.64	13	36	374	70	22	20	4	20	8.12
Ca_70_7	10	0.70	9	36	374	70	22	20	3	20	7.73
Ca_70_8	9	0.56	15	36	382	70	22	20	4	20	8.98
Ca_70_9	10	0.60	8	36	374	70	22	20	4	20	8.26
Ca_80_0	11	0.64	15	36	430	80	22	20	4	20	6.86
Ca_80_1	13	0.62	13	36	440	80	22	20	5	20	7.61
Ca_80_2	13	0.62	10	36	434	80	22	20	5	20	8.13
Ca_80_3	12	0.58	9	36	410	80	22	20	5	20	9.48
Ca_80_4	12	0.58	17	36	410	80	22	20	5	20	8.33
Ca_80_5	12	0.67	10	36	414	80	22	20	4	20	8.76
Ca_80_6	12	0.58	16	36	424	80	22	20	5	20	8.17
Ca_80_7	11	0.64	12	36	424	80	22	20	4	20	8.23
Ca_80_8	12	0.58	19	36	434	80	22	20	5	20	8.12
Ca_80_9	12	0.67	12	36	434	80	22	20	4	20	8.20
Ca_90_0	12	0.58	23	36	472	90	22	20	5	20	8.71
Ca_90_1	13	0.62	29	36	486	90	22	20	5	20	8.82
Ca_90_2	12	0.67	18	36	460	90	22	20	4	20	8.68

Table A.4 continued from previous page

Instance	Fitness	Gap	CPU_time (sec)	#_spots	#_nodes	#_groups	#_tours	VCap	LB	Peek_Load	Av_Load
Ca_90_3	12	0.58	21	36	484	90	22	20	5	20	9.02
Ca_90_4	13	0.62	15	36	480	90	22	20	5	20	7.78
Ca_90_5	13	0.62	14	36	472	90	22	20	5	20	8.09
Ca_90_6	13	0.62	24	36	462	90	22	20	5	20	8.69
Ca_90_7	13	0.62	23	36	478	90	22	20	5	20	8.18
Ca_90_8	14	0.64	12	36	478	90	22	20	5	20	9.15
Ca_90_9	11	0.64	46	36	494	90	22	20	4	20	8.12
Ca_100_0	13	0.62	32	36	520	100	22	20	5	20	9.09
Ca_100_1	13	0.62	23	36	550	100	22	20	5	20	9.14
Ca_100_2	13	0.62	24	36	510	100	22	20	5	20	8.49
Ca_100_3	13	0.62	39	36	536	100	22	20	5	20	8.66
Ca_100_4	14	0.64	15	36	544	100	22	20	5	20	8.24
Ca_100_5	14	0.57	22	36	520	100	22	20	6	20	8.60
Ca_100_6	14	0.64	18	36	508	100	22	20	5	20	8.48
Ca_100_7	14	0.64	23	36	526	100	22	20	5	20	7.92
Ca_100_8	14	0.57	24	36	536	100	22	20	6	20	8.57
Ca_100_9	13	0.62	25	36	552	100	22	20	5	20	8.28

Appendix B

Results of the Bi-Objective Solution Methods

B.1 NSGA-II

Table B.1: Results of the NSGA-II algorithm

Instance	HyperVol	K-dist(2)	CPU_time	#_spots	#_nodes	#_groups	#_tours	#_points	points
C0_5_0	nan	0	3600	7	40	5	1	1	<2,0>
C0_5_1	0.89	27	2933	7	40	5	1	2	<1,54><2,0>
C0_5_2	0.9	7.2	3600	8	50	5	1	3	<1,37><2,6><3,0>
C0_5_3	nan	0	2894	7	40	5	1	1	<2,0>
C0_5_4	0.74	2.55	3600	7	40	5	1	2	<2,5><3,0>
C0_5_5	nan	0	3600	8	40	5	1	1	<2,0>
C0_5_6	1	3.04	3600	6	20	5	1	2	<1,6><2,0>
C0_5_7	0.74	3.04	3600	8	40	5	1	2	<2,6><3,0>
C0_5_8	0.9	42.5	3600	7	30	5	1	2	<1,85><2,0>
C0_5_9	nan	0	3260	8	40	5	1	1	<2,0>
C0_10_0	0.72	13.35	3600	12	80	10	1	3	<3,70><4,10><5,0>
C0_10_1	0.76	4.63	3600	10	80	10	1	3	<3,25><4,8><6,3>
C0_10_2	0.54	4.03	3602	12	100	10	1	7	<4,59><5,26><6,17><7,10><8,9><9,2><10,0>
C0_10_3	0.72	13.16	3600	10	80	10	1	4	<3,100><4,17><5,5><6,0>

Table B.1 continued from previous page

Instance	HyperVol	K-dist(2)	CPU_time	#_spots	#_nodes	#_groups	#_tours	#_points	points
C0_10_4	0.64	8.35	3600	10	80	10	1	3	<4,43><5,19><6,6>
C0_10_5	0.68	8.93	3601	11	80	10	1	5	<3,84><4,37><5,17><6,5><7,0>
C0_10_6	0.88	10.51	3600	9	40	10	1	2	<2,21><3,0>
C0_10_7	0.69	11.64	3601	10	80	10	1	4	<3,87><4,42><5,22><6,8>
C0_10_8	0.75	9.69	3600	10	60	10	1	3	<3,50><4,8><5,0>
C0_10_9	0.7	8.27	3600	11	80	10	1	4	<3,57><4,27><5,9><6,0>
C0_30_0	0.82	6.28	3601	13	240	30	1	4	<8,89><9,56><10,46><12,43>
C0_30_1	0.79	2.43	3602	13	240	30	1	7	<9,74><10,63><11,56><12,55><13,52><15,49><17,43>
C0_30_2	0.78	3.37	3603	14	300	30	1	6	<11,84><12,76><13,67><14,61><15,50><21,49>
C0_30_3	0.78	5.49	3601	13	240	30	1	5	<9,115><10,91><11,77><12,62><13,61>
C0_30_4	0.8	2.31	3600	13	240	30	1	6	<9,88><10,84><11,65><12,59><13,56><14,49>
C0_30_5	0.72	6.52	3601	13	240	30	1	5	<10,114><11,95><12,88><13,56><14,40>
C0_30_6	0.86	11.43	3603	11	120	30	1	5	<5,176><6,96><7,83><8,43><9,39>
C0_30_7	0.76	5.98	3603	13	240	30	1	7	<9,133><10,104><11,83><12,55><13,51><14,46><15,44>
C0_30_8	0.76	6.59	3600	12	180	30	1	7	<8,143><9,119><10,105><11,86><12,75><13,54><14,45>
C0_30_9	0.78	3.26	3602	13	240	30	1	6	<9,82><10,73><12,63><13,50><14,47><16,43>
C0_50_0	0.83	3.29	3604	13	400	50	1	5	<14,100><15,90><16,86><18,64><25,62>
C0_50_1	0.86	2.27	3600	13	400	50	1	5	<15,94><16,92><17,88><18,78><19,71>
C0_50_2	0.83	1.96	3602	14	500	50	1	10	<18,104><19,95><20,90><21,82><22,79><23,78><24,74><27,70><30,68> <31,65>
C0_50_3	0.81	3.41	3605	13	400	50	1	6	<14,141><15,133><16,109><18,102><19,90><20,85>
C0_50_4	0.84	1.53	3602	13	400	50	1	11	<14,91><15,80><16,76><17,74><18,73><19,72><20,69><21,63> <22,61><24,60><26,59>
C0_50_5	0.83	2.5	3604	13	400	50	1	8	<16,129><17,118><18,104><19,100><20,89><21,86><22,85><24,82>
C0_50_6	0.87	9.3	3602	11	200	50	1	7	<8,246><9,168><10,139><11,136><12,103><13,89><14,88>
C0_50_7	0.83	2.39	3600	13	400	50	1	7	<16,104><17,95><18,91><19,86><20,83><21,79><25,75>
C0_50_8	0.85	3.07	3604	12	300	50	1	7	<13,185><14,171><15,146><16,142><17,139><18,108><19,106>
C0_50_9	0.87	2.07	3601	13	400	50	1	6	<15,93><16,87><17,81><20,80><21,76><22,73>
C0_70_0	0.87	1.75	3601	13	560	70	1	7	<20,110><21,103><22,98><23,91><24,89><25,83><26,82>
C0_70_1	0.86	2.8	3603	13	560	70	1	8	<20,114><21,104><22,99><23,94><24,91><25,85><27,81><29,72>
C0_70_2	0.88	1.59	3602	14	700	70	1	8	<26,99><28,93><29,91><30,90><31,89><32,84><35,82><37,77>
C0_70_3	0.84	3.8	3611	13	560	70	1	8	<19,167><20,153><21,135><22,132><23,129><24,124><25,112><26,103>
C0_70_4	0.86	1.91	3605	13	560	70	1	8	<21,92><23,89><24,83><25,78><26,74><28,71><31,70><35,69>
C0_70_5	0.86	1.64	3603	13	560	70	1	9	<23,126><24,119><25,118><26,117><27,112><29,107><30,106><31,100> <36,99>
C0_70_6	0.85	7.12	3605	11	280	70	1	6	<13,189><14,171><15,150><16,125><17,122><19,100>
C0_70_7	0.87	3.44	3609	13	560	70	1	8	<20,140><21,108><22,104><23,99><24,97><25,94><26,90><29,88>
C0_70_8	0.89	1.96	3602	12	420	70	1	6	<19,163><20,161><21,145><22,138><24,136><26,130>
C0_70_9	0.87	2.09	3602	13	560	70	1	6	<22,101><24,97><25,90><27,89><28,83><31,78>
C0_90_0	0.88	1.82	3600	13	720	90	1	11	<25,129><26,122><27,113><28,111><29,108><30,105><31,99><33,96> <35,93><36,92><40,89>
C0_90_1	0.9	2.29	3605	13	720	90	1	7	<27,113><28,108><29,101><31,97><33,94><34,89><39,88>
C0_90_2	0.91	2.72	3621	14	900	90	1	5	<34,100><36,94><38,88><44,85><48,84>
C0_90_3	0.87	5.04	3612	13	720	90	1	6	<24,191><25,172><26,165><27,140><28,128><35,125>
C0_90_4	0.87	2.76	3611	13	720	90	1	9	<28,119><29,102><30,98><31,93><32,88><33,86><34,82><38,80> <44,78>
C0_90_5	0.88	1.8	3606	13	720	90	1	10	<28,149><29,147><30,139><31,128><33,124><35,121><37,115><39,114> <45,112><47,109>
C0_90_6	0.88	7.35	3603	11	360	90	1	9	<15,308><16,215><17,213><18,196><20,172><21,170><22,157><23,153> <26,149>
C0_90_7	0.88	1.86	3607	13	720	90	1	7	<27,126><28,124><29,116><30,111><31,103><34,99><37,98>

Table B.1 continued from previous page

Instance	HyperVol	K-dist(2)	CPU_time	#_spots	#_nodes	#_groups	#_tours	#_points	points
C0_90_8	0.89	2.01	3606	12	540	90	1	11	<23,199><24,189><25,176><26,174><27,162><28,158><29,154><30,153><31,149><34,139><35,137>
C0_90_9	0.9	1.32	3616	13	720	90	1	10	<27,113><28,107><29,105><30,103><31,99><32,96><33,93><34,91><35,90><37,89>
C1_5_0	nan	0	3600	9	34	5	2	1	<2,0>
C1_5_1	0.9	42.5	3600	9	36	5	2	2	<1,85><2,0>
C1_5_2	nan	0	3600	9	38	5	2	1	<2,0>
C1_5_3	nan	0	3600	10	40	5	2	1	<2,0>
C1_5_4	nan	0	3600	9	36	5	2	1	<2,0>
C1_5_5	nan	0	3600	10	34	5	2	1	<2,0>
C1_5_6	0.98	9.01	3600	9	28	5	2	2	<1,18><2,0>
C1_5_7	0.73	7.02	3600	10	40	5	2	2	<2,14><3,0>
C1_5_8	nan	0	3242	9	30	5	2	1	<2,0>
C1_5_9	nan	0	3600	10	32	5	2	1	<2,0>
C1_10_0	0.72	21.84	3600	16	64	10	3	3	<3,100><4,31><5,0>
C1_10_1	0.71	25.51	3600	14	66	10	3	3	<3,135><4,18><5,0>
C1_10_2	0.62	8.78	3600	15	84	10	3	4	<4,64><5,21><6,6><7,0>
C1_10_3	0.73	14.86	3600	15	74	10	3	3	<3,81><4,8><5,0>
C1_10_4	0.62	12.18	3600	15	72	10	3	3	<4,55><5,18><6,0>
C1_10_5	0.61	13.51	3600	15	78	10	3	3	<4,56><5,31><6,0>
C1_10_6	0.76	27	3600	15	64	10	3	2	<3,54><4,0>
C1_10_7	0.74	42	3600	14	70	10	3	2	<3,84><4,0>
C1_10_8	0.73	16.17	3600	14	60	10	3	3	<3,71><4,26><5,0>
C1_10_9	0.69	25.69	3601	16	70	10	3	4	<3,203><4,26><5,2><6,0>
C1_30_0	0.72	6	3600	30	204	30	8	5	<10,111><11,70><12,66><14,50><16,45>
C1_30_1	0.69	14.79	3601	27	216	30	8	7	<10,274><11,97><12,81><13,76><14,74><15,47><16,45>
C1_30_2	0.72	13.84	3601	25	188	30	8	3	<10,120><11,77><12,57>
C1_30_3	0.74	5.5	3605	28	194	30	8	6	<9,113><10,68><12,66><14,51><15,47><16,40>
C1_30_4	0.73	5.39	3601	31	214	30	8	6	<10,112><11,90><12,66><13,63><14,61><15,48>
C1_30_5	0.72	4.6	3602	25	192	30	8	6	<10,102><11,84><12,68><13,42><14,41><16,40>
C1_30_6	0.7	8.83	3602	31	206	30	8	5	<10,120><11,77><12,47><13,40><14,36>
C1_30_7	0.72	11.22	3601	26	202	30	8	5	<10,148><11,87><12,76><13,64><14,47>
C1_30_8	0.68	6.6	3606	26	214	30	8	8	<10,140><11,95><12,92><13,75><14,59><15,51><16,43><17,29>
C1_30_9	0.74	12.62	3601	28	204	30	8	6	<9,206><10,126><11,83><12,73><13,70><14,58>
C1_50_0	0.76	1.91	3600	39	342	50	13	6	<16,120><17,116><18,111><19,106><20,84><21,82>
C1_50_1	0.7	4.91	3605	42	370	50	13	7	<18,143><19,110><20,97><21,94><22,88><23,84><24,76>
C1_50_2	0.73	7.06	3601	35	326	50	13	5	<17,148><18,113><19,103><20,94><22,86>
C1_50_3	0.78	16.64	3600	38	348	50	13	4	<15,219><16,121><17,111><18,96>
C1_50_4	0.75	4.66	3600	40	344	50	13	4	<18,123><19,106><20,93><22,90>
C1_50_5	0.76	8	3604	32	300	50	13	5	<15,164><16,113><17,99><18,96><23,89>
C1_50_6	0.73	6.81	3601	43	350	50	13	4	<17,139><18,111><19,89><20,87>
C1_50_7	0.76	4.91	3600	34	344	50	13	6	<17,142><18,99><19,95><20,87><21,84><22,82>
C1_50_8	0.7	3.64	3604	38	360	50	13	5	<19,116><20,106><21,95><22,86><24,83>
C1_50_9	0.74	5.08	3602	38	330	50	13	7	<16,152><17,100><18,98><20,95><21,93><22,88><23,83>
C1_70_0	0.79	3.94	3608	48	460	70	18	5	<23,152><24,140><25,125><27,122><28,114>
C1_70_1	0.72	1.97	3612	53	542	70	18	7	<25,129><26,123><28,113><29,109><30,106><32,105><34,102>
C1_70_2	0.75	7.16	3600	48	456	70	18	4	<23,158><24,134><25,125><27,110>
C1_70_3	0.77	5.44	3600	47	456	70	18	7	<22,174><23,138><24,125><25,111><26,107><27,106><28,99>
C1_70_4	0.76	5.87	3610	51	490	70	18	8	<25,202><26,133><27,125><28,119><29,117><30,116><31,104><33,102>
C1_70_5	0.76	7.26	3602	41	440	70	18	7	<21,211><22,165><23,144><24,131><25,127><26,123><28,114>
C1_70_6	0.71	5.04	3602	54	502	70	18	6	<25,156><26,129><27,119><28,110><31,107><34,102>
C1_70_7	0.79	6.35	3604	41	460	70	18	7	<22,199><23,153><24,125><25,122><26,121><27,110><31,108>

Table B.1 continued from previous page

Instance	HyperVol	K-dist(2)	CPU_time	#_spots	#_nodes	#_groups	#_tours	#_points	points
C1_70_8	0.74	5.33	3612	49	494	70	18	5	<25,155><26,139><27,131><28,124><29,109>
C1_70_9	0.75	4.1	3610	50	498	70	18	6	<24,157><25,130><26,123><27,104><28,102><31,100>
C1_90_0	0.74	5.83	3607	55	600	90	23	7	<31,203><32,165><33,149><34,146><35,142><36,136><37,125>
C1_90_1	0.72	3.62	3613	58	646	90	23	5	<33,139><34,121><37,116><40,114><41,109>
C1_90_2	0.75	3.76	3602	59	612	90	23	9	<31,182><32,157><33,139><34,131><35,127><36,121><37,120><38,119> <39,115>
C1_90_3	0.76	2.84	3615	57	590	90	23	6	<29,155><30,142><31,133><32,130><33,123><35,121>
C1_90_4	0.8	3.89	3609	58	626	90	23	5	<32,158><33,141><34,136><35,129><38,124>
C1_90_5	0.76	4.5	3609	51	570	90	23	7	<29,183><30,160><31,149><33,142><34,135><36,129><40,128>
C1_90_6	0.72	5.17	3611	64	630	90	23	6	<32,172><33,134><34,132><35,124><37,123><39,108>
C1_90_7	0.77	8.54	3616	49	604	90	23	7	<30,247><31,179><32,172><33,152><34,140><36,132><40,124>
C1_90_8	0.73	4.86	3600	59	632	90	23	7	<32,187><33,173><34,151><35,130><36,128><38,124><39,114>
C1_90_9	0.73	4.49	3606	61	638	90	23	8	<31,191><32,157><33,149><34,131><35,127><36,124><38,118><42,114>
C2_5_0	nan	0	3600	11	34	5	3	1	<2,0>
C2_5_1	nan	0	3600	11	34	5	3	1	<2,0>
C2_5_2	0.74	4.03	3600	11	42	5	3	2	<2,8><3,0>
C2_5_3	nan	0	3600	12	36	5	3	1	<2,0>
C2_5_4	0.71	11.51	3600	12	36	5	3	2	<2,23><3,0>
C2_5_5	0.75	1.12	3600	12	38	5	3	2	<2,2><3,0>
C2_5_6	nan	0	3600	12	36	5	3	1	<2,0>
C2_5_7	nan	0	3403	12	36	5	3	1	<2,0>
C2_5_8	nan	0	3000	11	30	5	3	1	<2,0>
C2_5_9	nan	0	3600	13	34	5	3	1	<2,0>
C2_10_0	0.65	4.38	3600	21	66	10	5	4	<4,33><5,4><6,1><7,0>
C2_10_1	0.76	12.38	3600	18	64	10	5	3	<3,70><4,4><5,0>
C2_10_2	0.65	12.01	3600	19	68	10	5	2	<4,24><5,0>
C2_10_3	0.75	13.69	3600	19	68	10	5	3	<3,75><4,7><5,0>
C2_10_4	0.63	5.17	3601	21	76	10	5	5	<4,54><5,23><6,10><7,7><8,5>
C2_10_5	0.64	10.02	3600	19	72	10	5	3	<4,50><5,10><6,0>
C2_10_6	0.76	12.84	3600	20	72	10	5	3	<3,59><4,18><5,0>
C2_10_7	0.65	8.04	3600	17	70	10	5	3	<4,44><5,4><6,0>
C2_10_8	0.76	9.52	3600	18	58	10	5	3	<3,44><4,13><5,0>
C2_10_9	0.75	40.5	3600	20	70	10	5	2	<3,81><4,0>
C2_30_0	0.71	17.32	3601	43	212	30	15	5	<10,217><11,96><12,81><13,56><14,45>
C2_30_1	0.69	6.75	3601	47	228	30	15	5	<11,121><12,92><13,70><15,61><17,58>
C2_30_2	0.73	6.19	3600	38	186	30	15	4	<10,106><11,75><12,61><13,59>
C2_30_3	0.75	9.42	3600	42	202	30	15	5	<9,135><10,100><11,84><12,72><13,53>
C2_30_4	0.71	6.46	3600	45	206	30	15	5	<11,114><12,84><13,64><15,58><16,54>
C2_30_5	0.73	8.44	3600	36	190	30	15	4	<10,120><11,62><12,59><13,56>
C2_30_6	0.72	8.22	3600	47	218	30	15	5	<10,132><11,102><12,92><13,79><14,60>
C2_30_7	0.74	4.44	3600	37	196	30	15	5	<10,98><11,86><12,72><13,58><14,55>
C2_30_8	0.69	6.33	3600	42	216	30	15	4	<11,98><12,80><13,55><15,48>
C2_30_9	0.74	17.11	3602	43	198	30	15	3	<9,141><10,65><13,52>
C2_50_0	0.73	7.66	3601	60	330	50	25	6	<16,173><17,135><18,112><19,100><23,93><25,88>
C2_50_1	0.74	7.48	3603	61	342	50	25	6	<16,173><17,129><18,109><19,81><21,79><22,78>
C2_50_2	0.74	6.27	3603	62	336	50	25	4	<17,128><18,110><19,100><20,88>
C2_50_3	0.77	5.5	3602	60	336	50	25	6	<16,167><17,132><18,122><19,113><20,100><21,99>
C2_50_4	0.74	3.9	3602	62	340	50	25	5	<18,121><19,102><20,92><21,89><23,86>
C2_50_5	0.74	4.97	3604	55	324	50	25	6	<16,141><17,115><18,104><19,95><21,84><22,82>
C2_50_6	0.71	1.39	3601	66	356	50	25	5	<18,103><19,102><20,92><22,87><24,85>
C2_50_7	0.75	2.14	3600	54	334	50	25	6	<17,120><18,119><19,112><20,108><21,100><23,93>
C2_50_8	0.74	10.78	3601	62	356	50	25	6	<17,208><18,141><19,113><20,86><22,85><24,83>

Table B.1 continued from previous page

Instance	HyperVol	K-dist(2)	CPU_time	#_spots	#_nodes	#_groups	#_tours	#_points	points
C2_50_9	0.75	5.95	3605	64	344	50	25	7	<16,167><17,125><18,99><19,94><21,92><22,89><24,88>
C2_70_0	0.75	7.13	3603	76	466	70	35	7	<22,204><23,152><24,139><25,129><26,122><27,120><29,107>
C2_70_1	0.73	4.97	3603	80	494	70	35	8	<24,172><25,137><26,116><27,115><28,110><29,104><30,100><32,93>
C2_70_2	0.74	7.76	3609	77	464	70	35	5	<24,178><25,125><26,118><27,116><28,103>
C2_70_3	0.75	3.14	3601	74	476	70	35	8	<23,148><25,126><26,119><27,115><28,113><30,110><31,108><32,100>
C2_70_4	0.74	4.46	3608	76	458	70	35	5	<23,165><24,145><25,135><26,119><27,117>
C2_70_5	0.72	4.26	3605	74	480	70	35	6	<25,153><26,136><27,116><28,115><29,112><30,101>
C2_70_6	0.73	15.79	3604	80	480	70	35	6	<24,288><25,144><26,128><27,111><28,104><31,103>
C2_70_7	0.76	3.22	3605	74	500	70	35	7	<24,157><25,144><26,129><27,119><29,118><30,110><32,109>
C2_70_8	0.71	3.52	3609	80	502	70	35	7	<26,143><27,126><28,113><29,111><30,108><32,106><33,97>
C2_70_9	0.74	2.1	3603	78	484	70	35	7	<24,132><25,130><26,126><27,115><28,106><31,104><33,100>
C2_90_0	0.74	3.87	3613	83	618	90	45	6	<32,169><33,134><34,130><35,129><37,127><39,126>
C2_90_1	0.73	4.71	3622	96	662	90	45	6	<33,172><34,152><35,133><36,128><37,120><39,117>
C2_90_2	0.75	12.09	3617	91	576	90	45	5	<29,234><30,162><31,141><33,124><38,122>
C2_90_3	0.74	7.6	3614	91	600	90	45	5	<31,183><32,151><33,123><37,121><38,114>
C2_90_4	0.72	2.86	3609	90	624	90	45	4	<32,142><34,141><35,125><37,116>
C2_90_5	0.74	5.12	3610	83	604	90	45	6	<31,182><32,148><33,141><34,136><35,129><36,124>
C2_90_6	0.73	3.68	3613	89	630	90	45	6	<33,175><34,142><36,141><37,126><38,124><40,123>
C2_90_7	0.76	4.56	3619	86	620	90	45	4	<33,156><34,138><36,128><37,124>
C2_90_8	0.73	6.71	3621	96	652	90	45	6	<34,180><35,139><37,125><39,119><41,113><45,108>
C2_90_9	0.72	1.64	3617	88	626	90	45	7	<32,150><33,148><34,128><36,122><37,121><38,119><39,112>
C3_5_0	nan	0	3600	13	34	5	4	1	<2,0>
C3_5_1	nan	0	1940	12	30	5	4	1	<2,0>
C3_5_2	nan	0	3600	13	38	5	4	1	<2,0>
C3_5_3	nan	0	3600	15	36	5	4	1	<2,0>
C3_5_4	0.69	16.01	3600	15	38	5	4	2	<2,32><3,0>
C3_5_5	nan	0	2893	14	36	5	4	1	<2,0>
C3_5_6	nan	0	3600	13	28	5	4	1	<2,0>
C3_5_7	nan	0	3600	13	34	5	4	1	<2,0>
C3_5_8	nan	0	3600	14	32	5	4	1	<2,0>
C3_5_9	0.74	6.02	3600	15	38	5	4	2	<2,12><3,0>
C3_10_0	0.65	7.03	3600	29	68	10	8	3	<4,36><5,6><6,0>
C3_10_1	0.66	6.02	3600	24	70	10	8	2	<4,12><5,0>
C3_10_2	0.75	19.51	3600	23	66	10	8	3	<3,103><4,14><5,0>
C3_10_3	0.65	17.01	3600	25	68	10	8	2	<4,34><5,0>
C3_10_4	0.64	9.31	3600	28	72	10	8	3	<4,54><5,1><6,0>
C3_10_5	0.66	10.51	3601	23	66	10	8	2	<4,21><5,0>
C3_10_6	0.72	30.51	3600	29	70	10	8	3	<3,154><4,29><5,0>
C3_10_7	0.65	4.87	3600	23	68	10	8	3	<4,24><5,5><6,0>
C3_10_8	0.65	22.01	3600	24	68	10	8	2	<4,44><5,0>
C3_10_9	0.66	13.51	3600	26	68	10	8	2	<4,27><5,0>
C3_30_0	0.73	8.02	3602	55	202	30	23	4	<10,111><11,88><12,69><13,58>
C3_30_1	0.72	8.42	3600	58	216	30	23	5	<10,125><11,84><12,67><13,59><14,49>
C3_30_2	0.7	6.06	3604	59	204	30	23	5	<10,116><11,80><12,70><13,37><14,35>
C3_30_3	0.72	5.11	3600	57	204	30	23	5	<10,101><11,84><12,70><13,59><15,55>
C3_30_4	0.74	8.72	3600	58	200	30	23	5	<10,138><11,104><12,83><13,67><14,59>
C3_30_5	0.74	5.44	3601	51	190	30	23	5	<9,115><10,102><11,77><12,55><13,52>
C3_30_6	0.71	3.1	3600	64	212	30	23	7	<10,105><11,86><12,84><13,68><14,58><16,53><17,51>
C3_30_7	0.72	10.77	3600	49	202	30	23	4	<10,128><11,72><12,62><13,52>
C3_30_8	0.7	5.71	3602	59	216	30	23	6	<11,121><12,81><13,74><14,66><15,60><16,56>
C3_30_9	0.7	7.28	3600	61	216	30	23	4	<11,97><12,80><13,68><15,51>
C3_50_0	0.73	5.46	3602	79	346	50	38	4	<17,121><18,104><20,94><22,86>

Table B.1 continued from previous page

Instance	HyperVol	K-dist(2)	CPU_time	#_spots	#_nodes	#_groups	#_tours	#_points	points
C3_50_1	0.72	7.06	3600	86	360	50	38	6	<17,162><18,100><19,98><20,94><21,90><22,80>
C3_50_2	0.75	5.36	3609	81	324	50	38	6	<15,172><16,153><17,115><18,109><19,98><20,91>
C3_50_3	0.74	4.39	3600	82	336	50	38	6	<17,136><18,113><19,103><20,92><21,90><26,89>
C3_50_4	0.73	4.18	3602	81	328	50	38	8	<16,141><17,118><18,113><19,108><20,99><22,93><23,87><24,77>
C3_50_5	0.72	6.62	3603	77	344	50	38	6	<17,157><18,113><19,99><20,88><21,86><24,81>
C3_50_6	0.72	3.92	3600	82	362	50	38	8	<17,149><18,120><19,104><20,96><22,95><23,93><26,88><27,87>
C3_50_7	0.74	5.2	3603	78	344	50	38	6	<17,145><18,122><19,111><20,101><21,95><22,89>
C3_50_8	0.69	3.81	3601	85	360	50	38	9	<18,132><19,110><20,97><21,90><22,81><27,79><29,68><30,64><31,62>
C3_50_9	0.73	6.27	3603	82	350	50	38	7	<17,169><18,116><19,103><20,96><21,92><22,90><23,84>
C3_70_0	0.72	5.5	3603	89	482	70	53	7	<25,180><26,141><27,120><28,115><29,105><30,103><31,101>
C3_70_1	0.73	11.1	3612	102	486	70	53	4	<24,177><25,112><28,107><30,95>
C3_70_2	0.76	3.66	3604	99	464	70	53	7	<23,157><24,135><25,128><26,120><28,114><31,111><32,109>
C3_70_3	0.74	4.9	3607	99	470	70	53	6	<24,160><25,142><26,118><27,114><29,109><30,100>
C3_70_4	0.74	7.54	3609	96	466	70	53	6	<23,199><24,157><25,135><26,122><29,121><30,114>
C3_70_5	0.72	6.66	3603	93	476	70	53	4	<25,150><26,130><27,120><29,107>
C3_70_6	0.73	8.33	3607	98	480	70	53	6	<24,197><25,139><26,128><27,120><29,115><30,103>
C3_70_7	0.73	1.99	3604	97	484	70	53	6	<25,126><27,122><28,118><29,117><30,115><32,105>
C3_70_8	0.74	2.14	3604	103	494	70	53	8	<25,138><26,126><28,107><29,106><30,104><31,98><32,96><33,95>
C3_70_9	0.74	12.14	3609	101	486	70	53	8	<22,303><23,164><24,140><25,126><26,122><27,112><28,111><29,105>
C3_90_0	0.73	7.36	3606	105	616	90	68	5	<32,188><33,150><34,136><36,135><37,118>
C3_90_1	0.73	3.37	3609	117	628	90	68	7	<32,158><33,143><34,129><35,126><36,125><38,117><39,111>
C3_90_2	0.74	9.19	3618	112	598	90	68	5	<30,213><31,158><32,157><33,146><34,123>
C3_90_3	0.73	4.64	3604	109	610	90	68	5	<31,174><32,162><33,139><34,123><37,122>
C3_90_4	0.74	7.04	3604	109	618	90	68	5	<31,210><32,166><33,156><34,123><37,122>
C3_90_5	0.74	3.3	3608	112	608	90	68	7	<31,170><32,150><33,146><34,131><37,127><39,123><42,120>
C3_90_6	0.72	7.43	3605	112	630	90	68	3	<33,156><34,126><36,119>
C3_90_7	0.74	7.82	3610	115	616	90	68	6	<31,208><32,142><33,138><35,135><36,122><44,120>
C3_90_8	0.76	3.15	3610	113	626	90	68	9	<32,174><33,154><34,141><35,132><36,127><38,126><39,120><40,119><42,118>
C3_90_9	0.75	4.45	3605	117	626	90	68	6	<31,170><32,160><33,148><34,138><35,134><36,119>
C4_5_0	nan	0	3600	16	34	5	5	1	<2,0>
C4_5_1	nan	0	3600	15	32	5	5	1	<2,0>
C4_5_2	nan	0	3600	15	34	5	5	1	<2,0>
C4_5_3	nan	0	3600	16	34	5	5	1	<2,0>
C4_5_4	0.71	11.01	3600	18	38	5	5	2	<2,22><3,0>
C4_5_5	nan	0	3600	16	34	5	5	1	<2,0>
C4_5_6	nan	0	3600	17	34	5	5	1	<2,0>
C4_5_7	nan	0	3076	15	34	5	5	1	<2,0>
C4_5_8	nan	0	3600	15	30	5	5	1	<2,0>
C4_5_9	nan	0	3600	17	36	5	5	1	<2,0>
C4_10_0	0.66	2.55	3600	33	70	10	10	2	<4,5><5,0>
C4_10_1	0.65	21.01	3600	29	70	10	10	2	<4,42><5,0>
C4_10_2	0.75	20.18	3601	27	64	10	10	3	<3,105><4,16><5,0>
C4_10_3	0.66	4.53	3600	29	68	10	10	2	<4,9><5,0>
C4_10_4	0.63	11.68	3600	32	72	10	10	3	<4,56><5,14><6,0>
C4_10_5	0.73	11.04	3600	26	62	10	10	4	<3,84><4,19><5,4><6,0>
C4_10_6	0.65	7.08	3601	33	72	10	10	3	<4,40><5,2><6,0>
C4_10_7	0.64	9.35	3600	26	66	10	10	3	<4,38><5,18><6,0>
C4_10_8	0.65	29.5	3600	28	68	10	10	2	<4,59><5,0>
C4_10_9	0.65	6.02	3600	30	68	10	10	3	<4,27><5,9><6,0>
C4_30_0	0.72	5.55	3600	68	202	30	30	4	<10,100><11,81><12,55><13,52>
C4_30_1	0.71	4.44	3602	71	212	30	30	5	<11,93><12,80><13,73><14,66><16,56>

Table B.1 continued from previous page

Instance	HyperVol	K-dist(2)	CPU_time	#_spots	#_nodes	#_groups	#_tours	#_points	points
C4_30_2	0.71	9.53	3601	72	202	30	30	5	<10,134><11,89><12,62><13,58><14,43>
C4_30_3	0.75	10.04	3603	66	200	30	30	6	<9,174><10,114><11,77><12,71><13,68><14,57>
C4_30_4	0.75	10.89	3600	70	202	30	30	4	<10,139><11,105><12,74><13,63>
C4_30_5	0.71	5.51	3602	65	200	30	30	5	<10,110><11,78><12,60><13,58><14,57>
C4_30_6	0.71	4.57	3600	73	210	30	30	4	<10,115><11,107><12,63><14,53>
C4_30_7	0.71	10.11	3601	63	206	30	30	6	<10,163><11,113><12,77><13,65><14,56><15,49>
C4_30_8	0.7	4.23	3600	73	218	30	30	6	<11,108><12,83><13,75><14,67><15,65><16,60>
C4_30_9	0.73	7.52	3602	70	208	30	30	4	<10,109><11,83><12,76><13,56>
C4_50_0	0.73	4.82	3601	88	340	50	50	4	<17,125><18,106><19,91><20,89>
C4_50_1	0.71	1.29	3600	100	354	50	50	7	<18,107><19,100><20,99><21,89><22,88><23,86><24,83>
C4_50_2	0.75	4.68	3605	96	332	50	50	6	<16,142><17,116><18,99><20,98><21,97><22,89>
C4_50_3	0.73	5.39	3602	97	332	50	50	8	<16,169><17,114><18,112><19,104><20,94><21,91><22,88><24,79>
C4_50_4	0.72	3.48	3604	95	338	50	50	5	<17,117><18,106><20,99><21,95><23,87>
C4_50_5	0.72	3.44	3603	90	340	50	50	7	<17,130><18,113><19,101><20,96><21,83><22,77><23,76>
C4_50_6	0.73	6.72	3605	93	350	50	50	5	<17,155><18,105><19,104><20,98><22,90>
C4_50_7	0.74	4.44	3600	90	338	50	50	6	<16,143><17,136><18,123><19,104><20,96><26,88>
C4_50_8	0.72	5.42	3603	100	354	50	50	6	<18,153><19,111><20,104><21,85><22,84><23,78>
C4_50_9	0.73	3.38	3605	95	350	50	50	6	<17,124><18,109><19,103><20,98><21,95><22,87>
C4_70_0	0.72	4.35	3609	107	480	70	70	7	<24,160><25,144><26,123><28,114><29,109><30,107><31,97>
C4_70_1	0.73	3.9	3604	118	484	70	70	7	<24,152><25,128><26,120><27,119><28,116><30,103><32,95>
C4_70_2	0.75	2.16	3605	115	468	70	70	6	<24,138><25,129><26,126><27,120><28,107><30,106>
C4_70_3	0.75	3.81	3604	112	474	70	70	6	<24,154><25,141><26,122><27,113><28,107><30,106>
C4_70_4	0.73	6.4	3601	110	472	70	70	4	<24,150><25,133><26,121><27,110>
C4_70_5	0.72	3.23	3605	112	478	70	70	7	<25,148><26,123><27,113><29,111><30,110><31,106><32,104>
C4_70_6	0.73	1.74	3604	113	484	70	70	7	<24,130><26,126><27,113><31,108><33,107><34,102><36,101>
C4_70_7	0.72	3.89	3610	116	482	70	70	6	<25,149><26,135><27,114><30,111><33,110><34,102>
C4_70_8	0.74	2.71	3609	115	486	70	70	8	<25,144><26,133><27,125><28,118><29,109><31,107><32,100><33,98>
C4_70_9	0.74	3.23	3603	117	486	70	70	6	<24,144><25,125><26,124><27,120><29,110><33,105>
C4_90_0	0.74	7.22	3614	125	620	90	90	8	<31,234><32,165><33,154><34,138><35,132><36,126><37,121><40,119>
C4_90_1	0.75	11.12	3601	129	616	90	90	3	<31,173><32,133><35,120>
C4_90_2	0.75	5.19	3601	128	614	90	90	7	<31,198><32,159><33,151><34,134><36,130><37,128><40,120>
C4_90_3	0.74	5.37	3621	129	618	90	90	7	<32,185><33,142><34,137><35,128><36,120><37,118><40,109>
C4_90_4	0.72	3.77	3603	122	614	90	90	7	<32,170><33,146><34,140><35,138><36,127><40,125><42,116>
C4_90_5	0.73	5.5	3616	127	616	90	90	6	<32,185><33,150><34,140><35,136><36,124><42,122>
C4_90_6	0.75	4.41	3614	126	612	90	90	8	<30,211><31,199><32,166><33,145><34,137><35,132><36,126><38,123>
C4_90_7	0.72	5.96	3621	128	626	90	90	5	<33,170><34,132><36,121><40,120><41,117>
C4_90_8	0.73	4.54	3614	128	618	90	90	8	<32,189><33,145><35,135><36,131><37,128><38,127><39,121><40,117>
C4_90_9	0.76	6.94	3610	125	616	90	90	8	<30,234><31,164><32,149><33,144><34,138><35,127><36,123><38,121>

B.2 AUGMECON-CP1

Table B.2: Results of the AUGMECON-CP1

Instance	HyperVol	K-dist(2)	CPU_time	#_spots	#_nodes	#_groups	#_tours	#_points	points
C0_5_0	0.97	8.51	7	7	40	5	1	2	<1,17><2,0>
C0_5_1	0.96	10.01	13	7	40	5	1	2	<1,20><2,0>
C0_5_2	0.96	3.17	560	8	50	5	1	3	<1,17><2,0><3,0>
C0_5_3	0.99	3.54	10	7	40	5	1	2	<1,7><2,0>
C0_5_4	0.96	9.51	12	7	40	5	1	2	<1,19><2,0>
C0_5_5	0.99	3.04	10	8	40	5	1	2	<1,6><2,0>
C0_5_6	1	0.71	4	6	20	5	1	2	<1,1><2,0>
C0_5_7	0.95	14.01	25	8	40	5	1	2	<1,28><2,0>
C0_5_8	0.99	2.55	61	7	30	5	1	2	<1,5><2,0>
C0_5_9	0.97	6.52	14	8	40	5	1	2	<1,13><2,0>
C0_10_0	0.94	13.53	2490	12	80	10	1	3	<1,75><2,6><3,0>
C0_10_1	0.96	7.31	2263	10	80	10	1	3	<1,42><2,1><3,0>
C0_10_2	0.88	2.34	2656	12	100	10	1	3	<2,12><3,0><4,0>
C0_10_3	0.96	12.17	3494	10	80	10	1	3	<1,71><2,0><3,0>
C0_10_4	0.94	11.02	2124	10	80	10	1	3	<1,58><2,8><3,0>
C0_10_5	0.95	11.41	2498	11	80	10	1	3	<1,66><2,2><3,0>
C0_10_6	1	8.02	17	9	40	10	1	2	<1,16><2,0>
C0_10_7	0.89	0.71	1904	10	80	10	1	2	<2,1><3,0>
C0_10_8	0.95	15.81	2910	10	60	10	1	3	<1,93><2,1><3,0>
C0_10_9	0.88	3.54	2591	11	80	10	1	2	<2,7><3,0>
C0_30_0	0.9	5.73	3600	13	240	30	1	5	<3,53><4,10><9,4><10,3><12,0>
C0_30_1	0.9	4.21	3600	13	240	30	1	6	<3,45><4,19><5,10><9,5><10,0><12,0>
C0_30_2	0.82	1.08	3600	14	300	30	1	9	<5,16><9,16><10,13><11,8><12,8><14,6><15,1><17,0><18,0>
C0_30_3	0.92	7.94	3600	13	240	30	1	5	<3,72><4,18><5,8><6,1><10,0>
C0_30_4	0.89	2.96	3600	13	240	30	1	9	<3,58><4,26><5,19><7,15><8,14><9,5><10,4><11,0><13,0>
C0_30_5	0.88	5.51	3600	13	240	30	1	8	<3,96><4,40><5,36><6,30><7,15><8,9><10,2><12,0>
C0_30_6	0.96	4.03	3600	11	120	30	1	2	<2,8><3,0>
C0_30_7	0.88	1.55	3600	13	240	30	1	6	<4,22><6,16><7,7><8,7><10,0><12,0>
C0_30_8	0.92	2.77	3600	12	180	30	1	4	<3,26><4,20><5,3><9,0>
C0_30_9	0.87	5.83	3600	13	240	30	1	10	<3,115><4,40><5,22><6,16><7,9><9,8><10,3><11,2><12,0><14,0>
C0_50_0	0.85	3.48	3600	13	400	50	1	10	<5,72><6,61><7,51><8,33><9,24><10,23><15,20><16,16><18,11><25,0>
C0_50_1	0.83	4.82	3600	13	400	50	1	10	<5,102><6,46><7,46><8,32><9,30><11,26><12,20><15,18><17,15><28,0>
C0_50_2	0.75	3.6	3601	14	500	50	1	8	<7,46><10,39><11,29><12,21><23,17><30,14><32,12><37,0>
C0_50_3	0.88	6.96	3601	13	400	50	1	8	<5,107><6,43><7,29><8,25><13,21><14,19><19,9><24,0>
C0_50_4	0.81	2.07	3600	13	400	50	1	11	<6,35><8,29><9,26><11,25><12,16><14,16><17,15><18,14><28,13><34,12><38,0>

Table B.2 continued from previous page

Instance	HyperVol	K-dist(2)	CPU_time	#_spots	#_nodes	#_groups	#_tours	#_points	points
C0_50_5	0.84	4.6	3600	13	400	50	1	9	<5,95><6,71><8,70><9,48><12,28><13,20><14,16><16,15><25,0>
C0_50_6	0.96	22.51	3600	11	200	50	1	2	<3,45><4,0>
C0_50_7	0.83	4.03	3601	13	400	50	1	11	<6,84><8,47><9,31><12,27><13,23><15,22><16,18><18,16><19,15><21,12><25,0>
C0_50_8	0.87	8.32	3600	12	300	50	1	8	<5,126><6,69><9,25><13,23><14,20><16,14><19,8><21,0>
C0_50_9	0.81	3.17	3601	13	400	50	1	14	<6,97><7,49><8,45><9,31><10,29><11,28><12,26><13,25><14,24><15,18><16,18><17,16><19,14><27,0>
C0_70_0	0.82	3.6	3600	13	560	70	1	10	<7,74><8,65><9,61><10,46><12,34><21,28><22,24><28,19><30,17><35,0>
C0_70_1	0.8	2.94	3600	13	560	70	1	17	<7,98><8,76><9,51><10,47><12,41><13,38><14,33><15,30><19,27><21,27><22,26><23,24><27,20><33,19><35,18><36,18><38,0>
C0_70_2	0.72	3.64	3600	14	700	70	1	11	<12,67><13,51><15,47><16,40><17,35><18,32><19,29><20,21><32,17><35,12><52,0>
C0_70_3	0.87	5.34	3600	13	560	70	1	12	<6,129><7,85><8,59><9,56><10,50><11,47><13,40><14,39><16,38><17,25><25,21><33,0>
C0_70_4	0.8	3.09	3600	13	560	70	1	13	<8,62><9,43><10,43><11,37><14,31><15,29><19,28><20,26><21,20><26,15><36,12><45,10><53,0>
C0_70_5	0.78	4.22	3600	13	560	70	1	13	<9,109><10,94><12,82><13,70><14,64><15,56><18,49><19,43><20,31><24,29><25,26><27,24><35,0>
C0_70_6	0.93	4.86	3600	11	280	70	1	7	<5,61><6,30><10,26><13,26><15,18><19,17><21,0>
C0_70_7	0.82	5.74	3600	13	560	70	1	9	<8,91><9,65><10,50><12,39><18,28><22,27><26,26><27,21><35,0>
C0_70_8	0.83	7.56	3600	12	420	70	1	7	<7,102><8,96><9,77><12,58><18,39><27,37><28,0>
C0_70_9	0.8	5.79	3600	13	560	70	1	8	<8,86><9,48><11,47><13,35><22,24><23,19><31,18><39,0>
C0_90_0	0.8	2.81	3600	13	720	90	1	14	<10,68><11,60><12,53><13,53><14,50><16,48><21,40><26,31><28,28><29,25><32,25><35,23><42,23><45,0>
C0_90_1	0.79	3.15	3600	13	720	90	1	11	<11,62><13,40><16,38><22,34><23,32><24,26><25,25><30,23><31,22><36,20><50,0>
C0_90_2	0.68	3.26	3600	14	900	90	1	11	<20,47><21,41><22,36><30,28><32,23><41,21><43,17><47,15><56,15><64,15><67,0>
C0_90_3	0.83	3.15	3600	13	720	90	1	11	<12,70><13,53><15,53><20,43><21,36><22,34><23,30><26,26><27,26><30,22><43,0>
C0_90_4	0.77	4.04	3601	13	720	90	1	10	<12,58><16,50><18,36><20,22><25,20><27,20><40,15><56,15><61,13><68,0>
C0_90_5	0.78	4.33	3600	13	720	90	1	12	<12,106><13,82><16,79><17,71><19,57><22,41><26,39><28,33><33,29><34,27><37,24><45,0>
C0_90_6	0.92	26.92	3600	11	360	90	1	7	<6,381><7,88><9,75><10,65><11,36><19,31><26,0>
C0_90_7	0.8	4.62	3600	13	720	90	1	10	<11,86><12,73><15,54><17,54><18,50><20,38><23,29><26,22><37,21><45,0>
C0_90_8	0.78	4.35	3600	12	540	90	1	10	<15,85><22,71><24,51><25,48><26,43><27,41><28,39><32,38><33,33><38,0>
C0_90_9	0.79	3.45	3600	13	720	90	1	9	<11,45><23,32><24,27><27,25><28,22><33,22><39,20><40,19><50,0>
C1_5_0	0.96	12.51	8	9	34	5	2	2	<1,25><2,0>
C1_5_1	0.98	7.52	5	9	36	5	2	2	<1,15><2,0>
C1_5_2	0.97	19.01	11	9	38	5	2	2	<1,38><2,0>
C1_5_3	0.98	8.02	13	10	40	5	2	2	<1,16><2,0>
C1_5_4	0.89	35	13	9	36	5	2	2	<1,70><2,0>
C1_5_5	0.98	8.02	6	10	34	5	2	2	<1,16><2,0>
C1_5_6	0.99	5.52	4	9	28	5	2	2	<1,11><2,0>
C1_5_7	0.92	26	18	10	40	5	2	2	<1,52><2,0>
C1_5_8	0.98	7.52	3	9	30	5	2	2	<1,15><2,0>
C1_5_9	0.99	8.51	7	10	32	5	2	2	<1,17><2,0>
C1_10_0	0.88	9.01	2311	16	64	10	3	2	<2,18><3,0>

Table B.2 continued from previous page

Instance	HyperVol	K-dist(2)	CPU_time	#_spots	#_nodes	#_groups	#_tours	#_points	points
C1_10_1	0.89	1.35	2814	14	66	10	3	3	<2,6><3,0><4,0>
C1_10_2	0.88	3.67	2879	15	84	10	3	3	<2,20><3,0><4,0>
C1_10_3	0.94	17.41	3600	15	74	10	3	3	<1,102><2,2><3,0>
C1_10_4	0.86	6.86	2732	15	72	10	3	3	<2,34><3,7><4,0>
C1_10_5	0.88	5.52	2911	15	78	10	3	2	<2,11><3,0>
C1_10_6	0.89	2.06	2594	15	64	10	3	2	<2,4><3,0>
C1_10_7	0.89	0.71	1381	14	70	10	3	2	<2,1><3,0>
C1_10_8	0.89	2.55	2849	14	60	10	3	2	<2,5><3,0>
C1_10_9	0.88	3.84	2855	16	70	10	3	3	<2,21><3,0><4,0>
C1_30_0	0.84	5.48	3600	30	204	30	8	5	<5,50><7,32><8,21><10,7><12,0>
C1_30_1	0.84	8.18	3600	27	216	30	8	8	<5,133><6,50><7,32><8,22><10,19><11,9><12,6><14,0>
C1_30_2	0.88	5.75	3490	25	188	30	8	7	<4,85><5,50><6,46><7,34><8,17><9,8><10,0>
C1_30_3	0.88	4.98	3600	28	194	30	8	6	<4,77><5,58><6,19><8,8><9,2><10,0>
C1_30_4	0.86	3.36	3600	31	214	30	8	10	<4,76><5,63><6,45><7,37><8,22><9,14><10,10><12,8><13,3><15,0>
C1_30_5	0.83	14.53	3600	25	192	30	8	6	<4,169><5,80><6,57><7,52><8,42><11,0>
C1_30_6	0.85	4.51	3528	31	206	30	8	7	<5,70><6,38><7,32><8,23><9,6><11,1><13,0>
C1_30_7	0.83	7.07	3600	26	202	30	8	8	<5,119><6,53><7,35><9,23><10,5><11,4><13,0><14,0>
C1_30_8	0.85	6.34	3600	26	214	30	8	9	<4,198><5,158><6,49><7,25><9,24><10,5><11,4><12,1><13,0>
C1_30_9	0.85	5.61	3432	28	204	30	8	5	<5,64><6,35><8,29><9,6><13,0>
C1_50_0	0.82	8.06	3601	39	342	50	13	12	<7,198><8,104><9,86><10,72><12,55><13,54><14,48><16,40><17,35><19,35><20,31><23,0>
C1_50_1	0.81	5.96	3600	42	370	50	13	11	<8,156><9,92><10,89><12,66><13,62><14,51><15,48><17,45><18,39><19,36><24,0>
C1_50_2	0.82	5.94	3600	35	326	50	13	11	<8,144><9,91><10,71><11,70><12,61><13,46><16,45><17,35><19,31><21,31><22,0>
C1_50_3	0.83	3.81	3601	38	348	50	13	12	<7,103><8,93><9,78><11,55><12,49><14,43><15,39><16,36><17,30><19,28><20,24><21,0>
C1_50_4	0.8	2.85	3600	40	344	50	13	14	<8,103><9,84><10,82><11,63><13,54><14,53><15,44><16,42><17,38><19,36><20,34><21,29><22,26><23,0>
C1_50_5	0.84	2.29	3600	32	300	50	13	15	<7,92><8,91><9,83><10,80><11,69><12,60><13,56><14,44><15,44><16,37><17,35><18,33><19,31><20,29><21,0>
C1_50_6	0.84	7.33	3600	43	350	50	13	11	<7,177><8,108><9,100><10,82><11,72><13,66><14,50><15,43><17,34><20,32><21,0>
C1_50_7	0.81	3.8	3600	34	344	50	13	14	<7,114><8,87><9,74><10,66><12,57><13,55><14,47><15,47><16,42><17,34><20,29><22,26><23,23><24,0>
C1_50_8	0.82	4.75	3526	38	360	50	13	13	<8,131><9,89><10,71><11,69><12,58><13,53><14,48><15,47><16,43><18,38><19,36><21,35><23,0>
C1_50_9	0.84	9.03	3601	38	330	50	13	13	<7,248><8,86><9,77><10,73><11,69><12,56><13,46><14,46><15,42><16,33><17,28><18,27><21,0>
C1_70_0	0.78	8.36	3600	48	460	70	18	10	<11,159><12,116><14,107><15,95><16,91><17,86><22,83><23,76><29,74><30,0>
C1_70_1	0.8	8.22	3590	53	542	70	18	12	<12,198><13,127><14,99><18,96><19,88><21,78><23,69><24,65><26,58><28,55><30,50><32,0>
C1_70_2	0.8	12.5	3601	48	456	70	18	8	<11,185><12,123><13,106><14,93><16,84><25,79><26,62><31,0>
C1_70_3	0.8	6.58	3600	47	456	70	18	9	<10,127><11,124><13,101><14,91><15,77><17,67><25,63><29,54><31,0>
C1_70_4	0.77	4.49	3600	51	490	70	18	16	<11,149><12,115><13,100><14,95><15,93><17,90><20,81><21,78><22,70><24,70><25,68><26,66><29,63><30,60><32,55><33,0>
C1_70_5	0.84	8.39	3600	41	440	70	18	10	<9,174><10,127><11,117><12,110><13,88><21,81><23,77><27,68><28,59><29,0>

Table B.2 continued from previous page

Instance	HyperVol	K-dist(2)	CPU_time	#_spots	#_nodes	#_groups	#_tours	#_points	points
C1_70_6	0.79	6.05	3600	54	502	70	18	11	<12,144><13,128><14,122><15,103><19,88><22,80><23,76><24,73><27,63><30,57><31,0>
C1_70_7	0.79	6.16	3600	41	460	70	18	11	<10,144><11,135><12,114><13,95><14,88><15,84><16,83><17,67><26,66><27,56><31,0>
C1_70_8	0.79	5.29	3600	49	494	70	18	11	<11,168><12,166><13,109><15,98><17,97><19,96><21,92><23,87><26,75><27,66><31,0>
C1_70_9	0.78	9.38	3601	50	498	70	18	11	<11,215><12,139><13,101><14,100><15,81><22,74><23,68><25,60><29,58><31,53><32,0>
C1_90_0	0.8	6.2	3600	55	600	90	23	14	<14,183><15,161><16,146><18,128><19,123><21,111><22,104><23,100><25,100><27,94><29,93><33,83><35,75><38,0>
C1_90_1	0.8	4.81	3600	58	646	90	23	14	<14,152><16,137><18,121><19,116><23,109><25,85><29,82><30,78><32,76><33,67><34,66><37,65><39,64><41,0>
C1_90_2	0.79	7.52	3600	59	612	90	23	13	<14,215><15,143><16,139><17,118><20,116><21,109><23,100><24,98><25,90><27,84><30,81><33,65><40,0>
C1_90_3	0.82	5.02	3600	57	590	90	23	16	<12,163><13,137><16,113><19,109><21,101><22,91><25,89><27,83><28,81><29,80><31,73><33,67><36,67><38,66><39,60><41,0>
C1_90_4	0.75	5.77	3600	58	626	90	23	15	<14,192><15,156><16,146><17,132><19,117><21,107><25,100><27,98><28,83><30,81><31,76><34,75><37,65><38,65><41,0>
C1_90_5	0.8	6.44	3600	51	570	90	23	12	<14,190><15,176><16,125><18,120><20,110><21,107><22,102><25,97><27,87><28,78><34,71><38,0>
C1_90_6	0.79	8.98	3600	64	630	90	23	12	<14,215><15,168><16,142><18,131><20,116><24,104><27,104><28,94><31,89><33,83><36,73><41,0>
C1_90_7	0.78	11.16	3600	49	604	90	23	13	<12,307><13,187><14,156><17,119><20,114><24,98><26,94><28,94><30,85><31,84><35,77><36,62><39,0>
C1_90_8	0.79	6.95	3600	59	632	90	23	11	<14,157><15,136><19,123><23,101><25,92><26,88><29,81><30,70><38,69><39,67><40,0>
C1_90_9	0.79	6.24	3600	61	638	90	23	13	<15,174><16,148><17,126><18,115><21,105><22,103><24,86><27,85><28,75><34,70><35,67><38,61><40,0>
C2_5_0	0.95	28	14	11	34	5	3	2	<1,56><2,0>
C2_5_1	0.96	19.51	10	11	34	5	3	2	<1,39><2,0>
C2_5_2	0.93	27	16	11	42	5	3	2	<1,54><2,0>
C2_5_3	0.98	7.52	15	12	36	5	3	2	<1,15><2,0>
C2_5_4	0.87	41.5	23	12	36	5	3	2	<1,83><2,0>
C2_5_5	0.95	16.51	16	12	38	5	3	2	<1,33><2,0>
C2_5_6	0.97	15.51	14	12	36	5	3	2	<1,31><2,0>
C2_5_7	0.95	26.5	9	12	36	5	3	2	<1,53><2,0>
C2_5_8	0.98	11.01	6	11	30	5	3	2	<1,22><2,0>
C2_5_9	0.97	16.51	14	13	34	5	3	2	<1,33><2,0>
C2_10_0	0.88	3.84	2806	21	66	10	5	3	<2,21><3,0><4,0>
C2_10_1	0.97	18.2	2612	18	64	10	5	3	<1,103><2,6><3,0>
C2_10_2	0.88	3.01	2807	19	68	10	5	3	<2,16><3,0><4,0>
C2_10_3	0.87	20.01	2816	19	68	10	5	2	<2,40><3,0>
C2_10_4	0.86	7.69	2783	21	76	10	5	3	<2,37><3,9><4,0>
C2_10_5	0.88	3.84	2951	19	72	10	5	3	<2,21><3,0><4,0>
C2_10_6	0.89	4.53	2757	20	72	10	5	2	<2,9><3,0>
C2_10_7	0.88	6.52	2323	17	70	10	5	2	<2,13><3,0>
C2_10_8	0.88	9.01	2806	18	58	10	5	2	<2,18><3,0>
C2_10_9	0.89	3.04	2340	20	70	10	5	2	<2,6><3,0>
C2_30_0	0.82	3.71	3600	43	212	30	15	9	<5,84><6,65><7,41><9,41><10,22><11,17><12,14><13,12><14,0>
C2_30_1	0.8	4.61	3505	47	228	30	15	9	<6,96><7,76><8,42><9,28><10,26><11,20><12,15><13,7><14,0>
C2_30_2	0.87	4.62	3600	38	186	30	15	9	<4,108><5,63><6,58><7,41><8,35><9,12><10,1><11,0><13,0>
C2_30_3	0.84	15.14	3600	42	202	30	15	9	<4,287><5,85><6,58><7,35><8,33><9,14><10,8><11,7><12,0>

Table B.2 continued from previous page

Instance	HyperVol	K-dist(2)	CPU_time	#_spots	#_nodes	#_groups	#_tours	#_points	points
C2_30_4	0.83	4.09	3600	45	206	30	15	8	<5,69><6,59><7,47><8,40><9,26><10,15><12,11><14,0>
C2_30_5	0.84	3.73	3600	36	190	30	15	6	<5,68><6,54><8,31><10,25><11,0><13,0>
C2_30_6	0.83	8.72	3572	47	218	30	15	9	<5,159><6,71><7,44><9,30><10,20><11,18><12,9><13,4><14,0>
C2_30_7	0.84	3.29	3600	37	196	30	15	7	<5,54><6,39><7,26><8,17><10,16><11,2><12,0>
C2_30_8	0.83	3.72	3430	42	216	30	15	9	<5,88><6,66><7,35><8,34><10,31><11,21><12,18><13,5><14,0>
C2_30_9	0.85	1.03	3496	43	198	30	15	8	<5,56><6,53><7,23><8,22><10,6><11,4><12,1><13,0>
C2_50_0	0.83	3.31	3600	60	330	50	25	14	<8,105><9,97><10,78><11,69><12,61><14,51><15,42><16,39> <17,37><18,36><19,32><20,31><21,28><22,0>
C2_50_1	0.82	3.79	3600	61	342	50	25	14	<8,116><9,88><10,79><11,67><12,64><13,61><14,51><15,46> <16,39><17,37><19,33><20,31><21,29><22,0>
C2_50_2	0.82	3.72	3600	62	336	50	25	14	<8,113><9,98><10,84><11,76><12,76><13,63><14,54><16,47> <17,44><18,42><19,39><20,38><21,34><22,0>
C2_50_3	0.82	3.8	3600	60	336	50	25	15	<7,128><8,104><9,95><10,80><11,73><12,68><13,63><14,56> <15,51><16,49><17,46><18,44><19,30><22,29><23,0>
C2_50_4	0.79	5.52	3600	62	340	50	25	12	<8,143><9,95><10,88><11,82><12,71><15,53><17,48><19,42> <20,35><21,35><22,28><24,0>
C2_50_5	0.83	4.01	3600	55	324	50	25	14	<8,121><9,87><10,79><11,78><12,64><13,51><15,45><16,40> <17,36><18,34><19,34><20,32><21,29><22,0>
C2_50_6	0.82	5.36	3600	66	356	50	25	10	<8,102><10,84><12,68><14,58><15,53><16,50><18,47><20,43> <22,36><23,0>
C2_50_7	0.81	5.74	3600	54	334	50	25	12	<8,146><9,88><12,71><13,63><14,56><15,50><16,50><17,43> <18,43><19,33><21,30><22,0>
C2_50_8	0.8	8.96	3577	62	356	50	25	13	<8,241><9,103><10,84><11,75><12,69><14,62><16,50><17,45> <18,36><19,35><20,31><22,30><23,0>
C2_50_9	0.82	4.41	3600	64	344	50	25	12	<8,125><9,91><10,79><11,74><12,71><13,63><14,43><16,41> <18,32><19,29><20,27><22,0>
C2_70_0	0.81	6.79	3600	76	466	70	35	11	<11,165><12,113><13,112><14,99><15,94><18,92><19,91><20,72> <26,72><28,68><30,0>
C2_70_1	0.8	9.65	3601	80	494	70	35	10	<11,190><12,146><13,110><15,88><18,82><21,80><23,75><24,60> <29,51><32,0>
C2_70_2	0.81	5.32	3600	77	464	70	35	12	<10,166><11,158><12,106><13,95><14,85><17,79><21,76><23,72> <25,65><28,62><30,60><31,0>
C2_70_3	0.79	8.42	3601	74	476	70	35	12	<11,205><12,109><14,103><15,97><16,84><23,84><26,71><27,69> <28,69><29,66><31,58><32,0>
C2_70_4	0.83	6.35	3600	76	458	70	35	12	<10,166><12,108><14,102><15,94><16,92><17,81><24,75><25,73> <26,73><27,62><28,61><29,0>
C2_70_5	0.8	12.19	3600	74	480	70	35	10	<11,244><12,131><13,114><15,91><16,88><21,88><22,75><26,67> <29,58><32,0>
C2_70_6	0.8	8.76	3600	80	480	70	35	10	<11,169><12,114><14,104><18,92><20,87><24,86><25,83><26,76> <30,67><31,0>
C2_70_7	0.79	8.69	3601	74	500	70	35	9	<11,154><13,99><17,92><20,84><22,76><24,66><27,60><28,60><32,0>
C2_70_8	0.78	6.94	3589	80	502	70	35	12	<12,193><13,132><16,130><17,104><18,102><20,91><24,85><25,77> <27,72><28,71><29,63><32,0>
C2_70_9	0.8	9.13	3600	78	484	70	35	10	<11,172><12,123><14,101><16,84><20,74><25,73><26,68><29,66> <30,62><31,0>
C2_90_0	0.79	6.37	3600	83	618	90	45	16	<14,210><15,185><16,165><17,142><18,130><19,111><22,108><24,102> <25,96><26,90><27,89><32,83><33,76><34,68><39,63><40,0>
C2_90_1	0.77	4.05	3600	96	662	90	45	17	<15,159><16,152><18,134><21,126><22,120><23,115><24,104><25,103> <26,92><33,91><34,82><35,82><36,79><37,75><38,70><39,68><41,0>
C2_90_2	0.81	4.69	3600	91	576	90	45	15	<13,167><14,157><15,155><17,135><19,118><20,118><21,116><24,102> <25,95><27,93><28,84><29,79><34,76><35,63><38,0>

Table B.2 continued from previous page

Instance	HyperVol	K-dist(2)	CPU_time	#_spots	#_nodes	#_groups	#_tours	#_points	points
C2_90_3	0.79	5.51	3600	91	600	90	45	15	<14,177><15,152><17,142><18,130><19,129><21,118><22,114><23,109><25,105><27,100><28,96><29,85><32,81><35,67><40,0>
C2_90_4	0.8	4.06	3600	90	624	90	45	18	<14,166><15,160><16,146><17,144><18,134><19,121><20,108><24,100><28,94><30,83><31,81><34,78><35,77><37,73><38,71><39,67><40,60><41,0>
C2_90_5	0.8	7.68	3600	83	604	90	45	17	<13,286><15,174><16,146><17,129><18,123><19,120><20,113><21,111><23,103><24,100><27,86><32,82><34,81><35,68><36,67><37,63><40,0>
C2_90_6	0.78	4.2	3600	89	630	90	45	18	<15,209><16,194><17,144><18,135><19,128><20,123><23,114><24,112><25,106><26,105><27,91><32,90><33,89><34,88><36,75><37,74><39,73><41,0>
C2_90_7	0.76	5.84	3600	86	620	90	45	16	<14,211><15,190><16,147><18,138><19,132><22,121><23,113><25,109><26,100><28,92><31,92><32,85><35,83><38,82><40,75><41,0>
C2_90_8	0.75	6.59	3600	96	652	90	45	15	<16,220><17,159><18,139><19,130><21,129><22,109><25,106><29,96><30,88><31,87><34,84><37,75><38,71><39,63><42,0>
C2_90_9	0.8	5.63	3600	88	626	90	45	15	<14,183><15,154><17,112><20,108><21,104><23,101><26,94><28,81><30,80><31,77><33,77><34,74><37,71><38,67><40,0>
C3_5_0	0.95	26	13	13	34	5	4	2	<1,52><2,0>
C3_5_1	0.95	28	9	12	30	5	4	2	<1,56><2,0>
C3_5_2	0.91	37.5	17	13	38	5	4	2	<1,75><2,0>
C3_5_3	0.92	36.5	29	15	36	5	4	2	<1,73><2,0>
C3_5_4	0.75	1.12	66	15	38	5	4	2	<2,2><3,0>
C3_5_5	0.95	23.51	22	14	36	5	4	2	<1,47><2,0>
C3_5_6	0.98	9.01	6	13	28	5	4	2	<1,18><2,0>
C3_5_7	0.93	39.5	19	13	34	5	4	2	<1,79><2,0>
C3_5_8	0.95	20.51	10	14	32	5	4	2	<1,41><2,0>
C3_5_9	0.93	40	30	15	38	5	4	2	<1,80><2,0>
C3_10_0	0.86	10.35	2816	29	68	10	8	3	<2,53><3,9><4,0>
C3_10_1	0.86	10.87	2760	24	70	10	8	3	<2,60><3,5><4,0>
C3_10_2	0.88	16.51	2773	23	66	10	8	2	<2,33><3,0>
C3_10_3	0.87	6.31	2779	25	68	10	8	3	<2,36><3,1><4,0>
C3_10_4	0.86	10.52	2775	28	72	10	8	3	<2,52><3,11><4,0>
C3_10_5	0.86	8.14	2936	23	66	10	8	3	<2,47><3,1><4,0>
C3_10_6	0.88	6.14	2758	29	70	10	8	3	<2,35><3,1><4,0>
C3_10_7	0.87	7.54	2752	23	68	10	8	3	<2,41><3,4><4,0>
C3_10_8	0.88	6.89	2787	24	68	10	8	3	<2,38><3,3><4,0>
C3_10_9	0.87	9.64	2767	26	68	10	8	3	<2,56><3,1><4,0>
C3_30_0	0.84	5.25	3478	55	202	30	23	7	<5,97><6,62><7,50><8,35><9,33><10,4><13,0>
C3_30_1	0.83	3.17	3536	58	216	30	23	9	<5,79><6,75><7,56><8,40><9,34><10,26><11,18><12,1><14,0>
C3_30_2	0.84	5.51	3532	59	204	30	23	8	<5,96><6,62><7,38><9,34><10,29><11,14><12,0><13,0>
C3_30_3	0.84	1.74	3596	57	204	30	23	9	<5,77><6,75><7,47><8,42><9,24><10,24><11,15><12,2><14,0>
C3_30_4	0.81	5.58	3600	58	200	30	23	9	<5,111><6,77><7,55><8,52><9,38><10,22><11,11><13,0><14,0>
C3_30_5	0.85	5.2	3600	51	190	30	23	6	<5,60><7,39><8,25><9,13><10,12><13,0>
C3_30_6	0.84	4.69	3569	64	212	30	23	10	<5,95><6,68><7,54><8,45><9,32><10,29><11,24><12,20><13,13><14,0>
C3_30_7	0.83	4.86	3600	49	202	30	23	8	<5,84><6,67><7,48><8,35><10,26><11,10><13,0><14,0>
C3_30_8	0.82	11.23	3370	59	216	30	23	9	<5,210><6,75><7,50><9,38><10,24><11,19><12,17><13,4><14,0>
C3_30_9	0.8	3.61	3600	61	216	30	23	9	<6,68><7,59><8,50><9,36><10,21><11,15><12,13><13,8><14,0>
C3_50_0	0.82	3.57	3600	79	346	50	38	15	<8,122><9,96><10,83><11,75><12,70><13,56><14,55><15,50><16,47><17,38><18,36><19,35><20,34><21,30><22,0>
C3_50_1	0.81	3.16	3600	86	360	50	38	13	<9,102><11,72><12,72><13,62><14,60><15,55><16,44><17,43><18,35><19,35><20,34><22,28><23,0>

Table B.2 continued from previous page

Instance	HyperVol	K-dist(2)	CPU_time	#_spots	#_nodes	#_groups	#_tours	#_points	points
C3_50_2	0.84	7.01	3601	81	324	50	38	13	<7,191><9,96><10,78><12,67><13,59><14,54><15,45><16,44><17,35><18,33><20,30><21,28><22,0>
C3_50_3	0.81	5.34	3600	82	336	50	38	13	<8,151><9,107><10,88><11,87><12,74><13,67><14,56><15,51><17,43><18,37><19,32><22,31><23,0>
C3_50_4	0.84	5.11	3600	81	328	50	38	15	<7,167><8,121><9,101><10,81><11,76><12,66><13,65><14,59><15,51><16,46><17,43><18,43><19,34><21,34><22,0>
C3_50_5	0.82	6.12	3513	77	344	50	38	14	<8,174><9,94><10,80><11,76><12,72><13,62><14,55><15,51><16,47><17,43><18,39><19,38><21,35><23,0>
C3_50_6	0.82	5.85	3600	82	362	50	38	10	<8,127><9,99><12,78><13,75><14,57><16,50><17,45><19,39><21,35><23,0>
C3_50_7	0.8	4.92	3601	78	344	50	38	11	<8,121><10,94><12,75><13,72><14,60><16,54><17,38><20,38><21,37><22,34><23,0>
C3_50_8	0.79	3.63	3493	85	360	50	38	14	<9,114><10,101><11,78><12,78><13,71><14,64><15,60><16,56><18,53><19,49><20,48><21,39><22,30><23,0>
C3_50_9	0.81	4.99	3600	82	350	50	38	12	<8,132><9,99><10,85><11,73><12,71><13,65><14,64><15,50><17,34><20,32><22,30><23,0>
C3_70_0	0.8	7.35	3600	89	482	70	53	11	<11,170><12,123><13,111><14,104><15,86><20,82><22,80><24,75><26,64><30,62><32,0>
C3_70_1	0.79	6.39	3600	102	486	70	53	11	<11,150><12,140><13,117><14,106><15,88><20,87><23,87><24,76><25,68><31,67><32,0>
C3_70_2	0.8	11.52	3600	99	464	70	53	7	<11,150><13,106><16,89><21,82><23,74><28,69><30,0>
C3_70_3	0.79	9.07	3601	99	470	70	53	10	<11,184><13,128><14,100><15,96><16,80><22,78><27,71><29,67><31,63><32,0>
C3_70_4	0.82	7.28	3600	96	466	70	53	12	<10,177><11,152><12,127><14,110><16,94><19,91><21,90><23,81><25,80><28,73><29,68><30,0>
C3_70_5	0.8	9.58	3591	93	476	70	53	11	<11,209><12,135><13,114><14,104><15,92><21,79><25,73><27,71><28,64><30,58><32,0>
C3_70_6	0.79	8.37	3600	98	480	70	53	8	<12,131><13,111><14,104><15,102><21,88><24,75><28,62><32,0>
C3_70_7	0.8	5.38	3601	97	484	70	53	11	<11,135><12,123><14,119><15,92><20,84><23,78><26,71><27,68><29,62><30,56><31,0>
C3_70_8	0.76	6.75	3601	103	494	70	53	12	<12,162><13,132><14,110><18,105><19,93><23,92><24,86><26,80><27,77><28,68><30,60><32,0>
C3_70_9	0.81	8.98	3600	101	486	70	53	10	<11,173><12,134><13,103><16,83><20,82><21,78><23,70><27,62><30,57><31,0>
C3_90_0	0.79	8.12	3600	105	616	90	68	13	<14,225><15,179><16,148><18,137><19,118><22,115><24,97><27,85><29,80><37,76><38,67><39,63><40,0>
C3_90_1	0.78	5.58	3600	117	628	90	68	15	<14,180><16,156><17,131><19,131><20,126><21,121><23,108><26,99><28,87><31,83><32,82><33,73><38,67><40,66><41,0>
C3_90_2	0.8	7	3600	112	598	90	68	16	<14,233><15,169><16,134><20,132><22,125><23,111><25,109><26,104><28,101><29,101><30,93><31,84><34,80><36,75><37,71><39,0>
C3_90_3	0.8	4.84	3601	109	610	90	68	16	<14,171><15,145><17,144><19,128><20,125><21,122><22,112><24,110><25,105><27,100><29,98><30,92><31,86><32,80><33,70><40,0>
C3_90_4	0.8	4.96	3600	109	618	90	68	16	<14,176><15,156><16,147><17,130><19,123><20,117><22,113><23,106><26,101><27,98><30,85><32,81><34,80><35,70><39,69><40,0>
C3_90_5	0.78	4.72	3600	112	608	90	68	14	<15,155><18,138><20,137><23,128><24,123><25,118><28,98><30,93><34,88><35,87><36,86><37,75><38,70><40,0>
C3_90_6	0.78	5.01	3600	112	630	90	68	16	<15,178><16,147><18,131><20,120><22,106><27,99><29,98><30,84><31,84><32,78><34,78><35,76><38,67><39,62><40,60><41,0>
C3_90_7	0.79	4.94	3600	115	616	90	68	15	<14,171><15,142><16,140><18,134><20,121><23,96><27,91><29,87><30,82><34,77><35,76><37,74><38,67><39,67><41,0>
C3_90_8	0.77	7.64	3600	113	626	90	68	13	<14,208><16,171><17,129><18,114><22,112><23,100><26,96><29,92><32,81><35,76><37,68><40,68><41,0>

Table B.2 continued from previous page

Instance	HyperVol	K-dist(2)	CPU_time	#_spots	#_nodes	#_groups	#_tours	#_points	points
C3_90_9	0.78	11.66	3600	117	626	90	68	12	<14,313><15,156><17,152><19,121><20,114><24,105><25,104><28,100> <29,82><32,74><37,73><40,0>
C4_5_0	0.92	42	21	16	34	5	5	2	<1,84><2,0>
C4_5_1	0.95	28.5	14	15	32	5	5	2	<1,57><2,0>
C4_5_2	0.88	52	19	15	34	5	5	2	<1,104><2,0>
C4_5_3	0.92	41.5	46	16	34	5	5	2	<1,83><2,0>
C4_5_4	0.75	1.12	90	18	38	5	5	2	<2,2><3,0>
C4_5_5	0.93	31.5	34	16	34	5	5	2	<1,63><2,0>
C4_5_6	0.96	25	17	17	34	5	5	2	<1,50><2,0>
C4_5_7	nan	0	30	15	34	5	5	1	<2,0>
C4_5_8	0.94	24.51	15	15	30	5	5	2	<1,49><2,0>
C4_5_9	0.94	34	27	17	36	5	5	2	<1,68><2,0>
C4_10_0	0.85	14.04	2836	33	70	10	10	3	<2,80><3,4><4,0>
C4_10_1	0.86	14.85	2756	29	70	10	10	3	<2,79><3,10><4,0>
C4_10_2	0.87	10.7	2775	27	64	10	10	3	<2,59><3,5><4,0>
C4_10_3	0.86	9.08	2814	29	68	10	10	3	<2,52><3,2><4,0>
C4_10_4	0.85	14.34	2823	32	72	10	10	3	<2,69><3,17><4,0>
C4_10_5	0.87	8.38	2952	26	62	10	10	3	<2,46><3,4><4,0>
C4_10_6	0.87	9.81	2752	33	72	10	10	3	<2,57><3,1><4,0>
C4_10_7	0.87	9.56	2786	26	66	10	10	3	<2,54><3,3><4,0>
C4_10_8	0.85	19.84	2729	28	68	10	10	3	<2,101><3,18><4,0>
C4_10_9	0.86	12.03	2745	30	68	10	10	3	<2,66><3,6><4,0>
C4_30_0	0.83	4.96	3461	68	202	30	30	8	<5,88><6,58><7,46><8,46><9,28><10,21><11,13><13,0>
C4_30_1	0.82	4.64	3527	71	212	30	30	8	<5,97><6,81><7,45><9,44><10,35><11,21><12,7><14,0>
C4_30_2	0.83	3.9	3600	72	202	30	30	7	<5,92><6,74><9,31><10,31><11,19><12,2><13,0>
C4_30_3	0.82	3.58	3600	66	200	30	30	9	<5,92><6,84><7,61><8,34><9,30><11,27><12,12><13,12><14,0>
C4_30_4	0.81	6.25	3600	70	202	30	30	8	<5,108><6,72><7,66><8,46><9,35><10,28><11,13><14,0>
C4_30_5	0.84	4.61	3446	65	200	30	30	8	<5,84><6,58><7,50><8,47><9,38><10,19><12,5><13,0>
C4_30_6	0.81	3.83	3566	73	210	30	30	8	<6,67><7,60><8,52><9,42><10,27><11,23><12,22><14,0>
C4_30_7	0.83	4.25	3600	63	206	30	30	10	<5,90><6,56><7,52><8,40><9,38><10,33><11,26><12,18><13,10><14,0>
C4_30_8	0.8	4.16	3350	73	218	30	30	9	<6,76><7,62><8,46><9,44><10,36><11,28><12,20><13,14><14,0>
C4_30_9	0.81	13.3	3600	70	208	30	30	7	<5,188><6,82><8,50><9,31><11,18><13,1><14,0>
C4_50_0	0.82	3.62	3600	88	340	50	50	14	<8,116><9,86><11,78><12,71><13,66><14,58><15,54><16,51> <17,48><18,35><19,34><21,27><22,25><23,0>
C4_50_1	0.81	8.56	3600	100	354	50	50	10	<8,180><9,103><10,100><11,83><13,66><15,56><17,53><19,43> <20,30><23,0>
C4_50_2	0.82	2.97	3600	96	332	50	50	13	<8,112><9,107><10,79><11,74><12,73><13,62><14,60><15,55> <16,49><17,42><19,32><20,28><22,0>
C4_50_3	0.83	4.39	3601	97	332	50	50	13	<8,120><9,93><10,85><11,75><12,68><13,59><14,52><16,43> <18,41><19,35><20,33><21,32><22,0>
C4_50_4	0.82	3.68	3600	95	338	50	50	13	<8,127><9,123><10,89><11,78><12,68><14,61><15,51><17,44> <18,44><19,39><20,36><21,30><23,0>
C4_50_5	0.82	5.13	3512	90	340	50	50	16	<8,182><9,96><10,87><11,81><12,77><13,61><14,58><15,54> <16,53><17,53><18,42><19,37><20,34><21,32><22,28><23,0>
C4_50_6	0.82	6.22	3600	93	350	50	50	11	<8,150><9,114><11,81><12,76><14,61><16,55><17,53><18,44> <20,34><22,32><23,0>
C4_50_7	0.81	11.31	3600	90	338	50	50	12	<8,274><9,101><10,83><11,74><13,67><15,56><16,50><17,46> <18,39><20,32><21,29><22,0>
C4_50_8	0.79	4.74	3508	100	354	50	50	13	<9,130><10,93><12,72><14,67><15,58><16,51><17,47><18,47> <19,39><20,36><21,34><22,33><24,0>
C4_50_9	0.81	8.96	3600	95	350	50	50	14	<8,256><9,102><10,85><11,70><12,66><13,60><14,56><15,53> <16,51><17,47><18,38><19,33><20,31><22,0>

Table B.2 continued from previous page

Instance	HyperVol	K-dist(2)	CPU_time	#_spots	#_nodes	#_groups	#_tours	#_points	points
C4_70_0	0.79	4.51	3600	107	480	70	70	13	<12,126><13,114><14,100><16,96><17,95><18,94><20,92><21,78><23,77><24,69><28,64><30,56><31,0>
C4_70_1	0.79	7.31	3600	118	484	70	70	11	<11,175><12,130><14,118><16,95><18,94><20,88><21,88><22,86><24,71><28,61><32,0>
C4_70_2	0.79	5.85	3601	115	468	70	70	13	<11,169><12,139><13,130><14,104><16,95><20,90><21,84><24,83><25,73><28,66><29,65><30,64><31,0>
C4_70_3	0.79	4.99	3600	112	474	70	70	13	<11,154><12,131><13,119><17,108><18,108><19,105><21,85><24,82><25,75><27,74><29,63><30,62><31,0>
C4_70_4	0.81	5.44	3600	110	472	70	70	12	<11,147><12,143><13,126><14,121><15,108><16,97><19,86><22,84><25,75><26,68><29,61><31,0>
C4_70_5	0.79	5.58	3565	112	478	70	70	14	<12,165><13,145><14,114><15,97><16,94><21,88><23,82><24,76><26,76><27,75><28,72><30,66><31,58><32,0>
C4_70_6	0.79	5.09	3601	113	484	70	70	12	<12,135><13,129><14,106><16,91><22,85><23,80><24,77><25,76><26,71><28,68><30,66><32,0>
C4_70_7	0.8	6.93	3600	116	482	70	70	12	<11,165><12,117><16,93><19,88><20,84><23,78><24,71><27,70><28,65><29,61><30,60><31,0>
C4_70_8	0.77	6.33	3600	115	486	70	70	11	<12,143><14,129><15,106><19,98><20,93><22,86><23,76><26,68><30,68><31,60><32,0>
C4_70_9	0.79	8	3600	117	486	70	70	12	<11,190><12,140><13,112><15,103><19,97><22,87><24,78><27,73><28,71><29,69><30,68><31,0>
C4_90_0	0.79	5.96	3600	125	620	90	90	14	<15,191><16,145><17,134><20,113><22,112><25,98><26,96><27,95><28,85><32,81><35,79><38,72><39,68><40,0>
C4_90_1	0.78	7.31	3600	129	616	90	90	15	<14,234><15,154><16,133><17,132><19,131><20,118><23,102><27,100><29,92><30,88><32,84><33,84><35,75><37,70><41,0>
C4_90_2	0.78	10.54	3600	128	614	90	90	12	<14,256><15,203><16,154><19,133><22,113><28,105><29,88><31,86><33,79><36,79><37,78><39,0>
C4_90_3	0.78	8.58	3600	129	618	90	90	16	<14,300><15,168><16,161><18,146><19,126><22,123><23,111><26,111><27,100><28,92><31,92><32,79><36,78><37,75><38,73><40,0>
C4_90_4	0.8	5.33	3600	122	614	90	90	15	<14,180><16,140><17,131><18,126><19,116><21,112><22,109><23,108><25,100><30,82><32,78><37,71><38,70><39,67><40,0>
C4_90_5	0.79	8.3	3600	127	616	90	90	14	<14,242><15,185><17,137><19,132><20,118><25,107><27,95><28,93><31,85><33,82><35,76><37,75><39,74><40,0>
C4_90_6	0.79	6.67	3600	126	612	90	90	16	<14,220><15,194><16,157><17,140><18,130><20,120><22,112><25,110><26,101><27,94><31,94><32,88><33,81><34,75><37,66><40,0>
C4_90_7	0.79	6.44	3600	128	626	90	90	13	<14,191><15,166><17,150><18,122><20,117><22,111><25,102><26,100><30,83><34,73><37,73><38,72><41,0>
C4_90_8	0.79	3.62	3600	128	618	90	90	19	<14,208><15,197><16,152><17,146><18,136><19,133><20,114><23,112><24,106><26,98><27,98><28,97><30,82><32,80><33,78><34,77><36,69><37,66><40,0>
C4_90_9	0.79	6.57	3600	125	616	90	90	15	<13,224><15,157><16,155><18,138><20,132><22,124><24,122><25,120><26,103><27,98><29,93><31,86><34,76><37,71><39,0>

B.3 AUGMECON-CP2

TABLE B.3: Results of the AUGMECON-CP2

Instance	HyperVol	K-dist(2)	CPU_time	#_spots	#_nodes	#_groups	#_tours	#_points	points
C0_5_0	0.97	8.51	4	7	40	5	1	2	<1,17><2,0>
C0_5_1	0.96	10.01	4	7	40	5	1	2	<1,20><2,0>
C0_5_2	0.96	3.17	43	8	50	5	1	3	<1,17><2,0><3,0>
C0_5_3	0.99	3.54	11	7	40	5	1	2	<1,7><2,0>
C0_5_4	0.96	9.51	5	7	40	5	1	2	<1,19><2,0>
C0_5_5	0.99	3.04	5	8	40	5	1	2	<1,6><2,0>
C0_5_6	1	0.71	1	6	20	5	1	2	<1,1><2,0>
C0_5_7	0.95	14.01	12	8	40	5	1	2	<1,28><2,0>
C0_5_8	0.99	2.55	2	7	30	5	1	2	<1,5><2,0>
C0_5_9	0.97	6.52	3	8	40	5	1	2	<1,13><2,0>
C0_10_0	0.94	13.37	2104	12	80	10	1	3	<1,75><2,5><3,0>
C0_10_1	0.96	7.41	1904	10	80	10	1	3	<1,42><2,2><3,0>
C0_10_2	0.88	1.68	2606	12	100	10	1	3	<2,8><3,0><4,0>
C0_10_3	0.96	12.17	3050	10	80	10	1	3	<1,71><2,0><3,0>
C0_10_4	0.94	10.7	1939	10	80	10	1	3	<1,58><2,6><3,0>
C0_10_5	0.95	11.56	2157	11	80	10	1	3	<1,66><2,3><3,0>
C0_10_6	1	8.02	6	9	40	10	1	2	<1,16><2,0>
C0_10_7	0.89	2.55	2354	10	80	10	1	2	<2,5><3,0>
C0_10_8	0.95	15.81	908	10	60	10	1	3	<1,93><2,1><3,0>
C0_10_9	0.88	3.04	2310	11	80	10	1	2	<2,6><3,0>
C0_30_0	0.9	3.76	3332	13	240	30	1	6	<3,51><5,18><6,16><7,2><9,1><12,0>
C0_30_1	0.89	4.61	3393	13	240	30	1	5	<3,41><4,29><5,15><9,13><10,0>
C0_30_2	0.73	3.51	3483	14	300	30	1	6	<5,66><6,58><10,16><13,13><14,5><22,0>
C0_30_3	0.89	15.56	3307	13	240	30	1	5	<3,152><4,82><5,21><7,1><9,0>
C0_30_4	0.88	3.72	3436	13	240	30	1	8	<3,66><4,28><5,27><6,20><7,16><9,4><11,0><14,0>
C0_30_5	0.88	6.72	3275	13	240	30	1	6	<3,85><4,55><5,48><6,26><8,16><9,0>
C0_30_6	0.96	10.51	3600	11	120	30	1	2	<2,21><3,0>
C0_30_7	0.86	7.97	3360	13	240	30	1	6	<4,91><5,46><6,20><8,4><10,0><12,0>
C0_30_8	0.93	2.63	3102	12	180	30	1	4	<3,19><4,4><5,3><6,0>
C0_30_9	0.88	7.43	3379	13	240	30	1	7	<3,97><4,46><5,27><6,12><7,8><8,2><13,0>
C0_50_0	0.79	5.79	3544	13	400	50	1	8	<8,86><9,47><10,34><14,27><18,22><20,21><21,13><25,0>
C0_50_1	0.76	3.47	3554	13	400	50	1	9	<8,95><9,90><10,42><12,28><18,25><19,19><22,18><25,11><28,0>
C0_50_2	0.61	3.23	3675	14	500	50	1	10	<11,88><12,83><14,82><15,70><16,66><18,46><23,40><24,14><33,12><37,0>
C0_50_3	0.83	2.32	3525	13	400	50	1	13	<7,72><9,72><10,56><11,52><12,43><13,34><14,28><15,24><18,21><19,15><20,11><23,7><24,0>
C0_50_4	0.69	4.49	3597	13	400	50	1	10	<9,84><10,68><11,58><12,50><16,43><17,29><22,24><29,24><32,7><38,0>
C0_50_5	0.77	5.56	3546	13	400	50	1	8	<8,96><10,65><11,51><12,50><14,44><15,25><20,20><25,0>
C0_50_6	0.94	31	3259	11	200	50	1	2	<4,62><5,0>

Table B.3 continued from previous page

Instance	HyperVol	K-dist(2)	CPU_time	#_spots	#_nodes	#_groups	#_tours	#_points	points
C0_50_7	0.8	2.82	3536	13	400	50	1	11	<7,71><8,60><10,55><11,47><14,41><15,23><18,18><19,15><20,15><22,10><25,0>
C0_50_8	0.85	5.22	3288	12	300	50	1	9	<5,127><6,127><7,98><8,52><10,39><12,30><14,21><18,11><21,0>
C0_50_9	0.74	3.68	3565	13	400	50	1	9	<9,72><10,54><13,46><14,41><16,27><17,26><18,18><22,18><27,0>
C0_70_0	0.73	3.27	3758	13	560	70	1	15	<11,98><13,74><15,71><16,65><17,54><20,54><21,48><22,42><23,40><27,39><29,39><30,37><32,34><33,30><35,0>
C0_70_1	0.7	3.42	3771	13	560	70	1	13	<12,112><13,111><14,85><15,68><16,63><18,59><19,56><20,44><26,40><29,34><34,29><36,28><38,0>
C0_70_2	0.54	2.93	4252	14	700	70	1	17	<16,100><18,86><20,77><22,75><23,68><25,65><26,60><27,58><28,53><33,49><34,47><35,46><37,40><38,32><47,31><48,12><52,0>
C0_70_3	0.77	3.79	3750	13	560	70	1	15	<10,134><11,133><12,119><13,96><15,88><16,78><17,75><18,71><19,66><20,61><21,57><22,51><25,43><29,39><33,0>
C0_70_4	0.65	2.41	3786	13	560	70	1	20	<13,92><14,75><15,69><16,68><17,62><20,58><21,57><22,50><24,45><25,43><27,42><29,37><30,36><31,34><35,30><37,25><41,24><46,21><52,20><53,0>
C0_70_5	0.68	7.54	3760	13	560	70	1	9	<14,129><15,96><18,82><24,73><26,72><27,66><28,56><34,50><35,0>
C0_70_6	0.9	25.25	3272	11	280	70	1	6	<6,296><7,79><11,42><12,30><18,12><21,0>
C0_70_7	0.71	3.04	3767	13	560	70	1	13	<13,103><14,98><16,72><18,71><19,64><21,57><22,56><24,51><25,48><26,40><30,37><33,33><35,0>
C0_70_8	0.77	12.48	3584	12	420	70	1	4	<12,84><19,73><20,44><28,0>
C0_70_9	0.68	3.87	3778	13	560	70	1	12	<12,91><16,72><18,68><19,60><21,54><23,48><25,40><28,36><31,34><32,34><35,28><39,0>
C0_90_0	0.72	3.79	4288	13	720	90	1	16	<16,129><17,88><18,80><19,79><20,63><22,60><23,57><24,55><25,50><29,48><31,46><32,44><34,41><35,36><39,30><45,0>
C0_90_1	0.68	3.26	4294	13	720	90	1	17	<17,104><18,84><19,78><20,71><21,68><23,68><24,65><25,60><26,59><28,58><29,55><30,51><31,42><36,36><42,35><43,28><50,0>
C0_90_2	0.53	3.55	5554	14	900	90	1	14	<25,94><27,79><30,75><32,71><34,60><37,59><39,57><40,57><41,56><42,54><44,46><50,22><65,20><67,0>
C0_90_3	0.76	3.77	4272	13	720	90	1	16	<14,141><15,133><16,109><18,97><19,85><20,83><22,73><25,68><27,57><29,54><30,51><32,48><38,46><41,40><42,36><43,0>
C0_90_4	0.63	2.95	4282	13	720	90	1	16	<19,81><21,70><24,67><25,62><26,56><28,50><32,48><33,41><35,38><39,34><41,31><53,27><58,27><64,25><67,24><68,0>
C0_90_5	0.67	4.48	4279	13	720	90	1	12	<18,130><20,121><21,118><22,100><24,96><26,83><28,77><30,76><32,65><34,60><38,53><45,0>
C0_90_6	0.9	9.55	3238	11	360	90	1	9	<8,271><9,232><10,89><11,59><14,36><16,21><17,20><18,11><26,0>
C0_90_7	0.71	3.25	4282	13	720	90	1	14	<16,98><18,96><20,91><21,89><22,81><24,67><26,66><27,63><28,58><29,54><32,50><37,44><38,37><45,0>
C0_90_8	0.72	6.96	3708	12	540	90	1	11	<15,164><18,115><24,115><25,92><26,92><29,90><30,85><31,84><35,75><37,67><38,0>
C0_90_9	0.68	3.3	4303	13	720	90	1	15	<16,106><17,94><18,88><19,84><20,79><21,73><22,68><23,64><24,63><26,62><27,56><29,46><39,32><41,30><50,0>
C1_5_0	0.96	12.51	3	9	34	5	2	2	<1,25><2,0>
C1_5_1	0.98	7.52	3	9	36	5	2	2	<1,15><2,0>
C1_5_2	0.97	19.01	4	9	38	5	2	2	<1,38><2,0>
C1_5_3	0.98	8.02	8	10	40	5	2	2	<1,16><2,0>
C1_5_4	0.88	35	10	9	36	5	2	2	<1,70><2,0>
C1_5_5	0.98	8.02	3	10	34	5	2	2	<1,16><2,0>
C1_5_6	0.99	5.52	0	9	28	5	2	2	<1,11><2,0>
C1_5_7	0.92	26	16	10	40	5	2	2	<1,52><2,0>
C1_5_8	0.98	7.52	3	9	30	5	2	2	<1,15><2,0>
C1_5_9	0.99	8.51	1	10	32	5	2	2	<1,17><2,0>
C1_10_0	0.88	6.52	2682	16	64	10	3	2	<2,13><3,0>

Table B.3 continued from previous page

Instance	HyperVol	K-dist(2)	CPU_time	#_spots	#_nodes	#_groups	#_tours	#_points	points
C1_10_1	0.89	3.04	2026	14	66	10	3	2	<2,6><3,0>
C1_10_2	0.87	6.34	2900	15	84	10	3	3	<2,36><3,0><4,0>
C1_10_3	0.95	17.72	3600	15	74	10	3	3	<1,103><2,3><3,0>
C1_10_4	0.86	7.58	2765	15	72	10	3	3	<2,43><3,2><4,0>
C1_10_5	0.88	3.54	2155	15	78	10	3	2	<2,7><3,0>
C1_10_6	0.89	1.12	2214	15	64	10	3	2	<2,2><3,0>
C1_10_7	0.89	1.12	1906	14	70	10	3	2	<2,2><3,0>
C1_10_8	0.88	5.02	2009	14	60	10	3	2	<2,10><3,0>
C1_10_9	0.88	2.67	2623	16	70	10	3	3	<2,14><3,0><4,0>
C1_30_0	0.83	4.47	3334	30	204	30	8	9	<4,121><5,97><6,74><7,72><8,25><11,14><12,11><13,6><14,0>
C1_30_1	0.82	5.08	3337	27	216	30	8	8	<5,111><6,104><8,65><9,43><10,26><11,25><12,13><14,0>
C1_30_2	0.86	16.79	3308	25	188	30	8	6	<4,200><5,83><6,36><8,21><10,2><13,0>
C1_30_3	0.86	3.66	3309	28	194	30	8	8	<4,128><5,118><6,44><7,40><8,30><9,14><10,6><11,0>
C1_30_4	0.84	10.96	3357	31	214	30	8	7	<4,151><5,68><6,50><8,37><9,25><12,7><15,0>
C1_30_5	0.81	13.76	3332	25	192	30	8	7	<4,193><5,88><6,75><7,73><9,53><10,36><13,0>
C1_30_6	0.84	7.25	3312	31	206	30	8	7	<5,120><6,58><7,53><8,39><10,11><11,7><12,0>
C1_30_7	0.84	2.47	3335	26	202	30	8	9	<4,147><5,146><6,58><8,50><9,27><10,22><11,10><12,4><14,0>
C1_30_8	0.84	7.86	3314	26	214	30	8	6	<4,130><5,121><6,103><7,48><12,41><13,0>
C1_30_9	0.83	13.05	3312	28	204	30	8	7	<5,178><6,69><7,37><9,22><11,12><12,8><13,0>
C1_50_0	0.79	6.69	3541	39	342	50	13	11	<7,194><8,188><10,131><12,90><15,62><17,56><18,50><19,48><21,46><22,30><23,0>
C1_50_1	0.74	12.04	3561	42	370	50	13	7	<11,180><12,140><13,118><16,75><17,66><18,51><24,0>
C1_50_2	0.78	8.73	3499	35	326	50	13	9	<10,153><11,88><14,75><15,63><17,56><18,53><19,48><20,42><22,0>
C1_50_3	0.79	14.64	3512	38	348	50	13	8	<8,227><9,146><11,104><12,70><16,54><19,49><20,41><21,0>
C1_50_4	0.76	7.38	3534	40	344	50	13	9	<9,142><10,124><11,118><12,108><13,82><14,61><20,58><21,47><23,0>
C1_50_5	0.82	8.68	3475	32	300	50	13	11	<7,206><8,175><9,154><11,118><12,80><13,67><14,59><15,43><18,34><19,25><21,0>
C1_50_6	0.8	7.58	3528	43	350	50	13	8	<8,194><11,191><12,105><14,80><16,61><18,54><19,40><21,0>
C1_50_7	0.75	4.8	3536	34	344	50	13	13	<9,148><11,108><13,95><14,93><15,65><16,60><18,56><19,54><20,51><21,51><22,44><23,36><24,0>
C1_50_8	0.73	11.4	3560	38	360	50	13	10	<11,285><12,198><13,173><14,169><15,116><17,77><18,71><20,48><22,40><23,0>
C1_50_9	0.83	8.95	3505	38	330	50	13	10	<7,182><10,127><11,104><12,94><13,76><14,62><17,51><18,40><19,36><21,0>
C1_70_0	0.68	12.97	3651	48	460	70	18	7	<18,183><19,170><21,153><22,125><28,115><29,98><30,0>
C1_70_1	0.68	10.7	3832	53	542	70	18	8	<21,178><22,139><23,138><24,125><27,112><28,108><29,84><32,0>
C1_70_2	0.73	8.94	3646	48	456	70	18	9	<16,159><17,124><24,120><26,113><27,109><28,98><29,94><30,82><31,0>
C1_70_3	0.76	8.19	3646	47	456	70	18	9	<11,150><20,134><21,106><24,100><25,93><26,90><28,86><30,82><31,0>
C1_70_4	0.65	9.6	3713	51	490	70	18	10	<20,194><22,157><23,138><24,119><25,114><27,113><28,108><29,96><30,90><33,0>
C1_70_5	0.77	8.92	3614	41	440	70	18	9	<13,207><17,205><18,152><19,123><22,119><23,107><26,104><27,89><29,0>
C1_70_6	0.66	8.24	3737	54	502	70	18	9	<22,178><24,146><25,146><26,141><27,107><28,103><29,103><30,96><31,0>
C1_70_7	0.7	12.53	3651	41	460	70	18	8	<17,198><18,154><19,126><22,108><23,107><24,98><25,65><31,0>
C1_70_8	0.67	13.49	3711	49	494	70	18	7	<20,189><21,173><22,143><25,121><26,108><30,91><31,0>
C1_70_9	0.67	8.55	3727	50	498	70	18	9	<20,182><21,171><22,142><24,141><25,126><27,105><28,88><29,81><32,0>
C1_90_0	0.69	11.21	4033	55	600	90	23	8	<25,200><28,183><29,152><31,142><33,118><35,110><36,104><38,0>
C1_90_1	0.67	16.68	4237	58	646	90	23	6	<27,194><30,118><34,102><36,100><40,97><41,0>
C1_90_2	0.66	12.07	4096	59	612	90	23	9	<27,258><28,234><29,177><30,144><32,124><35,119><36,102><37,100><40,0>
C1_90_3	0.72	7	4007	57	590	90	23	11	<22,171><23,162><25,132><27,118><29,106><33,100><34,100><35,95><36,90><39,88><41,0>

Table B.3 continued from previous page

Instance	HyperVol	K-dist(2)	CPU_time	#_spots	#_nodes	#_groups	#_tours	#_points	points
C1_90_4	0.66	9.11	4151	58	626	90	23	9	<24,165><28,146><29,121><34,113><35,106><36,105><37,105><39,96><41,0>
C1_90_5	0.71	9.42	3941	51	570	90	23	8	<23,161><27,160><28,146><29,125><32,123><33,111><34,101><38,0>
C1_90_6	0.67	8.26	4166	64	630	90	23	10	<27,194><29,161><30,155><31,126><33,120><34,119><36,115><37,102><38,98><41,0>
C1_90_7	0.69	12.4	4053	49	604	90	23	7	<24,198><26,158><27,152><28,113><30,106><35,88><39,0>
C1_90_8	0.67	6.86	4172	59	632	90	23	10	<27,167><28,156><29,125><33,118><34,117><35,105><36,103><37,96><39,94><40,0>
C1_90_9	0.68	8.05	4204	61	638	90	23	9	<26,149><29,134><31,119><33,118><34,106><35,103><37,98><39,91><40,0>
C2_5_0	0.95	28	5	11	34	5	3	2	<1,56><2,0>
C2_5_1	0.96	19.51	2	11	34	5	3	2	<1,39><2,0>
C2_5_2	0.94	27	6	11	42	5	3	2	<1,54><2,0>
C2_5_3	0.98	7.52	10	12	36	5	3	2	<1,15><2,0>
C2_5_4	0.87	41.5	14	12	36	5	3	2	<1,83><2,0>
C2_5_5	0.95	16.51	7	12	38	5	3	2	<1,33><2,0>
C2_5_6	0.97	15.51	5	12	36	5	3	2	<1,31><2,0>
C2_5_7	0.95	26.5	3	12	36	5	3	2	<1,53><2,0>
C2_5_8	0.98	11.01	3	11	30	5	3	2	<1,22><2,0>
C2_5_9	0.97	16.51	3	13	34	5	3	2	<1,33><2,0>
C2_10_0	0.88	2.67	2455	21	66	10	5	3	<2,14><3,0><4,0>
C2_10_1	0.97	17.88	2023	18	64	10	5	3	<1,103><2,4><3,0>
C2_10_2	0.88	4.31	2799	19	68	10	5	3	<2,24><3,1><4,0>
C2_10_3	0.88	10.51	1978	19	68	10	5	2	<2,21><3,0>
C2_10_4	0.83	12.68	2778	21	76	10	5	3	<2,60><3,16><4,0>
C2_10_5	0.86	22.01	2222	19	72	10	5	2	<2,44><3,0>
C2_10_6	0.89	8.51	2003	20	72	10	5	2	<2,17><3,0>
C2_10_7	0.88	6.02	1890	17	70	10	5	2	<2,12><3,0>
C2_10_8	0.88	12.01	1943	18	58	10	5	2	<2,24><3,0>
C2_10_9	0.89	3.04	1929	20	70	10	5	2	<2,6><3,0>
C2_30_0	0.78	11.02	3338	43	212	30	15	8	<5,206><6,111><7,73><9,61><10,33><11,32><13,8><14,0>
C2_30_1	0.8	9.76	3343	47	228	30	15	7	<5,203><6,160><7,79><10,73><11,31><12,25><14,0>
C2_30_2	0.86	8.92	3308	38	186	30	15	7	<4,129><5,95><6,66><8,44><9,27><11,2><13,0>
C2_30_3	0.83	9.37	3600	42	202	30	15	9	<4,189><5,134><6,93><7,69><8,63><9,52><10,31><13,2><14,0>
C2_30_4	0.81	5.65	3336	45	206	30	15	9	<5,112><6,83><7,60><8,51><10,30><11,25><12,22><13,10><14,0>
C2_30_5	0.84	13.75	3332	36	190	30	15	8	<4,222><5,95><6,79><7,61><8,44><10,21><11,5><13,0>
C2_30_6	0.81	8.79	3339	47	218	30	15	9	<5,200><6,132><7,75><9,71><10,58><11,58><12,18><13,4><14,0>
C2_30_7	0.84	6.13	3334	37	196	30	15	11	<4,152><5,102><6,63><7,48><8,48><9,36><10,28><11,19><12,16><13,3><14,0>
C2_30_8	0.81	10.04	3339	42	216	30	15	8	<5,173><6,86><7,65><8,51><9,41><12,16><13,14><14,0>
C2_30_9	0.83	6.41	3311	43	198	30	15	8	<5,127><6,91><7,48><9,39><10,38><11,24><12,2><13,0>
C2_50_0	0.81	7.58	3514	60	330	50	25	10	<8,184><10,121><11,103><15,98><16,64><17,50><18,49><19,39><20,38><22,0>
C2_50_1	0.79	6.72	3519	61	342	50	25	9	<9,132><12,111><13,80><15,63><16,58><17,49><18,48><21,35><22,0>
C2_50_2	0.78	5.22	3514	62	336	50	25	9	<9,159><10,159><11,137><14,84><16,83><17,62><18,59><19,33><22,0>
C2_50_3	0.79	8.15	3522	60	336	50	25	9	<7,162><10,132><12,111><14,84><16,83><17,69><18,48><22,43><23,0>
C2_50_4	0.75	6.72	3528	62	340	50	25	10	<10,146><11,87><13,84><16,74><17,68><18,63><19,48><21,46><23,33><24,0>
C2_50_5	0.8	10.54	3502	55	324	50	25	8	<9,162><10,108><13,66><15,56><18,47><20,42><21,34><22,0>
C2_50_6	0.78	12.84	3550	66	356	50	25	10	<8,371><9,297><11,144><12,133><14,87><16,75><18,57><19,51><22,50><23,0>
C2_50_7	0.79	10.14	3514	54	334	50	25	9	<8,186><9,122><12,84><15,78><16,68><18,55><19,54><21,35><22,0>
C2_50_8	0.74	9.24	3529	62	356	50	25	8	<11,159><13,134><14,120><15,104><16,67><17,55><21,45><23,0>
C2_50_9	0.78	9.1	3532	64	344	50	25	9	<9,187><11,140><12,97><14,92><15,58><18,52><19,48><20,38><22,0>

Table B.3 continued from previous page

Instance	HyperVol	K-dist(2)	CPU_time	#_spots	#_nodes	#_groups	#_tours	#_points	points
C2_70_0	0.71	13.22	3664	76	466	70	35	6	<18,155><23,122><24,102><27,96><28,94><30,0>
C2_70_1	0.71	8.89	3727	80	494	70	35	9	<17,173><20,151><22,119><23,115><24,108><27,98><30,90><31,89><32,0>
C2_70_2	0.7	11	3660	77	464	70	35	7	<18,164><20,140><24,101><25,96><26,93><30,88><31,0>
C2_70_3	0.71	12.83	3684	74	476	70	35	5	<17,122><25,113><27,96><28,88><32,0>
C2_70_4	0.74	10.66	3644	76	458	70	35	10	<16,221><17,183><18,165><19,152><22,124><23,116><25,110><27,102><28,92><29,0>
C2_70_5	0.71	14.07	3697	74	480	70	35	9	<17,259><18,140><24,127><25,113><27,111><28,98><30,91><31,91><32,0>
C2_70_6	0.68	12.24	3689	80	480	70	35	5	<21,116><26,105><28,100><30,96><31,0>
C2_70_7	0.66	12.14	3734	74	500	70	35	7	<21,165><22,146><24,130><25,110><27,102><28,90><32,0>
C2_70_8	0.71	12.22	3750	80	502	70	35	6	<17,141><25,112><27,100><28,96><29,88><32,0>
C2_70_9	0.68	12.32	3702	78	484	70	35	8	<20,197><21,154><22,128><23,116><25,111><26,100><29,98><31,0>
C2_90_0	0.69	9.63	4120	83	618	90	45	9	<25,205><27,186><29,131><31,121><32,108><37,106><38,103><39,94><40,0>
C2_90_1	0.64	13.45	4319	96	662	90	45	7	<30,187><31,155><32,128><33,124><34,111><37,102><41,0>
C2_90_2	0.71	14.16	3959	91	576	90	45	8	<24,224><25,147><26,144><27,143><29,136><30,122><32,99><38,0>
C2_90_3	0.7	9.48	4053	91	600	90	45	7	<24,163><26,160><29,119><32,108><35,108><36,98><40,0>
C2_90_4	0.68	8.88	4153	90	624	90	45	8	<26,148><29,136><30,134><31,125><32,123><33,107><36,94><41,0>
C2_90_5	0.68	10.1	4068	83	604	90	45	9	<26,192><27,158><28,148><29,130><31,120><33,98><37,91><38,83><40,0>
C2_90_6	0.66	12.55	4178	89	630	90	45	9	<28,230><29,178><31,162><32,139><33,123><36,110><38,106><39,105><41,0>
C2_90_7	0.66	15.53	4134	86	620	90	45	7	<27,213><28,148><30,146><32,142><33,120><34,98><41,0>
C2_90_8	0.63	7.72	4266	96	652	90	45	10	<30,167><31,150><32,146><34,137><36,117><37,112><38,110><39,105><41,96><42,0>
C2_90_9	0.69	10.84	4149	88	626	90	45	9	<25,244><26,240><27,182><29,132><31,110><35,110><36,100><38,94><40,0>
C3_5_0	0.95	26	7	13	34	5	4	2	<1,52><2,0>
C3_5_1	0.95	28	4	12	30	5	4	2	<1,56><2,0>
C3_5_2	0.91	37.5	7	13	38	5	4	2	<1,75><2,0>
C3_5_3	0.92	36.5	27	15	36	5	4	2	<1,73><2,0>
C3_5_4	0.75	1.12	580	15	38	5	4	2	<2,2><3,0>
C3_5_5	0.95	23.51	11	14	36	5	4	2	<1,47><2,0>
C3_5_6	0.98	9.01	0	13	28	5	4	2	<1,18><2,0>
C3_5_7	0.93	39.5	10	13	34	5	4	2	<1,79><2,0>
C3_5_8	0.95	20.51	3	14	32	5	4	2	<1,41><2,0>
C3_5_9	0.87	40	17	15	38	5	4	2	<1,80><2,0>
C3_10_0	0.85	13.21	2807	29	68	10	8	3	<2,75><3,4><4,0>
C3_10_1	0.86	12.03	2767	24	70	10	8	3	<2,65><3,7><4,0>
C3_10_2	0.88	10.51	2380	23	66	10	8	2	<2,21><3,0>
C3_10_3	0.86	9.97	2789	25	68	10	8	3	<2,58><3,1><4,0>
C3_10_4	0.85	12.02	2783	28	72	10	8	3	<2,63><3,9><4,0>
C3_10_5	0.85	12.38	2671	23	66	10	8	3	<2,70><3,4><4,0>
C3_10_6	0.87	25	2788	29	70	10	8	2	<2,50><3,0>
C3_10_7	0.86	9.75	2469	23	68	10	8	3	<2,56><3,2><4,0>
C3_10_8	0.87	9.39	2787	24	68	10	8	3	<2,53><3,3><4,0>
C3_10_9	0.87	26	1923	26	68	10	8	2	<2,52><3,0>
C3_30_0	0.81	8.53	3312	55	202	30	23	8	<5,133><7,84><8,64><9,53><10,48><11,32><12,15><13,0>
C3_30_1	0.8	9.69	3338	58	216	30	23	7	<5,134><6,103><7,70><8,67><10,48><12,25><14,0>
C3_30_2	0.82	8.96	3312	59	204	30	23	7	<5,135><6,90><9,47><10,46><11,32><12,30><13,0>
C3_30_3	0.82	5.68	3335	57	204	30	23	8	<5,134><6,122><8,81><9,47><10,45><11,24><13,21><14,0>
C3_30_4	0.78	7.8	3334	58	200	30	23	8	<5,136><6,93><8,82><9,62><10,48><11,28><12,25><14,0>
C3_30_5	0.82	22.55	3309	51	190	30	23	8	<5,367><6,94><7,64><8,51><9,46><10,25><12,16><13,0>
C3_30_6	0.82	8.81	3338	64	212	30	23	9	<5,150><6,98><7,78><8,61><10,51><11,43><12,32><13,18><14,0>
C3_30_7	0.82	6.38	3335	49	202	30	23	9	<5,141><6,77><7,73><8,51><9,37><10,37><12,24><13,22><14,0>
C3_30_8	0.79	15.65	3339	59	216	30	23	8	<5,254><6,141><7,76><8,64><10,39><12,29><13,20><14,0>
C3_30_9	0.77	10.75	3340	61	216	30	23	7	<6,154><7,73><8,65><10,48><11,43><12,29><14,0>

Table B.3 continued from previous page

Instance	HyperVol	K-dist(2)	CPU_time	#_spots	#_nodes	#_groups	#_tours	#_points	points
C3_50_0	0.78	8.84	3523	79	346	50	38	9	<9,189><10,148><13,145><14,104><15,80><17,60><18,51><21,48><22,0>
C3_50_1	0.78	15.42	3537	86	360	50	38	7	<9,226><10,161><14,107><16,88><17,50><21,44><23,0>
C3_50_2	0.81	9.95	3503	81	324	50	38	9	<8,184><12,95><13,78><14,64><16,54><19,36><20,34><21,28><22,0>
C3_50_3	0.77	9.58	3537	82	336	50	38	8	<9,153><13,122><14,88><17,85><18,68><19,50><21,35><23,0>
C3_50_4	0.81	8.48	3510	81	328	50	38	8	<8,166><9,145><12,103><13,90><15,77><18,50><19,46><22,0>
C3_50_5	0.79	11.05	3537	77	344	50	38	7	<9,168><12,113><14,78><15,76><17,53><21,50><23,0>
C3_50_6	0.78	8.11	3537	82	362	50	38	8	<9,176><10,171><11,119><14,97><15,78><18,52><22,48><23,0>
C3_50_7	0.75	13.53	3533	78	344	50	38	8	<9,220><11,163><13,121><14,80><18,73><19,55><22,53><23,0>
C3_50_8	0.76	13.06	3545	85	360	50	38	7	<9,182><11,130><15,101><16,69><21,53><22,51><23,0>
C3_50_9	0.76	5.81	3540	82	350	50	38	9	<10,136><11,136><13,100><17,81><18,64><20,52><21,51><22,42><23,0>
C3_70_0	0.69	14.06	3701	89	482	70	53	6	<19,191><20,180><23,150><25,98><30,81><32,0>
C3_70_1	0.73	8.32	3704	102	486	70	53	9	<15,157><20,151><21,130><24,118><25,111><26,108><28,98><30,87><32,0>
C3_70_2	0.7	19.69	3661	99	464	70	53	7	<18,271><19,123><21,111><25,100><27,98><29,87><30,0>
C3_70_3	0.69	10.71	3674	99	470	70	53	7	<19,150><21,138><23,120><25,104><26,96><29,95><32,0>
C3_70_4	0.73	9.7	3663	96	466	70	53	8	<17,163><21,159><22,141><24,123><25,121><26,119><27,102><30,0>
C3_70_5	0.68	10.02	3691	93	476	70	53	9	<20,192><21,176><23,165><24,160><25,143><26,142><27,112><28,84><32,0>
C3_70_6	0.71	12.95	3693	98	480	70	53	6	<17,152><22,138><24,123><26,101><28,92><32,0>
C3_70_7	0.69	12.15	3701	97	484	70	53	9	<19,220><21,154><22,122><24,112><25,105><27,101><29,92><30,90><31,0>
C3_70_8	0.66	15.53	3722	103	494	70	53	7	<21,208><22,144><24,118><25,102><30,101><31,89><32,0>
C3_70_9	0.71	10.67	3707	101	486	70	53	9	<18,204><19,166><20,137><25,124><27,118><28,96><29,94><30,82><31,0>
C3_90_0	0.68	11.9	4114	105	616	90	68	7	<26,165><29,142><32,116><36,106><37,103><38,96><40,0>
C3_90_1	0.68	10.94	4167	117	628	90	68	7	<26,145><30,120><31,106><35,101><38,101><40,96><41,0>
C3_90_2	0.69	9.93	4043	112	598	90	68	7	<26,136><29,125><32,121><33,112><35,108><38,101><39,0>
C3_90_3	0.7	9.8	4087	109	610	90	68	9	<25,187><26,154><29,146><30,135><31,133><32,113><34,104><39,101> <40,0>
C3_90_4	0.71	9.02	4118	109	618	90	68	10	<23,194><26,168><27,136><29,135><32,123><33,105><35,105><37,105> <38,101><40,0>
C3_90_5	0.67	20.7	4089	112	608	90	68	6	<27,240><28,140><32,121><36,103><38,96><40,0>
C3_90_6	0.66	12.18	4179	112	630	90	68	6	<28,142><30,130><33,121><34,118><35,98><41,0>
C3_90_7	0.68	13.15	4117	115	616	90	68	9	<26,254><27,211><28,150><29,138><31,121><32,116><34,106><35,84><41,0>
C3_90_8	0.65	11.88	4165	113	626	90	68	7	<27,165><31,137><35,109><37,108><38,100><39,95><41,0>
C3_90_9	0.65	11.16	4131	117	626	90	68	6	<29,129><31,118><34,106><36,100><39,98><40,0>
C4_5_0	0.92	42	15	16	34	5	5	2	<1,84><2,0>
C4_5_1	0.95	28.5	7	15	32	5	5	2	<1,57><2,0>
C4_5_2	0.88	52	11	15	34	5	5	2	<1,104><2,0>
C4_5_3	0.92	41.5	24	16	34	5	5	2	<1,83><2,0>
C4_5_4	0.75	1.12	67	18	38	5	5	2	<2,2><3,0>
C4_5_5	0.93	31.5	27	16	34	5	5	2	<1,63><2,0>
C4_5_6	0.96	25	4	17	34	5	5	2	<1,50><2,0>
C4_5_7	nan	0	24	15	34	5	5	1	<2,0>
C4_5_8	0.94	24.51	4	15	30	5	5	2	<1,49><2,0>
C4_5_9	0.94	34	19	17	36	5	5	2	<1,68><2,0>
C4_10_0	0.85	13.72	2781	33	70	10	10	3	<2,79><3,3><4,0>
C4_10_1	0.86	11.35	2773	29	70	10	10	3	<2,58><3,10><4,0>
C4_10_2	0.87	9.5	2751	27	64	10	10	3	<2,55><3,0><4,0>
C4_10_3	0.87	26	2721	29	68	10	10	2	<2,52><3,0>
C4_10_4	0.85	16.01	2789	32	72	10	10	3	<2,82><3,14><4,0>
C4_10_5	0.87	5.58	3174	26	62	10	10	3	<2,31><3,2><4,0>
C4_10_6	0.86	13.75	2474	33	72	10	10	3	<2,80><3,2><4,0>
C4_10_7	0.86	11.14	2483	26	66	10	10	3	<2,65><3,1><4,0>
C4_10_8	0.85	17.85	2749	28	68	10	10	3	<2,95><3,12><4,0>
C4_10_9	0.86	14.38	2751	30	68	10	10	3	<2,82><3,4><4,0>

Table B.3 continued from previous page

Instance	HyperVol	K-dist(2)	CPU_time	#_spots	#_nodes	#_groups	#_tours	#_points	points
C4_30_0	0.81	9.59	3312	68	202	30	30	9	<5,186><6,111><7,84><8,77><9,68><10,44><11,34><12,31><13,0>
C4_30_1	0.78	14.36	3337	71	212	30	30	8	<5,220><6,113><7,71><8,57><10,48><11,38><13,27><14,0>
C4_30_2	0.81	7.5	3335	72	202	30	30	8	<5,120><7,94><8,75><9,56><10,48><11,36><13,34><14,0>
C4_30_3	0.8	4.63	3334	66	200	30	30	9	<5,144><6,143><7,80><8,67><10,52><11,35><12,31><13,21><14,0>
C4_30_4	0.76	11.25	3335	70	202	30	30	9	<5,208><6,111><7,91><8,77><9,67><10,46><11,41><12,34><14,0>
C4_30_5	0.82	16.25	3311	65	200	30	30	7	<5,244><6,108><7,100><8,63><10,32><12,20><13,0>
C4_30_6	0.82	7.29	3337	73	210	30	30	9	<5,139><6,93><7,82><8,61><9,51><11,47><12,35><13,26><14,0>
C4_30_7	0.8	7.87	3336	63	206	30	30	9	<5,152><6,120><7,90><8,67><10,55><11,49><12,27><13,22><14,0>
C4_30_8	0.78	9.87	3606	73	218	30	30	6	<6,112><7,81><10,54><11,48><12,26><14,0>
C4_30_9	0.8	9.79	3337	70	208	30	30	8	<5,162><6,97><7,78><9,65><10,52><11,46><12,17><14,0>
C4_50_0	0.79	6.57	3540	88	340	50	50	10	<9,141><10,119><12,93><14,90><15,78><16,77><18,65><19,52><21,39><23,0>
C4_50_1	0.77	7.51	3536	100	354	50	50	10	<9,169><12,129><13,113><14,112><15,101><16,74><18,58><21,54><22,45><23,0>
C4_50_2	0.78	9.91	3524	96	332	50	50	10	<9,212><11,151><13,80><15,69><17,65><18,54><19,44><20,37><21,36><22,0>
C4_50_3	0.8	11.31	3517	97	332	50	50	9	<8,244><10,163><12,138><14,85><16,73><17,70><18,48><20,43><22,0>
C4_50_4	0.8	5.56	3520	95	338	50	50	12	<8,169><11,142><13,128><14,108><15,69><16,68><17,56><19,56><20,49><21,43><22,43><23,0>
C4_50_5	0.77	10.05	3530	90	340	50	50	8	<9,233><11,185><15,73><16,71><17,60><20,43><21,37><23,0>
C4_50_6	0.75	9.83	3543	93	350	50	50	9	<11,190><12,148><14,101><15,90><16,68><18,64><20,60><21,59><23,0>
C4_50_7	0.76	6.28	3522	90	338	50	50	8	<11,137><12,132><13,108><14,86><16,81><18,49><20,41><22,0>
C4_50_8	0.71	7.34	3553	100	354	50	50	8	<12,143><14,129><15,122><16,88><19,78><20,60><21,52><24,0>
C4_50_9	0.74	6.84	3522	95	350	50	50	7	<12,136><13,118><14,118><15,69><18,64><19,48><22,0>
C4_70_0	0.67	20.85	3690	107	480	70	70	5	<21,204><22,126><27,100><29,100><31,0>
C4_70_1	0.67	19.78	3703	118	484	70	70	9	<20,359><21,192><22,123><24,114><26,98><27,98><30,96><31,88><32,0>
C4_70_2	0.7	11.75	3670	115	468	70	70	9	<18,229><20,152><21,145><22,132><24,104><27,101><28,100><30,87><31,0>
C4_70_3	0.68	15.42	3684	112	474	70	70	7	<19,219><20,185><22,141><24,118><25,111><27,105><31,0>
C4_70_4	0.72	7.24	3676	110	472	70	70	10	<17,188><18,180><19,166><21,127><23,121><24,119><25,100><27,94><29,93><31,0>
C4_70_5	0.67	8.24	3691	112	478	70	70	7	<21,142><23,133><26,106><28,105><29,84><31,76><32,0>
C4_70_6	0.69	12.53	3707	113	484	70	70	9	<19,250><20,203><21,160><22,150><24,138><25,98><28,94><29,90><32,0>
C4_70_7	0.73	9.31	3697	116	482	70	70	7	<15,164><20,158><23,128><24,123><25,101><26,98><31,0>
C4_70_8	0.65	15.33	3706	115	486	70	70	6	<22,175><23,130><24,116><28,100><31,90><32,0>
C4_70_9	0.67	11.41	3701	117	486	70	70	7	<21,158><23,125><24,123><27,116><28,110><29,94><31,0>
C4_90_0	0.66	16.95	4129	125	620	90	90	7	<29,230><30,122><34,119><35,112><36,105><37,96><40,0>
C4_90_1	0.67	9.44	4111	129	616	90	90	7	<27,146><28,144><29,136><32,131><33,105><36,95><41,0>
C4_90_2	0.68	9.76	4100	128	614	90	90	8	<25,176><29,169><30,134><31,127><34,118><35,106><37,96><39,0>
C4_90_3	0.68	8.93	4127	129	618	90	90	9	<25,176><29,165><30,142><32,130><33,109><36,108><37,103><39,101><40,0>
C4_90_4	0.69	8.48	4108	122	614	90	90	8	<25,186><26,184><28,178><31,120><32,108><34,104><36,101><40,0>
C4_90_5	0.67	9.2	4110	127	616	90	90	8	<28,167><29,150><30,110><35,105><36,105><38,102><39,102><40,0>
C4_90_6	0.69	9.87	4090	126	612	90	90	8	<24,162><30,131><31,125><35,118><36,111><37,98><38,98><40,0>
C4_90_7	0.66	7.73	4164	128	626	90	90	9	<28,148><31,135><32,133><33,119><36,106><37,101><38,100><39,88><41,0>
C4_90_8	0.66	9.54	4121	128	618	90	90	8	<28,184><29,176><30,134><33,126><34,114><37,113><39,95><40,0>
C4_90_9	0.68	7.72	4109	125	616	90	90	9	<26,168><29,148><30,144><31,144><32,112><33,108><37,101><38,98><39,0>

Bibliography

[1] Claudia Archetti and Maria Grazia Speranza. Vehicle routing problems with split deliveries. *International transactions in operational research*, 19(1-2):3–22, 2012.

[2] Charles Audet, Jean Bigeon, Dominique Cartier, Sébastien Le Digabel, and Ludovic Salomon. Performance indicators in multi-objective optimization. *European journal of operational research*, 292(2):397–422, 2021.

[3] John W Baugh Jr, Gopala Krishna Reddy Kakivaya, and John R Stone. Intractability of the dial-a-ride problem and a multi-objective solution using simulated annealing. *Engineering Optimization*, 30(2):91–123, 1998.

[4] Alexandre Beaudry, Gilbert Laporte, Teresa Melo, and Stefan Nickel. Dynamic transportation of patients in hospitals. *OR spectrum*, 32:77–107, 2010.

[5] Gerardo Berbeglia, Jean-François Cordeau, Irina Gribkovskaia, and Gilbert Laporte. Static pickup and delivery problems: a classification scheme and survey. *Top*, 15:1–31, 2007.

[6] Kris Braekers, An Caris, and Gerrit K Janssens. Exact and meta-heuristic approach for a general heterogeneous dial-a-ride problem with multiple depots. *Transportation Research Part B: Methodological*, 67:166–186, 2014.

[7] Yongtao Cao, Byran J Smucker, and Timothy J Robinson. On using the hypervolume indicator to compare pareto fronts: Applications to multi-criteria optimal experimental design. *Journal of Statistical Planning and Inference*, 160:60–74, 2015.

[8] Pasquale Carotenuto, Artem Serebriany, and Giovanni Storchi. Flexible services for people transportation: a simulation model in a discrete events environment. *Procedia-Social and Behavioral Sciences*, 20:846–855, 2011.

[9] Pasquale Carotenuto, Daniele Monacelli, Giuseppe Raponi, and Marco Turco. A dynamic simulation model of a flexible transport services for people in congested area. *Procedia-Social and Behavioral Sciences*, 54:357–364, 2012.

[10] João CN Clímaco, José MF Craveirinha, and Marta MB Pascoal. An automated reference point-like approach for multicriteria shortest path problems. *Journal of Systems Science and Systems Engineering*, 15:314–329, 2006.

[11] Filipe Conceição. Bus on demand. 2010. URL https://api.semanticscholar.org/CorpusID:216078003.

[12] Jean-François Cordeau. A branch-and-cut algorithm for the dial-a-ride problem. *Operations Research*, 54(3):573–586, 2006.

[13] Jean-François Cordeau and Gilbert Laporte. The dial-a-ride problem (darp): Variants, modeling issues and algorithms. *Quarterly Journal of the Belgian, French and Italian Operations Research Societies*, 1:89–101, 2003.

[14] Jean-François Cordeau and Gilbert Laporte. A tabu search heuristic for the static multi-vehicle dial-a-ride problem. *Transportation Research Part B: Methodological*, 37(6):579–594, 2003.

[15] Jean-François Cordeau and Gilbert Laporte. The dial-a-ride problem: models and algorithms. *Annals of operations research*, 153: 29–46, 2007.

[16] Canan G Corlu, Rocio de la Torre, Adrian Serrano-Hernandez, Angel A Juan, and Javier Faulin. Optimizing energy consumption in transportation: Literature review, insights, and research opportunities. *Energies*, 13(5):1115, 2020.

[17] George B Dantzig and John H Ramser. The truck dispatching problem. *Management science*, 6(1):80–91, 1959.

[18] Terry Davies and Sarah Cahill. Environmental implications of the tourism industry. 2000.

[19] Kalyanmoy Deb, Amrit Pratap, Sameer Agarwal, and TAMT Meyarivan. A fast and elitist multiobjective genetic algorithm: Nsga-ii. *IEEE transactions on evolutionary computation*, 6(2):182–197, 2002.

[20] Martin Desrochers, Jacques Desrosiers, and Marius Solomon. A new optimization algorithm for the vehicle routing problem with time windows. *Operations research*, 40(2):342–354, 1992.

[21] Marco Diana and Maged M Dessouky. A new regret insertion heuristic for solving large-scale dial-a-ride problems with time windows. *Transportation Research Part B: Methodological*, 38(6):539–557, 2004.

[22] Moshe Dror, Gilbert Laporte, and Pierre Trudeau. Vehicle routing with split deliveries. *Discrete Applied Mathematics*, 50(3): 239–254, 1994.

[23] Gunter Dueck and Tobias Scheuer. Threshold accepting: A general purpose optimization algorithm appearing superior to simulated annealing. *Journal of computational physics*, 90(1):161–175, 1990.

[24] Aish Fenton. The bees algorithm for the vehicle routing problem. *arXiv preprint arXiv:1605.05448*, 2016.

[25] Marshall L Fisher. The lagrangian relaxation method for solving integer programming problems. *Management science*, 27(1): 1–18, 1981.

[26] Xavier Gandibleux, Nazik Mezdaoui, and Arnaud Fréville. A tabu search procedure to solve multiobjective combinatorial optimization problems. In *Advances in Multiple Objective and Goal Programming: Proceedings of the Second International Conference on Multi-Objective Programming and Goal Programming, Torremolinos, Spain, May 16–18, 1996*, pages 291–300. Springer, 1997.

[27] Marc Goetschalckx and Charlotte Jacobs-Blecha. The vehicle routing problem with backhauls. *European Journal of Operational Research*, 42(1):39–51, 1989.

[28] Intaek Gong, Kyungho Lee, Jaewon Kim, Yunhong Min, and KwangSup Shin. Optimizing vehicle routing for simultaneous delivery and pick-up considering reusable transporting containers: Case of convenience stores. *Applied Sciences*, 10(12):4162, 2020.

[29] Yacov Haimes. On a bicriterion formulation of the problems of integrated system identification and system optimization. *IEEE transactions on systems, man, and cybernetics*, (3):296–297, 1971.

[30] Sin C Ho, Wai Yuen Szeto, Yong-Hong Kuo, Janny MY Leung, Matthew Petering, and Terence WH Tou. A survey of dial-a-ride problems: Literature review and recent developments. *Transportation Research Part B: Methodological*, 111:395–421, 2018.

[31] Carl H Hüll, Henrik Andersson, Jan T Lundgren, and Peter Värbrand. The integrated dial-a-ride problem. *Public Transport*, 1: 39–54, 2009.

[32] IBM. Cp optimizer user's manual, 2022. https://www.ibm.com/docs/es/icos/22.1.1?topic=optimizer-cp-users-manual, Last accessed on 2023-12-29.

[33] Jang-Jei Jaw, Amedeo R Odoni, Harilaos N Psaraftis, and Nigel HM Wilson. A heuristic algorithm for the multi-vehicle advance request dial-a-ride problem with time windows. *Transportation Research Part B: Methodological*, 20(3):243–257, 1986.

[34] Joshua Knowles and David Corne. On metrics for comparing nondominated sets. In *Proceedings of the 2002 Congress on Evolutionary Computation. CEC'02 (Cat. No. 02TH8600)*, volume 1, pages 711–716. IEEE, 2002.

[35] Amit Kohar and Suresh Kumar Jakhar. A capacitated multi pickup online food delivery problem with time windows: a branch-and-cut algorithm. *Annals of Operations Research*, pages 1–22, 2021.

[36] Gilbert Laporte, Yves Nobert, and Serge Taillefer. Solving a family of multi-depot vehicle routing and location-routing problems. *Transportation science*, 22(3):161–172, 1988.

[37] Allan Larsen, OBGD Madsen, and Marius Solomon. Partially dynamic vehicle routing-models and algorithms. *Journal of the operational research society*, 53:637–646, 2002.

[38] Mengyang Liu, Zhixing Luo, and Andrew Lim. A branch-and-cut algorithm for a realistic dial-a-ride problem. *Transportation Research Part B: Methodological*, 81:267–288, 2015.

[39] Carlos B Lucasius, Adrie D Dane, and Gerrit Kateman. On k-medoid clustering of large data sets with the aid of a genetic algorithm: background, feasiblity and comparison. *Analytica Chimica Acta*, 282(3):647–669, 1993.

[40] Ying Luo and Paul Schonfeld. A rejected-reinsertion heuristic for the static dial-a-ride problem. *Transportation Research Part B: Methodological*, 41(7):736–755, 2007.

[41] Oli BG Madsen, Hans F Ravn, and Jens Moberg Rygaard. A heuristic algorithm for a dial-a-ride problem with time windows, multiple capacities, and multiple objectives. *Annals of operations Research*, 60:193–208, 1995.

[42] Baldassarre Marco, Carotenuto Pasquale, and Raponi Giuseppe. Dynamic simulation of a flexible transport system. *IFAC Proceedings Volumes*, 45(6):315–320, 2012. 14th IFAC Symposium on Information Control Problems in Manufacturing.

[43] Renaud Masson, Fabien Lehuédé, and Olivier Péton. The dial-a-ride problem with transfers. *Computers & Operations Research*, 41:12–23, 2014.

[44] George Mavrotas. Effective implementation of the ε-constraint method in multi-objective mathematical programming problems. *Applied mathematics and computation*, 213(2):455–465, 2009.

[45] Nenad Mladenović and Pierre Hansen. Variable neighborhood search. *Computers & operations research*, 24(11):1097–1100, 1997.

[46] Jairo R Montoya-Torres, Julián López Franco, Santiago Nieto Isaza, Heriberto Felizzola Jiménez, and Nilson Herazo-Padilla. A literature review on the vehicle routing problem with multiple depots. *Computers & Industrial Engineering*, 79:115–129, 2015.

[47] Santiago Muelas, Antonio LaTorre, and Jose-Maria Pena. A distributed vns algorithm for optimizing dial-a-ride problems in large-scale scenarios. *Transportation Research Part C: Emerging Technologies*, 54:110–130, 2015.

[48] Patrick Ngatchou, Anahita Zarei, and A El-Sharkawi. Pareto multi objective optimization. In *Proceedings of the 13th international conference on, intelligent systems application to power systems*, pages 84–91. IEEE, 2005.

[49] Vikas Palakonda and Rammohan Mallipeddi. Pareto dominance-based algorithms with ranking methods for many-objective optimization. *IEEE Access*, 5:11043–11053, 2017.

[50] Julie Paquette, Jean-François Cordeau, and Gilbert Laporte. Quality of service in dial-a-ride operations. *Computers & Industrial Engineering*, 56(4):1721–1734, 2009.

[51] Julie Paquette, Jean-François Cordeau, Gilbert Laporte, and Marta MB Pascoal. Combining multicriteria analysis and tabu search for dial-a-ride problems. *Transportation Research Part B: Methodological*, 52:1–16, 2013.

[52] Sophie N Parragh. Introducing heterogeneous users and vehicles into models and algorithms for the dial-a-ride problem. *Transportation Research Part C: Emerging Technologies*, 19(5):912–930, 2011.

[53] Sophie N Parragh, Karl F Doerner, Richard F Hartl, and Xavier Gandibleux. A heuristic two-phase solution approach for the multi-objective dial-a-ride problem. *Networks: An International Journal*, 54(4):227–242, 2009.

[54] Sophie N Parragh, Karl F Doerner, and Richard F Hartl. Demand responsive transportation. *Wiley Encyclopedia of Operations Research and Management Science*, 2010.

[55] Sophie N Parragh, Karl F Doerner, and Richard F Hartl. Variable neighborhood search for the dial-a-ride problem. *Computers & Operations Research*, 37(6):1129–1138, 2010.

[56] Anthony Przybylski, Xavier Gandibleux, and Matthias Ehrgott. Two phase algorithms for the bi-objective assignment problem. *European Journal of Operational Research*, 185(2):509–533, 2008.

[57] Stefan Ropke, Jean-François Cordeau, and Gilbert Laporte. Models and branch-and-cut algorithms for pickup and delivery problems with time windows. *Networks: An International Journal*, 49(4):258–272, 2007.

[58] Francesca Rossi, Peter Van Beek, and Toby Walsh. *Handbook of constraint programming*. Elsevier, 2006.

[59] Bernard W Silverman. *Density estimation for statistics and data analysis*. Routledge, 2018.

[60] Marius M Solomon. Algorithms for the vehicle routing and scheduling problems with time window constraints. *Operations research*, 35(2):254–265, 1987.

[61] Statista. Number of international tourist arrivals worldwide 1950-2022, 2023. https://www.statista.com/statistics/209334/total-number-of-international-tourist-arrivals/, Last accessed on 2023-10-02.

[62] Statista. Travel and tourism in mexico, 2023. https://www.statista.com/topics/6323/travel-and-tourism-in-mexico/#topicOverview, Last accessed on 2023-10-05.

[63] Paolo Toth and Daniele Vigo. Heuristic algorithms for the handicapped persons transportation problem. *Transportation science*, 31(1):60–71, 1997.

[64] AC Wade and Said Salhi. An investigation into a new class of vehicle routing problem with backhauls. *Omega*, 30(6):479–487, 2002.

[65] Ka Io Wong and Michael GH Bell. Solution of the dial-a-ride problem with multi-dimensional capacity constraints. *International Transactions in Operational Research*, 13(3):195–208, 2006.

[66] Zhihai Xiang, Chengbin Chu, and Haoxun Chen. A fast heuristic for solving a large-scale static dial-a-ride problem under complex constraints. *European journal of operational research*, 174(2):1117–1139, 2006.

[67] Shangyao Yan and Chun-Ying Chen. An optimization model and a solution algorithm for the many-to-many car pooling problem. *Annals of Operations Research*, 191:37–71, 2011.

[68] Sandra Zajac and Sandra Huber. Objectives and methods in multi-objective routing problems: a survey and classification scheme. *European journal of operational research*, 290(1):1–25, 2021.

[69] Eckart Zitzler and Lothar Thiele. Multiobjective evolutionary algorithms: a comparative case study and the strength pareto approach. *IEEE transactions on Evolutionary Computation*, 3(4):257–271, 1999.

[70] Eckart Zitzler, Marco Laumanns, and Lothar Thiele. Spea2: Improving the strength pareto evolutionary algorithm. *TIK report*, 103, 2001.

Autobiography

Oscar Alejandro Hernández López

As a candidate to obtain the degree of Doctor in Systems Engineering

Universidad Autónoma de Nuevo León
Facultad de Ingeniería Mecánica y Eléctrica

Tesis:

The Dial-a-Tour Problem: A study of Single and Bi-Objective Optimization Methods

I was born on July 19th, 1993 in Havana City, Cuba. The only child of Oscar Hernández Pérez and Caridad Evarista López Guerra. In 2017, I obtained a degree in Industrial engineering at Universidad Tecnológica de La Habana "José Antonio Echeverría", Cuba. In 2020, I obtained my master's in Systems Engineering in the Graduate Program in Systems Engineering at Universidad Autónoma de Nuevo León, México. I continued my PhD. studies at the same graduate program and also working under the supervision of Vincent André Lionel Boyer, PhD.