The effect of locating remote offices on the utility of employees and traffic congestion in Mexico City

Master Thesis Econometrics and Management Science

Author: Marten Soer (544030)

Supervised by: Olga Kuryatnikova

Second assessor: Twan Dollevoet

Erasmus
University
Rotterdam

Erasmus School of Economics

July 28th, 2024

The content of this thesis is the sole responsibility of the author and does not reflect the view of the supervisor, second assessor, Erasmus School of Economics or Erasmus University.

Abstract

In this paper, we propose several algorithms to locate remote offices in Mexico City to improve the utility of employees working in the city and traffic congestion in one of the busiest cities in the world. Firstly, we exactly solve large scale multi objective uncapacitated facility location problems (UFLP) in a matter of seconds by means of a Benders Decomposition (BD) in a branch-and-bound framework. The BD incorporates the computational efficiency of a Knapsack problem to find the optimal solution. As a result, the total commuting time decreases with up to 83.8% in comparison to the situation where employees commute to their original office. The effect on utility of opening an additional remote office is shown to be diminishingly increasing in the number of open remote offices. Furthermore, we extend the UFLP by incorporating remote office capacities. A similar approach is implemented, but the subproblem of this approach is too time-consuming for instances with high number of possible remote office locations. Consequently, we employ a quantum based heuristic that finds high quality solutions in reasonable time. Limiting the number of employees assigned to a remote office results in a more balanced solution such that there are less overcrowded and undercrowded remote offices. Lastly, we exploit information on the traffic in Mexico City to create a road network. A bi-level model is solved by means of a heuristic approach. The information on congestion significantly impacts the solution, which results in an increase of 7.60% travel distance and a decrease of 4.47% in travelling speed. In conclusion, the information on congestion significantly impacts the solution and we show that employees have to travel longer to their remote office, resulting in lower utilities.

Contents

1 Introduction

Working in remote offices is an attractive alternative for office employees (Lara-Pulido & Martinez-Cruz, 2023). A remote office is defined as a decentralized office building that could facilitate the working environment of multiple companies in one building. Following a research from S. Zhang et al. (2020), remote working may reduce the commuting time and congestion by efficiently locating these buildings. Consequently, remote offices are not only beneficial for the employees but also for all other inhabitants in the city. In this paper, we quantify the effect of remote offices in Mexico City. The average commuting time in this city is 1.8 hours (Hess & Narteh-Yoe, 2020) while this is on average 25 minutes in Europe (Eurostat, 2020). Lara-Pulido & Martinez-Cruz (2023) collect data from over 1,000 employees living in and around Mexico City and estimate their utility from using a remote office. Based on this sample, traffic and geographical information of Mexico City we optimize possible locations of remote offices.

Initially, we optimize the location of uncapacitated remote offices based on the reported utility of the employees. In their ideal situation a remote office is located next to their residence, resulting in sky high cost. To counteract this, the problem has two objectives: maximizing utility and minimizing number of open remote offices. Consequently, we formulate the first research question: “How to efficiently allocate remote offices while maximizing the employee utility and minimizing the number of opened offices?”. This uncapacitated facility location problem (UFLP) is solved to optimality using an altered version of the Benders decomposition introduced in Fischetti et al. (2017). All instances are solved to optimality in relatively short computation times, resulting in an increase in utility of $\$200,602.97$ while the average commuting time decreases with 83.8% compared to the situation without any remote offices. Furthermore, we employ the ϵ-constraint method to find optimal Pareto fronts that shows the effect of restricting the number of remote offices on the utility. These fronts show that the marginal increase of the system utility is diminishing in the number of open remote offices. Hence, an additional office has more effect on the utility of employees when fewer remote offices are open.

Originally, we assume that the offices have infinite capacity (number of desks) which could result in overburdened facilities. Consequently, the model is altered such that possible capacities of remote offices are taken into account. This leads to the second research question: “How does a finite office capacity influence the allocation of remote offices?”. We introduce an exact and heuristic approach. Firstly, the exact approach extends the uncapacitated facility location problem by incorporating altered optimality cuts (Fischetti et al., 2016). Secondly, we utilize an altered version of the quantum-inspired genetic algorithm from T.-C. Lu & Yu (2013) to obtain high-quality heuristic solutions. We show that the heuristic approach finds high quality solutions in reasonable time and that the capacity limit does not impose a great restriction when large number of facilities are allowed to be opened, however the effect on utility increases when the number of open offices decreases. We show that the restriction results in a more balanced solution in the number of employees assigned to offices.

Lastly, congestion is a significant issue in Mexico City (Lara-Pulido & Martinez-Cruz, 2023). However, up to this point the travel times are assumed to be constant and determined beforehand. Realistically, the travel times are contingent upon the traffic in the highly-congested city and cannot be determined in advance. In consequence, the third research question is: “What is the effect of incorporating traffic congestion into the decision making for remote office locations?”. The employees want to minimize their commuting time (Gordon et al., 1989) while employers minimize their total cost, which is inherent to maximizing the system utility. These conflicting objectives result in a strategic game aiming to find the Nash-equilibrium. To tackle this problem, a bi-level model is introduced. We solve this model heuristically by employing the idea of Zaferanieh et al. (2023). This method solves the the lower and upper problem separately and relate the models by a linear regression. It is shown that the effect of congestion increases with lower number of open facilities. Compared to the UFLP without traffic decisions, the average travelling speed reduces with 4.47% while the travel distance increases with 7.60%. Moreover, the utility is significantly impacted: the decrease in utility ranges from 4.75% to 12.55% depending on the size of the instance. We conclude that incorporating congestion into the decision making has significant effect on the solution.

We contribute to the literature in several ways:

1. We adopt modified methods originally introduced in Fischetti et al. (2017) and Fischetti et al. (2016). These researchers implement a “lazy constraint callback” to cut off non-integer solutions while we implement a branch-and-bound (BB) scheme to find the optimal integer solution. The results obtained by (Fischetti et al., 2017) show that a “lazy” branch-and-cut framework results in exploring thousands of nodes at the cost of higher computation times, this could be due to the fact that in every node the entire LP is solved from scratch. In consequence, the simplicity of the branch-and-bound might be beneficial as we do not resolve the LP from scratch but rather tighten the formulation.

2. We adopt the idea originally introduced by T.-C. Lu & Yu (2013) for the Knapsack problem to heuristically solve the CFLP. We alter the implementation to make it applicable for the CFLP and introduce a quick k-Medoid assignment heuristic to efficiently assign employees to open remote offices.

3. We contribute by solving a single-source UFLP incorporating congestion on a real-life instance by clustering the employees into districts, solving these independently and combining them afterwards.

We proceed with reviewing research in the area of optimizing facility locations and the contribution of this paper in Section 2. We give a clear overview on the problem and assumptions in Section 3 and the used data in Section 4. Furthermore, we introduce the methods proposed to solve the problems in Section 5 and discuss the results of these methods extensively in Section 6. Lastly, we give a conclusion in Section 7 in which room for further research is outlined.

2 Literature Review

To our knowledge, the facility location problem is introduced by Stollsteimer (1961) and Balinski (1966). Since then, locating facilities has been well studied and known as the (un)capacitated facility location problem. These problems are NP-Hard ((Cornuéjols et al., 1983), (Mirchandani & Francis, 1990)), hence there are no polynomial time algorithms to solve them. We introduce several existing methods of interest in Section 2.1 that solve the uncapacitated variant exactly. The relevant literature about the capacitated facility location problem is discussed in Section 2.2. Lastly, we give an overview of the literature on facility location problems incorporating network traffic. The terms facility and remote office are used interchangeably in the entire paper.

2.1 Uncapacitated Facility Location problem

The Uncapacitated Facility Location problem (UFLP) optimizes the location of facilities by maximizing (or minimizing) the objective function and adhering to all constraints while assuming that every facility has infinite capacity. In this section, we introduce a variety of significant methods in the literature that solve this problem exactly.

Decomposition techniques (DT) are well studied to exactly solve this problem. These techniques offer several benefits for large-scale by breaking down a large problem into several smaller sub problems. Firstly, Column Generation (CG) is an often applied technique. CG starts of with a small problem, containing none or a few variables, and iteratively adds variables to achieve the optimal solution. Du et al. (2020) implement a column-and-constraint generation technique that not only iteratively add variables, but also constrains. They conclude that the CG performs the best on their large scale instance. Regularly, the integer variables in the UFLP are relaxed to find an upper bound (when maximizing) on the solution and a branch-and-cut (BC) framework is implemented to obtain the optimal integer solution. Combining the strengths of CG and BC is better known as a branch-and-price algorithm. Senne et al. (2005) present this algorithm, resulting in the search of smaller search trees and lower computation times than the traditional CG. It achieves superior results on medium sized instances but does not find optimal solutions for large instances.

Moreover, another popular DT is Benders Decomposition (BD). BD divides the variables in the problem into two sets and considers for each set a subproblem. The first subproblem finds a solution for the first set, while the second subproblem uses the answer of the first set to find a solution for the second set. The solution of the second set is integrated into the first subproblem by so-called feasibility or optimality cuts and re-optimized, this procedure is repeated until the optimal solution is found. This technique is known to find optimal solution in large instances, nevertheless several researches (e.g. (Wentges, 1996), (Rahmaniani et al., 2017)) have proven that the BD could converge slowly to optimality. Hence, many researchers adopt accelerating techniques to improve the convergence.

Magnanti & Wong (1981) define a Pareto optimal cut (not dominated by other cuts) which accelerates the BD significantly. Only 30 years later other accelerating techniques are implemented to reduce the number of iterations and memory storage. Rei et al. (2009) implement local branching search to find multiple feasible solutions resulting in better lower and upper bounds. Combining the Pareto optimal cuts and multiple feasible solutions into one algorithm finds the best-known solutions for difficult instances (Beheshti Asl & MirHassani, 2019). Although the instances are proven to be difficult, the instances are relatively small with at most 100 facilities and clients. We assess the performance of this algorithm on large-scale instances. Note that all aforementioned methods are tested on small to medium instances with at most 500 facilities. Fischetti et al. (2017) are an exception in the literature and are able to solve instances with millions of variables. They observe that the subproblem of an UFLP can be solved more efficient by means of an efficient Knapsack algorithm. Due to the high quality solving mechanisms available, the computation time decreases significantly and the accelerated BD finds the best results on most instances. Consequently, we implement a modified version and compare this idea to the algorithm introduced by Beheshti Asl & MirHassani (2019).

2.2 Capacitated Facility Location problem

Not only the uncapacitated, but also the Capacitated Facility Location problem (CFLP) is well studied (e.g. (Akinc & Khumawala, 1977), (Holmberg et al., 1999), (Wu et al., 2006)). It extends the UFLP by including maximum capacity for a facility such that a facility cannot accommodate an infinite number of clients. Most researchers focus on the development of efficient exact and heuristic methods, we mention a small portion of them.

Many researchers employ a branch-and-bound (BB) framework to optimally solve the problem. The integer restrictions are relaxed such that the LP relaxation is solved. The first authors to solve the CFLP by means of the BB are Akinc & Khumawala (1977). They find solutions for instances with up to 25 facilities and 50 clients within a few seconds and concluded that the production of better lower bounds (in a maximization problem) would significantly improve the efficiency. Görtz & Klose (2012) introduce an efficient BB algorithm to tackle the CFLP by utilizing a subgradient optimization technique to find the most influential fractional variable. The method's main idea is to average solutions obtained for the Lagrangian subproblem to estimate fractional primal solutions to the Lagrangian dual and to base branching decisions on the them, at the time of writing it outperformed all other state-of-the-art implementations.

Apart from BB algorithms, DTs are thoroughly studied as well. Lorena & Senne (2004) introduce a branch-and-price algorithm to find optimal solutions. In this framework a CG algorithm is implemented to find the optimal solution of the MP. They combine the power of the CG and Lagrangian relaxation to speed up the process and already find optimal solutions for instances with hundreds of employees within minutes. (Klose & Görtz, 2007) follow the same framework but implement a stabilized CG. In consequence, this algorithm is able to solve instances with hundreds of facilities and employees in matter of seconds. However, it is not able to solve instances with significantly more employees than facility locations. In addition to CG, BD is another often utilized method to find optimal solutions. Due to the slow convergence (Wentges, 1996), many papers have implemented a stabilization technique to improve convergence. Tang et al. (2013) decrease the iterations and computation time by adding high-density Pareto cuts in each iteration. Naoum-Sawaya & Elhedhli (2013) present an interior-point branch-and-cut (BC) BD algorithm, this type of configuration resulted to be eleven times faster than the standard BC BD. Fischetti et al. (2016) implement a similar idea to incorporate the BD into a BC, while solving a Knapsack problem to find the dual variables. This method outperforms other exact methods on most instances, hence we use a modified version of this method to find exact solutions.

Due to the complexity of the problem, optimally solving multi objective CFLP on large-size instances is not likely to converge within reasonable time. Hence, we discuss several significant heuristic methods. Alenezy (2020) introduce tightening constraints to the Lagrangian decomposition representation to improve the lower bound (minimization) while Kim & Kim (2013) combines Lagrangian relaxation and subgradient optimization to create good solutions in relatively short computation time. Furthermore, the non-dominated sorting genetic algorithm (NSGA) is another well studied multi objective algorithm. The NSGA is proposed by Mirchandani & Francis (1990) and modified by Deb et al. (2002) (NSGA-II), we explain this algorithm in Section 5.3.2. Deb & Jain (2013) introduce another altered version of the NSGA that is able to handle many objective problems more efficiently (more than three objectives), but as we are in a case of at most two objectives this is not of interest to us. The implementation of modified NSGA-II is well presented in the current literature (e.g. (Medaglia et al., 2009), (Esmikhani et al., 2022), (Huo et al., 2021)). Firstly, Medaglia et al. (2009) incorporate a Mixed-Integer Program (MIP) into the NSGA-II to locate waste management facilities resulting in better scaled solutions compared to existing literature. Secondly, H. Lu et al. (2013) adjusted the initial population, crossover and mutation operator to make it more “chaotic”. In consequence, a larger part of the feasible space is searched. Simulations show that the adjusted NSGA-II efficiently solved the scheduling problem. Thirdly, T.-C. Lu & Yu (2013) create probabilistic transition states by implementing a quantum operator in the NSGA-II. This model outperforms existing quantum inspired evolutionary algorithms (QEA) and non-dominated sorting genetic algorithms significantly in terms of quality and spread of the Pareto front. To our knowledge, the CFLP has not been solved by a QEA. In consequence, we solve the capacitated facility location problem by means of this quantum inspired method.

2.3 Facility Location problem with congestion

In literature, two types of congestion into facility locations problems are considered: in-facility and network. The former considers the queues within the facility, while the latter includes occupation of the road network. Whereas the literature on incorporating in-facility congestion into the decision making is thorough (e.g. (Berman & Krass, 2004), (Shavarani et al., 2019)), incorporating network decisions into the FLP is more lacking. This is remarkable because locating facilities does have significant impact on congestion in the neighbourhood (Meyer et al., 2016). Incorporating network decisions is regularly done via a bi-level approach. First, a general review on the literature of bi-level optimization is given. Secondly, we discuss literature on bi-level optimization combining facility locations and network decisions, denoted as c-UFL.

The interest rose in the 80s when researches realized that bi-level models have strong connection with game theory (Stackelberg et al., 1952). Nowadays, bi-level problems (BLP) are widely used in fields of facility locations, credit allocations, energy policy, etc. In game theory, two parties have their own (conflicting) goal and try to find the Nash equilibrium. In this equilibrium no party can gain an advantage by unilaterally changing their decision. The bi-level approach has two "levels": leader and follower with (potentially) opposing objectives. Finding the optimal solution is difficult due to the complexity of the problem and it is even complex to find locally optimal solutions (Vicente & Calamai, 1994). Note that a problem with integrality constraints further increases the complexity.

To exactly solve the BLP most papers employ the KKT conditions. The two-level problem can be transformed into a single-level problem by using the optimality conditions of a convex and differentiable problem. The resulting non-linear model can be linearized by additional constraints and linear approximations (Fortuny-Amat & McCarl, 1981). This model is either solved by a branch-and-bound framework (Bard & Moore, 1990) or some sort of sequential algorithm (Júdice & Faustino, 1992).

Most often, bi-level models are solved heuristically. Genetic Algorithm (GA), Simulated Annealing (SA), and Particle Swarm (PS) are regularly used (Camacho-Vallejo et al., 2023). Ziar et al. (2023) handle a intermodal rail-road p-hub median model to decide on the locations of dry ports by adopting a bi-level approach. The GA finds good solutions for large instances where the standard methods are not able to find near optimal solutions. Khanduzi & Rastegar (2022) implement a hybridization of GA and SA to solve a bi-level mixed-integer problem on the healthcare network by minimizing the cost and maximizing the coverage of patients. The hybridization turns out to outperform the distinct methods based on quality and computation time. Lastly, Kuo & Huang (2009) implement a PS algorithm to solve four problems in literature, the results indicate that the PS algorithm outperforms the GA in accuracy. Camacho-Vallejo et al. (2023) give a comprehensive review on the heuristics employed in bi-level optimization.

Decision making on networks based on the traffic network is better known as network design problems (NDPs). The leader problem minimizes the system travel time while the follower problem minimizes the individual travel time (e.g. (Poorzahedy & Rouhani, 2007)). These objectives could lead to different solutions, better known as the Braess' paradox (Braess, 1968). This problem is solved exactly by Bagloee et al. (2018), however it only finds solutions for small instances up to ten clients and fifteen facility locations.

Next to the heuristic methods introduced before, linear regressions (LR) are regularly proposed to estimate the solution of large scale problems. Cassioli et al. (2012) employ LR techniques to estimate the solution of a computationally expensive problem. As BLP are considered very difficult this is rather convenient. Bagloee et al. (2018) develop a hybrid method based on this supervised learning technique to solve a nonlinear problem and finds highly promising results. We adopt the idea of Bagloee et al. (2018) to solve the c-UFL.

3 Problem Description

In the current situation, a given set of employees commute daily to their main office, defined as the current physical workplace for employees of an employer. The set of employees might have different employers, hence different main offices. Our goal is to allocate remote offices such that the utility of employees is maximized. Firstly, a remote office is defined as a workplace that is geographically separated from the employer's main office. Secondly, the utility is defined as the willingness to accept a cut in monthly paycheck in exchange for teleworking two days a week from a remote office (Lara-Pulido & Martinez-Cruz, 2023), which is equivalent to the willingness to pay to work in a remote office. In consequence, a maximization of the utility results in the largest paycheck drop for the employers. Primarily, a short description of all problems is given. Secondarily, we present all suppositions. These are separated into three categories: Employees, Remote offices, and Road Network. Table 1 reflects whether a presumption is of essence for the problems covered in this paper.

The problems

As described before, a facility location problem is implemented to maximize utility. Firstly, no additional constraints are incorporated such that it reduces to the standard uncapacitated variant. Secondly, capacities are introduced to create a more realistic scenario. Thirdly, information on the traffic is incorporated to include routing of employees in the decision process. Subsequently, we discuss the definitions and underlying assumptions on the various components of these problems.

Employees

An employee is defined as an individual that works part-time or full-time under contract for a certain company. In the current situation, all employees commute to the main office every single working day. This is rather inconvenient in a congested city, hence they admire a reduction in travelling time. It is assumed that these employees have no difficulty walking or driving.

The utility is based on the commuting time. In most cases, the utility increases with a decreasing commuting time. If an employee is assigned to the main office, the utility of the employee is zero. Moreover, the commuting time is approximated by the Manhattan distance for the UFLP and CFLP with a speed of five kilometers per hour (walking speed). The rational behind promoting walking is that it will reduce the number of cars on the road and that the utility has been estimated by assuming walking speed (Lara-Pulido & Martinez-Cruz, 2023).

The c-UFLP incorporates real-life information on congestion for intersections and roads to approximate the travel time depending on the number of employees utilizing a road. The more congested a road, the higher the travel times. For simplicity, it is assumed that all employees commute to and from work at the same time during peak hours and that all employees with a positive attitude towards remote offices prioritize the proximity of a remote office such that it prefers a main office over a remote office that is further away. The definition of positive attitude is given in Section 4. Consequently, the objectives of the employers (reduce cost) and employees (reduce travel time) do not align and form a strategic game. The goal is to find the Nash Equilibrium such that no player can unilaterally change their decision resulting in a better outcome. In this game, we assume that the employers and employees have perfect information such that every employee knows the routes of other employees and the entire set of open remote offices.

Remote offices

A remote office is defined as a decentralized office building that could facilitate the working environment of employees working in different companies in one building. It is assumed that the remote offices can be located anywhere on the grid. Hence, it does not take current buildings, infrastructure, etc. into account. Note that this is a drastic simplification of reality. Although it may not be very realistic, it does give a well-distributed grid. A hexagon office grid is utilized, in which the distance from the center of one hexagon to the center of any of its neighboring hexagons is always the same such that the grid is more convenient than a square grid in this implementation. No information on the possible sizes of these facilities is given and it is assumed that every remote office is identical.

Road Network

We employ a network of data points, based on the road network of Mexico City, to include traffic decisions. These data points are (assumed to be) intersections and there exists a road between two horizontally and vertically neighbouring intersections. Real life data is used to monitor the congestion level of an intersection and road. We assume that the streets are highly congested such that only a small number of additional vehicles can be capacitated on all roads. This is later needed to approximate the travel time on the road. Lastly, the remote offices, main offices, and employees' households are connected to the road network by a road to the nearest intersection.

Table 1: Applied assumptions in this paper

Assumptions	**UFLP**	**CFLP**	**c-UFLP**
Employees			
Discrete set of employees	x	x	x
Manhattan Distance	x	x	x
Commuting speed: 5km per hour	x	x	
Commuting speed: 30km per hour			x
Congestion-adjusted travel time			x
Prioritize proximity			x
Game-theoretic setup			x
Facilities			
Hypothetical Locations*	x	x	x
Hexagon grid	x	x	x
Network			
Network of roads and intersections			x

* The current infrastructure on the locations is not taken into account

4 Data

To examine the quality of the methods, we use data supplied by Centro Transdisciplinar Universitario para la Sustentabilidad and the authors of Lara-Pulido & Martinez-Cruz (2023). The information on locations of the employees' households, main offices, possible remote office locations, and road network in Mexico City are represented by means of the geographic coordinate system. These are transformed to Cartesian coordinates by Lambert Conformal projection with the average location of the employees' households as the centre. We partition this section into three subsections: Employees (Section 4.1), Remote offices (Section 4.2), and Road network (Section 4.3). We not only introduce and analyze the data but we also discuss applicable assumptions.

4.1 Employees

We work with a sample of 1,023 employees living in or near Mexico City, we assume that this sample represents all employees in the city well enough. In consequence, the location of remote offices would serve not only this sample but the entire population of Mexico City. We denote the number of employees in this subsection as n. Firstly, we elaborate on the information known about employees. This includes:

1. *Longitud_Or*: the longitude coordinates of the employee's household,

2. *Latitud_Or*: the latitude coordinates of the employee's household,

3. *Longitud_De*: the longitude coordinates of the main office of the employee,
4. *Latitud_De*: the latitude coordinates of the main office of the employee,
5. *wtp_sq*: the utility when the office is located next to the household of the employee,
6. *wtp_time*: the change in utility when an employee has to travel an additional hour.

Consequently, the utility of employee e is calculated as:

$$utility_e = wtp\_sq_e + wtp\_time_e * commuting\ time_e \tag{1}$$

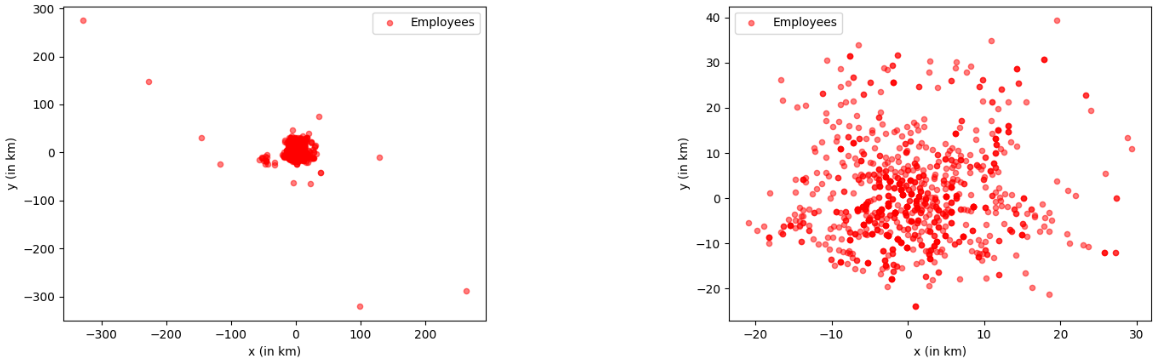
Where the $commuting\ time_e$ for employee e, is in hours. Employees with unrealistic characteristics are removed from the dataset. Unrealistic characteristics are positive *wtp_time* (increasing utility with larger commuting times) and absurdly high *wtp_sq* with low *wtp_time* (these employees only desire an office next to their household). Moreover, employees not within 50 kilometers of the centre of Mexico City are discarded as we focus on assigning employees to offices in a congested city and they will never yield a positive utility to a remote office due to the large distances. Note that employees with negative utility are not discarded, it assumed plausible that some employees do not want to work at a remote office in any situation. These employees affect the solution of the c-UFL as the routes of these employees to their main office could influence the routes of other employees. These modifications lead to the changes in characteristics shown in Table 2.

Table 2: Employee characteristics before and after cleaning (n = 1,100 before, n = 1,023 after)

Characteristics	Metric	Before Cleaning	After Cleaning
Longitude	Average	-99.152	-99.145
	Std. Dev	0.183	0.077
	Minimum	-102.341	-99.352
	Maximum	96.677	-98.872
Latitude	Average	19.410	19.414
	Std. Dev	0.177	0.095
	Minimum	16.522	19.194
	Maximum	21.852	19.763
Intercept Utility	Average	88.074	202.978
	Std. Dev	2,039.465	355.370
	Minimum	-49,659.946	-192.525
	Maximum	22,232.979	5,982.095
Slope Utility	Average	-84.570	-151.002
	Std. Dev	1,586.599	264.813
	Minimum	-31,678.400	-6,283.073
	Maximum	34,343.207	-22.334

The characteristics of longitude (x-axis) and latitude (y-axis) change mainly due to the deletion of employees not living in Mexico City. Consequently, the standard deviation decreases and the range is tighter. The deletion of outrages intercept (*wtp_sq*) and slope (*wtp_time*) utility results in a significantly lower standard deviation on both characteristics. For example, the employee with the smallest intercept utility of $-49,659.946$ has the highest slope utility $(34,343.207)$. This employee loves commuting, which would result in allocating the employee to the furthest remote office possible. Moreover, the employee with the highest intercept utility has the lowest slope utility such that the worker only benefits from a remote office next to their home. All employees with these characteristics are deleted. Lastly, we consider every employee with a positive intercept utility to be positive towards remote offices.

Figures 1a and 1b give a visual representation of the employees' household locations in Mexico City. Figure 1a visualizes the original set of employees. It is clear that most of the employees live in/near the city ($\approx 93\%$) while a fraction of the employees live hundreds of kilometres away. Following the adaptions mentioned in this section, the set of employees after cleaning is given in Figure 1b.



(a) Original employees data

(b) Employees after cleaning the data

Figure 1: Location of employees

Main offices

In the current situation, some employees have to travel hundreds of kilometers to the main office, these employees are willing to travel large distances to the remote office and highly value shorter commuting times. For example, the employee travelling to the upper right main office in Figure 2 is willing to travel two and a half hours while the average employee is willing to travel circa one hour to the remote office. Figure 3 shows the current situation of all employees that have to travel at least 50 kilometers to their main office.

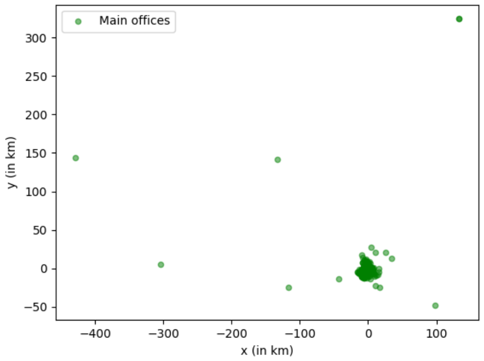


Figure 2: Main offices of employees after cleaning the data

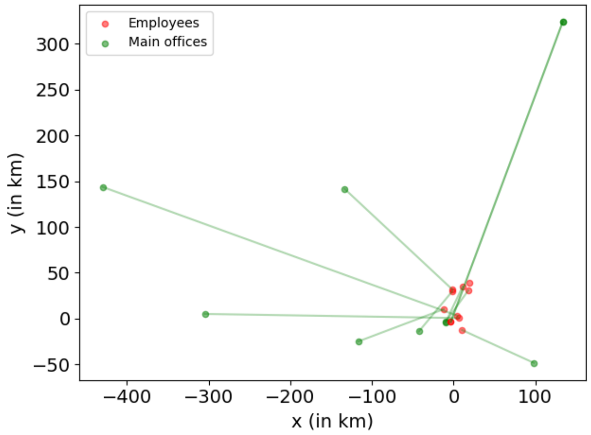


Figure 3: Current situation employees ≥50km to main office

It is clear that the current situation is far from ideal, hence the remote offices are likely to improve the happiness of these employees.

4.2 Remote offices

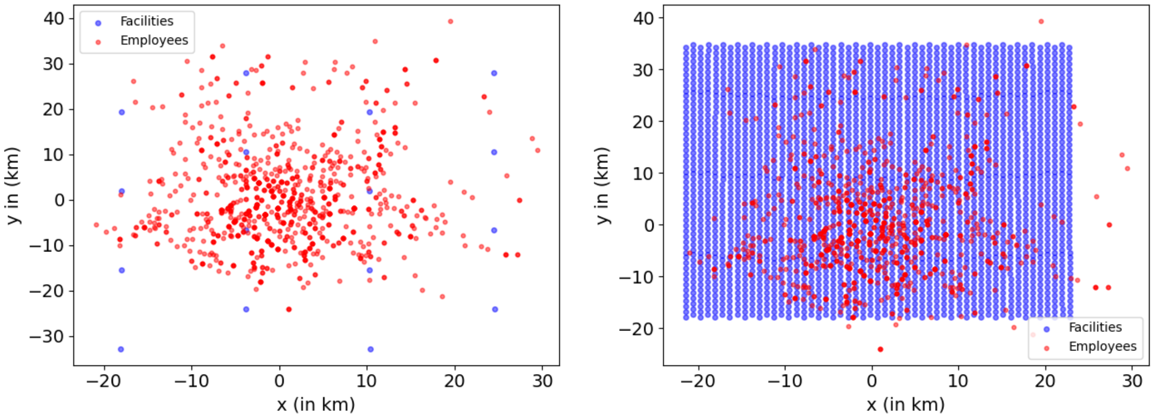
We create the possible remote office (facility) locations by utilizing a hexagon grid. This type of grid maximizes the coverage over an area better than other types of grids (Birch et al., 2007). We consider facilities within the city and a remote office is build in the centre of a hexagon. The characteristics of six distinct instances are displayed in Table 3. The instance names depend on the Euclidean distance between two neighboring facilities. For example, in the "20km" instance, the facilities are 20 kilometers apart.

Table 3: Characteristics remote office instances

		Longitude				**Latitude**				
Instance	**Facilities**	**Average**	**Std. Dev**	**Min**	**Max**	**Average**	**Std. Dev**	**Min**	**Max**	**Density***
20km	16	-99.122	0.156	-99.324	-98.919	19.388	0.185	19.115	19.661	0.006
10km	56	-99.155	0.156	-99.391	-98.919	19.486	0.159	19.232	19.739	0.020
4km	256	-99.149	0.125	-99.351	-98.946	19.497	0.144	19.255	19.739	0.112
2km	992	-99.142	0.125	-99.351	-98.932	19.493	0.140	19.255	19.731	0.426
1.2km	2,703	-99.146	0.124	-99.357	-98.935	19.486	0.138	19.486	19.722	1.161
0.6km	10,608	-99.144	0.122	-99.353	-98.935	19.494	0.138	19.256	19.731	4.579

* Density = Number of remote offices per km^2

The covered area of the facilities remain approximately the same such that there are solely located in the centre of Mexico City. Hence, the more facilities in an instance, the higher the density. To give an idea on the visual representation of the facility locations. Figures 4a and 4b show the remote office grids for 20 and 1.2 kilometers apart respectively. All other facility grids are placed in Appendix 8.1.



(a) Instance 20km

(b) Instance 1.2km

Figure 4: Figures remote offices and employees

4.3 Road network

The road network includes $1,469,433$ data points. A data point reflects a point of the road network with a corresponding traffic parameter, indexed from zero to four, zero being "non-congested" and four "highly congested". This point could be a highway, residential street, intersection, etc. We consider a data point if it falls within the minimum and maximum coordinates of employees/facilities such that every facility and employee can be accessed from an intersection. Furthermore, the number of data points is reduced by making a subset that selects one per 100 data points in a way that the points are evenly spaced, reflected in Figure 5, in which we retain the selected intersections.

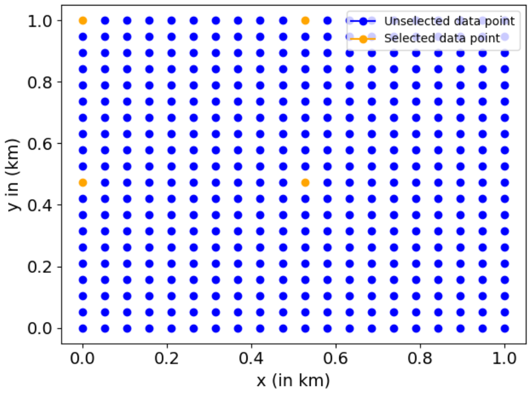


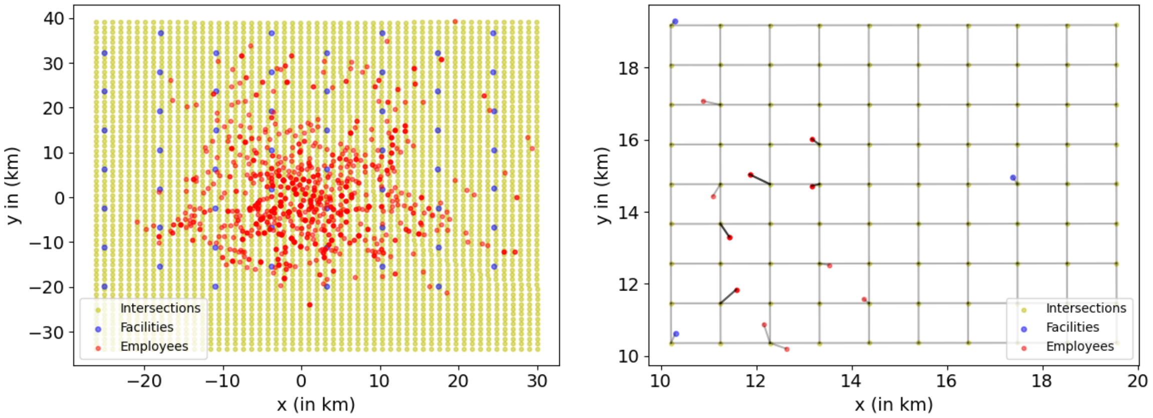
Figure 5: Subset network data points

We show the summary statistics of the subset compared to the original set in Table 4. It can be concluded that the effect on the statistics of sampling is marginal. The range and standard deviation of the coordinates decreases and the average traffic parameter increases.

Table 4: Network characteristics before and after cleaning (n = 1,469,433 before, n = 3,685 after)

Characteristics	Metric	Before Cleaning	After Cleaning
Longitude	Average	-99.123	-99.137
	Std. Dev	0.304	0.158
	Minimum	-99.649	-99.402
	Maximum	-98.597	-99.004
Latitude	Average	19.502	19.439
	Std. Dev	0.327	0.191
	Minimum	18.936	19.117
	Maximum	20.068	19.609
Traffic Parameter	Average	0.017	0.047
	Std. Dev	0.159	0.262
	Minimum	0.000	0.000
	Maximum	4.000	4.000

Figure 6a visualizes the network on the set of employees and remote office locations of instance 10km. The "Congested" intersections have traffic parameter greater than zero, while others have a parameter of zero. Every horizontally or vertically neighbouring intersection is connected by a road such that there exist no diagonal roads connecting two intersections and the employees, remote office, and main offices are connected to the nearest intersection. The traffic parameter tp_a of a road (or edge) a is equal to the maximum of the traffic parameters of the adjacent nodes. In consequence, $tp_a = \max\{tp_{n_1}, tp_{n_2}\}$ with n_1 and n_2 the edge's end nodes. The distance between two data points (employees, (remote) offices, intersections) is computed by means of Manhattan distance. Figure 6b reflects a small portion of the entire graph to give an idea on the road network.



(a) Intersection network with employees/remote offices

(b) Road network subplot

Figure 6: Road network

5 Methodology

To answer the three research questions, we divide this section into four subsections. We introduce the general sets, variables, parameters, and objectives of all problems in Section 5.1. Section 5.2 introduces the methods to solve the UFLP while we expound on the methods for solving the capacitated version of the FLP in Section 5.3. These problems assume a constant travelling speed. However, Mexico City is a highly congested city such that this is a harsh simplification of reality. Consequently, we extend the standard UFLP by taking the traffic situation into account in Section 5.4.

5.1 Variables, sets and objectives

Every method in our paper ultimately has the same goals: maximizing utility of employees and minimizing the number of open remote offices. We index the set of employees E by e, the set of potential remote offices F by f and the set of main offices M by m. To clarify, the terms facility and remote office are used interchangeably but have the same definition. Given the sets, we use the following decision variables in all methods:

- $u_{ef} = 1$ if employee e is assigned to remote office f and 0 otherwise, $\forall e \in E$, $\forall f \in F$,
- $b_f = 1$ if remote office f is opened and 0 otherwise, $\forall f \in F$

Not only the decision variables remain the same, but also this set of parameters has the same interpretation for all problems:

- d_{ef} = the travelling distance in hours between employee e and remote office f, $\forall e \in E$, $\forall f \in F$
- $\alpha_e = wtp\_sq_e$ = the utility of employee e with a remote office around the corner of his/her residence, $\forall e \in E$
- $\beta_e = wtp\_time_e$ = the change in utility per commuting hour of employee e, $\forall e \in E$

The first objective is to maximize the utility of employees. This objective follows the idea implied by (1) and is denoted as B_1:

$$\max_u \quad B_1 = \sum_{e \in E} \sum_{f \in F} u_{ef}(\alpha_e + \beta_e d_{ef}) \tag{2}$$

Solely maximizing the utility could result in a large number of open facilities with low number of assigned employees per remote office. Therefore, the second objective minimizes the number of open remote offices which is referred to as B_2:

$$\min_b \quad B_2 = \sum_{f \in F} b_f \Rightarrow \quad \sum_{f \in F} b_f \leq \epsilon \tag{3}$$

Observe that these objectives are conflicting, maximizing the utility will lead to more open facilities while less remote offices will lead to lower system utility. Hence, a multi objective approach needs to be implemented. Examples of well researched approaches are weighted sum (Marler & Arora, 2010), lexicographic optimization (Isermann, 1982), and Achievement Scalarizing Function (ASF) (Wierzbicki, 1982). We adopt the ϵ-constraint approach to yield a well distributed Pareto front (Haimes, 1971). This approach treats one objective as a constraint and the set of Pareto optimal solutions is obtained by varying the ϵ. In this paper, we restrict the number of opened facility such that objective B_2 is implemented as the constraint. Defining the range of the ϵ is straightforward in this case. Maximizing objective B_1 results in the highest possible number of open facilities (F_{B_1}, Nadir point) while zero facilities is the Utopia point ($\epsilon \in [0, F_{B_1}]$). Although every method implements the constraint to account for a multi objective approach, the implementation differs and will be elaborated on in the corresponding subsections.

5.2 Uncapacitated Facility Location Problem

The UFLP involves locating facilities to efficiently supply customer demand. We define efficiency as maximizing the system utility (summed utility of all employees). The mixed integer linear program (MILP) of the UFLP is displayed in model (4) - (7). We employ the decision variables and parameters introduced in Section 5.1 such that the MILP can be formulated as:

$$
\begin{aligned}
\max_{b,u} \quad & B_1 = \sum_{e\in E}\sum_{f\in F} u_{ef}(\alpha_e + \beta_e d_{ef}) && && (4)\\
\text{s.t.} \quad & 0 \le u_{ef} \le b_f, && \forall e \in E, \forall f \in F && (5)\\
& \sum_{f\in F} u_{ef} \le 1, && \forall e \in E, && (6)\\
& b_f \in \{0,1\}, && \forall f \in F. && (7)
\end{aligned}
$$

The objective (4) maximizes the system utility. The utility for employee e can be shortened to $\alpha_e + \beta_e d_{ef} = c_{ef}$ in case of constant travel speed. Constraint set (5) ensures that an employee can only be assigned to an open facility and constraints (6) make sure that an employee $e \in E$ can be assigned to at most one facility. If an employee is not assigned to a remote office, the employee e commutes to their main office and receives zero utility. The set of constraints (7) states that a facility can either be closed “0” or opened “1”. Note that there is no binary constraint needed for u_{ef} as we are maximizing the objective and there is no restriction on the number of employees assigned to one remote office. Hence, it is never optimal to fractionally assign an employee to a remote office. The standard MILP is not expected to solve large instances in reasonable time due to the NP-hardness so we implement two variants of Benders Decomposition (BD). To start off, we give a general derivation of the BD.

The standard BD splits the original problem (4) - (7) into two problems: the master problem (MP) and subproblem (SP). The former identifies the optimal solution for the variables b, while the latter identifies the optimal solution of u given the set of open remote offices. In every iteration, the dual of the SP supplies optimality cuts for the master problem to improve the solution. Firstly, we introduce the MP, (dual) SP, and optimality cuts. Secondly, we show that these cuts are valid.

The optimality cuts are generated by the dual of the SP. The SP is equal to model (4) - (7) for a given $\bar{b}$:

$$
\begin{aligned}
\max_{u} \quad & B_1 = \sum_{e\in E}\sum_{f\in F} u_{ef} c_{ef} && && (8)\\
\text{s.t.} \quad & 0 \le u_{ef} \le \bar{b}_f, && \forall e \in E, \forall f \in F && (9)\\
& \sum_{f\in F} u_{ef} \le 1, && \forall e \in E, && (10)
\end{aligned}
$$

Let us define $\rho_{ef} \ge 0$, $\forall e \in E$, $\forall f \in F$ and $\gamma_e \ge 0$, $\forall e \in E$ as the dual variables of constraints (9) and (10) respectively. The Lagrangian relaxation of the SP is:

$$
\begin{aligned}
\max_{u\ge 0} \mathcal{L}(\rho,\gamma,u) &= \max_{u\ge 0} \sum_{e\in E}\sum_{f\in F} u_{ef}c_{ef} + \sum_{e\in E}\sum_{f\in F}\rho_{ef}(\bar{b}_f - u_{ef}) + \sum_{e\in E}\gamma_e(1 - \sum_{f\in F} u_{ef}) && (11)\\
&= \max_{u\ge 0} \sum_{e\in E}\sum_{f\in F} u_{ef}(c_{ef} - \rho_{ef} - \gamma_e) + \sum_{e\in E}\sum_{f\in F}\rho_{ef}\overline{b_f} + \sum_{e\in E}\gamma_e && (12)
\end{aligned}
$$

Given the Lagrangian relaxation $\mathcal{L}(\rho, \gamma, u)$ and the solution $\bar{b}_f$ of the MP, $\forall f \in F$. The dual of the SP is:

$$\min_{\rho,\gamma} \quad (\sum_{e \in E} \sum_{f \in F} \rho_{ef} \bar{b}_f + \sum_{e \in E} \gamma_e) \tag{13}$$

$$\text{s.t.} \quad \gamma_e + \rho_{ef} \geq c_{ef}, \qquad \forall e \in E, \forall f \in F, \tag{14}$$

$$\gamma_e \geq 0, \qquad \forall e \in E, \tag{15}$$

$$\rho_{ef} \geq 0, \qquad \forall e \in E, \forall f \in F. \tag{16}$$

By weak duality, substituting any feasible solution of the dual SP, $\bar{\rho}_{ef}$, $\forall e \in E$, $\forall f \in F$ and $\bar{\gamma}_e$, $\forall e \in E$, forms an upperbound and thus a valid optimality cut. This optimality cut is added to the MP in every iteration. Generally, feasibility cuts are added if the primal SP is infeasible (dual SP unbounded). Note that in our problem it is possible not to assign an employee to any remote office such that every solution is feasible, therefore no feasibility cuts are needed. We denote the MP as:

$$\max_{\eta,b} \quad \eta \tag{17}$$

$$\text{s.t.} \quad \eta \leq \sum_{e \in E} \sum_{f \in F} \bar{\rho}^h_{ef} b_f + \sum_{e \in E} \bar{\gamma}^h_e, \qquad \forall h \in H \tag{18}$$

$$b_f \in \{0, 1\}, \qquad \forall f \in F, \tag{19}$$

Where constraint set (18) is the set of optimality cuts. The set H contains all distinct dual variables solutions and these cuts are added until the optimal solution is found. A valid optimality cut is defined as a cut that does not cut off any feasible solution. The optimal solution of the dual variables is the result of solving the dual SP. It is clear that model (13) - (16) is not restricted by the value of $\bar{b}$, hence it cannot cut off any feasible solution. Additionally, the solution is optimal when the lower bound (objective (dual) SP) is equal to the upper bound (objective MP).

Lastly, we add the ϵ-constraint (3) to the MP for all BD variants in this paper to account for the multi objective approach. We introduce two more efficient BDs: Section 5.2.1 introduces an altered form, introduced by Beheshti Asl & MirHassani (2019), of the well-known BD and Section 5.2.2 incorporates a more efficient SP into the BD to solve large scale instances.

5.2.1 Multi-cut Benders Decomposition (MCBD)

In this section we describe the Multi-cut Decomposition (MCBD) from Beheshti Asl & MirHassani (2019). Whereas the standard BD implements a Kelley's scheme (Kelley, 1960), which adds a single cut to the MP, produced by the SP, every iteration i. The MCBD deviates from the standard Kelley's scheme by solving the MP and storing at most L best feasible solutions, $L \geq 1$, in set O_i in iteration i. If the number of feasible solutions is smaller than L, all feasible solutions are stored and $|O_i| < L$. The set of best feasible solutions is attained by the "PoolSearchMode" in Gurobi. The $|O_i|$ solutions $\bar{b}^l$, $\forall l \in O_i$, are the parameters for the SP. In consequence, $|O_i|$ independent problems have to be solved. These are combined into one dual subproblem:

$$\begin{aligned}
\min_{\rho,\gamma} \quad & \sum_{l \in O_i} (\sum_{e \in E} \sum_{f \in F} \rho^l_{ef} \bar{b}^l_f + \sum_{e \in E} \gamma^l_e) & & (20) \\
\text{s.t.} \quad & \gamma^l_e + \rho^l_{ef} \geq c_{ef}, & \forall l \in O, \forall e \in E, \forall f \in F, & \quad (21) \\
& \gamma^l_e \geq 0, & \forall l \in O, \forall e \in E, & \quad (22) \\
& \rho^l_{ef} \geq 0, & \forall l \in O, \forall e \in E, \forall f \in F. & \quad (23)
\end{aligned}$$

Although the dual of the SP (20) - (23) has changed compared to the general BD, the MP is identical to (17) - (19). The solution of the dual is used for the right-hand side of the optimality cut (18). Note that the size of set H increases with $|O_i|$ in iteration i and that the MP is an ILP.

5.2.2 Uncapacitated Knapsack Benders Decomposition (UKBD)

The MCBD is solely tested on small size instances (Beheshti Asl & MirHassani, 2019), in contrary to the size of the data we use. It is reasonable to assume that the dual of the SP (20)-(23) with millions of variables will not be solved in reasonable time. Fischetti et al. (2017) tackles this problem by making use of the computational benefits of a Knapsack SP. Like the authors of the paper, we solve a relaxation of the MP such that fractional solutions are feasible. To obtain the optimal binary solution, we extend the original paper by adding linear constraints to the MP to tighten the formulation. In particular, this is better known as a branch-and-bound framework and is elaborated on in this section. In later stages, the uncapacitated Knapsack BD is referred to as UKBD.

Benders Decomposition

The MP (24) - (27), in essence, has the same characteristics as the MP (17) - (19) in the previous subsection. Nevertheless, this MP is an LP relaxation and the variable η is now decomposed into $|E|$ variables η_e, $\forall e \in E$. These cuts are called "fat" cuts while the "slim" cuts (18) is a summation over all employees. This results in altered optimality cuts, displayed as constraint set (25). The number of optimality cuts, initialized at zero, for each employee $e \in E$ is o^u_e. Here o^u_e increments by at most one per iteration t of the BD.

$$\begin{aligned}
\max_{\eta,b} \quad & \sum_{e \in E} \eta_e & & (24) \\
\text{s.t.} \quad & \eta_e \leq \psi^i_e(b), & \forall e \in E, i = 1, ..., o^u_e, & \quad (25) \\
& 0 \leq b_f \leq 1, & \forall f \in F, & \quad (26) \\
& \eta_e \geq 0, & \forall e \in E. & \quad (27)
\end{aligned}$$

Where we exactly define $\psi_e^i(b)$ in (25) further in this subsection. Given the (fractional) solution $\bar{b}$, the SP (28) - (31) can be seen as a Knapsack problem for every employee $e \in E$ (Fischetti et al., 2017):

$$\begin{aligned}
\psi_e(b) = \max\nolimits_u \quad & \sum_{f \in F} u_{ef} c_{ef} && && (28)\\
\text{s.t.} \quad & u_{ef} \leq \bar{b}_f, && \forall f \in F && (29)\\
& \sum_{f \in F} u_{ef} \leq 1, && && (30)\\
& u_{ef} \geq 0, && \forall f \in F, && (31)
\end{aligned}$$

Constraints (30) is an inequality while this is an equality constraint in the standard Knapsack problem. However, this does not matter as it only increases the feasible region but does not impact the optimal solution. The Knapsack problem makes the creation of optimality cuts computationally more attractive. Following Fischetti et al. (2017), we obtain them by employing Generalized Benders Cuts for which we give a derivation.

Intermezzo Generalized Benders Cut

Given a convex maximization program. The Generalized Benders decomposition cut is (Geoffrion, 1972):

$$\eta_e \leq \psi_e(\bar{b}) + \sum_{f \in F} \bar{s}_f (b_f - \bar{b}_f) \tag{32}$$

with $\bar{s}_f$ a subgradient of ψ_e in $\bar{b}$, $f(u,b)$ is the objective function (28) and $g_r(u,b)$ the rth constraint of the set of R constraints (let R be constraints (29) and (30)) and μ_r^* the corresponding optimal solution of the dual variable. All constraints in the model are of the form $g_r(\cdot) \leq 0$. Consequently, the subgradient can be computed by:

$$\bar{s}_f = \frac{\partial f(u,b)}{\partial b_f} + \sum_{r=1}^{R} \mu_r^* \frac{\partial g_r(u,b)}{\partial b_f} \tag{33}$$

This cut imposes an upper bound (maximization problem) on the objective value depending on the solution of b but does not excessively restrict the feasible region in any way such that this cut is valid.

The Lagrangian relaxation of the convex problem (28) - (31) for employee $e \in E$ is:

$$\max_{u \geq 0} \mathcal{L}_e(\rho, \gamma, u) = \max_{u \geq 0} \sum_{f \in F} u_{ef} c_{ef} + \sum_{f \in F} \rho_{ef}(\bar{b}_f - u_{ef}) + \gamma_e (1 - \sum_{f \in F} u_{ef}) \tag{34}$$

Given the Lagrangian relaxation for employee e, the subgradient is equal to the dual variable of constraint (29), $\bar{s}_f = \bar{\rho}_{ef}$. Hence, the Benders cut can be reduced to (Intermezzo 5.2.2):

$$\eta_e \leq \psi_e(\bar{b}) + \sum_{f \in F} \bar{\rho}_{ef}(b_f - \bar{b}_f) = \psi_e^i(b) \tag{35}$$

For employee e, it remains to find the optimal values of scalar $\psi_e(\bar{b})$ (the objective value of the knapsack problem given $\bar{b}$) and $\rho_{ef}, \forall f \in F$. The Dantzig algorithm produces the optimal primal and dual variable solutions (Cormen et al., 2022). Initially, it sorts the facilities in non-increasing utility for employee e and defines a critical facility k. Assuming that the sorted order of $\bar{b}$ corresponds to the sorted order of the facilities:

$$\sum_{f=1}^{k-1} \bar{b}_f < 1 \leq \sum_{f=1}^{k} \bar{b}_f$$

Given the critical facility k, we check if $c_{ek} < 0$ and if so the critical facility k is updated to the main office of employee $e \in E$ with corresponding $c_{ek} = 0$. The optimal primal solution is:

$$\bar{u}_{ef} = \begin{cases} \bar{b}_f & \text{for } f < k, \\ 1 - \sum_{f=1}^{k-1} \bar{b}_f & \text{for } f = k \\ 0 & \text{for } f > k \end{cases}$$

In consequence, $\psi_e(\bar{b}) = \sum_{f \in F} c_{ef} u_{ef} = c_{ek}(1 - \sum_{f=1}^{k-1} \bar{b}_f) + \sum_{f=1}^{k-1} c_{ef} \bar{b}_f$. The optimal solution for the dual variable corresponding to constraint (29) γ is $\bar{\gamma} = c_{ek}$, Note that the dual variable is non-negative, hence the utility of employee to facility k cannot be lower than 0. This holds as we assume that an employee will go to the main office, with utility zero, if the best open facility results in negative utility. Moreover, the optimal solution for the variables $\rho_{ef}, \forall f \in F$, corresponding to constraints (30), is:

$$\bar{\rho}_{ef} = \begin{cases} c_{ef} - c_{ek} & \text{for } f < k, \\ 0 & \text{for } f \geq k \end{cases}$$

The dual objective, derived from the Lagrangian, equals $\bar{\gamma} + \sum_{f \in F} \bar{\rho}_{ef} \bar{b}_f = c_{ek} + \sum_{f=1}^{k-1} (c_{ef} - c_{ek}) \bar{b}_f$ which corresponds to the expression obtained before on $\psi_e(\bar{b})$. Strong duality holds, confirming that the values for the dual and primal variables are indeed optimal and that we can implement the cost $\psi_e(\bar{b})$ in (35):

$$\eta_e \leq c_{ek} + \sum_{f=1}^{k-1} (c_{ef} - c_{ek}) b_f = \psi_e^i(b) \tag{36}$$

The cut is valid because it does not cut off any feasible solutions by weak duality. Moreover, since we use the dual optimal solution in the creation of the cut, this cut can be interpreted as the attainable objective for an employee.

Consequently, at most $|E|$ cuts are added every iteration. According to Fischetti et al. (2017) for linear problems this 'fat' model outperforms the 'slim' model in which the constraints for every employee are combined to one constraint, therefore we implement the 'fat' model. In every node, every fifth iteration of the BD all generated non-binding optimality cuts are removed from the model to increase computational efficiency. When the optimality cut does not improve the upper bound, the BD is terminated and all non-binding optimality cuts are removed. The subsequent steps are considered next.

Branch-and-bound framework

The MP is an LP relaxation such that the optimal solutions of the BD are possibly fractional. Fischetti et al. (2017) implement a so called "lazy" strategy to find non-fractional solutions using a state-of-the-art solver (IBM ILOG Cplex 12.6). In the root node, they solve the LP relaxation of (4) - (7) using the UKBD described above. In all other branching nodes, they solve the LP-relaxation using the solver, check if the LP solution of the node violate any BD cuts. If so, they add them to the pool of cuts and the model is re-optimized in the node. The process continues until the maximum number of iterations in the node is met or the solution does not violate any cuts. Branching is applied, the pool of cuts is deleted and everything is repeated in the new nodes. It seems rather ineffective to solve the model from scratch in every node, hence we implement a branch-and-bound scheme to find the best possible non-fractional solution.

The fractional solution is cut off by restricting a fractionally open remote office to either '0' or '1' and optimize the more restricted versions again by means of the BD. The node is pruned if the upper bound is smaller than the best known lower bound, the optimal solution is integer, or the problem is infeasible. This process is repeated until there are no more non-pruned nodes such that the optimal integer solution is found. A proof that the framework converges to the optimal solution is given:

Proposition 1. *The branch-and-bound with Benders Decomposition finds the optimal non-fractional solution.*

Proof. Prove that 1) The BB does not cut relevant solutions, 2) Assume that BB does not cut relevant solutions, it converges to the optimal solution.

1) If a non-integer solution is found in a node, branching is applied to tighten the relaxation. The feasible region is split by restricting a fractional variable x_i, $0 \leq i \leq n-1$, to either 0 or 1, and the restricted version is solved in the next node. Hence, no integer feasible region is cut off. Moreover, no continuous region is cut off by the BD as we work with valid optimality cuts obtained by a valid LP relaxation.

2) The number of possible integer solutions is finite (2^n) with n being the number of variables. As a result, the branching tree terminates finitely as there are finite solutions. The tightened formulations in the nodes are solved by BD, hence it always gives a upper bound (we are maximizing). If the solution is integer, it is a lower bound such that the lower and upper bound are equal, hence it terminates. In conclusion, the BB terminates finitely and no possibly relevant solutions are cut off, hence it finds the optimal solution.

□

Within the nodes, the BD with Kelley's scheme is known to converge slowly. Hence, we implement a relatively easy and effective method in the root node: in-out (Fischetti et al., 2017). At each iteration two points exist, the current optimal (fractional) solution $\overline{b}$ of the MP and the stabilizing point b^*. The stabilizing point is initialized as a unit vector of length $|F|$. At every iteration t of BD, the stabilizing point moves towards the optimal solution $b^*_t \leftarrow \frac{1}{2}(\overline{b}_t + b^*_t)$. These points are combined:

$$\bar{b}_t \leftarrow \lambda \bar{b}_t + (1-\lambda) b_t^* + \delta(1, ..., 1) \tag{37}$$

This updated $\bar{b}_t$ is the input for the subproblem of the BD. The parameters λ and δ are set to 0.2 and $2 * 10^{-6}$ respectively. Adding a small value δ to the solution results has proven to positively effect the convergence of the algorithm (Ben-Ameur & Neto, 2007). If the BD does not find an improvement on the upper bound for five consecutive BD iterations, set $\lambda = 1$. After that, if no improvements are found for two consecutive iterations we set it to Kelley's scheme ($\delta = 0$, $\lambda = 1$). Lastly, if no improvements are found with Kelley's scheme for two more consecutive iterations, the BD is terminated and all non-binding generated cuts are removed. If the optimal solution of the MP in the root node is integer, the optimal IP solution is found. If not, branching is applied and the standard Kelley's scheme is implemented in all other nodes. Following some tests, utilizing the in-out method in other nodes than the root node does not significantly improve the convergence of the algorithm.

5.3 Capacitated Facility Location Problem

Infinite capacity offices is not only unrealistic but also impractical. Model (38) - (42) is the MIP for the capacitated facility location problem. With C_f the integer capacity for facility $f \in F$.

$$\begin{aligned}
\max\nolimits_{b,u} \quad & B_1 = \sum_{e \in E} \sum_{f \in F} u_{ef}(\alpha_e + \beta_e d_{ef}) && && (38)\\
\text{s.t.} \quad & \sum_{f \in F} u_{ef} \leq 1, && \forall e \in E, && (39)\\
& \sum_{e \in E} u_{ef} \leq C_f b_f, && \forall e \in E, \forall f \in F, && (40)\\
& 0 \leq u_{ef} \leq b_f, && \forall e \in E, \forall f \in F. && (41)\\
& b_f \in \{0, 1\}, && \forall f \in F, && (42)
\end{aligned}$$

Constraints (40) represent the capacity constraint. It is not possible to assign more (fractional) employees to a facility than its capacity. Notice that there are no integrality constraints on u as these are unnecessary in our case, the optimal solution is always binary due to the unit demand of the employees and the integer capacity. The problem (38) - (42), given a solution of open remote offices $\bar{b}$, reduces to the transportation problem, which is a special type of network flow problem with only edges between suppliers and clients.

Theorem Theorem 1, S. et al. (2012): Any minimum cost network flow problem instance whose demands are all integers has an optimal solution with integer flow on each edge.

This characteristic will be useful to derive the optimality cuts. Initially, we describe an exact approach from Fischetti et al. (2016) in Section 5.3.1 and propose a heuristic approach from T.-C. Lu & Yu (2013) in Section 5.3.2.

5.3.1 Capacitated Knapsack Benders Decomposition (CKBD)

This section is a follow-up on Section 5.2.2, the solution of the root node of the UKBD is the starting point of the CKBD. When the optimal (fractional) solution has been found in the root node, a different type of optimality cut is added to the MP until optimality is ensured. Fischetti et al. (2016) are the first to implement this idea but we extend their CKBD by a branch-and-bound framework.

Benders Decomposition

The master problem is defined as (43) - (49). Let o^c be the number of added capacitated optimality cuts. These cuts are obtained by the SP.

$$
\begin{aligned}
\max_{\eta_e,\eta,b} \quad & \eta && && (43)\\
\text{s.t.} \quad & \sum_{e\in E} \eta_e = \eta, && && (44)\\
& \eta_e \leq \psi_e^i(b), && \forall e \in E, i = 1, ..., o_e^u, && (45)\\
& \eta \leq \Phi^i(b), && i = 1, ..., o^c, && (46)\\
& 0 \leq b_f \leq 1, && \forall f \in F, && (47)\\
& \eta_e \geq 0, && \forall e \in E. && (48)\\
& \eta \geq 0. && && (49)
\end{aligned}
$$

The objective of this LP relaxation remains the same: maximizing the system utility. Constraints (45) are the result of UKBD in the root node, and ψ_e^i are defined as in (36). Constraints (46) are the generated capacitated optimality cuts. In the uncapacitated variant, the uncapacitated optimality cuts $\psi_e^i(b)$ could be separated by the employees, but this is not a viable approach for the capacitated cuts due to the capacity constraint. In consequence, the optimality cuts $\Phi(b)$ are generated in the following manner (the superscript is deleted for simplicity), and the final expression for them is provided later in (59).

The subproblem is (8) - (10) plus the capacity constraint and due to this constraint it cannot be decomposed in $|E|$ independent SPs and it reduces to the Transportation Problem:

$$\Phi(\bar{b}) = \max_u \sum_{f\in F}\sum_{e\in E} u_{ef}c_{ef} \tag{50}$$

$$\text{s.t.} \sum_{f\in F} u_{ef} \leq 1, \qquad \forall e \in E \tag{51}$$

$$\sum_{e\in E} u_{ef} \leq \bar{b}_f C_f, \qquad \forall f \in F \tag{52}$$

$$0 \leq u_{ef} \leq \bar{b}_f, \qquad \forall e \in E, \forall f \in F, \tag{53}$$

Let γ_e, $\forall e \in E$ be the dual variable of constraints (51). We obtain the optimal values of γ_e, $\forall e \in E$ by solving (50) - (53). The Lagrangian of this problem, given $\overline{\gamma}$ and $\bar{b}$, is:

$$\begin{aligned}\Phi(b) \leq &\quad \max_u\{\sum_{f\in F}\sum_{e\in E} u_{ef}c_{ef} + \sum_{e\in E}\overline{\gamma}_e(1 - \sum_{f\in F} u_{ef})|(52),(53)\} &(54)\\ = &\quad \sum_{e\in E}\overline{\gamma}_e + \max_u\{\sum_{f\in F}\sum_{e\in E} u_{ef}(c_{ef} - \overline{\gamma}_e)|(52),(53)\} &(55)\end{aligned}$$

The maximization part of (55) is equivalent to the continuous knapsack problem, which can be decomposed into $|F|$ independent SPs. Denote the objective values of these problems as $KP^f_{\overline{\gamma}}$. Hence, $\forall f \in F$:

$$KP^f_{\overline{\gamma}} = \quad \max_{u_f \in \mathbb{R}^{|E|}} \sum_{e\in E} u_{ef}(c_{ef} - \overline{\gamma}_e) \tag{56}$$

$$\text{s.t.} \sum_{e\in E} u_{ef} \leq C_f \tag{57}$$

$$0 \leq u_{ef} \leq 1, \qquad \forall e \in E. \tag{58}$$

Due to the fact that variables u are continuous, the problem is linear, strong duality holds such that the primal and dual optimal objective of the subproblem are equivalent. Hence, the primal problems (56) - (58) for fixed $\overline{\gamma}$ can be quickly solved via a sorting algorithm and serve as inputs for the optimality cuts (46). The sorting algorithm orders for every facility f in F the employees from highest to lowest utility and add them until capacity is reached. In consequence, the optimality cut in iteration i can be simplified to:

$$\eta \leq \sum_{e\in E}\overline{\gamma}_e + \sum_{f\in F} KP^f_{\overline{\gamma}} b_f = \Phi^i(b) \tag{59}$$

The optimal values for the variables γ are obtained by solving (50) - (53), which can be time consuming. Hence, approximating the solution of differentiable model (50) - (53) by techniques such as gradient optimization might be beneficial to reduce computation time at the cost of potentially less strong cuts. Either way, this cut does not influence the feasible region (valid) and restricts the objective value of the master problem. The effect of (fractionally) opening a facility is measured by the objective of the

Knapsack problems.

Branch-and-bound framework

We implement the BD explained in Section 5.3.1 in every node of the branch-and-bound tree. This algorithm is exactly the same as in Section (5.2.2). However, the parameters for the in-out implementation in the root node differ such that it is more conservative, resulting in a significant speed up. Although, the parameters α and λ are set to 0.9 and 0.1 respectively. The updating scheme is identical.

5.3.2 Heuristic approach (APMQEA)

Solving large instances of the CFLP exactly is not likely to terminate in reasonable time due to the non-polynomial time complexity. In this section, we discuss a quantum inspired meta-heuristic that is based on the NSGA-II. To give a clear understanding of the algorithm, we start of by explaining the cornerstones: NSGA-II and crowding distance. This is followed by a detailed elaboration and pseudo-code of the adaptive population multi objective quantum-inspired evolutionary algorithm (APMQEA).

NSGA-II

The NSGA-II is a form of genetic algorithm to obtain an approximated Pareto front, which is the set of non-dominated solutions. The overall idea of the algorithm is shown in Figure 7.

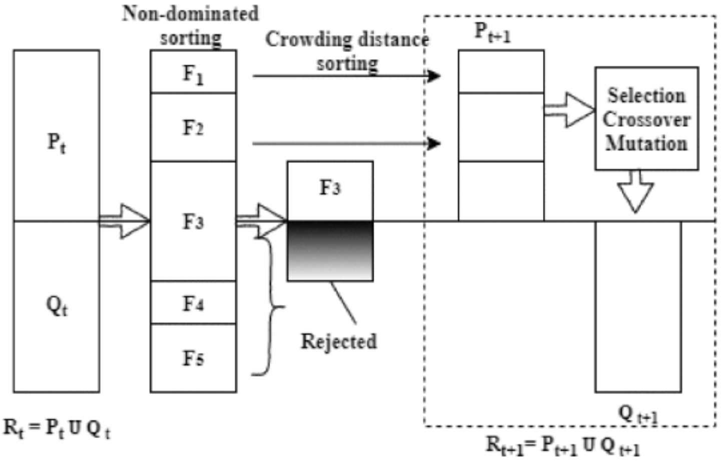


Figure 7: Procedure of NSGA-II, (Verma et al., 2021)

In each iteration t, new solutions (children solutions C_t) are created by altering the parent solutions P_t. The parents are defined as the solutions that are kept from the previous iteration. Here Q_t is the pool of solutions in iteration t containing the parent solution P_t and children solutions C_t ($Q_t = P_t \cup C_t$). These solutions are ranked by the non-dominated sorting. A rank is a set of solutions that are not strictly better than each other. However, rank 1 is strictly better than rank 2, etc. A solution is non-dominated (rank 1) if there is no solution that improves at least one objective while keeping the other objectives at least as good. If the solutions are sorted, the best O_{max} solutions are stored (the new parents). However, it is unlikely that the number of solution in the first number of ranks exactly sum up to O_{max}. In consequence, a subset of solutions in a rank have to be stored but it is not possible to sort them as it is given that solutions in the same rank are of identical quality. Hence, within a rank the solutions are stored based on the crowding distance measure, which attains the diversity of the solution in a rank. A solution not close to other solutions has a large crowding distance and is more likely to be selected. The mathematical formulation of the crowding distance d_i of solution i is given as:

$$\sum_{r=1}^{2} \frac{f_r(i+1) - f_r(i-1)}{f_r^{max} - f_r^{min}} \tag{60}$$

Where $f_r(i+1)$ and $f_r(i-1)$ are the values for objective r of the surrounding solutions of solution i in the rank respectively. Here f_r^{max} and f_r^{min} are the largest and smallest values of objective r in the entire rank. The solutions with minimum or maximum objective values are always retained. Crossover (combining solutions) and mutation (altering solutions), is applied to the new set of parents P_{t+1}, and the set C_{t+1} is obtained. The way of combining and altering the solutions depends on the implementation. This procedure is repeated till the termination criterion is met, normally this is when the a certain number of iterations is met or when the solution pool has not changed for a number iterations.

APMQEA

We incorporate the non-dominated sorting and crowding distance assignment in the APMQEA. The main idea evolves around the quantum theory and implements Q-bits. These bits employ a probability method to represent binary codes. Instead of '1' or '0' a Q-bit $\mathbf{q}_i$ is defined as:

$$\mathbf{q}_i = \left\| \begin{array}{c|c|c|c} \alpha_{i1} & \alpha_{i2} & .. & \alpha_{i|F|} \\ \beta_{i1} & \beta_{i2} & .. & \beta_{i|F|} \end{array} \right\| \tag{61}$$

Note that $0 \leq \alpha_f \leq 1$, $0 \leq \beta_f \leq 1$ for $f = 1, ..., |F|$. Here α_f^2 and β_f^2 are the probabilities that a Q-bit i has state '1' and '0' for facility f, respectively. In quantum physics $\alpha_f^2 + \beta_f^2 = 1$, in consequence (61) can be simplified to:

$$\mathbf{q}_i = \left\| \begin{array}{c|c|c|c} \alpha_{i1} & \alpha_{i2} & .. & \alpha_{i|F|} \end{array} \right\| \tag{62}$$

Every Q-bit $\mathbf{q}_i$ has one corresponding non-dominated solution $\mathbf{b}_i$ with corresponding objective values $\mathbf{o}_i$. The set of Q-bits in iteration t is Q_t ($|Q_t| = v$). Every Q-bit $\mathbf{q}_i \in Q_t$ is used to create k binary solutions $\mathbf{b}_{ij}$, $i = 1, ..., v$ and $j = 1, ..., k$ by generating k random number vectors $\mathbf{r}_{ij} = \begin{bmatrix} r_{ij}^1 & \cdots & r_{ij}^m \end{bmatrix}$ uniformly where $0 \leq r_{ij}^f \leq 1$ for $f = 1, ..., |F|$ and $j = 1, .., k$. The fth element of $\mathbf{b}_{ij}$ is '1' if $r_{ij}^f \leq \alpha_{if}^2$ and '0' otherwise.

The set B_t contains all solutions $\mathbf{b}_{ij}$, $i = 1, ..., v$ and $j = 1, ..., k$. The set O_{t-1} contains all non-dominated solutions of the previous iteration. The new non-dominated set $O_t = O_{t-1} \cup B_t$ is defined by applying non-dominated sorting and crowding distance assignment such that the best solutions are retained, note that at most O_{max} solution can be retained. Consequently, every set O_t for every iteration

t contains solely solutions of the first rank. If a $\mathbf{q}_i$ does not produce a non-dominated solution, it is discarded. Lastly, if $\mathbf{b}_{ij}$ is a non-dominated solution with corresponding objective value o_{ij} every α_{if}, $f = 1, ..., |F|$, of Q-bit $\mathbf{q}_i$ is updated by the rotation gate:

$$R(b_{il}, o_{ij}) : \alpha'_{if} = (\cos \Delta\theta_f - \sin \Delta\theta_f) \begin{bmatrix} \alpha_{if} \\ \sqrt{1 - \alpha_{if}^2} \end{bmatrix} \tag{63}$$

where the value of $\Delta\theta_f$ is formulated as ($\Delta\theta$ is given parameter):

1. if rank($\mathbf{b}_{ij}$) < rank($\mathbf{b}_{il}$) value and $\mathbf{b}_{il}^f = 0$ and $\mathbf{o}_{ij}^f = 1$: $\Delta\theta_f = -\Delta\theta$;
2. if rank($\mathbf{b}_{ij}$) < rank($\mathbf{b}_{il}$) and $\mathbf{b}_{il}^f = 1$ and $\mathbf{o}_{ij}^f = 0$: $\Delta\theta_f = \Delta\theta$;
3. otherwise: $\Delta\theta_f = 0$.

If $\mathbf{b}_{ij}$ is a newly generated non-dominated solution, the solution is updated based on $\mathbf{q}_i$, every $\mathbf{b}_{il}$, $l = 1, ..., k$, s.t. $l \neq j$ and the corresponding objective values $\mathbf{o}_{ij}$. If a non-dominated solution $\mathbf{b}_i$ from iteration $t - 1$ remains non-dominated in the current iteration t, we update the corresponding $\mathbf{q}_i$ based on every $\mathbf{b}_{ij}$, $j = 1, ..., k$ and $\mathbf{o}_i$. This updating scheme results in a dynamic list of non-dominated solutions. This procedure finds the set of open offices, but to assess the quality of the solution and apply non-dominated sorting, another problem has to be solved. In particular, we have to assign employees to the offices which is discussed further.

Evaluating the heuristic solution (k-MGAP)

Given a set of open facilities, evaluating the solution reduces to assigning the employees to one of the open facilities or the main office. We have shown that the binary constraints on the u decision variables can be relaxed, hence this problem reduces to a relaxation of the Generalized Assignment Problem. This LP is solvable in polynomial time but due to size of the instances, the larger instances cannot be solved within seconds. Hence, clustering algorithms align perfectly (e.g. (X. Zhang et al., 2016), (Peacock & Nicholson, 1991)). The most famous clustering technique is k-means (MacQueen et al., 1967). This unsupervised machine learning algorithm partitions a dataset into k subsets (clusters) consisting of similar data points, the center of all clusters (centroids) represents the mean of the cluster. A similar, but more applicable, unsupervised machine learning algorithm is k-medoids (Rdusseeun & Kaufman, 1987). Although this clustering technique similarly partitions the dataset into k clusters, it takes one of the data points as the centroid of a cluster. Hence, these possible centroids are the open facility locations in our case which makes this clustering technique more applicable for assigning employees to an open facility. We give the general framework of the k-MGAP heuristic in Figure 8.

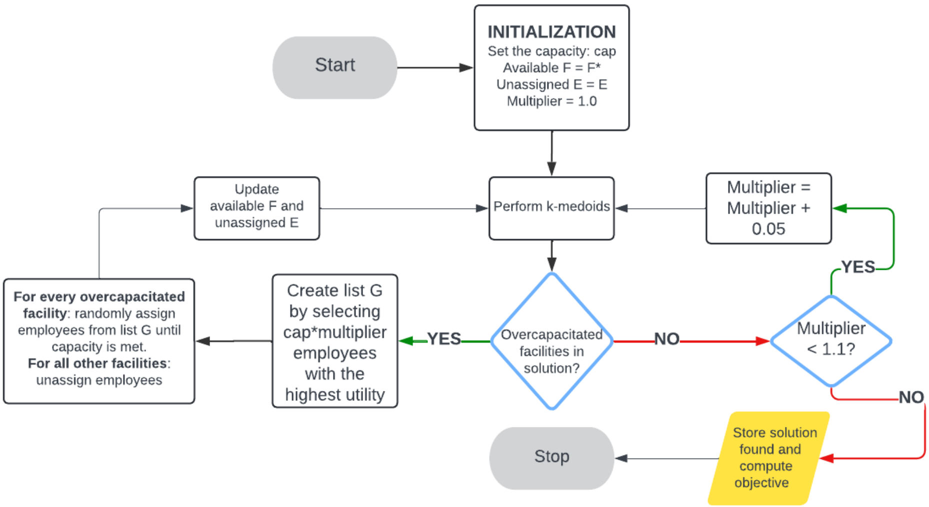

Start
INITIALIZATION
Set the capacity: cap
Available F = F*
Unassigned E = E
Multiplier = 1.0
Update available F and unassigned E
Perform k-medoids
Multiplier = Multiplier + 0.05
YES
For every overcapacitated facility: randomly assign employees from list G until capacity is met.
For all other facilities: unassign employees
Create list G by selecting cap*multiplier employees with the highest utility
YES
Overcapacitated facilities in solution?
NO
Multiplier < 1.1?
NO
Stop
Store solution found and compute objective

Figure 8: k-MGAP Heuristic Flowchart

This iterative algorithm solves a k-medoids problem up to the point that no capacity constraints are violated. Initially, given the open facilities $F^* \subset F$, all facilities are available and all employees $e \in E$ are unassigned. All main offices are denoted as centroid C^M. The k-medoids problem maximizes the total utility by assigning the employees to a remote office in F^* or to C^M. The solution is checked on overcapacitated facilities. For every overcapacitated facility, a list containing the $cap \cdot multiplier$, we define cap as the maximum capacity of a facility, employees with highest utility for that remote office is created. Thereafter, cap students are randomly selected from this list and assigned to the facility, which is then removed from F^*. Hence, the size of set F^* decreases per iteration. We unassign all other employees and perform another iteration. Stop when there exists no overcapacitated facility in the solution. Initially, the multiplier is set to 1.0 and update the multiplier by incrementing 0.05 every iteration. Following some experiments, this setting results in the best results. We terminate the algorithm if the multiplier is 1.1 and return the best solution and objective.

Overview algorithm On overview of the APMQEA algorithm is presented in Algorithm 1.

Algorithm 1 APMQEA Algorithm

```
Input: $E$ set of employees, $F$ set of facilities
Output: Feasible solution UFLP
t = 0
    /* The set $Q_t$ contains all Q-bits of iteration t */
/* $O_t$ contains all non-dominated solutions of iteration t */
initialize $Q_t$ and $O_t$
while $t < 1000$ and $Q_t$ has changed at least once in the last 5 iterations do
    t = t + 1
    Create the binary solution set $B_t$ from the Q-bits in $Q_{t-1}$
    Rebuild $O_t$ by selecting the non-dominated solutions from $O_{t-1} \cup \mathrm{B}_t$
    Create $Q_t$ by updating $Q_{t-1}$ following the updating scheme
```

5.4 Uncapacitated Facility Location Problem with congestion

Currently, the commuting times in many metropolises are impractically high. E.g. the average commuting time in Mexico City is 1.8 hours (Eurostat, 2020). However, incorporating traffic congestion into the decision making of facility locations has been underrepresented in literature. Hence, minimizing the individual travel time of the employees is the third objective of the model. In this set-up, we consider a game where the employers (leaders) maximize the system utility to decrease costs, while the employees (followers) minimize their own travel time. The decision making contains a hierarchical structure; the employers make strategic decisions on locating the remote offices while the employees react with their tactical strategies. The objectives could result to different solutions. For example, an employee can be assigned to a remote office if the total utility is maximized while it could be assigned to the main office if the individual travel time is minimized. Hence, we introduce a bi-level model to tackle this problem.

5.4.1 Problem formulation

The road network consists of nodes and undirected edges. The set of nodes $n \in N$ is a union of the sets of remote offices F, main offices M, employees E, and intersections U ($N = F \cup M \cup E \cup U$) and each road we describe in Section 4.3 corresponds to a road $a \in A$. To integrate traffic information, the travel time developed by the Bureau of Public Roads (BPR) (Council, 2000) on edge $a \in A$ is employed:

$$t_a(y_a) = T_a(1 + \pi_a(1 + tp_a)(\frac{y_a}{c_a})^4) \tag{64}$$

We replace the constant travel time by expression (64), which depends on the travel routes of all employees using a given road a. The free flow travel time T_a is the travel time on edge $a \in A$ given a Manhattan distance and constant travel speed. The congestion parameter is π_a, tp_a is the traffic parameter ranging from zero to four introduced in Section 4.3 while c_a is the capacity per hour of edge

$a \in A$ and equals 10. Here y_a is the variable that represents the total flow on edge $a \in A$ in the current solution, that is the number of employees commuting along the edge a.

Model (65) - (75) represents the bi-level model. Next to the sets, variables and parameters introduced in Section (5.1) we introduce several others. The newly employed sets are:

- K_{ef}, contains the K, with K given, shortest paths from employee $e \in E$ to remote office $f \in F$, $\forall f \in F$, $\forall e \in E$,
- K_e, contains the K, with K given, shortest paths from employee $e \in E$ to their main office, $\forall e \in E$,
- A^k_{ef}, contains all edges a of path k from employee e to remote office f, $\forall e \in E$, $\forall f \in F$.

The K shortest paths are predetermined by the K-shortest path algorithm introduced by Yen (1971), the algorithm can be found in Appendix 8.2. We give the definitions of the new variables and all unmentioned variables remain the same as introduced in Section 5.1:

- $u_e = 1$ if employee e goes to the main office and 0 otherwise, $\forall e \in E$,
- $z^k_{ef} = 1$ if employee e travels to remote office f using path k, $\forall e \in E$, $\forall f \in F$, $\forall k \in K_{ef}$,
- $z^k_e = 1$ if employee e travels to their main office using path k, $\forall e \in E$, $\forall k \in K_e$,

Finally, we introduce two additional parameters:

- $\delta^{ef}_{ak} = 1$ if path k that employee e uses to travel to remote office f traverses over undirected edge a, $\forall e \in E$, $\forall f \in F$, $\forall k \in K_{ef}$, $\forall a \in A^k_{ef}$,
- $\delta^e_{ak} = 1$ if path k that employee e uses to travel to their main office traverses over undirected edge a, $\forall e \in E$, $\forall k \in K_e$, $\forall a \in A$,

These sets, variables, and parameters are used to construct the following model:

$$\max_b \quad \sum_{e\in E}\sum_{f\in F}\sum_{k\in K_{ef}} z_{ef}^k(\alpha_e + \beta_e \sum_{a\in A_{ef}^k} t_a(y_a)) \tag{65}$$

$$\text{s.t.} \quad 0 \leq u_{ef} \leq b_f, \qquad \forall e \in E, \forall f \in F \tag{66}$$

$$b_f \in \{0,1\}, \qquad \forall f \in F \tag{67}$$

$$\min_{y,u_{ef},u_e,z_{ef}^k,z_e^k} \quad \sum_{a\in A}\int_0^{y_a} t_a(v)dv \tag{68}$$

$$\text{s.t.} \quad \sum_{f\in F} u_{ef} + u_e = 1, \qquad \forall e \in E \tag{69}$$

$$u_{ef} \leq b_f, \qquad \forall e \in E, \forall f \in F \tag{70}$$

$$\sum_{k\in K_{ef}} z_{ef}^k = u_{ef}, \qquad \forall e \in E, \forall f \in F \tag{71}$$

$$\sum_{k\in K_e} z_e^k = u_e, \qquad \forall e \in E \tag{72}$$

$$\sum_{e\in E}\sum_{f\in F}\sum_{k\in K_{ef}} z_{ef}^k \delta_{a,k}^{ef} + \sum_{e\in E}\sum_{k\in K_e} z_e^k \delta_{a,k}^e = y_a, \qquad \forall a \in A \tag{73}$$

$$z_{ef}^k \in \{0,1\}, \qquad \forall k \in K_{ef}, \forall e \in E, \forall f \in F \tag{74}$$

$$z_e^k \in \{0,1\}, \qquad \forall k \in K_e, \forall e \in E \tag{75}$$

$$u_{ef} \in \{0,1\}, \qquad \forall e \in E, \forall f \in F \tag{76}$$

$$u_e \in \{0,1\}, \qquad \forall e \in E \tag{77}$$

$$y_a \geq 0. \qquad \forall a \in A \tag{78}$$

This model reflects the strategic game between the employers and employees, the optimal solution corresponds to a Nash Equilibrium of the game where first the leader decides whether a facility will be opened and the follower reacts by deciding on his route to a remote or main office (it is likely multiple equilibria exist). Hence, no party can unilaterally change their decisions and obtain a better solution. The upper level model (65) - (67) optimizes the system utility while the lower level model (68) - (78) minimizes the individual travel time. The prove that objective (68) minimizes the individual travel time of every employee can be found in Patriksson (2015). Note that the utility from employee $e \in E$ to office $f \in F$ depends on the solution of the lower model, hence this objective function is non-linear, and to solve the problem we find linear approximations later in this section. Constraint sets (71) and (72) ensure that employee $e \in E$ travels on exactly one path $k \in K_{ef}$ when he/she travels to facility $f \in F$ or to the main office, respectively. Constraints (73) identify the flow on every edge $a \in A$ by summing over all routes. All unmentioned constraints and objectives have the same interpretation as before.

A bi-level problem is known to be a complex problem, hence the problem is not expected to be solved within reasonable time if the instance size increases and we introduce a heuristic approach in Section 5.4.2.

5.4.2 Heuristic approach

This heuristic approach, first introduced by Zaferanieh et al. (2023), is implemented to obtain high quality solutions. Objective (65) of model (65) - (78) is non-linear, hence we approximate it by linear a regression model. Firstly, we give an overview of the heuristic followed by a detailed explanation.

Algorithm

An overview of the heuristic algorithm is given in Figure 9.

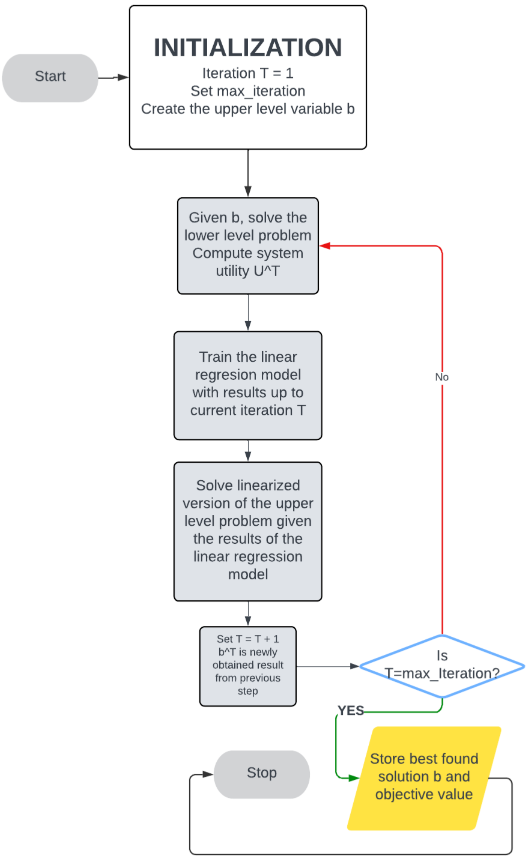

Start
INITIALIZATION
Iteration T = 1
Set max_iteration
Create the upper level variable b
Given b, solve the lower level problem Compute system utility U^T
Train the linear regresion model with results up to current iteration T
Solve linearized version of the upper level problem given the results of the linear regression model
Set T = T + 1 b^T is newly obtained result from previous step
Is T=max_Iteration?
No
YES
Store best found solution b and objective value
Stop

Figure 9: Flowchart SMLH

Initialization: The iteration counter T is set to 1 and the maximum iterations T_{max} of the heuristic is determined. The upper level variable $\bar{b}^T$ of iteration T, representing if a facility is open ('1') or closed ('0'), is randomly generated.

Step 1: Given $\bar{b}^T$, we solve the lower level problem by piecewise linear approximation of the objective. An example of a linear approximation of an edge a can be found in Appendix 8.2. Model (68) - (75) determines the value for variables $u_{ef}, \forall e \in E, \forall f \in F$ and $y_a, \forall a \in A$. Compute the system utility (65) U^T for iteration T.

Step 2: Given U^T and $\bar{b}^T$, train the following linear regression and determine the values for r_0 and $r_f, \forall f \in F$:

$$\bar{U_b} = \sum_{f \in F} r_f b_f + r_0 \tag{79}$$

Step 3: Given the current values of r_0 and $r_f, \forall f \in F$, solve model (80) - (82) to obtain a (possibly better) solution b^{T+1}. Set T = T + 1.

Step 4: If T = T_{max}, stop the heuristic and return the best found solution. Else, return to Step 1.

The coefficients of the trained linear regression are used to find a distinct set of open facilities in model (80) - (82). The set F_1^T contains all open facilities ($b_f^T = 1$) in the solution of iteration T, while F_0^T contains all not open facilities ($b_f^T = 0$) in the solution of iteration T. Here ϵ is defined as the maximum number of open remote offices, as before.

$$
\begin{aligned}
b^{T+1} = &\text{argmax}_b \quad \sum_{f \in F} r_f b_f && && (80)\\
&\text{s.t.} \quad \sum_{f \in F_1^t} b_f - \sum_{f \in F_0^t} b_f \leq \epsilon - 1, && t = 1, ..., T && (81)\\
&\qquad\quad b_f \in \{0, 1\}, && \forall f \in F. && (82)
\end{aligned}
$$

This model maximizes an approximation of the system utility by a linear regression and the constraints (81) ensure that the obtained solutions differs from the solution in past iterations. For the multi objective approach, constraint (3) is added to the upper level model.

Clustering and combining heuristic

Due to their complexity, the single level problem and lower level problem of the bi-level problem are not expected to be solvable in reasonable time. Hence, the problem is clustered into regions (districts) to decrease the number of variables in a single model. These districts are solved independently and are combined following a combining heuristic. Both steps are explained next.

To create balanced districts, the general k-means clustering is employed with a given threshold Γ, the maximum number of employees in a cluster. One issue arises with the threshold, k-means cannot be implemented such that the clusters do not have more than Γ employees. The input is the number of clusters but the size of the clusters is not restricted. Hence, a hierarchical approach is implemented. Initially, the dataset is clustered into $\lceil \frac{\text{Number of employees}}{\Gamma} \rceil$ districts. Note that the data clustered contains the employees, remote offices, and intersections. However, it is likely that an arbitrary cluster c has more employees than the threshold Γ. This cluster is re-clustered following the same procedure until all clusters have less than or equal to Γ employees in the cluster. All final clusters without any employee are discarded as they serve no purpose. For every cluster, the c-UFL is solved. Nevertheless, these clusters are not independent by definition. Some employees have to travel to a main office outside their district thus influencing the solution of other districts. To ensure independence, employees travelling to main offices outside their district are assumed to have no effect on the solution of other districts and travel at constant speed. These independent clusters are combined with a combining heuristic, a flowchart of the algorithm is displayed in Figure 10.

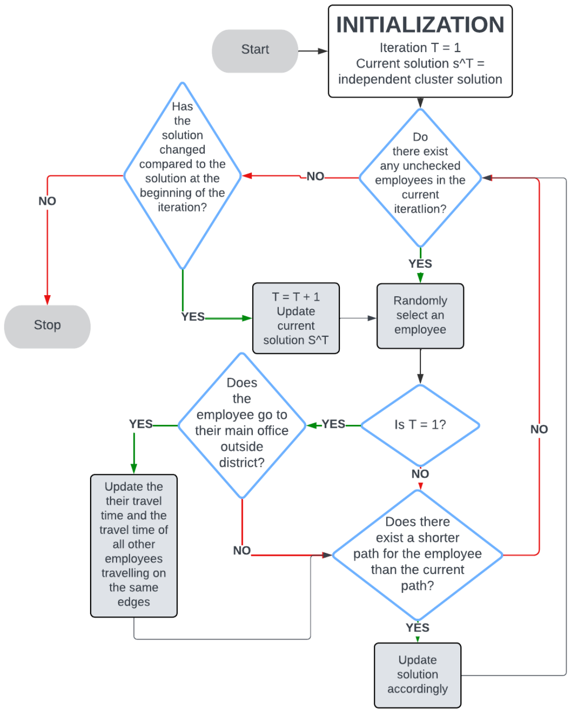

Start
INITIALIZATION
Iteration T = 1
Current solution s^T = independent cluster solution
Do there exist any unchecked employees in the current iteratlion?
NO
Has the solution changed compared to the solution at the beginning of the iteration?
NO
Stop
YES
T = T + 1
Update current solution S^T
YES
Randomly select an employee
Is T = 1?
YES
Does the employee go to their main office outside district?
YES
Update the their travel time and the travel time of all other employees travelling on the same edges
NO
NO
Does there exist a shorter path for the employee than the current path?
NO
YES
Update solution accordingly

Figure 10: Combining heuristic districts

We implement the heuristic to combine the solutions of all districts and relax the independence assumption. Given the current solution of the districts, we iterate over all employees and check if there exists a shorter path to any facility (possibly not in the sets of k shortest paths). If so, we update the path and solution. For every employee travelling to a main office, we update the current travelling time to take other traffic of the edges on the path into account and check if there exists a different path with lower travel time. We repeat this procedure until no more improvements can be found.

6 Results

The results of the methods introduced in Section 5 are obtained by using Python on a Dell laptop with 16GB RAM and all linear programs are solved by Gurobi. This section is divided into three sections: UFLP, CFLP and c-UFLP. Initially, we discuss all problems with solely one objective and extend the problem to consider the multi objective approach. In all figures, we attain the same color pattern: employees, main offices, remote offices, and intersections are coloured red, green, blue, and yellow, respectively. An instance is defined larger if the number of possible office locations is greater.

6.1 Uncapacitated Facility Location Problem

In this section, we discuss the results of the methods explained in Section 5.2. All methods are compared in efficiency for the single and multi objective approach. For the MCBD, we consider ten solutions per

BD iteration ($L = 10$).

6.1.1 Single Objective Approach

We display the computation times for the different approaches in Table 5. BD corresponds to the regular Benders Decomposition with Kelley's scheme in which one cut is added per iteration. We explain the MCBD in Section 5.2.1, while we expound on the UKBD in Section 5.2.2.

For all methods, the computation time increases with the number of facilities $|F|$. The classical BD outperforms the MCBD. This is due to the low number of iterations needed to converge. The BD only implements one additional cut per iteration and convergence within 3 iterations for all instances. In consequence, adding multiple cuts per iteration is "overkill". The multiple cuts lead to a more sophisticated subproblem, hence higher computation time in the SP. It can be concluded that the MCBD might only outperform the BD in problems that are considered difficult. This is equivalent to the conclusion of Beheshti Asl & MirHassani (2019). The UKBD improves the computation time significantly compared to the others BDs. Solving the SP as a Knapsack problem leads to a dramatic decrease in computation time enabling us to solve all instances to optimality.

Table 5: Computation times UFLP methods

Instance		**BD**		**MCBD**		**UKBD**	
	Facilities	**MP-time**	**SP-time**	**MP-time**	**SP-time**	**MP-time**	**SP-time**
20km	16	0.134	2.601	1.000	22.736	1.809	0.826
10km	56	0.457	9.519	2.820	71.496	1.373	0.734
4km	256	2.942	61.234	18.450	468.301	1.583	1.359
2km	992	10.221	201.951			1.574	3.570
1.2km	2,703	25.441	500.052	-	-	1.739	9.588
0.6km	10,608	-	-	-	-	6.627	43.090

$^{-}$ No solution within 30 minutes

MP-time: Computation time in seconds of the MP over all iterations

SP-time: Computation time in seconds of the SP over all iterations

The results in Table 6 show the optimal system utility, number of open facilities and fraction of employees assigned to one of the current open remote offices (coverage). All other employees are assumed to go to their main office resulting in zero utility.

Table 6: Results UFLP obtained by UKBD

Instance	**Total utility**	**Facilities**	**Coverage**
20km	63,032.64	14	0.453
10km	118,806.56	43	0.666
4km	168,765.19	156	0.801
2km	187,360.24	324	0.805
1.2km	195,109.33	433	0.811
0.6km	200,602.97	493	0.827

Coverage: Fraction of employees commuting to remote office

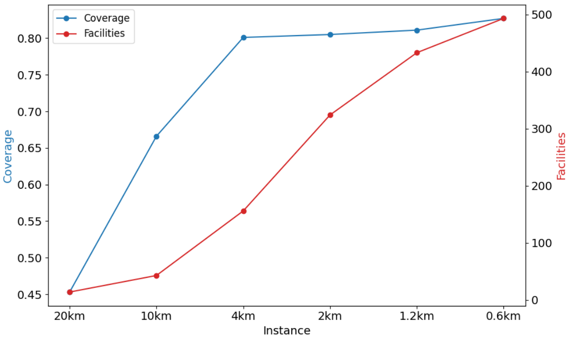
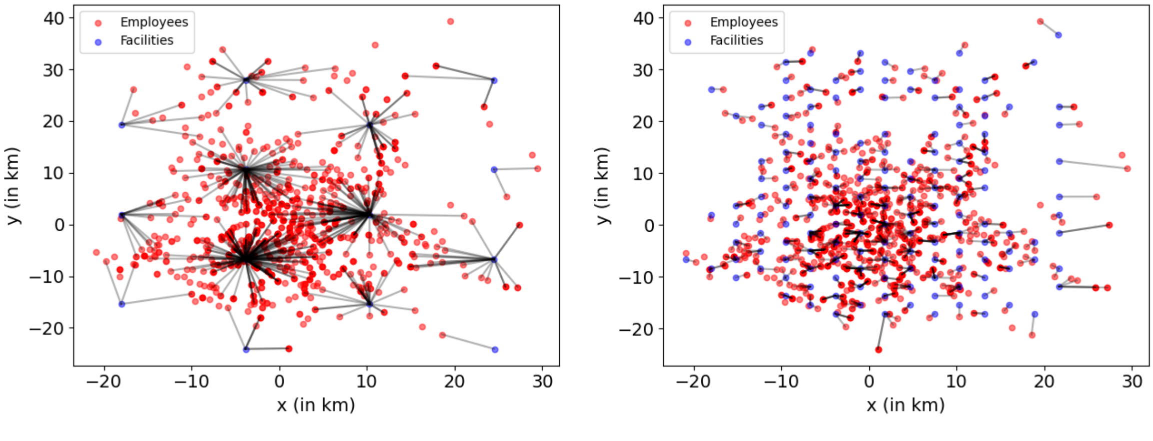


Figure 11: Relation facilities and coverage

It is clear that the utility gain per open remote office is diminishing with the number of open remote offices. For example, to achieve approximately 3.2 times the utility corresponding to the 20km instance, we should open 35 times the number of facilities in the 0.6km instance. The relation between the number of open offices and coverage of employees is displayed in Figure 11. As expected, the coverage increases with the number of open offices. It is noteworthy that the coverage is diminishingly increasing in the number of open facilities. Hence, the effect of an additional remote office on the coverage is greater when fewer offices are opened. To give an interpretation of the solution, Figures 12a and 12b illustrate the allocation of the employees.



(a) Result 20km

(b) Result 4km

Figure 12: Employee allocation, 2 instances

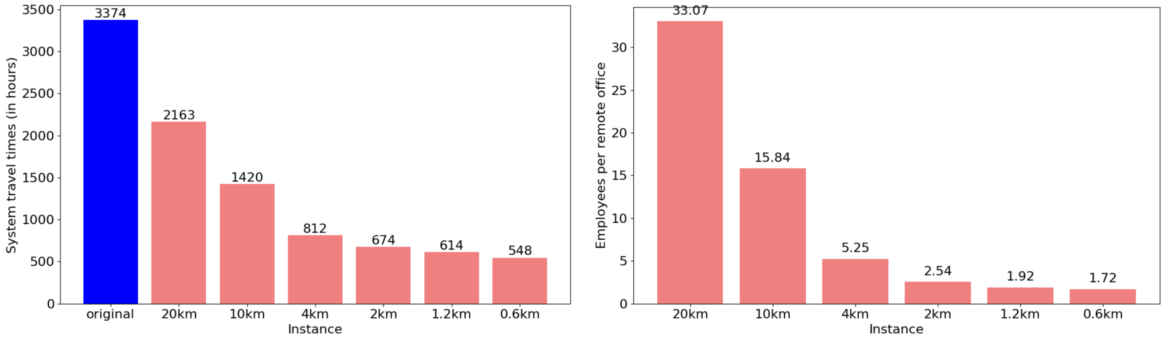
Figures 12a and 12b display the results of solving the 20km and 4km instance respectively. Not only the average number of employees allocated to a remote office increases (see Figure 13b), but also the average commuting time decreases. It can be concluded that the employees would choose the solution of the 4km over the 20km instance.

It is clear that a large share of employees are assigned to a remote office. Although, the model maximizes the utility of employees, identifying the total travel time decrease of all employees compared to the original situation gives valuable insights. This is displayed in Table (7) and Figure 13a. The table displays the characteristics of the original situation, where all employees travel to their main office, and the situation where remote offices are introduced in all instances. The average commuting time decreases if the number of facilities increases, which is self-evident. The standard deviation and median decreases significantly. In all instances, the weight of the commuting time distribution is close to zero while there is a relatively slim tail in the larger commuting times. These long travelling times correspond to employees commuting to their main office.

Table 7: Characteristics travel time UFL original situation and situations with remote offices

Instance	Average	Median	Std. Dev.	Minimum	Maximum
Original	3.298	2.412	6.243	0.001	114.959
20km	2.114	1.617	2.502	0.002	63.032
10km	1.388	0.905	2.377	0.002	63.032
4km	0.794	0.348	2.287	0.002	63.032
2km	0.659	0.169	2.308	0.002	63.032
1.2km	0.600	0.103	2.316	0.002	63.032
0.6km	0.535	0.053	2.308	0.001	63.032

Travel time in hours with a speed of 5 kilometers per hour

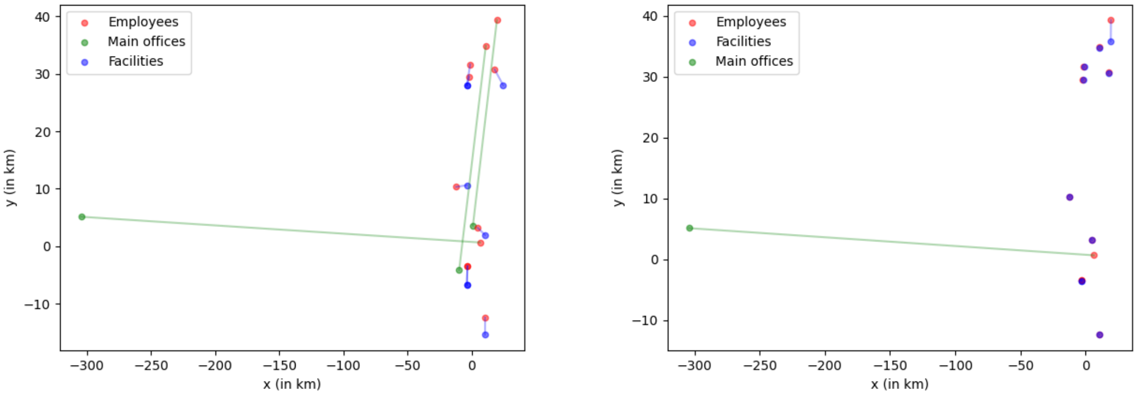


(a) Histogram system travel times (walking)

(b) Average employees per facility

Figure 13: UFL characteristics all instances

Figure 13a illustrates the decrease in system travel times, defined as the summed travel times of all employees, of all instances compared to each other and the original situation. In the latter, the system travel time is $3,374.13$ hours while the system travel time decreases significantly when we introduce the remote offices. In the smallest instance, the rearrangement of employees results in a decrease of 35.9% travel hours compared to the original situation. This decreases to circa 547.61 hours with the highest number of open facilities (more than 83.7% reduction). The employees that used to travel large distances from their household to the main office but have to travel only a few kilometers in the new situation have a large contribution to the drop in system travel time. We display these employees in Figures 14a and 14b, they have to travel more than 50 kilometers to their main office.



(a) Instance 20km

(b) Instance 0.6km

Figure 14: Figures employees > 50 km to main office

It is clear that the system travel time decreases significantly in both instances. However, one of the employees does not use a remote office in both instances as he/she has negative intercept and slope utility. Furthermore, some employees travel to their main office in the smaller instance while they travel to a remote office in the larger one. This is a result from their intercept and slope utility. For example, the employee in the upper right corner has a intercept utility of 344.94 and slope of -148.08 such that he/she is willing to walk at most 2.33 hours to the remote office. The traveltime to the closest facility is 3.25 and 0.74 hours for the smaller and larger instance respectively. In consequence, the employee is assigned to a remote office in the larger but not in the smaller instance.

A downside of the larger instances is the number of employees per shared office. In instance 0.6km, the average number of employees is circa 1.7 persons (see Figure 13b). Note that we use only a small subset of employees in the city such that the current location of remote offices might still be profitable due to large amount of employees. Nevertheless, opening a remote office for only a handful of persons may not be profitable. Consequently, restricting the number of open remote offices potentially results in higher number of employees per remote office and lower cost but decreases the system utility. We examine the multi objective approach in the next sub-section.

6.1.2 Multi Objective Approach

The Pareto front is a valuable measurement to display the trade-off between two or more contrary objectives. In this case, the system utility and number of opened remote offices. Figure 15 visualizes the Pareto front of instances 2km and 0.6km. These Pareto fronts are significantly different. All other

Pareto-fronts can be found in Appendix 8.3.

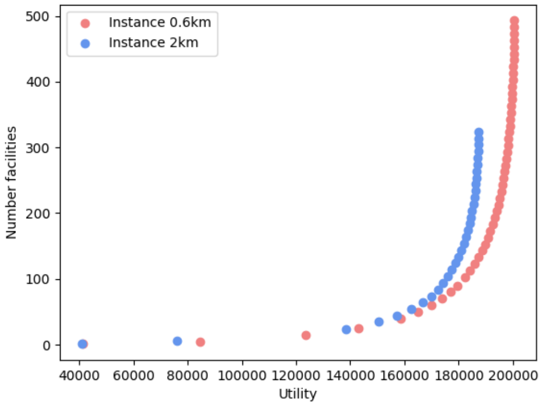
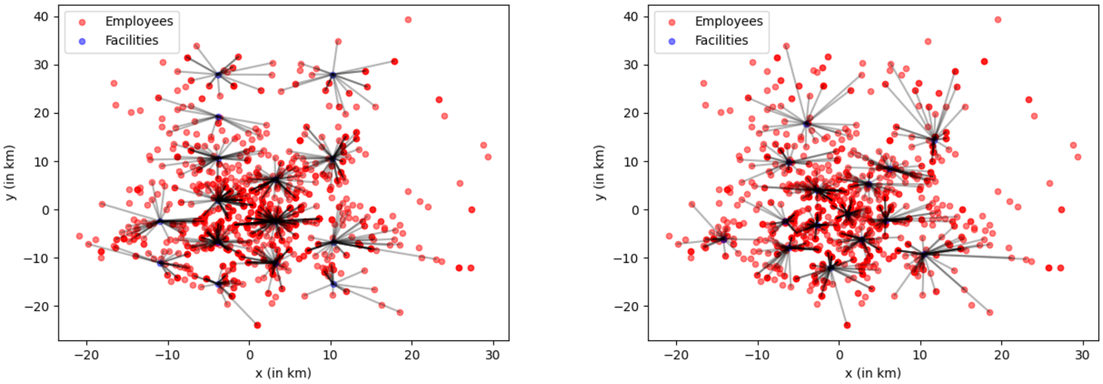


Figure 15: Pareto front 2km and 0.6km

The marginal increase of opening an additional facility on the system utility is diminishing. The Pareto fronts show substantial differences for the instances. For example, if one remote office is open, the system utility is 40, 768.13 and 41, 125.39 for the larger and smaller instance respectively. This is an increase in utility of 357.26 per open facility. In comparison, the utility is 121, 718.48 and 123, 404.61 with fifteen open facilities for the smaller and larger instance respectively. In consequence, the increase in utility is 112.41 per open facility. It can be concluded that while the difference in utility continues to increase with the number of open offices, the rate of increase diminishes as more remote offices are opened. The employee allocations of the solution with fifteen open remote offices for both instances is given in Figures 16a and 16b.



(a) Instance 2km

(b) Instance 0.6km

Figure 16: Employee allocation with 15 open offices

At first, these solutions do not seem to differ much. However, the number of facilities in the city centre of Mexico city is significantly higher for the larger instance compared to the smaller one (10 vs

5). Most employees live in the city centre such that the main advantage of the large instances is that it offers a wider variety of facility locations.

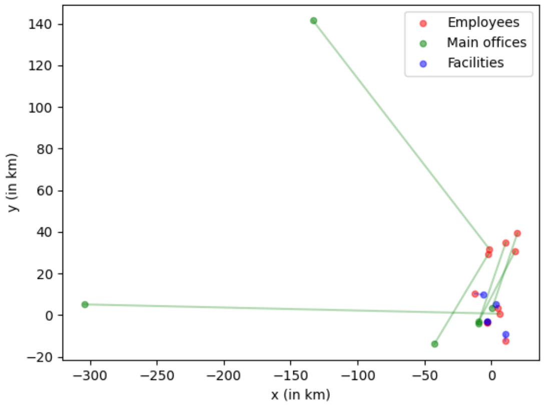
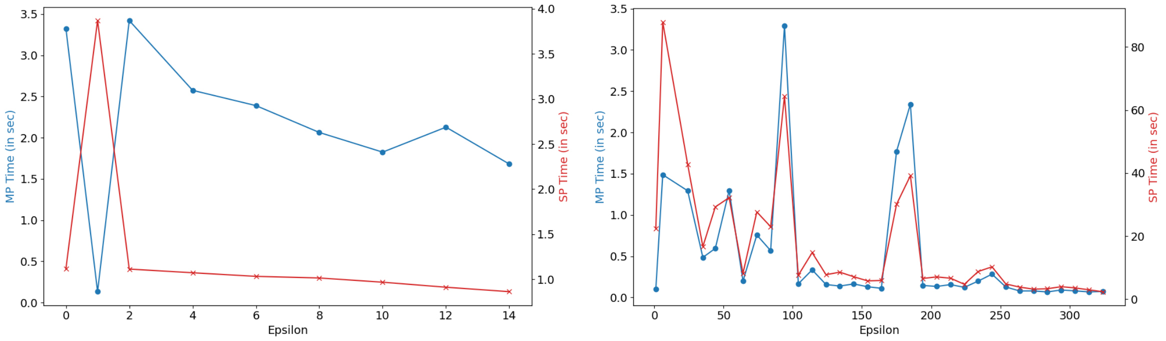


Figure 17: Employee allocation > 50km to main office, 15 open facilities, instance 0.6km

Figure 17 shows the result of the same employees in Figures 14a and 14b but with only 15 open facilities. The effect is rather drastic such that only four employees travel to a remote office while 7 employees travel to their main office. This results in a significant increase in travelled kilometers and possibly more congestion in the road network.



(a) Instance 20km

(b) Instance 2km

Figure 18: Computation time multi objective UFLP

Figures 18a and 18b reflect the computation time of the multi objective problem with a given ϵ. It is an expected observation that the computation time increases with a decreasing ϵ. If the allowed number of open remote offices decreases but the number of remote office locations remains unchanged, the problem becomes more challenging due to the exponential increase in possible solutions in the solution space. However, there exist some spikes in the computation time of the 2km instance. The increased computation time is due to the branching tree. If the optimal solution of the root node is fractional, the branching is rather time consuming. Hence, further research could be to develop a more efficient branching strategy.

6.2 Capacitated Facility Location Problem

In this section, we elaborate on the results of the methods explained in Section 5.3. First, we discuss the results of the model maximizing solely the system utility. Afterwards, we consider the second objective to minimize the number of facilitates and discuss the effect.

6.2.1 Single Objective Approach

Table 8 displays the system utility and number of open facilities for three facility capacities: 20, 40, and 60 employees. The last column represents the results of the uncapacitated variant.

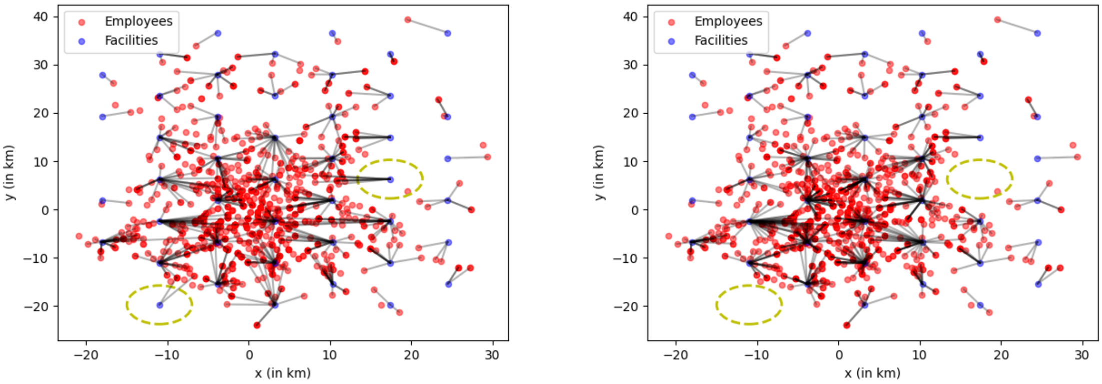
Table 8: Results CKBD /UKBD

Instance\Capacity	**20**		**40**		**60**		∞	
	Utility	**Facilities**	**Utility**	**Facilities**	**Utility**	**Facilities**	**Utility**	**Facilities**
20km	40,291.69	14	50,209.08	14	55,925.10	14	63,032.64	14
10km	101,689.90	45	113,885.43	43	117,681.06	43	118,806.56	43
4km	168525.27	156	168,765.19	156	168,765.19	156	168,765.19	156
2km	187,360.24	324	187,360.24	324	187,360.24	324	187,360.24	324
1.2km	-	-	-	-	-	-	195,109.33	433
0.6km	-	-	-	-	-	-	200,602.97	493

- No solution within 30 minutes

∞ Uncapacitated variant

The results shown in the table demonstrate an anticipated pattern. The larger the instance, the smaller the effect on the objective values. Instances 20km and 10km are the only one that affects all capacity sizes. Figure 13b already showed that the average number per facility is the largest in these instances. Moreover, an interesting result is the difference between the solutions of the 10km instance with a capacity of 20 and 40/60. These solutions are displayed in Figures 19a and 19b.



(a) Capacity 20

(b) Capacity 40/60

Figure 19: Employee allocation CFL, 10km

An important difference between the solutions is circled. Due to the lower capacity limit in the first figure, additional facilities are opened and the employees are more equally distributed over the open remote offices. For example, the office in the centre of the map has 20 employees assigned in the first solution while it has 37 employees assigned in the 40/60 capacity solution. In the both instances, there are a few "popular" remote offices which are (nearly) fully filled while others are about empty. The characteristics of average employees per office are shown in Table 9 for instances 20km and 10km. All other instances do not have any value to assess the capacity constraint.

Table 9: Characteristics of employees per open remote office

		Capacity			
		20	**40**	**60**	∞
20km	**Average**	14.71	21.21	24.64	33.07
	Median	20.0	21.5	21.5	16.5
	Std. Dev.	7.08	15.65	21.98	46.86
	Minimum	1.0	1.0	1.0	1.0
	Maximum	20.0	40.0	60.0	166.0
10km	**Average**	10.51	14.09	15.14	15.84
	Median	8.0	7.0	7.0	7.0
	Std. Dev.	7.90	14.09	18.01	21.01
	Minimum	1.0	1.0	1.0	1.0
	Maximum	20.0	40.0	60.0	97.0

$^{\infty}$ Uncapacitated variant

With the uncapacitated variant, the standard deviation of number of employees per office is relatively large in both instances, due to difference in popularity of remote offices. This metric decreases more severely if the capacity is restricted as more remote offices will facilitate the maximum number of employees per remote office. Another interesting observation is that the median for the smallest instance with a capacity of 20 employees per remote office equalizes the capacity. Hence, only a few offices facilitate less than 20 employees while the rest is fully capacitated. For the 10km instance, the median remains approximately unchanged but the standard deviation decreases with a stricter capacity. The reason being that especially offices with high number of employees assigned in the uncapacitated case are heavily impacted. The number of employees assigned to these offices reduces while the number of employees to other offices increases slightly. In conclusion, the capacity has a significant impact on the average employees per remote office and balances the the number of employees assigned per remote office in smaller instances.

The computation times of the exact method are shown in Table 10. Whereas the four smallest instances can be solved exactly within 30 minutes, the two largest can not. This is remarkable as the maximum number of employees assigned to a single office in the UFLP is 13 for both instances. Hence, the optimal solution of the UFLP is equivalent to the optimal solution of the CFLP. The difference between both implementations is the solving mechanism to obtain the optimality cuts. Due to the capacity constraint, the optimal dual variable cannot be the result of the Dantzig algorithm such that a transportation problem is solved. This problem takes significant time in all instances (more than 90%). To reduce the computational burden, we could implement an approximation technique to decrease the computation time at the cost of possibly less strong optimality cuts.

Table 10: Computation times CKBD /UKBD

Instance\Capacity	**20**		**40**		**60**		**∞**	
	MP-time	**SP-time**	**MP-time**	**SP-time**	**MP-time**	**SP-time**	**MP-time**	**SP-time**
20km	0.004	0.201	0.003	0.290	0.003	0.337	1.809	0.826
10km	0.628	68.873	0.535	19.776	0.515	15.705	1.373	0.734
4km	0.581	86.702	0.592	76.314	0.518	88.130	1.583	1.359
2km	0.709	662.377	0.748	691.803	0.736	687.573	1.574	3.570
1.2km	-	-	-	-	-	-	1.739	9.588
0.6km	-	-	-	-	-	-	6.627	43.090

MP-time: Computation time in seconds of the MP over all iterations

SP-time: Computation time in seconds of the SP over all iterations

$^-$ No solution within 30 minutes

$^\infty$ Uncapacitated variant

Lastly, the performance of the k-MGAP is evaluated against the optimal solution. This method is the assignment method in the APMQEA, hence the main focus of this algorithm is to find relatively good solution in short time manners. To test the quality of the assignment heuristic, we assign employees to the optimal set of open facilities obtained by the CKBD, or UKBD if the CKBD does not find an optimal solution within 30 minutes, and compare it to the optimal solution. The results of the k-MGAP are displayed in Table 11.

Table 11: Objective CKBD vs k-MGAP

Instance\Capacity	**20**			**40**			**60**		
	CKBD	**k-MGAP**	**Gap (%)**	**CKBD**	**k-MGAP**	**Gap (%)**	**CKBD**	**k-MGAP**	**Gap (%)**
20km	40,291.69	39,531.18	1.89	50,209.08	49,494.15	1.42	55,925.10	55,228.12	1.25
10km	101,689.90	98,699.28	2.94	118,806.56	114,164.35	3.90	118,806.56	116,997.57	1.52
4km	168,765.19	167,865.89	0.53	168,765.19	168,492.75	0.16	168,765.19	168,593.57	0.10
2km	187,360.24	185,136.99	1.19	187,360.24	185,302.58	1.10	187,360.24	186,018.76	0.72
1.2km	195,109.33	191,214.41	1.99	195,109.33	191,499.47	1.85	195,109.33	191,621.87	1.79
0.6km	200,602.97	196,527.82	2.03	200,602.97	196,044.73	2.27	200,602.97	196,524.61	2.03

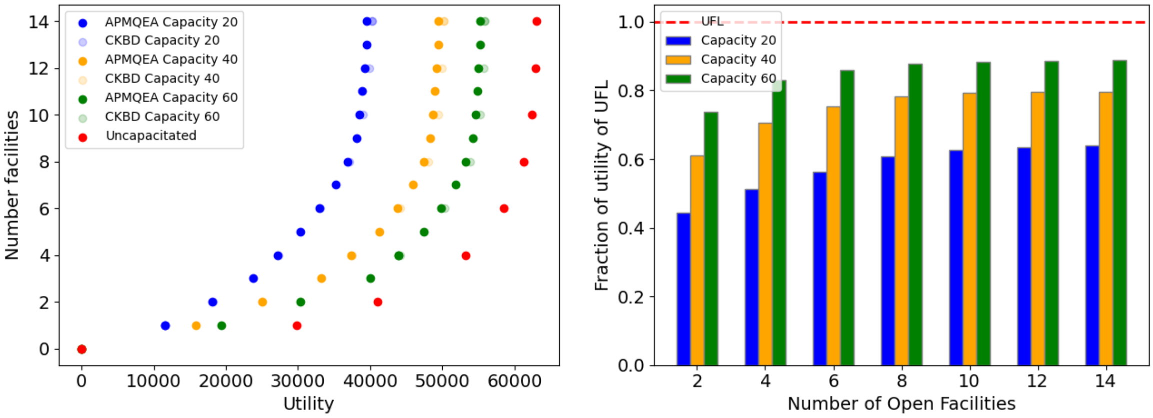
The results show that the heuristic finds close-to-optimal solutions, often within one or two percent of the optimal solution. It assigns a subset of the best employees to a facility that is overcapacitated and it does not take other facilities into account. To reduce the effect of this simplification, we implement a GRASP procedure by incrementing the multiplier parameter. Consequently, the increase in multiplier solely effects the solution of smaller instances and the capacity for the larger instances does not play a roll. Additionally, the computation times are given in Table 12. These results are promising, the solutions are found within 0.5 seconds for every instance but the largest. It can be concluded that this heuristic finds good solutions in a quick manner.

Table 12: Computation time (in sec) k-MGAP

Instance\Capacity	**20**	**40**	**60**
20km	0.084	0.067	0.063
10km	0.132	0.085	0.087
4km	0.233	0.090	0.091
2km	0.192	0.186	0.193
1.2km	0.472	0.435	0.422
0.6km	2.247	2.091	2.249

6.2.2 Multi Objective Approach

The results in the previous subsections show that the k-MGAP finds good solutions in a short time manner. Hence, we can use the APMQEA to find approximated Pareto fronts. The value for parameter $\Delta\theta = 0.01 * \pi$, this setting seemed to perform the best. The larger the value, the more likely it is that it converges to a local optimum, while a smaller value could not converge at all. For the smallest instance, we compare this approximated to the exact Pareto fronts obtained by the CKBD for all capacities in Figure 20a. It shows the effect of a capacity constraint on the solution.



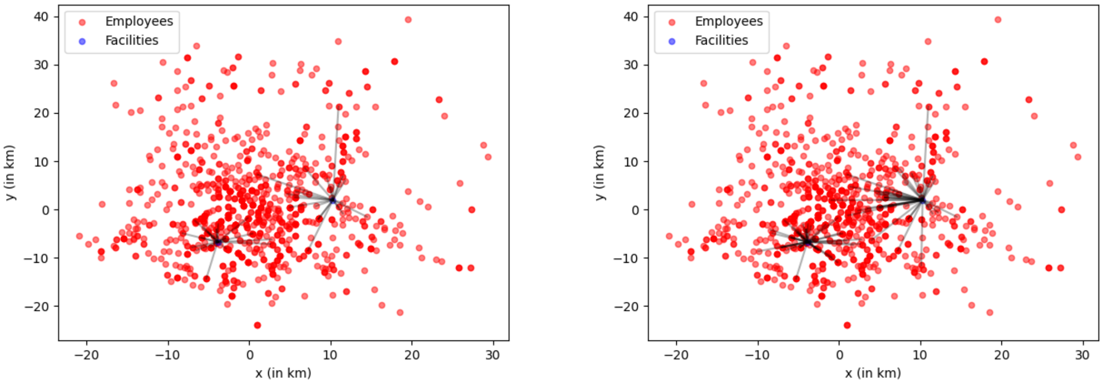
(a) Pareto fronts per capacity

(b) Utility capacitated vs uncapacitated

Figure 20: Instance 20km, CFLP

Firstly, the Pareto fronts show once more that the marginal increase in utility is diminishing with the number of open remote offices. The effect of the capacity is clearly visible. The system utility increases with the capacity for all number of open remote offices. The effect of an increase in capacity from 20 to 40 is larger than from 40 to 60 employees. This is nicely displayed in Figure 20b, which visualizes the effect of the capacity on utility compared to the uncapacitated solution. Hence, the marginal effect of additional capacity is also diminishingly increasing in the utility. Secondly, the approximation of the Pareto front by the APMQEA is displayed in Figure 20a as well. Although it can be concluded that the heuristic finds high quality solutions, the optimality gap is increasing with the number of open remote offices.

Furthermore, the exact optimal solutions with 2 open facilities are displayed in Figures (21a) and (21b). These solutions show that the open facilities are identical. It is just the number of employees assigned that differs which results in the difference of system utility.

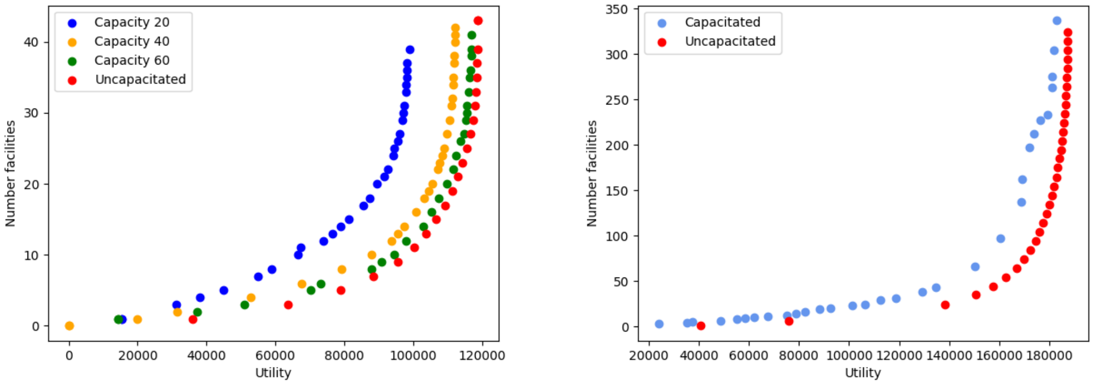


(a) 20 capacity

(b) 40 capacity

Figure 21: Solution 2 remote offices CUFL, instance 20km

However, the CKBD does not find the optimal Pareto front for all other instances. Hence the APMQEA is implemented to obtain an approximated Pareto front. Figure 22a shows the APMQEA Pareto front of instance 10km with different capacities, compared to the optimal Pareto front of the uncapacitated version. Once more, the effect of additional capacity is diminishingly increasing in the utility. Figure 22b displays the quality of the Pareto front in one of the largest instances. It is clear that it finds a good optimal front. The difference in utility increases with the number of open facilities and this is due to the evaluation heuristic (as explained before). The difference in utility is largest between 150 and 250 open facilities, this has likely to do with the maximum number of evaluations (1000) for the problem. Increasing the number of evaluations will result in even better fronts. Hence, the APMQEA finds high-quality Pareto fronts.



(a) Instance 10km, all capacities

(b) Instance 1.2km, capacity 20

Figure 22: Pareto Fronts, CFLP

6.3 Facility Location problem with congestion

In this section, we discuss the effect of incorporating the traffic situation of Mexico City into the decision process. We start off by elaborating on some distinct components of the model and afterwards we introduce and examine the results of the entire model.

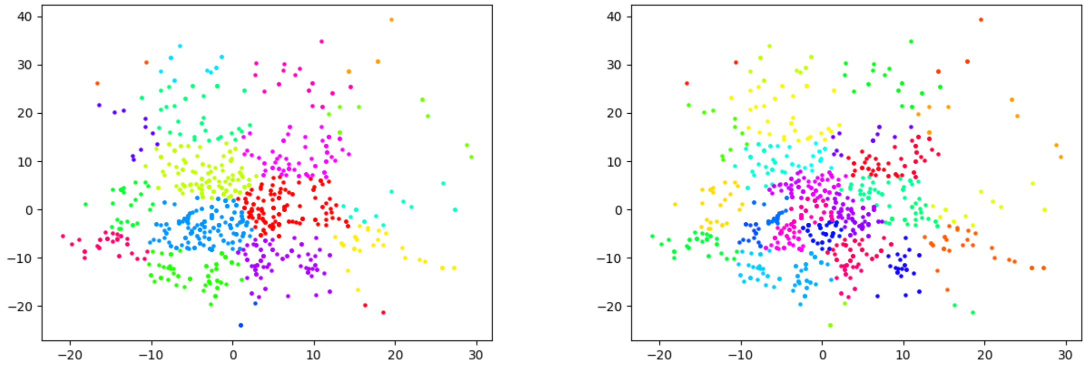
6.3.1 Clustering approach

Not only the entire bi-level problem has been proven to be difficult to solve, but also the lower-level problem. Hence, the city is clustered into regions following the procedure in Section 5.4. The k-means clusters every point in the data: main offices, remote offices, employees, and intersections. In consequence, it is possible that a cluster has no employees, these clusters are discarded as they serve no purpose, and that others have a lot of employees. The clustering is shown in Figure 23a while balancing out the clusters results in the clustering shown in Figure 23b. We show the characteristics of number of employees per cluster in Table 13. Before balancing, the difference between two clusters could be large (0 to 258), but this reduces after balancing to only 56 employees. Furthermore, the average number of employees is roughly in the middle of the tightened range, and the standard deviation has decreased significantly. The number of clusters increases from 20 to 36.

Table 13: Characteristics clustering approach, instance 0.6km

Number of employees in cluster	**Before balancing**	**After balancing**
Average	51.150	28.417
Std. Dev.	65.462	14.548
Minimum	0	2
Maximum	258	58

The goal of these figures is to show the effect of balancing the clusters. Most clusters in the outskirts of Mexico City remain intact. The groups in the city centre are divided into smaller ones. Due to the assumption that employees commuting to their main office have no effect on the traffic outside their cluster, the c-UFL can be solved independently for every cluster.



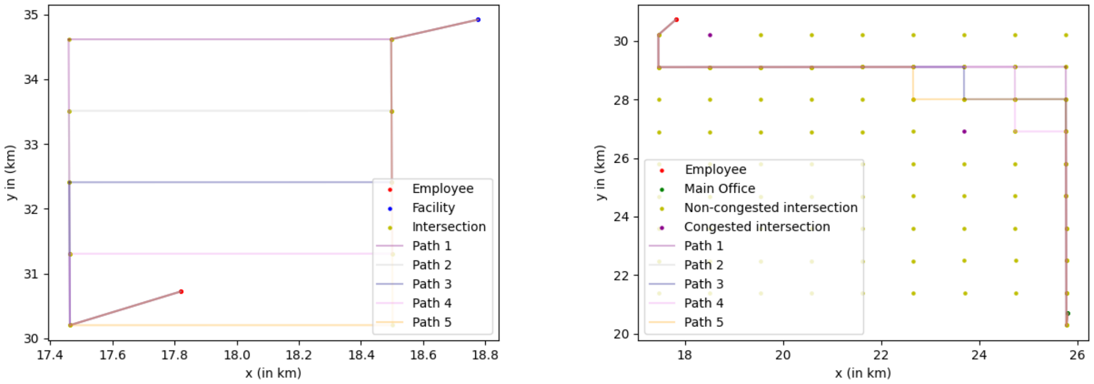
(a) Non-hierarchical clustering

(b) Hierarchical clustering

Figure 23: Clustering employees, instance 0.6km

6.3.2 K-shortest paths

Another component of the bi-level approach is the determination of the k shortest paths of a employee-facility pair, explained in Appendix 8.2. We set the number of paths k to five. The effect of higher number of paths has shown to increase with lower number of open remote offices per cluster, although not significantly. The value for the congestion parameter π_a on each edge $a \in A$ is 0.15 and follows from Council (2000), moreover the capacity c_a on each edge $a \in A$ is assumed to be 10. Figure 24a shows the five shortest paths from an employees household to a nearby remote office. This figure nicely displays the different paths. The first path is the quickest while the fifth is the slowest of the five. Figure 24b exhibits the paths from the same employee to their main office. Firstly, the distance is much larger such that more routes are possible. The purple points in the figure are the congested intersections and should be avoided. It is clear that all paths avoid these intersections and that all paths are relatively the same. A higher number of shortest paths would result in more distinct paths. Room for further improvement is to implement a restriction that all paths should differ in at least n intersections from each other, with n an integer.



(a) Shortest paths to remote office

(b) Shortest paths to main office

Figure 24: 5 shortest paths from employee to office, instance 2km

A this point we have the shortest paths for the employees to every remote office in a cluster and their main office. We can use this information to solve the model with traffic congestion.

6.3.3 Single Objective Approach

First of all, we consider the situation without a limit on the number of open facilities. This section focuses on the three largest instances: 2km, 1.2km and 0.6km. Compared to the UFL, the main adaption is the variable speed of the employees. This effect is shown in Table 14 and Figure 25. The table states the system utility, distance, travel time, and coverage of instances 2km and 1.2km. The clusters of 0.6km are not solvable within a few minutes due to the high number of possible remote office locations. A heuristic method could be introduced to heuristically solve the lower level problem of the bi-level approach. To quantify the effect of traffic, the results are compared with case of constant Manhattan travelling distance with a speed of 30 kilometers per hour.

Table 14: Single objective
Comparison UFL vs c-UFL

	2km			1.2km		
Characteristics	**UKBD***	**c-UFL** **Pre-combining**	**c-UFL** **Post-combining**	**UKBD***	**c-UFL** **Pre-combining**	**c-UFL** **Post-combining**
Utility	205,063.20	191,989.74	194,579.11	206,386.84	196,581.42	196,581.42
Travel Distance (in km)	3,378.74	3,687.11	3,712.48	3,059.07	3,211.91	3,241.04
Travel Time (in hours)	112.63	123.91	132.20	101.97	107.37	116.22
Coverage	0.805	0.789	0.798	0.811	0.802	0.802

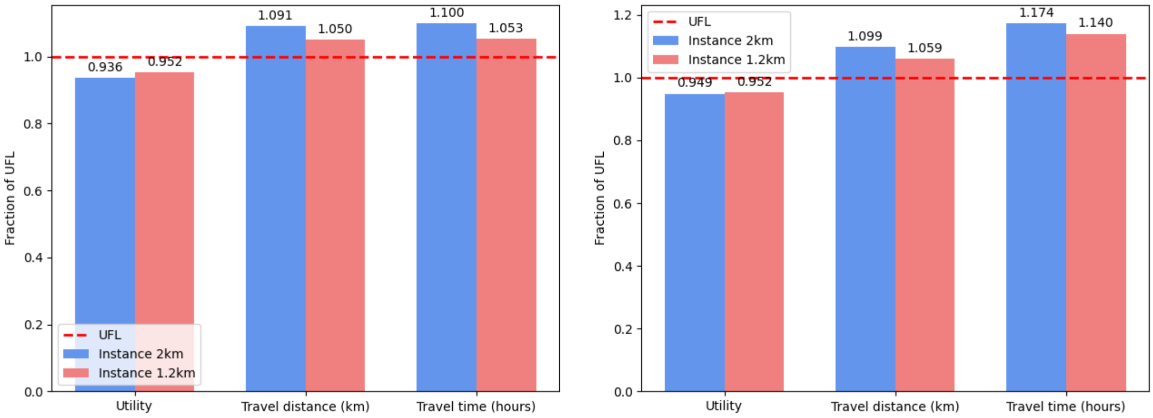
* Employees have assumed constant speed of 30 km/hour
Pre-combining: before combining clusters
Post-combining: after combining clusters

We show the characteristics of the computation time of solving the clusters independently, without combining them, in Table 15. The average computation time increases per cluster with the larger instances due to the size of the lower-level problem. Furthermore, the standard deviation is much larger for the larger instance which has to do with the larger differences in cluster sizes. The solutions of the clusters are later combined, this takes 436.52 and 446.66 seconds for instance 2km and 1.2km, respectively.

Table 15: Computation time (in sec) per cluster c-UFL heuristic

Characteristics	**2km**	**1.2km**
Average	0.66	2.82
Median	0.29	0.91
Std. Dev.	0.95	5.81
Minimum	0.03	0.06
Maximum	4.85	30.18

If there is no restriction on the number of open facilities. Every employee willing to work at a remote office goes to the the nearest possible location, resulting in low saturation of offices. Due to the short distance travelled, the employees do not really have to worry about the traffic situation and the deviations are minor. This is displayed in the table and figure. The average speed of the employees in the UFL is, logically, 30 km per hour. In the c-UFL (pre-combining), this reduces to 29.76 and 29.91 km/h for instance 2km and 1.2km respectively while the average speed after combining all clusters decreases to 28.08% and 27.89% for the smaller and larger instance, respectively. At first, the clusters are solved independently such that employees on the edge of a cluster could have a lower travelling time to an office in a neighbouring cluster than to a facility in the cluster. This problem is solved when the combining heuristic is applied resulting in an increase in the utility. Another interesting observation is that the system travel time increases following the combining heuristic. This is elaborated on in a later stage. Figure 25a reflects the percentage change of the characteristics with the c-UFL pre-combining compared to the UFL solution. It is clear that the travelling distance and time increase with a similar percentage, hence the average speed of the employees is not significantly impacted. However, the traffic situation (highly congested intersections) could result in a detour and thus possibly in an increase in travelling time and distance. Figures 26a and 26b reflect this.



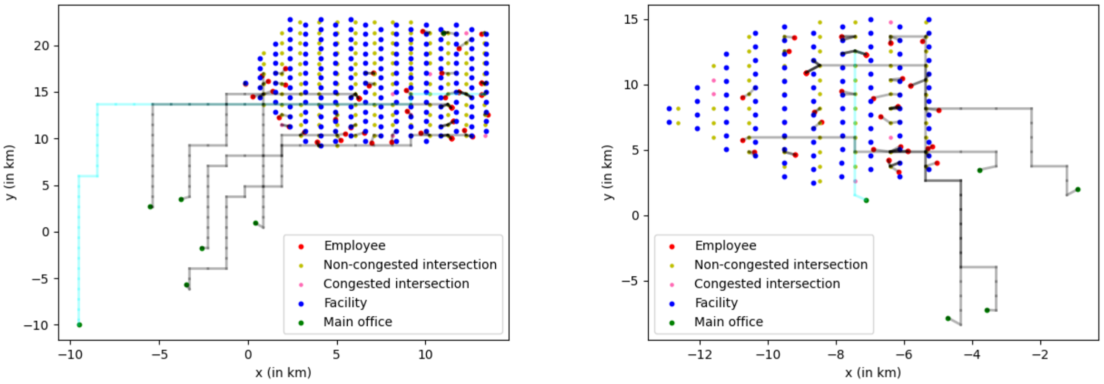
(a) Pre-combining

(b) Post-combining

Figure 25: Single objective

Characteristics compared to UFL

Figure 26a shows the result of solving the bi-level problem of one of the clusters. This cluster reflects the effect of the congested intersections on the routing decisions of employees. The cyan colored path is the commuting path of an employee negative towards working at remote offices (negative intercept utility), hence it commutes to his/her main office. The pink intersections are congested and thus result in higher travelling times. The employee wisely avoids this point. In this case, the alternative route is not likely to result in additional travel time. Nevertheless, in more crowded areas the detour could result in higher distance but lower travelling time. This will be touched upon in the multi objective case. An alternative case is displayed in Figure 26b. This figure shows the solution of the problem in the 15th cluster. This is an interesting cluster due to the highlighted cyan commuting path. As before, this employee is negative towards working at remote offices. However, this path is contradictory to the expectation that every employee avoids congested intersections as the second to last crossed intersection is congested with traffic parameter 2. A detour would lead to a significantly higher travelling time than travelling through the intersection. The intersection is expected to increase the travelling time on the connected roads by circa 30% for that road segment while a detour would result in an increase of approximately 100%. In conclusion, in some situations it is still beneficial to use congested intersections as they are likely to be faster than the detours.

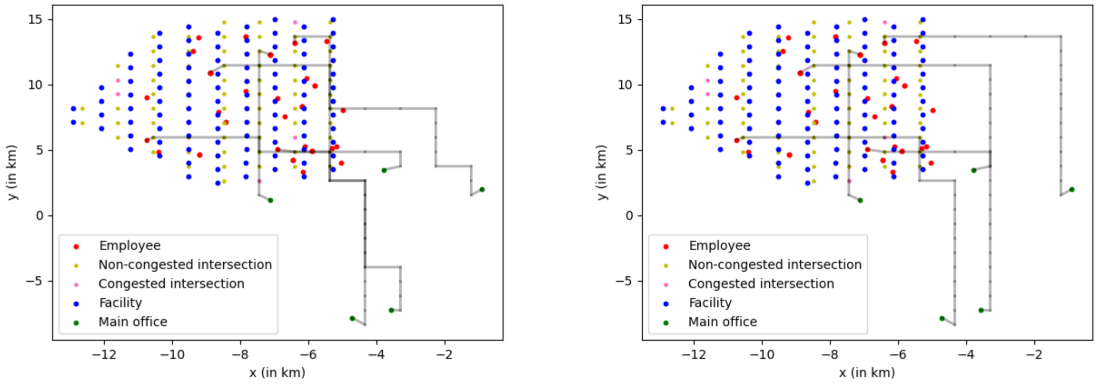


(a) Instance 1.2km, cluster 10

(b) Instance 1.2km, cluster 15

Figure 26: Solutions c-UFL, instance 1.2km

At first, the results of c-UFL post-combining may seem odd. Although the travel time and utility are highly correlated, the utility remains constant or increases while the system travel time increases as well. To make sure that every cluster could be solved independently, it is assumed that the employees travelling outside their cluster do not affect the traffic for others and could travel with 30 km/hour, this assumption is relaxed by the combining heuristic. In consequence, the employees travelling to the main office choose other routes and have a slightly increased travelling time as some of the roads are congested due to the employees in other clusters. An example is given in Figures 27a and 27b.



(a) Pre combining heuristic

(b) Post combining heuristic

Figure 27: Solutions main offices c-UFL, instance 1.2km, cluster 15

The total travel time of this cluster increases from 2.46 to 4.05 hours, while the travel distance remains approximately the same. Hence, relaxing the independence assumption results in significantly different routes and increases the travel time of the employees to main offices. Another improvement is that the average number of employees on a road segment decreases, resulting in less congestion. Not only the paths for employees going to the main offices changes, but also the paths of employees going to a remote office. It could be that an employee on the edge is reallocated to another remote office outside their district or that there exists a quicker route than the k predefined paths. These changes results in higher utilities.

6.3.4 Multi Objective Approach

In a city like Mexico City, the possibilities for opening remote office locations are usually limited. Building a remote office is a costly endeavour. If a cluster is seen as a district, it is viable solution to build a single facility in every district. We discuss the effect of limiting the number of open facilities in this section, we do not consider the entire Pareto front due to its computational extensiveness. Table 16 presents the system utility, travel distance, travel time, and coverage of the standard UFL, pre-combining c-UFL, and post-combining c-UFL. Both problems assume a free flow travel speed of 30 kilometers an hour.

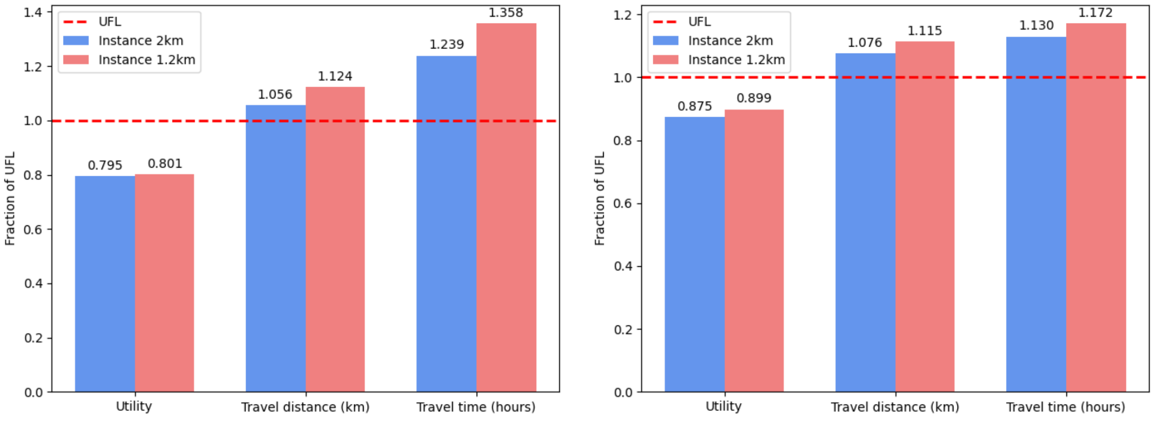
Table 16: Multi objective

Comparison UFL vs c-UFL

	2km			**1.2km**		
Characteristics	**UKBD***	**c-UFL Pre-combining**	**c-UFL Post-combining**	**UKBD***	**c-UFL Pre-combining**	**c-UFL Post-combining**
Utility	195,571.24	155,429.46	171,034.79	195,924.56	156,917.77	176,039.46
Travel Distance (in km)	5,758.49	6,080.24	6,195.95	5,635.55	6,333.24	6,281.96
Travel Time (in hours)	191.95	237.80	216.81	187.85	255.20	220.22
Coverage	0.762	0.674	0.732	0.765	0.661	0.745

* Employees have assumed constant speed of 30 km/hour

Before combining the clusters, incorporating traffic information into the decision process results in a drop of 20.53% and 19.91% in utility for the smaller and larger instance respectively. The main reason of this increase are the different facility locations and the increase in travel distance due to the congestion. The more people have to go one place, the more congested it is. This is reflected in the average speed of the employees, which reduces to 25.55 and 24.82 km/hour. However, it is assumed that employees are not allowed to use remote offices of other districts. The combining heuristic relaxes this constraint, resulting in a more diversified planning of the routes. After combining, the utility decreases with only 12.55% and 10.15% for the smaller and larger instance in comparison to the UKBD respectively. Employees can visit remote offices in other clusters resulting in a decrease in system travel time. This is reflected in Figure 28. We discuss several solutions of clusters to show the effect of the congestion and combining heuristic.



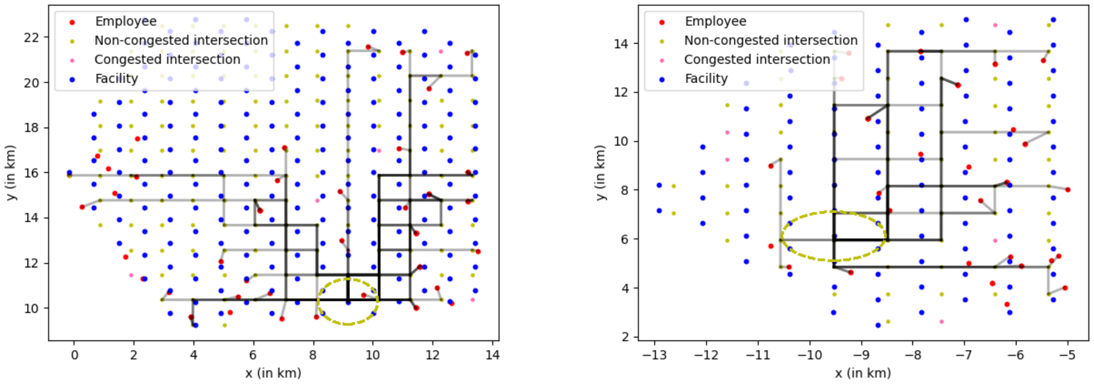
(a) Pre-combining

(b) Post-combining

Figure 28: Multi objective

Characteristics compared to UFL

Figures 29a and 29b show the solution of clusters 10 and 15 with at most one facility per cluster. The black lines represent the routes of an employee, the thickness of a line increases with the number of employees travelling an edge/road. The green circles clarify the position of the open facility. The open remote office maximizes the gained utility for the employers. The pink dots represent congested intersections. It is clear that employees try to avoid these as much as possible such that no employee travels via the congested intersections. However, the total travel time of all employees in this cluster is 31.18 hours with a distance of 342.01 kilometers. Compared to the single objective case, this is an increase of 233.05% and 440.64%, respectively. These solutions are improved if the clusters are combined. The solutions of the same clusters post-combining are given in Figures 30a and 30b.



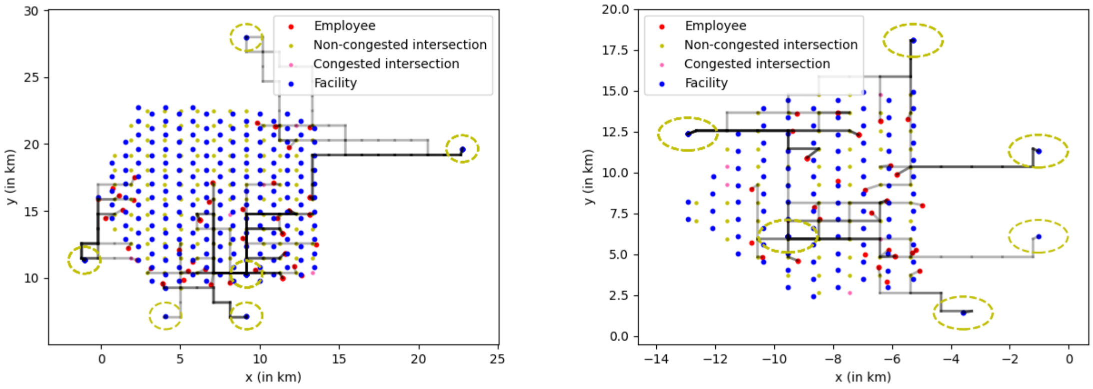
(a) Cluster 10

(b) Cluster 15

Figure 29: Solutions c-UFL (pre-combining) $\epsilon = 1$, instance 1.2km

Post-combining, while the system travel time decreases to 18.67 hours, the distance increases to 375.90

km. Hence, the travel time decreases significantly while the travel distance increases. This is due to the fact that some employees will travel a greater distance but these roads are non-congested which results in lower travelling time. Several employees on the outskirt of the cluster travel to facilities in other clusters such that their individual travelling time decreases. The same holds for cluster 15, the travelling time decreases from 12.75 to 10.14 hours and the travelling distance increases from 191.38 to 193.24 kilometers. Hence, it can be concluded that employees will travel further to mitigate congestion. We conclude that incorporating travel information into the decision making changes the outcome. It is therefore valuable and requires further research.



(a) Cluster 10

(b) Cluster 15

Figure 30: Solutions c-UFL (post-combining) $\epsilon = 1$, instance 1.2km

7 Conclusion

Mexico City is the third most congested city in the world, with only Paris and London performing worse (Traffic Index, 2024). This is reflected in the average commuting time; the average commuting from employees to their office is approximately 1.8 hours while this is 25 minutes in an average European city. The congestion peak can be reduced by efficiently locating the remote offices in the city S. Zhang et al. (2020). In this paper, we aim to efficiently locate remote offices in large-size instances.

7.1 Main conclusions

We answer the first research question, “How to efficiently allocate remote offices while maximizing the employee utility and minimizing the number of opened offices?”, by implementing a branch-and-bound framework in methods originally introduced by Fischetti et al. (2017) and solve the BD in every node. The SP of the BD is solved with the Dantzig algorithm for Knapsack problems, resulting in significant speed ups compared to the original BD and methods introduced in Beheshti Asl & MirHassani (2019). The opening of remote offices results in an increase of $\$200,602.97$ in utility of employees while the average commuting time decreases with 83.7%. The ϵ-constraint is implemented to restrict the number of open offices. We observe that the marginal effect of opening an additional remote office is diminishingly increasing.

Furthermore, we also address the second research question, “How does a finite office capacity influence the allocation of remote offices?”, by implementing a branch-and-bound framework in which the problem in every node is solved by a BD. Due to the capacity constraints, the Knapsack problem cannot be solved with the Dantzig algorithm and is solved by means of an LP. Due to the high number of variables, this could not be done in reasonable times for the larger instances. Hence, we implement a quantum-inspired heuristic and extend it by a k-Medoids heuristic to allocate the employees to open remote offices. The heuristic algorithm finds high-quality solutions in reasonable time and the k-Medoids problem efficiently allocates employees to open remote offices. By including capacities, we observe that the effect on utility is larger with a lower number of open facilities. Moreover, the number of employees per office is affected; if no capacities are ensured some offices are overcrowded and others have only a few employees but with the capacity constraints the number of employees per remote office is more balanced.

Lastly, constant travel speed and walking for hours throughout the city are harsh simplifications in a highly-congested city like Mexico City. In consequence, the last research question is “What is the effect of incorporating traffic congestion into the decision making for remote office locations?”. To tackle this problem, a bi-level approach is introduced with the objective of the employers as the upper level problem and objective of the employees as the lower level problem. The area of Mexico City is split into districts by a hierarchical clustering approach and each district is solved independently by solving a heuristic that approximates the non-linear objectives. Subsequently, we combine the independent districts with a combining heuristic. Compared to the constant travel speed, the average speed reduces with 4.5% while the travel distance increases with 12.55%. Hence, the congestion has significant impact on the solution and employees take other routes to minimize their personal travel time.

7.2 Limitations and further research

For further research, we mention some limitations of our methods and inherently some possible room for further research. First of all, the branch-and-bound framework of the UFL has shown to lack efficiency if the number of nodes to be branched is large. Hence, a cutting plane approach that more efficiently tightens the formulation could decrease the size of the tree and reduce the computation time. Regularly, the computation of these cutting planes is computationally expensive, hence an efficient manner for creating these cuts has to be implemented.

The dual variables of a constraint are needed to find the optimality cuts in the BD of the CUFL. Finding the exact solution has proven to be time-consuming. Hence, an approximation technique like gradient optimization could result in a faster algorithm. Moreover, the k-Medoids assignment heuristic in the APMQEA finds solution in a quick manner but these are heuristic. To our knowledge, there is no quick algorithm that solves the (relaxation of) generalized assignment problem to (near) optimality, hence this is possible room for research.

For the bi-level problem, we have considered the view of employers and employees. Considering the view of social planners or any other entity could give other solutions that might be more beneficial for the city planning, hence broadening to more viewpoints is room for further improvements. Alternatively, Mexico City is clustered into regions by a hierarchical method of k-means but these clusters remain approximations. Exactly clustering the employees into different regions may have a positive effect on the solution. Another limitation is that the entire region has to be split into regions to solve the c-UFL heuristically. Solving the problem as a whole could improve the solution significantly. The bottleneck of the solving mechanism is the lower level problem. Hence, approximating this could significantly decrease the computation time and thus could possibly solve large instances. Lastly, we have not yet quantified the quality of the heuristic for the congested problem formulation. Comparing the heuristic to optimal solutions or upper bounds could provide useful insights in the quality of the heuristic. Therefore, constructing a efficiently solvable upper bound on the problem formulation could give valuable insights.

References

Akinc, U., & Khumawala, B. M. (1977). An efficient branch and bound algorithm for the capacitated warehouse location problem. *Management Science*, *23*(6), 585–594.

Alenezy, E. J. (2020). Solving capacitated facility location problem using lagrangian decomposition and volume algorithm. *Advances in Operations Research*, *2020*(1), 5239176.

Bagloee, S. A., Asadi, M., Sarvi, M., & Patriksson, M. (2018). A hybrid machine-learning and optimization method to solve bi-level problems. *Expert Systems with Applications*, *95*, 142–152.

Balinski, M. L. (1966). On finding integer solutions to linear programs. In *Proceedings of the ibm scientific computing symposium on combinatorial problems* (pp. 225–248).

Bard, J. F., & Moore, J. T. (1990). A branch and bound algorithm for the bilevel programming problem. *SIAM Journal on Scientific and Statistical Computing*, *11*(2), 281–292.

Beheshti Asl, N., & MirHassani, S. A. (2019). Accelerating benders decomposition: multiple cuts via multiple solutions. *Journal of Combinatorial Optimization*, *37*, 806–826. Retrieved from `https://doi.org/10.1007/s10878-018-0320-8` doi: 10.1007/s10878-018-0320-8

Ben-Ameur, W., & Neto, J. (2007). Acceleration of cutting-plane and column generation algorithms: Applications to network design. *Networks: An International Journal*, *49*(1), 3–17.

Berman, O., & Krass, D. (2004). 11 facility location problems with stochastic demands and congestion. *Facility location: applications and theory*, 329.

Birch, C. P., Oom, S. P., & Beecham, J. A. (2007). Rectangular and hexagonal grids used for observation, experiment and simulation in ecology. *Ecological modelling*, *206*(3-4), 347–359.

Braess, D. (1968). Über ein paradoxon aus der verkehrsplanung. *Unternehmensforschung*, *12*, 258–268.

Camacho-Vallejo, J.-F., Corpus, C., & Villegas, J. G. (2023). Metaheuristics for bilevel optimization: A comprehensive review. *Computers & Operations Research*, 106410.

Cassioli, A., Di Lorenzo, D., Locatelli, M., Schoen, F., & Sciandrone, M. (2012). Machine learning for global optimization. *Computational Optimization and Applications*, *51*, 279–303.

Cormen, T. H., Leiserson, C. E., Rivest, R. L., & Stein, C. (2022). *Introduction to algorithms.* MIT press.

Cornuéjols, G., Nemhauser, G., & Wolsey, L. (1983). The uncapicitated facility location problem.

Council, N. (2000). Highway capacity manual. *Transportation Research Board, Washington, DC.*

Deb, K., & Jain, H. (2013). An evolutionary many-objective optimization algorithm using reference-point-based nondominated sorting approach, part i: solving problems with box constraints. *IEEE transactions on evolutionary computation*, *18*(4), 577–601.

Deb, K., Pratap, A., Agarwal, S., & Meyarivan, T. (2002). A fast and elitist multiobjective genetic algorithm: Nsga-ii. *IEEE transactions on evolutionary computation*, *6*(2), 182–197.

Du, B., Zhou, H., & Leus, R. (2020). A two-stage robust model for a reliable p-center facility location problem. *Applied Mathematical Modelling*, *77*, 99–114.

Esmikhani, S., Kazemipoor, H., Sobhani, F. M., & Molana, S. M. H. (2022). Solving fuzzy robust facility layout problem equipped with cranes using mps algorithm and modified nsga-ii. *Expert Systems with Applications*, *210*, 118402.

Eurostat. (2020, 10). Majority commuted less than 30 minutes in 2019. Retrieved from `https://ec.europa.eu/eurostat/web/products-eurostat-news/-/ddn-20201021-2`

Fischetti, M., Ljubić, I., & Sinnl, M. (2016). Benders decomposition without separability: A computational study for capacitated facility location problems. *European Journal of Operational Research*, *253*(3), 557–569.

Fischetti, M., Ljubić, I., & Sinnl, M. (2017). Redesigning benders decomposition for large-scale facility location. *Management Science*, *63*(7), 2146-2162. Retrieved from `https://doi.org/10.1287/mnsc.2016.2461` doi: 10.1287/mnsc.2016.2461

Fortuny-Amat, J., & McCarl, B. (1981). A representation and economic interpretation of a two-level programming problem. *Journal of the operational Research Society*, *32*, 783–792.

Geoffrion, A. M. (1972). Generalized benders decomposition. *Journal of optimization theory and applications*, *10*, 237–260.

Gordon, P., Kumar, A., & Richardson, H. W. (1989). The influence of metropolitan spatial structure on commuting time. *Journal of urban economics*, *26*(2), 138–151.

Görtz, S., & Klose, A. (2012). A simple but usually fast branch-and-bound algorithm for the capacitated facility location problem. *INFORMS Journal on Computing*, *24*(4), 597–610.

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

Hess, A., & Narteh-Yoe, S. (2020, 05). Productivity, sustainability, and economic growth in metropolises: Estimates of long-time commuting effects in developing countries.. doi: 10.5281/zenodo.3751057

Holmberg, K., Rönnqvist, M., & Yuan, D. (1999). An exact algorithm for the capacitated facility location problems with single sourcing. *European Journal of Operational Research*, *113*(3), 544–559.

Huo, J., Liu, J., & Gao, H. (2021). An nsga-ii algorithm with adaptive local search for a new double-row model solution to a multi-floor hospital facility layout problem. *Applied Sciences*, *11*(4), 1758.

Isermann, H. (1982). Linear lexicographic optimization. *Operations-Research-Spektrum*, *4*(4), 223–228.

Júdice, J. J., & Faustino, A. M. (1992). A sequential lcp method for bilevel linear programming. *Annals of Operations Research*, *34*(1), 89–106.

Kelley, J. E., Jr. (1960). The cutting-plane method for solving convex programs. *Journal of the society for Industrial and Applied Mathematics*, *8*(4), 703–712.

Khanduzi, R., & Rastegar, A. (2022). An efficient and robust hybrid metaheuristic method to solve a hierarchical bi-level protection-interdiction problem on real healthcare system. *Transactions on Emerging Telecommunications Technologies*, *33*(10), e4396.

Kim, D.-G., & Kim, Y.-D. (2013). A lagrangian heuristic algorithm for a public healthcare facility location problem. *Annals of Operations Research*, *206*, 221–240.

Klose, A., & Görtz, S. (2007). A branch-and-price algorithm for the capacitated facility location problem. *European Journal of Operational Research*, *179*(3), 1109–1125.

Kuo, R., & Huang, C. (2009). Application of particle swarm optimization algorithm for solving bi-level linear programming problem. *Computers & Mathematics with Applications*, *58*(4), 678–685.

Lara-Pulido, J. A., & Martinez-Cruz, A. L. (2023). Stated benefits of teleworking in mexico city: a discrete choice experiment on office workers. *Transportation*, *50*, 1743–1807. Retrieved from https://doi.org/10.1007/s11116-022-10293-w doi: 10.1007/s11116-022-10293-w

Lorena, L. A., & Senne, E. L. (2004). A column generation approach to capacitated p-median problems. *Computers & Operations Research*, *31*(6), 863–876.

Lu, H., Niu, R., Liu, J., & Zhu, Z. (2013). A chaotic non-dominated sorting genetic algorithm for the multi-objective automatic test task scheduling problem. *Applied Soft Computing*, *13*(5), 2790–2802.

Lu, T.-C., & Yu, G.-R. (2013). An adaptive population multi-objective quantum-inspired evolutionary algorithm for multi-objective 0/1 knapsack problems. *Information Sciences*, *243*, 39-56. Retrieved from https://www.sciencedirect.com/science/article/pii/S0020025513003150 doi: https://doi.org/10.1016/j.ins.2013.04.018

MacQueen, J., et al. (1967). Some methods for classification and analysis of multivariate observations. In *Proceedings of the fifth berkeley symposium on mathematical statistics and probability* (Vol. 1, pp. 281–297).

Magnanti, T. L., & Wong, R. T. (1981). Accelerating benders decomposition: Algorithmic enhancement and model selection criteria. *Operations Research*, *29*(3), 464–484. Retrieved from http://www.jstor.org/stable/170108

Marler, R. T., & Arora, J. S. (2010). The weighted sum method for multi-objective optimization: new insights. *Structural and multidisciplinary optimization*, *41*, 853–862.

Medaglia, A. L., Villegas, J. G., & Rodríguez-Coca, D. M. (2009). Hybrid biobjective evolutionary algorithms for the design of a hospital waste management network. *Journal of Heuristics*, *15*, 153–176.

Meyer, M. D., et al. (2016). *Transportation planning handbook.* John Wiley & Sons.

Mirchandani, P. B., & Francis, R. L. (1990). *Discrete location theory.*

Naoum-Sawaya, J., & Elhedhli, S. (2013). An interior-point benders based branch-and-cut algorithm for mixed integer programs. *Annals of Operations Research*, *210*, 33–55.

Patriksson, M. (2015). *The traffic assignment problem: models and methods.* Courier Dover Publications.

Peacock, J., & Nicholson, D. (1991). The large-scale clustering of radio galaxies. *Monthly Notices of the Royal Astronomical Society*, *253*(2), 307–319.

Poorzahedy, H., & Rouhani, O. M. (2007). Hybrid meta-heuristic algorithms for solving network design problem. *European journal of operational research*, *182*(2), 578–596.

Rahmaniani, R., Crainic, T. G., Gendreau, M., & Rei, W. (2017). The benders decomposition algorithm: A literature review. *European Journal of Operational Research*, *259*(3), 801–817.

Rdusseeun, L., & Kaufman, P. (1987). Clustering by means of medoids. In *Proceedings of the statistical data analysis based on the l1 norm conference, neuchatel, switzerland* (Vol. 31).

Rei, W., Cordeau, J.-F., Gendreau, M., & Soriano, P. (2009). Accelerating benders decomposition by local branching. *INFORMS Journal on Computing*, *21*(2), 333–345.

S., R., et al. (2012). *Network flow.* Retrieved from `https://courses.cs.duke.edu/fall12/compsci590.1/network_flow.pdf` (Accessed: 2024-07-27)

Senne, E. L., Lorena, L. A., & Pereira, M. A. (2005). A branch-and-price approach to p-median location problems. *Computers & operations research*, *32*(6), 1655–1664.

Shavarani, S. M., Mosallaeipour, S., Golabi, M., & İzbirak, G. (2019). A congested capacitated multi-level fuzzy facility location problem: An efficient drone delivery system. *Computers & Operations Research*, *108*, 57–68.

Stackelberg, H. v., et al. (1952). Theory of the market economy.

Stollsteimer, J. F. (1961). *The effect of technical change and output expansion on the optimum number, size, and location of pear marketing facilities in a california pear producing region.* University of California, Berkeley.

Tang, L., Jiang, W., & Saharidis, G. K. (2013). An improved benders decomposition algorithm for the logistics facility location problem with capacity expansions. *Annals of operations research*, *210*, 165–190.

Traffic Index. (2024). *Mexico city traffic index.* Retrieved from `https://trafficindex.org/mexico-city/` (Accessed: 2024-07-17)

Verma, S., Pant, M., & Snasel, V. (2021). A comprehensive review on nsga-ii for multi-objective combinatorial optimization problems. *IEEE access*, *9*, 57757–57791.

Vicente, L. N., & Calamai, P. H. (1994). Bilevel and multilevel programming: A bibliography review. *Journal of Global optimization*, *5*(3), 291–306.

Wentges, P. (1996). Accelerating benders' decomposition for the capacitated facility location problem. *Mathematical Methods of Operations Research*, *44*, 267–290.

Wierzbicki, A. P. (1982). A mathematical basis for satisficing decision making. *Mathematical modelling*, *3*(5), 391–405.

Wu, L.-Y., Zhang, X.-S., & Zhang, J.-L. (2006). Capacitated facility location problem with general setup cost. *Computers & Operations Research*, *33*(5), 1226–1241.

Yen, J. Y. (1971). Finding the k shortest loopless paths in a network. *management Science*, *17*(11), 712–716.

Zaferanieh, M., Abareshi, M., & Jafarzadeh, M. (2023). A bi-level p-facility network design problem in the presence of congestion. *Computers & Industrial Engineering*, *176*, 109010.

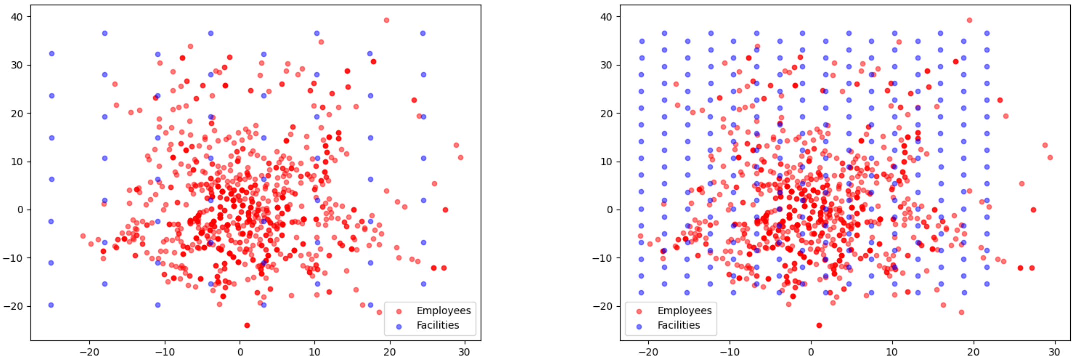
Zhang, S., Moeckel, R., Moreno, A. T., Shuai, B., & Gao, J. (2020). A work-life conflict perspective on telework. *Transportation Research Part A: Policy and Practice*, *141*, 51-68. Retrieved from `https://www.sciencedirect.com/science/article/pii/S0965856420307059` doi: https://doi.org/10.1016/j.tra.2020.09.007

Zhang, X., Tian, Y., Cheng, R., & Jin, Y. (2016). A decision variable clustering-based evolutionary algorithm for large-scale many-objective optimization. *IEEE Transactions on evolutionary Computation*, *22*(1), 97–112.

Ziar, E., Seifbarghy, M., Bashiri, M., & Tjahjono, B. (2023). An efficient environmentally friendly transportation network design via dry ports: a bi-level programming approach. *Annals of Operations Research*, *322*(2), 1143–1166.

8 Appendix

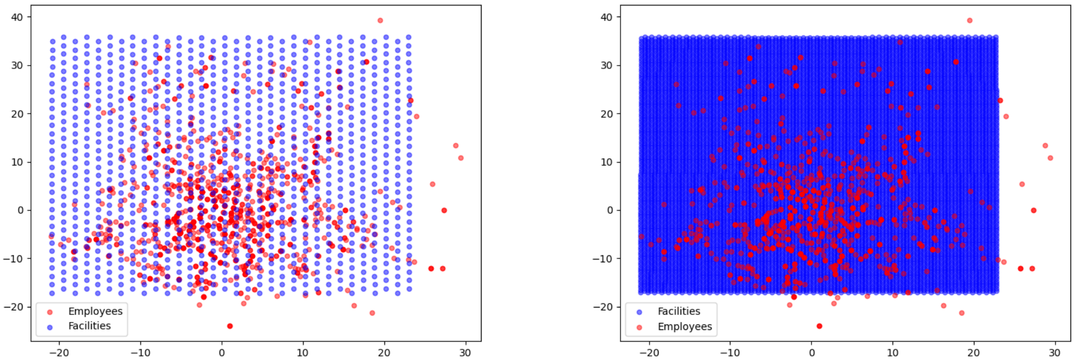
8.1 Data



(a) Facilities 10km

(b) Facilities 4km

Figure 31: Figures facilities compared to employees



(a) Facilities 2km

(b) Facilities 0.6km

Figure 32: Figures facilities compared to employees

8.2 Methodology

K- shortest path algorithm

A well known algorithm to find the K shortest paths is Yen's algorithm (Yen, 1971). This algorithm starts of by finding the shortest path from the source to the sink. This path is altered such that at least $K-1$ distinct paths are found and the K shortest paths are retained. A pseudocode of this algorithm is given by Algorithm 2.

Algorithm 2 k-shortest path Algorithm

```
Input: the graph $G$, the source $S$, the sink $Z$, number of shortest paths $K$
Output: K shortest paths from S to Z
/* Set the set of K shortest paths to an empty set */
A = ∅
/* Get the shortest path from S to Z by Dijkstra's algorithm */
shortest path = Dijkstra's algorithm(G, S, Z)
A = A ∪ shortest path
for k from 1 to K do
    /* For every node from the first to second last, create a new distinct spurpath */
    B = ∅
    for a from 0 to size(A[k-1])-2 do
        Spur node = node a from  A[k-1]
        Root path = every node up to, but not including $a$ in path $A[k-1]$
        Remove every node in the root path from graph G
        for p in A do
            Remove arcs that are already in a shortest path p
        /* Perform Dijkstra's algorithm on the changed graph G */
        Spur path = Dijkstra's algorithm(G, spurnode, Z)
        Combine the root path and the spur path
        Add potential shortest path to set B
    Order the set B from smallest to highest cost and add smallest cost path to set $A$
```

Linear approximation objective function

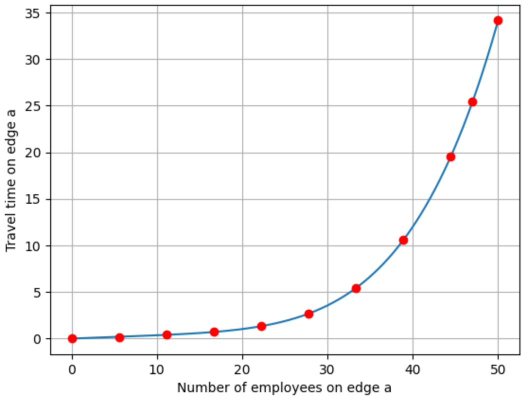
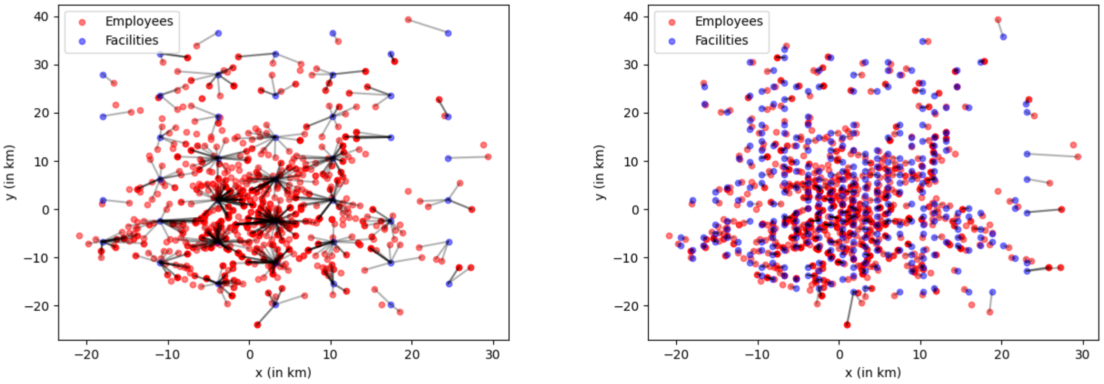


Figure 33: Linear approximation edge with $tp = 0$

The travel time of the non-linear function is approximated by the piecewise linear approximation that connects the neighbouring read dots in Figure 33 by linear functions. Ten of the red dots are equally distributed over the region, while the eleventh is added on the point with highest second derivative.

8.3 Results

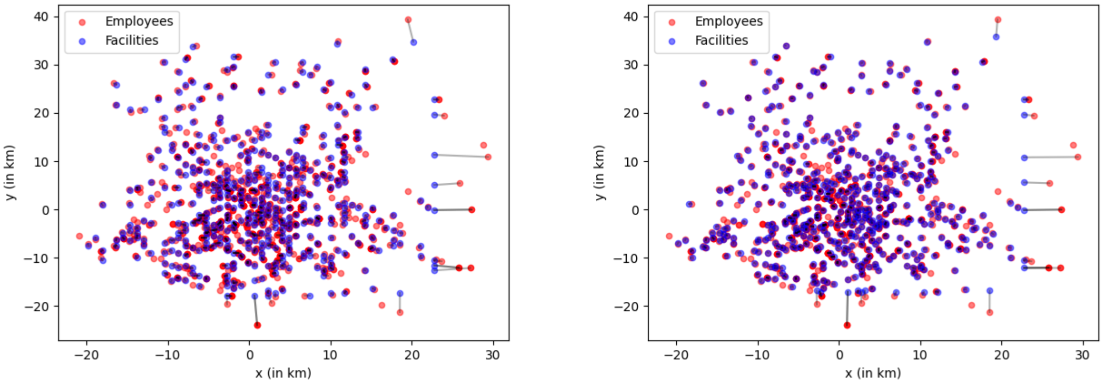
Single objective UFLP



(a) Result 10km

(b) Result 2km

Figure 34: Employee allocation

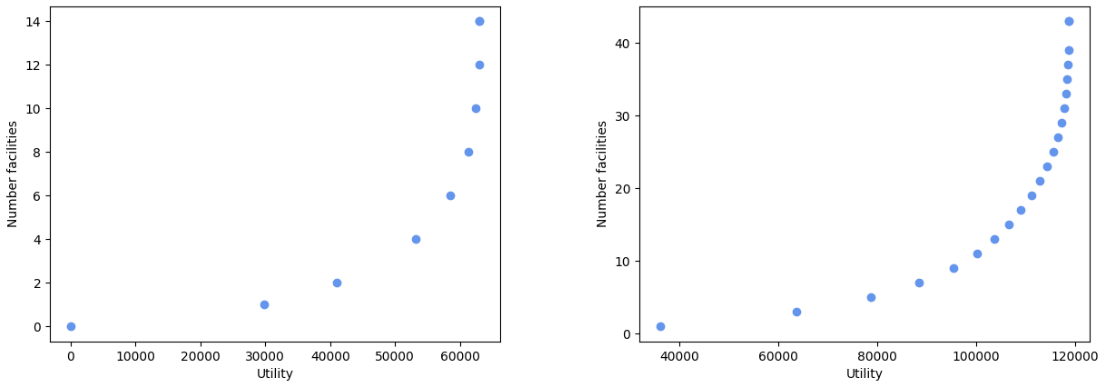


(a) Result 1.2km

(b) Result 0.6km

Figure 35: Employee allocation

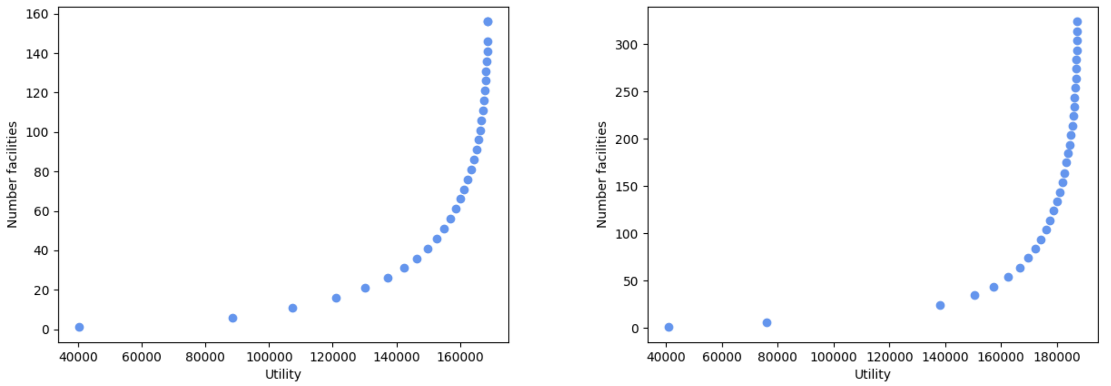
Multi objective UFLP



(a) Result 20km

(b) Result 10km

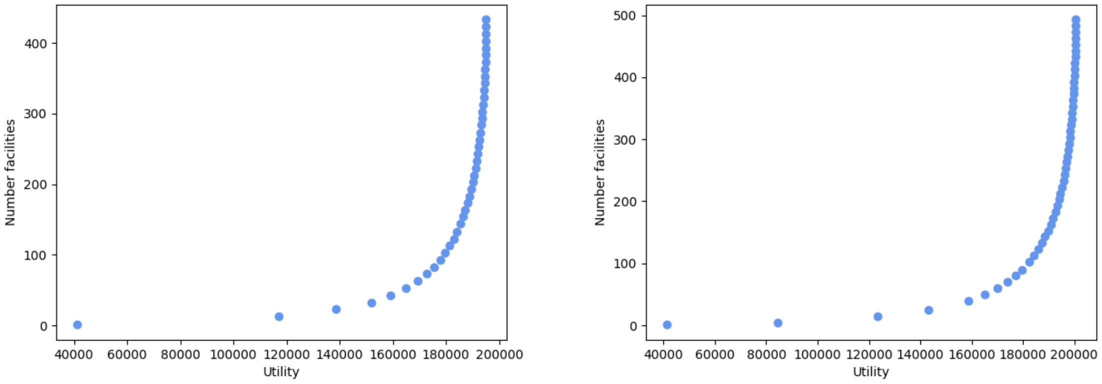
Figure 36: Pareto Front, UFLP



(a) Result 4km

(b) Result 2km

Figure 37: Pareto Front, UFLP



(a) Result 1.2km

(b) Result 0.6km

Figure 38: Pareto Front, UFLP

Multi objective CFLP

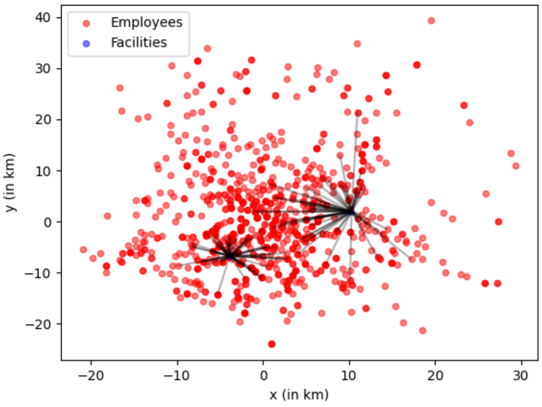
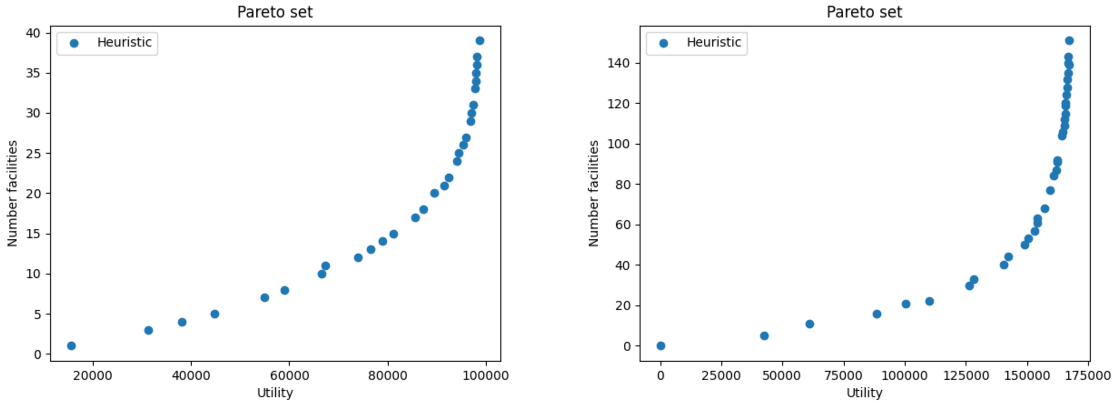


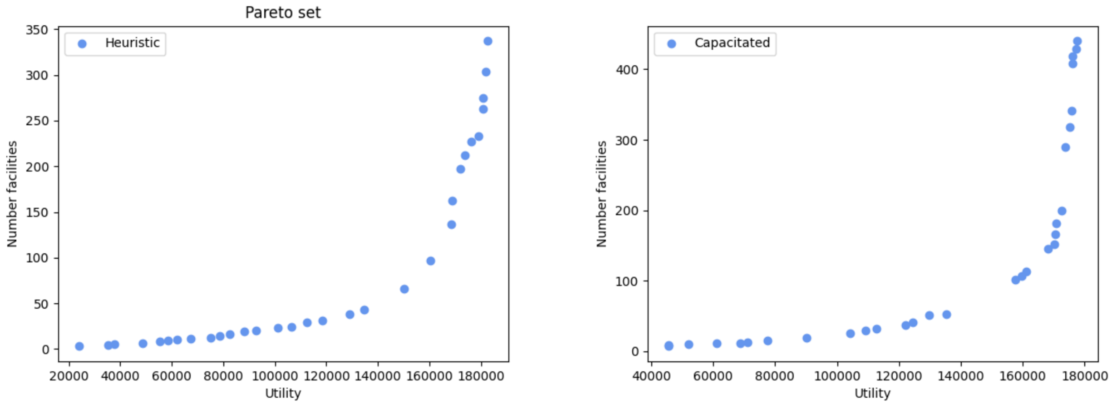
Figure 39: Result 2 facilities, 60 capacity, 20km



(a) Result 10km

(b) Result 4km

Figure 40: Pareto Front, CFLP



(a) Result 2km

(b) Result 0.6km

Figure 41: Pareto Front, CFLP