Generation and optimisation of real-world static and dynamic location-allocation problems with application to the telecommunications industry.

ANKRAH, R.B.

2019

Generation and Optimisation of Real-World Static and Dynamic Location-Allocation Problems with Application to the Telecommunications Industry

Reginald Boafo Ankrah

Computing Science and Digital Media

Robert Gordon University

A thesis submitted in partial fulfilment of the requirements of the

Robert Gordon University for the degree of

Doctor of Philosophy

December, 2019

Abstract

The Location-allocation (LA) problem concerns the location of facilities and the allocation of demand to minimise or maximise a particular function such as cost, profit or a measure of distance. Many formulations of LA problems have been presented in the literature to capture and study the unique aspects of real-world problems. However, some real-world aspects, such as resilience is still lacking in the literature. Resilience ensures uninterrupted supply of demand and enhances the quality of service. Due to changes in population shift, market size, economic and labour market which often causes demand to be stochastic, a reasonable LA problem formulation should consider some aspect of future uncertainties. Almost all LA problem formulations in the literature that capture some aspect of future uncertainties fall in the domain of dynamic optimisation problems where new facilities are located every time the environment changes. However, considering the substantial cost associated with locating a new facility, it becomes infeasible to locate facilities each time the environment changes.

In this study, we propose and investigate variations of LA problem formulations. Firstly, we develop and study new LA formulations that extend the location of facilities and the allocation of demand to add a layer of resilience. We apply the Population-based incremental learning algorithm for the first time in the literature to solve the new novel LA formulations. Secondly, we propose and study a new dynamic formulation of the LA problem where facilities are opened once at the start of a defined period and are expected to be satisfactory in servicing customers demands irrespective of changes in customer distribution. The problem is based on the idea that customers will change locations over a defined period, and these changes have to be taken into account when establishing facilities to service changing customers distributions. Thirdly, we employ a simulation-based optimisation approach to tackle the new dynamic formulation. Owing to the high computational costs associated with simulation-based optimisation, we investigate the concept of Racing, an approach used in model selection, to reduce the high computational cost by employing the minimum number of simulations for solution selection.

Declaration of Authorship

I declare that I am the sole author of this thesis and that all verbatim extracts contained in the thesis have been identified as such and all sources of information have been specifically acknowledged in the bibliography. Parts of the work presented in this thesis have appeared in the following publications:

- ANKRAH, R., REGNIER-COUDERT O, MCCALL, J., CONWAY, A, HARDWICK, A. Performance Analysis of GA and PBIL Variants for Real-World Location-Allocation Problems. In2018 IEEE Congress on Evolutionary Computation (CEC) 2018 Jul 8 (pp. 1-8). IEEE.
(**Chapter 3**)

- ANKRAH, R., LACROIX, B., MCCALL, J., HARDWICK, A. and CONWAY, A. A Holistic Metric Approach to Solving the Dynamic Location-Allocation Problem. In International Conference on Innovative Techniques and Applications of Artificial Intelligence 2018 Dec 11 (pp. 433-439). Springer, Cham
(**Chapter 4**)

- ANKRAH, R., LACROIX, B., MCCALL, J., HARDWICK, A. and CONWAY, A. 2019. Introducing the dynamic customer location-allocation problem. In Proceedings of the 2019 Institute of Electrical and Electronics Engineers (IEEE) Congress on evolutionary computation (IEEE CEC 2019), 10-13 June 2019, Wellington, NZ.
(**Chapter 5**)

- ANKRAH, R., LACROIX, B., MCCALL, J., HARDWICK, A. and CONWAY, A., Owusu, G. 2020. Racing Strategy for the Dynamic-Customer Location-Allocation Problem. In Proceedings of the 2020 Institute of Electrical and Electronics Engineers (IEEE) Congress on evolutionary computation (IEEE CEC 2019), 19-24 July 2020, Glasgow, UK.
(**Chapter 6**) (Accepted for publication)

Dedication

I dedicate this work to my father, Mr Samuel A. Ankrah. Thank you, Dad, for always believing in me. You will always have a special place in my heart.

Acknowledgements

I want to express my most profound appreciation to my supervisory team, Prof. John McCall, Dr Benjamin Lacroix and Dr Olivier Regnier-Courdert from Robert Gordon University (RGU), and Dr Andrew Hardwick and Anthony Conway from British Telecommunications plc (BT). You always offered me the best support and advice whenever I needed them. I am also grateful for the critical and robust supervision you have provided for my research - thank you very much.

This research benefited from generous funding from RGU and BT, for which I am very grateful. In the same light, I wish to further thank RGU for offering the training and infrastructure to conduct my research, and travel opportunities to participate in conferences and other related events. Furthermore, BT, who helped immensely through relevant activities and resources.

I appreciate the many insightful feedback received from fellow research students and staff members from the AI research group who provided forums for personal development. I also appreciate the help and support from the administrative staff at the School of Computing Science and Digital Media.

Finally, I would like to express my gratitude to my family for their constant love, support and encouragement.

Contents

List of Figures

List of Tables

Abbreviations

CEC Congress on Evolutionary Computation

CSD Coefficient of Spatial Differentiation

DC-LA Dynamic-customer location-allocation

DOP Dynamic optimisation problem

EDA Estimation distribution algorithm

EMD Earth movers distance

FGTS Fine-grained tournament selection

h-BOA Hierarchical-Bayesian optimisation algorithm

LA Location-allocation

LARP Location-allocation resilience problem

LARPR Location-allocation resilience problem with restrictions

MLS Maximum likelihood solution

PBIL Population-based incremental learning algorithm

PSS Partial Space Search

PV probability vector

ROOT Robust optimisation over time

S-BO Simulation-based optimisation

TD Transformed distance

Chapter 1

Research Background and Motivation

Location-allocation (LA) problem is a branch of location problems that involves choosing facility locations to service the demands of customers aimed at reducing the overall total costs or maximising total profits [105]. The LA problem often occurs in practical settings where facilities provide similar services. The location of facilities and the allocation of demand to these facilities are critical elements in strategic planning for a wide range of private and public organisations. In the past decade, the LA problem has attracted much attention from the operation research community. Research has focused on both problem formulations and algorithms to solve them in diverse settings within the private sectors and public sectors. LA problem is considered to be NP-hard, which means that the use of exact methods can be impracticable, especially for large instances of the problem. For this reason, meta-heuristic methods have been proposed by the operational research community that is capable of addressing the difficult real-world formulations of LA problem.

Many real-world problems have been formulated as LA problems within the private sectors such as the location of industrial plants, banks, telecommunication facilities, retail facilities and the public sectors such as hospitals, fire stations, police stations, and post offices [60]. The distinctive characteristics of establishing facilities to service demand within the private and public sectors have led to many different formulations of LA problems. Although many formulations of LA problem exists in the literature that captures many aspects of real-world problems, no LA problem formulation exists that captures the aspect of resilience. Resilience here is the option of providing backup services to customers to ensure uninterruptible supply of services. The aspect of resilience is vital, especially in settings where the success of customers operations primary rely on the consistent and uninterruptible supply of services. E.g. data centres require a constant high-speed network connection to operate; E-commerce organisations such as Amazon, Google, Facebook and E-bay require constant high-speed network connections to keep their website running efficiently; Financial institutions require consistent network connections to keep their servers up and running in order to provide online banking services to customers. For such companies, an interruption in network connections could result in huge financial reputational risk.

Although LA problem formulations are essential study topics, the changes in population, market size, environmental factors and the rapid advancement in technology often drives the need of consumers, which causes demand to be stochastic [8]. Therefore a reasonable LA formulation should consider some aspect of future uncertainty. In response to the dynamic nature of customers distribution, many dynamic formulations of LA problems have been presented in the literature to capture many of the dynamic characteristics of problems within the private and public sector. A study of the dynamic formulations of LA problem in the literature shows that almost all the dynamic formulations consider the location and relocation of facilities overtime to service the changing distribution of customers [18; 23; 43; 50; 109; 160; 165; 178]. These formulations are known in the literature as multi-period or explicit LA problems and falls in the context of dynamic optimisation problems. In dynamic optimisation problems, the decision-maker has to determine and implement a solution every time the environment changes. In multi-period LA formulations, anytime the distributions of customers change, new facilities are required to be opened, or existing facilities are relocated to service customers needs adequately.

In the absence of substantial costs of opening new facilities or relocating existing facilities to service the changing distribution of customers, multi-period LA problem formulations offer a good approach to tackling real-world dynamic problems. However, in settings where the opening or relocation of facilities attract considerable capital and resource investment, which often brings to bear significant financial, technical and reputational risk, it becomes infeasible to open new facilities or relocate existing facilities to adequately service customers demands each time customers change locations. In such problem scenarios, it becomes imperative that facility location choices executed today consider expected future circumstances. Considering changing distributions of customers when locating facilities for the first time will help to ensure that facilities are not only ideal for current conditions but also stay useful over a defined horizon [45]. This type of LA problem formulation requires that facilities are established once at the start of a defined horizon and are expected to be effective in servicing the changing demands of customers over a specified period. Such dynamic LA formulations are referred to as Implicit LA problems and fall in the domain of Robust optimisation over time (ROOT) [63]. In ROOT the quality of a solution, i.e. the location of facilities continues to be satisfactory and is relatively indifferent to the environmental fluctuations during the defined time interval. Very few works exist in the literature that studies LA problems in the context of ROOT. To the best of our knowledge, the only work that studied an LA problem formulation in the context of ROOT was the work presented by Daskin et al. in [45] where authors sought to find an optimal or near-optimal location of facilities to service changing distribution of customers over an infinite horizon. The low level of research work in this area makes it an area of interest for study.

In the dynamic LA problem literature, two main approaches are considered for predicting potential changes in customers distributions. These are forecast and scenario planning. Forecast concerns predicting the future as accurately as possible, given all the information available, including historical data and knowledge of any future events that might impact the forecasts [45]. The use of forecast is often favourable in instances with low uncertainty and low complexity because it is justifiable to make numerical predictions about variables of interest without extensively conditioning the forecast on key assumptions about other variables that may influence it. Scenario planning on the other-hand concerns predicting the future by generating many scenarios of how the future might unfold. Scenario planning is often employed in problems with high uncertainties where predetermined elements, common to the scenario are combined with critical uncertainties, that vary between the scenarios [167]. Forecasting predicts the potential changes in one future while scenario planning projects the potential changes in multiple futures. Due to the dynamic nature of real-world problems, scenario planning offers a robust approach to predicting future changes.

However, many of the work presented on dynamic LA problem formulations in the literature employed forecast models to predict future changes [16; 26; 45; 64; 158]. Very few works on dynamic LA problem formulations exist in the literature that employs scenario planning for predicting future changes [12; 66; 110]. To the best of our knowledge, no dynamic LA problem formulation exists in the literature that employs scenario planning for predicting future changes in the context of ROOT. The reason for the lack of work in the literature that employ scenario planning for predicting future changes maybe the fact that the class of the dynamic LA problems studied in the literature have low uncertainties and hence forecast models are more appropriate for such formulations or the fact that scenario planning comes with a high computational cost as a result of evaluating the many scenarios against decision variables. Decision variables represent the solution to a problem, i.e. the location of facilities that can be opened. In the context of LA problem, decision variables are often presented as a binary string x = $\{x_1, ..., x_m\} \in \{0, 1\}^m$ where 1 represents an opened facility and 0 represent a closed facility. The evaluation of a decision variable against a scenario is called a simulation. The decision variables are provided by an optimisation algorithm which is then evaluated against a scenario, and a result is produced according to the optimisation objective function. The process will repeatedly continue until it results in a satisfactory solution or terminated according to a prescribed condition. This solution approach is termed as simulation-based optimisation. Because all decision variables in a Simulation-based optimisation (S-BO) are evaluated against many scenarios many times, they often have very high computational costs. E.g. suppose we have 50 decision variables and 200 scenarios. We evaluate each decision variable against a scenario 100 times. That will give us a total of 1,000,000 simulations. If a simulation takes 0.5 seconds, then on average, it will take us 500,000 seconds to evaluate all 50 decision variables against all 200 scenarios. Considering that the time it takes to evaluate a decision variable against a scenario is problem dependent, the more significant the problem, the more costly the computational cost.

A review of the literature on the LA problem found three significant areas of study that have received very little attention in the literature. These areas are (1) the aspect of resilience; (2) the study of a dynamic LA problem formulation in the context of ROOT; and (3) the use of scenario planning in modelling changes in a dynamic LA problem in the context of ROOT. We are thus motivated to focus our study in these three key areas using a large practical scale real-world example of service company from the telecommunications industry as a case study. We introduce the case study in Section 1.1.

1.1 Case Study

To adequately and efficiently supply bandwidth demand over connection lines to customers locations, service companies have to strategically position facilities in areas that offer optimal coverage to customers concerning the minimal distance between a facility location and customers premises. Customers here refer to big corporations that include financial institutions, data centres and technology companies. A facility receives prodigious amounts of bandwidth from international leased lines and then breaks down the bandwidth into smaller bandwidth sizes according to customers demands. The requested bandwidth of a customer is carried over a wired connection from a facility to the customer site. Hence, the further away a customer site is from a facility, the more expensive it is to service the demand of the customer as this will require an extra wired connection to be constructed to the customer site.

The quality of connection a customer receives is also primarily influenced by how close or far a customer site is to a facility. The further away a customer is from a facility, the more likely the customer is to experience a disruption in connectivity or high latency, i.e. the delay before a transfer of data begins following an instruction for its transfer. To ensure the quality of service delivery and competitive cost of connections, facilities have to be located concerning customers locations ensuring that customers are close to facility sites to ensure the best of services. In light of this, the best option to ensure the best service to customers is to build facilities close to customers sites. However, considering that establishing facilities involve considerable capital and resource investment and the fact that customers locations are often spread across a wide area, it becomes infeasible to build facilities next to all customers sites. Hence there is the need to find optimal locations to establish the facilities that can supply the demand of customers whiles at the same time minimising the overall operational cost of building facilities and servicing customers demands.

The nature of customers businesses requires constant and uninterrupted connections from facilities to customer sites. Any disruption to customers connections will harm their operations, often leading to financial and market loss. As a means of ensuring constant up-time for customers, the concept of resilience becomes an essential factor to consider when deciding the location of facilities. Resilience here is the option of a second bandwidth connection to customer site from a different facility that serves as a backup in the event of a disruption to the primary facility or the primary wired connection to a customer's site.

In the real-world several factors causes the markets to be dynamic. Hence, for customers to stay relevant in a competitive market requires them to adapt to changing trends. The changes in trends are often prompted by factors such as technological advancement, economic and labour markets shifts. For this reason, customers are continually relocating to cities that offer them more significant market shares as well as sound economic policies for operations. For service companies to adequately service the changing distributions and demands of their customers, they are required to find the optimal locations of their facilities that offer them the best trade-off between the costs involved in establishing and running facilities and the quality of service their customers receive. Due to the high uncertainty in the movement of customers, it becomes difficult for the service companies to predict with ease how their customers will evolve. The high uncertainty is as a result of the many factors that contribute to how attractive a city is to a customer such as economic policies, market competition, life stage of the market, consumer preferences. The constant differentiation of customers products or services also influences their target market. Therefore it becomes a difficult task to forecast how customers will move between cities in the future. The dynamic nature of the real-world problem described above requires the use of scenario planning to capture or model the future distribution of customers.

The real-world case study presented above highlights the relevance of the three key areas of this research: (1) The aspect of resilience, (2) Formulation of dynamic LA problem in the context of ROOT and (3) Use of scenario planning for predicting future changes in customer movements.

1.2 Research Aims

Our work builds on the knowledge that establishing facilities to adequately service customers demands over an extended period of time must take into consideration how customers will move locations over time. However, simulating customers movements in a dynamic environment comes with a high computational cost since many movement scenarios of customers have to be generated to provide a good measure for making decisions to locate facilities. Hence this research intends to find out if we can decide the location of facilities without taking customers movements into consideration over a defined period and thereby avoid high computational overhead. Alternatively, ascertain if there is value in expending the extra computational effort by simulating the movements of customers when deciding the locations of facilities which are expected to be operable over an extended period of time. If there is value in simulating the movements of customers to decide the location of facilities, we intend to find out how we can reduce the computational effort required to simulate and evaluate the many customer movements scenarios. The following questions in Section 1.2.1 are set as guidelines to fulfil these aims.

1.2.1 Research Questions

Based on our interest in this study, we seek to investigate the following research questions ((RQ1)-(RQ3)):

(RQ1) What suitable LA formulations can capture the real-world problem complexities of a service company such as telecommunication company?

(RQ2) How can simulation-based optimisation be exploited to help find a satisfactory and robust solution concerning the changing customers' demands over time in the context of ROOT?

(RQ3) How can the high computational costs that come with simulation-based optimisation be addressed?

1.2.2 Research Objectives

In order to address the research questions asked in Section 1.2.1, this thesis has the following objectives:

(O1) Develop new formulations that capture the real-world complexities of service companies as an LA problem.

This objective addresses (RQ1) in part. We are keen to capture the complexities of the real-world service company in an LA problem formulation. The new LA problem formulations will extend existing LA problem formulations to capture the aspect of resilience, and the changes in customers demand over time in the context of ROOT.

(O2) Propose a new problem instance to study the new LA problem formulations.

This objective address (RQ1) in part. Due to the specific characteristics of our new LA problem formulations, existing problem instances such as the uncapacitated facility location datasets in the OR Library [13] will not be adequate to study the new formulations. Hence, we will develop new problem instances in collaboration with industry experts from the telecommunication industry to help study the new problem formulations.

(O3) Investigate optimisation algorithms suitable for solving the new LA problem formulations.

This objective address (RQ1) in part. Many state-of-the-art algorithms exist in the literature that has been used to tackle different LA problem formulations. We will investigate the major-state-of-the-arts optimisation algorithms presented in the literature to find a suitable algorithm that can find satisfactory solutions to the new LA problem formulations.

(O4) Develop a stochastic simulation model to simulate the changes in customers distributions over time.

This objective address (RQ2) in part. In order to capture the changing demand of customers over time, we will develop a stochastic simulation model that takes as input some predetermined values and random values to simulate possible changes in customers demands over a specified period.

(O5) Investigate a way to help reduce the high computational cost associated with the simulation-based optimisation.

This objective address (RQ3). The idea here is to find a way to reduce a large number of simulations often required to evaluate a solution in a simulation-based optimisation.

1.3 Contributions

In this Section, we present the original research generated in the process of meeting our objectives.

In Chapter 3, we introduce two new novel non-linear models of LA problem motivated by real-world scenarios from the telecommunication industry in collaboration with industry experts. The first formulation we call Location-allocation resilience problem (LARP), which extends the location of facilities and the allocation of customers to include a resilience layer. The second formulation, called Location-allocation resilience problem with restrictions (LARPR) is a constrained version of LARP due to budget constraints for establishing facilities. We also present a new problem instance in collaboration with industry experts from the telecommunication field to help study the new non-linear LA problem formulations. In addition to finding a suitable state-of-the-art algorithm from the literature to tackle the new LA problem formulations, we study the effectiveness of Population-based incremental learning algorithm (PBIL) algorithm [58; 59] for solving the LA problem formulations. PBIL has been shown in the literature to be effective in tackling combinatorial problems. To the best of our knowledge, this is the first time in the literature, PBIL will be employed to solve an LA problem formulation. Research work presented in Chapter 3 has been published in IEEE Congress on Evolutionary Computation (CEC) 2018 Jul 8 (pp. 1-8) [5].

In Chapter 4 we introduce a new dynamic formulation of LA problem we call Dynamic-customer location-allocation (DC-LA) problem in the context of Robust optimisation over time (ROOT) which takes into account the actualised servicing costs and the movement of customers over a defined period. The new dynamic variant stems from the telecommunication industry where customers change locations over a defined horizon, and these changes have to be taken into account when establishing facilities to service changing customers locations. To help measure how good a solution is to the new problem, we define a stochastic dynamic evaluation function that takes the movement of customers into account when evaluating a solution to the DC-LA problem. We also define a baseline evaluation function by which to compare the performance of the stochastic dynamic evaluation function. The baseline evaluation function assumes that customers do not move over time. Research work presented in Chapter 4 has been published in the International Conference on Innovative Techniques and Applications of Artificial Intelligence 2018 Dec 11 (pp. 433-439). Springer, Cham. [3]

In Chapter 5, we develop 1440 new problem instances based on the different parameters of DC-LA problem to help study the new dynamic LA problem formulation. Research work presented in Chapter 5 has been published in the Proceedings of the 2019 Institute of Electrical and Electronics Engineers (IEEE) Congress on evolutionary computation (IEEE CEC 2019), 10-13 June 2019, Wellington, NZ. [4].

In Chapter 6, we adapt the concept of racing [104] to help reduce the number of evaluations required to find good and robust solutions to DC-LA problem. Racing uses statistical tests to compare solutions in the evolutionary process until a statistical difference is found between solutions. After the test, solutions found to be statistically significant from the best solution in the population are discarded. The advantage of using a statistical test in the context of simulation-based optimisation is that the test can be performed iteratively until statistical significance is found. The adaptation of racing is to help ensure that the minimum number of simulations is performed to detect statistical difference to support solution selection. By adapting racing to DC-LA problem, we can reduce the total number of simulations required to find a robust solution to the new dynamic LA formulation which leads to a reduction in the high computational cost associated with simulation-based optimisation.

1.4 Thesis Structure

The rest of the Chapters are organised as follows: In Chapter 2, we review relevant works in the literature, focusing on how real-world problems have been formulated as an LA problem and highlight areas for further research. We discuss the recent solutions presented in the literature for tackling LA problem. In Chapter 3, we introduce two new formulations of LA problem motivated by real-world scenarios from the telecommunication industry in collaboration with industry experts and explore state-of-the-art solutions from the literature to solve the new formulations. In Chapter 4 we introduce a new dynamic formulation of LA problem we call DC-LA problem in the context of Robust optimisation over time (ROOT) which takes into account the actualised servicing costs and the movement of customers over a defined period. In Chapter 5, we develop 1440 new problem instances based on the different parameters of DC-LA problem to help study the new dynamic LA problem formulation. In Chapter 6, we adapt the concept of racing [104] to help reduce the number of evaluations required to find good and robust solutions to DC-LA problem. In Chapter 7, we apply our adaptation of racing to find robust solutions to a changing real-world scenario. Finally, in Chapter 8, we present a summary of our contributions and a review of the extent to which we met our research objectives. We also outline the limitations of the work presented in this thesis and considerations for future extensions.

Chapter 2

Literature Review

This chapter explores the literature around Location-allocation (LA) problem and examines how many real-world problems have been formulated as LA problems. In reviewing the literature on LA problem, we also explore how formulations of LA problem have been tackled.

2.1 Location-Allocation (LA) Problems

Location-allocation problem [157] is a branch of location problem that involves choosing facility locations in a region of concern to service the demands of customers aimed at minimising or maximising costs. Costs here could represent operational costs, service delivery, service coverage or profit the aspect of locating facilities and allocating demand to these facilities impacts on various logistics and operational decisions. In order to formulate a real-world problem as a Location-allocation (LA) problem, we first have to define the essential elements that impact the decision of locating facilities and allocating customers. These elements are facilities, customers, location and efficiency criteria. Facilities are the objects to be located to provide a service or good; customers refer to the users of the facilities who demand certain services or goods; locations are the set of candidate points for facility sites, and efficiency criteria defines the quality of interaction between facilities and customers which is expressed as the cost to be minimised or maximised.

2.1.1 Facilities

Facilities [54] denote a vast class of objects for which we must determine a spatial position to optimise their intercommunication with pre-existing objects. Typical examples of facilities within the context of LA problem are warehouses, manufacturing factories, schools, clinics, retail outlets and several other productions, industrial or government structures. The main properties characterising facilities are their *number*, their *type* and the *cost* associated with them.

- Number: In the simplest case of LA problem [52], only one facility is to be located relative to some existing facilities. This kind of LA problem is referred to as a *single-facility*. However, in the general case, LA formulations involve the concurrent location of multiple facilities called *multi-facility*. In multi-facility LA problem, the decision involves a trade-off among the improved accessibility of the customers to the facilities obtained by opening a larger number of centres, and the increased costs for establishing and operating the facilities.
- Type: Type [52] involves the capacity, service, and structural considerations of facilities. The first case of LA problem requires the placement of identical facilities concerning both size and kind of service they can supply. However, other applications may require the location of simultaneous facilities which differ from each other such as the location of both manufacturing plants and warehouses to produce and distribute goods in an efficient way. The number of levels at which different facilities operate distinguishes LA problems between *single-echelon* and *multi-echelon*. LA problems can also be contrasted according to *single-service* and *multi-service*, based upon whether the facilities can render a single or several varieties of service respectively. Moreover, some models admit facilities that supply an infinite demand, whereas other models look for the best placement of facilities with limited production or supply capacity. In this respect, models are denoted as capacitated or uncapacitated, respectively.
- Cost: The costs of facilities involve the fixed expenses incurred for their opening and the variable charges related to the service delivery. While the first type of costs are usually connected to the specific location where the facilities are established, the second is usually some function of the distance from the user of the service.

2.1.2 Customers

The word *customer* can be used in its most traditional meaning to denote a person or entity requiring accessibility to a service or supply of a good. In an LA problem, it is vital to know customers *distribution*, *demand* and *behaviour*.

- Distribution: It may be assumed that customers are either spread uniformly over a given set or that they are located at specific points in space.
- Demand: Each customer is assigned a weight which expresses the amount of service the customer requires, i.e. its *demand* [53]. The weight assigned to a customer can be a measure of the distance between the customer and the facility or the actual cost incurred for servicing a customer. When a customer is a single user, the associated weight can be a unit weight, or a fixed weight representing the sufficient demand of the user for the good or service. When

the demand point is symbolic of an area destination for the service (such as a community or a city), the weight is often to account for the total demand arising in that area (for instance it might be a function of the population size). In both cases of single and demand areas, the demand may not necessarily be known with certainty. If facilities provide essential services (such as issuing driver's licenses or providing polling places on election day), consumer demand may be deterministic and known a priori. However, for facilities that provide non-essential services (for example, fast food restaurants, retail stores or ATMs), consumer demand may be a function of the total cost of receiving service.

- Behaviour: In some LA problem formulations customers are free to choose from which facility to be served, in which case the question is whether they will always patronise the closest facility or use some other criterion which reflects their preferences. In this respect, they can behave individually or as a group, meaning that when choosing a facility, they might consider the convenience of all the other members of the group. Conversely, location problems exist where the assignment of customers to specific facilities is compulsory, as in the case of schools located in some districts.

2.1.3 Locations

The physical site where facilities can be located. Concerning the set of available points, three spatial descriptions are studied in literature: *discrete*, *continuous* and *network*.

- Discrete (site selection): The decision-maker can specify a list of plausible sites for facility locations [55]. This kind of solution space proves to be very flexible because it makes it possible to incorporate several geographical and economic features into the model. Furthermore, the discrete space is the most natural option for designing problems when land availability, zoning regulations or the presence of pre-existing structures require that new facilities be opened only at some pre-specified points within the area under consideration.

- Continuous (site generation): In a continuous space [55], no a-priori knowledge of particular candidate sites are assumed, and the generation of appropriate sites is left to the model at hand.

- Network: LA problem represented on networks can be perceived as both discrete or continuous [55], depending on whether connections are regarded as a continuous collection of candidate points for facility location, or only the nodes are available for the location of new facilities. For some applications in both public and private service systems, the graph-theoretic approach lends itself in an excellent way to an intuitive representation of the problem. Some examples include the set up of plants in a transportation system to reduce production and shipment costs and the placement of emergency services in rural areas to guarantee fast intervention to population centres.

2.1.4 Efficiency criteria

An essential part in the formulation of an LA problem is to identify an efficiency measure [51; 54] of the interaction occurring among the locations where the facilities are positioned and the customers using the service, to provide a tool for driving the location process towards a satisfactory result concerning many different objectives. The quality of the interactions is considered to be directly related to the relative spatial position of the interacting points (namely, customers and facility locations) and can be a measure of the distance between the customer and the facility or the actual cost incurred for servicing a customer. Most work presented in the literature usually express the measure of efficiency by some notion of distance. Many different distance measures may be of interest depending on the application, and the study and choice of adequate distance concepts have almost become a research field in its own right. The definition of distance measures represents the first step towards the specification of different efficiency criteria, which can be built by converting estimated distances into appropriate costs. For instance, distances can be adjusted to reflect travel or response times, by including factors such as physical or social barriers to travel, congestion and road conditions. In some cases, distance-related elements do not appear directly as objectives but might be necessary to capture some particular aspects of the problem in the form of additional constraints.

Below we describe some distance measures which have been extensively analysed and used in LA problem literature to approximate distances between two spatial coordinates. For the sake of simplicity, we denote the coordinates of two points p and q by (p_1, p_2) and (q_1, q_2) which identifies the position of the customer and the facility location for which we want to measure the distance.

The most familiar and widely used distance measure is the *straight-line* or *euclidean* measure, denoted by $l_2(x, y)$. It is derived from the euclidean norm and can be mathematically written as:

$$f(p, q) = ((p_1 - q_1)^2 + (p_2 - q_2)^2)^{1/2} \tag{2.1}$$

Euclidean distance often applies when movement is allowed homogeneously in all directions. The second topper distance measure, $l_1(p, q)$, is the variously referred *rectangular*, *rectilinear*, *metropolitan*, or *manhattan* distance. Rectilinear distances derive from the rectangular norm and can be mathematically expressed as:

$$f(p, q) = |p_1 - q_1)| + |p_2 - q_2)| \tag{2.2}$$

2.2 Existing formulations of LA problem

Many real-world problems have been formulated as an LA problem in the literature by leveraging on the important elements of facilities, customers, location and efficiency criteria. The first of these formulations recorded in the literature is the work presented in [41] by Cooper Leon in 1963. Cooper's formulation involved locating two facilities m to service seven customer n demands. In order to formulate his problem as an LA problem, Cooper made some assumptions about the facilities, customers, location and efficiency criteria. Cooper assumed facilities to be uncapacitated, which meant that a single facility could fully service the demand of a customer. However, the location of the facilities was unknown and had to be determined by solving the LA problem. Cooper also assumed that the location and demand of each customer are known. The efficiency criteria employed by Cooper was the minimum euclidean distance between a facility and a customer multiplied by the service cost of servicing a customer. Hence, in his formulation cost is proportional to distance. The decision variables were represented by a binary string x $= \{x_1, ..., x_m\} \in \{0, 1\}^m$ where 1 represents an opened facility and 0 represent a closed facility. The objective of the formulation was to minimise the total cost of establishing m facilities and the cost of supplying n customers with m facilities. Cooper formulated his problem as:

$$f(x) = \sum_{i=1}^{m} \sum_{j=1}^{n} x_i c_i + w_{ij} d(x_i, a_j) \tag{2.3}$$

subject to

$$\sum_{i=1}^{m} w_{ij} = r_j, \quad j = 1, 2.....n \tag{2.4}$$

$$w_{ij} \geq 0, \quad i = 1, 2...., m \qquad j = 1, 2...., n \tag{2.5}$$

where m: is the number of facilities; n: is the number of customers; $x_i = 1$ if facility is opened and 0 otherwise; c_i: is the cost of opening facility i; w_{ij}: is the quantity of goods supplied to customer j by facility i; $d(x_i, a_j)$: is the euclidean distance between customer j and facility i; a_j: is the location of customer j; r_j: is the demand of customer j. Equation (2.3) minimises the total cost of establishing facilities and the cost of supplying n customers with m facilities. Equation (2.4) ensures that all customer demand is satisfied. Since there exist no capacity restrictions on the facilities, an optimal solution will have the requirement of each customer serviced by the facility that is the nearest to it. Constraint (2.5) is ensures that the quantity of goods supplied to a customer is not less than 0.

Over the years, many real-world problems within the public and private sectors have been formulated as an LA problem. Below we explore some of the different formulations presented in the literature.

2.2.1 Emergency response

Verma and Gaukler in [166] studied the optimal placement of disaster response facilities that will be used to pre-position emergency supplies such as food, medicine, potable water, medical equipment's, generators and tents in the event of natural disasters such as earthquakes, floods, and large scale fires or non-natural events such as terrorist attacks. They formulated an LA problem that explicitly considered the impact a disaster could have on disaster response facilities and population centres in the surrounding areas in California. The authors used the response time from a facility to an affected population centre defined by the euclidean distance, which is proportional to transportation costs as the efficiency criteria. The objective, therefore, is to minimise the total transportation cost from facilities to affected population centres. The decision variables were represented by a binary string x = $\{x_1, ..., x_m\} \in \{0, 1\}^m$ where 1 represents an opened facility and 0 represent a closed facility. Similar works to Verma and Gaukler in field of emergency response includes [14; 29; 31; 73; 83; 86; 111; 138; 146; 152; 173].

2.2.2 Utility allocation

Patel in [133] studied the problem of locating 45 social service centres m within the Dharampur region of India to provide agricultural extension, primary schools, public health centres, cooperative service societies, fair-price shops and post offices to 237 villages n. The objective of the study was to minimise the total distance of accessibility of the service centres from villages. Due to budget constraint, the total costs of establishing all service centres were not to exceed Rs. 1.4 million. Also, the maximum distance of any village from a service centre was not to exceed the maximum allowed distance D. Patel used the Euclidean distance as an efficiency criterion, however, to compensate for the terrain in Dharampur a factor of 1.5 was applied to the euclidean distance between a service centre and a village. The decision variables were represented by a binary string x = $\{x_1, ..., x_m\} \in \{0, 1\}^m$ where 1 represents an opened service centre and 0 represent a closed service center.

The work presented in [79] by Ali et al. aimed to determine the optimal site for the parking facilities of a Steel company and designate travels connecting departments to all parking facility. The objective of their work seeks to minimise the total cost of the system, including the costs of equipment, maintenance and operation of the facilities plus the cost of travelled distance in the company. The cost of travelled distance included the cost of fuel exhaustion, driver's salaries and the cost of vehicle deprecation. The decision variables were represented by a binary string x = $\{x_1, ..., x_m\} \in \{0, 1\}^m$ where 1 represents a chosen parking facility and 0 otherwise. Similar research works to Ali et.al. were conducted in [75; 120; 168].

2.2.3 Administration

In work by Lolonis et al. in [112], the authors formulate an LA problem to group a collection of basic spatial units (US counties) into bordering administrative regions, so that the supplies of services are efficient, and the regions are uniform guaranteeing that specific types of services are rendered to beneficiaries at an affordable cost. The uniformity of the regions improves the spatial uniqueness of regions, which facilitates developmental plans. Since the cost of providing services increases with the distance separating a demand point from the assigned centre, they employ distance as an efficiency criterion. However because the allocation of a demand point to a facility depends on its similarity to the centre concerning selected socio-economic and physical characteristics, the authors assume that spatial difference has an interference consequence that artificially raises the cost of designating demand to regional centres. The measure of this increase is decided by the Coefficient of Spatial Differentiation (CSD), which is a rigidly growing function of the difference of two demand points. Hence the product of CSD and the actual distance separating demand point j and centre i denotes the total cost of allotting a demand unit j to centre i and includes both the cost of spatial separation and the impact of dissimilarity as well. The product is called Transformed distance (TD) of demand point j and centre i and is used as the criterion for allocating demand points to centres. Hence a demand point is allocated to the centre that is closer in terms of Transformation Distance. The decision variables were represented by a binary string $\mathrm{x} = \{x_1, ..., x_n\} \in \{0,1\}^n$ where 1 represents a selected administrative region and 0 otherwise. Similar works to Lolonis et.al. in the area of administrative LA problem are conducted in [6; 21; 124; 130; 151; 174].

2.2.4 Healthcare

In recent works, Shariff et al. in [159], presented an LA problem formulation which was used to study and address the location of healthcare facilities of one of the districts in Malasia. The objective of the study was to maximise the population assigned to a facility within a coverage distance. The decision variables were represented by a binary string $\mathrm{x} = \{x_1, ..., x_m\} \in \{0,1\}^m$ where 1 represents an opened facility and 0 represent a closed facility. Similar research works to Shariff et al. problem has been conducted in [15; 36; 37; 90; 129; 137].

2.2.5 Agriculture

Tong et al. [164] examined the problem of determining farmers' markets in Tucson, Arizona, state. Farmers' market is established locations where farmers periodically gather, to peddle their farm produce. Given that many farmers' market operates a few hours a week, the problem formulation considers not only locations but also time frames. By taking into considerations the different travel patterns and work schedule of customers, the variations of travel distances with the time of day when customers originate their trips from non-home places can be devised to ensure the effective use of the farmers' market. The objective of the authors' work was to minimise the overall additional travel accrued by visiting farmer's markets. The decision variables were represented by a binary string x = $\{x_1, ..., x_m\} \in \{0, 1\}^m$ where 1 represents a chosen farmers' market site and 0 otherwise. Similar LA formulations in the area of agriculture include [20; 34; 171].

2.2.6 Education

Neema et al. formulated an LA problem in [128] to find the optimal location of high schools and the allocation of students within a sub-district of Bangladesh. The objective of their work was to minimise the total travelled distance of students from home to high schools. The decision variables were represented by a binary string x = $\{x_1, ..., x_m\} \in \{0, 1\}^m$ where 1 represents an opened high school and 0 otherwise. Similar research works to Neema et. el. problem have been conducted in [76; 123; 126].

2.2.7 Energy

Bojic et al. in [19] formulated an LA problem to solve the problem of locating solid biomass power plants for Vojvodina, an agricultural, energy-deficient province of Serbia. Several factors were considered when locating biomass plants in Vojvodina; the plant had to be located close enough to biomass source to reduce the transportation cost but also close enough to urban centres to ensure competitive pricing for delivering electricity to customers. The type of the plant also affected the decision to locate a biomass plant, the bigger the capacity of the plant, the more expensive it was to operate. Also, high operation costs affected the cost of service rendered. The objective of the authors, therefore, was to ascertain the capacities, type and locations of solid biomass power plants that generate minimal electricity costs, for areas with known resources and targeted electricity generation. The decision variables were represented by a binary string x = $\{x_1^t, ..., x_m^t\} \in \{0, 1\}^m$ where 1 represents an open facility with plant type t and 0 otherwise. Similar research work within this area are presented in [98; 135; 141; 143].

Although formulations of real-world problems as LA problems are interesting to study topics, they, however, do not capture many of the features of real-world problems. The fundamental characteristics of LA problem require that any rational formulation reflect some aspects of future changes. Changes in population growth and migration often drive the need of consumers. For this reason, facilities are expected to be effective in servicing demand over an extended planning horizon, especially in cases where considerable capital and resource investment is required in locating a facility such as a telecommunication infrastructure. It, therefore, makes sense to plan the location of facilities with consideration to the varying time aspect of the problem [131]. Taking the varying time aspect of the problem into consideration when planning the location of facilities will ensure that facilities are not only ideal for current conditions but also stay useful over a defined time horizon. We are, therefore, motivated to explore dynamic formulations of LA problem in section 2.3.

2.3 Dynamic LA formulations in the literature

In the literature, the dynamic formulations of LA problem are mainly categorised into Explicit and Implicit dynamic models [52]. In explicit dynamic formulations, facilities will be opened and possibly closed over a defined period, also known as multi-period. They include other factors such as relocation time, number of relocations and number of facilities to be relocated. This categorisation of LA problem falls in the domain of dynamic optimisation. Implicit dynamic formulations concern selecting profitable facilities locations to be opened once at the start and remain open over a defined time. Implicit dynamic formulations are dynamic because they recognise that problem parameters such as demand may vary across time and endeavour to anticipate these changes in the facility location scheme generated. This implicit dynamic formulations of LA problem falls in the domain of Robust Optimisation Over Time (ROOT). Section 2.3.1 highlights the main difference between dynamic optimisation problems and robust optimisation overtime problems.

2.3.1 Dynamic Optimisation problems vs Robust Optimisation over time problems (ROOT)

In a Dynamic optimisation problem (DOP), the fitness functions of the optimisation change over time, resulting in the global optima to change as well [22; 92]. In *DOPs*, it is presumed that the decision-maker has to plan and implement a solution each time the environment changes. For *DOPs* the objective of the current time is to obtain an optimum solution in terms of fitness for the current environment and then relocate a new optimum solution in terms of fitness for the new environment once the environment shifts or changes. In dynamic optimisation problems, the fitness function is deterministic at each time instant, but dependent on time t, i.e.

$$f(\overline{x}, \overline{\alpha}(t)) \tag{2.6}$$

where $\overline{x}$ describes the configuration parameters, $\overline{\alpha}(t)$ denotes time-dependent problem parameters. The most basic aim in solving dynamic optimisation problems is to pursue the changing optima over time. The formalisation of *DOPs* is logical in circumstances where implementing a solution, e.g. establishing or relocating new facilities, can be completed promptly and cheaply. However, in circumstances where the implementation of a solution requires features of human operation and high costs in establishing facilities, it will be highly capital intensive and sometimes infeasible to execute a new solution each time the environment varies. Taking a practical dynamic LA problem, for example, the environmental states, e.g. location of customers vary from year to year. For this kind of problem, if we implement a new solution every year by building new facilities to service changing customer locations, it will incur high financial risks. Therefore, in such circumstance, it would be more cost-effective having a fixed solution implemented and used for a long-time period than implementing a new solution each time the environment varies as it is done in DOPs. This process of finding and implementing a fixed solution to a changing problem over time is known as ROOT.

In ROOT, we aim to obtain solutions that are reliable or robust over the defined time, rather than pursuing the shifting optima [63]. A solution is described as robust over a specific period when its quality continues to be satisfactory and is relatively indifferent to the environmental fluctuations during the defined time interval. An obtained solution that is robust over time will be employed until its quality diminishes to an unacceptable level in the present environment. A new robust solution should be obtained when the solution quality diminishes to an undesirable state. Assuming a solution x is determined at time t. We define the robustness of a solution during a period as:

$$F^a(x,t,T) = \frac{1}{t}\int_t^{t+T} f(x,\alpha(i))di \tag{2.7}$$

where x signifies the decision variables, i.e. the solution; T is a user-specified parameter asserting the duration a solution is employed; α defines the environmental condition that stipulates the fitness function f. The environment condition at period i is expressed as $\alpha(i)$.

To the best of our knowledge almost all LA problem formulations studied in the literature fall in the domain of dynamic optimisation problem except for work presented by Daskin et al. in [45]. When formulating dynamic LA problem, two processes are mainly considered in the literature for predicting potential changes. These are forecasting and scenario planning.

2.3.2 Forecasting vs Scenario planning

Forecasting [156] concerns prognosticating the future as precisely as feasible, given all the information accessible, including past data and information of any forthcoming events that might influence the predictions. Forecasting predicts potential changes in one future. Forecasting models are most favourable in instances with low uncertainty and low complexity because it is defensible to make numerical predictions about the variable of concern without widely conditioning the forecast on crucial assumptions about other variables that may change it. However, forecast models often fail to prognosticate quick vital shifts in conditions. A limitation of using forecasting to evaluate risk is that in dynamic conditions where there exist high uncertainty and high complexity, forecast models usually break down because they are encountering new events that are not represented in their test statistics. Works in the literature that employs forecasting for predicting future changes include [16; 26; 45; 64; 106; 154; 158; 177].

Scenario planning [167], on the other hand, is a strategic planning method used to conceive flexible continued term strategies for the future by considering the plausibility and possibility that discusses the weaknesses of forecasting. In scenario planning, many scenarios of how the future might unfold are generated in considerable detail. Predetermined elements, common to the scenario, are combined with critical uncertainties, that vary between the scenarios. In many cases of scenario planning, the goal is to choose strategies that are robust to all scenarios. A scenario is a rich data-driven story concerning the future that can support organisations more reliable decision-making now where hypothesis shows a variety of likelihoods for the future. Works in the literature that employs scenario planning for predicting future changes include [12; 66; 110].

2.4 Existing dynamic formulation of LA problem

To capture many of the dynamic aspects of real-world problems, varying dynamic formulations of LA problem has been presented in the literature. A generalised formulation of the dynamic LA problem was first introduced in the literature by Wesolowsky and Truscott [169]. In their formulation, they aimed to devise a plan of optimal locations and relocations in response to predicted changes in the demand volume originating at demand points over a planning period T. Wesolowsky and Truscott formulated the dynamic LA problem as:

$$f(x) = \sum_{t=1}^{T}\sum_{i=1}^{m}\sum_{j=1}^{n} a_{ijt}y_{ijt} + \sum_{k=2}^{T}\sum_{i=1}^{m}(c_{it}x_{it} + c'_{it}x'_{it}) \tag{2.8}$$

Subject to:

$$\sum_{i=1}^{m} y_{ijt} = 1 \quad \forall j, t, \tag{2.9}$$

$$\sum_{j=1}^{n} y_{ijt} \leq N x_{iit} \quad \forall i, t, \tag{2.10}$$

$$\sum_{i=1}^{m} y_{ijt} = G \quad \forall t, \tag{2.11}$$

$$\sum_{i=1}^{m} x_{it} \leq m_t \quad \forall t \geq 2, \tag{2.12}$$

$$y_{iit} - y_{iit-1} + x_{it} - x_{it}^{'} = 0 \quad \forall i, t \geq 2, \tag{2.13}$$

$$y_{ijt} \geq 0 \quad \forall j \neq i \quad x_{it}, x_{it}^{'} \geq 0, \quad \forall i, t \qquad y_{iit} \in 0, 1, \quad \forall i, t. \tag{2.14}$$

where a_{ijt}: is the present value of the cost of assigning demand point j to facility i in period t; c_{it}: is the present value of the cost of removing a facility from site i in period t; $c_{it}^{'}$: is the present value of the cost of establishing a facility at site i in period t; m_t: the maximum number of facility location changes allowed in period t; y_{ijt}: is 1 if demand point j is assigned to facility location i in period t and 0 otherwise; x_{it}: is 1 if facility is established at site i in period t and 0 otherwise; $x_{it}^{'}$: is 1 if a facility is removed from site i in period t and 0 otherwise; G: is the total number of facilities that can be established in period k; m: number of facility locations; n: number of demand points. The objective function is to minimize the costs of distribution from the facilities to the demand points. Based on Equation (2.9), each demand point j is assigned to exactly one facility location i . In Equation (2.10) demand point j can be assigned to facility location i only if i is self assigned. Equation (2.11) guarantee that G self-assignments are made among the m locations. Equation (2.12) limit the number of sites vacated in each of periods *2* through T. Since constant number of facilities, G, is required in all periods, placing an upper bound on the number of facility removals in a period is equivalent to limiting the number of facility location changes in the period. Equation (2.13) in conjunction with the second term of Equation (2.8) ensure that the appropriate relocation costs are charged. The required minimisation of costs forces the following binary values of x_{it} and $x_{it}^{'}$ for each possible combination of values for y_{iit} and $y_{ii,t-1}$ in (2.14).

In succeeding sections, we examine the dynamic LA formulations of some real-world scenarios.

2.4.1 Manufacturing/Production

Canel and Khumawala [26] proposed a multi-period LA problem aimed at locating facilities across different countries. The model seeks to decide which countries to establish manufacturing facilities, quantities of goods to be manufactured and the amounts to be dispatched from the facilities to the customers. The objective function maximises profits rather than minimising costs in order to capture the different prices in different countries. The problem formulation assumes the demand of customers to vary over time based on the growth rate of the market. The demand of customers is forecasted over the future time horizon. The decision variables were represented by a binary string x = $\{x_1, ..., x_m\} \in \{0, 1\}^m$ where 1 represents an opened facility at time t and 0 otherwise. Similar work in this field includes [38; 39; 74; 81]

2.4.2 Emergency Supply

Gama et al. in [64] present a multi-period LA problem for self-evacuation towards shelter sites in Wake County, North Carolina, USA. Shelters here are facilities in which evacuees can find health assistance, food and safety. The proposed LA formulation considers shelter location, warning signals dissemination, and evacuation routing decisions under flood forecast. The study aims to optimally identify opening times and locations for shelter sites, timings for evacuation order dissemination, and optimal evacuees-to-shelter allocation while minimising the total travelling time between evacuation zones and shelter destinations. The decision variables were represented by a binary string $\mathrm{x} = \{x_1, ..., x_m\} \in \{0, 1\}^m$ where 1 represents an opened shelter at time t and 0 otherwise. Similar research work in this area includes [18; 23; 50; 109; 178]

2.4.3 Healthcare

Benneyan et al. presents a multi-period LA problem in [16] that considers the location of speciality care clinics for veteran Administration (VA) health in New England, United States. The LA problem seeks to minimise the overall total costs over a defined period relative to geographic demand by determining the optimal sleep bed capacity for each open clinic concerning the maximum acceptable travel distance of a patient. Forecast changes in geographic demand pattern are assumed for the formulation. The decision variables are represented by a binary string $\mathrm{x} = \{x_1, ..., x_m\} \in \{0, 1\}^m$ where 1 represents an opened clinic at time t and 0 otherwise. Similar in this area includes [43; 160; 165]

2.4.4 Environmental

Whiles multi-period LA problem has received much attention from the research community it is important to note that the location and relocation of facilities due to shifting in population over a defined period can be very expensive and sometimes not feasible especially in the case of telecommunication where opening and running facilities incur considerable cost and risk. For this reason, it is best to open facilities once at the start and remain open over the defined period. These LA formulations are termed as Implicit LA problem belonging to robust optimisation problems (ROOT) [175]. To the best of our knowledge, the only paper within the context of dynamic LA problem, in which focus is placed on formulating and solving implicit LA problem is the one by Daskin et al. [45].

Daskin et al. acknowledge that the challenge in tackling dynamic LA problems stems from the uncertainty encompassing future circumstances. The authors contend that the most reliable way to handle uncertainty is to delay decision making as long as feasible, accumulating information and advancing forecasts as time progresses. The study, therefore, strives to find an optimal or near-optimal first-period location of facilities over an infinite period.

2.5 Exacts, Heuristics and Meta-heuristics approaches to LA problem in the literature

Different solution approaches involving exact methods, heuristics and meta-heuristics have been proposed and used to solve different formulations of the LA problem. Exact methods [139] use optimisation mathematical models are guaranteed to find the optimal solution for the problem by evaluating all possible solutions. LA problem is considered as an NP-hard problem, i.e. the computation time required to solve them increases as the size of the problem increases. Hence, the larger the problem, the more complex the solution space which in-turn can make exact methods slower. Heuristics and meta-heuristics are techniques designed for solving a problem more quickly when exact methods are too slow, or for finding an approximate or near-optimal solution when exact methods fail to find the optimal solution. While heuristic algorithms are problem-specific meta-heuristics, have an independent problem structure consisting of components which exploit problem-related information. Meta-heuristics start with a set of initial solutions, generated either randomly or by exploiting some information about the problem at hand. This set of solutions is iteratively improved by applying a set of operations on the solutions in the set, such as combining two solutions or searching for a better neighbour of a solution in the set for a pre-specified number of iterations.

Since almost all formulations of the LA problem are NP-hard, exact methods are limited. Hence most LA problem formulations are solved through heuristics and meta-heuristics. A look through the literature shows a significant amount of research on meta-heuristic based solution approaches for solving LA problem formulations in the last decade. In the succeeding Sections, we examine some of the most widely used exact methods, heuristics and meta-heuristics for solving the LA problem. We also explore the Population-Based Incremental Learning Algorithm (PBIL) [9], which has been reported in the literature to be effective in tackling optimisation problems. To the best of our knowledge, PBIL has not been used in the literature to tackle LA problem. We are motivated to explore the potential benefits of PBIL (i.e. a small number of parameters and the probabilistic model that reveals much information about the problem being solved but with a lightweight modelling cost) and how it can be applied to solve the LA problem.

2.5.1 Exact methods

Ceselli [30] presented two exact algorithms, branch-and-bound and branch-and-price technique for solving the LA problem. In this algorithm, the Lagrangian relaxation and subgradient optimization are utilized for branch-and-bound technique and column generation is used for the branch-and-price algorithm. The author analysed and compared some performance details, showed how a fine-tuning could improve the performances of their technique.

Canos et al. .[27] proposed an exact algorithm for the LA problem. They considered a set of restrictions such that the decision-maker can select partially feasible solutions which partially cover the demands as they extensively decrease the cost. The researchers proposed an alternative enumeration algorithm for solving the problem, which is based on Hakimi's seminal papers [77; 78], and it was appropriate when the number of vertices is not too large. Neebe [127] considered a branch and bound algorithm for the LA problem and for providing lower bound the Lagrangian relaxation is used.

Jrvinen et al. [89] constructed a branch-and-bound algorithm for the LA problem. The authors showed how the vertex-substitution technique could lead to premature convergence and consequently local optimum, and give a heuristic method for finding a good initial solution for this technique. In addition, they studied four techniques, namely, branch-and-bound, branch-and-bound without backtracking, substitution so that the initial solution was formed by the first p vertex heuristic, and substitution with an initial heuristic solution.

The Branch-and-bound algorithms developed for the LA problem by Kuenne and Soland [103] and Ostresh [145] used very small instances, of the order of customers $n = 15$, facilities $m = 4$ and $n = 50$, $m = 3$. Rosing [144] is able to incorporate improvements in the methodology that allow problems with $n = 30$, $m = 5$ and $n = 25$, $m = 6$ to be solved exactly.

2.5.2 Heuristic methods

Taillard [162] utilized a clustering technique to solve the LA problem. They applied a candidate list search (CLS), local optimization (LOPT) and decomposition/recombination (DEC). The CLS started with an alternate heuristic technique introduced by Maranzana [113] and obtained a locally optimal solution. Regarding the solutions, in the CLS clustering technique, the solution is found by eliminating a vertex and adding another, similar to vertex substitution. In this case, the new solution is selected only if it is better than the initial one. The author utilized the LOPT technique in DEC clustering method for finding a good solution in the overall problem.

Resende and Werneck [142] introduced a multi-start hybrid heuristic that combined elements of several metaheuristics as Greedy Randomized Adaptive Search Procedure (GRASP). In this process for each generation, a greedy randomized algorithm is applied by a local search technique. The author utilized the idea of path-relinking from tabu search and scatter search for storing a group of the best solutions of the previous generation. This algorithm was useful from strategies that improve diversity: selecting solutions from the pool in a biased way, returning a local minimum in the path if no improving solution is found, and applying local search to the solution returned.

In [170] Whitaker develops a Greedy heuristic starts with an empty set of open facilities. Facilities are then added one by one until the total number of facilities is reached; each time the location which most reduces total cost is selected.

In[56] Feldman et al., develops a Stingy heuristic, also known as Drop or Greedy-Drop. The heuristic starts with all m facilities opened, and then removes them one by one until the number of facilities has been reduced to the maximum number of facilities required to be opened; each time the location which least cost of opening a facility is selected. Salhi and Atkinson [149] modified the implementation of the stingy heuristic to start from a subset instead of the entire set of potential sites.

In the Greedy heuristic developed by Captivo [28], the Alternate procedure is run for each step. A combination of Alternate and Interchange heuristics has was suggested in [136] by Pizzolato. In [121], a variant of Stingy (or Greedy-Drop) is compared with Greedy combined with Alternate and Multi start Alternate. In [147] perturbation heuristic, Stingy and Greedy is run one after another, each having a given number of steps. The search allows exploration of infeasible regions by oscillating around feasibility. The combination of Greedy and Interchange, where the Greedy solution is chosen as the initial one for Interchange, has been most often used for comparison with other newly proposed methods in the LA problem literature.

In the alternate heuristic developed by Maranzana [113], facilities are located at location points, and users are assigned to the closest facility. Each facility is selected evaluated according to each facilitys set of users. Then the procedure is iterated with the new locations of the facilities until no more changes in assignments occur. Since the iterations consist of alternately locating the facilities and then allocating users to them, this method is referred to as the alternating heuristic. The heuristic may switch to an exhaustive exact method if all possible subsets of locations are chosen as an initial solution. However, this is not usually the case since the complexity of the algorithm is then increased by an O(mp).

A heuristic that uses a dynamic programming idea is suggested by Hribar and Daskin [85]. It may be viewed as reduced dynamic programming or as an extended greedy constructive method. Instead of considering only the best facility as in Greedy, the q best solutions are stored in each iteration (q is a parameter). The procedure stops when the required maximum of opened facilities are reached, as in Greedy. This heuristic was tested using three small data-sets of size $m = n = 49$, 55, and 88.

2.5.3 Meta-Heuristic methods

Genetic algorithm (GA) is one of the most widely used population-based meta-heuristics applied to solve LA problems. The earliest application to solve an LA problem with a GA was by Hosage and Goodchild in [84]. Other works that used

GA to tackle LA problems include but not limited to [72; 87; 117; 128; 150]. In their solution approach using a GA, authors either generated initial solutions randomly or employed a heuristic approach. The most commonly used of the two was random initialisation. Authors also fixed the size of the population to a fixed number for the whole run of the algorithm irrespective of the size of the problem instance [32; 42; 102; 161] or the size of the population was set a function of the problem size [2; 88; 91]. The solution approaches also differed in their choice of crossover and mutation operators. The most commonly used crossover operators were the One-point and uniform crossover operators. The most commonly used mutation parameter was the bit-flip or bit reversal operator. On the whole crossover probability were much higher than the mutation operators with some work recording as high as 0.99 [44] for crossover probability. Typically the mutation probability value changed between 0.005 and 0.8 with most papers taking shallow values of between 0.01 to 0.2. Concerning termination criteria, three main criteria were employed in the literature: (1) the maximum number of generations, (2) the maximum number of generations without improvement in the best solution value and (3) the execution time. The most commonly used termination criteria were the maximum number of generations; however, some works combined two or more of the criteria to terminate the algorithm. GAs was recorded in the literature to offer better results when compared to other methods for solving LA problem presented in the literature [87; 128; 163] on the 15 Uncapacitated Warehouse data-sets from the OR library [13].

Tabu search (TS) is among the widely used local search meta-heuristics for tackling LA problem. Works that employed tabu search to solve LA problem include [24; 71; 148]. TS begins with the first solution to the problem and hunts for the fittest solution in a suitably distinguished neighbourhood of the solution. Amongst the regularly used approaches for creating the first solution are random initialisation [1; 7; 119], greedy approach [33; 46; 67] and heuristic approach [25; 148]. TS then assigns the best solution in the neighbourhood as the current solution and begins the exploration process anew. TS ends the search if specified stopping criteria, either concerning execution time, solution quality or both have been met [69]. During the exploration process, TS retains the current best solution, and the best solution discovered so far. In order to check TS from evaluating solutions that it has already evaluated in previous iterations, TS keeps a record of neighbour generation moves it deems forbidden, or tabu and eliminates solutions that can be attained only by tabu moves from the neighbourhood. Once a move joins the list of tabu moves, it tarries there for several consecutive iterations. The list of tabu moves alternates continuously throughout the execution of the search, making tabu search an adaptive memory search algorithm [69]. More details about TS and its components can be found in the original paper by Glover [70]. For solving LA problem by a simple probabilistic TS, good results on Kochetov test instances [97] are reported in [71].

Scatter search is a population-based meta-heuristic originally proposed by Fred Glover [69]. Works that employed scatter search to solve LA problem include [40; 48; 49; 94; 95; 132; 155]. Scatter search keeps a set of solutions termed as the reference set, which is initially generated from a seed solution. In generating the initial solutions, some works used a construction heuristic to ensure diversity [155] whiles others used randomly generated seed [94], and others used a combination of greedy heuristic [95]. After generating the initial solution, Scatter search iteratively combines solutions in the reference set to create new solutions which are in turn employed to update the reference set. The process ends when either the set of solutions does not change or after a pre-specified time limit or after a pre-specified number of iterations [69]. Although the structure of scatter search resembles that of GA, the fundamental principles on which the two methods work are significantly different. Unlike GA, scatter search utilises a much smaller reference set and employs more precise methods to guarantee diversity and coverage of the solution space. Furthermore, while in GA, solutions are selected at random for recombination, scatter search adopts a precise and exhaustive scheme to select solutions for combination. Scatter search also employs intensification procedures such as local search and tabu search to improve upon each solution created from the combination process. Most implementation of scatter search for solving LA problem in the literature used local search with different neighborhood structures as improvement methods [49; 132] such as swap neighborhood [95] and shift neighborhood [40; 48]. Good results are reported on TSP-Lib instances [140].

A basic Simulated annealing (SA) heuristic for LA problem was proposed in Murray and Church [125]. The SA heuristic proposed by Chiyoshi and Galvo in [35] combines elements of the vertex substitution method with the general methodology of simulated annealing. The cooling schedule adopted incorporates the notion of temperature adjustments rather than just temperature reductions. Computational results are given for OR-Library test instances [13]. Optimal solutions were found for 26 of the 40 problems tested. Recently, an SA heuristic that uses the 1-interchange neighbourhood structure has been proposed by Levanova and Loresh in [107]. Results of good quality are reported on Kochetov data sets [97], and on the first 20 (among 40) OR-Library [13] test instances. For example, 17 out of the 20 OR-Library instances are solved exactly.

Variable neighbourhood search (VNS). There are several papers that use VNS [80; 121; 122] for solving the LA problem. In the first one [122], a basic VNS is applied and extensive statistical analysis of various strategies performed. Neighbourhood structures are defined by moving $1, 2, , k_{max}$ facilities and correspond to sets of 01 vector at Hamming distance $2, 4, , 2k_{max}$ from x. The descent heuristic used is 1interchange, with the efficient, fast interchange (FI) computational scheme. Results of a comparison of heuristics for OR-Library [13] and some TSP-Lib [140] problems are reported. In order to solve larger LA problem instances, in [80], both reduced VNS and a decomposition variant of VNS (VNDS) are applied. Sub-problems with increasing numbers of users (that are solved by VNS) are obtained by merging subsets of customers. Results on instances of 1400, 3038 and 5934 users from the TSP library [140] show that VNDS improves notably upon VNS in less computing time, and gives much better results than FI, in the same time that FI takes for a single descent. Moreover, reduced VNS, which does not use a descent phase, gives results similar to those of FI in much less computing time.

Among the recent popular meta-heuristics for tackling optimisation problems are Estimation Distribution Algorithms (EDAs) particularly Population-based incremental (PBIL) learning algorithm, which has been recorded in the literature to lead to better results than a standard GA in many optimisation problems [59]. PBIL also allows for problem encoding, which makes it suitable for tackling problems where prior knowledge of the problem can be encoded in the algorithm to aid in finding an optimal or near-optimal solution. Therefore, we proposed to explore the effectiveness of PBIL in solving the LA problem. PBIL is discussed into detail in section 2.5.4. To the best of our knowledge, this is potentially the first time PBIL has been applied to the LA problem.

2.5.4 Population-Based Incremental learning Algorithm (PBIL)

Population-based Incremental Learning (PBIL) Algorithm is a simple Estimation of Distribution Algorithm (EDA). EDAs are stochastic optimisation algorithms that explore the space of candidate solutions by sampling an explicit probabilistic model constructed from promising solutions found so far [82]. PBIL originally proposed by Baluja [9] [11] is an optimisation algorithm that integrates the characteristics of a GA efficiently with those of competitive learning. This combination results in a tool that has proven through experiments to be simpler than a GA and sometimes transcends the performance of a GA (in terms of speed and precision) on a broad range of optimisation problems [93].

PBIL strives at extricating the population statistics instead of keeping a significant number of samples. Population-based Incremental Learning initialises a probability vector (PV) that acts as a model for high evaluation solutions. Through the probability vector, the succeeding population for the next generation is created. An evident characteristic of this model is that it demands less memory and executes faster than a traditional GA.

PBIL employs the binary encoding scheme to create the probability vector. In this bit string representation, the likelihood of each bit position containing a 1 is specified. The probability that a bit position contains a 0 can be determined by subtracting the probability in the vector from 1. Based on these probabilities, the members of a population can be extracted. It should be mentioned here that the diversity of the population depends on the probability values in the probability vector. A probability vector where the value in each bit position is assigned to 0.5 introduces the most diversity. In such a vector, the generation of 1 or 0 in each bit location is entirely random.

The learning rate in PBIL directly affects how fast the probability vector gets altered to resemble a classified point [9] correctly. As the population samples are created using the probability vector, the learning rate influences which section of the search space is searched. The learning rate has a primary impact on the trade-off amid exploration and exploitation of the search space. Exploration describes the capability of the algorithm to scour the search space while exploitation describes the capability of the algorithm to efficiently use the obtained knowledge to decrease and focus the future search. For instance, if the learning rate is set as 0, there will be no exploitation of knowledge obtained during the search. As the learning rate increases, exploitation also increases, and the capacity to explore a vast portion of the search space is reduced. In other words, the higher the learning rate, the quicker the algorithm focuses on the search. The lower the learning rate, the more exploration takes place. To evolve the PV in PBIL a truncation size is set which determines the number of solutions to be discarded from the population. The truncation size is multiplied by the population size to determine the number of solutions to be eliminated from the population, i.e. a truncation size of 0.5 means half of the population will be discarded, and the remaining solutions are used to evolve the PV.

In our LA problem the decision variables or solutions in PBIL are represented by a binary string x = $\{x_1, ..., x_m\} \in \{0, 1\}^m$ where 1 represents an opened facility and 0 represent a closed facility. A Pseudocode of how PBIL is implemented to tackle our LA problem is presented in Algorithm 1.

Algorithm 1 Population-based Incremental learning Algorithm pseudo-code

```
$P$: population
$n$: number of solutions in population $P$
$l$: length of solution
$pv$: probability vector
$nv$: number of solutions used in updating the $pv$
$LR$: Learning rate
1. Initialise probability vector $pv$
for $i$ := 1 to $l$ do
  $pv[i]$ = 0.5;
end for
2. Generate $n$ solutions from $pv$
for $i$ := 1 to $n$ do
  $P[i]$ = generate($pv$);
end for
3. Evaluate the solutions in $P$
for $i$ := 1 to $n$ do
  evaluate($P[i]$);
end for
4. Sort solutions in $P$ according to fitness
5. Remove worst half of the solutions from population $P$ to get $nv$
6. Update the probability vector $pv$
for $i$ := 1 to $l$ do
  double sum = 0;
  for $j$ := 1 to $nv$ do
    sum += $P$[j][i];
  end for
  $pv[i]$ := ($LR$ * (sum/n)) + $pv$[i] * (1.0 - $LR$)
end for
for $j$ := 1 to $nv$ do
  for $i$ := 1 to $l$ do
    $pv[i]$ := $pv[i]$* (1.0 - $LR$)+$P[j][i]$*($LR$);
  end for
end for
7. If termination condition is not reached return to step 2
8. Return the best solution from population $P$
```

2.6 Chapter Summary

In this chapter, we focused on understanding the concept of LA problem and how they have been formulated to capture many of the unique aspects of real-world scenarios presented in the literature. Our study showed that LA problems formulations are categorised into static formulations which only takes into account present conditions and dynamic formulations which aims to capture some aspect of future changes in customer demand. Our review of the literature showed that within the dynamic LA problem domain, two main categories exist. These are dynamic optimisation LA problem and robust optimisation over time (ROOT) LA problem. In dynamic optimisation LA problem, each time the environment changes, i.e. customer demand changes, new facilities will have to built or relocated in order to meet the needs of customers. This is often feasible in scenarios where the cost of building new facilities or relocating new facilities do not attract high facility and operational costs. In ROOT facilities are established once at the start of a defined period and are expected to be satisfactory in servicing the demands of customers over the defined period irrespective of how much the demand of customers change. ROOT often lends itself in situations where the cost of building facilities or relocating existing facilities comes with high financial and operational costs. Although many dynamic LA formulations have been presented in the literature, almost all of the dynamic formulations fall in the domain of dynamic optimisation problems. Only a single work in the literature presented an LA problem formulation with the context of ROOT.

In modelling future changes of customers demands, two methods are mostly used in the literature. These are forecasting and scenario planning. From the literature, forecasting was often used when there was enough historical data to predict the future with a level of certainty. Most dynamic formulations of LA problem reviewed in the literature review used forecasting as a mode of predicting future changes. Scenario planning, on the other hand, was used when there was a lot of uncertainty about the future and many scenarios of future changes had to be simulated in order to find a robust solution to the problem. A few of the research work surveyed in the literature review employed scenario planning in predicting future changes in customer demands and no work was found which employed scenario planning in the context of ROOT.

Another area we found lacking attention in the literature review was the aspect of resilience within the LA problem domain. Resilience is a very important aspect in real-world scenarios because it resolves the issue of downtime and ensures uninterrupted supply of demand. From the research work surveyed in the literature review, no formulation of LA problem captured the aspect of resilience in their formulation.

From the literature review, we identified three major areas that have received little to no attention from the LA problem research community, which forms the focus of our research. These are: (1) The aspect of resilience in LA problem formulation, (2) Formulation of dynamic LA problem in the context of ROOT and (3) Use of scenario planning for predicting future changes in customer movements in the context of ROOT.

In the literature review, we also explored the many ways LA problem formulations have been solved in the literature. We observed that exact methods, heuristic and meta-heuristic methods have all been employed within the literature. A surveyed of these methods showed that exact methods were often used for very small problems with instances of about three facilities and 50 customers. Heuristics and meta-heuristics proved to be more popular with meta-heuristics been the most widely used within the last decade as they were recorded to offer better results on larger LA problems. Considering that our research is focused on the large real-world instance, we are motivated to explore meta-heuristic solutions. Since our focus in this research is on problem modelling and not to develop a new algorithm to solve our problem. We are motivated to find a good enough algorithm presented in the literature to solve our LA problem. The most popular among the meta-heuristics methods surveyed in the literature review were Genetic Algorithms (GA), and hence in the succeeding Section, we hope to explore the effectiveness of GA to solve our problem. We are also motivated to explore the effectiveness of Population-Based incremental learning (PBIL) algorithm as it has been recorded in the literature to perform well on many optimisation problems. Our motivation for PBIL aside the recorded good performance on optimisation problems is based on the fact that PBIL has few parameters to configure with a univariate modelling cost. Also, PBIL allows for problem encoding when prior knowledge is known about the problem. To the best of our knowledge, this is potentially the first time PBIL will be employed to tackle an LA problem in the literature.

Chapter 3

GA Variants and PBIL for solving Real-World Location-Allocation Problems

In this Chapter, we present two novel non-linear models of LA problem motivated by a real-world problem from the telecommunication industry. The first formulation Location-Allocation Resilience Problem (LARP) is not restricted to the number of facilities that can be established to service customer demands. The second formulation; Location-Allocation Resilience Problem with Restriction (LARPR) is a constrained version of LARP having a limit on the number of facilities that can be established to service customers demands due to budget constraint. Both formulations capture an additional layer of resilience, where resilience is the backup connection provided to a customer to ensure uninterrupted bandwidth connection. Our focus in this research is primarily on problem modelling and not to develop a new algorithm to solve our problem. Therefore we surveyed the literature to find a good enough algorithm to solve our new LA problem formulations. From the literature review in Section 2, we observed that among the most successful algorithms applied to solve the LA problem are Genetic Algorithms (GAs). We also observed that the GAs employed, differed in their component and configuration choices. The GAs differed in the way they initialise solutions; the selection method they employ to choose parents to generate new offsprings; the crossover process they employ to recombine genes of parents to create new solutions and the mutation process they employ to maintain diversity within the population. In the hopes of designing an optimal GA to solve the new LA problem formulations, we are motivated to understand the contribution each GA configuration choice makes to the GA performance. To do this, we combine the different configuration choices to create new GA variants. We give a detail description of the different GA components in Section 3.3.

We are also motivated to explore Population-based Incremental Learning algorithm (PBIL) [9], which has been shown in the literature to be efficient in tackling challenging combinatorial problems in recent years [9; 58; 59; 118; 153]. Our motivation in selecting PBIL is twofold: (1) PBIL has the potential benefits of an Estimation distribution algorithm (EDA), i.e. a small number of parameters and employing a probabilistic model that reveals much information about the problem being solved but with a lightweight (uni-variate) modelling cost. (2) We observe that specifically for the novel LA problem formulations proposed in this Chapter, useful problem knowledge can be encoded directly into the probabilistic model. To the best of our knowledge, this is potentially the first time PBIL has been applied to the LA problem.

Our aim, therefore, in this Chapter, is to explore the effectiveness of GA variants and PBIL for solving the new LA problem formulations. The work presented in this Chapter has been published in the 2018 IEEE Congress on Evolutionary Computation (CEC) publications.

The rest of the Chapter are organised as follows: In Section 3.1 we present the problem background. Section 3.2.1 and Section 3.2.2 presents the formulation of LARP and LARPR respectively. Solution approach is presented in Section 3.3. Experimental setup is presented in Section 3.4 and discussion of results are presented in Section 3.5. We conclude the Chapter in Section 3.6

3.1 Problem background

In this Section, we extend the case study presented in Section 1.1 to help formulate the new LA problems. We consider a service company that needs to establish new facilities to service the bandwidth demand of its customers adequately. Customers are large corporate entities that require the highest quality of service. The quality of service a customer receives is determined by how close the customer is to an opened facility. A facility is assumed to be able to service all the demand of a customer adequately. Facilities are also assumed to be adaptive to customer demands, i.e. they upgrade their core bandwidth to meet the demands of customers they serve. Establishing a facility incurs substantial costs covering land acquisition costs, workforce costs, energy costs, equipment costs, maintenance costs, operational costs. The costs involved in servicing a customer is based on the distance in kilometres a customer is from a facility by a unit of cost per kilometre. Customers also require backup connections linking them to different facilities other than the ones they are currently serviced by to ensure resilience in connection.

We assume an already set of existing facilities that currently supply the demands of all existing customers. Due to the emergence of new customer locations in places that are further from existing facilities and the relocation of some existing customers, existing facilities are unable to efficiently service the demands of customers without attracting high service costs. This is because the further away a customer is from a facility less the quality of service a customer receives and the more expensive it is to provide a service to the customer. Also, the relocation of some existing customers renders some existing facilities unprofitable due to the significant capital required to run the facility to service a few existing customers connected to these facilities. If a decision is made to shut down unprofitable facilities, existing customers who are connected to the facilities to be shut down have to be reassigned to new facilities. The decommissioning of a facility and the reassignment of customers to new facilities both attract costs e_i and g respectfully. Opening a new facility i attracts a cost of c_i however, the existing facility does not attract an opening cost. Every opened facility attracts a running cost of h_i. To accommodate the demands of all the customers j allocated to a facility, the core bandwidth of the facility needs to be adapted, resulting in a cost calculated by the function $o_i(x_{ij})$ which is a step cost function where x_{ij} determines if a customer is connected to a facility or not. The core bandwidth to be used at a facility is the smallest value from the set $S = \{s_1, ..., s_q\}$ of all core bandwidths available that can accommodate the sum of all facilitys customer expected bandwidth.

To ensure competitive pricing, quality of service and uninterrupted connections, we are motivated to formulate two new LA problems. The new formulations seek to find the optimal locations of facilities that minimise the total costs incurred by the service company for establishing new facilities, the running costs of all established facilities, shutting down unprofitable facilities, reassigning customers, and servicing customer demands, i.e. both primary and backup connections.

3.2 Problem Formulation

In this Section, we present the formal description of LARP and LARPR. These formulations look for the optimal location for facilities and assignment decisions of customers, that lead to the minimisation of the total operating costs. To mathematically formulate our LA problem we use the following notations:
We assume the decision space and variables to be discrete.

- A set $A = \{a_i, ..., a_m\}$ of potential locations.
- A set $B = \{b_j, ..., a_n\}$ of customer locations.
- A set S of available core bandwidth. The core bandwidth to be used by facility a_i is the smallest value of the set $S = \{s_1, ..., s_q\}$ that can accommodate the sum of all expected bandwidths of customers connected to facility a_i
- c_i: cost for opening a new facility
- h_i: running cost of an opened facility
- e_i: cost of shutting down a facility
- g: reassignment cost of moving customer j from facility i to facility k
- l: the number of reassignment of customer j. $l \in \{0, 1, 2\}$
- p_q: cost associated core bandwidth s_q
- w_j: expected bandwidth of customer b_j

- d_{ij}: service cost of servicing customer b_j from facility a_i
- $x_{ij} = 1$ if customer b_j is connected to facility a_i and 0 otherwise.
- U: maximum number of total facilities that can be opened.
- The decision variables are represented by a binary string $x = \{x_1, ..., x_m\} \in \{0,1\}^m$ where 1 represents an opened facility and 0 represent a closed facility.

Possible solutions to the LA problem formulation are assessed based on the facility and service costs. Nevertheless, considering the capacity of facilities are not limited, once locations have been determined, the highest performance is achieved when each customer employs the facility giving the least expensive service. Therefore for the LA problem formulations, a solution to the problem is wholly determined by choice to place a facility or not at each of the m locations.

3.2.1 Location-Allocation Resilience Problem (LARP)

LARP aims to minimise the operational costs made up of establishing new facilities, the running costs of all established facilities, shutting down unprofitable facilities, reassigning customers, and servicing customer demands, i.e. both primary and backup connections. We formulate the problem of LARP as:

$$f(x) = \sum_{i=1}^{m} \left[(c_i + h_i) * x_i + \sum_{j=1}^{n} (d_{ij} * x_{ij}) + e_i * (1 - x_i) + \sum_{j=1}^{n} (g * l * x_{ij}) + o_i(x_{ij}) \right] \tag{3.1}$$

where:

$$o_i(x_{ij}) = \begin{cases} p_1 & \sum_j^n w_j x_{ij} < s_1 \\ p_2 & s_1 \leq \sum_j^n w_j x_{ij} < s_2 \\ \cdots & \\ p_q & s_{q-1} \leq \sum_j^n w_j x_{ij} \end{cases} \tag{3.2}$$

Subject to:

$$\sum_{i=1}^{m} x_{ij} = 2, \forall j \tag{3.3}$$

$$x_i \in \{0, 1\} \tag{3.4}$$

Equation (3.1) minimises the total operational costs. Equation (3.2) determines the total core bandwidth of a facility whiles Equation (3.3) ensures that every customer is connected to two facilities. Constraint (3.4) defines the decision variables.

3.2.2 Location-Allocation Resilience Problem with Restrictions (LARPR)

Unlike LARP, which has no restrictions on the number of facilities that can be opened, LARPR has a budget constraint that limits the total number of facilities that can be established. The objective of LARPR remains the same as the objective of LARP presented in Equation (3.1). However, we introduce a new constraint shown in Equation (3.5) that limits the number of opened facilities.

$$\sum_{i=1}^{m} a_i \leq U \quad | \quad a \in \{0, 1\}, \tag{3.5}$$

3.3 Proposed Methods

All algorithms employ a binary problem representation and so can be directly compared in the same search space.

3.3.1 Genetic Algorithm (GA)

In this Section, we describe the different GA operators presented in the literature for tackling LA problem formulations.

Two initialisation methods are presented.

- Random Initialisation (R): In random initialisation, the initial population is populated with random solutions with a uniform probability. In random initialisation, every gene or element in a solution has a probability of 0.5 of been assigned a 0 or 1 [99].
- Heuristic Initialisation (H): The Heuristic initialisation method [163] functions on the basis that an approximate number of facilities in the optimal solution to the LA problem can be estimated from the ratio of facility costs to service costs. Hence the more efficient search for optimal solutions can be achieved by configuring the initial population in the areas of the search space where optimal solutions are likely to exist. For this purpose, a classification index t is introduced:

$$t \quad = \frac{\frac{1}{m}\sum_{i=1}^{m} c_i}{\frac{1}{mn}\sum_{i=1}^{m}\sum_{j=1}^{n} d_{ij}} \tag{3.6}$$

  The classification index t is the ratio of the average facility cost to the average service cost. The authors [163] of this heuristic method argue that by using index t, one can determine the number of opened facilities in an optimal solution. Nevertheless, with too detailed a classification, the first solutions would converge in a single search space, consequently inducing early convergence of solutions. To avoid the problem of early convergence, the heuristic considers two cases when deciding the number of facilities for initial solutions: whether facility costs are similar to service costs or facility costs are more significant than service costs. The initial solution $x = \{x_1, ...x_m\}$ is generated using the classification index t.

Three selection methods are presented.

- Tournament selection: In this selection procedure [163], a parent is selected from the population using the tournament selection parameter N_{tour}. N_{tour} solutions are randomly selected from the population and a competition is held amongst the solutions. The fittest of the solutions is chosen as a parent for recombination. The process is repeated to chose the second parent.

- Roulette-Wheel selection: In this selection process [99], the selection of a solution to be a parent is dependent on a probability which is equivalent to the solutions fitness. Hence, a solution with a higher fitness will have a higher chance of being chosen as a parent. The strategy employed by Roulette wheel employs a selection pressure to the fitter solutions in the population, evolving better solutions to its fitness. Consider a circular wheel which is split into k pies, k representing all solutions in a population. Each solution is assigned a piece of the pie on the wheel. The size of the pie is relative to the proportion of the solutions fitness. On the circumference of the wheel, a fixed spot is selected and the wheel is revolved. When the wheel comes to a stop, the solution whose fitness area lies in front of the fixed spot is selected as a parent. The second parent is selected by repeating the process.

- Fine-Grained tournament selection: The Fine-grained tournament selection (FGTS) [57] is similar to the tournament selection; however FGTS is managed by the real value parameter F_{tour} (which is the desired mean tournament size) rather than the integer parameter N_{tour} (which is the tournament size). Much like the tournament selection, a solution is selected if the solution is the fittest in the tournament. Unlike the tournament selection, the tournament size is not unique in the entire population. In FGTS, different tournaments are held with a varying number of opponents within a single step of the selection process. The F_{tour} parameter manages the selection process. Sizes of the tournaments are $F_{tour^-} = [F_{tour}]$, $F_{tour^+} = [F_{tour}]+1$. The size of all z held tournaments is either F_{tour^-} or F_{tour^+}. The number of tournaments with size F_{tour^-} is denoted as z^- and the number of tournaments with size F_{tour^+} is denoted as z^+. The sum of z^- and z^+ is z. The pseudocode of FGTS is presented in algorithm 2

Algorithm 2 Fine-Grained Tournament Selection pseudo-code

.

```
Input: Population $p$ (size of array $p$ is $z$)
Input: Desired average tournament size $F_{tour}$, $F_{tour} \in R$;
Output: Population after selection $p'$ (size of array $p'$ is z)
begin
$F^{-}_{tour}$ = trunc($F_{tour}$);
$F^{+}_{tour}$ = trunc($F_{tour}$)+1;
$z^{-}$ = trunc($n * (F^{+}_{tour} - F_{tour})$);
$z^{+} = n - z^{-}$;
1. Conduct tournament with size $F_{tour^{-}}$
for i = 1 to $z^{-}$ do
        $p[i]'$ =best fitted among $F_{tour^{-}}$ solutions randomly selected from population $p$.
end for
2. Conduct tournament with size $F_{tour^{+}}$
for i = $z^{-}$ + 1 to $n$ do
        $p[i]'$ =best fitted among $F_{tour^{+}}$ solutions randomly selected from population $p$.
end for
    return $p^{'}$;
end
```

Two crossover operators are presented:

- One-Point Crossover (1P): In 1P [114] a single random point $p_i(i = 0 \quad to \quad n{-}1)$ is chosen from the length of the solution. Once a point is selected, the tails of the two parents selected for crossover are exchanged to create new offsprings. The 1P crossover is employed with a crossover probability of 0.9.
- Uniform crossover (U): BF [163] affords the uniformity in merging the genes of two-parent solutions. It accomplishes this process of exchanging bits in the parents to be added in the child solution by adopting a uniform random real number u (between 0 to 1). The random real value determines whether the child choose the i^{th} genes from the first or second parent. The U crossover is employed with a high crossover probability of 0.9.

Two mutation operators are presented.

- Bit-Flip mutation (BF): In BF [163] one bit from a solution is chosen at random and flipped, i.e. if the randomly chosen bit is 0, the bit is then flipped to become 1. The BF mutation is employed with a low mutation probability of 0.2.
- Partial Space Search (PSS): In PSS [163] a parent solution is divided into two subsets of genes A_1 and A_2 and different actions are applied to each subset. For A_1 the best potential sites are found by ranking facilities according to facility costs and the set of customers for which costs are at a minimum at site a_i. For

A_2, if the number of facilities open exceeds the reference value t described in equation 3.6, then some of the open facilities are randomly closed to keep the number of opened facilities in A_2 within the reference value. The two subsets of $A_1 \cup A_2$ are then combined to generate a new solution. The PSS mutation is employed with a low mutation probability of 0.2. Algorithm 3 presents the pseudocode of Partial Space Search proposed for the LA problem.

Algorithm 3 Partial Space Search pseudo-code

```
Require: $A = \{a_1, ...., a_m\}$, Mutation rate for Partial Space Search: $\sigma$, Classification index $t$.
Generate subset $A_1$ and $A_2$ from A
for Subset $A_1$ do
    Rank facilities according to the set of customers for which the service costs are minimum at site $a_i$ and the set of potential sites with facility costs greater than $a_i$ and $a_j$.
end for
for Subset $A_2$ do
    if number of opened facilities exceed $t$ then
        Randomly close some of the opened facilities to keep the number of opened facilities in $A_2$ within the reference value
    end if
end for
Recombine subsets to create new solution.
```

3.4 Experimental Setup

To evaluate the performance of the GA variants and PBIL, we test them against two datasets. The first dataset is the uncapacitated warehouse location problem set from the OR-Library [13] referred to as the CAP dataset. This is a standard dataset that has facility and customer locations and does not capture the aspect of resilience. The choice for selecting the CAP dataset is motivated by the availability of optimum values to the problems instances and its use in a few of the LA problem literature for comparing the performance of algorithms [1; 100; 101; 119; 163]. The CAP test instances consist of four small-sized problems (cap71,cap72,cap73 and cap74) with 16 facility locations by fifty demand points; four medium-sized problems (cap101,cap102,cap103 and cap104) having 25 facility locations and 50 demand points; seven large-size problems namely cap131, cap132, cap133 and cap134 having 50 facility locations and 50 demand points and capa, capb and capc having 100 facility locations and 1000 demand points.

The second dataset is typical of a real-world service company data and captures the aspect of resilience. The generated instance contains 100 facilities which correspond to the first 100 most populous cities in the United States and 10000 customer locations.

The second dataset is tested on LARP and LARPR.

For every problem instance, we run an algorithm twenty (20) times to obtain a distribution of results. We then employ the Friedman test described in Section 3.4.1 to conduct the performance evaluation of the algorithms. Parameters for the GA presented in Table 3.1 were set according to the best parameter values recorded in [163] determined by experimenting with the CAP data-set from [13]. We set the learning rate parameter and truncation size for PBIL in Table 3.1 as the default value presented in the literature [9; 10] which has been experimentally shown to offer a good trade-off between exploitation and exploration of the problem search space. To allow for comparison, we employ the same population size and fitness evaluations for all algorithms.

Table 3.1: Parameters for GAs and PBIL

Parameter	Value
Population size	50
Fitness evaluation	20000
GA crossover rate	0.9
GA mutation rate	0.2
PBIL learning rate	0.1
PBIL truncation size	0.5

In order to understand the contribution each GA configuration component makes to the performance of a GA, we exhaustively combine the configuration components to create 24 GA variants. The configuration of each GA variant is represented by {Initialiser/Selection/Crossover/Mutation}. The configuration components are presented in Table 3.2.

Table 3.2: GA components

GA operator	Representation
Initialisers	
Random	R
Heuristic	H
Selection	
Tournament	T
Fined-grained Tournament	FGTS
Roulette Wheel	RW
Crossover	
Uniform	U
1-point	1P
Mutation	
Bit-flip	BF
Partial space Search	PSS

We present the 24 GA variants in Table 3.3.

Table 3.3: GA Configurations

Initialiser	Selection	Crossover	Mutation
R	T	U	BF
H	T	U	BF
R	T	1P	BF
H	T	1P	BF
R	T	U	PS
H	T	U	PS
R	T	1P	PS
H	T	1P	PS
R	FGTS	U	BF
H	FGTS	U	BF
R	FGTS	1P	BF
H	FGTS	1P	BF
R	FGTS	U	PS
H	FGTS	U	PS
R	FGTS	1P	PS
H	FGTS	1P	PS
R	RW	U	BF
H	RW	U	BF
R	RW	1P	BF
H	RW	1P	BF
R	RW	U	PS
H	RW	U	PS
R	RW	1P	PS
H	RW	1P	PS

To conduct the performance evaluation of the GA variants and PBIL, we employ the Frieman test. We describe the Friedman test is Section 3.4.1.

3.4.1 Friedman test

The Friedman test [61; 62], also known as the Friedman two-way analysis of variances by ranks can be described as the non-parametric analogue of the parametric two-way analysis of variance. The Friedman test is useful in answering the question: within a collection of k samples, where the size of k is greater or equal to 2, does two of the samples at the least depicts populations with different mean values. The Friedman test is considered as a multiple comparison test which strives to identify significant or notable differences among the behaviour of at least two algorithms. The null hypothesis of Friedmans test denotes that there is no difference between the means of populations. On the other hand, the alternative hypothesis states that there is a difference between the means of populations. The hypothesis is thus non-directional, i.e. the alternative hypothesis does not state the direction of the difference; it only indicates that a difference exists. In computing the test statistic for the Friedman test, the first step is to transform the initial results into ranks. The following steps describe the process for computing the ranks:

1. The first step is to collect all observed values or results for each algorithm or problem pair.
2. To obtain the ranks, values of each problem i of algorithm j is ranked from 1, which is the best result to k which is the worst result. The ranks are therefore expressed as $r_i^j (1 \leq j \leq k)$.
3. To achieve the definitive rank for each algorithm j, the average of the ranks calculated for all problems are computed. This is expressed as $R_j = \frac{1}{n} \sum_i r_i^j$.

Hence, the Friedman test separately ranks the algorithms for each problem; this means that the algorithm with the best performance will have the rank of 1, the second-best performing algorithm will have the rank of 2 and so on. In the event of ties, it is still suggested to calculate the average ranks.

Because the Friedman test is non-directional, it can only identify significant differences across the entire multiple comparisons. As a result, the Friedman test is incapable of ascertaining a precise comparison between the algorithms being compared.

However, when the purpose of performing multiple comparison test is to ascertain a precise comparison between the algorithms being compared using a control method, then a family of hypotheses can be defined relating to the control method. A control method is the algorithm of interest to the experimental study whose performance can be compared with other algorithms. In the case where we hope to find the precise comparison between algorithms, a post-hoc test can be administered to obtain a *p-value*. Post-hoc methods help to determine which of the algorithms are deemed to be significantly better or worse when compared to the control method. In this work, we employ the Holms procedure. Our choice for the Holm procedure is motivated by the fact that it is easy to compute and considered in the literature [47] to be more potent than other post hoc methods such as the Bonferroni-Dunn test.

3.5 Experimental Results and Analysis

In this Section, we present and analyse the results obtained from experiments by the GA variants and PBIL on data instances described in Section 3.4

3.5.1 Results of GA configurations and PBIL on CAP problem instances

We first examine the performance of the algorithms on the CAP dataset. In Table 3.4, the column labelled *Algorithm* shows the GAs configurations and PBIL. The column labelled *Mean Rank* shows the average ranking each algorithm obtained over the 15 CAP problem instances. Here the lower the mean rank, the better the performance of the algorithm. The best mean ranked algorithm is highlighted in bold.

Table 3.4: Mean Rank of GA Variants and PBIL on CAP dataset

Initialiser	Selection	Crossover	Mutation	Mean Rank
R	T	U	BF	6.5
H	T	U	BF	5.17
R	T	1P	BF	15.57
H	T	1P	BF	12.33
R	T	U	PS	8.73
H	T	U	PS	6.37
R	T	1P	PS	22.73
H	T	1P	PS	20.47
R	FGTS	U	BF	7.2
H	FGTS	U	BF	6.33
R	FGTS	1P	BF	11.63
H	FGTS	1P	BF	10.07
R	FGTS	U	PS	8.83
H	FGTS	U	PS	7.23
R	FGTS	1P	PS	21.8
H	FGTS	1P	PS	19.33
R	RW	U	BF	8.93
H	RW	U	BF	7.47
R	RW	1P	BF	22.13
H	RW	1P	BF	18.93
R	RW	U	PS	16
H	RW	U	PS	12.93
R	RW	1P	PS	23.13
H	RW	1P	PS	20.73
PBIL				**4.43**

We apply the Friedman test defined in Section 3.4.1 to test for statistical difference between results obtained by all algorithms on the CAP dataset. When a statistical difference is detected, we employ the Holms procedure defined in Section 3.4.1 with a significance level of α= 0.05 to make a proper comparison between the algorithms. The statistical comparison of results is presented in Table 3.5. In the Holms procedure, if the p-value obtained by an algorithm is less or equal to (α/i) where i is the position of the algorithm in the Table, the algorithm is seen to be statistically different from the best mean-ranked algorithm highlighted in Table 3.4 which is PBIL. Algorithms in Table 3.5 are arranged in descending order with the statistically worse algorithm highlighted in bold.

Table 3.5: Holm test on Friedman results to compute the differences between results obtained by GA configurations and PBIL on CAP instances. $\alpha = 0.05$

i	Initialiser	Selection	Crossover	Mutation	p-value	α/i
24	**R**	**RW**	**1P**	**PS**	**3.44E-12**	**2.08E-03**
23	**R**	**T**	**1P**	**PS**	**9.79E-12**	**2.17E-03**
22	**R**	**RW**	**1P**	**BF**	**4.51E-11**	**2.27E-03**
21	**R**	**FGTS**	**1P**	**PS**	**1.03E-10**	**2.38E-03**
20	**H**	**RW**	**1P**	**PS**	**1.32E-09**	**2.50E-03**
19	**H**	**T**	**1P**	**PS**	**2.43E-09**	**2.63E-03**
18	**H**	**FGTS**	**1P**	**PS**	**2.95E-08**	**2.78E-03**
17	**H**	**RW**	**1P**	**BF**	**6.83E-08**	**2.94E-03**
16	**R**	**RW**	**U**	**PS**	**1.68E-05**	**3.13E-03**
15	**R**	**T**	**1P**	**BF**	**3.43E-05**	**3.33E-03**
14	**H**	**RW**	**U**	**PS**	**1.56E-03**	**3.57E-03**
13	**H**	**T**	**1P**	**BF**	**3.29E-03**	**3.85E-03**
12	R	FGTS	1P	BF	7.38E-03	4.17E-03
11	H	FGTS	1P	BF	3.61E-02	4.55E-03
10	R	RW	U	BF	9.40E-02	5.00E-03
9	R	FGTS	U	PS	1.02E-01	5.56E-03
8	R	T	U	PS	1.10E-01	6.25E-03
7	H	RW	U	BF	2.59E-01	7.14E-03
6	H	FGTS	U	PS	2.97E-01	8.33E-03
5	R	FGTS	U	BF	3.03E-01	1.00E-02
4	R	T	U	BF	4.42E-01	1.25E-02
3	H	T	U	PS	4.72E-01	1.67E-02
2	H	FGTS	U	BF	4.80E-01	2.50E-02
1	H	T	U	BF	7.85E-01	5.00E-02

Results in Table 3.5 shows the GA configurations from thirteen to twenty-four to be statistically different from the control method and algorithms 1 to 12. The best mean ranked algorithm in Table 3.4, i.e. PBIL, is used as the control method. Algorithms one to twelve are deemed to be mutually statistically indistinguishable from the control method.

To understand the contribution each component makes to the performance of a GA, we examine the GA configurations. To do this, consideration is given to the first twelve algorithms, whose results are deemed to be mutually statistically indistinguishable from the PBIL. In the Table, both the heuristic and random method of initialisation appears in six GA configuration out of the twelve GA configurations of interest. The heuristic method generates initial solutions in the area of the problem search space where possible good solutions are assumed to exist. It does this by using an index calculated from the ratio of the average of facility opening costs and the average of service costs. This helps to provide good initial solutions for breeding new solutions. However, as seen from later experiments, the Heuristic approach seems to work best on problems with fewer or no constraints, i.e. LA problem formulations that only gives considerations to only facility and service costs. Random initialisation creates much diversity in the population at the start of the search, which allows a greater area of the search space to be explored. Indeed it has been observed experimentally in the literature that random solutions often are the ones to drive the population to optimality [172].

Fine-Grained tournament selection (FGTS) appears in six of the GA configurations out of the twelve GA configurations of interest when compared to Tournament (T) and Roulette wheel (RW) selection with four and two appearances respectively. The ability of FGTS to conduct several tournaments in one selection process gives it a better chance of finding fitter parents for recombination when compared to the tournament selection which allows for a single tournament in one selection process. In FGTS the population is divided into different groups, and a varying number of parents are selected from each group to compete in a tournament where the fittest parents are selected for each tournament. This process allows for a good exploration of the population in finding good parents for recombination. Whiles in tournament (T) selection a single tournament is held within a single step of the selection process which can sometimes lead to missing out on the more fitter solutions in the population. The performance of the Roulette wheel method of selection can be attributed to its naive way of selecting solutions. For example, if the best solution in the population has a 90% fitness of the entire roulette wheel, then the other solutions have very few chances to be selected, and this can often cause early convergence of the search.

Uniform (U) crossover appears in ten GA configurations out of the twelve GA configurations of interest when compared to One-Point (1-P) crossover with two appearances. The success of Uniform (U) crossover over 1-P crossover may be attributed to the ability of uniform crossover to keep a right balance between diversity and convergence. In Uniform crossover, new solutions will be different from their parents if their parents are not similar which helps with diversity in the population and similar to parents if parents are similar which helps to preserve good genes. Since tails of two parents are swapped after a cross-point to get a new offspring in the 1-P crossover, there is a high percentage of preserving bad genes in an offspring from the parent which can affect the performance of the GA configuration in finding optimal solutions.

Finally, BF mutation appears in eight out of the twelve algorithms of interest when compared to PSS with four appearances. The success of Bit-Flip mutation can be attributed to its ability to introduce diversity into the population without causing too much disruption. BF does this by altering a single gene in the solution which ensures diversification without entirely changing the solution. The process of BF mutation, therefore, helps to prevent the search from converging too quickly but also ensures that the search process is not heavily disrupted to prevent convergence after a certain period of exploration of the search space. PSS method works by dividing the solution into two subsets, then employs a local search process using a classification index described in Section 3.3.1 to find the best facility sites or genes in the first subset. However, PSS randomly rearranges the genes in the second subset. The second process introduces a higher degree of diversification which causes a later convergence of the algorithm, which in part affects the performance of the algorithm to find optimal solutions within the designated run time of the algorithm.

PBIL showed good and competitive performance by achieving the best mean rank. The performance of PBIL can be attributed to its ability to quickly focus the search in the region of the search space where good solutions are likely to exist. The competitive learning in PBIL helps it to focus its search by evolving the probability vector with good solutions and then sampling equally or similarly good solutions from the probability vector.

3.5.2 Results of GA configurations and PBIL on LARP LARPR problem instance

In LARP there are no restrictions to the number of facilities that can be opened to service demands whiles LARPR restricts the number of facilities that can be opened due to budget constraints. Hence, solutions found in LARPR are penalised by adding a high cost when the number of opened facilities exceeds the allowed number of facilities to be opened. The primary choice of PBIL is highlighted here as we can directly encode problem knowledge into the probabilistic model. Encoding problem knowledge here means, we can feed the model at the start of the search with a probable number of facilities that we are allowed to open out of the maximum number of potential locations.

In encoding the problem knowledge in PBIL, we assume three scenarios. In the first scenario, we assume that each facility location has an even chance to be selected for establishing a facility, and hence we initialise the probability vector with the default value of 0.5%. In the second scenario, we assume that the total number of facilities locations that can be opened out of the 100 potential locations must not exceed 34 facilities hence we initialise the probability vector with the value of 0.34. The restriction on the total number of facilities that can be opened is due to budget constraint, and the total number of 34 is set based on a real-world problem with industry experts. The third scenario extends the second scenario, by assuming an initial set of 29 opened facilities which attracts zero opening costs. As in the second scenario, the total number of facilities that can be opened out of the 100 locations is 34. Due to the high cost involved in establishing a facility, we would prefer to have the already opened facilities stay open if favourable and possibly open five additional new facilities. We, therefore, initialise the first 29 elements of the PBILs probability vector with the value of 0.85 and remaining elements with 0.15. Here we hope to give a higher probability to the first 29 facility locations of staying opened since they attract no opening cost and a 15% chance to the remaining facilities. The value of 29 existing facilities was set according to a real-world problem in collaboration with industry experts. To allow for a good comparison, we initialise all GA configurations with the same probabilities of generating initial solutions.

To make for easy discussion, we grouped succeeding Tables according to the three scenarios described above. In Tables 3.6, 3.7 and 3.8 we present the mean ranks achieved by each algorithm on LARP and LARPR. In all three Tables, *Algorithm* represents the GA configurations and PBIL, *Mean* represent the mean value achieved overall twenty runs for each algorithm on LARP and LARPR whiles *Mean Rank* shows the mean ranks achieved by each algorithm on LARP and LARPR. In all Tables, the best algorithm, i.e. the algorithm that achieved the least mean rank, is highlighted in bold.

Table 3.6 shows the mean values obtained by all algorithms on LARP and LARPR by initialising the probability vector with the default value of 0.5.

Table 3.6: Mean rank results of GA configurations and PBIL on LARP and LARPR with an initialisation probability of 0.5

Algorithms				LARP		LARPR	
Initialiser	Selection	Crossover	Mutation	Mean	Rank	Mean	Rank
R	T	U	BF	9.91E+07	2	1.61E+08	2
H	T	U	BF	9.95E+07	6	1.61E+08	3
R	T	1P	BF	1.14E+08	10	1.90E+08	15
H	T	1P	BF	1.77E+08	20	1.85E+08	14
R	T	U	PS	9.96E+07	7	1.82E+08	12
H	T	U	PS	1.32E+08	17	1.79E+08	10
R	T	1P	PS	1.33E+08	18	2.89E+08	22
H	T	1P	PS	3.96E+08	24	4.17E+08	25
R	FGTS	U	BF	9.91E+07	3	1.62E+08	4
H	FGTS	U	BF	9.93E+07	4	1.63E+08	5
R	FGTS	1P	BF	1.06E+08	9	1.81E+08	11
H	FGTS	1P	BF	1.34E+08	19	1.74E+08	8
R	FGTS	U	PS	9.94E+07	5	1.83E+08	13
H	FGTS	U	PS	1.30E+08	14	1.77E+08	9
R	FGTS	1P	PS	1.30E+08	16	2.17E+08	17
H	FGTS	1P	PS	3.90E+08	23	3.99E+08	23
R	RW	U	BF	1.03E+08	8	1.68E+08	7
H	RW	U	BF	1.16E+08	11	1.67E+08	6
R	RW	1P	BF	1.30E+08	15	2.06E+08	16
H	RW	1P	BF	2.92E+08	22	2.86E+08	21
R	RW	U	PS	1.23E+08	12	2.22E+08	18
H	RW	U	PS	2.52E+08	21	2.59E+08	20
R	RW	1P	PS	1.24E+08	13	2.42E+08	19
H	RW	1P	PS	4.08E+08	25	4.12E+08	24
PBIL				**9.91E+07**	**1**	**1.61E+08**	**1**

In Table 3.6, PBIL is seen to have achieved the best mean rank for both LARP and LARPR when initial solutions are initialised with a default probability of 0.5 for both GA configurations and PBIL. The performance of PBIL shows that even without encoding problem knowledge in the model, the learning process of PBIL saves the knowledge obtained about good solutions in the early stages of the search and reuse this knowledge to generate similar likely good solutions within each generation of the search process. This helps PBIL to quickly focus the search in the regions of the search space where good solutions are likely to exist. GAs with configurations $R/T/U/BF$

and $R/FGTS/U/BF$ rank second and third respectively with almost similar values to PBIL on LARP. However, even-though GA configuration $R/T/U/BF$ maintains the second rank on LARPR, configuration $R/FGTS/U/BF$ is taken over by $H/T/U/BF$. A look at the three GA configurations shows that they all employ uniform crossover and Bit-Flip mutation. However, configurations $R/T/U/BF$ and $H/T/U/BF$ employ the tournament selection method whiles $R/FGTS/U/BF$ used FGTS method. Configurations $R/T/U/BF$ and $R/FGTS/U/BF$ both employ random initialisation whiles $H/T/U/BF$ employs the heuristic method of initialisation. Although the choice of initialisation and selection methods impacts the performance of a GA to a degree as seen in the three GA configurations, considering that exploration and exploitation of the search space are essential concepts that meaningfully impact on the performance of a GA. We can safely assume that the choice of a crossover and mutation operator in a GA which are the main operators that define the degree of exploration and exploitation in a GA play a significant role in the performance of a GA configuration. In this situation, both the Uniform crossover and mutation operator combine well together to enhance the performance of the GA configurations.

Table 3.7 shows the mean values obtained by all algorithms on LARP and LARPR by initialising the probability vector with the default value of 0.34.

Table 3.7: Mean rank results of GA configurations and PBIL on LARP and LARPR with an initialisation probability of 0.34

Algorithms				LARP		LARPR	
Initialiser	Selection	Crossover	Mutation	Mean	Rank	Mean	Rank
R	T	U	BF	9.91E+07	2	1.61E+08	2
H	T	U	BF	9.95E+07	6	1.61E+08	4
R	T	1P	BF	1.17E+08	11	1.78E+08	13
H	T	1P	BF	1.77E+08	20	1.85E+08	16
R	T	U	PS	9.95E+07	7	1.63E+08	7
H	T	U	PS	1.32E+08	17	1.79E+08	14
R	T	1P	PS	1.29E+08	15	1.98E+08	19
H	T	1P	PS	3.96E+08	24	4.17E+08	25
R	FGTS	U	BF	9.91E+07	3	1.61E+08	3
H	FGTS	U	BF	9.93E+07	4	1.63E+08	6
R	FGTS	1P	BF	1.07E+08	9	1.72E+08	10
H	FGTS	1P	BF	1.34E+08	18	1.74E+08	11
R	FGTS	U	PS	9.94E+07	5	1.62E+08	5
H	FGTS	U	PS	1.30E+08	16	1.77E+08	12
R	FGTS	1P	PS	1.27E+08	13	1.92E+08	18
H	FGTS	1P	PS	3.90E+08	23	3.99E+08	23
R	RW	U	BF	1.03E+08	8	1.64E+08	8
H	RW	U	BF	1.16E+08	10	1.67E+08	9
R	RW	1P	BF	1.28E+08	14	1.83E+08	15
H	RW	1P	BF	2.92E+08	22	2.86E+08	22
R	RW	U	PS	1.25E+08	12	1.88E+08	17
H	RW	U	PS	2.52E+08	21	2.59E+08	21
R	RW	1P	PS	1.62E+08	19	1.98E+08	20
H	RW	1P	PS	4.08E+08	25	4.12E+08	24
PBIL				**9.90E+07**	**1**	**1.60E+08**	**1**

In Table 3.7 PBIL is again seen to have achieved the best mean rank on both LARP and LARPR when the initial population is generated with a probability value of 0.34. Once the initial population is generated, PBIL updates the probability vector with the best half of the population and then samples new solutions from the probability vector. In this way, PBIL can maintain the encoded problem knowledge to drive the search to find the optimum or near optimum solution to the LARP and LARPR. Configurations $R/T/U/BF$ and $R/FGTS/U/BF$ achieve second and third ranks respectively on both LARP and LARPR with a noticeable difference in results when compared to PBIL. The performance of the GA configuration can be explained by the fact that even though the initial population of GAs are initialised with the same probability as PBIL. The encoded knowledge is lost during the process of recombination and mutation. The loss of knowledge is due to the random nature of the crossover and mutation operators. In the uniform crossover operator, because genes from the two parents are randomly chosen to form a new offspring, the new population generated is likely to lose some of the knowledge provided at the start of the search. The Bit-Flip mutation operator also randomly flip genes in a solution which is likely to have the same effect of losing the encoded knowledge in the search process. However, the consistent performance of GA configurations $R/T/U/BF$ and $R/FGTS/U/BF$ shows that the uniform crossover combines well with the bit-flip mutation to enhance the performance of the GAs. In both GA configuration, the Random initialiser is employed. Because solutions generated through the random initialiser are likely to be spread across the search space, it provides much diversity at the start of the search, which can lead to finding reasonable good solutions for recombination. The tournament and FGTS methods are employed by configurations $R/T/U/BF$ and $R/FGTS/U/BF$, respectively. The two selection methods appear to combine well with Random initialiser, Uniform crossover and Bit-flip mutation.

Table 3.8 shows the mean values obtained by all algorithms on LARP and LARPR by initialising the probability vector with the default value of 0.85/0.15.

Table 3.8: Mean rank results of GA configurations and PBIL on LARP and LARPR with an initialisation probability of 0.85/0.15

Algorithms				LARP		LARPR	
Initialiser	Selection	Crossover	Mutation	Mean	Rank	Mean	Rank
R	T	U	BF	9.91E+07	2.5	1.61E+08	3
H	T	U	BF	9.95E+07	6	1.61E+08	3
R	T	1P	BF	1.16E+08	10.5	1.78E+08	13
H	T	1P	BF	1.77E+08	20	1.85E+08	16
R	T	U	PS	9.96E+07	7	1.63E+08	6.5
H	T	U	PS	1.32E+08	17	1.79E+08	14
R	T	1P	PS	1.31E+08	16	1.98E+08	19.5
H	T	1P	PS	3.96E+08	24	4.17E+08	25
R	FGTS	U	BF	9.91E+07	2.5	1.61E+08	3
H	FGTS	U	BF	9.93E+07	4	1.63E+08	6.5
R	FGTS	1P	BF	1.07E+08	9	1.72E+08	10
H	FGTS	1P	BF	1.34E+08	18	1.74E+08	11
R	FGTS	U	PS	9.94E+07	5	1.62E+08	5
H	FGTS	U	PS	1.30E+08	15	1.77E+08	12
R	FGTS	1P	PS	1.29E+08	13.5	1.92E+08	18
H	FGTS	1P	PS	3.90E+08	23.5	3.99E+08	23
R	RW	U	BF	1.03E+08	8	1.64E+08	8
H	RW	U	BF	1.16E+08	10.5	1.67E+08	9
R	RW	1P	BF	1.29E+08	13.5	1.83E+08	15
H	RW	1P	BF	2.92E+08	22	2.86E+08	22
R	RW	U	PS	1.24E+08	12	1.88E+08	17
H	RW	U	PS	2.52E+08	21	2.59E+08	21
R	RW	1P	PS	1.43E+08	19	1.98E+08	19.5
H	RW	1P	PS	4.08E+08	25	4.12E+08	24
PBIL				**9.90E+07**	**1**	**1.60E+08**	**1**

From Table 3.8, a similar trend is observed as in previous Tables when algorithms are initialised with a value of 0.85/0.15. PBIL again is seen to achieve the best mean ranks on both LARP and LARPR. GA configurations $R/T/U/BF$ and $R/FGTS/U/BF$ achieves the same rank on LARP whiles $R/T/U/BF$ and $H/T/U/BF$ and $R/FGTS/U/BF$ all achieve the same rank on LARPR. The emergence of the three GA configurations employing the same crossover operator (Uniform) and mutation parameter (Bit-flip) shows that the combination of these two operators plays a defining role in the overall performance of a GA on the problems presented. Random initialisation appears in two of the GA configuration making it more preferred to the heuristic method. The tournament selection method appears in two out of the three GA configurations however FGTS has been consistent in all scenarios of the problem even on the CAP instance where it appeared in six out of the twelve algorithms of interest. The performance of both selection methods appears to both contribute positively to the performance of the GA when combined with random initialiser, Uniform crossover and bit-flip mutation. Based on the performance of the GA configurations, $R/T/U/BF$

and $R/FGTS/U/BF$ are considered as the best performing GA configurations.

3.6 Summary of Chapter

In this Chapter, we set out to find a suitable algorithm for solving the two novel non-linear formulations of LA problem introduced in Sections 3.2.1 and 3.2.2. The first model Location-Allocation Resilience Problem (LARP) has no restrictions on the number of facilities that can be established to service customer demands. The second model; Location-Allocation Resilience Problem with Restriction (LARPR) is a constrained version of LARP.

Literature within the field of LA problem showed GAs to be among the most successful algorithms presented for solving the LA problems in recent years. A search through literature found that the GAs employed different components and configuration choices. To develop an optimal GA configuration for solving the new models, we needed to understand the contribution each GA component made to the performance of the GA.

Combining the different GA components gave us 24 different configurations. The components included:

- Random and Heuristic Initialisation
- Tournament, Fine-Grained Tournament and Roulette Wheel selection
- Uniform and 1-Point crossover
- Bit-Flip mutation and Partial space search.

We applied the Population-Based Learning (PBIL) Algorithm to solve the formulated LA problems. Our choice for PBIL was motivated by the fact that PBIL has the potential benefits of an EDA but with a lightweight (univariate) modelling cost; and we observed that specifically for the novel LA problem formulations proposed, useful problem knowledge can be encoded directly into the probabilistic model.

Addition of PBIL to the 24 GA variants gave us a total of 25 algorithms. To test the performances of all algorithms, we generated a new benchmark problem instance motivated by a real-world problem from the telecommunication industry. The problem contained 100 facilities by 10000 customers. We also used fifteen CAP test instances [13] presented in the literature. To ensure a Uniform platform for result comparison, we initialised all algorithms with the same probability on the new benchmark problems.

Results from experiments on CAP dataset showed that among the GA configurations, 12 were considered to be mutually statistically indistinguishable when compared to the PBIL, which was the best algorithm among the 25 algorithms.

Experiments on the LARP and LARPR saw PBIL outperforming all GA configuration. The performance of PBIL can be attributed to its ability to use and retain encoded problem knowledge to drive the search for finding an optimal or near-optimal solution. In analysing the results of the top three best mean rank GA configurations, we found that the combination of the GA components: Random initialisation, tournament and FGTS, Uniform crossover and bit-flip mutation gave the best results on all problems. Thus our optimal GAs found were GAs with configurations $R/T/U/BF$ and $R/FGTS/U/BF$.

Although the GA configurations such as $R/T/U/BF$ and $R/FGTS/U/BF$ showed competitive performance on problem instances, PBIL emerged the best algorithm of the 25 algorithms, especially when consideration is made to the performance of the algorithms on the LARP and LARPR. PBIL was seen to perform best on all instances of LARP and LARPR when compared to the GA variants. In this work, we are primarily interested in a good enough algorithm for solving our LA problem. Hence the successful performance of PBIL over the GA variants, therefore, makes PBIL our algorithm of choice for tackling the formulated LA problems.

Chapter 4

Dynamic-Customer Location-Allocation Problem

4.1 Introduction

The location of facilities in both private and public sector systems critically affects the ability of these systems to deliver the essential services [45]. Because facility location decisions are long-term strategic decisions, they impact on shorter-term decisions such as resource allocation. The strategic nature of location problems requires that any reasonable formulation consider some aspects of future uncertainty. Changes in population, market size, environmental factors and the rapid advancement in technology often drives the need of consumers, which causes customers to relocate between cities. In the absence of substantial costs of opening a facility, the optimal option of locating facilities will be to site them optimally in each period to service changing distribution of customers. However, due to the significant capital investments required for locating or relocating a facility, facilities are expected to remain operable for an extended period [45]. The significant capital involved in locating facilities coupled with the changing distribution of customers makes it imperative that location choices executed today consider expected future circumstances. Considering changing distribution of customers when locating facilities will help to ensure that facilities are not only ideal for current conditions but also stay useful over the defined horizon [45]. The nature of this problem gives rise to the study of a dynamic LA problem formulation motivated by a real-world problem from the telecommunication industry, where customers frequently move between cities over time. We call this new formulation Dynamic-Customer Location-Allocation (DC-LA) Problem. DC-LA problem is formulated in the context of Robust Optimisation Over Time (ROOT) where facilities are established only once at the beginning of the defined period and are expected to be satisfactory in adequately servicing the changes in customers demands over the defined period. Although motivated by a real-world problem, the problem of DC-LA problem can be generalised to extend to broader location problems. DC-LA problem takes into account the actualised servicing costs and movement of customers over time. The new dynamic LA problem formulation generates random customer movements driven by the attractivity of cities.

This Chapter is organised as follows; we defined the problem of DC-LA problem in Section 4.2. Experiments are conducted in Section 4.3. We conclude the Chapter in Section 4.5. Work presented in this Chapter was published in the International Conference on Innovative Techniques and Applications of Artificial Intelligence, Springer, Cham, 2018.

4.2 Dynamic-Customer Location-Allocation (DC-LA) Problem

In defining Dynamic-Customer Location-Allocation (DC-LA) Problem, we first need to define an initial set of notations. Let a set of m cities $A = a_1, a_2, ..., a_m$ be a set of m potential locations, and $B = b_1, b_2, ...b_n$ be a set of n customers. Each $a_i \in [0, 1]^2$ and $b_j \in [0, 1]^2$ define the coordinates in a 2-dimensional plane. Here, the cost d_{ij} of connecting customer b_j to location a_i is defined by the euclidean distance between b_j to location a_i. Let $T = t_1, t_2, ..., t_{max}$ denote the defined time horizon where t_{max} is the maximum length of time. DC-LA problem considers the potential movement of customers over a given time horizon t_{max}. In DC-LA problem, the pattern of customer location changes is assumed to be stochastically driven by the attractivity of cities, i.e. how attractive a city is to a customer. In DC-LA problem a facility is assumed to be located within the centre of a city. Hence for the remainder of the work, a city and facility will be used interchangeably to refer to the location of a facility. Locations employed in DC-LA problem are assumed to be discrete, and the optimal number of facilities are found by solving the problem. The problem formulation assumes that facilities offer similar services to customers and facilities are unconstrained in the number of customers they can adequately service. Given the substantial costs incurred in locating facilities, and recognising that the first-period decision is the one that must be implemented [45], we believe that the goal of DC-LA problem planning should be to determine an optimal or near-optimal first-period choice of facilities for the defined horizon. For this reason, we model the problem as Robust optimisation over time (ROOT) problem defined in Section 2.3.1. Thus DC-LA problem concerns finding the optimal or near-optimal locations for establishing facilities in the first-period to service changing demands over a defined horizon to reduce the overall total costs.

In DC-LA problem, the movement of customers influences the connection cost of customers to facilities. DC-LA problem considers an initial set $A^{'} = \{a^{'}_1, a^{'}_2, ..., a^{'}_m\}$ of attraction rates for cities. The attraction rates define the probabilities at which customers will be placed in each city. In Section 4.2.1, we define the simulation model that simulates the movement of customers over a defined horizon t_{max}.

4.2.1 Simulation model

A simulation is based on the assumption that customers will move over time *i.e.* disappear from a location and reappear in another location. We also assume that the attraction rate of each city in the future is unknown. To develop our simulation model, we first define and understand the choice of parameters and environmental constants and how they influence the simulation model.

Movement Rate (mr) is an important parameter that is varied in the simulation process to regulates the mobility of customers. mr ranges from 0 to 1 and is used for determining the movement dates for a customer. Movement dates here refers to the times at which a customer changes locations between cities. The movement dates of each customer are sampled from a normal distribution centred in $mr \cdot t_{max}$. Hence, each customer will move on average, $mr \cdot t_{max}$ times during the simulation. A low mr value means that customers make frequent movements between cities over time whiles a higher mr means customers will make little or no movements between cities over time. E.g. if mr=0.25 and t_{max} = 30years, then per our model we will expect a customer to move on average every 7.5 years. However if mr= 0.75, then for the same value of t_{max} we will expect a customer to move on average every 22.5 years. Hence, for this work a low mr relates to high customer movements over t_{max} whiles a higher mr relates to lower customer movements over t_{max}. The values of mr were set in consultation with an expert from the telecommunication industry to realistically capture the varying rate of at which customers move between cities over time within the telecommunications industry.

Attraction Rates is an important parameter that is varied within the simulation process to define the probability of a city to attract customers. A set $A^{'} = \{a^{'}_1, a^{'}_2, ..., a^{'}_m\}$ represent a set of attraction rates where $\sum_{i=1}^{m} a^{'}_i = 1$. The attractivity of cities drives the movement of customers. We assume the attractiveness of a city will vary over time; hence, they are randomly generated within a simulation. The process to generate attraction rates is shown in Algorithm 4.

Algorithm 4 Cities attraction rates generation

```
Require: Number of facilities : $m$
  Set of attraction rates: $A = \emptyset$
  $a = random.nextDouble()$
  for i=1 to m do
    $a = (a + random.nextDouble())$
    $A = A \cup a$
  end for
  A/max(A)
```

The number of facilities (m) is an important environmental constant which will change from problem to problem; however, in this work, we are not going to focus on the changes in the problem search space. Rather our focus is on a real-world example.

The number of cities represents the possible number of locations where a facility can be established. Facility locations can range from ten locations to hundreds of locations. Each facility location $l_i \in [0,1]^2$ define the coordinates in a 2-dimensional plane. Each facility location also has a cost c_i of opening that facility. To get a good spread of the number of facility locations, we select a varying number of locations in consultation with industry experts within the telecommunication industry.

Much like the number of cities, the number of customer locations (n) is an important environmental constant which will change from problem to problem. The number of customers locations indicates the total number of customers whose demands need to be satisfied with operating facilities. The number of customers can range from a few hundred to thousands of customers. Customers are assumed to exits in cities. Each customer $b_i \in [0,1]^2$ define the coordinates in a 2-dimensional plane. To get a good spread of the number of customers, we select a varying number of customer locations in consultation with industry experts within the telecommunication industry.

We assumed a finite time in which we wish to simulate the movement of customers. Time is defined by a set $T = t_1, t_2, ..., t_{max}$, where t_{max} is the maximum length of time. The value of t_{max} was determined in consultation with industry experts from the telecommunication industry. The choice of t_{max} is informed by the extended period a facility is expected to be efficiently operable in servicing customers demands within the telecommunication industry.

Discount Rate (dr) is an important environmental constant which allows us to compare costs incurred at different times over t_{max}. The choice of the discount rate dr used in this work is motivated by historical data of interest rates from the Bank of England, which has settled at the low rate of 0.05 since March 2009 [96]. However, the choice of dr is only an estimate.

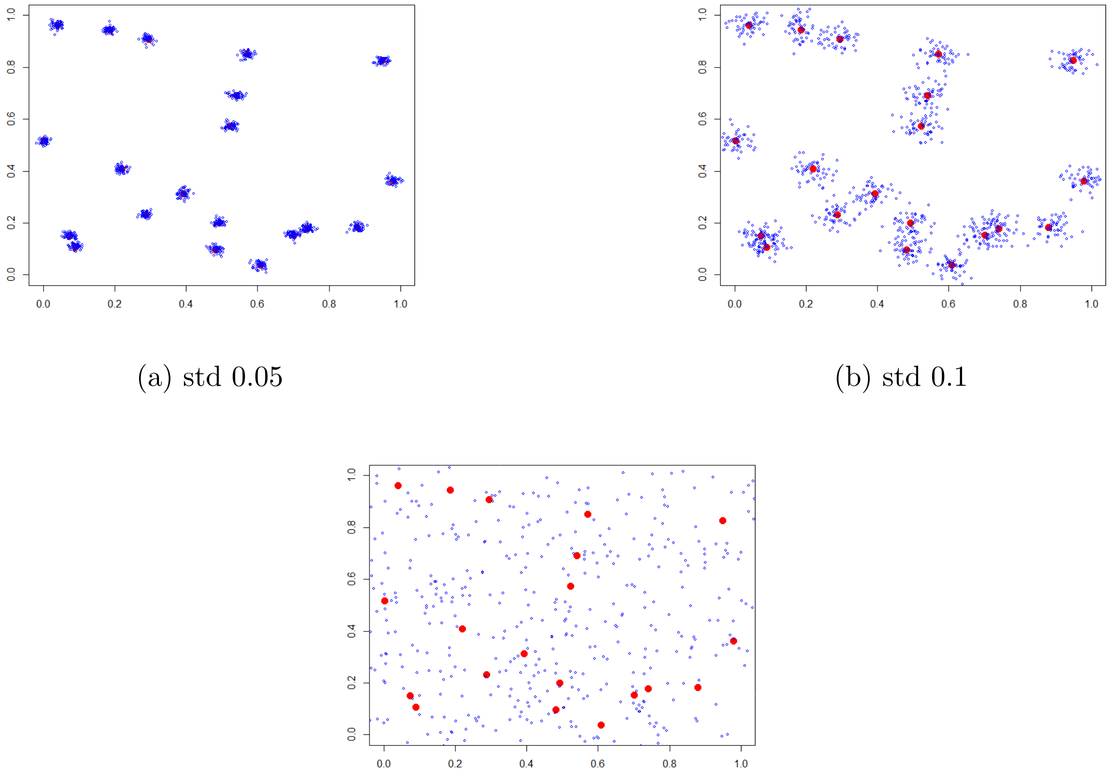
The Standard deviation for calculating Movement dates (*sd for mr*): *sd* here is the measure used to quantify the amount of dispersion in the movement dates generated for a customer. The value of *sd* here is motivated by the fact that we want the dates generated for a customer to be closer to the expected date a customer is expected to change locations.

The standard deviation for generating customers locations (*sd for n*): *sd for n* is employed to generate new locations for customers within a city. When a customer has to change locations, the new coordinates of the customer are obtained by sampling its location from a normal distribution centred in the coordinates of the city. Hence, the coordinates b_j of customer j placed in city i is given by:

$$b_j = \{\mathcal{N}(x_{a_i}, 0.1), \mathcal{N}(y_{a_i}, 0.1)\} \tag{4.1}$$

sd for n here indicates the range or radius a customer is to be located from the centre of a city. Choosing a high value for *sd for n* will cause customers to cover the whole space of interest; however, this is not realistic in the real world. In the real-world customers' locations in cities are clustered together and especially in places like the United States where we have got cities in small areas with relatively large gaps between cities. Therefore choosing a smaller *sd for n*, gives a more realistic picture of the real-world scenario. In Figure 4.1, we show the clustering of customers based on some values of *sd for n*. We generate 20 cities and for each city generate 100 customer locations by varying the values of *sd for n*. Figure 4.1 shows the effect the values of *sd for n* has on the distribution of customers. For our model to be representative of the realistic distribution of customers in a city, we are motivated to choose a lower value of *sd for n*. There is, however, an area to explore where the grouping of cities and customers of problems might change, which could be a potential area for further work.

Figure 4.1: Distributions of customers in cities according to values of *sd for n*



(a) std 0.05

(b) std 0.1

(c) std 0.5

Each simulation starts by generating new attraction rates for each city. For each customer, we then generate the movement times at which the customer is going to move over t_{max} years using mr. The movement times of each customer are sampled from a normal distribution centred in $mr \cdot t_{max}$. Hence, each customer will move on average, $mr \cdot t_{max}$ times during the simulation. The process to generate movement times is shown in Algorithm 5.

Algorithm 5 Customer movement dates generation

```
Require: Movement rate : $mr$
  Set of movement dates: $M = \emptyset$
  $t = 0$
  while $t < t_{max}$ do
    $t = t + \mathcal{N}(mr \cdot t_{max}, 0.1 \cdot t_{max})$
    $M = M \cup t$
  end while
```

Each simulation consists of generating customer movements and calculating their service costs over the t_{max}. For each simulation, we first generate a set of attraction rates for each city. Then for each customer b_j, we generate several possible dates along the defined horizon t_{max}, which represent the point at which a customer will change cities. We generate the possible movements of each customer b_j using the approach described in Algorithm 5. It is assumed that at the start of the simulation, each customer b_j is already located and serviced by a city. If a customer has to move at a point in time along the defined horizon t_{max}, a new city is chosen for the customer based on the attraction rate of a city. The new city for a customer b_j is determined by generating a random number between 0 and 1 each time customer b_j has to move. The customer is then assigned to the city within which range of attraction rate the random value generated happens to fall. Once a new city has been chosen for a customer b_j, new coordinates are obtained for the customer by sampling the location from a normal distribution centred in the coordinates of the city. The new cost for customer b_j is computed, which is the Euclidean distance between the new customer's location and facility locations. The service cost of each customer b_j is added to the total cost for each period of t_{max}. The total costs are then actualised at the end of the simulation at a discount rate of dr. The steps of a simulation are outlined in Algorithm 6.

Algorithm 6 Simulation Model

```
Require: $A$, $B$, $t_{max}$
  for Each Simulation do
    Generate attraction rate $a'$ for each city $a_i \in A$
    for Each customer $b_j \in B$ do
      Set of movement dates: $M = \emptyset$
      $t = 0$
      while $t < t_{max}$ do
        $t = t + \mathcal{N}(mr \cdot t_{max}, 0.1 \cdot t_{max})$
        $M = M \cup t$
      end while
      for  $t_y$ in 1 to $t_{max}$ do
        if $t_y \in M$ then
          Choose a new city for the customer based on $A'$
          Generate new location for customer in the new city
          Update cost for servicing customer based on open facilities
        end if
        Add cost of servicing customer $b_j$ to total cost for year $t_y$
      end for
    end for
    Actualise costs obtained for $t_{max}$ using discount rate $dr$
  end for
```

In Section 4.2.2, we present two ways of measuring how good a solution is to DC-LA problem.

4.2.2 Measure of the goodness of a solution to DC-LA problem

To assess the necessity of simulating future movements of customers, we introduce two evaluation methods for measuring how good a solution is to DC-LA problem. The first evaluation function forms the baseline evaluation for comparison and consists in using a deterministic function which assumes that customers will not move over time. We refer to this function as static evaluation. The decision variables are represented by a binary string $x = \{x_1, x_2, ..., x_m\} \in \{0,1\}^m$ where 1 represents an opened facility and 0 represents a closed facility. c_i is the cost of opening a facility. We formulate the static evaluation as:

$$f_{static}(x) = \sum_{i=1}^{m} c_i x_i + C_0(x) + \sum_{t=1}^{t_{max}} C_0(x)(1+dr)^{-t} \tag{4.2}$$

Where C_0 is the connection cost of each customer to an opened facility.

$$C_0(x) = \sum_{i=1}^{m} \sum_{j=1}^{n} d_{ij} x_{ij} \tag{4.3}$$

where $x_{ij} = 1$ if customer j is connected to facility i and $x_{ij} = 0$ if not.

Subject to:

$$\sum_{i\,=1}^{m} \sum_{j\,=1}^{n} x_{ij} = 1 \tag{4.4}$$

Here the cost function C_0, which is the connection costs of customers is assumed to be deterministic. Service costs of customers for times $\{t_1, t_2, ..., t_{max}\}$ are actualised and discounted using the discount rate *dr*. The static evaluation function is motivated by the current trend in the telecommunication industry in locating facilities which are mostly informed by current parameters such as the concentration of customers without consideration to how customers will evolve. The pseudo-code of the static evaluation method is presented in algorithm 7.

Algorithm 7 Static evaluation method

```
Require: $x$, $B$, $t_{max}$
  for Each Solution $x$ do
    for Each Customer $b$ do
      for $t_y$ in 0 to $t_{max}$ do
        if $t_y == 0$ then
          Allocate customer $b_j$ in $B$ to an open facility $a_i$ in $A$ that has the least
          distance from $b_j$
          Calculate the fitness of $x$ at $t_0$ by summing up the facility costs of all
          opened facility $a_i$ and the cost (distance) between $a_i$ and all $b_j$
        else
          Add cost (distance) of servicing customer $b_j$ to total cost for year $t_y$
        end if
      end for
    end for
    Actualise costs obtained for $t_{max}$ using discount rate $dr$
  end for
```

However, in the real world, several factors will influence a customer to move between cities. We are therefore motivated to adopt an evaluation method that captures how customers move and the frequency with which they move. The future possible movements of customers necessitates the dynamic stochastic evaluation where a solution to the problem is evaluated against a number of scenarios. The concept of ROOT is emphasized here, where the facilities are open at the start of the defined period and remain open over the defined period. Customer movements are then simulated over the defined period and evaluation is made concerning the opened facilities. The decision variables are represented by a binary string $x = \{x_1, x_2, ..., x_m\} \in \{0, 1\}^m$

where 1 represents an opened facility and 0 represents a closed facility. The dynamic approach reformulates Equation 4.2 as:

$$f_{dynamic}(x) = \sum_{i=1}^{m} c_i x_i + C_0(x) + E\left[\sum_{t=1}^{t_{max}} C_t(x)(1+dr)^{-t}\right] \tag{4.5}$$

Subject to equations 4.3 and 4.4.

Where the opening cost of facilities and the connection cost $C_0(x)$ is a deterministic function. $E\left[\sum_{t=1}^{t_{max}} C_t(x)(1+dr)^{-t}\right]$ represents the expected discounted service costs of customers for times $\{t_1, t_2, ..., t_{max}\}$. dr is a discount rate, typically applied to allow comparison of costs incurred at different times. A pseudo-code of the dynamic evaluation is presented in Algorithm 8.

Algorithm 8 Dynamic evaluation method

```
Require: $x$, $B$, $t_{max}$
  for Each Solution $x$ do
    for Each Customer $b$ do
      for $t_y$ in 0 to $t_{max}$ do
        if $t_y == 0$ then
          Allocate customer $b_j$ in $B$ to an open facility $a_i$ in $A$ that has the least distance from $b_j$
          Calculate the fitness of $x$ at $t_0$ by summing up the facility costs of all opened facility $a_i$ and the cost (distance) between $a_i$ and all $b_j$
        else
          Generate attraction rates $A'$ for each city $a_i$
          Generate movement dates of customer $M$ from Algorithm 5
          Add cost (distance) of servicing customer $b_j$ to total cost for year $t_y$
          if $t_y \in M$ then
            Choose a new city for the customer based on $A'$
            Generate new location for customer in the new city
            Update cost for servicing customer $b_j$ to total cost for year $t_y$
          end if
        end if
      end for
    end for
    Actualise costs obtained for $t_{max}$ using discount rate
  end for
```

4.2.3 Problem Generation

An instance of DC-LA problem comprises of three main parameters: the number of facility locations m, number of customer n, and the movement rate mr. The choice of these parameters is set in consultation with experts from the telecommunications industry. To generate an instance of DC-LA problem, we first uniformly generate cities locations and their attraction rate randomly. Based on those locations, customers are then iteratively generated by randomly selecting a city (based on the attraction rates). In a city, the coordinates of a customer are obtained by sampling its location from a normal distribution centred in the coordinates of the city. A pseudo-code for generating a problem instance is presented in Algorithm 9:

Algorithm 9 Generation of a problem instances for DC-LA problem

```
Require: Movement rates: $mr$, Number of facilities: $m$, Number of customers: $n$,
    Attraction rates $A'$
  for each $mr$ do
    Uniformly generate $m$ cities locations where coordinates of $m \in \{0, 1\}$
    Generate attraction rates for cities using Algorithm 4
    for each city $a_i$ do
      for (q=1 to ($a_i^{'}$ * $n$)) do
        Generate customers locations from a normal distribution with $a_i$ as the mean
          and standard deviation of 0.1.
      end for
    end for
  end for
```

4.3 Experimental Setup

Table 4.1 presents a summary of the values of the parameters and environmental constants described in Section 4.2.1.

Table 4.1: Parameters for defining DC-LA problem

DC-LA problem Parameters	Description	Values
mr	Movement rate	0.25,0.5,0.75,1.0
m	Number of facility locations	10,20,50,100
n	Number of customer locations	100, 500 , 1000
$a^{'}$	attraction rate	randomised
dr	Discount rate	0.05
sd for mr	SD for movement dates	0.1
sd for n	SD for customer locations	0.1
t_{max}	Defined horizon	30

By varying the parameters m,n and mr we generate 48 problem configurations. For each of the 48 configurations, we generate 30 instances creating a benchmark of 1440 problems. The cost of a facility is set as m/n, the rationale for setting the cost of the facility is to try and minimise the number of facilities to open considering that the cost of opening facilities incur substantial costs. In this Section, we seek to observe the influence the parameters defined may have on the problem. We do this by running an initial set of experiments with a focus on each of the problem parameters. Results and observations of influence for each parameter are presented below.

To examine how the distribution of customers changes between facilities over the length of a simulation, we apply a measure called the Earth movers distance (EMD) [108]. EMD is a measure of the distance between two distributions over a region D. In simple terms, EMD is the minimum amount of effort requires to convert one distribution into the other. In order to compute the EMD between the distribution of customers to facilities for each year of the simulation, we first rank facilities according to the number of customers they serve. The facility servicing the higher number of customers gets the rank one, the facility servicing the second higher number of customers gets the rank two, and so on. For each year of the simulation, facilities are ranked according to the total number of customers they serve. The EMD is then calculated between the ranks of facilities at the beginning of the simulation, i.e. year$_0$ and year$_{t=1}$ to year$_{t_{max}}$. The bigger the distance recorded, the more significant the difference between the ranks of facilities at year$_t$ and year$_0$. Algorithm 10 shows how EMD is computed.

Algorithm 10 Computing EMD for rank of cities

```
Require: P= Ranks of facilities at $year_{t=0}$, Q= Ranks of facilities at $year_{t=1}$
  $TotalDistance = 0$
  $EMD_0 = 0$
  for $i = 0$ to (P.length-1) do
    $EMD_{i+1} = (P_i + EMD_i) - Q_i$
    $TotalDistance = \sum |EMD_i|$
  end for
```

4.4 Experimental Results and Analysis

In this Section, we present and analyse the results obtained from experiments on the 1440 problem instances. We aim to understand the effects the parameters of DC-LA problem exerts on the problem instances.

4.4.1 Effects of Movement Rate (mr)

To observe the effect of movement rate parameter, we conduct an exhaustive evaluation of all solutions to the problem configurations with the smaller number of facilities, i.e. 10 facilities. This is because an exhaustive evaluation of solutions for more than 10 facilities can be computationally costly. For example, an exhaustive search for a problem with a facility size of 20 will require us to evaluate 2^{20} (i.e. 1,048,576) solutions. However, a problem instance with 10 facilities is more manageable as we will only be required to evaluate 2^{10} (i.e. 1024) solutions. We therefore run an exhaustive search for all 30 instances of m_10 by n_100, m_10 by n_500 and m_10 by n_1000.

Table 4.2 shows the sensitivity of the movement rates. The Table shows the correlation of costs between the movement rates when compared to a static problem, i.e. when customers are assumed to make no movements between cities over time. Tables 4.3 and 4.4 shows the ranks of solutions of all 30 problems for problem configuration m_10 by n_100, m_10 by n_500 and m_10 by n_1000 respectively. For the dynamic evaluation, we consider four mr giving us four dynamic evaluation functions. Hence, for all problems, there are five ways of evaluating a solution, the static method which is deterministic, and the dynamic method with mr_0.25, mr_0.50, mr_0.75 and mr_1.0. For each solution, we obtain fitness using each of the five evaluation functions. We use 5000 simulations to evaluate a solution to problems where customers are assumed to make movements because it is stochastic and return the median of the 5000 distribution as the fitness of the solution. For each of the method, we ranked the solutions giving us a permutation of rankings of solutions. Then we performed an experiment correlation on all the ranks. The results presented in the Tables 4.2, 4.3 and 4.4 below are the correlation between the static evaluation function and the dynamic evaluation functions on m_10 by n_100, m__10 by n_500 and m_10 by n_1000 respectively.

Table 4.2: Correlation of results for static and dynamic methods of evaluation

m	n	Problem	$mr0.25$	$mr0.5$	$mr0.75$	$mr1.0$
10	100	p1	0.18	0.16	0.19	0.35
10	100	p2	0.16	0.15	0.21	0.48
10	100	p3	0.11	0.09	0.16	0.50
10	100	p4	0.22	0.27	0.37	0.65
10	100	p5	0.43	0.50	0.50	0.69
10	100	p6	0.64	0.66	0.70	0.80
10	100	p7	0.06	0.11	0.17	0.48
10	100	p8	0.13	0.09	0.11	0.33
10	100	p9	0.12	0.15	0.12	0.34
10	100	p10	0.23	0.19	0.29	0.50
10	100	p11	0.36	0.38	0.46	0.65
10	100	p12	0.19	0.22	0.25	0.53
10	100	p13	0.16	0.12	0.14	0.39
10	100	p14	0.45	0.49	0.54	0.62
10	100	p15	0.33	0.28	0.35	0.55
10	100	p16	0.17	0.18	0.32	0.70
10	100	p17	0.70	0.66	0.69	0.74
10	100	p18	0.32	0.32	0.39	0.57
10	100	p19	0.03	0.12	0.11	0.40
10	100	p20	0.18	0.17	0.14	0.36
10	100	p21	0.07	0.08	0.09	0.36
10	100	p22	0.21	0.22	0.26	0.45
10	100	p23	0.19	0.21	0.23	0.43
10	100	p24	0.19	0.18	0.23	0.54
10	100	p25	0.37	0.35	0.41	0.61
10	100	p26	0.25	0.28	0.26	0.46
10	100	p27	0.46	0.46	0.50	0.75
10	100	p28	0.29	0.22	0.31	0.49
10	100	p29	0.33	0.32	0.36	0.46
10	100	p30	0.56	0.56	0.61	0.69

Table 4.3: Correlation of results for static and dynamic methods of evaluation

m	n	Problem	$mr0.25$	$mr0.5$	$mr0.75$	$mr1.0$
10	500	p1	0.23	0.24	0.26	0.73
10	500	p2	0.43	0.46	0.53	0.84
10	500	p3	0.13	0.18	0.25	0.54
10	500	p4	0.46	0.49	0.54	0.79
10	500	p5	0.11	0.19	0.30	0.73
10	500	p6	0.21	0.21	0.31	0.79
10	500	p7	0.18	0.19	0.28	0.68
10	500	p8	0.26	0.21	0.31	0.62
10	500	p9	0.23	0.28	0.41	0.84
10	500	p10	0.17	0.20	0.20	0.59
10	500	p11	0.53	0.52	0.60	0.84
10	500	p12	0.28	0.28	0.34	0.74
10	500	p13	0.17	0.22	0.17	0.55
10	500	p14	0.26	0.34	0.44	0.75
10	500	p15	0.38	0.40	0.40	0.64
10	500	p16	0.39	0.37	0.52	0.75
10	500	p17	0.45	0.48	0.52	0.75
10	500	p18	0.48	0.50	0.50	0.61
10	500	p19	0.38	0.39	0.44	0.71
10	500	p20	0.16	0.13	0.17	0.67
10	500	p21	0.25	0.31	0.37	0.73
10	500	p22	0.20	0.18	0.27	0.60
10	500	p23	0.15	0.21	0.25	0.67
10	500	p24	0.19	0.23	0.24	0.62
10	500	p25	0.12	0.19	0.26	0.61
10	500	p26	0.30	0.31	0.37	0.70
10	500	p27	0.13	0.21	0.17	0.62
10	500	p28	0.17	0.24	0.30	0.58
10	500	p29	0.16	0.14	0.19	0.59
10	500	p30	0.09	0.14	0.18	0.53

Table 4.4: Correlation of results for static and dynamic methods of evaluation

m	n	Problem	$mr0.25$	$mr0.5$	$mr0.75$	$mr1.0$
10	1000	p1	0.38	0.38	0.43	0.73
10	1000	p2	0.36	0.42	0.52	0.82
10	1000	p3	0.57	0.61	0.70	0.91
10	1000	p4	0.45	0.44	0.52	0.85
10	1000	p5	0.37	0.40	0.44	0.64
10	1000	p6	0.28	0.26	0.42	0.79
10	1000	p7	0.50	0.50	0.53	0.82
10	1000	p8	0.44	0.40	0.50	0.75
10	1000	p9	0.23	0.26	0.39	0.79
10	1000	p10	0.45	0.48	0.55	0.85
10	1000	p11	0.23	0.29	0.37	0.74
10	1000	p12	0.21	0.23	0.43	0.78
10	1000	p13	0.33	0.33	0.45	0.79
10	1000	p14	0.19	0.22	0.25	0.59
10	1000	p15	0.21	0.25	0.30	0.70
10	1000	p16	0.19	0.21	0.25	0.62
10	1000	p17	0.17	0.24	0.32	0.71
10	1000	p18	0.14	0.16	0.21	0.61
10	1000	p19	0.29	0.34	0.39	0.73
10	1000	p20	0.45	0.46	0.51	0.80
10	1000	p21	0.20	0.25	0.38	0.76
10	1000	p22	0.18	0.19	0.33	0.70
10	1000	p23	0.15	0.21	0.27	0.71
10	1000	p24	0.22	0.24	0.33	0.71
10	1000	p25	0.15	0.19	0.29	0.71
10	1000	p26	0.48	0.46	0.49	0.75
10	1000	p27	0.27	0.35	0.47	0.84
10	1000	p28	0.29	0.35	0.45	0.72
10	1000	p29	0.31	0.36	0.41	0.66
10	1000	p30	0.47	0.48	0.52	0.78

4.4.2 Visual representation of the effects of Movement Rate (mr)

Figures 4.2, 4.3 and 4.4 shows the visual representation of the correlations results presented in Tables 4.2, 4.3 and 4.4 for 10 facilities by 100 customers, 10 facilities by 500 customers and 10 facilities by 100 customers.

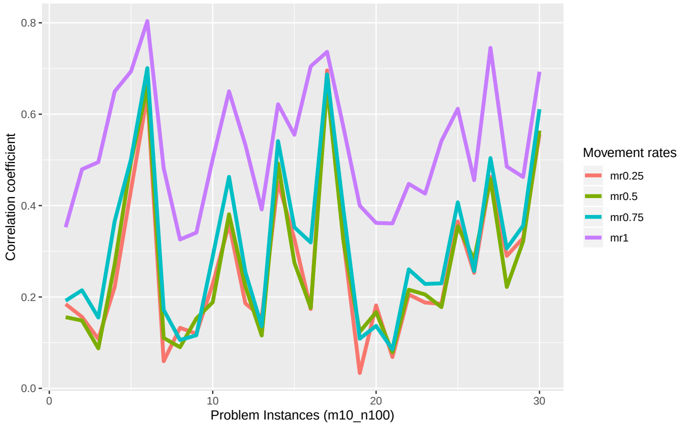


Figure 4.2: Correlation between static and dynamic evaluations for movement rates on 10 facilities by 100 customers

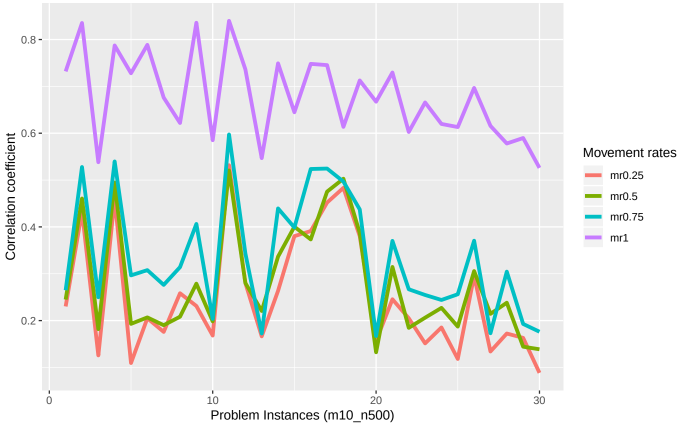


Figure 4.3: Correlation between static and dynamic evaluations for movement rates on 10 facilities by 500 customers

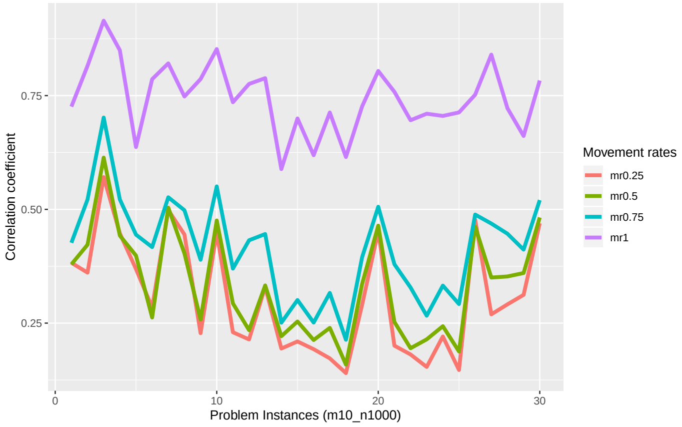


Figure 4.4: Correlation between static and dynamic evaluations for movement rates on 10 facilities by 1000 customers

From the Tables 4.2, 4.3, 4.4 and Figures 4.2, 4.3 and 4.4, we observe that on most problem instances for problem configuration of $m10\_n100$ there exist a low correlation between dynamic and static when customers are assumed to make frequent movement over the defined horizon. However, we also observe that when customers are assumed to make little or no movement over the defined period, a low correlation exists between static and dynamic on most problem instances. A look at problem configurations $m10\_n500$ and $m10\_n1000$ shows an increasing correlation between static and dynamic when customers are assumed to make little or no movement over the defined horizon.

From the results presented in Tables 4.2, 4.3, 4.4 and Figures 4.2, 4.3 and 4.4 shows that the difference in correlation between results of static and dynamic is influenced by the number customers. A small number of customers appears create varying distributions which creates low correlation between results obtained by static and dynamic especially in situations where customers are assumed to make frequent movement over a defined period. However, a more significant number of customers appears to create less different distributions, especially when customers are assumed to make little or no movement over a defined period. Hence, we observe a higher correlation between the results obtained by static and dynamic in scenarios where customers are assumed to make little or no movement over the defined period. We also observe that for problem configuration with m10_n100 the distance between $mr0.25$ and $mr1$ is lesser when compared to problem configurations with a more significant number of customers showing that a smaller number of customers creates more varying distribution across all movement scenarios of customers.

A visual representation of the sensitivity of the movement rates are shown in Figures 4.5, 4.6,4.7 and 4.8. Each Figure shows the movement distribution of customers over 20 simulations across 30 years for an open set of facilities.

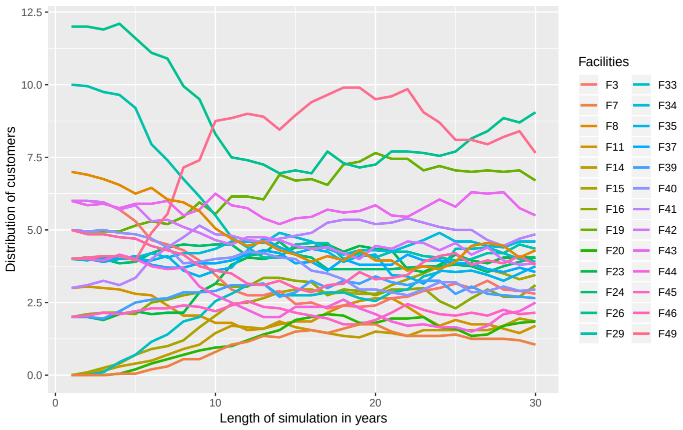


Figure 4.5: Simulation of customers movements at $mr = 0.25$ with 50 facilities 100 customers

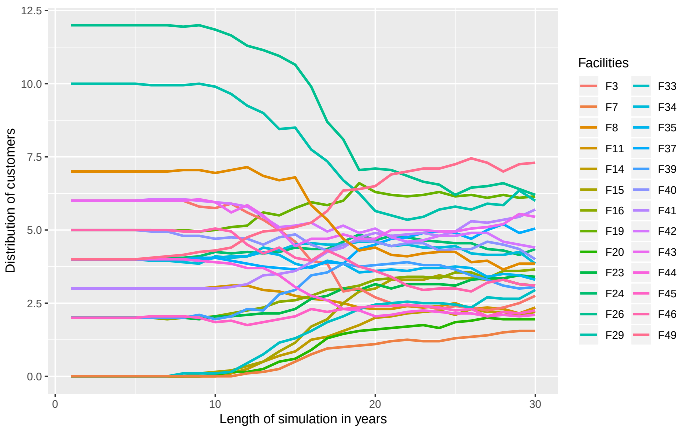


Figure 4.6: Simulation of customers movements at $mr = 0.5$ with 50 facilities 100 customers

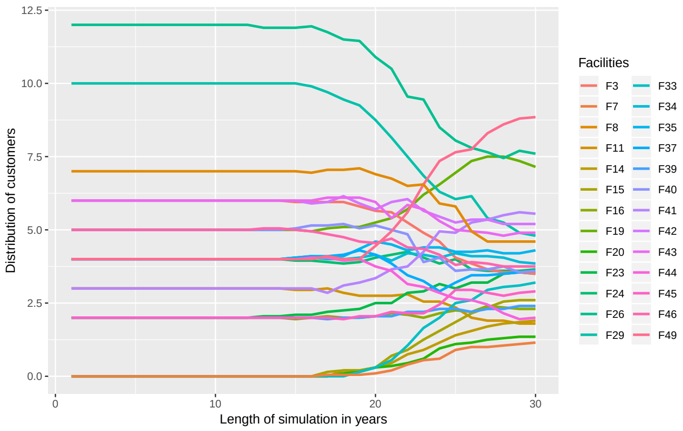


Figure 4.7: Simulation of customers movements at $mr = 0.75$ with 50 facilities 100 customers

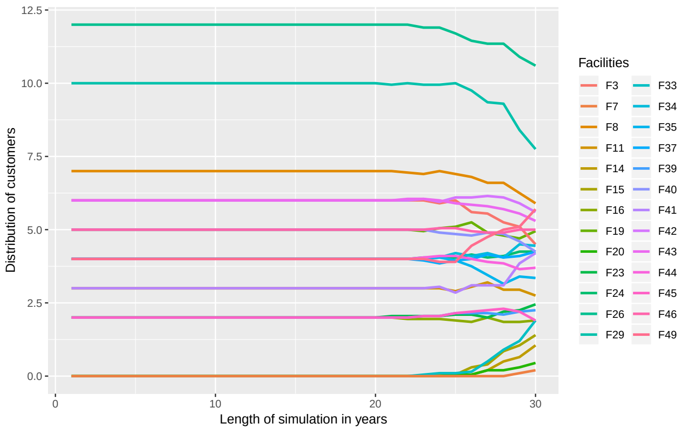


Figure 4.8: Simulation of customers movements at $mr = 1.0$ with 50 facilities 100 customers

A look at Figure 4.5 to 4.8 clearly shows how the movement rates impact the movement of customers over the defined horizon. The smaller the value of mr the more movement a customer makes across simulation and hence creates more varying costs while the higher the value of mr the less movement a customer makes creating less varying costs.

4.4.3 Effects of the number of facilities (m)

To understand what effect the number of facilities has on a problem, we calculate the EMD between the ranks of cities over the length of a simulation averaged over 20 simulations. The EMD is calculated for facilities 10,20,50 and 100 using 100 customers. We show the effect observed in Figure 4.9

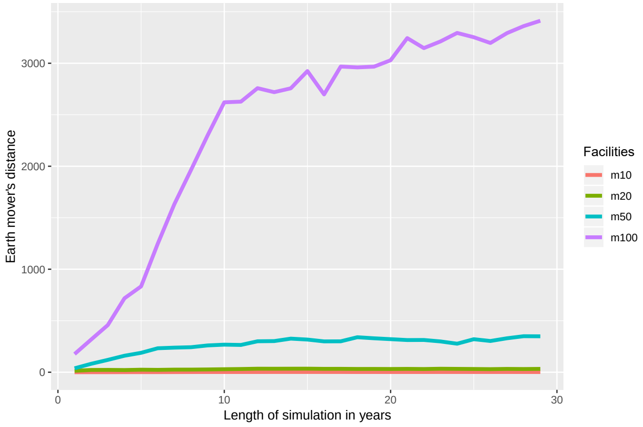


Figure 4.9: Earth movers distance between the number of facilities over the length of simulation

In Figure 4.9 we observe that as the number of facilities increases, the distance between the ranks of cities at the start of the simulation and ranks of cities over the length of the simulation also increases. Figure 4.9 shows that the larger the number of facilities, the more variations in customers distributions are observed creating more varying costs. This is because a larger number of facilities creates unevenly distributed customers over the defined period.

4.4.4 Effects of the number of customers (n)

In Figure 4.10, we show the EMD calculated between the ranks of cities over the length of a simulation averaged over 20 simulations for the number of customers. The EMD is calculated for 100, 500 and 1000 customers using the same number of facilities (50).

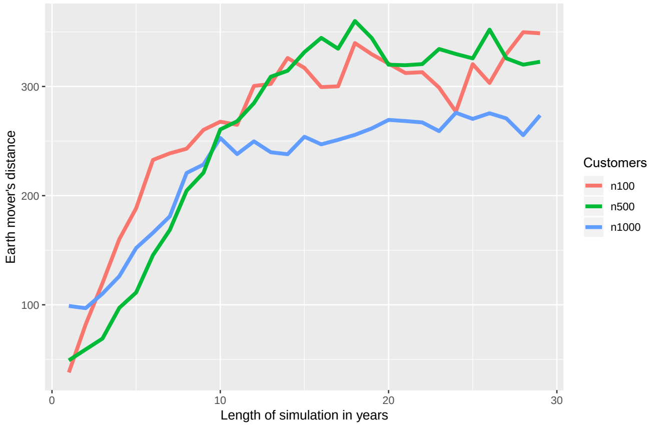


Figure 4.10: Earth movers distance between the number of customers over the length of simulation

From Figure 4.10, we observe that more variations in the distribution of customers are observed for 100 and 500 customers. For the more significant number of customers, we observe that the difference in the distribution of customers tends to even out after some years. This means that a smaller number of customers tend to create more variations in customer movement over the length of the simulation. This observation is also evident in Tables 4.2,4.3 and 4.4 where we observe that for a more significant number of customers the correlation between the results obtained by the static evaluation and the dynamic evaluation when a customer is assumed to make fewer movement over the defined period is higher when compared with a smaller number of customers. This shows that a more significant number of customers movements (i.e. 1000 customers) turns to even out at the end of the simulation and hence create less varying costs especially when customers are assumed to make little or no movement over the defined period.

4.4.5 Effect of attractive rates

To understand the impacts the attractive rates of cities make to simulation, we calculate the earth movers distance (EMD) between the ranks of facilities over the length of the simulation. In Figure 4.11, we simulate 20 customer movements and calculate the EMD between the ranks of cities when the attraction rate is different for every simulation. In Figure 4.12, we simulate 20 customer movements and calculate the EMD between the ranks of cities when the attraction rate is the same for all 20 simulations.

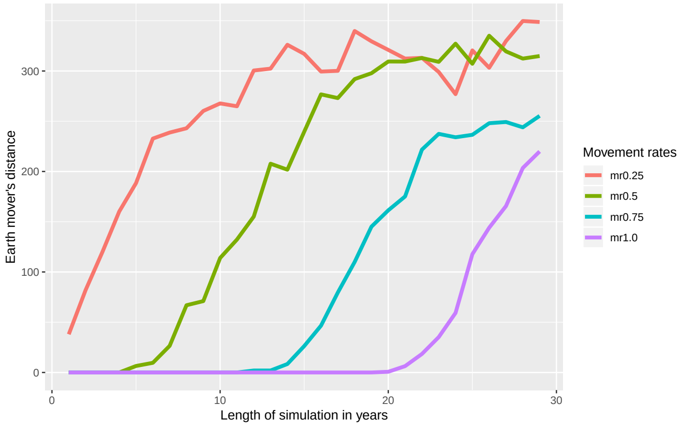


Figure 4.11: Simulation of customers movements with different attractive rates with 50 facilities 100 customers

From Figure 4.11, we observe that when the attraction rates of cities are different for each simulation, there is more variation in the movement of customers. Throughout a simulation, we observe that the difference in ranks between cities at the start of the simulation and ranks of cities at the later stage of the simulation keeps increasing showing a significant difference in the distribution of customers over the simulation. The opposite effect is seen in Figure 4.12, where a smaller difference is observed between the ranks of cities at the start of the simulation and the ranks of cities over the length of the simulation. In particular, we observe that for later years of the simulation, the difference between the ranks of cities appears to decrease, showing a little variation of customer distribution between the start and end of the simulation. Because the same attraction rates are fixed for all simulations, the only variations in the distribution of customers are created from customers movements. Hence there is a likelihood of customers movements in a later simulation to resemble earlier simulation causing low variations in costs.

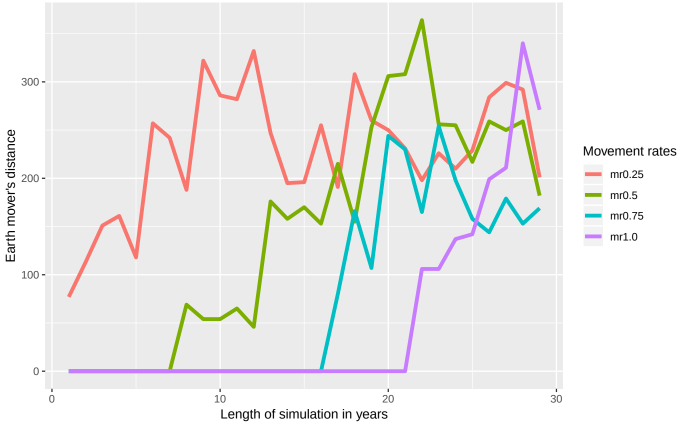


Figure 4.12: Simulation of customers movements with the same attractive rates with 50 facilities 100 customers

By using different attraction rates of cities for each simulation, we can ensure that the distributions of customers of later simulations are different from earlier simulations. By doing this, we will have different scenarios by which to evaluate our solutions.

4.5 Summary

In this Chapter, we introduced and defined the Dynamic-Customer location-Allocation (DC-LA) problem. DC-LA problem considers the potential movement of customers over a given time horizon t_{max}. Dc-LA problem is primarily defined by three main parameters; these are movement rate which defines the rate at which customers will move locations over a defined period; Number of facilities and number of customers. In DC-LA problem, the pattern of customer location changes is assumed to be stochastically driven by the attractivity of cities, i.e. how attractive a city is to a customer. In order to model the future expected changes in customers movements, we presented and discussed a stochastic model for simulating future events for each customer. The simulation model works by assigning each city an attractive rate at the start of the simulation. Customers are then assigned to cities based on how attractive the city is to the customer.

To evaluate solutions to DC-LA problem, we presented and discussed two evaluation methods: dynamic and static evaluation functions. The dynamic evaluation function assumes that customers will move locations over a defined period and account for customer movement in its evaluation process. The static evaluation function forms the baseline for comparison with the dynamic evaluation functions and assumes that customers will not move over the defined time.

In order to study our DC-LA problem, we generated 1440 problem instances by combining the parameters of movement rates mr, the number of facilities m and the number of customers n. We then conducted an initial set of experiments to observe and understand the influence the different parameters exerted on DC-LA problem. Our aim in the experiment was to understand if there was a justification to expend extra computational effort to simulate the movement of customers when deciding the location of facilities or can we make good decisions concerning the location of facilities using the static evaluation which requires less computational effort because we do not need to simulate future movements of customers.

From experiments, we observed that for movement rate, when customers are assumed to make frequent movement over a defined period, high variations in costs are recorded. These variations in costs reduce as customers are observed to make little or no movements over the defined period.

For the number of facilities, we observed that as the number of facilities increases the greater the recorded variations in costs. The variations, however, reduces for a smaller number of facilities.

For the number of customers, we observed that a smaller number of customers tend to create more variations in customer movements which creates a high variation in costs recorded over the defined period. For the bigger number of customers, we observed low variations in costs as the simulation tends to even out after some years. This is because a similar number of customers tend to replace customer who leaves a city.

Variations in costs determine how different the problem is from a static problem which assumes that customers movement will remain the same over the defined period and hence there will be little or no variations in cost. High variations in costs show that the problem changes over the defined period. This means that a static solution which does not take future changes into consideration will be an infeasible solution.

The high variations in costs observed in some scenarios give justification to experiment further to see if there is value in simulating customer movements and in what scenarios will it be worth expending the extra effort to find a robust solution to DC-LA problem. In the next Chapter, we will conduct a set of experiments on the new 1440 problem instances to assess the necessity of simulating the future movement of customers.

Chapter 5

Solution Approach to the Dynamic Customer Location-Allocation Problems

5.1 Introduction

The dynamic model of Location-Allocation (LA) problem introduced in Chapter 4 generates random customer movements based on varying attractivity of locations. To assess the necessity of simulating future movements of customers, we aim to study the DC-LA problem in this Chapter by conducting a set of experiments. To do this, we experiment with the 1440 DC-LA problem instances generated by varying the parameters of movement rates mr, the number of facilities m and the number of customers n in Section 4.3 of Chapter 4.

The experimental results in Chapter 3, showed Population-Based Incremental Learning Algorithm (PBIL) to perform better than the GA variants when tested on the CAP datasets and the new problem instance for LARP and LARPR. Hence we run PBIL described in Section 2.5.4 on the 1440 problems using the two evaluation methods described in Section 4.2.2. The first method referred to as, the dynamic evaluation, simulates customer movements to estimate the expected cost over time. The second method, called, the static evaluation, which forms the baseline for comparison with the dynamic evaluation function, assumes no customer movements and only evaluates the actualised service costs of customers. We compare the results obtained using the two evaluation methods concerning the different problem parameters to analyse the efficiency of each method. Our aim in this Chapter is to ascertain if there is value in simulating customer movements over a defined horizon when making decisions to site facilities to service changing demand bearing in mind the associated high computational cost or can we rely on the static solution to decide the location of facilities to a changing problem? The work presented in this Chapter has been published in Proceedings of the 2019 Institute of Electrical and Electronics Engineers

(IEEE) Congress on Evolutionary computation (IEEE CEC 2019), 10-13 June 2019, Wellington, NZ.

The Chapter is organised as follows. In Section 5.2 we describe the experimental setup and discuss result in Section 5.3. Finally, we conclude the Chapter in Section 5.4.

5.2 Experimental Setup

Parameters used for generating the problem instances are presented in Table 4.1 in Chapter 4. Values defined for the number of customers are informed by a purely commercial customer base often located in countries other than the united kingdom such as the United States of America. In such countries, the commercial customers such as financial institutions, data centres and engineering firms range from a couple of hundreds to a few thousand.

Combining PBIL and the two evaluation functions gives us the configurations: $PBIL_{static}$ and $PBIL_{dynamic}$ for all movement rates. However, in analysing the results obtained, we refer to $PBIL_{static}$ and $PBIL_{dynamic}$ as static and dynamic respectively. The parameters used for PBIL are presented in Table 5.1. Each run is allowed 10000 fitness evaluations.

Table 5.1: Parameters for PBIL

Parameter	Value
Population size	50
Fitness evaluation	10000
PBIL learning rate	0.1
Truncation size	0.5

This experiment aims to ascertain whether it is worth taking customer movements over a defined horizon into account when deciding to locate facilities to service customer needs. Considering that the two evaluation functions evaluate solutions differently, and the fact that we seek to investigate if there is value in employing the more expensive dynamic evaluation function, we evaluate the best solution found at the end of each run with *dynamic* over 5000 simulations to allow for proper comparison of the two evaluation functions. We run each evaluation for each of the 1440 problems 20 times. Experiments were performed on an Intel(R) Xeon(R) E5620 @2.40GHz cluster.

Due to the stochastic nature of PBIL and the fact that we hope to compare only two methods i.e. static and dynamic, we employ the Wilcoxon signed ranks test to assess the performance of static and dynamic. Our choice of the Wilcoxon signed-rank test is motivated by the fact that within the statistic community, the Wilcoxon signed ranks test is preferred and considered to be safer when compared to the t-test because the test does not assume normal distributions. Outliers (particularly good or bad performances of a few problems) also tend to have less of an effect on the Wilcoxon signed ranks test than on the t-test.

5.2.1 The Wilcoxon signed ranks test

The Wilcoxon signed ranks test is a non-parametric method used in hypothesis testing conditions, comprising a study with two samples. The Wilcoxon signed ranks test is employed to answer the question: do two samples denote two diverse populations? The Wilcoxon signed ranks test is comparable to the paired t-test in non-parametric statistical procedures; hence, it is a pairwise test that strives to recognise significant differentiation among two samples averages, i.e., the performance of two algorithms.

The Wilcoxon signed ranks test is described as follows: d_i denote the variations in the performance scores of two algorithms on ith out of n problems. In the case where the scores of the algorithms lie in different ranges, to prevent prioritising one problem over the other, these scores can be normalised to the interval [0,1] [65].

d_i, which denotes the differences in the scores of the algorithms being compared are ranked concerning their absolute values. In the event of a tie, computing the average ranks is a recommended approach. For instance, if two d_i are tied in the allocation of ranks 1 and 2, then the rank 1.5 is assigned to both d_i. However, one can also adopt some of the available approaches in the literature such as [68] which disregards ties and designates the highest ranks and then calculate all the potential assignments and compute the average of the results gained within the test. The aggregate of ranks on the problems for algorithm one is denoted as R^+. The aggregate of ranks on the problems for algorithm two is denoted as R^-. When $d_i = 0$, the rank is divided equally between the sums; however, one is discarded if there is an odd number of the ranks.

$$R^+ = \sum_{d_i>0} rank(d_i) + \frac{1}{2}\sum_{d_i=0} rank(d_i) \tag{5.1}$$

$$R^- = \sum_{d_i<0} rank(d_i) + \frac{1}{2}\sum_{d_i=0} rank(d_i) \tag{5.2}$$

T denotes the least of the sums, T = min(R+, R). If T is smaller than or equivalent to the value of the distribution of Wilcoxon for n degrees of freedom [176], the null hypothesis which states that the means of the two algorithms are the same is rejected. The rejection implies that one algorithm of the two performed better than the other algorithm with the associated *p-value*. In practice, the Wilcoxon signed ranks test is considered to be more sensitive when compared to the t-test.

5.3 Experimental Results and Analysis

In this Section we present and discuss the effects DC-LA problem parameters exerts on DC-LA problem instances. We seek to determine if there is value in expending the extra computational effort to simulate the movements of customers over a defined period when deciding the locations of facilities.

We perform the Wilcoxon signed ranks test described in Section 5.2.1 on the results obtained from the experiments on all 30 problems for each of the 48 problem configurations. We aim to determine if there is a significant differentiation between the averages of results obtained using the static and dynamic evaluation functions.

The results of the Wilcoxon signed-rank test are shown in Table 5.2. In Table 5.2, the aggregate of ranks on the problems for dynamic evaluation is denoted as R^+ and the aggregate of ranks on the problems for static evaluation is denoted as R^-. For each problem configuration, the aggregate of ranks for the problem configurations on which an evaluation function performed better than the other is highlighted in bold. Problem configurations are represented by movement rate mr, the number of facilities m and the number of customer n. We employ a 95% confidence level for the test. A 95% confidence level means that if the p-value obtained by an evaluation function, for example, static evaluation, on a problem configuration is less than the significance level α=0.05, then the results obtained by the static evaluation function is deemed to be statistically significant when compared to the results obtained by dynamic on the same problem configuration. In Table 5.2 we obtain statistical difference for 21 out of the 48 problem configurations.

Table 5.2: Wilcoxon comparison of dynamic vs static evaluations grouped by configurations of the DC-LA problem

			Dynamic	Static		Statistical
mr	m	n	$R+$	$R-$	$p - value$	difference?
0.25	10	100	**210**	0	9.57E-05	**Yes**
0.25	10	500	**111**	9	4.13E-03	**Yes**
0.25	10	1000	**50**	5	2.49E-02	**Yes**
0.25	20	100	**422**	13	1.03E-05	**Yes**
0.25	20	500	**218**	35	3.13E-03	**Yes**
0.25	20	1000	**321**	57	1.58E-03	**Yes**
0.25	50	100	**465**	0	1.86E-09	**Yes**
0.25	50	500	**462**	3	9.31E-09	**Yes**
0.25	50	1000	**449**	16	3.15E-07	**Yes**
0.25	100	100	**465**	0	1.86E-09	**Yes**
0.25	100	500	**464**	1	3.73E-09	**Yes**
0.25	100	1000	**451**	14	2.05E-07	**Yes**
0.5	10	100	**105**	15	1.15E-02	**Yes**
0.5	10	500	**51**	27	3.67E-01	No
0.5	10	1000	**22**	6	2.05E-01	No
0.5	20	100	**269**	109	5.61E-02	No
0.5	20	500	82	**128**	4.01E-01	No
0.5	20	1000	**210**	90	8.91E-02	No
0.5	50	100	**464**	1	3.73E-09	**Yes**
0.5	50	500	**419**	46	3.45E-05	**Yes**
0.5	50	1000	**356**	109	9.93E-03	**Yes**
0.5	100	100	**460**	5	1.86E-08	**Yes**
0.5	100	500	**371**	94	3.48E-03	**Yes**
0.5	100	1000	**300**	165	1.71E-01	No
0.75	10	100	**40**	5	4.40E-02	**Yes**
0.75	10	500	**9**	1	2.01E-01	No
0.75	10	1000	**10**	5	5.90E-01	No
0.75	20	100	81	**109**	5.87E-01	No
0.75	20	500	12	**93**	1.20E-02	**Yes**
0.75	20	1000	**64**	56	8.42E-01	No
0.75	50	100	**372**	93	3.22E-03	**Yes**
0.75	50	500	213	**252**	7.00E-01	No
0.75	50	1000	222	**243**	8.39E-01	No
0.75	100	100	**305**	160	1.40E-01	No
0.75	100	500	205	**260**	5.84E-01	No
0.75	100	1000	190	**275**	3.93E-01	No
1	10	100	**0**	**0**	0.00E+00	No
1	10	500	**2**	1	1.00E+00	No
1	10	1000	0	**3**	3.71E-01	No
1	20	100	16	**39**	2.62E-01	No
1	20	500	9	**36**	1.24E-01	No
1	20	1000	12	**33**	2.36E-01	No
1	50	100	**247**	218	7.77E-01	No
1	50	500	182	**283**	3.09E-01	No
1	50	1000	**262**	203	5.56E-01	No
1	100	100	**263**	202	5.43E-01	No
1	100	500	**250**	215	7.30E-01	No
1	100	1000	**248**	217	7.61E-01	No

From the Table 5.2, it is observed that the dynamic evaluation function achieves a statistical significance in results over the Static evaluation function when customers are assumed to make frequent movements over the defined period. An opposing effect is seen as customers make fewer movements over the defined period. From the aggregate of ranks for all problem configurations, it can be observed that even on most problems where no statistical difference was recorded between the dynamic and static evaluation, the dynamic evaluation shows more favourable results than the static evaluation. Based on the observation of results from Table 5.2 we can conclude that in scenarios where customers are assumed to make frequent movements over the planning period there is value in employing the dynamic evaluation function to decide the locations of customers. In scenarios where customers are assumed to make little or no movements over the defined period, the dynamic evaluation is still recommended if the computational cost involved is not an issue because the dynamic evaluation function achieves better results even if the results between dynamic and static evaluations are mutually statistically indistinguishable. However, in the situation where the computational cost is of concern, then using the static evaluation function will offer similarly better results.

Although the movement of customers seems to be the primary drive in the performance of the dynamic evaluation function, results from the Table show that other parameters such as the number of facilities and the number of customers appear to have an impact on the performance of the evaluation functions. To better understand how the parameter values influence the performance of the evaluation functions we discussed them into details in Section 5.3.1

5.3.1 Analysis of the effects of DC-LA problem parameters on problem instances

To help in analysing the influence of the problem parameters on the performance of the evaluation functions we present Table 5.3 which shows the number of wins, losses and ties of the dynamic evaluation overall 30 instances of the 48 problem configurations when compared to the static evaluation function. On each problem configuration, the highlighted value shows the wins, loses or ties of the dynamic evaluation function on a problem configuration.

Table 5.3: Wins, Losses and Ties of dynamic evaluation when compared to static evaluation grouped by configurations of DC-LA problem

mr	m	n	Wins	Loss	Ties
0.25	10	100	**20**	0	10
0.5	10	100	10	5	**15**
0.75	10	100	7	2	**21**
1	10	100	0	0	**30**
0.25	10	500	12	3	**15**
0.5	10	500	6	6	**18**
0.75	10	500	3	1	**26**
1	10	500	1	1	**28**
0.25	10	1000	8	2	**20**
0.5	10	1000	4	3	**23**
0.75	10	1000	3	2	**25**
1	10	1000	0	2	**28**
0.25	20	100	**25**	4	1
0.5	20	100	**14**	13	3
0.75	20	100	7	**12**	11
1	20	100	2	8	**20**
0.25	20	500	**15**	7	8
0.5	20	500	8	**12**	10
0.75	20	500	1	13	**16**
1	20	500	1	8	**21**
0.25	20	1000	**18**	9	3
0.5	20	1000	**15**	9	6
0.75	20	1000	6	9	**15**
1	20	1000	2	7	**21**
0.25	50	100	**30**	0	0
0.5	50	100	**29**	1	0
0.75	50	100	**23**	7	0
1	50	100	14	**16**	0
0.25	50	500	**29**	1	0
0.5	50	500	**24**	6	0
0.75	50	500	**15**	**15**	0
1	50	500	12	**18**	0
0.25	50	1000	**26**	4	0
0.5	50	1000	**22**	8	0
0.75	50	1000	**16**	14	0
1	50	1000	**15**	**15**	0
0.25	100	100	**30**	0	0
0.5	100	100	**28**	2	0
0.75	100	100	**19**	11	0
1	100	100	**17**	13	0
0.25	100	500	**29**	1	0
0.5	100	500	**22**	8	0
0.75	100	500	14	**16**	0
1	100	500	**17**	13	0
0.25	100	1000	**28**	2	0
0.5	100	1000	**19**	11	0
0.75	100	1000	13	**17**	0
1	100	1000	**17**	13	0

5.3.2 Movement Rate (mr)

We first analyse the influence of the movement rate. As seen from Table 5.3, the necessity of using the dynamic evaluation is diminished when customers are assumed to make little or no movement over the defined period. This conclusion is affirmed by the high number of ties achieved by the dynamic evaluation of problem configurations with 10 and 20 facilities. Even for problem configurations with 50 and 100 facilities, the dynamic evaluation is seen to achieve more loss when customers make little or no movements over the defined period. Little or no movement of customers makes the problem an almost static one hence the similarity in performance of the two evaluation functions. On the other hand in all problem configurations when customers are assumed to make frequent movement over the defined period the dynamic evaluation function is seen to achieve the more wins as expressed in Table 5.4 where the problem configurations are grouped according to movement rates.

Table 5.4: Wilcoxon comparison of dynamic vs static evaluations grouped by movement rate of the DC-LA problem

	Dynamic	Static		Statistical
mr	$R+$	$R-$	$p - value$	difference?
0.25	**44945**	1111	9.36E-47	**Yes**
0.5	**33002**	7753	1.24E-19	**Yes**
0.75	**15949**	14432	4.97E-01	No
1	**11437**	11141	8.69E-01	No

The necessity of using the dynamic evaluation function to locate facilities when customers are assumed to make frequent movements over the defined period is further affirmed in Figure 5.1 which shows the percentage difference in costs savings when compared to the static evaluation function. In Figure 5.1 negative values indicate that the dynamic approach resulted in better performance.

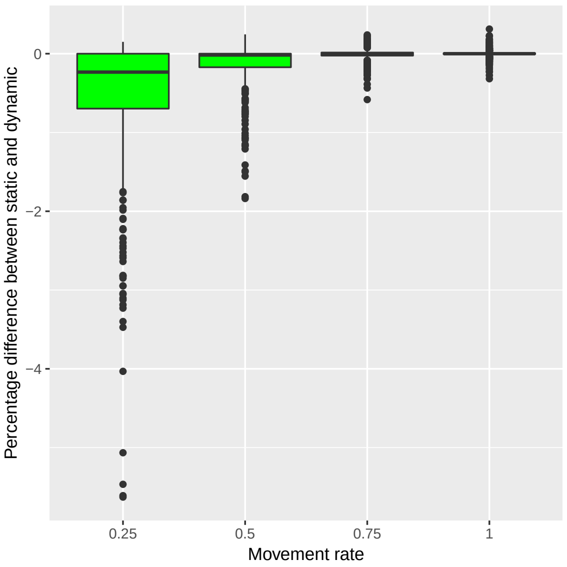


Figure 5.1: Percentage difference between the dynamic and static evaluation grouped by movement rate

Indeed when customers are assumed to make the most movements over the defined period, the dynamic evaluation achieves about 6% [1] in cost savings and decreases to 0.3% when customers make little or no movements over the defined period. Although the dynamic evaluation shows such good performance when customers make frequent movements over time, the profoundly different costs created as a result of the high movements emphasise the necessity of simulating a large number of scenarios to obtain a fitness for a solution which comes with very high computational costs as we will observe in later Sections.

5.3.3 Number of Facilities (m)

Secondly, the number of facilities is seen to influence the performance of the evaluation functions. From Table 5.3 we observe that the number of wins achieved by the dynamic evaluation is less for problem configurations with a smaller number of facilities, i.e. 10 facilities and more pronounced for problem configurations with a more significant number of facilities such as 50 and 100 facilities. However, irrespective of the problem size the dynamic evaluation achieves a significant difference in results when compared to the static evaluation for all number of facilities as presented in Table 5.5 where problem configurations are grouped by the number of facilities.

[1] Percentage cost savings recorded in this work translates into millions in cost savings over t_{max}

Table 5.5: Wilcoxon comparison of dynamic vs static approach grouped by the number of facilities of the DC-LA problem

m	Dynamic $R+$	Static $R-$	$p-value$	Statistical difference?
10	**4609**	542	5.70E-12	**Yes**
20	**17126**	8299	6.35E-06	**Yes**
50	**51722**	13258	2.18E-22	**Yes**
100	**50414**	14566	1.18E-19	**Yes**

Even though the dynamic evaluation achieves statistical difference for all facility sizes, we observe in Figure 5.2 that the higher amount of costs savings of about 6% [2] is achieved for a more significant number of facilities, i.e. 100 facilities; making the dynamic evaluation more suitable of the two evaluation functions for problem configurations with a large facility size.

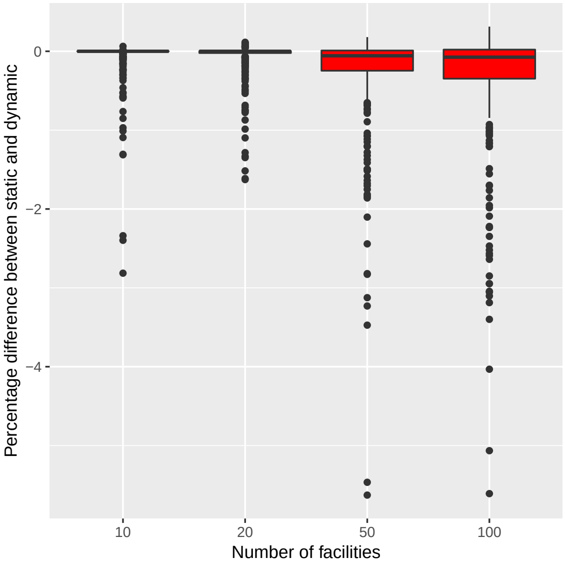


Figure 5.2: Percentage difference between the dynamic and static evaluation grouped by number of facilities

Although the dynamic evaluation is the favourable choice for locating facilities, especially when the problem has a larger number of facilities, the size of the facilities add to the computational costs of the dynamic evaluation function. The larger the facility size, the more costs have to be computed in the evaluation process.

[2] Percentage cost savings recorded in this work translates into millions in cost savings over t_{max}

5.3.4 Number of Customers (n)

Finally, the number of customers also appears to influence the choice of the evaluation function. From Figure 5.3, we see that more cost savings are achieved for a smaller number of customers than a larger number of customers.

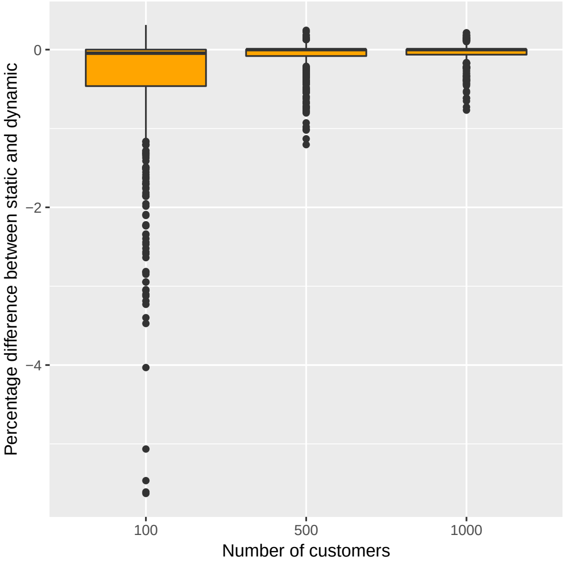


Figure 5.3: Percentage difference between the dynamic and static evaluation grouped by number of customers

The effect of the number of customers can be explained by the fact that the movement of a smaller number of customers creates more varying costs than a larger number of customers as seen in Figure 5.4. Figure 5.4 shows the earth mover's distance described in Section 5.2 calculated across 20 simulations on the ranking of facilities for 100, 500 and 1000 customers on 50 facilities when customers are assumed to make the most movements over a defined horizon.

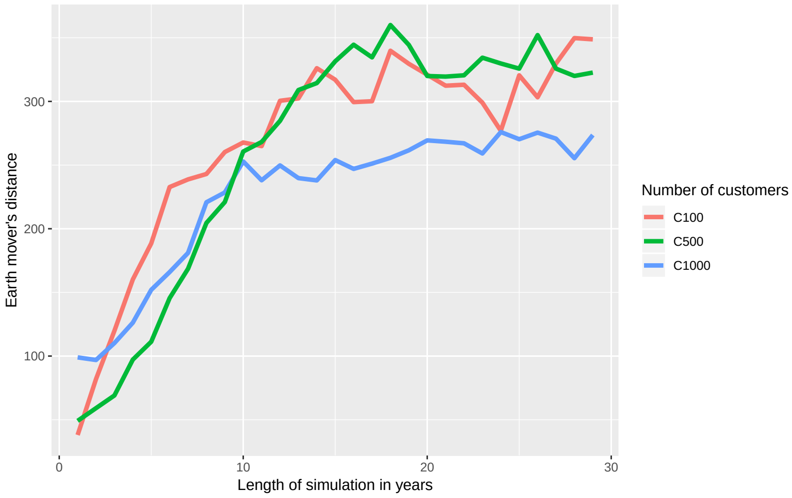


Figure 5.4: Earth mover's distance between the distributions of customers over the length of simulation for 100, 500 and 1000 customers

Here we see that as the years progresses the difference in the distribution of customers is greater for 100 customers than 500 and 1000 customers showing that a bigger change in customer distributions over the simulation. The big changes tend to affect the number of resources and where the resource needs to be allocated. The distances between the distributions for 1000 customers are seen to average out after a number of years with slight changes, showing that the number of customers per a city does not change much as the gains of customers who move out are about the same number of customers who move in causing the distribution of customers to even out over the simulation. It should, however, be noted that the level of variations in customer distribution is dependent on how much customers move over time hence from Table 5.2, even for larger number of customers a statistical significance is recorded in results between the two evaluation functions when customers are assumed to make frequent movement over the defined horizon.

So far, we aimed to ascertain if there was value in using a stochastic simulation model to simulate the movement of customers over a defined period. If so, when do we use a more expensive evaluation function to decide to locate facilities because it will have a cost impact? From the discussion of results, we see that when customers make frequent movements over a defined period, there is value in simulating customer movements to help make a decision. In such a scenario, we employ a more expensive dynamic evaluation function to help decide the location of facilities. To understand the computational effort exerted by the evaluation functions, we discuss the computational time complexities of the two evaluation functions in Section 5.3.5.

5.3.5 Computational time complexity

Table 5.6 shows the average computational time taken by the static and dynamic evaluations for each of the 48 problem configurations. *Dynamic/Static* shows the ratio of computational time in seconds between dynamic and static. Problem configurations are defined by the movement rates mr, the number of facilities m and the number of customers n.

Table 5.6: Computational times in seconds of static and dynamic for each problem configurations

			Time (seconds)		
mr	m	n	Static	Dynamic	Dynamic/Static
0.25	10	100	0.65	100.99	155.37
0.5	10	100	0.46	56.88	123.65
0.75	10	100	0.38	45.69	120.24
1	10	100	0.26	33.37	128.35
0.25	20	100	0.86	121.45	141.22
0.5	20	100	0.47	65.48	139.32
0.75	20	100	0.35	51.22	146.34
1	20	100	0.26	35.77	137.58
0.25	50	100	1.19	172.44	144.91
0.5	50	100	0.67	85.36	127.40
0.75	50	100	0.58	64.49	111.19
1	50	100	0.44	41.44	94.18
0.25	100	100	1.64	241.85	147.47
0.5	100	100	0.96	112.95	117.66
0.75	100	100	0.79	82.89	104.92
1	100	100	0.59	49.69	84.22
0.25	10	500	3.12	493.43	158.15
0.5	10	500	1.88	275.83	146.72
0.75	10	500	1.6	220.5	137.81
1	10	500	1.25	159.39	127.51
0.25	20	500	4.19	595.03	142.01
0.5	20	500	2.4	319.39	133.08
0.75	20	500	1.95	249.18	127.78
1	20	500	1.48	171.52	115.89
0.25	50	500	5.84	865.87	148.27
0.5	50	500	3.3	427.85	129.65
0.75	50	500	2.62	322.91	123.25
1	50	500	1.93	203.44	105.41
0.25	100	500	8.22	1225.31	149.06
0.5	100	500	4.52	576.5	127.54
0.75	100	500	3.63	423.63	116.70
1	100	500	2.64	248.26	94.04
0.25	10	1000	6.84	999.69	146.15
0.5	10	1000	4.03	547.91	135.96
0.75	10	1000	3.33	441.84	132.68
1	10	1000	2.58	318.82	123.57
0.25	20	1000	8.23	1191.34	144.76
0.5	20	1000	4.8	628.33	130.90
0.75	20	1000	4.03	498.67	123.74
1	20	1000	2.91	342.57	117.72
0.25	50	1000	12	1723	143.58
0.5	50	1000	6.6	845.15	128.05
0.75	50	1000	5.34	645.87	120.95
1	50	1000	3.96	408.19	103.08
0.25	100	1000	16.71	2467.96	147.69
0.5	100	1000	9.07	1159.61	127.85
0.75	100	1000	7.27	857.84	118.00
1	100	1000	5.06	501.67	99.14

From Table 5.6, it is seen that for all problem configurations, the dynamic evaluation is computationally about 84 to 158 times higher than static. To understand why this is so, we need to analyse the impact each problem parameter has on the computational time of the dynamic evaluation.

Firstly, we observe that the lower the mr, the more expensive the time recorded. A low mr means that customers make more movements over time. Whenever a customer makes a move, new coordinates are computed for the customer and the distances calculated between the new location of the customer and the cities to obtain the least cost. Hence, the more movement a customer makes the more time it takes to generate new locations and compute the costs. On the other hand, a higher mr means that a customer will make little or no movement over t_{max}. Which means the simulation models generate fewer movement times and hence fewer new customer locations. From the Table, the average computational time taken by dynamic when customers are assumed to make frequent movements over the defined period is 3 to 5 times higher than when customers are assumed to make little or no movement over the defined time.

Secondly, the number of facilities also contribute to the computational time. To obtain the least cost of a new customer location, we calculate the cost of service between the customer location and the facilities. Hence, the more facility locations there is, the longer the time taken to compute the distance between the facility locations and a customer location. The computational cost of the dynamic increases by a factor of 2.5 to 3 between problems with 10 and 100 facilities.

Thirdly, the number of customers adds to the computational effort. Since costs are computed for customers each time they move, the more customers there are, the more the time taken to compute distances between new customer locations and facilities locations. The computational cost of the dynamic evaluation function increases by a factor of 9 to 10 between problem configurations with 100 and 1000 customers.

Due to the stochastic nature of PBIL, we run the algorithm on each problem instance 20 times for both static and dynamic evaluations. We do this so we can better access the performance of the algorithm based on the distribution of results. Running the algorithms many times comes with high computational overhead. For example, if it takes 1.5 hours to make a single run of PBIL using dynamic evaluation on a small problem instance with 10 facilities by 100 customers with mr=0.25, then it will take us about 30 hours to complete all 20 runs of the algorithm. Not considering the time it will take on a larger problem of 100 facilities by 1000 customers with mr=0.25. We, therefore, explore the concept of the Maximum likelihood solution (MLS) [116] which is a property of PBIL in Section 5.3.6 to see if we can avoid making many runs of PBIL on a problem instance thereby saving us much experimental time.

5.3.6 The Maximum Likelihood Solution

In PBIL, the entire genetic population is generated from the probability vector (PV). Throughout a run, PBIL will generate whole populations of solutions, and the best solutions in the population are used to evolve the PV from which new solutions are generated as described in Section 2.5.4. At the end of a run, the solution that is generated from the evolved PV is the solution of interest to us. We call this solution the *Maximum likelihood Solution (MLS)* [116]. The solution is called the maximum likelihood solution because if we run PBIL many times the value of MLS is most likely to be the average solution fitness we will observe as it will come up a bit more often than other solutions. Although we might not know the exact distributions of the values or fitness of solutions that will be generated by PBIL. However, if PBIL is run many times, we will expect MLS to show up more often than all other solutions. In essence, we can refer to MLS as the mode of the distribution of solutions that will be generated by PBIL over the many runs. MLS, therefore, is the maximum likelihood estimate of what solution we will generate and hence, its fitness is a maximum likelihood value we will get. The probability of obtaining MLS is defined as:

$$P(x) = \prod_{i=1}^{m} p_i(x_i) \tag{5.3}$$

Because the PV in PBIL is a uni-variate model, i.e. it assumes full independence of problem variables the probability of MLS: *P(MLS)* is the product of the probabilities of the individual properties. Although the probability of obtaining MLS is minimal, MLS has a whole region around it that is quite similar to it. Taking the region into consideration gives us a large concentration of probabilities. It should be noted that the *MLS* for an evaluation function is only obtained from the evolved probability vector at the end of a run after solutions have been evaluated using an evaluation function In Figure 5.5, we show how the MLS evolves throughout a run. We plot the MLS over the number of generations against the best solution found in the population for each generation. The red line indicates the best solution whiles the blue line shows the evolution of MLS for a run.

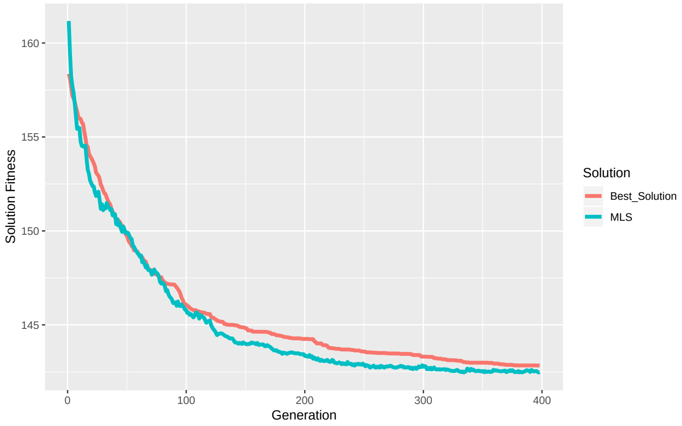


Figure 5.5: Evolution of MLS plotted against the best solution in the population

We employ MLS for comparison to see if we can get an estimate consistent with results obtained by an evaluation function when we run the experiments many times. To allow for comparison of results, we evaluate MLS with the same 5000 scenarios used to evaluate the best solutions found for the static and dynamic evaluation functions.

In Table 5.7, we show the number of wins and ties recorded between each of the evaluation functions and their respective MLS for all 30 problems of each problem configuration. The highest wins or ties are highlighted in bold for each problem configuration. The *Maximum-Likelihood Solution* is represented as $Static_{MLS}$ and $Dynamic_{MLS}$ for static and dynamic evaluations, respectively. Each problem configuration is defined by movement rate mr, the number of facilities m and the number of customer n.

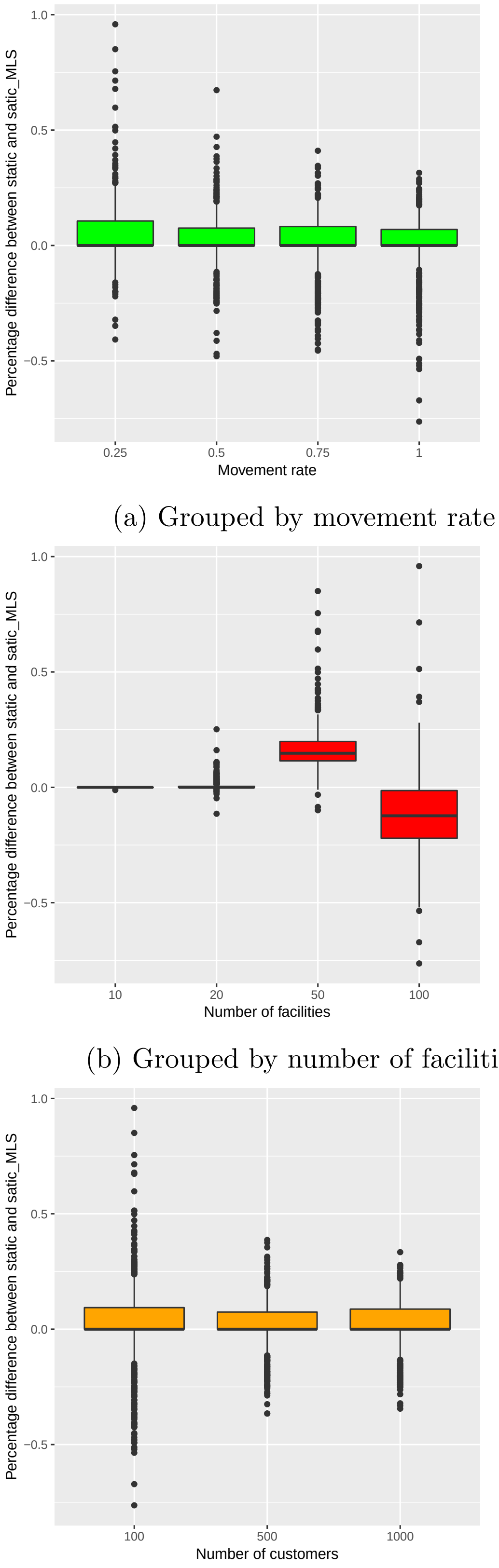
Table 5.7: Wins recorded between evaluations and their respective MLS

mr	m	n	Static	Static$_{MLS}$	Ties		Dynamic	Dynamic$_{MLS}$	Ties
0.25	10	100	0	0	**30**		0	5	**25**
0.5	10	100	0	1	**29**		1	1	**28**
0.75	10	100	0	0	**30**		0	2	**28**
1	10	100	0	0	**30**		0	0	**30**
0.25	10	500	0	0	**30**		1	0	**29**
0.5	10	500	0	0	**30**		2	1	**27**
0.75	10	500	0	0	**30**		1	0	**29**
1	10	500	0	0	**30**		1	1	**28**
0.25	10	1000	0	0	**30**		0	2	**28**
0.5	10	1000	0	0	**30**		1	2	**27**
0.75	10	1000	0	0	**30**		0	0	**30**
1	10	1000	0	0	**30**		0	0	**30**
0.25	20	100	7	1	**22**		**16**	4	10
0.5	20	100	4	1	**25**		8	4	**18**
0.75	20	100	7	0	**23**		12	0	**18**
1	20	100	8	0	**22**		10	0	**20**
0.25	20	500	8	2	**20**		**14**	2	**14**
0.5	20	500	7	3	**20**		10	3	**17**
0.75	20	500	10	0	**20**		12	1	**17**
1	20	500	10	0	**20**		11	0	**19**
0.25	20	1000	7	2	**21**		**13**	4	**13**
0.5	20	1000	11	2	**17**		**14**	3	13
0.75	20	1000	11	1	**18**		11	2	**17**
1	20	1000	11	0	**19**		**15**	0	**15**
0.25	50	100	**27**	3	0		**28**	2	0
0.5	50	100	**29**	1	0		**30**	0	0
0.75	50	100	**30**	0	0		**30**	0	0
1	50	100	**30**	0	0		**30**	0	0
0.25	50	500	**30**	0	0		**30**	0	0
0.5	50	500	**30**	0	0		**30**	0	0
0.75	50	500	**30**	0	0		**30**	0	0
1	50	500	**30**	0	0		**30**	0	0
0.25	50	1000	**30**	0	0		**30**	0	0
0.5	50	1000	**30**	0	0		**30**	0	0
0.75	50	1000	**30**	0	0		**30**	0	0
1	50	1000	**30**	0	0		**30**	0	0
0.25	100	100	**20**	10	0		2	**27**	1
0.5	100	100	8	**22**	0		1	**29**	0
0.75	100	100	2	**28**	0		0	**30**	0
1	100	100	0	**29**	1		0	**30**	0
0.25	100	500	14	**15**	1		2	**28**	0
0.5	100	500	8	**22**	0		3	**27**	0
0.75	100	500	4	**25**	1		2	**28**	0
1	100	500	0	**30**	0		1	**29**	0
0.25	100	1000	9	**21**	0		5	**25**	0
0.5	100	1000	10	**20**	0		5	**25**	0
0.75	100	1000	5	**25**	0		5	**25**	0
1	100	1000	3	**27**	0		1	**29**	0

From Table 5.7, we observe that a higher number of ties is achieved for both static and static$_{MLS}$ and dynamic and dynamic$_{MLS}$ on problems with 10 and 20 facilities. We also observe that for problem configurations with 50 facilities, both static and dynamic evaluations achieve a higher number of wins than their respective MLS. However, on the more significant problems with 100 facilities, we observe that static$_{MLS}$ and dynamic$_{MLS}$ both achieve a higher number of wins than static and dynamic evaluations respectively. The good performance of MLS for larger number of facilities in both evaluation functions can be explained by the fact that when the search space is larger such as in the case of a larger number of facilities, MLS achieves almost consistent results over the 20 runs as it is the most likely solution to be obtained for each run of the algorithm. However, due to the stochastic nature of the algorithm, results obtained for static and dynamic may vary for each run of the algorithm especially for problems with a larger number of facilities which have a larger search space. This is because the search space is too large to be adequately explored by the algorithm. For smaller number of facilities which have a smaller search spaces, static and dynamic evaluations both obtain consistent results as the search space can be adequately explored by the algorithm to find the optimal or near-optimal solution on each run of the algorithm. MLS on the other hand will almost always return the mode solution of the distribution of solutions and this might not necessary be the optimal or near-optimal solution. Hence we observe that on smaller number of facilities, static and dynamic achieves better or similar results to MLS. However on problems with a larger number of facilities, MLS is observed to obtain the better results for both static and dynamic evaluations.

The results observed in Table 5.7 shows that when we employ PBIL to solve more significant problems having 100 facilities, using the MLS offers better results than the results obtained with static and dynamic evaluations. This means that for much larger problems, we can employ the MLS as a measure of making decisions to locate facilities without having to run many experiments.

To understand how the performance of static$_{MLS}$ and dynamic$_{MLS}$ translates into percentage cost savings we study Figure 5.6 and 5.7 grouped according to DC-LA problem parameters of mr, m and n.

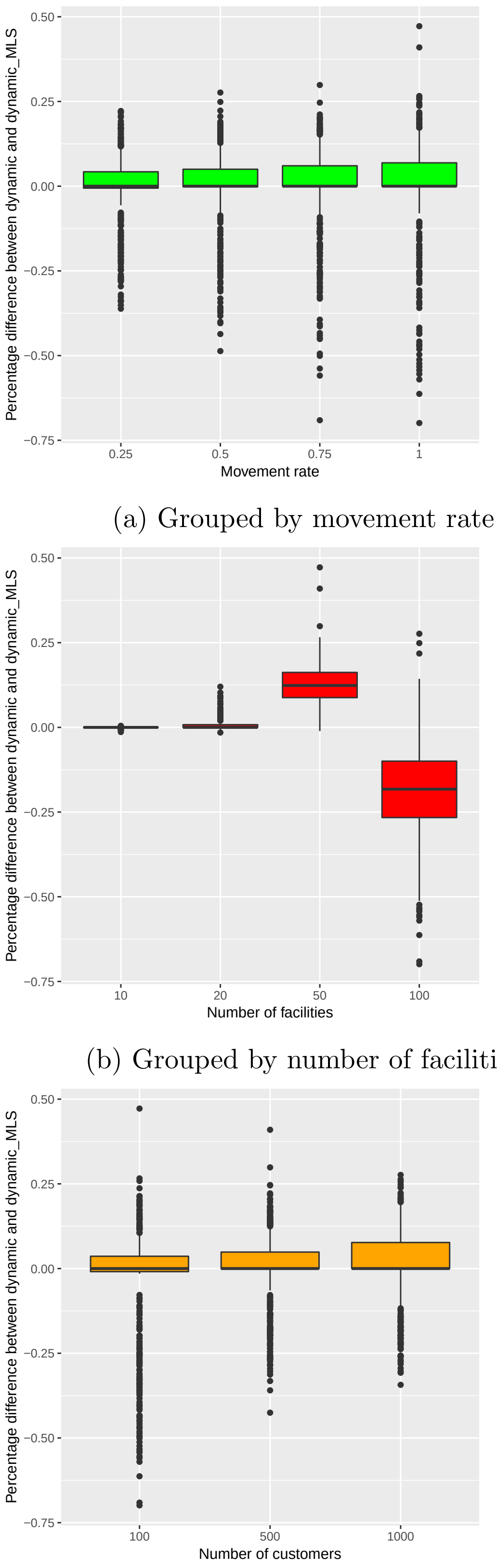


(a) Grouped by movement rate

(b) Grouped by number of facilities

(c) Grouped by number of customer

Figure 5.6: Percentage difference between the static and $static_{MLS}$



(a) Grouped by movement rate

(b) Grouped by number of facilities

(c) Grouped by number of customer

Figure 5.7: Percentage difference between the dynamic and $dynamic_{MLS}$

From both Figure 5.6 and 5.7 we observe an improve cost savings for a 100 number facilities when MLS is employed for both static and dynamic evaluations. We see that for static, MLS achieves an improved cost savings of about 0.77% [3] for a 100 facilities in Figure 5.6b whiles for dynamic, MLS is seen to achieve an improve cost savings of about 0.75% for a 100 facilities in Figure 5.7b. The results observed corresponds to the observation made in Table 5.7, which affirms the choice of MLS as a good measure for making decisions about locating facilities on problems with 100 facilities.

5.4 Chapter Summary

Our focus in this Chapter was to determine if there was value in simulating the movement of customers when deciding to locate facilities to service the changing distributions of customers. To do this, we run a set of experiments on the 1440 DC-LA problem instances generated in Section 4.3 by using PBIL 2.5.4 with the static and dynamic evaluation functions described in Section 4.2.2. The dynamic evaluation takes into consideration the customers movements between cities over the defined period. The static evaluation function assumes that customers do not make movements between cities over time and also forms the baseline for comparison with the dynamic evaluation method.

We observed that dynamic evaluation obtained globally better results than the static. However, we also noted that the performance of the dynamic evaluation function in-terms of wins and cost savings was highly dependent on the parameters of the problem. By analysing the problem parameters of movement rate mr, the number of facilities m and the number of customer n we observed that when customers make frequent movement over the defined horizon, the dynamic evaluation achieves higher cost savings. We also observe that the greater the number of facilities, the more cost savings are achieved by the dynamic evaluation. Finally, for the number of customers, we observe that the dynamic evaluation favoured a smaller number of customers.

It is also important to note that the computational time of the dynamic method can be extremely costly, especially in problems with a high frequency of customer movement, and a large number of facilities and customers. This computation overhead should be alleviated with the improvement in results obtained and should be taken into account when facing a given problem.

The performances of the dynamic and static evaluation functions to the generated benchmark raise the question of the computational effort one should dedicate to each solution evaluation in a stochastic environment concerning the global budget allocated to a search and the gain in performances. In the next Chapter, we will investigate a new technique to help find the right balance between the number of simulations required and computational time complexity of the dynamic evaluation method. We will then aim at relating this problem to a real-world application.

[3] Percentage cost savings recorded in this work translates into millions in cost savings over t_{max}

Chapter 6

Racing Strategy for the Dynamic-Customer Location-Allocation Problem

6.1 Introduction

Experimental results in Chapter 5 showed that the dynamic evaluation function offered better results when compared to the static evaluation function, especially when customers were assumed to make frequent movement over the defined horizon. However, the performance of the dynamic evaluation function comes with a high computational cost. The high computational cost is due to a large number of simulations required by the dynamic evaluation function to compare solutions in a population, and a large number of simulations often leads to a considerable computational effort been wasted at the early stages of the search on poor solutions. To help reduce the number of simulations required to compare solutions in a population, we explore the concept of iterated racing [115]. Racing uses a statistical test to compare solutions in the population after they have been evaluated against a number of simulations. Racing was first proposed in machine learning to deal with the problem of model selection [115]. Racing was then adapted by [17] for the configuration of an optimisation algorithm. In this Chapter, we seek to adapt the concept of racing to help reduce the number of simulations required to compare solutions in a population. The work presented in this Chapter has been accepted for publication in Proceedings of the 2020 Institute of Electrical and Electronics Engineers (IEEE) Congress on Evolutionary Computation (IEEE CEC 2020), 19-24th July 2020, Glasgow, UK.

This Chapter is organised as follows: In Section, 6.2 we introduce the concept of racing and how it has been used for parameter configurations. Then in Section 6.3, we discuss our adaptation of racing as a selection method for tackling DC-LAP. Finally Section 6.4 discusses experiments and results.

6.2 Racing

The process of model selection is to find a model that best describe an observed data. However, finding the model with the lowest generalisation error (a measure of how accurately a model can predict observation of data) can be a computationally expensive process, especially if the number of models is significant. Optimisation algorithms such as hill climber and genetic algorithms have been employed in model selection [17]; however, these algorithms sometimes end up with a model that is arbitrarily worse than the best. To tackle the high computational effort expended in finding the best model to describe a set of data, the concept of racing was developed in [115] by Maron et.al. Racing worked by testing the various models in parallel, one test point at a time. In this way, a running average could be maintained for each model's generalisation error. The average generalisation error is an estimate of the model's exact generalisation error had it been tested on all of the test points. By using a statistical bound, the closeness of the estimated generalisation error to the exact error could be determined. After a small number of test points, the best models, i.e. the models with the lowest generalisation error can be distinguished from the worst models (i.e. models with the highest generalisation error). The models that are significantly worse than the best ones are discarded from the race. A race here is a single iteration of the search process. The more test points that are observed, the tighter the estimated generalisation error is to the exact error. Hence many models can be differentiated from each other and discarded, thereby concentrating the computational effort on differentiating among the better model [115].

The concept of racing was later adapted in [17] as Iterated racing to automatically configure optimisation algorithms. The process of iterated racing primarily involves three steps: (1) sampling new configurations according to a particular distribution, (2) selecting the best configurations from the newly sampled ones utilising racing, and (3) updating the sampling distribution in order to bias the sampling towards the best configuration. These three actions are iterated until a termination condition is reached.

From a general perspective, an iterated racing approach is any process that iterates the generation of candidate configurations with some form of racing algorithm to select the best configurations. Hence a search process of an iterated racing approach could be, in principle, very different from the current use of finding the best candidate configurations and instead make use for example, of local searches, population-based algorithms or surrogate models. The essential element here is the appropriate combination of a search process with an evaluation that takes the underlying stochasticity of the evaluation into account [17]. Based on this reasoning, we are motivated to adopt the concept of iterated racing to the problem of Dynamic-Customer Location-Allocation (DC-LA) problem to help reduce the total number of evaluations which will help to reduce the considerable computational effort expended by the dynamic evaluation function. In Section 6.3, we describe our adaptation of racing to DC-LAP.

6.3 Adaptation of Racing to Dynamic Customer Location-Allocation problem

We employ the concept of racing in DC-LAP as a selection process to quickly discard the statistically worse solutions from the best solutions at the early stages of the search process thereby concentrating the computational effort on differentiating among the better solutions. An essential aspect of our adaptation of racing is in the truncation mechanism which strives to use the least number of simulations to compare solutions in the population.

In describing our adaptation of racing to DC-LAP, we first define the input parameters:

- A population size $\mathcal{P}$ of k solutions
- S_{max}: defines the maximum number of customer movement simulations per race. In the situation where solutions become mutually statistically indistinguishable, the race will continue to evaluate solutions against new simulations until the maximum number per race S_{max} is exhausted.
- S_{min}: minimum number of simulations before running a statistical test.
- A truncation rate μ: Based on the size k of the initial population, the race terminates when the size of the population $\mathcal{P}$ is decreased to μk. $\mu \in \{0, 1\}$

Once the initial population $\mathcal{P}$ of k solutions are generated by PBIL, for each race i every solution x in the population is evaluated on a customer movement scenario S_i by $f(\mathcal{P}_x, S_i)$. Before a statistical test is performed, each solution has to have performed S_{min} simulations. We employ the Friedman test described in Section 3.4.1 as the statistical test for determining statistical differences between the solutions. Once a statistical difference has been recorded, the solution(s) considered to be statistically worst when compared to the best solution in the population are removed from the population. If neither of the terminating criteria for the race has been satisfied, i.e. S_{max} or μk, the race continues by generating a new scenario and evaluating the remaining solutions against the new scenario. After the first statistical test has been performed using S_{min} simulations, subsequent statistical tests are performed more frequently after the single evaluation of all remaining solutions on every new scenario. After every test, statistically worse solutions from the best solutions are discarded from the population. Race continues until the size of $\mathcal{P}$ is decreased to μk or S_{max} is reached.

An example of a race is shown in Figure 6.1. In Figure 6.1 there exist 10 solutions. At every step of the race, the solutions are evaluated on a single scenario A' based on a new attraction rate. After several steps, those solutions that are deemed to be statistically worse than the best solution in the population are discarded from the population, and the race proceeds with the surviving solutions. Because the initial elimination test is essential in performing the statistical test, typically a higher number of simulations (A'_{first}) are observed before making the initial statistical test. Succeeding statistical tests are performed for each (A'_{each}) scenario. The process proceeds until a termination criterion is reached, i.e. a set minimum number of solutions in the population is reached, or the set maximum number of scenarios is exhausted, or a set number of solutions have been evaluated. Each node is the evaluation of a solution on a scenario. 'v' indicates that no statistical test is performed, '-' indicates that the test removed at least one solution from the population, '=' indicates that the test did not remove any solutions from the population. In the example below, $A'_{first} = 5$ and $A'_{each} = 1$.

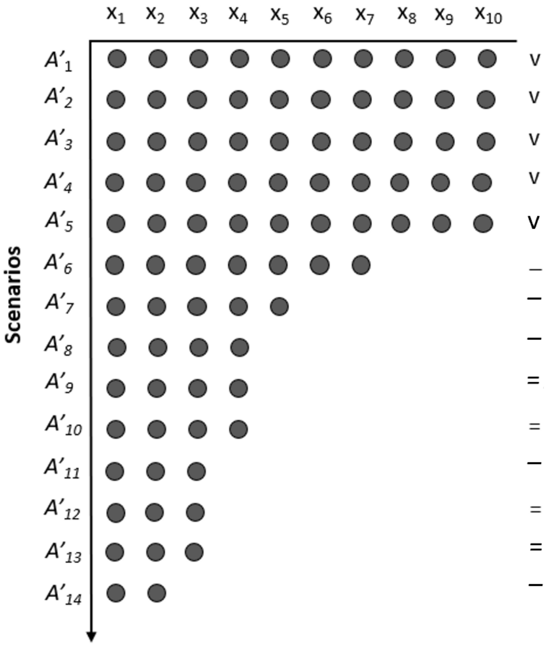


Figure 6.1: Racing for solution selection.

After the race, PBIL updates the probability vector PV with the surviving solutions. PBIL then generates new solutions to reset the population to its initial value of k. Solutions surviving from the previous generation are carried on to the next generation. Because the surviving solutions are not evaluated again on the same customer movement scenarios, it allows for the algorithm to save simulations in further generations. Algorithm 11 shows the pseudo-code of racing.

Algorithm 11 Racing as a selection method

```
    $k$ : population size
    $\mu$ : truncation rate
    $S_{min}$ : minimum number of simulations before running statistical test
    $S_{max}$ : maximum number of simulation per race
Generate initial population at random of $k$ solutions $\mathcal{P} = \{x_1, x_2, ..., x_k\}$
Generate set of customer movement scenarios $\mathcal{S} = \{S_1, ..., S_{max}\}$
while termination criterion not reached do
    i=0;
    while $|\mathcal{P}| > \mu k$ AND $i < S_{max}$ do
        $i = i + 1$
        for j= 1 to k do
            Evaluate $F_{ij} = f(x_j, S_i)$
        end for
        if $i \geq S_{min}$ then
            Perform statistical test on $F_{ij}$
            Remove from $\mathcal{P}$ all $x_j$ that are significantly worse than the best individual
            in the population.
        end if
    end while
    Update probability vector of PBIL with remaining solutions.
    Generate new solution from PBIL probability vector.
    Add new solutions to $\mathcal{P}$ until the size of $|\mathcal{P}| = k$
end while
```

6.4 Experimental Setup

For our experiment, we use the generated 1440 DC-LA problem instances used in Chapter 5. Parameter settings for PBIL are the same presented in Table 5.1 however we replace the truncation size of 0.5 with S_{min} or μk. The parameters for racing are the default values presented in the literature [104] and are presented in Table 7.1. Experiments were performed on an Intel(R) Xeon(R) E5620 @2.40GHz cluster.

Table 6.1: Racing parameters

Parameter	Description	Value
k	Population size	50
μ	Truncation rate	0.5
S_{min}	Minimum number of iterations per race	20
S_{max}	Maximum number of iterations per race	1000

6.5 Experimental Results and Analysis

To give better clarity to the performance of racing, we compare racing with the static and dynamic evaluation functions on the 1440 instances.

In Table 6.2, we show the average ranking of static, dynamic, and racing across all thirty instances for each problem configuration. The best average rank among the three evaluation functions on a problem configuration is highlighted in bold.

Table 6.2: Average ranking of static, dynamic and racing overall 48 problem configurations

mr	m	n	Static	Dynamic	Racing
0.25	10	100	2.17	**1.1**	1.27
0.5	10	100	1.57	**1.27**	1.43
0.75	10	100	1.37	1.23	**1.17**
1	10	100	**1**	**1**	1.07
0.25	10	500	1.77	1.33	**1.1**
0.5	10	500	1.4	1.4	**1.13**
0.75	10	500	1.2	1.1	**1.07**
1	10	500	**1.03**	1.07	**1.03**
0.25	10	1000	1.5	1.13	**1.03**
0.5	10	1000	1.27	1.27	**1**
0.75	10	1000	**1.1**	1.13	**1.1**
1	10	1000	**1**	1.1	1.13
0.25	20	100	2.6	1.6	**1.57**
0.5	20	100	1.93	1.87	**1.6**
0.75	20	100	1.47	1.8	**1.4**
1	20	100	1.17	1.57	**1**
0.25	20	500	2	1.67	**1.33**
0.5	20	500	1.57	1.9	**1.33**
0.75	20	500	1.27	1.87	**1.07**
1	20	500	1.3	1.57	**1**
0.25	20	1000	2.27	1.83	**1.17**
0.5	20	1000	1.97	1.87	**1.23**
0.75	20	1000	1.43	1.7	**1.07**
1	20	1000	1.2	1.53	**1**
0.25	50	100	3	**1.4**	1.57
0.5	50	100	2.9	**1.53**	**1.53**
0.75	50	100	2.6	2.1	**1.23**
1	50	100	2.3	2.33	**1.3**
0.25	50	500	2.93	1.77	**1.3**
0.5	50	500	2.73	2	**1.27**
0.75	50	500	2.23	2.27	**1.5**
1	50	500	2.3	2.47	**1.2**
0.25	50	1000	2.83	2	**1.17**
0.5	50	1000	2.63	2.07	**1.3**
0.75	50	1000	2.43	2.4	**1.17**
1	50	1000	2.43	2.4	**1.17**
0.25	100	100	3	1.73	**1.27**
0.5	100	100	2.93	2	**1.07**
0.75	100	100	2.57	2.27	**1.13**
1	100	100	2.47	2.33	**1.17**
0.25	100	500	2.97	1.97	**1.07**
0.5	100	500	2.7	2.2	**1.07**
0.75	100	500	2.37	2.53	**1.1**
1	100	500	2.53	2.4	**1.03**
0.25	100	1000	2.93	2.07	**1**
0.5	100	1000	2.57	2.33	**1.1**
0.75	100	1000	2.23	2.53	**1.23**
1	100	1000	2.5	2.37	**1.13**

From Table 6.2, it can be observed that racing achieves the best mean ranks for forty out of the forty-eight problem configurations. In Chapter 5, we established through experimentation that the dynamic evaluation offers better performance than the static approach. Therefore to have a better insight into the performance of racing when compared to the dynamic evaluation, we present a summary of wins and ties of racing against the dynamic evaluation in Table 6.3. In Table 6.3 each problem configuration has thirty problem instances. The wins obtained by racing and dynamic for each problem configuration is presented under racing and dynamic, respectively. Ties recorded between the two evaluation functions for each problem configurations are presented under *Ties*. For each problem configuration, the highest wins or ties are highlighted in bold.

Table 6.3: Wins, Losses and Ties of racing and dynamic evaluations grouped by the configuration of DC-LA problem

mr	m	n	Racing	Dynamic	Ties
0.25	10	100	3	8	**19**
0.5	10	100	3	9	**18**
0.75	10	100	5	4	**21**
1	10	100	0	1	**29**
0.25	10	500	7	2	**21**
0.5	10	500	6	2	**22**
0.75	10	500	2	1	**27**
1	10	500	1	1	**28**
0.25	10	1000	2	1	**27**
0.5	10	1000	5	0	**25**
0.75	10	1000	2	3	**25**
1	10	1000	1	2	**27**
0.25	20	100	**14**	12	4
0.5	20	100	**14**	10	6
0.75	20	100	**12**	6	**12**
1	20	100	9	0	**21**
0.25	20	500	**13**	7	10
0.5	20	500	**15**	6	9
0.75	20	500	13	1	**16**
1	20	500	9	0	**21**
0.25	20	1000	**16**	3	11
0.5	20	1000	**17**	4	9
0.75	20	1000	12	1	**17**
1	20	1000	9	0	**21**
0.25	50	100	12	**17**	1
0.5	50	100	**15**	14	1
0.75	50	100	**26**	3	1
1	50	100	**25**	5	0
0.25	50	500	**22**	8	0
0.5	50	500	**24**	6	0
0.75	50	500	**23**	7	0
1	50	500	**26**	4	0
0.25	50	1000	**26**	4	0
0.5	50	1000	**24**	6	0
0.75	50	1000	**28**	2	0
1	50	1000	**27**	3	0
0.25	100	100	**22**	8	0
0.5	100	100	**28**	2	0
0.75	100	100	**27**	2	1
1	100	100	**27**	3	0
0.25	100	500	**28**	2	0
0.5	100	500	**28**	2	0
0.75	100	500	**30**	0	0
1	100	500	**29**	0	1
0.25	100	1000	**30**	0	0
0.5	100	1000	**29**	1	0
0.75	100	1000	**29**	1	0
1	100	1000	**28**	2	0

From Table 6.3, we observe that the maximum number of wins is obtained by racing on problems with a more significant number of facilities. Problems with a larger number of facilities have a more extensive search space; this means that racing can effectively eliminate weak solutions from the population at the early stages of the search which helps to focus the search on good solutions to the problem. A look at the problems configurations with a smaller number of facilities shows more ties recorded between racing and dynamic evaluation. However, racing is seen to achieve more wins on average than the dynamic evaluation.

To know if there exists a statistical difference between the results obtained by static, dynamic and racing we perform a statistical test on the results in Section **??** using the Friedman test described in Section 3.4.1.

6.5.1 DC-LA problem parameters influence on results

Table 6.4 shows the average ranking of static, dynamic and racing evaluations overall 1440 problem instances. Here, the lower the average ranking, the better the overall results obtained by an evaluation function.

Table 6.4: Average Rankings of the algorithms

Algorithm	Ranking
Static	2.38
Dynamic	2.09
Racing	**1.52**

p-value computed by Friedman Test: 2.19E-10

In Table 6.4, racing is seen to have achieved the best average rank of 1.52, followed by the dynamic evaluation and then the static evaluation. To know if there is a significant difference between the results, we examine the p-value obtained by the Friedman test. Here the p-value of $2.19E - 10$ shows that there is a significant difference between the results obtained. We, therefore, apply the Holms procedure described in Section 3.4.1 as a post-hoc test to obtain p-values for each evaluation with $\alpha = 0.05$, which will help to determine where the significant differences exist. We present the results of Holm's test in Table 6.5. The Holm's adjusts the value of α, which is the significance level in a step-down manner. Holm's step-down procedure starts with the most significant p-value. If the p-value of an evaluation function is below $\alpha/(k-1)$, we conclude that there is a significant difference between the selected evaluation function and the control algorithm. In Table 6.5, the evaluation functions are presented under *algorithm* and their corresponding p-value is presented under p. Because racing achieved the least mean rank in Table 6.4, racing is used as the control algorithm on which to compare static and dynamic evaluation functions for statistical differences in results. If an evaluation function is deemed to be statistically different in results to racing, that evaluation function is highlighted in bold.

Table 6.5: Holm / Hochberg Table for $\alpha = 0.05$

i	algorithm	p	Holm
2	**Static**	**1.70E-118**	0.025
1	**Dynamic**	**2.49E-53**	0.05

From the Table, both static and dynamic evaluations are highlighted in bold signifying a significant difference between racing and static and racing and dynamic. It is, therefore, safe to conclude that racing obtains globally better results than static and dynamic evaluations.

To understand how the problem parameters influence the performance of racing or on what problem configurations racing performs better than static and dynamic, we analyse the results obtained according to the problem parameters of movement rate mr, the number of facilities m and number of customers n.

6.5.2 Movement Rate (mr)

Firstly, we examine the performance of racing according to the movement rate (mr) in Table 6.6. In Table 6.6 the different movement rates are presented under *mr*, the p-value obtained by the Friedman test for each movement rate is presented under *p-value*, α is set to 0.05 as done in Table 6.5. *Significance* shows whether or not a statistical difference is attained in results for a particular movement rate between racing, static and dynamic evaluations. On the movement rates where a statistical difference is detected, we perform the Holm's procedure to determine which of the evaluation functions is statistically different from the control method.

Table 6.6: Grouped by mr

mr	p-value	Alpha	Significance
0.25	**1.28E-10**	0.05	**Yes**
0.5	**1.05E-10**	0.05	**Yes**
0.75	**6.82E-11**	0.05	**Yes**
1	**7.71E-11**	0.05	**Yes**

From Table 6.6, it is seen that on all movement rate a statistical difference is detected, so we perform the Holm's procedure and presents results in Table 6.7

Table 6.7: Holm / Hochberg Table for $\alpha = 0.05$ on mr

i	algorithm	p	Holm
		mr0.25	
2	**Static**	**2.28E-59**	0.025
1	**Dynamic**	**7.89E-07**	0.05
		mr0.5	
2	**Static**	**4.61E-34**	0.025
1	**Dynamic**	**5.40E-12**	0.05
		mr0.75	
2	**Static**	**2.87E-20**	0.025
1	**Dynamic**	**6.20E-19**	0.05
		mr1.0	
2	**Static**	**3.34E-22**	0.025
1	**Dynamic**	**2.67E-19**	0.05

From 6.7, we observe that racing achieves statistically better results when compared to the static and dynamic evaluations on all movement scenarios of customers. In Section 5.3 where we compared the results of static and dynamic evaluations, we observed that when customers made little or no movement over the defined period, the dynamic evaluation obtained mutually indistinguishable results from the static evaluation showing that the static evaluation was as good an option as dynamic in such as scenario.

However, the performance of racing shows that even in scenarios where customers are assumed to make little or no movements over the defined period, racing appears to achieve significantly better results than the static and dynamic evaluations. The performance of racing terms from the ability of racing to discard weak solutions from the population at the early stages of the search thereby focusing the computational effort on the good solutions in the population to evolve better solutions. To see how the statistical difference in results achieved by racing translates into cost savings, we direct our attention to Figure 6.2 and 6.3. Figure 6.2 shows the percentage cost savings between racing and static evaluation whiles Figure 6.2 shows the percentage cost savings between racing and dynamic evaluation. Negative values mean cost savings.

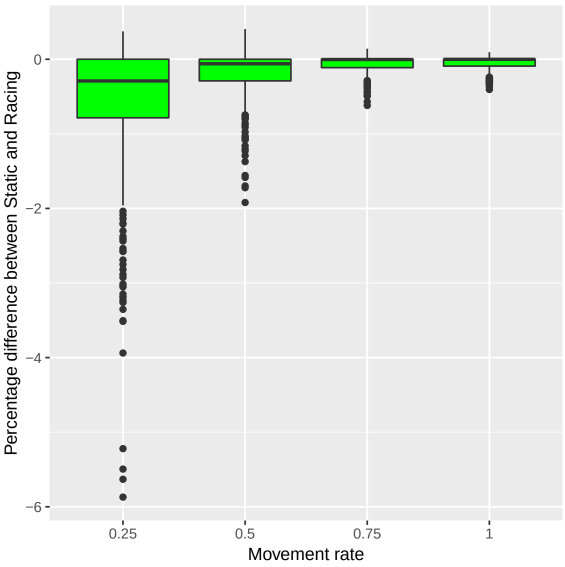


Figure 6.2: Percentage difference between the racing and static evaluation grouped by movement rate

From Figure 6.2 we observe that when customers make frequent movements over the defined period, racing achieves a costs savings of about 6% [1]. The cost savings reduces to about 0.5% when customers are assumed to make little or no movement over the defined period.

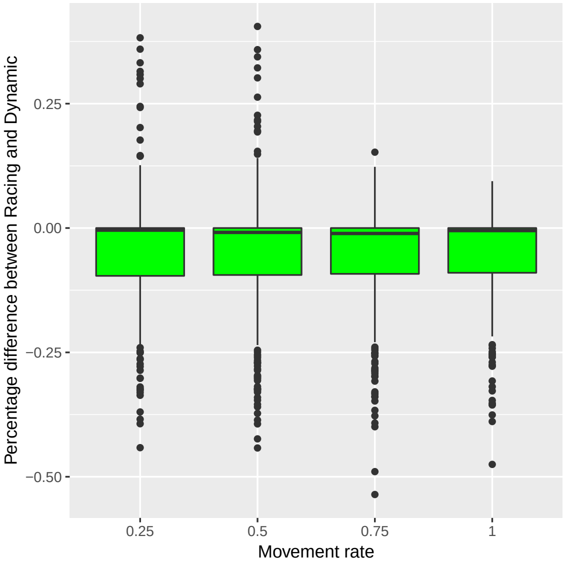


Figure 6.3: Percentage difference between the racing and dynamic evaluation grouped by movement rate

[1] Percentage cost savings recorded in this work translates into millions in cost savings over t_{max}

In Figure 6.3, racing is seen to improve on the cost savings achieved by the dynamic evaluation of up to about 0.5% [2] across all customer movement scenarios. The performance of racing on movement rates shows racing to be the best choice among the three evaluation functions when consideration is made to the movement of customers in deciding the locations of facilities.

6.5.3 Number of Facilities (m)

Secondly, we examine the influence of the number of facilities m on the performance of racing. In Table 6.8 the number of facilities are presented under m, the p-value obtained by the Friedman test for each facility size is presented under *p-value*, α is set to 0.05 as done in Table 6.6. *Significance* shows whether or not a statistical difference is attained in results for a particular number of facilities between racing, static and dynamic evaluations. On the number of facilities where a statistical difference is detected, we perform the Holm's procedure to determine which of the evaluation functions is statistically different from the control method.

Table 6.8: Grouped by m

m	p-value	Alpha	Significance
10	**1.21E-02**	0.05	**Yes**
20	**7.52E-10**	0.05	**Yes**
50	**1.45E-10**	0.05	**Yes**
100	**1.56E-10**	0.05	**Yes**

From Table 6.8, we observe that a statistical difference is obtained for all number of facilities. We, therefore, compute the Holm's procedure to find where the statistical difference exists for each number of facilities. The results of the Holm's procedure is presented in Table 6.9.

[2] Percentage cost savings recorded in this work translates into millions in cost savings over t_{max}

Table 6.9: Holm / Hochberg Table for $\alpha = 0.05$ on m

i	algorithm	p	Holm
		m10	
2	**Sim0**	**9.09E-03**	0.025
1	Sim100	9.41E-01	0.05
		m20	
2	**Sim0**	**1.64E-08**	0.025
1	**Sim100**	**2.27E-08**	0.05
		m50	
2	**Sim0**	**2.09E-68**	0.025
1	**Sim100**	**3.79E-24**	0.05
		mr100	
2	**Sim0**	**7.21E-94**	0.025
1	**Sim100**	**1.69E-50**	0.05

Results obtained in Table 6.9 shows that there exists a statistical difference in results between racing which is the control method and the two evaluation functions (static and dynamic) on 20, 50 and 100 facilities. However, for 10 facilities, racing is mutually statistically indistinguishable from dynamic evaluation, but racing is statistically different in results when compared to the static evaluation the results in Table 6.9 shows racing to be favourable to a more significant number of facilities. To understand how the performance of racing translates into cost savings concerning the number of facilities, we examine Figures 6.4 and 6.5. Figure 6.4 shows the percentage difference in cost savings between racing and the static evaluation whiles Figure 6.5 shows the percentage in cost savings achieved by the dynamic evaluation. Negative values mean cost savings.

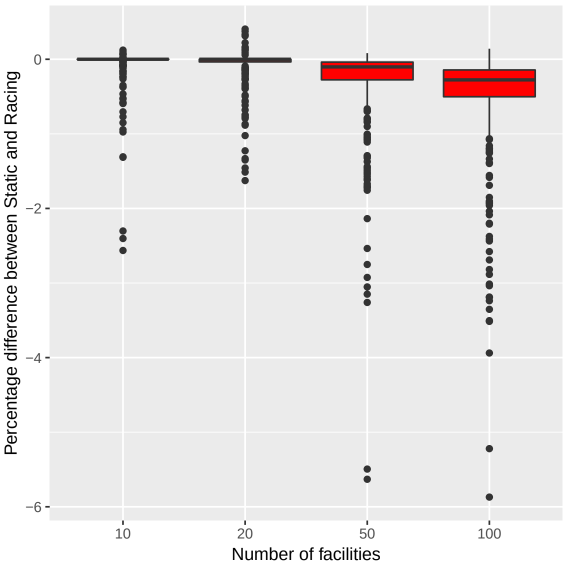


Figure 6.4: Percentage difference between the racing and static evaluation grouped by number of facilities

In Figure 6.4, we observe that for a smaller number of facilities, racing achieves a cost savings of about 2.6% and increases to about 6% for a more significant number of facilities when compared to the static evaluation.

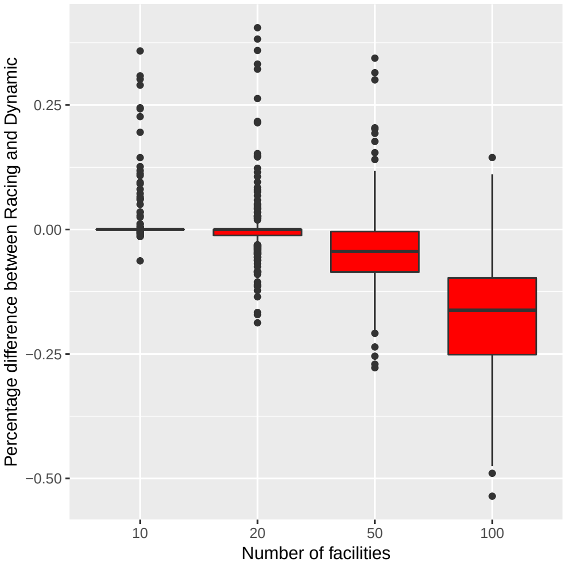


Figure 6.5: Percentage difference between the racing and dynamic evaluation grouped by number of facilities

In Figure 6.5, racing is observed to improve on the cost savings achieved by the dynamic for a smaller number of facilities of about 0.125% and this increases to about 0.55% for a more significant number of facilities.

6.5.4 Number of Customers(m)

Finally, we examine the influence the number of customers has on the performance of racing. In Table 6.10 the number of customers are presented under n, the p-value obtained by the Friedman test for each facility size is presented under *p-value*, α is set to 0.05 as done in Table 6.6. *Significance* shows whether or not a statistical difference is attained in results for a particular number of customers between racing, static and dynamic evaluations. On the number of customers where a statistical difference is detected, we perform the Holm's procedure to determine which of the evaluation functions is statistically different from the control method.

Table 6.10: Grouped by n

n	p-value	Alpha	Significance
100	**1.28E-10**	0.05	**Yes**
500	**9.13E-11**	0.05	**Yes**
1000	**1.09E-10**	0.05	**Yes**

From Table 6.10, we observe that a statistical difference is obtained for all number of customers. We, therefore, compute the Holm's procedure to find where the statistical difference exists for each number of customers. The results of the Holm's procedure is presented in Table 6.11.

Table 6.11: Holm / Hochberg Table for = 0.05 on n

i	algorithm	p	Holm
		n100	
2	**Static**	**1.77E-41**	0.025
1	**Dynamic**	**1.34E-08**	0.05
		n500	
2	**Static**	**3.88E-39**	0.025
1	**Dynamic**	**3.30E-24**	0.05
		n1000	
2	**Static**	**1.42E-41**	0.025
1	**Dynamic**	**3.60E-27**	0.05

Results obtained in Table 6.11 shows that there exists a statistical difference in results between racing which is the control method and the two evaluation functions (static and dynamic) on all number of customers. To understand how the performance of racing translates into cost savings concerning the number of customers, we examine Figures 6.6 and 6.7. Figure 6.6 shows the percentage difference in cost savings between racing and the static evaluation whiles Figure 6.7 shows the percentage in cost savings achieved by the dynamic evaluation. Negative values mean cost savings.

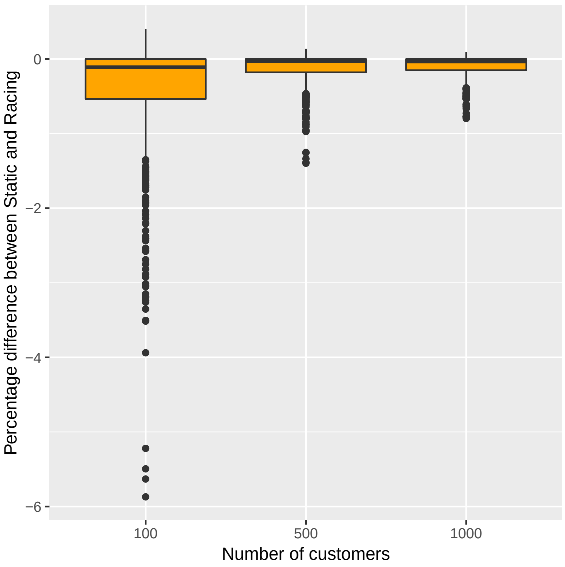


Figure 6.6: Percentage difference between the racing and static evaluation grouped by number of customers

In Figure 6.6, racing achieves a cost savings of about 6% for a smaller number of customers, and this decreases to about 0.8% for a more significant number of customers when compared to the static evaluation.

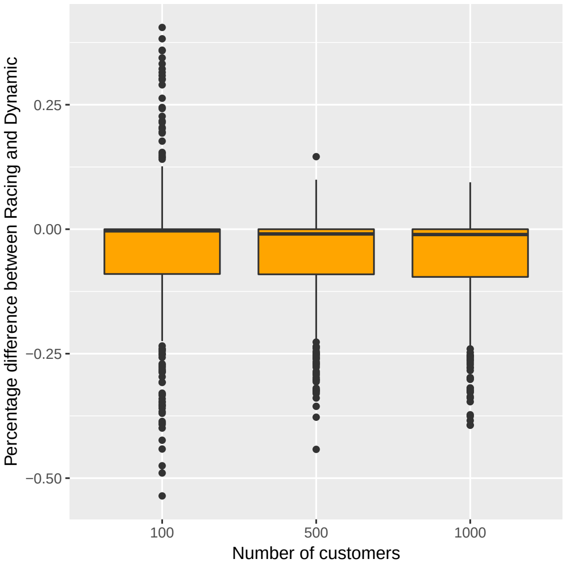


Figure 6.7: Percentage difference between the racing and dynamic evaluation grouped by number of customers

In Figure 6.7, racing is observed to have improved on the savings achieved by dynamic evaluation with an improved cost savings of about 0.55% for a smaller number of customers and decreases to about 0.35% for a more significant number of customers. The improved cost savings of racing makes it the best evaluation function among the three evaluation functions concerning the number of customers.

Results discussed in this Section shows that racing achieves improved cost savings when compared to the dynamic evaluation function concerning all problem parameters of movement rate mr, the number of facilities m and number of customers n. An important reason of adapting racing to our problem was to help reduce the number of simulations needed to efficiently compare solutions in a population during the search process thereby reducing the considerable computational effort that comes with evaluating solutions with many simulations as done in the dynamic evaluation. We, therefore, examine the computational effort expanded by racing in terms of time on the DC-LA problem instances in Section 6.5.5.

6.5.5 Computational Time Complexity

In this Section, we examine the computational time of racing on problem instances based on the problem parameters of DC-LAP. Table 6.12 shows the average computational time taken by racing and dynamic evaluation for each of the 48 DC-LAP problem configurations. The various movement rates are presented under *mr*, number of facilities under *m*, number of customers under *n*, computational time of dynamic is presented under *Dynamic* and computational time of racing under *Racing*. The ratio between time obtained by dynamic and racing is presented under *Dynamic/Racing*.

Table 6.12: Computational times in seconds of Racing and Dynamic for each problem configuration

			Time (s)		
mr	m	n	Dynamic	Racing	Dynamic/Racing
0.25	10	100	100.99	23.49	4.3
0.5	10	100	56.88	14.06	4.04
0.75	10	100	45.69	11.69	3.91
1	10	100	33.37	8.98	3.71
0.25	10	500	493.43	103.59	4.76
0.5	10	500	275.83	57.6	4.79
0.75	10	500	220.5	46.53	4.74
1	10	500	159.39	33.87	4.71
0.25	10	1000	999.69	205.38	4.87
0.5	10	1000	547.91	114.43	4.79
0.75	10	1000	441.84	91.78	4.81
1	10	1000	318.82	66.46	4.8
0.25	20	100	121.45	26.64	4.56
0.5	20	100	65.48	14.83	4.41
0.75	20	100	51.22	11.91	4.3
1	20	100	35.77	8.59	4.17
0.25	20	500	595.03	125.79	4.73
0.5	20	500	319.39	66.87	4.78
0.75	20	500	249.18	52.71	4.73
1	20	500	171.52	36.45	4.71
0.25	20	1000	1191.34	246.98	4.82
0.5	20	1000	628.33	131.78	4.77
0.75	20	1000	498.67	103.6	4.81
1	20	1000	342.57	71.43	4.8
0.25	50	100	172.44	37.06	4.65
0.5	50	100	85.36	19.01	4.49
0.75	50	100	64.49	14.76	4.37
1	50	100	41.44	9.83	4.22
0.25	50	500	865.87	179.79	4.82
0.5	50	500	427.85	89.14	4.8
0.75	50	500	322.91	67.83	4.76
1	50	500	203.44	42.84	4.75
0.25	50	1000	1723	354.54	4.86
0.5	50	1000	845.15	177.22	4.77
0.75	50	1000	645.87	134.62	4.8
1	50	1000	408.19	84.95	4.81
0.25	100	100	241.85	51.1	4.73
0.5	100	100	112.95	24.55	4.6
0.75	100	100	82.89	18.49	4.48
1	100	100	49.69	11.45	4.34
0.25	100	500	1225.31	253.42	4.84
0.5	100	500	576.5	119.88	4.81
0.75	100	500	423.63	88.94	4.76
1	100	500	248.26	52.15	4.76
0.25	100	1000	2467.96	506.33	4.87
0.5	100	1000	1159.61	241.54	4.8
0.75	100	1000	857.84	178.6	4.8
1	100	1000	501.62	103.87	4.83

From Table 6.12, we observe that as customers make frequent movement over time, the average computational time of racing is 4.8 times lower than the average time recorded by the dynamic evaluation. Also, as customers make little or no movement over time, the average time recorded by racing is 4.5 times lower than the average time recorded by the dynamic evaluation.

For the number of facilities, we observe that on the smallest number of facilities, the average time recorded by racing is 4.5 times lower than the average time recorded by the dynamic evaluation while the average time recorded by racing on the most significant number of facilities is 4.7 times lower than the average time recorded by dynamic evaluation.

For the number of customers, we observe that on the smallest number of customers, the average time recorded by racing is 4.3 times lower than the average time recorded by the dynamic evaluation while the average time recorded by racing on the most significant number of customers is 4.8 times lower than the average time recorded by the dynamic evaluation.

The improved computational time of racing can be attributed to the ability of racing to discard weak solutions at the beginning of the search process through the use of statistical tests to compare solutions in the evolutionary framework. This approach allows simulations to be performed iteratively until a statistical difference is reached, which ensures that the minimum number of simulations is performed to detect statistical difference to support solution selection. On the other hand, due to the larger number of simulations required by the dynamic approach, a considerable effort is often wasted in the early stages of the search process on weak solutions. The waste of effort and a large number of simulations all contribute to the expensive computational cost of the dynamic evaluation, which is on average 4.5 times higher than the average time recorded by racing.

For all problem parameters, the ratio between the variance in computational time recorded for racing is 24 times lower than the variance in computational time recorded by the dynamic. The variance shows that the computational time recorded for racing on problem configurations are relatively closer to the mean recorded time than for dynamic evaluation. On average racing employs about 21 simulations within each race of a run to quickly discard weak solutions from the population. The use of the minimum amount of simulation to discard weak solutions accounts for the low variance in the computational costs recorded for racing.

In this Section, we examined the computational time complexity of racing and compared it to that of the dynamic evaluation. Results showed that racing achieved improved costs savings over the dynamic evaluation. The improved costs savings by racing was achieved at an average of about 4.5 times less the computational time taken by the dynamic evaluation. The improvement of racing in both costs savings and computational time makes establishes racing as the best evaluation amongst the three evaluations for deciding the location of facilities concerning the movement of customers.

Even though we achieve an improvement in computational time with racing, we still run experiments many times in order to better assess the results obtained by racing. We, therefore, explore the concept of the Maximum likelihood solution (MLS) as we did in Section 5.3.6 for dynamic and static to see if we can avoid making many runs of racing on a problem instance to save time.

6.5.6 Maximum-Likelihood Solution (MLS)

In Table 6.13, we show the number of wins and ties recorded between racing and the MLS for all 30 problems of each problem configuration. The highest wins or ties are highlighted in bold for each problem configuration. The *Maximum-Likelihood Solution* is represented as *Racing_MLS*. Each problem configuration is defined by the movement rate of mr, the number of facilities m and the number of customer n.

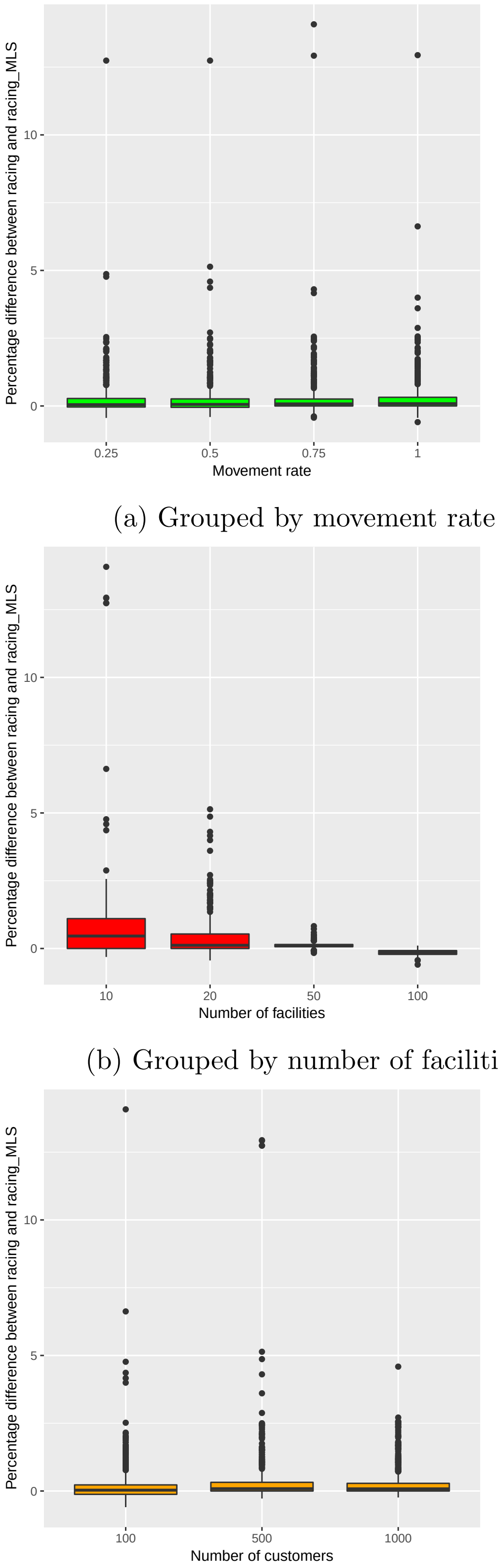
From Table 6.13, we observe that on problem configurations with 10, 20 and 50 facilities, racing achieves the greater number of wins. In Table 5.7 of Chapter 5, we observed that a greater number of ties was recorded for problem configurations with 10 and 20 facilities between static and dynamic and their respective MLS. The ability of racing to achieve more wins on the same number of facilities where static and dynamic recorded more ties when compared to their respective MLS affirms an improved performance of racing over static and dynamic evaluations.

Similar to observations made in Table 5.7, racing$_{MLS}$ achieves a greater number of wins over racing for problem configurations with 100 facilities. The good performance of MLS on the larger number of facilities can be explained by the fact that when the search space is larger such as in the case of a larger number of facilities, MLS achieves almost consistent results over the 20 runs as it is the most likely solution to be obtained for each run of the algorithm. However, due to the stochastic nature of the algorithm, results obtained for racing may vary for each run of the algorithm, especially for problems with a larger number of facilities which have a larger search space. This is because the search space is too large to be adequately explored by the algorithm. The consistent performance of MLS on the problem configuration with 100 facilities affirms the choice of using MLS as a measure for making decisions to locate facilities to service the changing demands of customers over time without having to run many experiments.

Table 6.13: Recorded wins and times between racing and racing$_{MLS}$ on problem configurations

mr	m	n	Racing	Racing$_{MLS}$	Ties
0.25	10	100	**19**	3	8
0.5	10	100	**17**	3	10
0.75	10	100	**19**	0	11
1	10	100	**23**	0	7
0.25	10	500	**24**	0	6
0.5	10	500	**21**	0	9
0.75	10	500	**24**	0	6
1	10	500	**27**	0	3
0.25	10	1000	**21**	0	9
0.5	10	1000	**24**	0	6
0.75	10	1000	**23**	1	6
1	10	1000	**25**	0	5
0.25	20	100	**21**	4	5
0.5	20	100	**18**	8	4
0.75	20	100	**21**	1	8
1	20	100	**21**	0	9
0.25	20	500	**21**	1	8
0.5	20	500	**22**	1	7
0.75	20	500	**21**	0	9
1	20	500	**24**	0	6
0.25	20	1000	**21**	2	7
0.5	20	1000	**23**	1	6
0.75	20	1000	**23**	1	6
1	20	1000	**22**	0	8
0.25	50	100	**18**	11	1
0.5	50	100	**18**	11	1
0.75	50	100	**30**	0	0
1	50	100	**30**	0	0
0.25	50	500	**29**	1	0
0.5	50	500	**30**	0	0
0.75	50	500	**30**	0	0
1	50	500	**30**	0	0
0.25	50	1000	**30**	0	0
0.5	50	1000	**30**	0	0
0.75	50	1000	**30**	0	0
1	50	1000	**30**	0	0
0.25	100	100	0	**30**	0
0.5	100	100	0	**30**	0
0.75	100	100	0	**30**	0
1	100	100	1	**29**	0
0.25	100	500	3	**27**	0
0.5	100	500	2	**28**	0
0.75	100	500	3	**27**	0
1	100	500	2	**28**	0
0.25	100	1000	3	**27**	0
0.5	100	1000	5	**25**	0
0.75	100	1000	2	**27**	1
1	100	1000	3	**27**	0

To observe how the performance of racing$_{MLS}$ translates to cost savings, we study Figure 6.8.



(a) Grouped by movement rate

(b) Grouped by number of facilities

(c) Grouped by number of customer

Figure 6.8: Percentage difference between the racing and $racing_{MLS}$

From Figure 6.8, we observe that racing$_{MLS}$ achieves an improved cost savings of about 0.63% on problem configurations with 100 facilities. The improved cost savings of racing$_{MLS}$ on the larger number of facilities (100) correlates to the performance of $dynamic_{MLS}$ and $static_{MLS}$ for the same number of facilities. The performance of racing$_{MLS}$ affirms the choice of MLS to help make decisions about locating facilities without having to run many experiments on problem instances with a significant number of facilities.

6.6 Chapter Summary

In Chapter 5, we explored the dynamic evaluation in the context of simulation-based optimisation to tackle the problem of DC-LA. Although the dynamic evaluation achieved better costs savings when compared to the static evaluation function, the dynamic evaluation came with a high computational cost due to a large number of simulations required in the evaluation process.

To help achieve a balance between a large number of simulation and the high computational cost, we adapted the concept of racing as a selection method to our problem. Our adaptation of racing uses the Friedman test to compare solutions in PBIL statistically. Racing allows simulations to be performed iteratively, ensuring that the minimum number of simulations is performed to detect a statistical difference.

We observed that racing obtained globally better results than the static and dynamic evaluation functions. In terms of cost savings, racing showed good performance by achieving improved cost savings over the dynamic evaluation.

We also observed that on average, the computational cost of racing was about 4.5 times lower than the computational cost recorded for the dynamic evaluation. The improved performance of racing over the dynamic approach and the improved computational time of racing makes it the best out of the evaluation functions to find robust solutions to DC-LA problem.

Experimentation with the maximum likelihood solution *MLS* showed that for problem configurations with a more significant number of facilities (100) we could employ the MLS to decide the locations of facilities without having to run many experiments; thereby saving much computational effort.

Having improved on costs savings and computational time with racing, we aim to apply the concept of racing to solve a real-world scenario of DC-LA problem in the next Chapter.

Chapter 7

Application of Racing to Real-World Dynamic-Customer Location-Allocation Problem

Experimental results in Chapter 6 showed the effectiveness of our adaption of racing to tackle the Dynamic-customer location-allocation (DC-LA) problem. The results showed that racing improved on the costs savings achieved by the dynamic evaluation in lesser time.

Owing to the successful performance of racing on DC-LA problem, in this Chapter, we seek to explore the effectiveness of racing on a real-world problem from the telecommunication industry using a service telecommunication company as a case study. The Chapter is structured as follows: In Section 7.1, we describe the real-world problem. In Section 7.2, we describe the process of generating an instance of a real-world problem. Experiments are conducted in Section 7.3 and results are discussed in Section 7.4. Finally, the Chapter summary is presented in Section 7.5.

7.1 Problem definition

This Section extends the case study in Section 1.1. The problem scenario is based on the service company's operations in the United States. In this problem, we seek to find optimal locations for establishing facilities out of 100 potential locations within the United States to service the changing locations of customers. The 100 potential locations correspond to the 100 most populous cities in the United States. This work aims to reduce the overall operational costs of establishing facilities to service the changing demand of customers. We assume that there are ten thousand customers spread across the United States. The distance in kilometres from a facility to a customer location by the unit cost of bandwidth per kilometre defines the major cost of a customer's connection to a facility. However, other factors add to the final operational costs at the end of the defined period. These factors are discussed below:

- Bandwidth: Each facility is assumed to have an initial core bandwidth of one gigabyte, which attracts a fixed cost. Because a facility is not restricted to the number of customers, it can service, the bandwidth need of a facility is thus determined by the total amount of bandwidth demands of customers serviced by the facility. i.e., if the bandwidth demands of customers currently being served by a facility are higher than one gigabyte, then an additional one gigabyte is added to the existing bandwidth. For every additional gigabyte of bandwidth added, there is an associated fixed cost. The core bandwidth for a facility is thus updated when required.

  At the start of the defined period, every customer is assigned a bandwidth size based on their demand. For all bandwidth sizes, there is an associated unit cost of bandwidth per kilometre. There is a 50% chance that a customer will change the bandwidth size when they make a move from one city to another. For every bandwidth size change a customer makes, the unit cost of the customer's bandwidth is updated with associated bandwidth cost.

- Net location: Net location refers to the customer site or location. When a customer location already has an established connection from a facility, the location is termed as an *on-net* location. If the customer location has no established connection from a facility, the location is termed as an *off-net* location. At the start of the defined period, each customer location is assumed to be either on-net or off-net. There exists a 50% chance that a customer may change locations from an on-net to an off-net location or vice versa when the customer moves between cities. In the situation that a customer relocates to an off-net location, the unit cost of bandwidth is increased by 50% in the first month to cover for new connection costs to the customer location. However, no extra cost is attracted if the new location of a customer is an on-net location.

- Buck-Up Connection: We assume that 20% of the total customer base have a backup connection to a new facility in addition to their primary connection to ensure resilience in connection. When a customer moves between cities, there is a 20% chance that a customer will request a backup connection which attracts a cost based on the requested bandwidth size.

7.2 Problem Instance Generation

The data for generating an instance of a problem is provided to us by a telecommunications service company whose operations are located in the united states. To generate, an instance of the problem, we randomly sample 10000 customer locations from a pool of 120000 customer locations from the United States based on the attractivity of the first one hundred most populous cities in the United States, i.e. the attraction rate of cities defines the distribution of customers among the cities at the start of the defined period. The number of facility locations and customers is defined together with industry experts from the service telecommunication company. By sampling customer locations from the data pool, we create four problem configurations based on the movement rates mr. For each problem configuration, we generate 30 problem instances. In all problem instances, the facilities locations remain the same; however, the customer locations vary for each problem instance as well as the initial bandwidth for each customer.

7.2.1 Simulation model

In simulating customer movements in the real world, we make some changes and additions to the generic simulation model presented in Section 4.2.1 for DC-LA problem formulation.

For each customer, we generate movement dates according to the movement rate mr. In the generic simulation model presented Section 4.2.1, when a customer had to move, the new location of the customer was randomly generated from a normal distribution with the centre of the new city as the mean of the distribution. In the real-world problem, the location of the customer is randomly sampled from a list of existing locations for that city provided by the telecommunication service company.

In the generic simulation model, the cost of servicing a customer was primarily determined by the Euclidean distance between a facility location and a customer location. In the real-world problem, the primary cost of servicing a customer is defined by the distance in kilometres from a facility location to the customer location by the unit cost of bandwidth a per a kilometre. However, other factors add up to the final cost of a customer. These factors are whether or not a customer moves to an on-net or off-net location; whether or not a customer decides to change their existing bandwidth size; and whether or not the customer decides to take up a back-up connection for resilience.

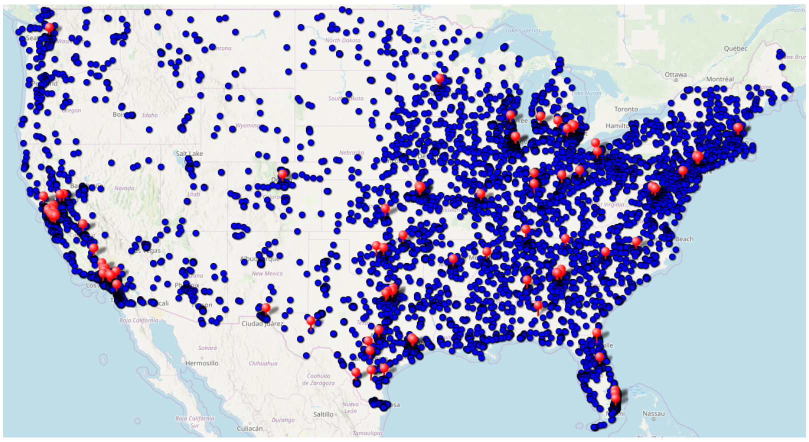


Figure 7.1: An example of facilities locations (red dots) and customers locations (blue dots) distribution in the United States

7.3 Experimental setup

All associated costs were defined based on industry standards with experts from the service telecommunications company.

- Facility equipment cost: 50000
- Facility Maintenance cost: 5000
- Initial bandwidth cost for facility: 3000
- Additional bandwidth cost for facility: 3000
- Bandwidth sizes in megabytes: 2, 4, 5, 8, 10, 20, 50, 100, 150, 250, 600, 1000
- Unit cost per kilometer for each bandwidth size respectively: 2.75, 3.12, 3.73, 4.05, 4.58, 6.14, 8.66, 9.66, 13.04, 14.03, 16.75, 19.12
- Off-Net cost: 50% increase of the unit of bandwidth cost for the first month

The costs presented are indicative of monthly costs. Like previous experiments we set the discount rate $r = 0.05$. The parameters for racing and *PBIL* are presented in Table 7.1. Experiments were performed on an Intel(R) Xeon(R) E5620 @2.40GHz cluster.

Table 7.1: Racing parameters

Parameter	Description	Value
k	Population size	50
μ	Truncation rate	0.5
S_{min}	Minimum number of iterations per race	20
S_{max}	Maximum number of iterations per race	1000

For each problem instance, we evaluate solutions using the static and racing evaluation functions. For this real-world problem, it will be infeasible to evaluate solutions using the dynamic evaluation function. This is evident from experiments conducted in Section 6.4 where we observed that the dynamic evaluation was on average 4.5 times greater than the time recorded by racing even on the larger problem with 100 facilities and 1000 customers. It will therefore be infeasible to apply the dynamic evaluation to a problem of 100 facilities by 10000 customers. Each run is allowed 10000 fitness evaluations. At the end of each run, the best solution is evaluated using the dynamic evaluation over 5000 simulations to allow for comparison. We run each evaluation function 20 times on all problem instances.

7.4 Results and Discussions

In Table 7.2, we present the average mean ranking achieved by racing and static overall 30 problem instances for each problem configuration. The smaller the mean value, the better the results achieved by the evaluation function.

Table 7.2: Average ranking of results overall problem instances

mr	m	n	racing	static
0.25	100	10000	**1**	2
0.5	100	10000	**1**	2
0.75	100	10000	**1**	2
1	100	10000	**1**	2

From Table 7.2, we observe that for the real world problem racing achieves the best mean rank on all problem configurations.

In Table 7.3, we show the wins of racing and static evaluation for all 30 instances for each problem configuration.

Table 7.3: Wins and losses of racing when compared to static evaluation grouped by the configuration of DC-LAP

mr	m	n	Racing	Static
0.25	100	10000	**30**	0
0.5	100	10000	**30**	0
0.75	100	10000	**30**	0
1	100	10000	**30**	0

Results in Table 7.3 shows that for all problem configurations, racing achieves the ultimate number of wins for each problem configurations. To understand if the wins achieved by racing translates into a statistical difference in results, we perform the Wilcoxon rank signed test described in Section 5.2.1 on the results obtained by the evaluation functions and presents the test results in Table 7.4.

In Table 7.4, the aggregate of ranks for racing on problem configuration is denoted as R^+. The aggregate of ranks for static evaluation on problem configurations is denoted as R^-. The aggregate of ranks for the problem configurations on which an evaluation function performed better than the other is highlighted in bold. Under the column labelled *significance*, a *Yes* indicate that there exists a significant difference in results between the evaluation functions on a problem configuration.

Table 7.4: Wilcoxon comparison of static and racing evaluations grouped by configurations of DC-LAP

mr	m	n	Racing R+	Static R-	p-value	significance?
0.25	100	10000	**465**	0	1.86E-09	**Yes**
0.5	100	10000	**465**	0	1.86E-09	**Yes**
0.75	100	10000	**465**	0	1.86E-09	**Yes**
1	100	10000	**465**	0	1.86E-09	**Yes**

Results from Table 7.4 shows that racing achieves significantly better results than static on all problem configurations irrespective of the number of movements customers make across the defined period. To study the cost-savings implication of results represented in Table 7.4, we refer to Figure 7.2, which shows the percentage difference in cost-savings between racing and static. Negative values mean cost savings.

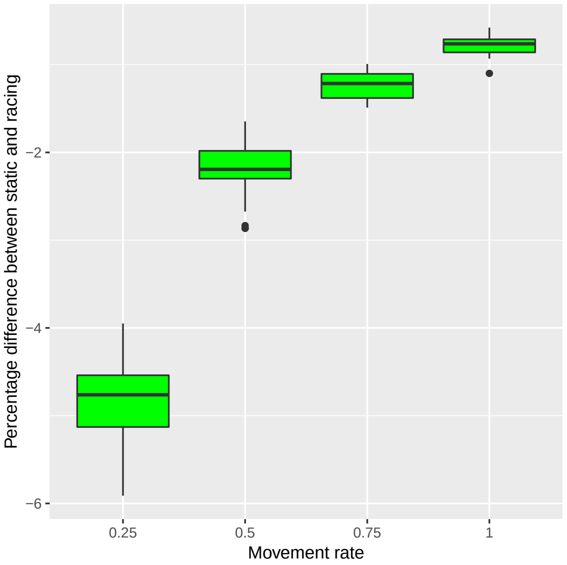


Figure 7.2: Percentage difference between the racing and static evaluation

From Figure 7.2 we observe that when customers make frequent movement over the defined period racing achieves cost-savings of about 5.9%[1], and this reduces to about 1.1% when customers are assumed to make little or no movement over the defined period.

The performance of racing makes it the best among the evaluation functions to tackle large real-world DC-LAP problems. The adaptation of racing as an evaluation function can also be extended to tackle other location problems that employ a stochastic evaluation.

We observed in Sections 5.3.6 and 6.5.6 that the maximum likelihood solution (MLS) obtained for each evaluation function showed better performances than their respective evaluation functions on problem configurations with 100 facilities. As a result of this performance, we study the MLS of static and dynamic evaluation functions on the real-world problem in Section 7.4.1.

[1] Percentage cost savings recorded in this work translates into millions in cost savings over t_{max}

7.4.1 Maximum Likelihood Solution (MLS)

In Table 7.5, we present the wins achieved by static, racing and their respective MLS over the 30 instances for each problem configuration. It should be noted that the *MLS* for an evaluation function is only obtained from the evolved probability vector at the end of a run after solutions have been evaluated using an evaluation function.The maximum number of wins on each problem configuration is highlighted in bold.

Table 7.5: Recorded wins for racing, static and their respective MLS on problem configurations

mr	m	n	Static	Static$_{MLS}$	Ties		Racing	Racing$_{MLS}$	Ties
0.25	100	10000	0	**30**	0		0	**30**	0
0.5	100	10000	0	**30**	0		0	**30**	0
0.75	100	10000	0	**30**	0		1	**29**	0
1	100	10000	0	**30**	0		2	**28**	0

From the results presented in Table 7.5, we observe that static$_{MLS}$ and racing$_{MLS}$ achieves the maximum number of wins on all problem configurations when compared to their respective evaluation functions.

To study how the wins of static$_{MLS}$ translate to cost-savings, we refer to Figure 7.3, which shows the percentage difference in cost-savings between static and static$_{MLS}$. Negative values mean cost savings.

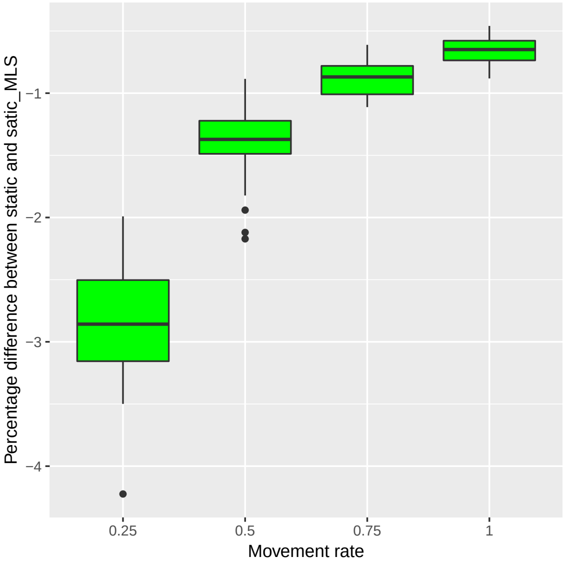


Figure 7.3: Percentage difference between the static evaluation and static$_{MLS}$

From Figure 7.3, we observe that in all scenarios of customer movement, an improved cost saving is achieved by static$_{MLS}$ over static evaluation function. When customers are assumed to move frequently over the defined period, we observed an improved percentage cost savings of about 3.5%[2], and when customers are assumed to make little or no movement over the defined period, we observe an improved cost savings of about 0.8%.

A look at figure 7.4 shows the percentage difference in cost savings between racing and racing$_{MLS}$. Negative values mean cost savings. Here also we observe improved cost savings by racing$_{MLS}$ over racing.

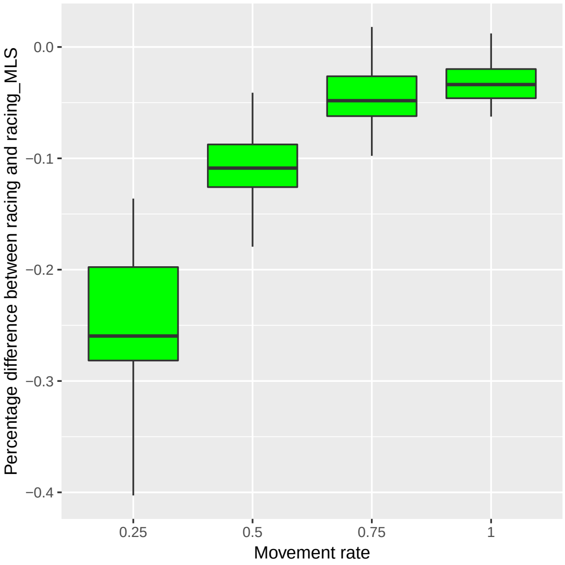


Figure 7.4: Percentage difference between racing and racing$_{MLS}$

From Figure 7.4 we observe that as customers make frequent movement over the defined period, racing$_{MLS}$ achieves an improved cost savings of about 0.4% over racing and as customers make less movement over the defined period we observe that racing$_{MLS}$ achieves an improved cost savings of about 0.06% over racing. The performance of racing$_{MLS}$ over racing on the larger number of facilities can be explained by the fact that when the search space is larger such as in the case of a 100 facilities, racing$_{MLS}$ achieves almost consistent results over the 20 runs as it is the most likely solution to be obtained for each run of the algorithm. However, due to the stochastic nature of the algorithm, results obtained for racing may vary for each run of the algorithm especially for problems with a larger number of facilities which have a larger search space. This is because the search space is too large to be adequately explored by the algorithm.

[2] Percentage cost savings recorded in this work translates into millions in cost savings over t_{max}

7.4.2 Computational time complexity

Table 7.6 shows the average computational time taken by static and racing. The time recorded for racing is presented under the column labelled *Racing* and the time recorded for static under the column labelled *Static*. The ratio between the time recorded for racing and static is recorded under racing/static.

Table 7.6: Computational times in seconds of racing and static for each problem configuration

			Time (s)		
mr	m	n	Racing	Static	racing/static
0.25	100	10000	1.23E+08	3.61E+04	3.41E+03
0.5	100	10000	5.17E+07	3.68E+04	1.40E+03
0.75	100	10000	3.50E+07	3.66E+04	9.57E+02
1	100	10000	1.58E+07	3.61E+04	4.37E+02

From Table 7.6, we observe that the computational time recorded by racing is 3413 times higher than the time taken by static evaluation when customers are assumed to make frequent movement over the defined period. The ratio in computational time between racing and static reduces to about 437 when customers are assumed to make little or no movement over the defined period. The computational time recorded by racing shows that using the dynamic evaluation function to evaluate solutions to this real-world problem would have been infeasible. Especially considering that for the extensive problem configuration of 100 facilities by 1000 customers of DC-LAP studied in Chapter 6, the time taken by the dynamic evaluation was on average about 4.8 times higher than racing when customers were assumed to make frequent movement over the defined period and 4.5 times higher when customers were assumed to make little or no movement over the defined period.

7.5 Chapter Summary

In this Chapter, we presented a real-world problem of DC-LAP using a service telecommunications company in the United States as a case study. To tackle the problem, we used our adaptation of racing introduced in Chapter 6 to find a robust solution to the problem and help reduce the computational cost incurred in evaluating solutions with a stochastic evaluation. To assess the performance of racing, we used static evaluation as a baseline for comparison. For each problem configuration, we sampled 30 problem instances from an existing set of 120000 customer locations. In all problem instances, facility locations were located in the centre of the first 100 most populous cities in the United States.

Results from the experiments showed racing to achieve a statistical difference in results on all problem configurations when compared to the static evaluation function.

We observed that racing achieved cost-savings of about 5.9% when customers were assumed to make frequent movement over the defined period and a cost-savings of about 1.1% when customers were assumed to make little or no movement over the defined period. The performance of racing makes it the best out of the evaluation methods introduced in this work for deciding the location of facilities in DC-LA problem.

To avoid running more experiments on a problem instance than necessary, we studied the maximum likelihood solution (MLS) obtained for static and racing. The results obtained by the MLS showed improved cost-savings over the savings achieved by racing and static evaluations overall problem configurations. We observed that on average racing$_{MLS}$ achieved an improved costs savings over racing of about 0.4% when customers were assumed to make frequent movements over the defined period and about 0.07% when customers were assumed to make little or no movement over the planning period.

The performance of racing$_{MLS}$, showed that we could confidently employ the MLS obtained from racing to make a good and robust decision of locating facilities to service the changing distribution of customers over a defined period.

Chapter 8

Conclusion and Further Work

The purpose of this thesis was to propose and investigate variations of LA problem formulations in the context of Robust optimisation over time (ROOT) [175]. In this Chapter, we present a summary of our contributions and a review of the extent to which we met our research objectives.

8.1 Contribution Summary

To aid in summarising the contribution of work in this thesis, we first present a summary of the objectives below. For a detailed description of the objectives, please refer to Section 1.2.2.

(O1) Develop new formulations that capture the real-world complexities of the telecommunications industry as an LA problem.

(O2) Propose a new problem instance to study the new LA problem formulation.

(O3) Investigate optimisation algorithms suitable for solving the new LA problem formulations.

(O4) Develop a stochastic simulation model to simulate the changes in customer demand over time.

(O5) Investigate a way to help reduce the high computational cost associated with the simulation-based optimisation.

8.1.1 Introduction of Resilience to Location Allocation (LA) Problems

We introduce two novels, non-linear formulations of LA problem that extends the primary aspect of locating facilities and the allocating demand to capture the aspect of resilience. Resilience here is the option of providing backup services to customers to ensure uninterruptible supply of demand. We called these formulations Location-allocation resilience problem (LARP) 3.2.1 and Location-allocation resilience problem with restriction (LARPR) 3.2.2 which is a constrained version of LARP. Other aspects captured in the formulations include closing down of profitless and inefficient existing facilities, costs involved in reassigning customers from closed facilities to opened facilities, step costs function involved in determining the bandwidth required by a facility based on the demand exerted on the facility from customers, and the running cost of facilities. This contribution address objective (O1) in part. Most of the work relating to this contribution can be found in Chapter 3.

8.1.2 New problem instance for studying new formulations of Location-Allocation Problem

Due to unique characteristics of our two new LA problem formulations (LARP 3.2.1, LARPR 3.2.2), we are unable to study the formulations using existing problem instances such as uncapacitated warehouse Location problem set [13] which do not reflect the unique aspect of resilience. We, therefore, generate a new problem instance typical of a real-world telecommunication problem. By using the new problem instance which reflects the new characteristics of our LA problem formulations, we are adequately able to study the new formulations. Detailed work on this contribution can be found in Chapter 3. This contribution addresses objective (O2) in part.

8.1.3 Application of Population-based incremental learning (PBIL) algorithm to solve Location-Allocation Problems

PBIL has been recorded in the literature to perform well on many combinatorial problems. However, to the best of our knowledge, PBIL has never been applied to tackling the LA problem in the literature. We, therefore, introduced the PBIL to solve the new formulations. Our motivation in selecting PBIL was in twofold: Firstly, we observed that specifically for the new LA problem formulation, useful problem knowledge could be encoded directly into the probabilistic model of PBIL to aid in finding an optimal or near-optimal solution. Secondly, PBIL has the potential benefits of an EDA, such as employing a probabilistic model that reveals much information about the problem being solved but with a lightweight (univariate) modelling cost. To the best of our knowledge, PBIL has never been applied to tackling the LA problem in the literature.

We compared PBIL to 24 variants of Genetic Algorithms (GA) generated by combining the components of four GAs presented in the literature for solving LA problem formulations. Experiments showed PBIL to outperform all GA variants on all problem instances. Detailed work showing the effectiveness of PBIL for solving LA problem formulations can be found in Chapters 3, 5, 6 and 7. This contribution addresses objective (O3).

8.1.4 Introduction of Dynamic-Customer Location-Allocation Problem in the context of Robust Optimisation Over Time

We introduced a dynamic formulation of the LA problem called Dynamic-customer Location-allocation (DC-LA) problem. DC-LA problem does not fall in the domain of dynamic optimisation but rather robust optimisation over time [175], as described in Section 2.3.1. In DC-LAP facilities are established once at the start of the defined horizon and are expected to be operable and perform satisfactorily to the end of the defined period. DC-LA problem considers the potential movement of customers over a given time horizon. It does this by generating movement dates for a customer over a defined period. The movement dates indicate the time a customer will relocate from their current city to a new city. The model drives the movement of customers using the attractivity of a city to a customer. DC-LA problem assumes that the attractiveness of a city in the future is unknown; hence, it randomly generates the attraction of cities within the simulation. When a customer has to move cities, the choice of a new city is driven by how attractive the new city is to the customer. Once a customer has relocated to a new city, the new costs for the customer are calculated as the euclidean distance between a customer and facility. The customer is then assigned to the facility that offers the customer the least cost of service. DC-LA helps to capture and study the real-world properties of LA problem within the telecommunication industry. Even though DC-LA problem is motivated by a real-world problem from the telecommunications industry, it can be extended to other location problems. Most of the work relating to this contribution can be found in Chapters 4, 5, 6, and 7. This contribution addresses objective (O1) in part.

8.1.5 Introduction of a Stochastic Simulation Model to simulate customer movements in Dynamic-Customer Location-Allocation Problem

We develop a simulation model for simulating the potential movement of customers over a planning period. The simulation model is based on the assumption that customers will move over time *i.e.* disappear from a location and reappear in another location. The model also assumes that the attraction rate of each city in the future is unknown and hence randomly generates the attraction of cities with the model. Most of the work relating to this contribution can be found in Chapters 4, 5, 6, and 7. This contribution addresses objective (O4).

8.1.6 New problem instance for studying new dynamic formulation of Location-Allocation Problem

We develop a new set of problem instances to study the new dynamic formulation of the LA problem. An instance of DC-LA problem comprises of three main parameters: the number of facility locations m, number of customer n and the movement rate mr. Real-world, telecommunication problems motivate the choice of these parameters. By combining the parameters of DC-LA problem, we generate 1440 problem instances to help study DC-LA problem. Most of the work relating to this contribution can be found in Chapters 5, and 6. This contribution addresses objective (O1) in part.

8.1.7 Adaptation of Racing to reduce the high computational cost associated with the Simulation-based optimisation

Racing was first proposed in machine learning to deal with the problem of model selection [115]. Racing was then adapted by [17] for the configuration of the optimisation algorithm. We adapt the concept of racing as a selection approach to help address the high computational cost that comes with the simulation-based optimisation. Our adaptation of racing works by comparing the fitnesses of solutions in the search process using a statistical test. The innovation of our adaptation of racing is in its truncation mechanism, which strives to use the least number of simulations in a simulation-based optimisation where the cost function is stochastic. Experiments showed racing to reduce the computational cost by 4.5 times when compared to the stochastic dynamic evaluation function. We also observed that racing improved on the cost-savings achieved by the stochastic dynamic evaluation function. The improved performance of racing in terms of computational time and cost-savings is due to the ability of racing to discard poor solutions at the early stages of the search process and the ability of racing to use the minimum number of simulation to find statistical differences between solutions. Most of the work relating to this contribution can be found in Chapters 6 and 7. This contribution addresses objective (O5) in part.

8.2 Future Work

This Section highlights the limitations of the work presented in this thesis and outlines some areas for consideration for future work.

The choice of using a Monte Carlo simulation approach attracts a high computational cost due to a large number of simulations employed. In this work, we manage to reduce the high computational cost by adopting the concept of racing to reduce the number of simulations required for finding a robust solution. So a further step would be to explore how the Maximum-likelihood solution (MLS) generated from the probability vector (PV) of PBIL could be used to reduce the computational effort further. A possible area of study in this regard will be to evolve the PV using a single but different simulation in each generation and observe the outcome. Also, in this work, we employed the Friedman rank test and the Holm's procedure as a complementary post hoc test for testing for the statistical difference between solutions in racing. The statistical power of the Friedman test is dependent on the sample size, i.e. a larger sample size has more statistical power. Hence, before we can perform the initial statistical test in the race, we have to evaluate each solution in the population 20 times to give us enough of a sample size for testing. Future work will explore other non-parametric statistical tests other than the Friedman test to see if we can find statistical differences between solutions with a lesser number of simulations.

The successful performance of PBIL on LARP, LARPR and DC-LA problem formulations highlights the importance of exploring other Estimation of distribution algorithms. Although PBIL has been shown in the literature to perform well on many combinatorial problems, other EDA's that are bi-variate in nature has been shown to improve on the performance of uni-variate EDA's such as PBIL especially in considering the resilience constraints and strong trend in customer movements of our LA problem formulations. This is because univariate EDAs treat each decision variable independently and hence, they are often not representative enough to provide the best performance. Future work will, therefore, explore bi-variate EDA such as the Hierarchical-Bayesian optimisation algorithm (h-BOA) [134] to see if the results obtained in this work can be improved.

The LA problems formulated in this work all considered discrete locations for establishing facilities. A continuous dynamic LA problem model where the locations of facilities are not determined ahead of time is also an area of interest for further work. Also, we assumed fixed sizes for cities in this work; it will be interesting to examine the effect of different parameters for the city sizes and explore the impact this will have on the problem.

The value in simulating customer movements and the use of racing to help reduce the computational effort of simulating customer movements opens the door to exploring the application of the racing in other fields of study. Additionally, other alternatives aside racing can be explored to tackle DC-LA problem. Finally, we would seek to investigate whether and where the ideas in this thesis could be extended to other varieties of LA problems.

Bibliography

[1] Al-Sultan, K.S., Al-Fawzan, M.A.: A tabu search approach to the uncapacitated facility location problem. Annals of Operations Research **86**, 91–103 (1999)

[2] Alp, O., Erkut, E., Drezner, Z.: An efficient genetic algorithm for the p-median problem. Annals of Operations research **122**(1-4), 21–42 (2003)

[3] Ankrah, R., Lacroix, B., McCall, J., Hardwick, A., Conway, A.: A holistic metric approach to solving the dynamic location-allocation problem. In: International Conference on Innovative Techniques and Applications of Artificial Intelligence. pp. 433–439. Springer (2018)

[4] Ankrah, R., Lacroix, B., McCall, J., Hardwick, A., Conway, A.: Introducing the dynamic customer location-allocation problem. (2019)

[5] Ankrah, R., Regnier-Coudert, O., McCall, J., Conway, A., Hardwick, A.: Performance analysis of ga and pbil variants for real-world location-allocation problems. In: 2018 IEEE Congress on Evolutionary Computation (CEC). pp. 1–8. IEEE (2018)

[6] Armstrong, M.P., Rushton, G., Honey, R., Dalziel, B.T., Lolonis, P., De, S., Densham, P.J.: Decision support for regionalization: A spatial decision support system for regionalizing service delivery systems. Computers, Environment and Urban Systems **15**(1-2), 37–53 (1991)

[7] Arostegui Jr, M.A., Kadipasaoglu, S.N., Khumawala, B.M.: An empirical comparison of tabu search, simulated annealing, and genetic algorithms for facilities location problems. International Journal of Production Economics **103**(2), 742–754 (2006)

[8] Ballou, R.H.: Dynamic warehouse location analysis. Journal of Marketing Research pp. 271–276 (1968)

[9] Baluja, S.: Population-based incremental learning. a method for integrating genetic search based function optimization and competitive learning. Tech. rep., Carnegie-Mellon Univ Pittsburgh Pa Dept Of Computer Science (1994)

[10] Baluja, S.: An empirical comparison of seven iterative and evolutionary function optimization heuristics (1995)

[11] Baluja, S., Caruana, R.: Removing the genetics from the standard genetic algorithm. In: Machine Learning: Proceedings of the Twelfth International Conference. pp. 38–46 (1995)

[12] Bashiri, M., Moslemi, A.: A robust scenario based approach in an uncertain condition applied to location-allocation distribution centers problem (2011)

[13] Beasley, J.E.: Or-library: distributing test problems by electronic mail. Journal of the operational research society **41**(11), 1069–1072 (1990)

[14] Ben-Tal, A., Do Chung, B., Mandala, S.R., Yao, T.: Robust optimization for emergency logistics planning: Risk mitigation in humanitarian relief supply chains. Transportation research part B: methodological **45**(8), 1177–1189 (2011)

[15] Bennett, V.L., Eaton, D.J., Church, R.L.: Selecting sites for rural health workers. Social Science & Medicine **16**(1), 63–72 (1982)

[16] Benneyan, J.C., Musdal, H., Ceyhan, M.E., Shiner, B., Watts, B.V.: Specialty care single and multi-period location–allocation models within the veterans health administration. Socio-economic planning sciences **46**(2), 136–148 (2012)

[17] Birattari, M., Stützle, T., Paquete, L., Varrentrapp, K.: A racing algorithm for configuring metaheuristics. In: Proceedings of the 4th Annual Conference on Genetic and Evolutionary Computation. pp. 11–18. Morgan Kaufmann Publishers Inc. (2002)

[18] Bish, D.R., Sherali, H.D.: Aggregate-level demand management in evacuation planning. European Journal of Operational Research **224**(1), 79–92 (2013)

[19] Bojić, S., atkov, ., Brcanov, D., Georgijević, M., Martinov, M.: Location allocation of solid biomass power plants: Case study of vojvodina. Renewable and Sustainable Energy Reviews **26**, 769–775 (2013)

[20] Boudahri, F., Sari, Z., Maliki, F., Bennekrouf, M.: Design and optimization of the supply chain of agri-foods: Application distribution network of chicken meat. In: 2011 International conference on communications, computing and control applications (CCCA). pp. 1–6. IEEE (2011)

[21] Bowen, W.M., Weeks, K.D., Batra, D., Hill, T.R.: A deep knowledge planning decision support system for aiding nuclear waste transportation decisions. Computers, environment and urban systems **13**(1), 15–27 (1989)

[22] Branke, J.: Evolutionary optimization in dynamic environments, vol. 3. Springer Science & Business Media (2012)

[23] Bretschneider, S., Kimms, A.: Pattern-based evacuation planning for urban areas. European Journal of Operational Research **216**(1), 57–69 (2012)

[24] Brimberg, J., Mladenovic, N.: Solving the continuous location-allocation problem with tabu search. Studies in Locational Analysis **8**(23-32), 41 (1996)

[25] Calık, H., Alumur, S.A., Kara, B.Y., Karasan, O.E.: A tabu-search based heuristic for the hub covering problem over incomplete hub networks. Computers & Operations Research **36**(12), 3088–3096 (2009)

[26] Canel, C., Khumawala, B.M.: Multi-period international facilities location: An algorithm and application. International Journal of Production Research **35**(7), 1891–1910 (1997)

[27] Canós, M.J., Ivorra, C., Liern, V.: An exact algorithm for the fuzzy p-median problem. European Journal of Operational Research **116**(1), 80–86 (1999)

[28] Captivo, M.E.: Fast primal and dual heuristics for the p-median location problem. European Journal of Operational Research **52**(1), 65–74 (1991)

[29] Caunhye, A.M., Nie, X., Pokharel, S.: Optimization models in emergency logistics: A literature review. Socio-economic planning sciences **46**(1), 4–13 (2012)

[30] Ceselli, A.: Two exact algorithms for the capacitated p-median problem. Quarterly Journal of the Belgian, French and Italian Operations Research Societies **1**(4), 319–340 (2003)

[31] Chanta, S., Sangsawang, O.: Shelter-site selection during flood disaster. Lect. Notes Manag. Sci **4**, 282–288 (2012)

[32] Chaudhry, S.S., He, S., Chaudhry, P.E.: Solving a class of facility location problems using genetic algorithms. Expert Systems **20**(2), 86–91 (2003)

[33] Chen, J.F.: A hybrid heuristic for the uncapacitated single allocation hub location problem. Omega **35**(2), 211–220 (2007)

[34] Chen, X., Zhong, C.y.: An improved genetic algorithm for location problem of logistic distribution center for perishable products. In: International Asia conference on industrial engineering and management innovation (IEMI2012) proceedings. pp. 949–959. Springer (2013)

[35] Chiyoshi, F., Galvao, R.D.: A statistical analysis of simulated annealing applied to the p-median problem. Annals of Operations Research **96**(1-4), 61–74 (2000)

[36] Church, R., ReVelle, C.: The maximal covering location problem. Papers in regional science **32**(1), 101–118 (1974)

[37] Cocking, C., Flessa, S., Reinelt, G.: Improving access to health facilities in nouna district, burkina faso. Socio-economic planning sciences **46**(2), 164–172 (2012)

[38] Cohen, M.A., Fisher, M., Jaikumar, R.: International manufacturing and distribution networks: a normative model framework. Managing international manufacturing **13**, 67–93 (1989)

[39] Cohen, M.A., Lee, H.L.: Resource deployment analysis of global manufacturing and distribution networks. Journal of manufacturing and operations management **2**(2), 81–104 (1989)

[40] Contreras, I.A., Díaz, J.A.: Scatter search for the single source capacitated facility location problem. Annals of Operations Research **157**(1), 73–89 (2008)

[41] Cooper, L.: Location-allocation problems. Operations research **11**(3), 331–343 (1963)

[42] Correa, E.S., Steiner, M.T.A., Freitas, A.A., Carnieri, C.: A genetic algorithm for the p-median problem. In: Proceedings of the 3rd Annual Conference on Genetic and Evolutionary Computation. pp. 1268–1275. Morgan Kaufmann Publishers Inc. (2001)

[43] Côté, M.J., Syam, S.S., Vogel, W.B., Cowper, D.C.: A mixed integer programming model to locate traumatic brain injury treatment units in the department of veterans affairs: A case study. Health care management science **10**(3), 253–267 (2007)

[44] Cunha, C.B., Silva, M.R.: A genetic algorithm for the problem of configuring a hub-and-spoke network for a ltl trucking company in brazil. European Journal of Operational Research **179**(3), 747–758 (2007)

[45] Daskin, M.S., Hopp, W.J., Medina, B.: Forecast horizons and dynamic facility location planning. Annals of Operations Research **40**(1), 125–151 (1992)

[46] Delmaire, H., Díaz, J.A., Fernández, E., Ortega, M.: Reactive grasp and tabu search based heuristics for the single source capacitated plant location problem. INFOR: Information Systems and Operational Research **37**(3), 194–225 (1999)

[47] Derrac, J., García, S., Molina, D., Herrera, F.: A practical tutorial on the use of nonparametric statistical tests as a methodology for comparing evolutionary and swarm intelligence algorithms. Swarm and Evolutionary Computation **1**(1), 3–18 (2011)

[48] Díaz, J.A., Fernandez, E.: Hybrid scatter search and path relinking for the capacitated p-median problem. European Journal of Operational Research **169**(2), 570–585 (2006)

[49] Du, F., Evans, G.W.: A bi-objective reverse logistics network analysis for post-sale service. Computers & Operations Research **35**(8), 2617–2634 (2008)

[50] Duhamel, C., Santos, A.C., Brasil, D., Châtelet, E., Birregah, B.: Connecting a population dynamic model with a multi-period location-allocation problem for post-disaster relief operations. Annals of Operations Research **247**(2), 693–713 (2016)

[51] Farahani, R.Z., Abedian, M., Sharahi, S.: Dynamic facility location problem. In: Facility Location, pp. 347–372. Springer (2009)

[52] Farahani, R.Z., Asgari, N., Heidari, N., Hosseininia, M., Goh, M.: Covering problems in facility location: A review. Computers & Industrial Engineering **62**(1), 368–407 (2012)

[53] Farahani, R.Z., Drezner, Z., Asgari, N.: Single facility location and relocation problem with time dependent weights and discrete planning horizon. Annals of Operations Research **167**(1), 353–368 (2009)

[54] Farahani, R.Z., Hekmatfar, M.: Facility location: concepts, models, algorithms and case studies. Springer (2009)

[55] Farahani, R.Z., Hekmatfar, M., Fahimnia, B., Kazemzadeh, N.: Hierarchical facility location problem: Models, classifications, techniques, and applications. Computers & Industrial Engineering **68**, 104–117 (2014)

[56] Feldman, E., Lehrer, F., Ray, T.: Warehouse location under continuous economies of scale. Management Science **12**(9), 670–684 (1966)

[57] Filipović, V.: Fine-grained tournament selection operator in genetic algorithms. Computing and Informatics **22**(2), 143–161 (2012)

[58] Folly, K.A.: Performance evaluation of power system stabilizers based on population-based incremental learning (pbil) algorithm. International Journal of Electrical Power & Energy Systems **33**(7), 1279–1287 (2011)

[59] Folly, K.A.: An improved population-based incremental learning algorithm. International Journal of Swarm Intelligence Research (IJSIR) **4**(1), 35–61 (2013)

[60] Fortz, B.: Location problems in telecommunications. In: Location science, pp. 537–554. Springer (2015)

[61] Friedman, M.: The use of ranks to avoid the assumption of normality implicit in the analysis of variance. Journal of the american statistical association **32**(200), 675–701 (1937)

[62] Friedman, M.: A comparison of alternative tests of significance for the problem of m rankings. The Annals of Mathematical Statistics **11**(1), 86–92 (1940)

[63] Fu, H., Sendhoff, B., Tang, K., Yao, X.: Robust optimization over time: Problem difficulties and benchmark problems. IEEE Transactions on Evolutionary Computation **19**(5), 731–745 (2015)

[64] Gama, M., Santos, B.F., Scaparra, M.P.: A multi-period shelter location-allocation model with evacuation orders for flood disasters. EURO Journal on Computational Optimization **4**(3-4), 299–323 (2016)

[65] García-Martínez, C., Lozano, M.: Evaluating a local genetic algorithm as context-independent local search operator for metaheuristics. Soft Computing **14**(10), 1117–1139 (2010)

[66] Ghosh, A., Craig, C.S.: A location allocation model for facility planning in a competitive environment. Geographical analysis **16**(1), 39–51 (1984)

[67] Ghosh, D.: Neighborhood search heuristics for the uncapacitated facility location problem. European Journal of Operational Research **150**(1), 150–162 (2003)

[68] Gibbons, J.D., Chakraborti, S.: Nonparametric statistical inference. Springer (2011)

[69] Glover, F.: A template for scatter search and path relinking. Lecture notes in computer science **1363**, 13–54 (1998)

[70] Glover, F., Laguna, M.: Tabu search. In: Handbook of combinatorial optimization, pp. 2093–2229. Springer (1998)

[71] Goncharov, E., Kochetov, Y.: Probabilistic tabu search for the unconstrained discrete optimization problems. Discrete Analysis and Operations Research **9**(2), 13–30 (2002)

[72] Gong, D., Gen, M., Yamazaki, G., Xu, W.: Hybrid evolutionary method for capacitated location-allocation problem. Computers & industrial engineering **33**(3-4), 577–580 (1997)

[73] Görmez, N., Köksalan, M., Salman, F.S.: Locating disaster response facilities in istanbul. Journal of the Operational Research Society **62**(7), 1239–1252 (2011)

[74] Gray, A.: Modeling the selection of markets and the migration of manufacturing across countries. In: Proceedings of the Symposium on Globalization of Operations Management. pp. 7–8. Georgetown University Washington, DC (1993)

[75] Guimarães, V.d.A., Ribeiro, G.M., Forte, V.L.d., Lucena, A.: A location-allocation model for logistics integration centers. International Journal of Transport Economics= Rivista Internazionale de Economia dei Trasporti **44**(2) (2017)

[76] Haase, K., Müller, S.: Management of school locations allowing for free school choice. Omega **41**(5), 847–855 (2013)

[77] Hakimi, S.L.: Optimum locations of switching centers and the absolute centers and medians of a graph. Operations research **12**(3), 450–459 (1964)

[78] Hakimi, S.L.: Optimum distribution of switching centers in a communication network and some related graph theoretic problems. Operations research **13**(3), 462–475 (1965)

[79] Hamadani, A.Z., Ardakan, M.A., Rezvan, T., Honarmandian, M.M.: Location-allocation problem for intra-transportation system in a big company by using meta-heuristic algorithm. Socio-Economic Planning Sciences **47**(4), 309–317 (2013)

[80] Hansen, P., Mladenović, N.: Variable neighborhood search: Principles and applications. European journal of operational research **130**(3), 449–467 (2001)

[81] Haug, P.: A multiple-period, mixed-integer-programming model for multinational facility location. Journal of Management **11**(3), 83–96 (1985)

[82] Hauschild, M., Pelikan, M.: An introduction and survey of estimation of distribution algorithms. Swarm and evolutionary computation **1**(3), 111–128 (2011)

[83] Hong, J.D., Xie, Y., Jeong, K.Y.: Development and evaluation of an integrated emergency response facility location model. Journal of Industrial Engineering and Management (JIEM) **5**(1), 4–21 (2012)

[84] Hosage, C., Goodchild, M.: Discrete space location-allocation solutions from genetic algorithms. Annals of Operations Research **6**(2), 35–46 (1986)

[85] Hribar, M., Daskin, M.S.: A dynamic programming heuristic for the p-median problem. European Journal of Operational Research **101**(3), 499–508 (1997)

[86] Hu, F., Yang, S., Xu, W.: A non-dominated sorting genetic algorithm for the location and districting planning of earthquake shelters. International Journal of Geographical Information Science **28**(7), 1482–1501 (2014)

[87] Jahantigh, F.F., Malmir, B.: A hybrid genetic algorithm for solving facility location allocation problem. In: Proceedings of the 2016 International Conference on Industrial Engineering and Operations Management, Kuala Lumpur, Malaysia (2016)

[88] Jaramillo, J.H., Bhadury, J., Batta, R.: On the use of genetic algorithms to solve location problems. Computers & Operations Research **29**(6), 761–779 (2002)

[89] Järvinen, P., Rajala, J., Sinervo, H.: A branch-and-bound algorithm for seeking the p-median. Operations Research **20**(1), 173–178 (1972)

[90] Jia, H., Ordóñez, F., Dessouky, M.: A modeling framework for facility location of medical services for large-scale emergencies. IIE transactions **39**(1), 41–55 (2007)

[91] Jia, H., Ordóñez, F., Dessouky, M.M.: Solution approaches for facility location of medical supplies for large-scale emergencies. Computers & Industrial Engineering **52**(2), 257–276 (2007)

[92] Jin, Y., Branke, J., et al.: Evolutionary optimization in uncertain environments-a survey. IEEE Transactions on evolutionary computation **9**(3), 303–317 (2005)

[93] Karray, F., Karray, F.O., De Silva, C.W.: Soft computing and intelligent systems design: theory, tools, and applications. Pearson Education (2004)

[94] Keskin, B.B., Üster, H.: Meta-heuristic approaches with memory and evolution for a multi-product production/distribution system design problem. European Journal of Operational Research **182**(2), 663–682 (2007)

[95] Keskin, B.B., Üster, H.: A scatter search-based heuristic to locate capacitated transshipment points. Computers & Operations Research **34**(10), 3112–3125 (2007)

[96] King, M., Low, D.: Measuring the"world"real interest rate. Tech. rep., National Bureau of Economic Research (2014)

[97] Kochetov, Y., Stolyar, A.: Evolutionary local search with variable neighborhood for the resource constrained project scheduling problem. In: Proceedings of the 3rd international workshop of computer science and information technologies. vol. 132 (2003)

[98] Kocoloski, M., Griffin, W.M., Matthews, H.S.: Impacts of facility size and location decisions on ethanol production cost. Energy Policy **39**(1), 47–56 (2011)

[99] Korac, V.M., Kratica, J., Savić, A.: An improved genetic algorithm for the multi level uncapacitated facility location problem. International Journal of Computers Communications & Control **8**(6), 845–853 (2013)

[100] Korkmaz, S., Babalik, A., Kiran, M.S.: An artificial algae algorithm for solving binary optimization problems. International Journal of Machine Learning and Cybernetics **9**(7), 1233–1247 (2018)

[101] Kratica, J.: Solving the uncapacitated warehouse location problem using a simple genetic algorithm. In: Proc. XIV ECPD International Conf. on Material Handling and Warehousing, 1996 (1996)

[102] Kratica, J., Tošic, D., Filipović, V., Ljubić, I.: Solving the simple plant location problem by genetic algorithm. RAIRO-Operations Research **35**(1), 127–142 (2001)

[103] Kuenne, R.E., Soland, R.M.: Exact and approximate solutions to the multisource weber problem. Mathematical Programming **3**(1), 193–209 (1972)

[104] Lacroix, B., McCall, J., Lonchampt, J.: Iterated racing algorithm for simulation-optimisation of maintenance planning. In: 2018 IEEE Congress on Evolutionary Computation (CEC). pp. 1–7. IEEE (2018)

[105] Laporte, G., Nickel, S., da Gama, F.S.: Location science, vol. 528. Springer (2015)

[106] Lee, S.M., Green, G.I., Kim, C.S.: A multiple criteria model for the location-allocation problem. Computers & Operations Research **8**(1), 1–8 (1981)

[107] Levanova, T.V., Loresh, M.: Algorithms of ant system and simulated annealing for the p-median problem. Automation and remote control **65**(3), 431–438 (2004)

[108] Levina, E., Bickel, P.: The earth mover's distance is the mallows distance: Some insights from statistics. In: Proceedings Eighth IEEE International Conference on Computer Vision. ICCV 2001. vol. 2, pp. 251–256. IEEE (2001)

[109] Li, A.C., Nozick, L., Xu, N., Davidson, R.: Shelter location and transportation planning under hurricane conditions. Transportation Research Part E: Logistics and Transportation Review **48**(4), 715–729 (2012)

[110] Lin, C.K.Y.: Solving a location, allocation, and capacity planning problem with dynamic demand and response time service level. Mathematical Problems in Engineering **2014** (2014)

[111] Lin, Y.H., Batta, R., Rogerson, P.A., Blatt, A., Flanigan, M.: Location of temporary depots to facilitate relief operations after an earthquake. Socio-Economic Planning Sciences **46**(2), 112–123 (2012)

[112] Lolonis, P., Armstrong, M.P.: Location-allocation models as decision aids in delineating administrative regions. Computers, environment and urban systems **17**(2), 153–174 (1993)

[113] Maranzana, F.: On the location of supply points to minimize transport costs. Journal of the Operational Research Society **15**(3), 261–270 (1964)

[114] Marić, M.: An efficient genetic algorithm for solving the multi-level uncapacitated facility location problem. Computing and Informatics **29**(2), 183–201 (2012)

[115] Maron, O., Moore, A.W.: The racing algorithm: Model selection for lazy learners. Artificial Intelligence Review **11**(1-5), 193–225 (1997)

[116] Maxwell, B., Anderson, S.: Training hidden markov models using population-based learning. In: Genetic and Evolutionary Computation Conference, GECCO-99. vol. 994 (1999)

[117] McCormack, R., Coates, G.: A simulation model to enable the optimization of ambulance fleet allocation and base station location for increased patient survival. European Journal of Operational Research **247**(1), 294–309 (2015)

[118] Meng, X., Li, J., Zhou, M., Dai, X., Dou, J.: Population-based incremental learning algorithm for a serial colored traveling salesman problem. IEEE Transactions on Systems, Man, and Cybernetics: Systems **48**(2), 277–288 (2016)

[119] Michel, L., Van Hentenryck, P.: A simple tabu search for warehouse location. European Journal of Operational Research **157**(3), 576–591 (2004)

[120] MirHassani, S., Ebrazi, R.: A flexible reformulation of the refueling station location problem. Transportation Science **47**(4), 617–628 (2012)

[121] Mladenović, N., Brimberg, J., Hansen, P., Moreno-Pérez, J.A.: The p-median problem: A survey of metaheuristic approaches. European Journal of Operational Research **179**(3), 927–939 (2007)

[122] Mladenović, N., Hansen, P.: Variable neighborhood search. Computers & operations research **24**(11), 1097–1100 (1997)

[123] Møller-Jensen, L.: Assessing spatial aspects of school location-allocation in copenhagen. Geografisk Tidsskrift-Danish Journal of Geography **98**(1), 71–80 (1998)

[124] Murray, A.T.: Advances in location modeling: Gis linkages and contributions. Journal of geographical systems **12**(3), 335–354 (2010)

[125] Murray, A.T., Church, R.L.: Applying simulated annealing to location-planning models. Journal of Heuristics **2**(1), 31–53 (1996)

[126] Ndiaye, F., Ndiaye, B.M., Ly, I.: Application of the p-median problem in school allocation. American Journal of Operations Research **2**(02), 253 (2012)

[127] Neebe, A.W.: Branch and bound algorithm for the p-median transportation problem. Journal of the Operational Research Society **29**(10), 989–995 (1978)

[128] Neema, M.N., Maniruzzaman, K.M., Ohgai, A.: Application of advanced hybrid genetic algorithms for optimal locations of high school. In: International Conference on Computational Science and Its Applications. pp. 294–303. Springer (2013)

[129] Oppong, J.R.: Accommodating the rainy season in third world location-allocation applications. Socio-Economic Planning Sciences **30**(2), 121–137 (1996)

[130] Ossenbruggen, P.J., Ossenbruggen, P.C.: Swap: a computer package for solid waste management. Computers, Environment and Urban Systems **16**(2), 83–100 (1992)

[131] Owen, S.H., Daskin, M.S.: Strategic facility location: A review. European journal of operational research **111**(3), 423–447 (1998)

[132] Pacheco, J.A., Casado, S.: Solving two location models with few facilities by using a hybrid heuristic: a real health resources case. Computers & operations research **32**(12), 3075–3091 (2005)

[133] Patel, N.R.: Locating rural social service centers in india. Management Science **25**(1), 22–30 (1979)

[134] Pelikan, M.: Hierarchical bayesian optimization algorithm. In: Hierarchical Bayesian Optimization Algorithm, pp. 105–129. Springer (2005)

[135] Perpina, C., Alfonso, D., Pérez-Navarro, A., Penalvo, E., Vargas, C., Cárdenas, R.: Methodology based on geographic information systems for biomass logistics and transport optimisation. Renewable Energy **34**(3), 555–565 (2009)

[136] Pizzolato, N.D.: A heuristic for large-sizep-median location problems with application to school location. Annals of operations research **50**(1), 473–485 (1994)

[137] Rahman, S.U.: Location-allocation modelling for primary health care provision in Bangladesh. Ph.D. thesis, University of Exeter (1991)

[138] Rawls, C.G., Turnquist, M.A.: Pre-positioning planning for emergency response with service quality constraints. OR spectrum **33**(3), 481–498 (2011)

[139] Reeves, C.R.: A genetic algorithm for flowshop sequencing. Computers & operations research **22**(1), 5–13 (1995)

[140] Reinhelt, G.: {TSPLIB}: a library of sample instances for the tsp (and related problems) from various sources and of various types. URL: http://comopt. ifi. uniheidelberg. de/software/TSPLIB95 (2014)

[141] Rentizelas, A.A., Tolis, A.J., Tatsiopoulos, I.P.: Logistics issues of biomass: The storage problem and the multi-biomass supply chain. Renewable and sustainable energy reviews **13**(4), 887–894 (2009)

[142] Resende, M.G., Werneck, R.F.: A hybrid heuristic for the p-median problem. Journal of heuristics **10**(1), 59–88 (2004)

[143] Rogers, J., Brammer, J.G.: Analysis of transport costs for energy crops for use in biomass pyrolysis plant networks. Biomass and Bioenergy **33**(10), 1367–1375 (2009)

[144] Rosing, K.E.: An optimal method for solving the (generalized) multi-weber problem. European Journal of Operational Research **58**(3), 414–426 (1992)

[145] Rushton, G., Goodchild, M.F., Ostresh, L.M.: Computer programs for location-allocation problems. Department of Geography, University of Iowa (1973)

[146] Saeidian, B., Mesgari, M.S., Ghodousi, M.: Evaluation and comparison of genetic algorithm and bees algorithm for location–allocation of earthquake relief centers. International Journal of Disaster Risk Reduction **15**, 94–107 (2016)

[147] Salhi, S.: A perturbation heuristic for a class of location problems. Journal of the Operational Research Society **48**(12), 1233–1240 (1997)

[148] Salhi, S.: Defining tabu list size and aspiration criterion within tabu search methods. Computers & Operations Research **29**(1), 67–86 (2002)

[149] Salhi, S., Atkinson, R.: Subdrop: A modified drop heuristic for location problems. Location Science **3**(4), 267–273 (1995)

[150] Salhi, S., Gamal, M.: A genetic algorithm based approach for the uncapacitated continuous location–allocation problem. Annals of Operations Research **123**(1-4), 203–222 (2003)

[151] Sánchez-Lozano, J.M., Teruel-Solano, J., Soto-Elvira, P.L., García-Cascales, M.S.: Geographical information systems (gis) and multi-criteria decision making (mcdm) methods for the evaluation of solar farms locations: Case study in south-eastern spain. Renewable and Sustainable Energy Reviews **24**, 544–556 (2013)

[152] Santos, M.G., Meriño, M., Sore, N., Quevedo, V.C.: Flood facility location-allocation in marikina city using mclp with lagrange. In: Proc. of 14th Asia Pacific Industrial Engineering and Management Systems Conference (2013)

[153] dos Santos Coelho, L., Grebogi, R.B.: Chaotic synchronization using pid control combined with population based incremental learning algorithm. Expert Systems with Applications **37**(7), 5347–5352 (2010)

[154] Sayarshad, H.R., Chow, J.Y.: Non-myopic relocation of idle mobility-on-demand vehicles as a dynamic location-allocation-queueing problem. Transportation Research Part E: Logistics and Transportation Review **106**, 60–77 (2017)

[155] Scheuerer, S., Wendolsky, R.: A scatter search heuristic for the capacitated clustering problem. European Journal of Operational Research **169**(2), 533–547 (2006)

[156] Schoemaker, P.J.: Multiple scenario development: Its conceptual and behavioral foundation. Strategic management journal **14**(3), 193–213 (1993)

[157] Scott, A.J.: Location-allocation systems: a review. Geographical Analysis **2**(2), 95–119 (1970)

[158] Sha, Y., Huang, J.: The multi-period location-allocation problem of engineering emergency blood supply systems. Systems Engineering Procedia **5**, 21–28 (2012)

[159] Shariff, S.R., Moin, N.H., Omar, M.: Location allocation modeling for healthcare facility planning in malaysia. Computers & Industrial Engineering **62**(4), 1000–1010 (2012)

[160] Sinangil, S., Musdal, H., Ceyhan, M.E., Benneyan, J.C., Watts, B.V., Shiner, B.: Service location models in the va healthcare system. In: IIE Annual Conference. Proceedings. p. 1. Institute of Industrial and Systems Engineers (IISE) (2011)

[161] Stanimirović, Z., Kratica, J., Dugošija, D.: Genetic algorithms for solving the discrete ordered median problem. European Journal of Operational Research **182**(3), 983–1001 (2007)

[162] Taillard, É.D.: Heuristic methods for large centroid clustering problems. Journal of heuristics **9**(1), 51–73 (2003)

[163] Tohyama, H., Ida, K., Matsueda, J.: A genetic algorithm for the uncapacitated facility location problem. Electronics and Communications in Japan **94**(5), 47–54 (2011)

[164] Tong, D., Ren, F., Mack, J.: Locating farmers markets with an incorporation of spatio-temporal variation. Socio-Economic Planning Sciences **46**(2), 149–156 (2012)

[165] Uzunlar, O., Ceyhan, M.E., Benneyan, J.C., Watts, B.V., Shiner, B.: Optimal longitudinal relocation of colonoscopy services within the veterans health administration. In: IIE Annual Conference. Proceedings. p. 1. Institute of Industrial and Systems Engineers (IISE) (2012)

[166] Verma, A., Gaukler, G.M.: Pre-positioning disaster response facilities at safe locations: An evaluation of deterministic and stochastic modeling approaches. Computers & Operations Research **62**, 197–209 (2015)

[167] Wade, W.: Scenario planning: A field guide to the future. John Wiley & Sons (2012)

[168] Wang, Y.W., Wang, C.R.: Locating passenger vehicle refueling stations. Transportation Research Part E: Logistics and Transportation Review **46**(5), 791–801 (2010)

[169] Wesolowsky, G.O., Truscott, W.G.: The multiperiod location-allocation problem with relocation of facilities. Management Science **22**(1), 57–65 (1975)

[170] Whitaker, R.: A fast algorithm for the greedy interchange for large-scale clustering and median location problems. INFOR: Information Systems and Operational Research **21**(2), 95–108 (1983)

[171] Xiaohui, Q., Wen, Y.: Studies on spatio-temporal collaboration model for location analysis of vegetable and fruit logistics. In: 2009 Sixth International Conference on Fuzzy Systems and Knowledge Discovery. vol. 5, pp. 619–626. IEEE (2009)

[172] Yang, S., Jiang, Y., Nguyen, T.T.: Metaheuristics for dynamic combinatorial optimization problems. IMA Journal of Management Mathematics **24**(4), 451–480 (2012)

[173] Ye, F., Zhao, Q., Xi, M., Dessouky, M.: Chinese national emergency warehouse location research based on vns algorithm. Electronic Notes in Discrete Mathematics **47**, 61–68 (2015)

[174] Yi, C.S., Lee, J.H., Shim, M.P.: Site location analysis for small hydropower using geo-spatial information system. Renewable Energy **35**(4), 852–861 (2010)

[175] Yu, X., Jin, Y., Tang, K., Yao, X.: Robust optimization over timea new perspective on dynamic optimization problems. In: IEEE Congress on evolutionary computation. pp. 1–6. IEEE (2010)

[176] Zar, J.: Biostatistical analysis. ed. 5ta. new jersey, usa: Ed (2009)

[177] Zhao, L., Li, H., Sun, Y., Huang, R., Hu, Q., Wang, J., Gao, F.: Planning emergency shelters for urban disaster resilience: An integrated location-allocation modeling approach. Sustainability **9**(11), 2098 (2017)

[178] Zhi, J., Keskin, B.B., Melouk, S.H.: A multi-period dynamic location planning model for emergency response. IIE Transactions on Healthcare Systems Engineering **5**(4), 211–224 (2015)