Maria do Céu Lourenço Marques

DYNAMIC LOCATION PROBLEMS UNDER UNCERTAINTY: MODELS AND OPTIMIZATION TECHNIQUES

Tese de doutoramento em Gestão – Ciência Aplicada à Decisão, orientada por Prof. Doutora Joana Matos Dias e apresentada à Faculdade de Economia da Universidade de Coimbra

Março de 2015

· U C ·

Universidade de Coimbra

• U C •

FEUC FACULDADE DE ECONOMIA
UNIVERSIDADE DE COIMBRA

Maria do Céu Lourenço Marques

Dynamic location problems under uncertainty: models and optimization techniques

Tese de Doutoramento em Gestão - Ciência Aplicada à Decisão,

apresentada à Faculdade de Economia da Universidade de Coimbra

para obtenção do grau de Doutor

Orientador: Prof. Doutora Joana Matos Dias

Coimbra, 2015

Abstract

This thesis is devoted to mathematical modelling and solution techniques for dynamic facility location problems under uncertainty. The uncertainty regarding the evolution of important problems' parameters along the planning horizon, such as setup and assignment costs, as well as level or location of demand, is explicitly incorporated into the dynamic models through a finite and discrete set of possible scenarios.

In the present work we first propose a two–stage stochastic model for the uncapacitated problem. The first decisions to be made are the strategic ones, where and when to locate the facilities throughout the planning horizon. The second–stage decisions refer to the assignment of the *existing* customers to the open facilities over the whole planning horizon under each possible scenario. As opposite to location decisions, that must be made here and now and should be valid for all possible future scenarios, assignment can be decided after the uncertainty has been resolved and thus can be adjusted in each time period to each possible scenario. The objective is to find a solution that minimizes the expected total cost over all possible scenarios. This model is then extended to other situations, recognizing that other features should be included in the mathematical model to be able to generate other possible solutions. A set of robust constraints is incorporated into that model, that in spite of restricting the set of admissible solutions, it offers more informed and robust solutions under uncertainty. A multi–objective problem wherein each scenario gives rise to an objective is then developed, and relations with other known problems are established as well. For this latter model, requirements about scenarios probabilities or risk profiles are dropped. Within this context, it is emphasized that the Decision Maker will have a better picture of the compromises that exist among the possible scenarios. In terms of models, we conclude with several extensions considering capacitated facilities. The possibility of unmet demand appears naturally in this class of problems, giving rise to other interesting and challenging questions. We propose and discuss both mono and multi–objective approaches.

We proceed with the description of the solution techniques that have been developed to tackle the uncapacitated problems. First we present a primal-dual heuristic approach inspired on classical works and a branch&bound scheme integrating this same heuristic. Afterwards, a Lagrangean relaxation approach developed to tackle the problem with robust constraints is detailed. The calculation of non–dominated solutions for the multi–

objective problem is discussed and illustrated. Finally, as the models and algorithms were tested over sets of randomly generated problems, the computational experiments and results obtained are provided including comparisons with general solvers.
The results of this work aim to help Decision Makers in the difficult process of decision making when dealing with location problems under uncertainty, and thus should be interpreted as decision support tools.

keywords: dynamic location problems, uncertainty, scenarios, primal-dual heuristics, optimization

Resumo

Esta tese versa sobre modelação matemática e algoritmos de resolução de problemas de localização dinâmica em contextos de incerteza. A incerteza acerca de como importantes parâmetros dos problemas irão evoluir ao longo do tempo, tais como custos de instalação de serviços e de afetação, localização ou nível da procura, é explicitamente incorporada nos modelos dinâmicos através de um conjunto finito e discreto de cenários.

Na presente dissertação, propomos em primeiro lugar um modelo estocástico de duas fases para o problema de localização sem restrições de capacidades. As primeiras decisões a serem tomadas são as estratégicas, onde e quando localizar os serviços ao longo do horizonte temporal. As decisões de segunda fase referem-se à afetação dos clientes com procura aos serviços abertos ao longo do horizonte temporal para todos os cenários possíveis. Ao contrário das decisões de localização, tomadas no presente e válidas para todos os futuros possíveis, as decisões de afetação podem ser tomadas após a realização da incerteza e ajustadas em cada período temporal a cada cenário. O objetivo do problema é encontrar uma solução que minimize o custo total esperado para todos os cenários possíveis. Este modelo é depois alargado a outras situações, reconhecendo-se que outras características devem ser incluídas no modelo de modo a gerar outras soluções para o problema. Um conjunto de restrições de robustez é incorporado no modelo que, apesar de restringir o conjunto de soluções admissíveis, oferece soluções mais informadas e robustas em situações de incerteza. Um problema multi–objetivo em que cada cenário origina um objetivo é depois apresentado, assim como relações com outros problemas conhecidos. Requisitos acerca das probabilidades associadas aos cenários ou acerca de perfis de risco são desnecessários. É ainda sublinhado que neste contexto o Agente de Decisão terá um melhor retrato dos compromissos existentes entre os possíveis cenários. Em termos de modelos, concluímos com várias extensões considerando serviços com capacidades limitadas. A possibilidade de procura insatisfeita surge naturalmente nesta classe de problemas, dando lugar a outras interessantes e desafiantes questões. Propomos e discutimos abordagens mono e multi–objetivo.

Procedemos à descrição dos algoritmos construídos para resolução dos problemas sem restrições de capacidades. Apresentamos uma heurística primal–dual inspirada em abordagens clássicas e um algoritmo *branch&bound* que integra aquela heurística. Uma técnica usando relaxação Lagrangeana é depois detalhada para resolução do problema com

as restrições de robustez. O cálculo de soluções não dominadas para o problema multi–objetivo é discutido e ilustrado com um exemplo. Finalmente, como tanto os modelos como os algoritmos foram testados com instâncias geradas aleatoriamente, as experiências e resultados computacionais são apresentados, incluindo comparações com *general solvers.*
Os resultados deste trabalho pretendem ajudar os Agentes de Decisão no difícil processo de decisão perante problemas de localização em contexto de incerteza, e assim devem ser interpretados como ferramentas de apoio à decisão.

palavras–chave: localização dinâmica, incerteza, cenários, heurísticas primais-duais, otimização.

Acknowledgements

I would like to thank my advisor Professor Joana Matos Dias for her guidance and sharing her knowledge in such a vast field. I am grateful for her encouraging and fruitful comments on my work. Acknowledgments also go out for her friendship, as well as her company during two international meetings, which were undoubtedly very rewarding experiences.

I am in debt to the Instituto Politécnico/Instituto Superior de Engenharia de Coimbra (IPC/ISEC) for financial support provided to attend conferences. I would also like to thank the Faculty of Economics of Coimbra for financial support to participate in ISOLDE 2014 held in Italy. To the Department of Physics and Mathematics at ISEC I am grateful for all the very kind help. Acknowledgements go out to Eng. João Alves for helping me during the first stages on MinGW compilers and C–language. Didi, Pascoal and Rui were always attentive with words of encouragement.

To my mother Julia, Filipe, Cátia, David, Lena and to my closest friends Ana Filipa, Gabi and Paula, I express my love.

Contents

Prelude **1**

1 Background and Related Literature **5**

1.1 Some classical facility location problems 5

1.2 Uncertainty modelling 7

1.3 Overview on Single-period and Dynamic facility location problems under uncertainty 10

2 Mathematical Models **15**

2.1 Dynamic uncapacitated location problem under uncertainty 17

2.1.1 Dual problem and complementary slackness conditions 20

2.1.2 Extensions regarding the uncertainty in potential facility sites . . 21

2.2 Dynamic uncapacitated location problem under uncertainty with a regret based measure of robustness 23

2.2.1 Expected total cost versus regret: illustrative examples 25

2.3 Multi–objective dynamic uncapacitated location problem under uncertainty 28

2.4 Dynamic capacitated location problems under uncertainty 31

2.4.1 Mono-objective approaches 32

2.4.2 Multi-objective approaches 37

3 Solution Approaches **43**

3.1 Primal-Dual heuristic 45

3.1.1 Dual ascent procedure 45

3.1.2 Primal procedure 46

3.1.3 Primal–Dual adjustment procedure 47

3.1.4 Illustrative examples 49

3.2 Branch&Bound approach 55

3.3 Lagrangean relaxation approach 56

3.4 Multi–objective approach 58

4 Computational Experiments **63**

4.1 Generation of test problems 64

4.2 Computational results 66
4.2.1 Primal-Dual heuristic 67
4.2.2 Branch&Bound approach 73
4.2.3 Lagrangean relaxation approach 81

Postlude **85**

Bibliography **87**

Prelude

A Facility Location Problem can be seen as the problem of efficiently deciding where to locate equipments/facilities, being public services, such as hospitals or schools, or private services (plants, warehouses,...). The question of *where* to locate may be associated with other questions: what size (capacity) should be established; when to locate; how long to keep the facilities operating; in the case of facilities whose purpose is to meet the needs of a set of customers, how to assign customers to the facilities, etc. Facility location problems have been widely studied by many researchers. From the literature we can witness the diversity of situations considered and the corresponding diversity of models developed, reflecting also the importance of such problems (e. g., Mirchandani and Francis [60], Daskin [18], Revelle et al. [71]). Discrete versus continuous or planar models, deterministic versus stochastic or under uncertainty, static versus dynamic, are only some classes of location models that can be found in the literature (Krarup and Pruzan [48]).

This work is concerned with dynamic discrete facility location problems where uncertainty is explicitly considered through the use of scenarios. Discrete location problems are those problems in which the facilities to be located can only be placed at a finite number of potencial sites selected via some prior analysis (Mirchandani and Francis [60]). The class of models that deal explicitly with the presence of uncertainty is usually called location under uncertainty or stochastic location models. In such cases some of the problem input parameters are only known with uncertainty, as opposed to their deterministic counterparts where all the parameters are assumed to be known precisely. The incorporation of uncertainty into classical (deterministic) location models comes from the recognition that at the time of decision making it may not be possible to know with certainty some of the problem input parameters (level or location of demand, costs, for instance). Considering that most location decision problems are strategic by nature, and that the decisions made are costly to revert, with consequences in the medium and long terms, the uncertainty inherent in most real facility location problems should be explicitly considered and represented in the constructed models (Owen and Daskin [66]). With such models Decision Makers "can better prepare for and respond to" uncertainty in strategic planning (Shapiro [81]).

During the last decades there has been considerable interest in location under uncertainty and a large volume of work is now available in specialized papers and monographs. We can find a primary division between uncertainty and risk decision problems (Rosenhead et al. [76]). In situations under uncertainty no probabilistic information about the uncertain parameters is advanced, whereas in risk decision problems it is assumed a perfect knowledge about probability distributions. However, uncertainty problems may be converted to risk decision problems by the consideration of some probabilistic information, and the term uncertainty has been also used in risk decision situations. Regardless this and other classifications, the works found in the literature may differ in the source of uncertainty (most of them in level or location of demand and/or costs), in the way uncertainty is represented (mainly, stochastic programming and scenario approaches), objective functions considered, solution methods, etc. A review about these challenging problems, where many situations are considered, is given by Snyder [82]. Even so, compared with the research devoted to deterministic versions, the literature related to stochastic location is still much more limited, particularly addressing discrete location problems. As stated by the authors cited above and others, as most deterministic discrete location problems are too complex, formulated as mixed integer programming problems and classified as NP-hard, the incorporation of randomness in such models increases their complexity and hinders its use in the computation of optimal solutions, which makes this class of problems less attractive than deterministic formulations.

Another class of problems within our scope of interest concerns Dynamic (or Multi-period) Location problems. Dynamic models are mainly concerned with planning the location and/or size of facilities over time, such that the time dimension is explicitly represented through the use of time dependent decision variables. Classical (static) models are enriched with the answer to questions such as “when” to locate (Jacobsen [39]). A dynamic location problem approach is usually necessary whenever the assignment costs change significantly during the planning horizon or there are significant costs for relocating facilities (Erlenkotter [30]). Dynamic models may require a large volume of data, which makes them also less attractive and less studied than static problems.

Dynamic and stochastic location models are strongly related. Whenever it is necessary to explicitly consider a planning horizon, uncertainty appears due to unknown future conditions that may lead to a limited knowledge about problem parameters (Owen and Daskin [66]). If the parameters of dynamic location models change deterministically over time, then it is not possible to incorporate the uncertainty inherent in real-world location problems even though time dimension is explicitly represented in the model. Considering both time and uncertainty in location models allows the consideration of more realistic situations, although the resulting models become more complex than static and deterministic ones. Most of the work that has been done addresses single-period (static) deterministic models, static under uncertainty models or deterministic dynamic models, although exploring many different and relevant situations. There has been much less work considering explicitly both time and uncertainty in discrete location models.

The main objective of this work is to support location decision making through the development of mathematical models and algorithms that deal explicitly with the uncertainty inherent in most dynamic facility location problems. The main contributions of this thesis are summarized as follows: (i) development of a new model for the uncapacitated discrete dynamic facility location problem that considers explicitly uncertainty in many of the problem's parameters via a set of scenarios, as well as solution approaches to tackle this problem, first a primal-dual heuristic approach inspired on classical works and then a branch&bound scheme integrating this same heuristic to solve the problem to optimality (ii) development of an extension of the first model considering robustness concerns and also a Lagrangean relaxation approach to tackle the problem (iii) development of a multi-objective approach for the uncapacitated dynamic location problem under uncertainty (iv) development of new models considering capacitated facilities.

Taking into account the vast existing literature on facility location, in Chapter 1 we address different location problems and perspectives that are somehow related to this work. First, in section 1.1 we review some classical (static and deterministic) and deterministic dynamic location problems. Some references to these classes of problems are also provided. Section 1.2 is devoted to the subject of Uncertainty, where Stochastic and Scenario approaches are addressed. We focus on those aspects that are more important to the forthcoming developments. In section 1.3 an overview on past works concerning facility location problems under uncertainty is given. These works address both static and dynamic approaches, from earlier to most recent ones, reflecting the variety and richness of the existing contributions on facility location under uncertainty.

In Chapter 2 we describe new models for discrete dynamic location problems under uncertainty. We generalize some well known location models by incorporating explicitly the uncertainty in these models through a set of scenarios. In section 2.1 we revisit the classical uncapacitated facility location problem (UFLP), proposing a dynamic and uncertain version of this problem. In this model, fixed and assignment costs are scenario dependent, as well as the set of customers and the set of potential locations for facilities. The problem is formulated as an integer linear programming model, that contains the deterministic static and dynamic UFLP as particular problems (*NP-hard* problems (Cornuejols et al. [16])). Taking into account the forthcoming developments in terms of solution approaches to this problem (a primal–dual heuristic) formulations for the dual problem and complementary slackness conditions are given as well. We end this section considering variations in the first model proposed. Due to the assumptions regarding uncertainty in potential facility sites, the model here presented is more general than the first introduced. Afterwards, the first model proposed is further extended to other situations. In section 2.2 a regret based measure of robustness is incorporated and the solutions provided by this problem are analysed through illustrative examples. In section 2.3 a Multi–objective approach is considered and relations with other locations problems are also provided. We advocate here the use of a multi–objective approach as a valuable tool in guiding the decision–making process under uncertainty, as the Decision Maker will have a much broader view of the compromises that exist among the possible scenarios. In section 2.4 we propose and discuss several extensions considering capacitated facilities.

Chapter 3 details the solutions approaches developed to tackle the problems presented in the previous chapter. In section 3.1 a primal-dual heuristic approach to tackle the first model presented is described along with illustrative examples. This heuristic approach is directly inspired on the approaches developed by Bilde and Krarup [13] and Erlenkotter [29], and Van Roy and Erlenkotter [88], designed for the static and dynamic versions of the UFLP, respectively. In section 3.2 this same heuristic is incorporated in a branch&bound algorithm in order to solve the problem to optimality. Afterwards, in section 3.3 a Lagrangean relaxation approach developed to tackle the problem with robustness constraints is described, which uses also the primal-dual heuristic. We end this chapter explaining in section 3.4 how Pareto–efficient solutions for the Multi-objective problem can be calculated following an interactive approach with an illustrative example.

Chapter 4 is devoted to the presentation and discussion of the computational experiences carried out to validate the proposed models and evaluate the performance of the corresponding algorithms both in terms of solution quality and computational time. First, in section 4.1 we discuss briefly the issue of scenarios' generation giving some references to the subject as well. The algorithm developed to generate test problems for the present work is then described. The proposed models and solution techniques were tested over sets of randomly generated test problems. In section 4.2 the computational results are presented. For the models and algorithms described in the previous chapters, we present some details about the solutions obtained for those problems, in particular the quality of the solutions in terms of gap, and also the computational time spent by the algorithms. Comparisons with the results of general solvers are provided as well.

The numbering system used in this work is the common one whereby (2.3.1) refers to the 1st numbered equation in section 3 of chapter 2. An analogous scheme is followed for propositions, figures, tables, etc. All references in the text are in the bibliography chapter ordered alphabetically.

Chapter 1

Background and Related Literature

The literature devoted to facility location problems is immense. Among the vast collection of works concerning location problems, we have chosen to review in this text only those works and perspectives that are somehow related to the location problems tackled in this thesis. Most of these works extend classical (static and deterministic) discrete location problems with differentiating characteristics, in a stochastic or/and dynamic setting. We start with a short review on some classical problems as well as on deterministic dynamic problems. Afterwards, the subject of uncertainty modelling is discussed. The focus goes to two main approaches, the Stochastic and Scenario approaches, given not only their relevance in the location literature but also the forthcoming developments of this thesis. Specially related with the Scenario approach, some notes and references on *robustness* are given. In the following section, we consider previous works that are devoted to discrete location problems under uncertainty (single-period and dynamic). We also review some recent works about supply chain design problems under uncertainty in which location decisions are included.

We stress that this chapter along with the additional works that will be cited throughout this text have no pretensions of completeness. For other references and extensive reviews on facility location under uncertainty, the reader is referred to Louveaux [55], Kouvelis and Yu [47] and Snyder [82].

1.1 Some classical facility location problems

The classical uncapacitated facility location problem (UFLP), also known as the simple plant location problem (SPLP), plays a central role in the location research field, not only by itself but also integrated in other problems. The UFLP consists of deciding where to locate a number of facilities among a finite set of potential sites, in order to minimize total costs (fixed facility costs plus variable production costs and transportation costs to customers). Since the facilities are uncapacitated, all demands will be assigned to the nearest open facility. The size of an open facility is computed as the sum of the demands it serves. The UFLP has been extensively studied since Kuehn and Hamburger [49] and is known to be *NP*-hard (Cornuejols et al. [16]). A well known variation of the UFLP is the capacitated facility location problem (CFLP) in which there is a known upper bound to the capacity of each facility. In terms of formulation it is similar to the UFLP, with additional capacity constraints. It is possible that customers can no longer be assigned to the closest open facility. It is necessary to define if the demand of each customer can be served by more than one open facility, or if it has to be fully assigned to one and only one facility. The p–median problem (introduced by Hakimi [34]) consists of finding the optimal location of exactly p facilities in order to meet a given demand at the lowest possible transportation cost.

The above problems are by far well known and detailed descriptions and its variations along with solution methods (mainly heuristic and approximation algorithms) may be found in several books, papers and in the references therein (e.g., Mirchandani and Francis [60], Daskin [18], Korte and Vygen [46]).

In a dynamic setting, the works found in the literature may differ in the way some timing aspects and other important issues are incorporated and handled. We can find models that consider both the possibility of opening new facilities during the planning horizon, or the closure of facilities that were opened at the beginning of the planning horizon. Most of the times, once a facility is opened, it stays open until the end of the planning horizon. Similarly, once a facility is closed it stays closed until the end of the planning horizon. Nevertheless, there are models that consider more flexible settings where a facility can be opened, closed and even reopened during the planning horizon. There are models that consider capacity constraints or other type of constraints like budget upper bounds. The number and diversity of proposed solution methods is significant. One of the earliest dynamic uncapacitated facility location problem (DUFLP) was proposed by Roodman and Schwarz [74]. The authors consider the problem of closing up to a pre-specified number of initially open and operating facilities as demand declines over a given multiperiod planning horizon. It is also presented a branch and bound algorithm and near optimal heuristic algorithms to solve the problem. In [75] the model is generalized to solve a facility phase-in/phase-out problem (i.e., opening new facilities or closing initially opened ones). A related model was proposed by Wesolowsky and Truscott [91] that considers the possibility of removing and establishing facilities in each time period and additional restrictions on the maximum number of facilities to be removed in each period. As solution method the authors propose a dynamic programming approach. Roy and Erlenkotter [88] also consider the DUFLP, where new facilities can be opened and initially opened facilities can be closed over the planning horizon. The authors present a branch-and-bound procedure incorporating a heuristic dual ascent method, the latter initially developed by Bilde and Krarup [13] and Erlenkotter [29] for the static UFLP. More recently, Dias et al. [21] present a new version of the DUFLP that not only allows for the opening and closing of facilities over the time horizon but also their reopening, where fixed costs include also reopening costs. A primal-dual heuristic is proposed and computational results are presented. Regarding the capacitated case, reference to Erlenkotter [30] and Jacobsen [39], where not only introductions to such problems are given and additional difficulties that arise in the capacitated case are emphasized, but also earlier models and solution methods are discussed. More recently, models and solution methods for dynamic capacitated problems are suggested by Dias et al. [22] and Soto and Uster[85]. The reader is referred to Dias [23] where an extensive study about dynamic facility location, both in terms of models and solution techniques, is given, and to a recent review given by Nickel and Saldanha da Gama [64], where many other references can be found.

1.2 Uncertainty modelling

Uncertainty has been explicitly incorporated in facility location models in several ways, giving rise to several classes of models and approaches. Uncertainty appears typically in the distribution costs or travel times, production costs, and mainly in the location or level of demand. A common approach to take uncertainty into account is through the design of a set of possible *scenarios*. In general, scenarios can be interpreted as a limited representation of the uncertainty in problem data or uncertainty about how the problem parameters will evolve (Rockafellar and J-B Wets [72], Van der Heijden [87]). Usually a scenario is any possible realization (discrete or interval) of the uncertain problem parameters, and depending on the approach, scenarios may require weights (probabilities) associated to them or not. Another possibility to take uncertainty into account is to consider the uncertain parameters as random variables with an explicit use of their probability distributions or density functions. The corresponding models and related methods can then be considered as belonging to the field of Stochastic Programming (SP) (Birge and Louveaux [14]). It should be noted here that a scenario approach does not exclude the possibility of using some stochastic programming technique. Two–stage stochastic programs with recourse and chance constrained programs, for instance, are two popular stochastic approaches that have been applied to facility location. The latter considers a confidence level type constraint, as two–stage stochastic programs with recourse are characterized by two sets of decisions: the first–stage decisions are the decisions that have to be made before the random events can be observed (*here and now* decisions) and the second–stage or recourse decisions are those that can be decided after the uncertainty has been revealed. Let us detail here only some features of two-stage stochastic problems with recourse, given its relevance within the stochastic location literature and the forthcoming results of this work. The reader is referred to books (Birge and Louveaux [14], Kall and Wallace [42]) and many papers wherein SP approaches are applied.

A standard two–stage stochastic programming problem with recourse, in short 2-SSPP, can be formulated as follows:

$$\text{(2-SSPP)} \qquad \min \quad a^T x + \mathbf{E}[\mathcal{Q}(x,\omega)] \tag{1.2.1}$$

$$\text{s.t. } x \in X,$$

with

$$\mathcal{Q}(x,\omega) = \min \quad g(\omega)^T y \tag{1.2.2}$$

$$\begin{aligned} \text{s.t. } & D(\omega)y = h(\omega) + W(\omega)x, \\ & y \in Y, \end{aligned}$$

where $X \subseteq \mathbb{R}^{n_1}$ denotes the set of constraints on the first stage variables, $a \in \mathbb{R}^{n_1}$, $Y \subseteq \mathbb{R}^{n_2}$ denotes the set of constraints on the second stage variables, ω is a random variable from a probability space $(\Omega, \mathcal{F}, \mathcal{P})$ with $\Omega \subseteq \mathbb{R}^k$, and $(g(\omega), D(\omega), h(\omega), W(\omega))$ are possible (real) uncertain problem parameters that we assume here well dimensioned. The symbol $\mathbf{E}[.]$ represents the mathematical expectation as usual.

The above formulation describes well the nature of two–stage stochastic problems with recourse, noticing however that other forms may be found in the literature. In the first stage problem, the decisions about the values of variables x must be made before the realization of uncertainty. Afterwards, for a given value of the first stage variables x and once the uncertainty is resolved, the values of the second stage or recourse variables y are selected (second stage problem). The objective (1.2.1) is to minimize the cost of the stage one decisions plus the expected cost of the stage two decisions. The above formulation emphasizes also that the second stage problem decomposes into independent subproblems (1.2.2), one for each realization of the uncertain parameters. Although variables y depend on the realization of ω, this is not explicitly represented here because the subproblem for each outcome is decoupled from all others outcomes. Those subproblems, also called recourse problems, are linked by the first stage decisions. Whenever the recourse problems are feasible for (at least) the first stage decisions, the stochastic problem is said to have *(relatively) complete recourse*. In SP the feasibility of the recourse problems is usually enforced by the introduction of artificial recourse variables.
In most applications, usually it is assumed that the random variable ω follows a discrete distribution with finite support $\Omega = \{\omega^1, ..., \omega^S\}$, called the scenario set. Denoting by p^s the probability of realization of the sth scenario ω^s, $P(\omega = \omega^s) = p^s$, and assuming that $p^s > 0$ for all $\omega^s \in \Omega$ and that $\sum_{s=1}^{S} p^s = 1$, it is possible to rewrite the 2–SSPP in an extensive form, the so–called *deterministic equivalent programming problem* of 2–SSPP. In what follows, the uncertain problem parameters $(g(\omega), D(\omega), h(\omega), W(\omega))$ associated with a particular realization ω^s, i.e. with a scenario, is succinctly denoted

by (g^s, D^s, h^s, W^s) with associated probability p^s. Then, the *deterministic equivalent programming problem* of 2–SSPP can be written as follows:

$$\min \quad a^T x + \sum_{s=1}^{S} p^s \mathcal{Q}^s(x) \tag{1.2.3}$$

$$\text{s.t. } x \in X,$$

with

$$\mathcal{Q}^s(x) = \min \quad (g^s)^T y \tag{1.2.4}$$

$$\text{s.t. } D^s y = h^s + W^s x$$
$$y \in Y.$$

As we will see in the next section, two–stage approaches have been applied both in static and dynamic location problems under uncertainty. An important feature in stochastic programming, implicit in the above formulations, is the so–called non–antecipativity principle that, in simple terms, requires that decisions are based only on the information available at the current stage of the decision process and cannot anticipate future outcomes of the uncertain parameters (i.e., Rockafellar and J-B Wets [72], Birge and Louveaux [14]). Multi-stage problems are an extension of two–stage problems in which uncertainty is resolved in more than one stage along the time horizon. More recently, these stochastic programs have also been applied to dynamic problems, which can be even harder to solve than two–stage programs (Dyer and Stougie [28]).

A related issue addressed in the literature is *robustness*, specially when faced with scenario-based models. However, the concept of *robustness* may have different meanings and interpretations, being in reality a multi-faceted issue (Roy [78]). A pioneer work about the use of the robustness concept in strategic management is due to Rosenhead et al. [76]. The criterion robustness is “a measure of the flexibility which an initial decision of a plan maintains for achieving near-optimal states in conditions of uncertainty”. The proposed concept is developed through the case study of a factory location problem over time, and here the robustness concept refers to individual facilities, the ones that should be opened first, when considering a time horizon under uncertainty. Ever since, several different robustness measures have been proposed in the literature, some of which have already been applied to facility location under uncertainty problems. As opposed to sensitivity analysis, that measures the sensitivity of solutions to changes in the input data (it is a reactive approach to tackle uncertainty), robustness should be taken into account *a priori* when the problem is formulated (Mulvey et al. [63], Kouvelis and Yu [47], Roy [78]). For instance, in decision environments with significant uncertainty, rather than the “optimal” solution for a specific scenario or even for the most likely scenario, a risk averse decision maker wants a *robust decision*, defined in this context as the one that performs well across all scenarios and hedges against the worst of all possible scenarios ([47]). Different criteria can then be used to select among robust solutions, such as min-max and min-max regret criteria. In brief, the min-max criterion aims at constructing solutions having the best possible performance in the worst case; regret criterion aims at obtaining a solution minimizing the maximum deviation, over all possible scenarios, between the value of the solution and the optimal value of the corresponding scenario (Aissi et al. [4]). A different robustness approach is given by Mulvey et al. [63]. The authors consider both solution robust and model robust concepts: a solution is robust if it remains close to optimal for any scenario, and it is model robust if it remains almost feasible for any scenario. As it is unlikely that a given solution will remain both feasible and optimal for all scenarios, the authors propose a multicriteria objective approach that allows to measure the tradeoff between solution and model robustness. Usually the above approaches are associated with the so–called Robust Optimization (Snyder [82]). The above and other robustness approaches are also discussed and compared in [19, 8, 12, 82, 83, 4, 11, 78], reflecting the importance of the subject.

1.3 Overview on Single-period and Dynamic facility location problems under uncertainty

One of the earliest stochastic location problems known was presented by Mirchandani and Odoni [61]. The authors extend the concept of p–median to stochastic networks where the distance (travel time) on any arc or the demand (call rate) at any node may be discrete random variables with known distributions. The authors prove that under a set of assumptions an optimal solution exists at nodes of the network (satisfying the Hakimi property, [34]). Thus, the stochastic median location problem can be formulated as an integer linear program (since there is a finite number of identifiable potencial facility sites). Later, Weaver and Church [90] propose two solution procedures for this problem, a heuristic and a bounding procedure based on the subgradient optimization of the Lagrangian dual. Louveaux [54] presents a stochastic version of the UFLP in which demands, variable production and transportation costs, and selling prices (incorporated in the model) can be random. The problem is formulated as a two–stage stochastic program with recourse, where the first–stage decisions are the location and the size (capacity) of the facilities to be established, and the second–stage or recourse decisions are the allocation of the available production to the most profitable demands. As opposed to the deterministic case, the choice of both the demands to be served and the size of the facilities to be established also becomes part of the decision process. In this work also a stochastic version of the p–median, defined as a two–stage stochastic program with recourse, is presented, and relations between the stochastic versions of the p–median and the UFLP are discussed. Solution methods are later presented by Louveaux and Peeters [56]. The authors propose a heuristic dual–based procedure, inspired on the method developed by Erlenkotter [29] for the classical (static and deterministic) UFLP. As the complexity of the problem increases with the randomness in the demands and costs, it is assumed that all random variables have discrete distributions with only a small number of scenarios. Laporte et al. [51] consider a CFLP in which customer demands are stochastic. The problem consists of optimally determining the location and size of facilities given that future customer demand is uncertain. The objective function minimizes the difference between the sum of fixed facility costs and average cost of operating transportation services between facilities and customers (assignment costs), and the expected net revenue from supplying customers. The problem can also be viewed as a two–stage stochastic integer program. Following the scenario approach, Current et al. [17] address location problems in which the total number of facilities to be sited is uncertain. Two decision criteria are considered in p-median based formulations: the minimization of the maximum regret and the minimization of expected opportunity loss. Under the decision criteria, each problem locates an initial number of facilities when the total number is unknown. The approaches are illustrated with a sample problem. Serra and Marianov [80] consider a p–median based model in which travel times between nodes and/or demand at nodes are uncertain, described by scenarios. Two p–median formulations are presented, the min–max and the regret approaches. The authors propose a heuristic method for both formulations, and a real application to the location of fire stations in Barcelona is presented. Snyder and Daskin [83] consider the classical (static) p–median and UFLP problems with uncertain demands and transportation costs, described by probabilistic scenarios. The models minimize expected costs while making sure that the relative regret for each scenario is no greater than a pre-specified value (a new robustness measure for optimization under uncertainty). The relative regret of a solution associated with a given scenario is calculated by the difference between the value of the solution under that scenario and the optimal value of the scenario divided by this latter value. The authors incorporate regret into the problems's formulations by considering constraints that guarantee that the relative regret associated with each solution, for each of the possible future scenarios, is upper bounded. They also propose a Lagrangian decomposition algorithm to solve the corresponding optimization problems. In a recent work (Lim and Sonmez [52]) the same robustness measure is considered in a static facility p–median relocation problem. Berman and Drezner [10] also consider the p–median problem when the total number of facilities to be sited in the future is uncertain. The problem seeks the location for p facilities that minimize the expected weighted distance when up to q new facilities are added to the system in the future. The probability of adding $0 \leq r \leq q$ new facilities (possible scenarios) is given. The authors prove that an optimal solution exists with all the facilities located on nodes (satisfying the Hakimi property), and formulate the problem as an integer program. Heuristic algorithms are suggested to solve the problem (focusing in the case $q = 1$; for $q \geq 2$ it seems more difficult). A similar integer programming model and a decomposition algorithm to solve it is presented by Sonmez and Lim [84]. As opposed to the previous work, in this paper the problem allows the closing of some of the facilities that were opened initially, due to future demand change, and considers also budget restrictions for the opening and closing of facilities. Ravi and Sinha [69] propose a two-stage stochastic version of the UFLP and an *8*–approximation algorithm[1] to solve it. Here, demand and fixed costs are both random, and facilities may be opened in either the first or second stage. A related two-stage stochastic program is proposed by Wang et al. [89] in which service installation costs are also considered (services must be installed at the open facilities and each customer must be assigned to an open facility at which the service requested by the customer is installed). The authors propose a primal-dual approximation algorithm to solve the optimization problem. Lin [53] proposes a stochastic version of the single–source capacitated facility location problem in which the demand is uncertain. The objective function is to minimize the total system costs including fixed facility costs and costs of servicing each demand point by its assigned facility. Simultaneously, recognizing that facilities should provide an adequate level of service, the model also incorporates facility service level requirements. These requirements are formulated as chance constraints, being the probability that each open facility can cope with the stochastic demand assigned. Mo and Harrison [62] propose a conceptual framework for robust supply chain design under demand uncertainty. The aim is to find a supply chain configuration (or a group of configurations) that provides robust performance under demand uncertainty. Uncertainty of demand is represented by discrete scenarios with known probabilities. First the authors define various performance measures of "robustness“ (minimum total expected cost, minimum variance of total cost, minimum of maximum deviation, multiple criteria) emphasizing different perspectives of robust supply chain. As solution methods, the authors discuss explicit enumeration methods and SP methods. In the SP approach the problem is formulated as a classic two-stage stochastic program. The objective function is to minimize total expected cost, which includes fixed costs of opening plants and warehouses, expected shipping cost from plants to warehouses and from warehouses to customers, and expected outsourcing cost when customers' demands cannot be satisfied from warehouses. The authors discuss the difficulties in using these approaches when the total number of scenarios is large and suggest that this number could be reduced by a sampling based approach. Albareda-Sambola et al. [7] consider a two-stage stochastic program for a facility location problem where uncertain demand is modelled by a Bernoulli distribution. Kiya and Davoudpour

[1] An approximation algorithm is a c–approximation algorithm (where c is the approximation ratio) if it can be proven that the solution found by the algorithm is at most c times worse than the optimal solution (in this case, c times larger as it is a minimization problem).

[44] extend the deterministic warehouse network re-design model to uncertain operational parameters (demand and operational costs) described by probability distributions. A two-stage stochastic program with recourse is presented and an algorithm based on the Sample Average Approximation method combined with Benders decomposition and other heuristic methods is developed.

Our attention returns now to those works wherein both uncertainty and time are explicitly considered. Jornsten and Bjorndal [40] consider the DUFLP under uncertainty, where the fixed and variable costs are described via a set of scenarios. To solve the dynamic and stochastic program, the authors use the scenario and policy aggregation described by Rockafellar and J-B Wets [72]. The method is applied to a set of small illustrative problems. Ahmed and Garcia [3] consider a dynamic capacity acquisition and assignment problem under uncertainty. The problem seeks a capacity expansion schedule for a set of resources and the assignment of resource capacity to tasks over the multi-period planning horizon. The problem can be viewed as the planning of locations and capacities of distribution centers (DCs) and the assignment of customers to the DCs. The model explicitly incorporates uncertainty in task processing requirements and assignments costs via a set of scenarios. Although the problem is a multi-period one, the capacity planning decisions for all periods are made in period/stage one (thus, a two-stage stochastic programming approach is adopted). Romauch and Hartl [73] consider a dynamic facility location problem with uncertain demand, described by scenarios. The problem seeks the optimal decisions for production, inventory and transportation, to serve the customers during a fixed number of periods. It is assumed that the production sites have limited storage capacities. The model is first solved by dynamic programming and then a heuristic is proposed, the Sample Average Approximation Method (SSA) adapted to the multi-period case. Albareda-Sambola et al. [5] present a multi-period location-assignment problem under uncertainty. It is a stochastic version of an earlier (deterministic and multi-period) problem studied by the same authors. Here, the service time periods of the customers and the minimum number of customers to be served at each time period are scenario dependent. The objective is to minimize the expected cost-penalty value (setup cost for the open facilities, assignment and service cost, and penalty cost for not servicing customers with demand). More recently, the same authors present in [6] a new algorithm for a multi–period location–assignment problem under uncertainty, a Fix–and–Relax–Coordination scheme. Hernández et al. [36] present a multi-period stochastic model to the location of prison facilities under uncertainty, where the uncertain future demand for capacity is represented by probabilistic scenarios. The problem seeks the location and sizes of a given number of new facilities (jails) and determines where and when to increase the capacity of both new and existing facilities over a time horizon. Subject to several constraints (maximum inmate transfer distances, upper and lower bounds for facility capacities, among others) the objective is to minimize the expected costs of the prison system. The model is solved by a branch–and–cluster coordination scheme (a heuristic mixture of branch–and–fix coordination and branch–and–bound schemes).

We next review some works where examples of facility location problems integrated in supply chain are proposed and where some other related references can be found. Aghezzaf [2] first developed a deterministic capacity planning and warehouse location model for the supply chain (which can be viewed as a multiple–source capacitated economic lot-sizing problem). Then the model is extended to uncertain realizations of future market demand (the only source of uncertainty) described by scenarios. The author uses the concept of robust optimization developed by Mulvey et al. [63] combined with Lagrangean relaxation methods. Pan and Nagi [67] also propose a robust optimization formulation for a multiple layer supply chain network under demand uncertainty. The uncertainty of demand is represented by probabilistic scenarios. The objective function includes expected total cost, cost variability and model infeasibility penalty by the consideration of a weighted penalty to unmet demand that may occur under a possible scenario. The problem includes several decisions: location, distribution, production, inventory. To solve the problem a heuristic is developed and extensive computational results are presented. Pimentel et al. [68] develop a stochastic capacity planning problem applied to a Global Mining Supply Chain which integrates lot sizing, capacity expansions, facility location and network design decisions. Facility location decisions include the opening, closing and reopening of facilities. The authors adopt a multi-stage integer stochastic formulation where the evolution of the uncertain parameters is represented by a discrete probability scenario tree [2]. An analysis of different solution approaches, from exact to approximate methods, with solutions provided by software CPLEX is given. Nickel et al. [65] propose a multi–period multi–commodity stochastic supply chain network design problem which integrates, in addition to location and distribution decisions, financial decisions such as what investments and loans to consider in each time period of the planning horizon. Uncertainty is associated with future demand and return rates, represented by a set of scenarios. Service level and risk measures are also included in the model, both in the objective function. The problem is formulated as a multi–stage stochastic mixed–integer linear programming problem. Due to computational reasons, a more compact formulation of the problem is proposed which is based upon the paths in the scenario tree. In order to measure the relevance of using a stochastic approach (the value of the stochastic programming approach), a deterministic problem derived from the stochastic one is presented. Computational results including comparisons between the stochastic and the deterministic solutions are presented.

[2] The nodes in period t constitute the states of the world that can be distinguished from the information available up to t; the leaf nodes define the scenarios, which represent the joint realizations of the risky parameters over all periods.

Chapter 2

Mathematical Models

The models proposed in this work can be applied to any situation in which a company has to do the planning of strategic location investments over a given period of time. As emphasized earlier, the motivation to study location models which explicitly incorporate uncertainty comes from the need to take into account in the decision process the environmental changes that may occur during the planning horizon. The main sources of uncertainty considered in the models developed come from the existence or lack of customers, as well as costs associated with the opening of facilities and satisfying the clients' demand. Costs for opening facilities can change due to the economic environment, behavior of the real estate market, changes in interest rates. Such costs can even hinder the opening of a facility. Assignment costs can change due to changes in road infrastructures, new roads can be built while others may become inaccessible, government policies, price of fuel, tolls, for instance.

We have witnessed that the representation of uncertainty in optimization models, applied also to location models, has been widely debated in the literature (e.g., Dembo [19], Mulvey et al. [63], Van der Heijden [87], Kouvelis and Yu [47], Snyder [82], Durbach and Stewart [27]). The scenario approach appears as "an extremely powerful, convenient and natural way to represent uncertainty" ([19]) and can be more appropriate than a stochastic one, especially when the available information may not be sufficient to support a stochastic programming model (Rockafellar and J-B Wets [72], Van der Heijden [87]). Under high uncertain conditions, such as those that may occur during a multi-period location problem, the design of scenarios can be more accurate than the use of probability distributions or stochastic process (Schoemaker [79], Van der Heijden [87]). A recent experimental study by Durbach and Stewart [27], about the effect of uncertainty representation on decision making in terms of several items (the difficulty experienced in making a decision, for instance), indicates that the use of probability distributions appeared to overload subjects, being more difficult to use than other concise formats such as the use of scenarios.

We have chosen to represent uncertainty in the models by a finite and discrete set of possible scenarios. The study cited above only reinforces our choice on the scenario approach, dealing with dynamic location problems under uncertainty that are by themselves harder to be understood by Decision Makers. Scenarios are interpreted as "a thinking tool and communication device that aid the managerial mind rather than replace it", an aid especially useful under conditions of high uncertainty and complexity (Schoemaker [79]). In some of the models presented, we also consider probabilistic scenarios and thus we do not exclude here the use of stochastic approaches. In particular, two–stage stochastic problems (briefly reviewed in section 1.2) that model well the real nature of location problems, though the probabilities associated with the scenarios must also be advanced. Several other questions (and difficulties) may arise whenever the uncertainty is explicitly incorporated into a model. For instance, it might be difficult to find a single solution defined as the *best* one in all possible future realizations of uncertainty. Within this context, the concept of *best* solution strongly depends on the attitude towards risk of the Decision Maker (DM). When the DM is assumed to be risk neutral, expected cost criterion are appropriated but, as already noted in section 1.2, in the presence of different risk profiles other features should be included in the mathematical models in order to generate other possible solutions.

This chapter is dedicated to the description of the problems, mathematical modelling, where integer and mixed-integer linear programming models are presented. We start in section 2.1 with an extension of the dynamic uncapacitated facility location problem to an uncertain future (Marques and Dias [58]). Later on, in section 2.2 a *regret* based measure of robustness is included in this model. This measure is not new in the location literature, but is explicitly incorporated in a dynamic location problem for the first time (as far as the authors know) (Marques and Dias [59]). By the analysis of some illustrative examples, it is possible to obtain a deeper knowledge about the problem and its possible solutions: the possibility of achieving more robust solutions from small changes in a given and less robust solution, or the discovery of the core facilities, those that remain open even if the robustness parameter varies. In section 2.3 the dynamic uncapacitated location problem under uncertainty is considered as a multi-objective problem, where each scenario will give rise to one objective (Dias and Marques [24]). Within this context, the aim is to achieve Pareto–efficient solutions. A single objective location problem under uncertainty is tackled by resorting to a multi–objective approach, and the concept of Pareto–efficiency is thus applied in the context of a single objective problem under uncertainty. It is quite difficult to find the concept of Pareto efficiency being applied in this context. We have found several publications dedicated to multi–objective stochastic programming, usually tackling the problem by reducing it to a single objective stochastic program or transforming it to a deterministic multi-objective program (e.g., Hulsurkar et al. [37], Teghem Jr et al. [41], Urli and Nadeau [86], Abdelaziz [1], Gutjahr [33], Guillén et al. [32],Cardona–Valdés et al. [15]). Additional references goes to the recent works proposed by Lamboray and Vanderpooten [50], Iancu and Trichakis [38], and Klamroth et al. [45] wherein multiple objective (deterministic) counterparts for uncertain optimization problems are introduced and their relations to well known scalar robust optimization problems are discussed.

In all the models proposed so far, as we assumed that facilities are uncapacitated, for the first–stage location decisions taken, it is certain that total demand will be satisfied in the second–stage (whatever the scenario that will occur). In section 2.4 we address capacitated problems, following mono and multi–objective approaches to tackle these challenging problems.

2.1 Dynamic uncapacitated location problem under uncertainty

In this section the dynamic uncapacitated facility location problem is extended to uncertain realizations of the potential locations for facilities and the existence of customers as well as fixed and variable costs. The *future* will be one of a finite set of possibilities, represented by *scenarios* where each scenario characterizes the value of all problem's parameters in a possible future.

The first decisions to be made are where and when to locate the facilities. We assume here that once a facility is opened, it stays open until the end of the planning horizon. Afterwards, it must be decided how to assign the *existing* customers over the whole planning horizon under each possible scenario. We are indeed in the presence of a two–stage decision problem: location decisions are strategic by nature so they must be decided *here and now* and must be valid for all possible future scenarios, whilst assignment decisions can be decided after the uncertainty has been resolved and thus can be adjusted in each time period to each possible scenario. The aim of the problem is to find a *good* solution that performs well across all possible scenarios without focusing in a particular scenario. More precisely, the objective is to find a solution that minimizes the expected total cost (fixed plus assignment costs) over all possible scenarios. A mixed linear programming formulation for this problem is proposed. Let us introduce the notation that will be used throughout this text.

The time horizon is represented by a finite set of discrete time periods $\mathcal{T} = \{1, ..., t, ..., T\}$. The set of possible future *scenarios* is denoted by $\mathcal{S} = \{1, ..., s, ..., S\}$. In what follows, suppose that each scenario $s \in \mathcal{S}$ will occur with probability p^s such that $p^s > 0$ and $\sum_{s \in \mathcal{S}} p^s = 1$.

The set of potential facility sites is denoted by $J = \{1, ..., j, ..., M\}$ and the set of possible customer locations (or demand points) by $I = \{1, ..., i, ..., N\}$. These sets include all the potential facility locations and all the potential customers for all possible scenarios, despite the fact that for each scenario in particular possibly only a subset of potential locations and a subset of customers is considered. Let us define δ_{it}^s as equal to 1 if customer i has a demand that has to be fulfilled during period t for scenario s (in short, an existing customer), and 0 otherwise. Then we have to guarantee that all customers such that $\delta_{it}^s = 1$ are assigned to an open facility, for all $(t, s) \in \mathcal{T} \times \mathcal{S}$.

In terms of costs, the model considers not only fixed costs (opening and operating), but also variable costs associated with the assignment of customers to the facilities. For $(j, t, s) \in J \times \mathcal{T} \times \mathcal{S}$, let f_{jt}^s be the fixed cost of establishing (opening) facility j at the beginning of period t plus the operating costs in all subsequent time periods, under scenario s; for $(i, j, t, s) \in I \times J \times \mathcal{T} \times \mathcal{S}$, c_{ijt}^s represents the assignment cost of customer i to facility j in period t and under scenario s. If it is not possible to open facility j at the beginning of time period t under scenario s, then the corresponding fixed cost will be considered equal to $+\infty$. Such a situation can only occur for $t > 1$, given the possibility that any new service opens in that period.

The decisions to be made are where and when to locate new facilities, and how to assign the existing customers over the whole planning horizon under each possible scenario. Let $x \in \{0, 1\}^{|J| \times |\mathcal{T}|}$ be the vector of location decisions such that x_{jt} equals 1 if facility j is opened at the beginning of period t, and 0 otherwise, and $y \in \{0, 1\}^{|I| \times |J| \times |\mathcal{T}| \times |\mathcal{S}|}$ the vector of assignment decisions such that y_{ijt}^s equals 1 if customer i is assigned to facility j in period t under scenario s, and 0 otherwise (we could also consider, for each $s \in \mathcal{S}$, vector $y^s \in \{0, 1\}^{|I| \times |J| \times |\mathcal{T}|}$, being the vector of assignment decisions for scenario s only). The objective is to minimize expected total cost including fixed and assignment costs over all scenarios.

The dynamic uncapacitated facility location problem under uncertainty, in short DUFLPU, can be formulated in an extensive form as follows:

$$\text{(DUFLPU)} \qquad \min \quad \sum_{t\in\mathcal{T}}\sum_{j\in J}\sum_{s\in\mathcal{S}} p^s f_{jt}^s\, x_{jt} + \sum_{s\in\mathcal{S}}\sum_{t\in\mathcal{T}}\sum_{i\in I}\sum_{j\in J} p^s c_{ijt}^s\, y_{ijt}^s \tag{2.1.1}$$

s.t.

$$\sum_{j\in J} y_{ijt}^s = \delta_{it}^s \qquad \forall i \in I, t \in \mathcal{T}, s \in \mathcal{S}, \tag{2.1.2}$$

$$\sum_{\tau=1}^{t} x_{j\tau} - y_{ijt}^s \geq 0 \qquad \forall i \in I, j \in J, t \in \mathcal{T}, s \in \mathcal{S}, \tag{2.1.3}$$

$$\sum_{t\in\mathcal{T}} (-x_{jt}) \geq -1 \qquad \forall j \in J, \tag{2.1.4}$$

$$x \in \{0,1\}^{|J|\times|\mathcal{T}|}, \tag{2.1.5}$$

$$y \in \{0,1\}^{|I|\times|J|\times|\mathcal{T}|\times|\mathcal{S}|}. \tag{2.1.6}$$

The objective function (2.1.1) minimizes the expected total costs (fixed plus variable costs). Constraints (2.1.2) require that in every time period under each scenario an existing customer is assigned to exactly one facility. Constraints (2.1.3) impose that an existing customer can only be assigned to open facilities. A customer can be assigned to different facilities at different time periods and different scenarios. Constraints (2.1.4) ensure that each facility is opened at most once during the time horizon (located at the same site in all scenarios). Finally, (2.1.5)–(2.1.6) restrict the decision variables to be binary.
The above formulation contains the UFLP ($|\mathcal{T}| = |\mathcal{S}| = 1$) and the DUFLP ($|\mathcal{T}| > 1, |\mathcal{S}| = 1$) as particular problems, and has $|J|\,|\mathcal{T}| + |J|\,|I|\,|\mathcal{T}|\,|\mathcal{S}|$ binary variables and $|I|\,|\mathcal{T}|\,|\mathcal{S}| + |J|\,|I|\,|\mathcal{T}|\,|\mathcal{S}| + |J|$ restrictions (not counting the zero-one constraints). Even for moderate dimensions of these sets, (2.1.1)–(2.1.6) becomes a quite large integer linear program.

Remark 2.1.1 *The DUFLPU is a two–stage stochastic model though a standard formulation has not been explicitly written here. In spite of the location decisions being scenario independent, in the sense that they cannot be changed according to each scenario in particular, the fixed cost can be considered scenario dependent as it was assumed here. Note that if we consider $f_{jt} = \sum_{s\in\mathcal{S}} p^s f_{jt}^s$, the objective function (2.1.1) can be rewritten as follows:*

$$\sum_{t\in\mathcal{T}}\sum_{j\in J} f_{jt}\, x_{jt} + \sum_{s\in\mathcal{S}}\sum_{t\in\mathcal{T}}\sum_{i\in I}\sum_{j\in J} p^s c_{ijt}^s\, y_{ijt}^s. \tag{2.1.7}$$

The model can now be explicitly written as a two–stage program wherein the fixed costs on the first stage are in fact expected fixed costs. Throughout this text we will consider *mainly the form (2.1.1), but it should be stressed once more that location decisions make the DUFLPU non–separable by scenarios as those decisions must be valid for all scenarios.*

The first technique developed to solve the DUFLPU is a primal-dual heuristic approach. In order to apply this heuristic, we present next the dual problem, the condensed dual problem and the complementary slackness conditions between the dual and primal problems. The forthcoming formulations are crucial for the algorithm's description which is only detailed in section 3.1 for the interested readers.

2.1.1 Dual problem and complementary slackness conditions

Consider the linear programming (LP) relaxation of the primal problem defined by (2.1.1)–(2.1.4) and where restrictions (2.1.5) and (2.1.6) are replaced by nonnegativity constraints. Defining in (2.1.1) $\mathcal{C}^s_{ijt} = p^s c^s_{ijt}$ and $\mathcal{F}^s_{jt} = p^s f^s_{jt}$, and considering dual variables v^s_{it}, w^s_{ijt} and u_j associated with the restrictions (2.1.2), (2.1.3) and (2.1.4), respectively, the dual problem is given by:

$$\max \quad \sum_{i\in I}\sum_{t\in\mathcal{T}}\sum_{s\in\mathcal{S}} \delta^s_{it}\, v^s_{it} - \sum_{j\in J} u_j \tag{2.1.8}$$

subject to

$$v^s_{it} - w^s_{ijt} \leq \mathcal{C}^s_{ijt} \qquad \forall i\in I, j\in J, t\in\mathcal{T}, s\in\mathcal{S}, \tag{2.1.9}$$

$$\sum_{i\in I}\sum_{s\in\mathcal{S}}\sum_{\tau=t}^{T} w^s_{ij\tau} - u_j \leq \sum_{s\in\mathcal{S}} \mathcal{F}^s_{jt} \qquad \forall j\in J, t\in\mathcal{T}, \tag{2.1.10}$$

$$w^s_{ijt} \geq 0 \qquad \forall i\in I, j\in J, t\in\mathcal{T}, s\in\mathcal{S}, \tag{2.1.11}$$

$$u_j \geq 0 \qquad \forall j\in J. \tag{2.1.12}$$

For feasible variables v^s_{it}, by constraints (2.1.9) and (2.1.11), we may set

$$w^s_{ijt} = \max\{0, v^s_{it} - \mathcal{C}^s_{ijt}\} \qquad \forall i,j,t,s, \tag{2.1.13}$$

to obtain the condensed dual problem:

$$\max \quad \sum_{i\in I}\sum_{t\in\mathcal{T}}\sum_{s\in\mathcal{S}} \delta^s_{it}\, v^s_{it} - \sum_{j\in J} u_j \tag{2.1.14}$$

subject to

$$\sum_{i\in I}\sum_{s\in\mathcal{S}}\sum_{\tau=t}^{T} \max\{0, v^s_{i\tau} - \mathcal{C}^s_{ij\tau}\} - u_j \leq \sum_{s\in\mathcal{S}} \mathcal{F}^s_{jt} \qquad \forall j,t, \tag{2.1.15}$$

$$u_j \geq 0 \qquad \forall j. \tag{2.1.16}$$

The corresponding slack variables π_{jt} for constraints (2.1.15) are given by:

$$\pi_{jt} = \sum_{s \in \mathcal{S}} \mathcal{F}^s_{jt} - \sum_{i \in I} \sum_{s \in \mathcal{S}} \sum_{\tau = t}^{T} \max\{0, v^s_{i\tau} - \mathcal{C}^s_{ij\tau}\} + u_j \qquad \forall j, t. \tag{2.1.17}$$

Then, the complementary slackness conditions are:

$$\pi_{jt}\, x_{jt} = 0 \qquad \forall j, t, \tag{2.1.18}$$

$$v^s_{it} \left(\sum_j y^s_{ijt} - \delta^s_{it} \right) = 0 \qquad \forall i, t, s, \tag{2.1.19}$$

$$w^s_{ijt} \left(\sum_{\tau=1}^{t} x_{j\tau} - y^s_{ijt} \right) = 0 \qquad \forall i, j, t, s, \tag{2.1.20}$$

$$u_j \left(1 - \sum_t x_{jt} \right) = 0 \qquad \forall j, \tag{2.1.21}$$

$$y^s_{ijt} \left(v^s_{it} - \mathcal{C}^s_{ijt} - w^s_{ijt} \right) = 0 \qquad \forall i, j, t, s. \tag{2.1.22}$$

As it is well known from duality theory, if the dual and primal solutions satisfy all complementary slackness conditions, then the solutions are optimal. If not, the corresponding primal solution is said to have gap.

2.1.2 Extensions regarding the uncertainty in potential facility sites

It was assumed for the DUFLPU that if it is not possible to open facility j at the beginning of time period t under scenario s, then the corresponding fixed cost is considered equal to $+\infty$. The fixed cost incurred under that scenario will be too high, and given the problem's objective function (2.1.1), the corresponding facility location certainly will not be selected to the set of open facilities in that period of time. Consequently, this assumption will only decrease the number of potential facility sites in that period of time.

Let us assume now that, even if it is not possible to open facility j at the beginning of time period t under scenario s, it is still possible to open that facility under other scenario(s) $s' \neq s$ for $s' \in \mathcal{S}$. In addition, the fixed cost can be equal to any value $< +\infty$, i.e., it is possible to attribute a finite fixed cost to the possibility of not opening that service in the future. This cost may be null (if the facility will not be opened there will be no fixed cost) or any positive value (representing costs no longer recoverable for instance).

In order to model this new and more realistic situation, let us assume that, for $(j,t,s) \in J \times \mathcal{T} \times \mathcal{S}$, the fixed cost f_{jt}^s will be equal to any value in $\mathbb{R}_0^+$. In addition, let us define, for $(j,t,s) \in J \times \mathcal{T} \times \mathcal{S}$, parameter ρ_{jt}^s as equal to 1 if it is possible to open facility j at the beginning of time period t under scenario s, and 0 otherwise. As opposite to the first model, in the present situation, even if $\rho_{jt}^s = 0$, facility location j remains as a potential facility site to open in period t, if and only if there is at least one $s' \neq s$ with $\rho_{jt}^{s'} = 1$. However, if $x_{jt} = 1$ (facility j is opened at the beginning of period t) and $\rho_{jt}^s = 0$ for some scenario s, no assignments can be made to that facility for all $\tau \geq t$ under that scenario s, even if $\rho_{j\tau}^s = 1$ for some $\tau > t$ as the facility is opened once and the important ρ is on that period when the facility is planned to be opened. Customers will not be able to use that facility under that scenario(s) and so assignments should not be made to that facility. In terms of decision variables, the definitions introduced earlier are still valid here, though to a decision $x_{jt} = 1$ should be also added the information about ρ_{jt}^s for all $s \in \mathcal{S}$. In terms of problem formulations and solution approach, small changes have to be introduced in the results already developed for the first problem.

The primal problem formulation is given by the primal problem (2.1.1)–(2.1.6) with constraints (2.1.3) replaced by

$$\sum_{\tau=1}^{t} \rho_{j\tau}^s x_{j\tau} - y_{ijt}^s \geq 0 \qquad \forall i \in I, j \in J, t \in \mathcal{T}, s \in \mathcal{S}. \tag{2.1.23}$$

The above constraints still impose that an existing customer can only be assigned to open facilities. However, in the present model, each customer i in period t under scenario s can only be assigned to a facility opened in τ and such that $\rho_{j\tau}^s = 1$, for $\tau \leq t$.

In terms of dual problem formulation, consider (2.1.8)–(2.1.12) where constraints (2.1.10) are replaced by

$$\sum_{i \in I} \sum_{s \in \mathcal{S}} \sum_{\tau=t}^{T} \rho_{j\tau}^s w_{ij\tau}^s - u_j \leq \sum_{s \in \mathcal{S}} \mathcal{F}_{jt}^s \qquad \forall j \in J, t \in \mathcal{T}. \tag{2.1.24}$$

Consequently, the condensed dual problem is given by (2.1.14)–(2.1.16) with constraints (2.1.15) replaced by

$$\sum_{i \in I} \sum_{s \in \mathcal{S}} \sum_{\tau=t}^{T} \rho_{j\tau}^s \max\{0, v_{i\tau}^s - \mathcal{C}_{ij\tau}^s\} - u_j \leq \sum_{s \in \mathcal{S}} \mathcal{F}_{jt}^s \qquad \forall j, t. \tag{2.1.25}$$

The corresponding slack variables π_{jt} for constraints (2.1.25) are given by:

$$\pi_{jt} = \sum_{s\in\mathcal{S}} \mathcal{F}^s_{jt} - \sum_{i\in I}\sum_{s\in\mathcal{S}}\sum_{\tau=t}^{T} \rho^s_{j\tau} \max\{0, v^s_{i\tau} - \mathcal{C}^s_{ij\tau}\} + u_j \qquad \forall j,t. \tag{2.1.26}$$

Finally, the complementary slackness conditions are given by (2.1.18)–(2.1.22) where conditions (2.1.20) are replaced by:

$$w^s_{ijt} \left(\sum_{\tau=1}^{t} \rho^s_{j\tau} x_{j\tau} - y^s_{ijt} \right) = 0 \qquad \forall i,j,t,s. \tag{2.1.27}$$

In spite of this model being more general than the DUFLPU first introduced, it requires not only more (input) parameters but also additional information must be given whenever location decisions are taken. Mainly due to simplicity reasons, throughout this work we will assume only the first situation described for the DUFLPU, hopping that this decision will contribute to an easier reading of this text.

2.2 Dynamic uncapacitated location problem under uncertainty with a regret based measure of robustness

We propose now a variation of the DUFLPU where a *regret* based measure of robustness is incorporated. The aim of this problem is still to find a *good* solution that performs well across all possible scenarios, through the minimization of the expected total cost over all possible scenarios, but the provided solution, if exists, is subject to additional constraints being a more robust solution in a context of uncertainty. The concept of regret is well known in the literature and has been used mainly in static scenario–based location models (e.g., Snyder [82], Snyder and Daskin [83], Lim and Sonmez [52]). In simple terms, taking into account that a decision has to be made considering several different scenarios, regret can be understood as a measure of how much will we lose due to the fact that the optimal solution of the scenario that came to occur was not implemented.

In order to formulate and describe the problem, let us first introduce additional notation as well as some important definitions that were adapted from the static case. For a given solution (x, y) and for each $s \in \mathcal{S}$, let us represent the total cost achieved under scenario s by $\zeta_s(x, y)$:

$$\zeta_s(x,y) = \sum_{t\in\mathcal{T}}\sum_{j\in J} f^s_{jt}\, x_{jt} + \sum_{t\in\mathcal{T}}\sum_{i\in I}\sum_{j\in J} c^s_{ijt}\, y^s_{ijt}. \tag{2.2.1}$$

As already noted, location decisions to the DUFLPU must be valid for all scenarios. Consider now each single–scenario minimization problem wherein the objective function is to minimize the total cost for a given scenario only. We are faced with $|\mathcal{S}|$ deterministic dynamic uncapacitated facility location problems (DUFLP), each corresponding to one single scenario. Throughout this text we will refer to each single–scenario minimization problem as DUFLP^s and represent its optimal objective function value by ζ_s^*. Let us assume that ζ_s^* is known and such that $\zeta_s^* > 0$, for all $s \in \mathcal{S}$.

Taking into account that we are faced with different possible scenarios (data change for different scenarios), the best solution of each DUFLP^s is expected to be different not only from the best ones achieved under other scenarios but from the best of the DUFLPU as well. In what follows, we are only interested in feasible solutions of the DUFLPU that are also feasible to DUFLP^s for all $s \in \mathcal{S}$. In the present case, this will always happen since we are dealing with an uncapacitated problem.

Definition 2.2.1 *The Regret of a feasible solution (x, y) of the DUFLPU associated with a given scenario $s \in \mathcal{S}$ is defined by the difference between the value of the solution under that scenario and the optimal value of that scenario:*

$$Reg^s(x, y) = \zeta_s(x, y) - \zeta_s^*. \tag{2.2.2}$$

The relative regret is given by $Reg^s(x, y)/\zeta_s^*$.

Throughout this text we will use the terms regret and relative regret interchangeably.

The aim is to minimize the expected total cost ensuring that the relative regret for each scenario does not exceed a pre-specified value α, $\alpha \geq 0$. Thus, for a given $\alpha \geq 0$, the dynamic uncapacitated location problem under uncertainty with a regret based measure of robustness, in short α-DUFLPU, is formulated by (2.1.1)–(2.1.6) and the following constraints:

$$\zeta_s(x, y) \leq (1 + \alpha)\zeta_s^* \qquad \forall s \in \mathcal{S}. \tag{2.2.3}$$

Constraints (2.2.3) impose that relative regret for each scenario is no greater than α. A solution for the problem α-DUFLPU is such that the objective function value under any scenario is at most $100\,\alpha\%$ worse than the scenario's optimal solution. Thus, and depending on the α value, a more demanding and robust solution is expected to be found for this problem than the solution to the DUFLPU, that can be seen as a ∞-DUFLPU. We will call throughout this text a feasible solution of the α-DUFLPU an α-robust solution.

Definition 2.2.2 *For a given $\alpha \geq 0$, a feasible solution of α-DUFLPU is called an α-robust solution.*

The approach developed to obtain α-robust solutions is described in section 3.3.

2.2.1 Expected total cost versus regret: illustrative examples

The effect of incorporating parameter α into the proposed dynamic location problem under uncertainty is now illustrated. The tradeoff between the expected total cost and α is also analysed. It is worthwhile to study the compromise that exists between expected total cost and maximum regret as the DM will be able to make a more informed decision, choosing the solution that is most fitted to his attitude towards risk.
Considering three randomly generated problem instances, problem α-DUFLPU has been solved iteratively for several values of α, and the best feasible solution found in each iteration was recorded. Initially, α was set to a large value and then it was reduced by 0.01 units at each iteration until no feasible solution could be found.

Example 2.2.1 *Consider an instance with* 10 *time periods,* 20 *potential facility sites,* 100 *possible customers and* 5 *scenarios.*

For this particular instance, it was possible to prove that α-DUFLPU is infeasible for $\alpha < 0.07$. The best expected total costs achieved for each α are plotted in Figure 2.2.1. We can see that the expected total cost has a non decreasing pattern as α decreases. In addition, the steep curve indicates that large reductions in regret are possible with small increases in expected total cost. These results are in accordance with similar results already observed in static models. Achieving a more robust solution can sometimes be accomplished by small changes in a given solution. This is depicted in Figure 2.2.2, where two situations are compared: considering a maximum relative regret of 19% and 7%. For this particular example, we can see that small changes in location decisions can lead to more robust solutions.

Figure 2.2.1: Example 2.2.1:Expected total cost *versus* α.

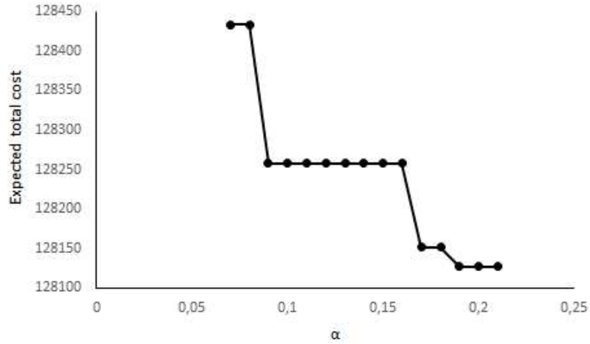


Table 2.2.1: Example 2.2.1: Expected total cost *versus* α.

α	Best Obj	Increase	Location Decisions				
			t=1	t=2	t=3	t=4	t=6
0.19	128127	0.0%	9;11;13;14;17	10;18	7	4	2
0.17	128151,2	0.02%	9;11;13;14;17	10;18	7	4	–
0.09	128257,8	0.09%	6;9;11;13;14;17	18	–	4;16	–
0.07	128433,4	0.24%	9;11;13;14;17	18	–	4;16	2

Table 2.2.1 depicts the solutions in detail. We report the best objective function values found for some values of α as well as the corresponding location decisions. In column 'Increase' we report the increase (in percentage) of the best objective function values relative to the best one achieved with $\alpha = 0.19$, given by the diference between the best objective function value for each α and the best one with $\alpha = 0.19$ divided by this latter value. We can see that it is possible to decrease the relative regret from 19% to only 7% with a slightly increase of 0.24% in the expected objective function value (illustrated in Figure 2.2.2). Furthermore, we can gather additional information about this particular problem, such as the discovery of a set of 'core' facilities, the ones that stay open for all values considered for parameter α.

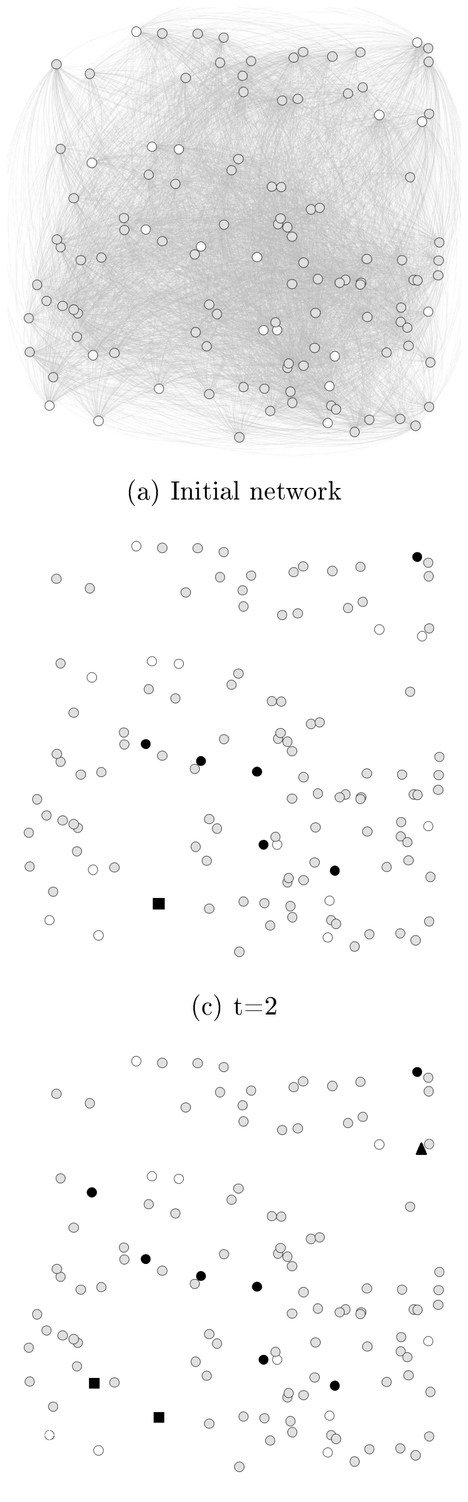
Example 2.2.2 *Consider two instances of the same size:* 10 *periods of time,* 20 *potential facility sites,* 100 *possible customers and* 10 *scenarios.*

The first instance proved to be infeasible for $\alpha < 0.06$ and the second one for $\alpha < 0.17$. The best solutions achieved for both problem instances, presented in Figure 2.2.3 and Table 2.2.2, show a similar behavior to the one observed in example 2.2.1. It is also possible to identify for both instances the corresponding set of core facilites.

Table 2.2.2: Example 2.2.2: Expected total cost *versus* α.

	α	Best Obj	Increase	Location Decisions					
				t=1	t=2	t=3	t=4	t=5	t=6
Inst 1	0.19	118189.8	0.00%	5;7;8;14	4;12;16	18	–	–	–
	0.18	118580.0	0.33%	5;7;8;14	12;16	18	–	–	–
	0.1	118614.8	0.36%	5;7;8;14;20	12;16	18	–	–	–
	0.06	118757.5	0.48%	5;7;8;14;18	12;16	–	–	–	–
Inst 2	0.22	106920.6	0.00%	6;7;10	–	–	–	5	17
	0.21	107088.5	0.16%	6;7;10	–	17	–	5	–
	0.2	108047.1	1.05%	6;10	–	17	–	5	–
	0.18	108251.6	1.24%	6;10	–	17	8	5	–
	0.17	108339.1	1.33%	6;10	–	17;20	–	5	–

Figure 2.2.2: Example 2.2.1: Best location decisions for $\alpha = 0.19$ and $\alpha = 0.07$.



(a) Initial network

(b) t=1

(c) t=2

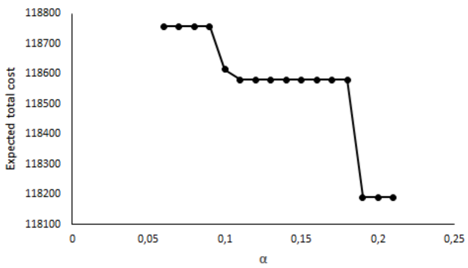
(d) t=3

(e) t=4

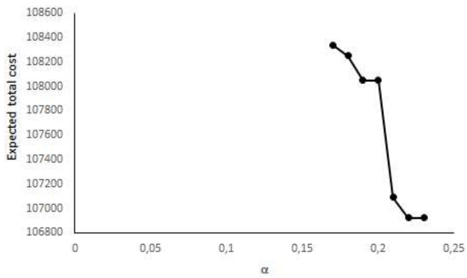
(f) t=6

(a) Initial network. White nodes represent potential facility sites and gray nodes possible customers. (b) – (f) Networks with best location decisions. (●) represent facilities opened both for $\alpha = 0.19$ and $\alpha = 0.07$. (■) represent facilities opened only for $\alpha = 0.19$. (▲) represent facilities opened only for $\alpha = 0.07$.

Figure 2.2.3: Example 2.2.2: Expected total cost *versus* α.



(a) Instance 1



(b) Instance 2

The three instances used here for illustration purposes depict the general behavior observed in similar problems. It is also possible to see that each problem has its own features, and there can be huge variations in the obtained results (namely regarding the minimum relative regret value for which the problem is still feasible) even for problems of the same dimension.

2.3 Multi–objective dynamic uncapacitated location problem under uncertainty

Let us assume that it is not possible to consider *a priori* any kind of assumptions regarding the risk profile of the DM or even about his preferences. Then one possible approach is to consider the dynamic facility location problem under uncertainty as a multi–objective problem where each scenario will give rise to one objective. Thus, a set of objective functions is defined instead of one single objective function and a set of solutions is calculated instead of only one. Within this context, the DM will have a much broader view of the compromises that exist among the possible scenarios.
Recalling that the definition of $\zeta_s(x, y)$ is (2.2.1), the multi–objective dynamic uncapacitated facility location problem under uncertainty, in short MODUFLPU, is defined as follows:

$$\text{(MODUFLPU)} \qquad \min \quad \{\zeta_1(x, y), ..., \zeta_s(x, y), ..., \zeta_S(x, y)\} \tag{2.3.1}$$

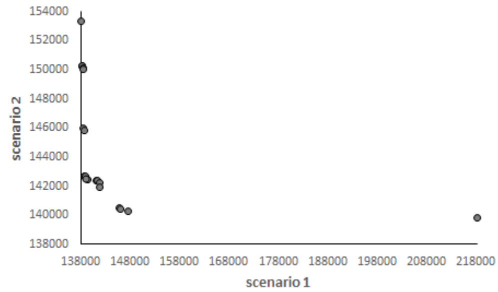
$$\text{s.t.} \quad (2.1.2)\text{–}(2.1.6).$$

In a multi-objective problem, the solutions of interest are designated Pareto–efficient/non–dominated solutions. In the present problem, non-dominated solutions will be the ones such that it is not possible to improve the objective function of one given scenario without deteriorating the objective function of at least one other scenario (definition 2.3.1).

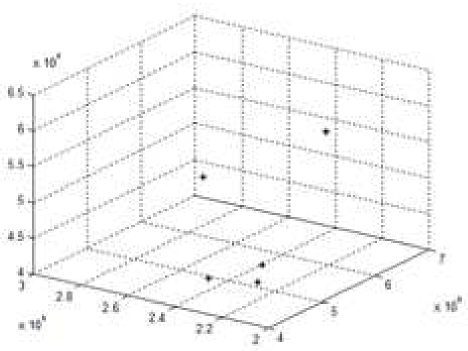
Definition 2.3.1 *Let (x, y) be an admissible solution for MODUFLPU. (x, y) is a Pareto–efficient solution if and only if there is no other solution (x_1, y_1) such that $\zeta_s(x_1, y_1) \leq \zeta_s(x, y)$ for all $s \in \mathcal{S}$ and $\zeta_s(x_1, y_1) < \zeta_s(x, y)$ for at least one scenario s. The image of an efficient solution in the objective space is called a non–dominated solution.*

Regardless the preferences or profile of the DM, assuming only his rationality, the interest goes to Pareto–efficient solutions only. The procedure followed in this work to generate non–dominated solutions to the MODUFLPU is only described and illustrated in section 3.4. In the rest of this section our attention is restricted to results in which the approach was designed and to establish relations with other problems well known from the literature.

Figure 2.3.1: Sets of non–dominated solutions.



(a) Instance with two scenarios



(b) Instance with three scenarios

The non–dominated solutions of a multi–objective problem can be achieved by solving auxiliary programming problems. When dealing with integer or mixed-integer problems, care has to be taken though to guarantee that the chosen procedure is capable of calculating non–supported non–dominated solutions (lying inside duality gaps). In this work we resort to a result due to Ross and Soland [77], where an auxiliary mono–objective programming problem is considered, the well known optimization of a weighted sum of the objective functions. The solutions to the original problem MODUFLPU are then achieved by solving the auxiliary problem that is defined next.

Let $\nu \in \mathbb{R}^S$ be a vector where each component ν_s represents the weight associated with each objective function ζ_s of MODUFLPU, such that $\nu_s > 0$ for all $s \in \mathcal{S}$ and $\sum_{s \in \mathcal{S}} \nu_s = 1$. In addition, let $\mathcal{M} \in \mathbb{R}^{\mathcal{S}}$ be a vector with components $\mathcal{M}_s$ being upper bounds to the objective function (total cost) achieved in each scenario s. It should be stressed here that those weights ν do not represent any kind of DM's preferences. Those weights can and should be changed in accordance with $\mathcal{M}$ for instance (further details about this issue are given in section 3.4).

The auxiliary programming problem to the MODUFLPU, in short AUX, is formulated as follows:

$$\text{(AUX)} \qquad \min \quad \sum_{s\in\mathcal{S}} \nu_s \zeta_s(x,y) \tag{2.3.2}$$

s.t.

$$(2.1.2)\text{–}(2.1.6)$$

$$\zeta_s(x,y) \le \mathcal{M}_s \qquad \forall s \in \mathcal{S}. \tag{2.3.3}$$

The next result, based in Ross and Soland [77], is particularly important in what concerns the calculus of non–dominated solutions to MODUFLPU. Afterwards, some results related with well known problems from the literature are given.

Proposition 2.3.1 *For any $\nu \in \mathbb{R}^S$ such that $\nu_s > 0$ for all $s \in \mathcal{S}$ and $\sum_{s\in\mathcal{S}} \nu_s = 1$, (x, y) is an efficient solution of MODUFLPU if and only if it is the optimal solution of AUX for some $\mathcal{M} \in \mathbb{R}^S$.*

Proposition 2.3.2 *The optimal solution of DUFLPU, the minimum expected total cost over all scenarios, is a non–dominated solution of MODUFLPU.*

Proof: Considering in AUX, for all $s \in \mathcal{S}$, $\nu_s = p^s$ and $\mathcal{M}_s$ large enough (constraints (2.3.3) are redundant), the optimal solution of AUX is the minimum expected total cost. From proposition 2.3.1 we can conclude that this solution is a non–dominated solution of MODUFLPU.

Proposition 2.3.3 *An α-robust solution is a non–dominated solution of MODUFLPU.*

Proof: Considering in AUX, for all $s \in \mathcal{S}$, $\mathcal{M}_s = (1+\alpha)\zeta_s^*$, the optimal solution of AUX is α-robust as AUX has an α-DUFLPU form. In addition, from proposition 2.3.1, the solution is a non-dominated solution to MODUFLPU.

AUX can also be used to calculate an efficient min-max solution. In a first stage, it is necessary to solve the problem of minimizing the maximum cost under all scenarios. This can be done by solving the following programming problem:

$$\text{(MIN-MAX)} \qquad \min \quad \varrho \tag{2.3.4}$$

s.t.

$$(2.1.2)\text{–}(2.1.6)$$

$$\zeta_s(x,y) \le \varrho \qquad \forall s \in \mathcal{S}. \tag{2.3.5}$$

Let ϱ^* be the optimal objective function value of MIN-MAX.

Proposition 2.3.4 *If in constraints (2.3.3) $\mathcal{M}_s$ is defined such that $\mathcal{M}_s = \varrho^*$ for all $s \in \mathcal{S}$, then AUX will generate an efficient min–max solution.*

Proof: Taking into account that ϱ^* is the optimum of MIN-MAX (the objective function value for any scenario s will be less than or equal to this value), it is easy to see that if in constraints (2.3.3) $\mathcal{M}_s$ is defined such that $\mathcal{M}_s = \varrho^*$ for all $s \in \mathcal{S}$, then any efficient solution calculated will also be a min-max solution.

A similar reasoning can be applied in order to obtain an efficient solution that minimizes maximum regret.

Proposition 2.3.5 *Consider problem MIN-MAX with restrictions (2.3.5) replaced by the following set:*

$$Reg^s(x, y) \leq \varrho \qquad \forall s \in \mathcal{S}. \tag{2.3.6}$$

If in AUX $\mathcal{M}_s$ is defined such that $\mathcal{M}_s = \zeta_s^* + \varrho^*$ for all $s \in \mathcal{S}$, then AUX will generate an efficient solution that minimizes maximum regret.

Proof: Taking into account that regret for any scenario is no greater than ϱ^*, it is easy to see that if in constraints (2.3.3) $\mathcal{M}_s$ is defined such that $\mathcal{M}_s = \zeta_s^* + \varrho^*$ for all $s \in \mathcal{S}$, then any efficient solution calculated minimizes maximum regret.

2.4 Dynamic capacitated location problems under uncertainty

The simultaneous consideration of different possible scenarios and capacities associated with facilities brings up other interesting questions and additional difficulties arise. This section is devoted to the modelling of capacitated facility location problems being extensions of some of the uncapacitated models presented earlier. We first propose several mono–objective approaches that later will lead us to multi–objective ones. We restrict our analysis to those problems in which capacities are inputs to the problem, assumed to be known precisely. We leave out of this study problems where the capacity (size) of facilities are decision variables (usually known as the class of capacity planning/expansion problems).

All the problem instances considered in the examples shown throughout this section have been randomly generated and solved by CPLEX MIP optimizer, v12.4.

2.4.1 Mono-objective approaches

Let us introduce the following notation, in addition to the one previously defined. For $j \in J$, K_j denotes the capacity of facility j in each time period (expressed in units of demand); for $(i, t, s) \in I \times \mathcal{T} \times \mathcal{S}$, let d_{it}^s be the total demand of customer i during time period t under scenario s; for $(i, j, t, s) \in I \times J \times \mathcal{T} \times \mathcal{S}$, c_{ijt}^s denotes the assignment cost of customer i's total demand to facility j in time period t under scenario s (in this case it is a function of d_{it}^s and the distance $dist_{ijt}^s$ between (i, j) in t under s, here the unit transportation cost, and thus $c_{ijt}^s = dist_{ijt}^s\, d_{it}^s$). In terms of decision variables: x_{jt} equals 1 if facility j is opened at the beginning of period t, and 0 otherwise; y_{ijt}^s represents the fraction of customer i's demand assigned to facility j in time period t under scenario s. We assume here that the demand of each customer can be assigned to more than one facility.

Considering the DUFLPU, defined by (2.1.1)–(2.1.6), a possible extension of this problem where capacities are associated with facilities, naturally called dynamic capacitated location problem under uncertainty, in short DCFLPU, can be formulated in an extensive form as follows:

$$\text{(DCFLPU)} \qquad \min \quad \sum_{t\in\mathcal{T}} \sum_{j\in J} \sum_{s\in\mathcal{S}} p^s f_{jt}^s x_{jt} + \sum_{s\in\mathcal{S}} \sum_{t\in\mathcal{T}} \sum_{i\in I} \sum_{j\in J} p^s c_{ijt}^s y_{ijt}^s \tag{2.4.1}$$

subject to

$$\sum_{j\in J} y_{ijt}^s = \delta_{it}^s \qquad \forall i \in I, t \in \mathcal{T}, s \in \mathcal{S}, \tag{2.4.2}$$

$$\sum_{i\in I} d_{it}^s y_{ijt}^s \le K_j \sum_{\tau=1}^{t} x_{j\tau} \qquad \forall j \in J, t \in \mathcal{T}, s \in \mathcal{S}, \tag{2.4.3}$$

$$\sum_{t\in\mathcal{T}} x_{jt} \le 1 \qquad \forall j \in J, \tag{2.4.4}$$

$$x_{jt} \in \{0, 1\} \qquad \forall j \in J, t \in \mathcal{T}, \tag{2.4.5}$$

$$y_{ijt}^s \ge 0 \qquad \forall i \in I, j \in J, t \in \mathcal{T}. \tag{2.4.6}$$

The above formulation is very similar to the one defined to the DUFLPU, namely the objective function (2.4.1) that minimizes the expected total cost (fixed plus assignment costs) over all scenarios. The difference goes to constraints (2.4.3) which dictate that customers' demand can only be assigned to open facilities and no facility can supply more than its capacity. This problem will have at least one admissible solution if and only if total demand does not exceed total capacity under all possible scenarios. However, it can be the case that total demand may not be satisfied under some scenario(s) given the established capacities. Consequently, the above problem can be infeasible, as opposite to the DUFLPU where an admissible solution always exists for all scenarios. For illustrative purposes, consider the small problem instance given in example 2.4.1.

Example 2.4.1 *Consider a problem instance with 2 possible scenarios, 2 time periods, 2 potencial facility locations and 4 possible customers. The possible demands of each customer in each time period for both scenarios are presented in table 2.4.1. The last row presents total demands. In addition, consider* $K_1 = 90$ *and* $K_2 = 150$ *(total potential capacity equals* 240 *units).*

Table 2.4.1: Possible customers'demand, (d^1_{it}, d^2_{it}).

t		1	2
	1	(85,85)	(93,88)
i	2	(49,49)	(48,53)
	3	(25,25)	(28,23)
	4	(68,68)	(73,61)
		(227,227)	**(242,225)**

We can see that total capacity will not be sufficient to satisfy total demand in time period two under scenario one. The above major problem can then be classified as infeasible or a problem without complete recourse, as it is designated in Stochastic Programming because there is not an admissible solution for all possible scenarios.

A possible extension of model DCFLPU is to consider unmet demand. More precisely, when location decisions are made, it is explicitly assumed by the DM that total demand may be unsatisfied in the future. In addition, it is also assumed that a penalty cost is incurred for each unit of demand not satisfied.
Let us represent the fraction of the unmet demand of customer i during t and under s by decision variable e^s_{it}, for all (i, t, s). In addition, β^s_{it} denotes the total cost of not fulfilling the customer i's total demand during t under s. We consider here a general situation where the penalty costs can be different for different customers, but an equal penalty cost for all customers could also be considered. An extension of the DCFLPU considering possible unmet demand can then be formulated as follows:

$$(\text{DCFLPU}_{II}) \qquad \min \qquad \sum_{t\in\mathcal{T}}\sum_{j\in J}\sum_{s\in\mathcal{S}} p^s f^s_{jt} x_{jt} + \sum_{s\in\mathcal{S}}\sum_{t\in\mathcal{T}}\sum_{i\in I}\sum_{j\in J} p^s c^s_{ijt} y^s_{ijt} + \sum_{s\in\mathcal{S}}\sum_{t\in\mathcal{T}}\sum_{i\in I} p^s\, \beta^s_{it}\, e^s_{it} \tag{2.4.7}$$

subject to

$$\sum_{j \in J} y_{ijt}^{s} + e_{it}^{s} = \delta_{it}^{s} \qquad \forall i \in I, t \in \mathcal{T}, s \in \mathcal{S}, \tag{2.4.8}$$

(2.4.3), (2.4.4),(2.4.5), (2.4.6),

$$e_{it}^{s} \geq 0 \qquad \forall i \in I, t \in \mathcal{T}, s \in \mathcal{S}. \tag{2.4.9}$$

The objective function (2.4.7) minimizes the expected total costs including fixed, assignment and penalty costs associated with unmet demands in the third term. Constraints (2.4.8) ensure that the total demand of each customer is distributed between met and unmet demand.

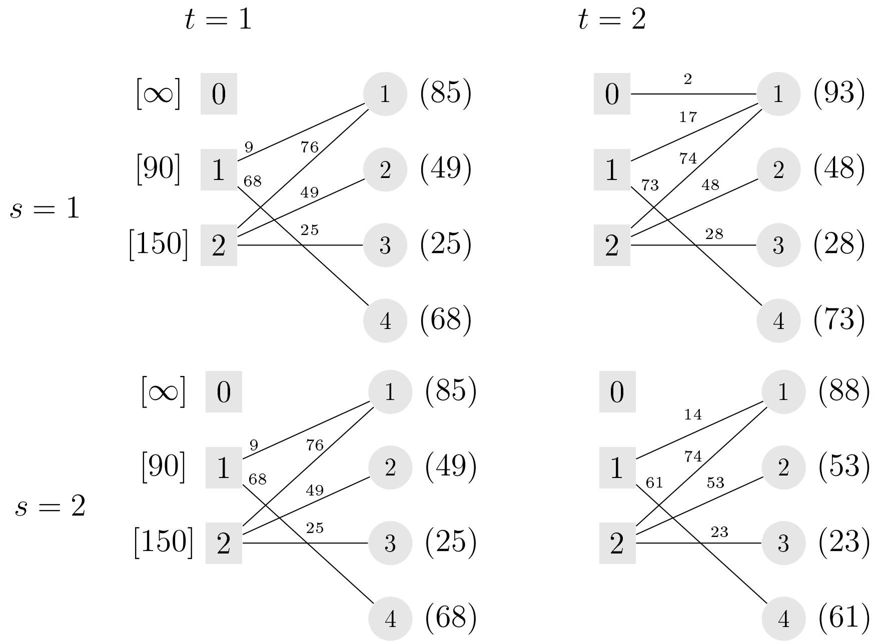
Feasibility is guaranteed by formulation DCFLPU$_{II}$ and its best solution will result of the compromise defined by the problem's data. In particular, the values of variables e_{it}^{s} will certainly depend on the diference between assignment costs and costs of not satisfying demand. Let us consider again the problem instance of example 2.4.1 in which total demand in period two under scenario one exceeds in two units the potential capacity. In order to fit this problem to this new situation, we have considered for all customers the penalty costs higher than the corresponding assignment costs, for illustrative purposes only. The optimal solution for this new problem, where obviously both facilities are opened, results with $e_{12}^{1} = 0.0215$ and $e_{it}^{s} = 0.0$ for all $(i, t, s) \neq (1, 2, 1)$. Hence, and as expected, only two units of demand in time period two under scenario one are not satisfied, in the present solution belonging to customer one.

Let us return to model DCFLPU and to those problems where the potential total capacity is sufficient to satisfy total demand. For instance, suppose that a third potential facility site with $K_3 \geq 2$ is added to the problem's data of example 2.4.1. First, it is easy to see that the DCFLPU is feasible and has several admissible solutions in which demand is fully satisfied. However, the *best* one will be dependent on the capacities, setup costs of those three facilities, assignment costs, in summary the problem's data. Assuming here the extreme situation in which the costs associated with that third facility are all higher than the costs associated with the other two services, the question goes to the practicability in terms of costs of one solution where three facilities have to be opened in order to satisfy total demand (in the present case, a third facility is opened to satisfy only the remaining two units under one single scenario). This extreme example is only to illustrate that, in spite of the DCFLPU being feasible, guaranteeing that total demand is satisfied under all possible scenarios, comes with a cost. Note that model DCFLPU$_{II}$ can also be applied whenever the potential total capacity is sufficient to satisfy total demand. It can be used to analyse the tradeoff between expected total costs, including fixed and assignment costs only, and expected total costs associated with unmet demand. The penalty costs represent the weight or importance given to satisfying demand. It is easy to see from the objective function (2.4.7) that if higher penalty costs are considered, more satisfied demand is expected, leading to an increase of the expected total costs associated with satisfied demand; on the other hand, smaller penalty costs will lead to solutions with more unsatisfied demand but also with smaller expected costs for satisfying demand. This reasoning leads us to multi-objective approaches that will be discussed in the following sub–section.

Before going any further, we shall remark that the above situations could be modelled through model DCFLPU with additional features instead of model DCFLPU$_{II}$. Assume that in the set of potential facility sites there is a potential facility site indexed by $j = 0$, for instance, with zero fixed costs and with a huge capacity (at least large enough to satisfy total demand). Throughout this text we will denote this new set of potential facility sites by $J_0 = J \cup \{0\}$ such that $f^s_{0t} = 0$ for all (t, s) and $K_0 = +\infty$. The demand assigned to this *virtual* facility, y^s_{i0t} for all (i, t, s), represents unsatisfied demand, and the assignment costs between this virtual facility and customers, c^s_{i0t} for all (i, t, s), are in fact penalty costs. Hence, if in model DCFLPU, defined by (2.4.1)–(2.4.6), set J is replaced by set J_0 we get also an extension of DCFLPU with possible unmet demand. Furthermore, considering $c^s_{i0t} = \beta^s_{it}$ for all (i, t, s), both models DCFLPU and DCFLPU$_{II}$ provide the same solution where $y^s_{i0t} = e^s_{it}$ for all (i, t, s). Considering this notation, the optimal solution for the problem of example 2.4.1 with unmet demand is partially depicted in figure 2.4.1.

A different perspective can be given of the above problem. Assume that total demand should be always satisfied (at any cost). A possibility is to assume explicitly future capacity shortages. Let us assume also that costs are associated to such shortages, interpreted in this context as penalty costs incurred by the increase of the capacities (by having to pay extra hours to employees, or buy some units in outsourcing for instance). Let us represent the capacity shortage of each open facility j during time period t and scenario s by decision variable o^s_{jt}. Let θ denote the cost of each unit of demand that is not satisfied by each open facility (equal for all facilities). We assume also that shortage costs are equal for all facilities. An extension of the DCFLPU considering possible capacity shortages can then be formulated as follows:

Figure 2.4.1: Optimal solution for example 2.4.1 with unsatisfied demand.



$$(\text{DCFLPU}_{III}) \qquad \min \quad \sum_{t\in\mathcal{T}}\sum_{j\in J}\sum_{s\in\mathcal{S}} p^s f^s_{jt} x_{jt} + \sum_{s\in\mathcal{S}}\sum_{t\in\mathcal{T}}\sum_{i\in I}\sum_{j\in J} p^s c^s_{ijt} y^s_{ijt} + \sum_{s\in\mathcal{S}} p^s \sum_{t\in\mathcal{T}}\sum_{j\in J} \theta\, o^s_{jt} \tag{2.4.10}$$

s.t.

(2.4.2),

$$\sum_{i\in I} d^s_{it} y^s_{ijt} \leq K_j \sum_{\tau=1}^{t} x_{j\tau} + o^s_{jt} \qquad \forall j \in J, t \in \mathcal{T}, s \in \mathcal{S}, \tag{2.4.11}$$

$$o^s_{jt} \leq \overline{M} \sum_{\tau=1}^{t} x_{j\tau} \qquad \forall j \in J, t \in \mathcal{T}, s \in \mathcal{S}, \tag{2.4.12}$$

(2.4.4),(2.4.5), (2.4.6),

$$o^s_{jt} \geq 0 \qquad \forall j \in J, t \in \mathcal{T}, s \in \mathcal{S}. \tag{2.4.13}$$

The objective function (2.4.10) minimizes the expected total costs including in the third term the costs associated with capacity shortages. Constraints (2.4.11) and (2.4.12), where $\overline{M}$ represents a very large number, ensure that customers' demand can only be assigned to open facilities and impose that the amount supplied by each open facility must be no greater than its available capacity plus its capacity shortage. Note that an a priori maximum shortage of each facility could be also imposed, instead of considering $\overline{M}$ in constraints (2.4.12), although, in this case, it is not possible to guarantee the existence of an admissible solution.

2.4.2 Multi-objective approaches

In this sub–section we propose several multi-objective approaches to the problems under study, given several perspectives to capacitated problems as well. In order to formulate the next problems, consider the set of potential facility sites given by $J_0 = J \cup \{0\}$, in order to include possible unsatisfied demand into the models as explained above. In what follows, we still represent the total cost (facility location and assignment of satisfied demand costs) achieved in scenario s by $\zeta_s(x, y)$. In addition, we represent the total cost associated with unmet demand in scenario s by $U_s(y)$:

$$U_s(y) = \sum_{t \in \mathcal{T}} \sum_{i \in I} c^s_{i0t} y^s_{i0t}. \tag{2.4.14}$$

We first propose a bi–objective problem where expected total costs, including fixed and assignment costs only, and the expected total penalty cost (associated with unmet demands) give rise to two distinct objective functions. We can formulate this bi–objective dynamic capacitated facility location problem under uncertainty, in short BODCFLPU, as follows, where set J is replaced by set J_0 in the set of constraints:

$$\text{(BODCFLPU)} \qquad \min \qquad \left\{ \sum_{s \in \mathcal{S}} p^s \zeta_s(x, y), \ \sum_{s \in \mathcal{S}} p^s U_s(y) \right\} \tag{2.4.15}$$

$$\text{s.t.}$$

$$(2.4.2)\text{–}(2.4.6).$$

The non-dominated solutions for this problem are the ones such that it is not possible to improve the expected total cost (fixed and assignment) for all scenarios without deteriorating the expected total penalty costs. Then, the analysis of the tradeoff between those two objectives, discussed earlier with model DCFLPU$_{II}$, can be made through model BODCFLPU, where a set of interesting solutions can be found and analyzed.

In order to offer a better picture of the compromises that exist among the possible scenarios, a multi-objective problem can be defined where each scenario will give rise to one objective. We are indeed proposing an extension of the multi-objective approach designed to the uncapacitated case, presented in section 2.3, to the capacitated problem. Thus, and now without making any assumptions about the risk profile or about the preferences of the DM, we can formulate a multi–objective dynamic capacitated facility location problem under uncertainty, in short MODCFLPU, as follows, where set J is replaced by set J_0 in the set of constraints:

$$\text{(MODCFLPU)} \qquad \min \quad \{\zeta_1(x,y)+U_1(y),...,\zeta_s(x,y)+U_s(y),...,\zeta_S(x,y)+U_S(y)\} \tag{2.4.16}$$

s.t.

(2.4.2)–(2.4.6).

The non–dominated solutions of MODCFLPU, as well as the non–dominated solutions of BODCFLPU, can be achieved by solving the corresponding auxiliary programming problems. We omit in this text their formulations taking into account its resemblance to the MODUFLPU considered in section 2.3. The non–dominated solutions can also be achieved following the procedure illustrated in section 3.4 for the MODUFLPU.

Strongly related with the type of facilities under study, as well as the products or services provided by such facilities, in reality it can be very difficult to estimate the unmet demand costs. This task can be easier if there are supply contracts that determine the fees that have to be paid for each unit of demand not satisfied, but it can be a hard task as in some health care services for instance. In the models proposed so far, those costs are given (possibly with uncertainty), but we now drop this requirement. In what follows, we may still have possible scenarios where total demand may not be satisfied. However, the costs associated with unsatisfied demand are not known, not even with uncertainty. For simplicity reasons, we will represent the total unmet demand in scenario s by $U_s(y)$ but, under such circumstances, defined as follows:

$$U_s(y) = \sum_{t\in\mathcal{T}} \sum_{i\in I} y^s_{i0t}. \tag{2.4.17}$$

Note that, if (2.4.17) is considered instead of (2.4.14) in BODCFLPU, then the non-dominated solutions of this model will represent compromises between expected total unmet demand and expected total cost.
Motivated by the previous model and taking into account the unknown penalty costs, a new problem can also be modelled that can provide additional information to the DM. To the objective functions corresponding to the total costs in each of the possible scenarios we add the set of functions corresponding to the total unmet demand in each scenario (if penalty costs are known, the total unmet demand cost could be considered instead). A new multi–objective problem can be defined with $2S$ objective functions, where each scenario will give rise to two distinct objectives. The aim is to minimize simultaneously total costs and total unmet demand for each of the possible scenarios.

The new multi–objective problem can then be formulated as follows, where set J is replaced by set J_0 in the set of constraints:

$$(\text{MODCFLPU}_{II}) \qquad \min \quad \{\zeta_1(x,y),...,\zeta_s(x,y),...,\zeta_S(x,y),U_1(y),...,U_S(y)\} \quad (2.4.18)$$

$$\text{s.t.}$$

$$(2.4.2)\text{–}(2.4.6).$$

The non–dominated solutions for the present problem are the ones such that it is not possible to improve the total cost (or total unmet demand) of one given scenario without deteriorating, at least, the total unmet demand (or total cost) of that scenario or the total cost or total unmet demand of one other scenario. Bellow, we present an illustrative example with a small problem instance. We report and analyse some of the non–dominated solutions calculated for this particular instance, with only two possible scenarios but where the tradeoff between the four objectives can be observed. The auxiliary programming problem to the MODCFLPU$_{II}$, that has been considered in the calculation of non–dominated solutions, is formulated next.
Let $\nu_1 \in \mathbb{R}^S$ and $\nu_2 \in \mathbb{R}^S$ be the vectors of weights associated with the objective functions of MODCFLPU$_{II}$, such that $\nu_{1s} > 0$ and $\nu_{2s} > 0$ for all $s \in \mathcal{S}$, and $\sum_{s\in\mathcal{S}}(\nu_{1s} + \nu_{2s}) = 1$. In addition, $\mathcal{M}_1 \in \mathbb{R}^S$ and $\mathcal{M}_2 \in \mathbb{R}^S$ represent the vectors of upper bounds to the objective functions. Then, the auxiliary programming problem to the MODCFLPU$_{II}$ is formulated as follows:

$$(\text{CAUX}) \qquad \min \quad \sum_{s\in\mathcal{S}} (\, \nu_{1s}\zeta_s(x,y) + \nu_{2s}U_s(y)\,) \quad (2.4.19)$$

$$\text{s.t.}$$

$$(2.4.2)\text{–}(2.4.6)$$

$$\zeta_s(x,y) \leq \mathcal{M}_{1s} \qquad \forall s \in \mathcal{S}. \quad (2.4.20)$$

$$U_s(y) \leq \mathcal{M}_{2s} \qquad \forall s \in \mathcal{S}. \quad (2.4.21)$$

Example 2.4.2 *Consider a problem instance with 2 possible scenarios, 5 time periods, 15 potencial facility locations (including the virtual one) and 50 possible customers.*

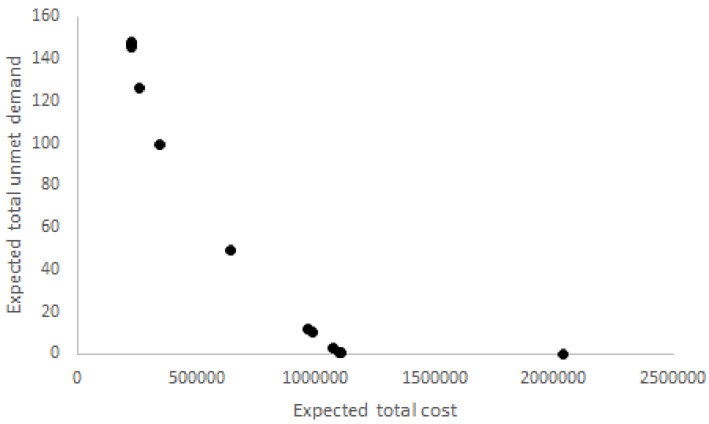
In table 2.4.2 we detail twenty non–dominated solutions of this problem instance that were found following an interactive procedure (see section 3.4 where this solution approach is applied to the MODUFLPU). For ease in the exposition of the results only, the solutions (objective function values and the corresponding location decisions) are ordered by non decreasing values of the total cost for scenario one, i.e. ζ_1. The best values found for each of the objectives are in bold. We can see that there are several sets of solutions with the same location decisions, same sites and time periods in which facilities are opened, although with different assignment decisions. Solutions number 1 and 2 are such an example, both with the best total cost for scenario 1, the total cost for scenario 2 improves but with an increase of the unmet demand for scenario 2. A similar behavior is observed between solutions 5 and 11, both with the best total cost for scenario 2, the total cost for scenario 1 worsens but the unmet demand for scenario 1 decreases. In solutions 3 and 4, with the same location decisions as well, total costs deteriorate in both scenarios with an improve of total unmet demand. The solutions from number 12 to 20 were obtained searching the regions defined by smaller upper bounds to the objectives U_1 and U_2, supposing that the DM is really interested in satisfying (almost) total demand and there will be sufficient resources to reach such goals. As shown by solution number 20, it is possible in this instance to satisfy total demand for both scenarios, though with the worst total costs observed. We note that we have chosen a problem instance where these solutions belong to the set of admissible solutions. However, such admissible solutions should be further analysed by the DM to decide if they are 'really' admissible (the increase in the cost that enables that total demand will be satisfied under all scenarios may be unbearable). It is out of our scope to present all the non–dominated solutions for this problem. Taking into account that in the present model we are dealing with $2S$ objectives, within an interactive approach the information given by the DM becomes crucial in order to restrict the regions of search, mainly in those problems where a huge number of possibilities may arise. For this instance some other non–dominated solutions were found with smaller values to total unmet demand, but no more by imposing smaller bounds to total costs than the ones presented here. We conclude stressing that facilities 9, 10, 13 and 14 are opened at the beginning of the planning horizon in all of the non-dominated solutions found.

Suppose that instead of model MODCFLPU$_{II}$ the DM is only interested in analyzing the compromise between expected total costs and expected total unmet demand. We return then to model BODCFLPU. For illustrative purposes, we have considered the problem's data of example 2.4.2 and fixed equal probabilities for both scenarios. By this example, we can confirm that models MODCFLPU$_{II}$ and BODCFLPU are indeed different problems. In fact, within the set of twenty non–dominated solutions of the multi-objective problem, eight become dominated on the bi-objective problem. The non–dominated solutions for this new problem are depicted in Figure 2.4.2, where it is easier to see that (expected) total costs increases as total satisfied demand also increases (or total unsatisfied demand decreases).

Table 2.4.2: Example 2.4.2: Time period in which each facility is opened.

					Opened Facilities														
	ζ_1	ζ_2	U_1	U_2	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14
1	**368907**	207845	165.14	125.98	1	1	2	1						1	1			1	1
2	**368907**	89370	165.04	128.31	1	1	2	1						1	1			1	1
3	369289	88547	166.95	128.68	1	1	2	4			1			1	1			1	1
4	369360.4	89459.8	165.50	125.50	1	1	2	4			1			1	1			1	1
5	381063	**85252**	166.99	129.10	1						1			1	1			1	1
6	420552.6	93738	125.50	127.46	1	1	2	1	2			1		1	1	3		1	1
7	420875.6	92719	125.50	127.46	1	1	2	1	2			1		1	1			1	1
8	421516.9	91581	125.50	127.12	1	1	2	4	2		1	1		1	1			1	1
9	555128	135632.2	99.50	99.50	1	1	2	4	2		1	1	3	1	1	1	1	1	1
10	904542	385418	49.50	49.50	1	1	2	1	2		1	1	3	1	1	1	1	1	1
11	998138	**85252**	145.16	129.10	1						1			1	1			1	1
12	1173015	806903	19.50	0.50	1	1	1	1	2		1	3	1	1	1	1	1	1	1
13	1222287.5	710482	14.50	8.50	1	1	1	1	2		1	3	1	1	1	1	1	1	1
14	1339496	806903	4.50	0.50	1	1	1	1	2		1	3	1	1	1	1	1	1	1
15	1395351	806903	0.50	0.50	1	1	1	1	2		1	3	1	1	1	1	1	1	1
16	1395351	814273	0.50	**0.00**	1	1	1	1	2		1	3	1	1	1	1	1	1	1
17	1395351	753989	0.50	4.50	1	1	1	1	2		1	3	1	1	1	1	1	1	1
18	1396208	608225	0.50	19.50	1	1	1	1	2		1	3	3	1	1	1	1	1	1
19	1402899	806903	**0.00**	0.50	1	1	1	1	2		1	3	1	1	1	1	1	1	1
20	2552968	1529019	**0.00**	**0.00**		1	1	1	1	1	1	1	1	1	1	1	1	1	1

Figure 2.4.2: Set of non–dominated solutions considering only two objective functions.



Chapter 3

Solution Approaches

Primal-dual heuristics have proven their value when dealing with facility location problems, whether being static and deterministic (Erlenkotter [30]), deterministic dynamic (Van Roy and Erlenkotter [88], Dias et al. [21]) or static under uncertainty (Louveaux and Peeters [56]). From the existing literature we have witnessed though that such techniques have not been applied in dynamic facility location under uncertainty yet. The complexity of the mathematical models under study as well as the success of such techniques when tackling related problems, were the main reasons to develop a primal–dual heuristic to tackle the DUFLPUD (Marques and Dias [58]). This dual-based heuristic is inspired on the classical approaches developed by Bilde and Krarup[13], Erlenkotter [29] and Van Roy and Erlenkotter [88]. The main idea of the approach is to obtain good solutions from the dual problem of the corresponding linear programming relaxation of the primal problem, more precisely from the so–called condensed dual problem. This technique is able to find admissible primal and dual solutions for feasible DUFLPUD. The heuristic's procedures (dual ascent, primal and adjustment procedure) detailed in section 3.1 are designed to reduce progressively the duality gap between dual and primal objective function values. In those problems for which the heuristic is unable to find the optimal solution, it is still able to provide upper and lower bounds to the optimum of DUFLPUD, being thus always possible to evaluate the quality of the best solution achieved. In order to solve DUFLPUD to optimality this primal–dual heuristic is integrated in a branch&bound approach (Marques and Dias [57]). Instead of solving to optimality relaxed versions of the original problems in each node of the branch&bound tree, we decided to use the dual-based heuristic to solve each problem. Considering now model α-DUFLPU, note that if constraints (2.2.3) are relaxed, a problem with the same structure of the DUFLPU is obtained, allowing then the use of the primal-dual heuristic to tackle that problem. Lagrangean relaxation is a well known technique that allows the calculation of lower bounds for integer programming problems (Reeves [70], Guignard [31]). Hence, a Lagrangean relaxation and a subgradient algorithm is developed to tackle α-DUFLPU.

There are several different ways of dealing with a multi–objective problem. One such way is the so-called interactive approach. The interactive approach considers interchanging calculation and dialogue phases. In the calculation phase a non-dominated solution is calculated and showed to the DM. The DM will then react by giving some new information that will guide the calculation of the new non-dominated solution to be calculated in the next iteration. The process continues until the DM is satisfied with a given solution or the whole set of non-dominated solutions is found (see, for instance, Dias et al. [20]). The major drawback of this approach has to do with the possibility of having calculation phases taking too much computational time, not promoting a real-time interaction and making the process not attractive to the DM. The main advantage has to do with the ability of searching areas of the solutions' surface that are interesting to the DM, not wasting time or resources calculating solutions that the DM will simply discard. Moreover, whenever a non-dominated solution is encountered, there is a region in the objective space that is no longer interesting (the one that is dominated by this solution), and another region where there cannot be any admissible solutions (or else this solution would not be non-dominated). So, it is possible, in each iteration, to eliminate regions from further searches.

Another way of dealing with multi-objective problems considers the *a priori* and off-line calculation of the whole set (or a significant number) of non-dominated solutions. The solutions can then be presented to the DM, all at the same time, or using an interactive approach similar to the one previously described. One of the advantages of this approach is that the computational burden of calculating the solutions is made *a priori*, promoting a faster action-reaction interaction with the DM since no optimizations will be done. The choice between an interactive or a generation approach should be done considering several aspects of the problem such as its dimension or the time needed to calculate a solution for instance. As stated in section 2.3, the set of non–dominated solutions of MODUFLPU is achieved by solving the auxiliary problem(s) AUX. It is quite *easy* to embed the use of AUX in both an interactive and an off-line generation procedure, where the whole set of efficient solutions can be calculated. Note that the AUX formulation presented can result in a computationally heavy integer programming problem. It is a *NP-hard* problem, and the computational time needed to calculate a given solution will be heavily dependent on the problem's dimension, especially the number of scenarios and the number of potential facility locations. To solve AUX we can resort to general solvers or use dedicated procedures, both exact and heuristic procedures. Although the latter will not be able to guarantee the optimality of the calculated solution, they can be a very good choice especially in the presence of an interactive procedure, where the most important thing will be to define a region of interest for the DM. It is even possible to think of using a heuristic procedure in a first stage, and then an exact procedure to actually guarantee the optimality of the solution of interest. For illustrative purposes, considering a problem instance we propose here an interactive procedure based on Dias et al. [20], where all AUX problem instances were solved by a general solver.

3.1 Primal-Dual heuristic

For ease in the exposition, let us reindex, for each scenario s, $\mathcal{C}_{ijt}^{s}$ for each (i,t) in nondecreasing order as $\mathcal{C}_{it}^{s(k)}$, for $k = 1, 2, ..., k_{it}^{s}$, where k_{it}^{s} denotes the number of facility-to-customer links for (i,t) under scenario s. Thus, $\mathcal{C}_{it}^{s(1)} = \min_{j \in J}\{\mathcal{C}_{ijt}^{s}\}$. For convenience, we also include $\mathcal{C}_{it}^{s(k_{it}^{s}+1)} = +\infty, \forall\ (i,t,s)$.
Let I^{+} be the set of *pseudo customers* (i,t,s) corresponding to the dual variables v_{it}^{s} that the procedure will try to increase. Initially, I^{+} will be equal to all possible combinations $(i,t,s) \in I \times \mathcal{T} \times \mathcal{S}$, except those such that $\delta_{it}^{s} = 0$. Later, I^{+} will be set within the respective procedures. We note that a customer without demand does not contribute to the improvement of the dual objective function value and does not also contribute to any violation of the complementary slackness conditions. Thus, these customers are excluded from the ascent procedures.
The steps of the heuristic are as follows:

1. Set $v_{it}^{s} = \mathcal{C}_{it}^{s(1)}, \forall\ (i,t,s)$, and $u_j = 0, \forall\ j$.
   Set $I^{+} = \{(i,t,s) \in I \times \mathcal{T} \times \mathcal{S} : \delta_{it}^{s} = 1\}$.
2. Execute the dual ascent procedure.
3. Execute the primal procedure. If an optimal solution is found, then stop.
4. Execute the primal–dual adjustment procedure.

The heuristic stops when the optimal solution is found or when there are no primal or dual improvements after a given number of trials within the adjustment procedure.

3.1.1 Dual ascent procedure

This procedure, that may start with any dual feasible solution, will try to increase the values of variables v_{it}^{s} belonging to set I^{+}. The increase of such variables will lead to an increase of the dual objective function value and, simultaneously, to the decrease of some slacks' values (see step 6). The maximum value that variables v_{it}^{s} can take is limited by restrictions (2.1.15). Equivalently, we can also consider slacks defined by (2.1.17) and acknowledge that these slacks have to remain nonnegative. Instead of increasing the value of each dual variable v_{it}^{s} as much as possible in one single step, the procedure follows an iterative approach: in each iteration, the algorithm will try to increase a dual variable v_{it}^s to the smallest $\mathcal{C}_{ijt}^s$ that is greater than or equal to the current v_{it}^s value. If this is not possible, due to the fact that at least one slack would become negative, than the variable is increased as much as possible guaranteeing that all slacks remain nonnegative (steps 4, 5 and 6). The procedure is repeated until it is not possible to increase the value of any variable v_{it}^s because of the slacks that are already equal to zero. The slacks that are equal to zero will define the set of candidate facility locations.

In what follows, $(i,t,s)_q$, with $q \leq |I \times \mathcal{T} \times \mathcal{S}|$, represents a given, but arbitrary, sequence of pseudo customers.

1. Consider any dual feasible solution $\{v_{it}^s\}$ such that $v_{it}^s \geq \mathcal{C}_{it}^{s(1)}, \forall\ (i,t,s)$, and $\pi_{jt} \geq 0, \forall\ (j,t)$.

   For each (i,t,s) define $k(i,t,s) = \min\{k : v_{it}^s \leq \mathcal{C}_{it}^{s(k)}\}$. If $v_{it}^s = \mathcal{C}_{it}^{s(k(i,t,s))}$, then $k(i,t,s) \leftarrow k(i,t,s)+1$.

2. $(i,t,s) \leftarrow (i,t,s)_1$ and $q \leftarrow 1$; $r = 0$.

3. If $(i,t,s) \notin I^+ \vee \delta_{it}^s = 0$, then go to step 7.

4. Set $\Delta_{it}^s = \min_j\{\pi_{j\tau} : v_{it}^s - \mathcal{C}_{ijt}^s \geq 0, \tau \leq t\}$.

5. If $\Delta_{it}^s > \mathcal{C}_{it}^{s(k(i,t,s))} - v_{it}^s$, then $\Delta_{it}^s = \mathcal{C}_{it}^{s(k(i,t,s))} - v_{it}^s$; $r = 1$; $k(i,t,s) \leftarrow k(i,t,s)+1$.

6. For all $j \in J$ with $v_{it}^s - \mathcal{C}_{ijt}^s \geq 0$, set $\pi_{j\tau} = \pi_{j\tau} - \Delta_{it}^s, \tau \leq t$; set $v_{it}^s = v_{it}^s + \Delta_{it}^s$.

7. If $q < |I^+|$, then $q \leftarrow q+1$, $(i,t,s) \leftarrow (i,t,s)_q$, and return to step 3.

8. If $r = 1$, then return to step 2, otherwise stop.

3.1.2 Primal procedure

From the dual ascent procedure results the dual feasible solution $\{v_{it}^{s+}\}$ with an objective function value v_D^+, and associated slacks $\{\pi_{jt}^+\}$. A corresponding primal feasible solution, $\{x_{jt}^+\}$ and $\{y_{ijt}^{s+}\}$, can be constructed, with an objective function value v_P^+.
In order to describe the primal procedure, let us first define the following sets:

$J^* = \{(j,t) \in J \times \mathcal{T} : \pi_{jt}^+ = 0\}$;

$J_t^* = \{j \in J : (j,\tau) \in J^*, \tau \leq t\}, \forall t \in \mathcal{T}$;

$J_t^+ = \{j \in J : \text{facility } j \text{ is open at time } t\}, \forall t \in \mathcal{T}$.

In addition, define $t_1(j) = \min\{\gamma : j \in J_\gamma^+\}$ and $t_2(j) = \max\{\gamma \leq t_1(j) : (j,\gamma) \in J^*\}$. Then,

$J^+ = \{(j, t_2(j)) \in J \times \mathcal{T} : j \in J_\tau^+ \text{ for some } \tau\}$.

The set J^* corresponds to all (j,t) such that j can be opened at the beginning of t without violating (2.1.18); set J_t^* corresponds to all j that can be opened up to t; set J_t^+ corresponds to all j that are actually open during t; set $J^+ \subseteq J^*$ corresponds to all j that open at the beginning of t, i.e., J^+ dictates what facilities are actually opened and when (location decisions).
The facilities that are considered first (step 2) are the ones that at a given time t should be assigned to a given customer (i,s), according to conditions (2.1.20), called *essential* facilities. Other facilities are only opened if strictly necessary (step 3). If a facility j needs to be open at some time period(s) and the first time period when it needs to be open is t, then it will be opened at the beginning of time period $t_2(j)$, defined as being the time period closest to t such that the corresponding slack is equal to zero. It should be noted that, as we are dealing with an uncapacitated location problem, there will always be an admissible solution that can be built in this way: we can be sure that there exists at least one facility j such that π_{j1} is equal to zero (at least one facility can be opened at the beginning of the first time period). If this was not true, then it would still be possible to improve the dual solution by increasing at least one v_{i1}^s dual variable.
The steps of the primal procedure are as follows:

1. Set $J^+ = J_t^+ = \emptyset$, $\forall t$. Build J^* and J_t^*, $\forall t$.

2. For each $t \in \mathcal{T}$, if $j \in J_t^*$ such that $\exists (i,s) : v_{it}^{s+} \geq \mathcal{C}_{ijt}^s$ and $v_{it}^{s+} < \mathcal{C}_{ij't}^s, \forall j' \in J_t^* \backslash \{j\}$, then $J_\tau^+ = J_\tau^+ \cup \{j\}$, $\forall \tau \geq t$.

3. For each (i,t,s), if $\nexists j \in J_t^+$ with $v_{it}^{s+} \geq \mathcal{C}_{ijt}^s$, then
   $J_\tau^+ = J_\tau^+ \cup \big\{ j \in J_t^* : \mathcal{C}_{ijt}^s = \min\{\mathcal{C}_{ij't}^s : v_{it}^s \geq \mathcal{C}_{ij't}^s\}\big\}$, $\forall \tau \geq t$.

4. Build J^+.

5. Update J_t^+, $\forall t$. Assign each (i,t,s) to facility $j \in J_t^+$ with lowest $\mathcal{C}_{ijt}^s$.

3.1.3 Primal–Dual adjustment procedure

The primal–dual adjustment procedure will try to enforce the conditions (2.1.20) that are still being violated by the current solution. The violation of these conditions means that, for a given scenario s, time period t and customer i, there are at least two variables w_{ijt}^s different from zero such that the corresponding facilities j are both open in period t. The only way of satisfying (2.1.20) would be to assign customer i to more than one opened facility, which is not admissible from the primal problem point of view. This procedure will try to change the current dual solution, by decreasing the value of at least one variable v_{it}^s (and thus possibly decreasing the value of some variables w_{ijt}^s), such that at least two slacks will be increased. The changes in the slacks' values may lead to the increase of other dual variables increasing the dual objective function value.

In order to describe the primal–dual procedure, let us first consider the additional sets:

$J_{it}^{s*} = \{j : \exists \tau \leq t \mid (j, \tau) \in J^* \text{ and } v_{it}^s \geq \mathcal{C}_{ijt}^s\}, \forall (i, t, s)$;

$J_{it}^{s+} = \{j : \exists \tau \leq t \mid (j, \tau) \in J^+ \text{ and } v_{it}^s > \mathcal{C}_{ijt}^s\}, \forall (i, t, s)$;

$I_{jt}^+ = \{(i, \tau, s) : J_{i\tau}^{s*} = \{j\} \text{ for } \tau \geq t\}, \forall (j, t)$.

In addition, we denote a best source and a second-best source for (i, t, s) in J_t^+ by $j(i, t, s)$ and $j'(i, t, s)$, respectively:

$\mathcal{C}_{ij(i,t,s)t}^s = \min_{j \in J_t^+}\{\mathcal{C}_{ijt}^s\}, \forall (i, t, s)$;

$\mathcal{C}_{ij'(i,t,s)t}^s = \min_{j \in J_t^+, j \neq j(i,t,s)}\{\mathcal{C}_{ijt}^s\}, \forall (i, t, s)$ for $|J_{it}^{s+}| > 1$.

And we define, $\mathcal{C}_{it}^{s-} = \max_j\{\mathcal{C}_{ijt}^s : v_{it}^s > \mathcal{C}_{ijt}^s\}$.

For a given (i, t, s), the set J_{it}^{s*} represents all facilities j that can be open at period t (because a slack $\pi_{j\tau}$ is equal to zero for some $\tau \leq t$) and such that if j is open then customer i can be assigned to j at period t under scenario s. Similarly, for a given (i, t, s), the set J_{it}^{s+} considers all facilities that are in operation during period t in the current primal solution, and such that customer i would have to be assigned to j in period t under scenario s to guarantee the satisfaction of (2.1.20). If $|J_{it}^{s+}| > 1$, for some (i, t, s), then a complementary slackness condition (2.1.20) is violated. In such case, the decrease of the variable v_{it}^s causes the increase of at least two slacks $\pi_{j\tau}$, associated with distinct facilities (step 4). Set I_{jt}^+ corresponds to all variables $v_{i\tau}^s$ whose value can be increased with the increase of slacks $\pi_{j\tau}, \tau \leq t$, and that must be constructed to the execution of the dual ascent procedure (step 5).

The steps of the primal-dual adjustment are:

1. $(i, t, s) \leftarrow (i, t, s)_1, q \leftarrow 1$; set $v_D = v_D^+$ and $v_P = v_P^+$; set $r = 0$.

2. If $|J_{it}^{s+}| \leq 1$, then go to step 9.

3. If $I_{j(i,t,s)t}^+ = \emptyset$ and $I_{j'(i,t,s)t}^+ = \emptyset$, then go to step 9.

4. For each (j, τ), with $\tau \leq t$ and $v_{it}^s > \mathcal{C}_{ijt}^s$, set $\pi_{j\tau} = \pi_{j\tau} + v_{it}^s - \mathcal{C}_{it}^{s-}$; set $v_{it}^s = \mathcal{C}_{it}^{s-}$.

5. (a) Set $I^+ = I_{j(i,t,s)t}^+ \cup I_{j'(i,t,s)t}^+$ and execute the dual ascent procedure.

   (b) Set $I^+ = I^+ \cup \{(i, t, s)\}$ and execute the dual ascent procedure.

   (c) Set $I^+ = I \times \mathcal{T} \times \mathcal{S}$ and execute the dual ascent procedure.

6. If v_{it}^s is changed, then return to step 2.

7. Execute the primal procedure.

8. If neither $v_D^+ > v_D$ nor $v_P^+ < v_P$, then $r \leftarrow r + 1$; otherwise $r \leftarrow 0$ and update v_D and v_P.

9. If $v_D \geq v_P$, or $r = r_{max}$ or $q = |I \times \mathcal{T} \times \mathcal{S}|$, then stop; otherwise $q \leftarrow q+1$, $(i,t,s) \leftarrow (i,t,s)_q$, and return to step 2.

As the primal–dual heuristic for the DUFLPUD has been described, we now explain the changes that have to be made in the above procedures in order to adjust the approach for the version of the problem considered in subsection 2.1.2. The procedures are in fact very similar for both situations, but the variations are crucial. First, it is worthwhile to compare slack variable values defined by (2.1.17) and (2.1.26), for the first and second situations, respectively. Note that (2.1.26) will not be decreased whenever $\rho^s_{j\tau} = 0$, for some $\tau \geq t$ and s. Consequently, during the research for the set of candidate facility locations, within the dual ascent procedure (subsection 3.1.1), the pseudo–customers under that scenario will no longer contribute to the decrease of the slack values and thus to the opening of these facility sites. However, it is possible that other pseudo–customers, under other scenarios $s' \neq s$ for which $\rho^{s'}_{j\tau} = 1$, might contribute to the decrease of the slack and thus to a new set of candidate facility locations for that scenarios only. Consequently, in terms of primal procedure (subsection 3.1.2), in addition to consider assignments only to open facilities, that were opened at the beginning of some time period t, it must be also guaranteed that those facilities are such that $\rho^s_{jt} = 1$.

3.1.4 Illustrative examples

We illustrate the heuristic by two small examples. Real-world problems are typically much larger and provide more challenging situations. For the sake of simplicity, we consider problems with only two scenarios, both with $p^1 = 0.70$ and $p^2 = 0.30$, three time periods ($T = 3$), three potencial facility locations ($M = 3$) and four potencial customers ($N = 4$). In terms of the primal formulations, we are dealing with problems with *only* 81 decision variables and 99 restrictions.

Example 3.1.1 *Consider the problem's data in Tables 3.1.1–3.1.3: possible customers, assignment and fixed costs, respectively. We note that at* $t = 1$ *(present time) the input data is the same for both scenarios. In table 3.1.1 we can see that, under scenario 2, customer* 1*'s demand's should not be considered in period* $t = 3$ *nor customer* 4*'s demand's for periods* $t > 1$.
The weighted assignment costs are presented in Table 3.1.4. The initial dual solution and the initial slacks (derived after the weighting of the fixed costs) are shown in Tables 3.1.5 and 3.1.6, respectively.

The dual ascent procedure tries to increase the variables v^s_{it} belonging to I^+, following an arbitrary sequence of these variables. We chose to consider the variables ordered by increasing values of t, s and i, respectively. We show below some of the first steps of the algorithm.

Table 3.1.1: Possible customers, $(\delta^1_{it}, \delta^2_{it})$.

	t	1	2	3
	1	(1,1)	(1,1)	(1,0)
i	2	(1,1)	(1,1)	(1,1)
	3	(1,1)	(1,1)	(1,1)
	4	(1,1)	(1,0)	(1,0)

Table 3.1.2: Assignment costs, (c^1_{ijt}, c^2_{ijt}).

	t	1			2			3		
	j	1	2	3	1	2	3	1	2	3
	1	(5,5)	(7,7)	(10,10)	(7,10)	(8,9)	(13,14)	(9,–)	(8,–)	(19,–)
i	2	(10,10)	(6,6)	(6,6)	(11,12)	(7,7)	(8,11)	(12,11)	(7,7)	(10,13)
	3	(6,6)	(10,10)	(12,12)	(7,9)	(11,13)	(13,13)	(7,10)	(13,15)	(13,14)
	4	(4,4)	(7,7)	(12,12)	(6,–)	(10,–)	(14,–)	(7,–)	(11,–)	(14,–)

Table 3.1.3: Fixed costs, f^s_{jt}.

t	1			2			3		
$s \diagdown j$	1	2	3	1	2	3	1	2	3
1	7	8	$+\infty$	9	10	11	$+\infty$	11	12
2	7	8	$+\infty$	12	10	12	$+\infty$	15	12

Table 3.1.4: Weighted assignment costs, $\mathcal{C}^s_{ijt}$.

		t	1			2			3		
		j	1	2	3	1	2	3	1	2	3
		1	3.5	4.9	7.0	4.9	5.6	9.1	6.3	5.6	13.3
$s = 1$	i	2	7.0	4.2	4.2	7.7	4.9	5.6	8.4	4.9	7.0
		3	4.2	7.0	8.4	4.9	7.7	9.1	4.9	9.1	9.1
		4	2.8	4.9	8.4	4.2	7.0	9.8	4.9	7.7	9.8
		1	1.5	2.1	3.0	3.0	2.7	4.2	–	–	–
$s = 2$	i	2	3.0	1.8	1.8	3.6	2.1	3.3	3.3	2.1	3.9
		3	1.8	3.0	3.6	2.7	3.9	3.9	3.0	4.5	4.2
		4	1.2	2.1	3.6	–	–	–	–	–	–

$(t, s) = (1, 1)$

$i = 1$:
$\min_j\{\pi_{j1} : v^1_{11} - \mathcal{C}^1_{1j1} \geq 0\} = \pi_{11} = 7$, $\Delta^1_{11} = \min\{7, 4.9 - 3.5\} = 1.4$, $\pi_{11} = 7 - 1.4 = 5.6$, $v^1_{11} = 3.5 + 1.4 = 4.9$;

Table 3.1.5: Initial dual solution, (v^1_{it}, v^2_{it}).

t		1	2	3
	1	(3.5, 1.5)	(4.9, 2.7)	(5.6,–)
i	2	(4.2, 1.8)	(4.9, 2.1)	(4.9, 2.1)
	3	(4.2, 1.8)	(4.9, 2.7)	(4.9, 3.0)
	4	(2.8, 1.2)	(4.2,–)	(4.9,–)

Table 3.1.6: Initial slacks, π_{jt}.

t		1	2	3
	1	7.0	9.9	$+\infty$
j	2	8.0	10.0	12.2
	3	$+\infty$	11.3	12.0

$i = 2$:

$\min_j\{\pi_{j1} : v^1_{21} - \mathcal{C}^1_{2j1} \geq 0\} = \min_j\{\pi_{21}, \pi_{31}\} = 8$, $\Delta^1_{21} = \min\{8, 4.2 - 4.2\} = 0$, $v^1_{21} = 4.2$;

$i = 3$:

$\min_j\{\pi_{j1} : v^1_{31} - \mathcal{C}^1_{3j1} \geq 0\} = \pi_{11} = 5.6$, $\Delta^1_{31} = \min\{5.6, 7 - 4.2\} = 2.8$, $\pi_{11} = 5.6 - 2.8 = 2.8$, $v^1_{31} = 4.2 + 2.8 = 7$;

$i = 4$:

$\min_j\{\pi_{j1} : v^1_{41} - \mathcal{C}^1_{4j1} \geq 0\} = \pi_{11} = 2.8$, $\Delta^1_{41} = \min\{2.8, 4.9 - 2.8\} = 2.1$, $\pi_{11} = 2.8 - 2.1 = 0.7$, $v^1_{41} = 2.8 + 2.1 = 4.9$.

The algorithm proceeds to $(t, s) = (1, 2)$, increasing v^2_{11} to 2.1 and v^2_{31} to 1.9. Afterwards, for $t = 2$ and $s = 1$, v^1_{12} is blocked by $\pi_{11} = 0$; for $i = 2$:

$\min_j\{\pi_{j\tau} : v^1_{22} - \mathcal{C}^1_{2j2} \geq 0, \tau \leq 2\} = \min\{\pi_{21}, \pi_{22}\} = \pi_{21} = 8$, $\Delta^1_{22} = \min\{8, 5.6 - 4.9\} = 0.7$, $\pi_{21} = 8 - 0.7 = 7.3$, $\pi_{22} = 10 - 0.7 = 9.3$, $v^1_{22} = 4.9 + 0.7 = 5.6$.

The dual ascent procedure continues until all the dual variables are blocked by some slack. At the end, we obtain the dual solution $\{v^{s+}_{it}\}$ and associated slacks $\{\pi^+_{jt}\}$ shown in Tables 3.1.7 and 3.1.8, respectively. In addition, at the end of this procedure $u_j = 0$, $\forall j$. The corresponding dual objective function value is equal to $v^+_D = 87.8$.

With sets $J^* = \{(1, 1), (2, 1)\}$, $J^*_t = \{1, 2\}$, $\forall t$, the primal procedure advances with sets $J^+ = J^*$ and $J^+_t = J^*_t, \forall t$. In fact, facilities 1 and 2 are both essencial for some customers at $t = 1$. For instance, $v^{1+}_{21} > \mathcal{C}^1_{221}$ but $v^{1+}_{21} < \mathcal{C}^1_{211}$, and $v^{2+}_{31} > \mathcal{C}^2_{311}$ but $v^{2+}_{31} < \mathcal{C}^2_{321}$, thus $t_1(j) = t_2(j) = 1$, $j = 1, 2$. Then, $v^+_P = 87.8 = v^+_D$, which means that the optimal solution has been found (illustrated in Figure 3.1.1). Despite the simplicity of this example, some of the inherent features of a nondeterministic and dynamic problem can be observed.

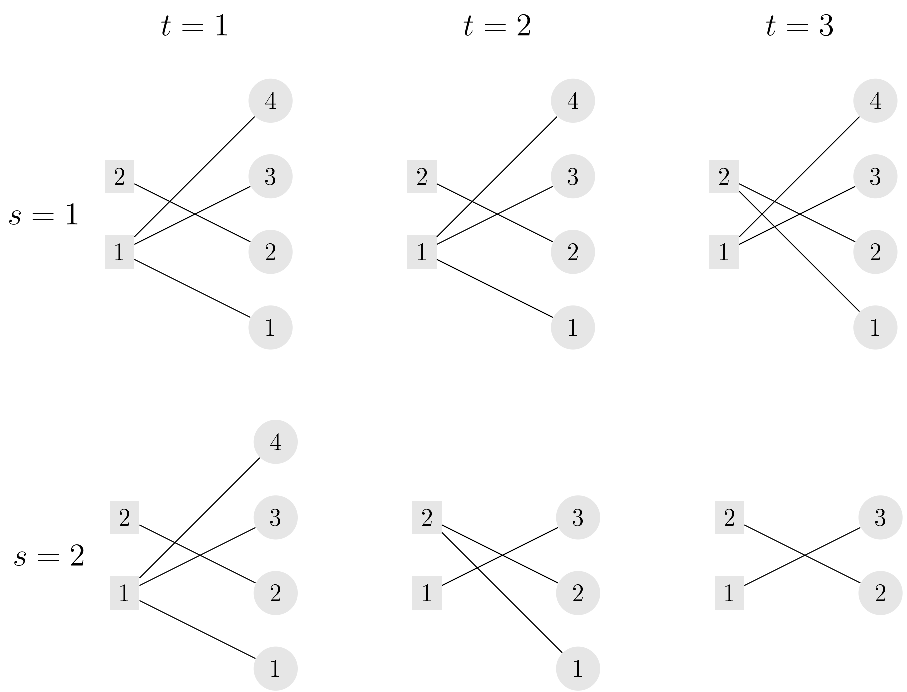
Table 3.1.7: Dual solution from the ascent procedure, $(v_{it}^{1+}, v_{it}^{2+})$.

t		1	2	3
	1	(4.9, 2.1)	(4.9, 3.0)	(6.3,–)
i	2	(6, 1.8)	(5.6, 3.3)	(7.0, 3.3)
	3	(7, 1.9)	(4.9, 2.7)	(4.9, 3.0)
	4	(4.9, 1.2)	(4.2,–)	(4.9,–)

Table 3.1.8: Slacks, π_{jt}^{+}.

t		1	2	3
	1	0	9.9	$+\infty$
j	2	0	3.8	8.2
	3	$+\infty$	11.3	12

Figure 3.1.1: Optimal solution for example 3.1.1



Example 3.1.2 *Consider the problem's data in Tables 3.1.9–3.1.11. As in the previous example, at* $t = 1$ *the input data is the same for both scenarios. The weighted assignment costs are presented in Table 3.1.12. The initial dual solution and the initial slacks are shown in Tables 3.1.13 and 3.1.14, respectively.*

Table 3.1.9: Possible customers, $(\delta^1_{it}, \delta^2_{it})$.

t		1	2	3
	1	(1,1)	(1,0)	(1,0)
i	2	(1,1)	(1,1)	(1,1)
	3	(1,1)	(1,1)	(1,1)
	4	(1,1)	(1,0)	(1,0)

Table 3.1.10: Assignment costs, (c^1_{ijt}, c^2_{ijt}).

t		1			2			3		
j		1	2	3	1	2	3	1	2	3
	1	(5,5)	(8,8)	(10,10)	(7,–)	(9,–)	(11,–)	(9,–)	(12,–)	(12,–)
i	2	(8,8)	(5,5)	(6,6)	(11,8)	(6,7)	(7,9)	(13,13)	(7,8)	(10,12)
	3	(6,6)	(5,5)	(7,7)	(7,7)	(6,8)	(8,12)	(7,8)	(9,8)	(8,13)
	4	(4,4)	(6,6)	(8,8)	(6,–)	(7,–)	(9,–)	(7,–)	(8,–)	(9,–)

Table 3.1.11: Fixed costs, f^s_{jt}.

t	1			2			3		
$s \diagdown j$	1	2	3	1	2	3	1	2	3
1	15	17	13	17	19	14	$+\infty$	20	15
2	15	17	13	18	19	15	$+\infty$	21	15

Table 3.1.12: Weighted assignment costs, $\mathcal{C}^s_{ijt}$.

	t		1			2			3		
	j		1	2	3	1	2	3	1	2	3
		1	3.5	5.6	7.0	4.9	6.3	7.7	6.3	8.4	8.4
$s=1$	i	2	5.6	3.5	4.2	7.7	4.2	4.9	9.1	4.9	7.0
		3	4.2	3.5	4.9	4.9	4.2	5.6	4.9	6.3	5.6
		4	2.8	4.2	5.6	4.2	4.9	6.3	4.9	5.6	6.3
		1	1.5	2.4	3.0	–	–	–	–	–	–
$s=2$	i	2	2.4	1.5	1.8	2.4	2.1	2.7	3.9	2.4	3.6
		3	1.8	1.5	2.1	2.1	2.4	3.6	2.4	2.4	3.9
		4	1.2	1.8	2.4	–	–	–	–	–	–

After the dual ascent procedure, we obtain the dual solution and associated slacks shown in Tables 3.1.15 and 3.1.16, respectively. At the end of this procedure $u_j = 0$, $\forall j$. We can see that all dual variables belonging to I^+ were increased, except the one corresponding to the pseudo customer $(i, t, s) = (3, 3, 2)$. The corresponding dual objective function value is equal to $v^+_D = 94.4$.

Table 3.1.13: Initial dual solution, (v_{it}^1, v_{it}^2).

t		1	2	3
	1	(3.5, 1.5)	(4.9,–)	(6.3,–)
i	2	(3.5, 1.5)	(4.2, 2.1)	(4.9, 2.4)
	3	(3.5, 1.5)	(4.2, 2.1)	(4.9, 2.4)
	4	(2.8, 1.2)	(4.2,–)	(4.9,–)

Table 3.1.14: Initial slacks, π_{jt}.

t		1	2	3
	1	15.0	17.3	$+\infty$
j	2	17.0	19.0	20.3
	3	13.0	14.3	15.0

Table 3.1.15: Dual solution from the ascent procedure, $(v_{it}^{1+}, v_{it}^{2+})$.

t		1	2	3
	1	(7, 3)	(6.3,–)	(8.4,–)
i	2	(5.6, 2.4)	(7.7, 2.4)	(8.1, 3.6)
	3	(4.9, 1.8)	(4.9, 2.4)	(5.6, 2.4)
	4	(5.6, 1.8)	(4.9,–)	(5.6,–)

Table 3.1.16: Slacks, π_{jt}^+.

t		1	2	3
	1	0.0	11.4	$+\infty$
j	2	0.0	10.1	15.9
	3	7.1	10.4	13.9

With sets $J^* = \{(1,1),(2,1)\}$, $J_t^* = \{1,2\}, \forall t$, the primal procedure advances with sets $J^+ = J^*$ and $J_t^+ = J_t^*, \forall t$. Facilities 1 and 2 are both essential at $t = 3$, then $t_1(1) = t_1(2) = 3$ and $t_2(1) = t_2(2) = 1$. The primal objective function value equals $v_P^+ = 98.5 > v_D^+$, so the heuristic continues to the primal–dual adjustment procedure.
The previous result means that at least one of the conditions (2.1.20) is violated. For instance, $v_{11}^{1+} > \mathcal{C}_{1j1}^1$, for $j = 1, 2$, thus $|J_{11}^{1+}| = 2$.
The best source and the second-best source for pseudo costumer $(i,t,s) = (1,1,1)$ are, respectively, $j(1,1,1) = 1$ and $j'(1,1,1) = 2$. In addition, $I_{11}^+ = \{(3,3,1)\}$ and $I_{21}^+ = \{(2,3,1),(2,3,2)\}$. Within the primal-dual adjustment procedure, slacks π_{11}^+ and π_{21}^+ are increased $v_{11}^{1+} - \mathcal{C}_{11}^{1-} = 7 - 5.6 = 1.4$ units and v_{11}^{1+} is decreased to $\mathcal{C}_{11}^{1-} = 5.6$. After the dual ascent procedures, initially with $I^+ = \{(3,3,1),(2,3,1),(2,3,2)\}$, no further

improvements are possible. The resulting dual solution is presented in Table 3.1.17, with associated slacks presented in Table 3.1.18. The dual objective function value is updated to $v_D = 95.1$.

Table 3.1.17: Dual solution after the dual ascent procedures within the primal-dual adjustment procedure.

t		1	2	3
	1	(5.6, 3)	(6.3,–)	(8.4,–)
i	2	(5.6, 2.4)	(7.7, 2.4)	(**9.2**, **3.9**)
	3	(4.9, 1.8)	(4.9, 2.4)	(**6.3**, 2.4)
	4	(5.6, 1.8)	(4.9,–)	(5.6,–)

Table 3.1.18: Slacks after the dual ascent procedures within the primal-dual adjustment procedure.

t		1	2	3
	1	0.6	10.6	$+\infty$
j	2	0.0	8.7	14.5
	3	5	8.3	11.8

From the primal procedure results $J^* = J^+ = \{(2,1)\}$, and $J_t^+ = \{2\}, \forall t$, then $v_P = 95.1 = v_D$, which means that the heuristic found the optimal solution.

3.2 Branch&Bound approach

The branch&bound algorithm can be summarized as follows. The original problem DU-FLPU is first solved in the root node using the dual-based heuristic. If the solution calculated is not the optimal solution (or in cases where it is, but we cannot prove it because of a duality gap), the searching proceeds with a branch&bound scheme that guarantees that the optimal solution is found (if enough time and computational resources are available). The branching is based on those location decision variables that contribute to the complementary slackness violations of the current solution. After some tests, we decided to follow a simple rule and choose the first location variable found that contributes to these violations. Other rules were tested (taking into account the fixed facility costs, expected gains/losses in terms of assignment costs in choosing a second–best source instead of selecting the best source for a given customer), but no significant improvements were observed, especially in large sized problems. Inspired on previous works (Erlenkotter [29], Van Roy and Erlenkotter [88] and Dias et al. [21]), location variables are fixed first to zero and then to one. The tree is searched using a depth search procedure. Setting a variable to one is achieved by changing the corresponding fixed cost to zero. To use the current dual solution in the next branch&bound tree node, some changes may have to be made to guarantee dual admissibility (some dual variables must be reduced, with a corresponding increase in some of the slacks). When fixing a variable to zero, its fixed cost is set equal to $+\infty$, guaranteeing the admissibility of the current dual solution that will be used in the next tree node. A node is fathomed only if the current problem is infeasible, the optimal solution of the current problem has been found or the current dual objective function value is worse than the best primal objective function value found so far.

The computational results are provided in subsection 4.2.2.

3.3 Lagrangean relaxation approach

To be able to formulate and solve the problem α-DUFLPU (section 2.2), it is necessary to calculate the optimal solution ζ_s^* for each scenario $s \in \mathcal{S}$. These (deterministic) $|\mathcal{S}|$ problems can be solved to optimality by the branch&bound procedure proposed earlier or by a general solver (CPLEX, for instance). Assume then that ζ_s^* is known and such that $\zeta_s^* > 0$, for all $s \in \mathcal{S}$.

The Lagrangean relaxation of problem α-DUFLPU, in short LR α-DUFLPU, with respect to the constraint set (2.2.3) can be defined through the introduction of the Lagrange multipliers $\lambda_s \geq 0, \forall s \in \mathcal{S}$. Each λ_s is associated with the corresponding constraint and brought into the objective function, as follows:

$$(LR\,\alpha\text{-DUFLPU}) \qquad \min \quad \sum_{t\in\mathcal{T}}\sum_{j\in J}\sum_{s\in\mathcal{S}} p^s f_{jt}^s x_{jt} + \sum_{s\in\mathcal{S}}\sum_{t\in\mathcal{T}}\sum_{i\in I}\sum_{j\in J} p^s c_{ijt}^s y_{ijt}^s + \sum_{s\in\mathcal{S}} \lambda_s \left(\sum_{t\in\mathcal{T}}\sum_{j\in J} f_{jt}^s x_{jt} + \sum_{t\in\mathcal{T}}\sum_{i\in I}\sum_{j\in J} c_{ijt}^s y_{ijt}^s - (1+\alpha)\zeta_s^* \right) \tag{3.3.1}$$

s. t.

(2.1.2)–(2.1.6).

The algorithm has been designed considering two well known results from Lagrangean Relaxation (e.g., Reeves [70], Guignard [31]) adapted for the present problem in the following proposition.

Proposition 3.3.1 *The optimal solution of LR α-DUFLPU, for $\lambda_s \geq 0, \forall s \in \mathcal{S}$, gives a lower bound to the optimal solution of the original problem α-DUFLPU. In addition,* *a solution of LR α-DUFLPU that satisfies also constraint set (2.2.3) provides an upper bound to the optimum of α-DUFLPU.*

We have decided to use the efficient primal–dual heuristic to solve problem LR α-DUFLPU. In order to apply the primal-dual heuristic to the present problem, the objective function (3.3.1) is rewritten as follows:

$$\sum_{t \in \mathcal{T}} \sum_{j \in J} \sum_{s \in \mathcal{S}} (p^s + \lambda_s) f_{jt}^s x_{jt} + \sum_{s \in \mathcal{S}} \sum_{t \in \mathcal{T}} \sum_{i \in I} \sum_{j \in J} (p^s + \lambda_s) c_{ijt}^s y_{ijt}^s. \tag{3.3.2}$$

Notice that constant $-\sum_{s \in \mathcal{S}} \lambda_s (1+\alpha)\zeta_s^*$ is not considered in (3.3.2), being only added to the final objective function value. Defining in (3.3.2) $\mathcal{F}_{jt}^s = (p^s + \lambda_s) f_{jt}^s$ and $\mathcal{C}_{ijt}^s = (p^s + \lambda_s) c_{ijt}^s$, the formulations already presented for the DUFLPU in subsection 2.1.1, for the dual problem, the condensed dual problem, as well as the complementary slackness conditions between dual and primal problems are still valid for the LR α-DUFLPU. Hence, LR α-DUFLPU can be solved by the primal–dual heuristic presented in section 3.1. Recall that the heuristic's procedures are designed to reduce progressively the duality gap between dual and primal objective function values. Even if the heuristic is unable to find the optimal solution of LR α-DUFLPU, it is still able to provide a good lower bound to the optimal objective function value of α-DUFLPU, in this case through the dual objective function value as stated in the next proposition.

Proposition 3.3.2 *The best dual solution calculated by the primal–dual heuristic applied to LR α-DUFLPU provides a lower bound to the optimal objective function value of α-DUFLPU.*

Proof: Let us represent the optimum of α-DUFLPU by $\mathcal{O}pt(\alpha$-DUFLPU) and the optimum of LR α-DUFLPU by $\mathcal{O}pt(LR\,\alpha$-DUFLPU). In addition, let $(z_P,\ z_D)$ be the primal and dual solutions calculated by the primal–dual heuristic for LR α-DUFLPU and its dual, respectively. If $z_P = z_D$, then $z_P = \mathcal{O}pt(LR\,\alpha$-DUFLPU) which provides a lower bound to $\mathcal{O}pt(\alpha$-DUFLPU) (proposition 3.3.1). If the heuristic's solutions are such that $z_D < z_P$, then, from duality theory, we know that $z_D < \mathcal{O}pt(LR\,\alpha$-DUFLPU) $\leq$ $\mathcal{O}pt(\alpha$-DUFLPU), so z_D is a valid lower bound to $\mathcal{O}pt(\alpha$-DUFLPU).

Let us now turn to the generation of upper bounds. Taking into account the objective function (2.1.1) and the set of constraints (2.2.3), it is trivial to prove that the objective function value of α-DUFLPU is bounded above by $\sum_s p^s(1+\alpha)\zeta_s^*$. This value can then be considered as a first upper bound to the optimum of α-DUFLPU. Furthermore, if a lower bound calculated at any iteration is greater than this value, then α-DUFLPU is infeasible.

The primal solution calculated by the heuristic can be admissible or not for α-DUFLPU. If it is admissible, then it represents an upper bound to the optimal solution of α-DUFLPU.

After executing the primal-dual heuristic to LR α-DUFLPU, a local search procedure is performed. This local search procedure will explore the neighborhood of the current solution, trying to reach feasibility or trying to improve the objective function value (reaching better upper bounds). The neighborhood is considered to be the set of solutions that are equal to the current one with the exception of the opening time period of one facility. The local search procedure tries to change the time period when a given facility is opened, or tries not to open the facility at all. Whenever a better solution is found, it becomes the current solution and the local search continues until it is not possible to find better solutions in the neighborhood of the current solution.

A standard subgradient algorithm is used to update the Lagrange multipliers. Let us define subgradients G_s for the relaxed constraints, evaluated at the current solution, by:

$$G_s = \sum_{t \in \mathcal{T}} \sum_{j \in J} f_{jt}^s x_{jt} + \sum_{t \in \mathcal{T}} \sum_{i \in I} \sum_{j \in J} c_{ijt}^s y_{ijt}^s - (1+\alpha)\zeta_s^*, \quad \forall s \in \mathcal{S}.$$

In addition, let π represent the step size for the Lagrange multipliers and z the step size coefficients for the Lagrange multipliers.

Initially, in iteration $k = 0$, $\lambda_s^{(k)} = 0, \forall s \in \mathcal{S}$,

and in iteration $k > 0$,

$\lambda_s^{(k+1)} = \max\{0, \lambda_s^{(k)} + \pi G_s\}$, with $\pi = z \dfrac{UB^{(k)} - LB^{(k)}}{\sum_s G_s^2}$,

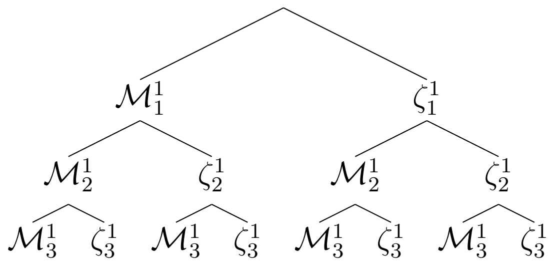
where $UB^{(k)}$ and $LB^{(k)}$ are the most recent upper and lower bounds achieved.

During the execution of the algorithm, the best upper and lower bounds achieved are updated and recorded, in order to calculate the solution gap, which is one of the established stopping criteria. The stopping criteria as other details of the algorithm will be discussed further in subsection 4.2.3.

3.4 Multi–objective approach

We will explain in this section a procedure to tackle the MODUFLPU. As stated in section 2.3, the knowledge of non–dominated solutions to the original MODUFLPU is achieved by solving the auxiliary problem(s) AUX. In an interactive approach, the dialogue phase with the DM consists in defining new values to the righthand side of constraints (2.3.3), the $\mathcal{M}_s$ values. These values will then define the regions of search. In a generating approach, $\mathcal{M}_s$ values can be automatically generated in a way that guarantees that the whole objective space is explored. The automatic generation of vector $\mathcal{M}$ can be done resorting to two simple data structures: a binary tree, with as much levels as the number of scenarios, and a matrix. Each time a new solution is calculated, based on a given vector $\mathcal{M}$, a binary tree is generated such that it will define all possible future vectors $\mathcal{M}$. These vectors are then recorded in a matrix so that they can be retrieved in future iterations. To give a simple example of this procedure, consider a problem with three scenarios. The initial vector $\mathcal{M}$ is set to $(\mathcal{M}_1^1, \mathcal{M}_2^1, \mathcal{M}_3^1)$. Solving AUX with this vector, assume that the non-dominated solution $(\zeta_1^1, \zeta_2^1, \zeta_3^1)$ is obtained, where $\zeta_1^1 \leq \mathcal{M}_1^1, \zeta_2^1 \leq \mathcal{M}_2^1, \zeta_3^1 \leq \mathcal{M}_3^1$ taking into account constraints (2.3.3). Based on both the given vector $\mathcal{M}$ and the achieved solution, a binary tree can be built as shown in figure 3.4.1.

Figure 3.4.1: Binary tree for automatic generation of vector $\mathcal{M}$.



The path from the root to each node of the tree will define a possible new future vector $\mathcal{M}$. In the present example, eight vectors are defined, $(\mathcal{M}_1^1, \mathcal{M}_2^1, \zeta_3^1)$, $(\mathcal{M}_1^1, \zeta_2^1, \mathcal{M}_3^1)$ or $(\mathcal{M}_1^1, \zeta_2^1, \zeta_3^1)$ for example, corresponding to eight possible search regions. These vectors can be stored in a matrix, so that they can be retrieved in a future iteration of the algorithm. Whenever a new solution is calculated, a new binary tree is built and the corresponding vectors added to the matrix. Note, however, that to some of these vectors will correspond infeasible problems and thus should not be recorded and used. For instance, $(\zeta_1^1, \zeta_2^1, \zeta_3^1)$ will not be interesting because it corresponds to an infeasible problem (otherwise $(\zeta_1^1, \zeta_2^1, \zeta_3^1)$ would not be a non-dominated solution). Other vectors will end up with optimal solutions that are already known such as $(\mathcal{M}_1^1, \mathcal{M}_2^1, \mathcal{M}_3^1)$ for instance. Furthermore, knowing that one given problem is impossible will allow us to conclude that other $\mathcal{M}$ vectors will also lead to impossible problems and then it is not worth to explore the corresponding region. This search method is easily implementable and will guarantee that the whole objective space is explored.

Let us now turn to the choice of the vector of weights ν in order to define the objective function of AUX. As noted before, these weights can and should be changed in accordance with vector $\mathcal{M}$ in order to help decreasing the computational time needed to calculate a solution ([20]). For instance, if $\mathcal{M}$ is more demanding for a given scenario, meaning that $\mathcal{M}_s$ is close to the best objective function value ζ_s^*, then the respective objective function weight should be increased. One simple way of doing this is setting ν as follows:

$$\nu_s = 1 - \frac{\mathcal{M}_s - \zeta_s^*}{\zeta_s^*}, \forall s \in \mathcal{S}, \tag{3.4.1}$$

$$\nu_s = \frac{\nu_s}{\sum_s \nu_s}, \forall s \in \mathcal{S}.$$

We next illustrate a solution approach to the MODUFLPU by one small problem, following an interactive procedure based on Dias et al. [20] where all AUX problem instances were solved by CPLEX v12.6 .

Example 3.4.1 *Consider a problem instance with* 25 *potential facility sites,* 100 *possible customers,* 10 *time periods and* 2 *scenarios.*

Initially, and in order to delineate the region of interest, the solutions with the best possible objective function value for each scenario should be calculated. These solutions can be achieved considering in AUX binary vectors ν and large values to $\mathcal{M}$. The solutions obtained for the present problem are depicted in the objective space in figure 3.4.2: (138023, 153313) with the optimum cost of scenario 1 and (218195, 139854) with the optimum of scenario 2. The DM is now free to set the vector $\mathcal{M}$. Let us assume that he does not want to explore any particular region, so he decides to consider ($\mathcal{M}_1$, $\mathcal{M}_2$) = (218195, 153313) based on the two non–dominated solutions already calculated. With weights $(\nu_1, \nu_2) = (0.32, 0.68)$, calculated according to (3.4.1), the new solution reached is (138902, 142526) (figure 3.4.3).

Figure 3.4.2: Solutions with the optimum of each scenario.

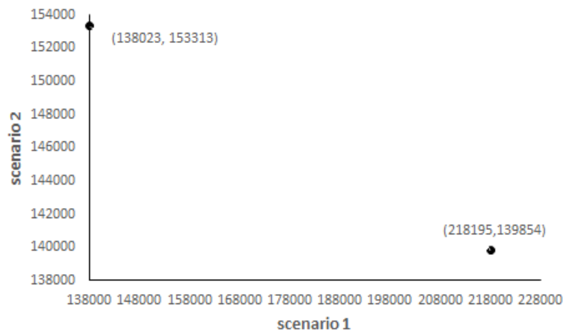
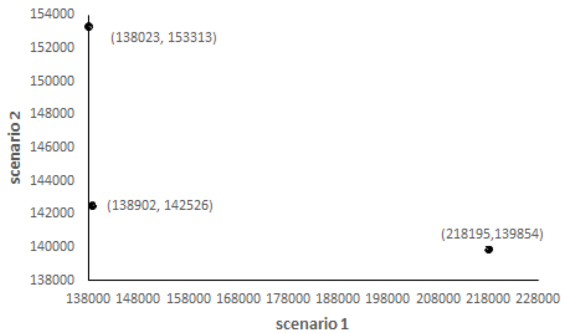


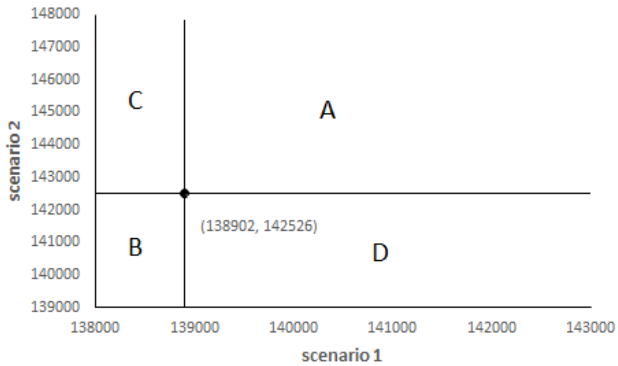
Figure 3.4.3: The first non–dominated solution calculated.



Considering the newly calculated non-dominated solution, it is easy to see that two regions of the objective space are no longer of interest. This is illustrated in figure 3.4.4: as region A has only solutions that are dominated by the solution calculated, region B has only non-admissible solutions.

The DM can then decide whether to explore region C or region D. Let us assume that he would explore region D. Then $\mathcal{M}_1$ will remain equal to 218195 and $\mathcal{M}_2$ will be set to 142526 (given by the new non–dominated solution just calculated). Figure 3.4.5 shows the new solution calculated, (141836, 141936). The procedure would be repeated until the DM is satisfied or the whole objective space has been explored. The whole set of non–dominated solutions found for this problem is depicted in figure 3.4.6. It is possible

Figure 3.4.4: Regions A and B discarded from further searches.



to observe the compromises that exist between the two scenarios. The location decisions in each of the non–dominated solutions, which facilities are to be opened and when, are detailed in table 3.4.1. We can observe that a set of seven facilities is opened exactly in the same time period in all solutions calculated.

Figure 3.4.5: A new non–dominated solution.

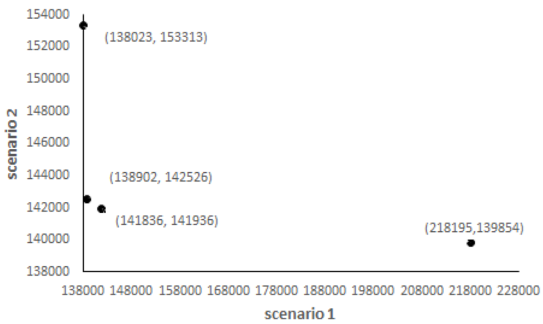


Figure 3.4.6: The set of non–dominated solutions.

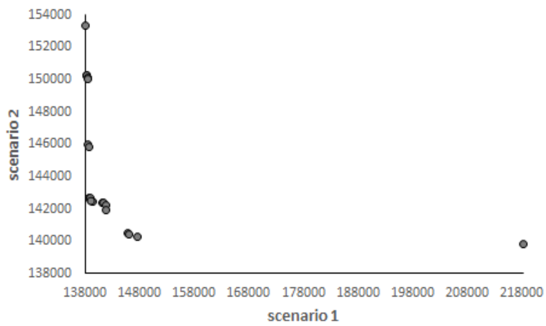


Table 3.4.1: Example 3.4.1: Time period in which each facility is opened.

		Opened Facilities														
ζ_1	ζ_2	2	3	4	5	6	7	9	11	14	16	18	20	22	24	25
138023	153313	1	1	–	7	6	4	2	2	3	2	4	6	3	2	–
138228	150276	1	1	–	7	6	4	2	2	3	2	4	6	3	2	1
138237	150257	1	1	–	7	6	4	2	2	3	3	4	6	3	2	1
138360	150238	1	1	–	7	6	4	2	2	3	3	4	–	3	2	1
138384	150093	1	1	–	–	6	4	2	2	3	2	4	6	3	2	1
138393	150074	1	1	–	–	6	4	2	2	3	3	4	6	3	2	1
138564	145957	1	1	–	7	6	4	2	5	3	2	4	6	3	2	–
138720	145827	1	1	–	–	6	4	2	5	3	2	4	6	3	2	–
138746	142709	1	1	–	7	6	4	2	5	3	2	4	6	3	2	1
138869	142690	1	1	–	7	6	4	2	5	3	2	4	–	3	2	1
138902	142526	1	1	–	–	6	4	2	5	3	2	4	6	3	2	1
139281	142430	1	1	7	–	6	4	2	5	3	2	4	6	3	2	1
141238	142389	–	1	–	7	6	4	2	5	3	2	4	6	3	2	1
141457	142365	–	1	–	–	6	4	2	5	3	2	4	6	3	2	1
141695	142200	–	1	7	–	6	4	2	5	3	2	1	6	3	2	1
141836	141936	–	1	7	–	6	4	2	5	3	2	4	6	3	2	1
145742	140500	1	1	–	–	6	4	2	5	3	2	4	2	3	2	1
146121	140404	1	1	7	–	6	4	2	5	3	2	4	2	3	2	1
147507	140307	–	1	–	7	6	4	2	5	3	2	4	2	3	2	1
218195	139854	–	1	7	–	6	4	2	5	3	2	4	2	3	2	1

Chapter 4

Computational Experiments

The algorithms developed to tackle the DUFLPU, the primal–dual heuristic approach (section 3.1) and the branch&bound approach (section 3.2), as well as the Lagrangean relaxation procedure developed to solve α-DUFLPU (section 3.3), have been tested over sets of different problem instances. As we are not aware of the existence of benchmark problem instances that could be easily adapted to conform to the presented models, we have chosen to randomly generate problem instances. It should be pointed out that the generation of the data to a decision model under uncertainty is in itself an active area of research, mainly in what concerns stochastic programming models (see, for instance, Dupacova [25], Dupacova et al. [26], Kaut and Wallace [43], Heitsch and Romisch [35]). Scenario based stochastic programs, in which the true underlying probability distributions are replaced by discrete distributions concentrated in a finite number of points (scenarios), or sequence of events, with probabilities, often require a specific form of the input (as multistage problems require scenario trees for example). The variety of methods for generating scenarios available in the literature is thus significant: sampling and sampling-based methods, moment matching, path-based methods which generate complete paths/scenarios, optimal discretization, etc. These methods depend on the decision model, level of knowledge about the underlying probability distributions or stochastic processes, availability of historical data, opinion of experts, etc. The total number of scenarios generated by some of these methods is too large and thus with higher computational difficulties. To overcome such difficulties, there are also methods for reducing the total number of scenarios (for details see the works cited above and the references therein for example).

There are possibly many ways in which one could generate the scenarios for the proposed models. In a real-world setting such scenarios may be advanced by experts for example. The purpose of the algorithm that has been developed for the generation of test problems (described in section 4.1) is only to create input data to the models, in a simple, understandable and fast manner, in order to make possible the realization of the tests. Herein, the generated scenarios are some kind of "what if" scenarios. As we are in the presence of a dynamic problem under uncertainty, data must change simultaneously over time and among the different scenarios. Furthermore, we considered different dimensions for the test problems, by varying the number S of scenarios, number T of time periods, number M of possible facility locations and number N of possible customers. Our purpose was first to evaluate the quality of the solutions achieved by the developed algorithms in terms of gap, given by the difference between the best objective function value found by each algorithm and the best known lower bound on the optimal value divided by this best known lower bound. We also analyzed the algorithms in terms of the computational time spent on the searching process. Even though we are dealing with strategic decisions, where time usually is not determinant, faster algorithms permit the consideration of larger and diverse problems, enriching the decision making process. For α-DUFLPU in particular, being able to solve it for several different values of maximum regret will allow the DM to get a better picture of the compromises that exist. However, it is desirable that this process takes place within a reasonable computational time. The results obtained by general solvers considering the same sets of problems are also presented.

4.1 Generation of test problems

The algorithm that was developed for the generation of test problems can be summarized as follows. First, the network of the problem is randomly generated, including the location of the nodes (potencial facility sites and possible customer locations) and arcs between them. This network will be valid for all time periods and scenarios. Then, we consider the generation of the data for all time periods of scenario 1: arc costs, consequently assignment costs, set of potential facility sites and corresponding fixed costs, and set of customer locations. Scenario 1 is called the basic scenario as it is from this scenario that all the others will be constructed. Thus, for the other scenarios, for the first time period we consider the data generated for the basic scenario (the first time period corresponds to the present situation that is not scenario dependent), as for each one of the other periods of time the data may change with some probability. For the sake of simplicity, these input probabilities are only dependent of the scenarios but these could be also dependent of other items such as periods of time, arcs, facility or customer locations. This is a very important feature of the procedure, since it will allow the generation of problems well distinct. As far as scenario probabilities (p^s) are concerned, these were randomly generated such that the sum of all probabilities is equal to 1. Below we provide the approach used in the generation of all test problems (in general). Table 4.1.1 presents some input values that were considered and that must be known before the generation procedure. For ease in the exposition, let us first consider the following additional notation:

J_t^s : Set of potencial facility locations that can be selected (opened) at the beginning of time period $t \in \mathcal{T}$ for scenario $s \in \mathcal{S}$,

I_t^s: Set of customer locations with demand during period $t \in \mathcal{T}$ for scenario $s \in \mathcal{S}$, where $J_t^s \subseteq J$ and $I_t^s \subseteq I$.

Table 4.1.1: Input values.

$MaxX$	1000
$MaxY$	1000
p_{arc}	0.75
d	50
p_{arcc}	0.80
p_f^s	0.80 for $s = 1$ and 0.5 $\forall s \neq 1$
p_c^s	0.80 for $s = 1$ and 0.3 $\forall s \neq 1$
p_c	0.10
p_a^s	0.40
p_{cf}^s	0.60

Data generation steps

1. Random generation of $(x, y)-$coordinates in a rectangular area of size $MaxX \times MaxY$ corresponding to the location of $|J| + |I|$ nodes (potencial facility sites plus possible customer locations).

2. Random generation of arcs between the network nodes with probability p_{arc}; afterwards, if there isn't an arc between two nodes "close" (the Euclidean distance between them is less than d), an arc is created between them with probability $p_{arcc} > p_{arc}$.

3. For $s = 1$ (basic scenario):

    3.1 for $t = 1$: random generation of costs associated with arcs, according to a Uniform distribution $\mathcal{U}[lc, uc]$;

    for each $t \geq 2$, each arc cost is equal to the cost generated in period $t - 1$ plus a changing factor randomly generated.

    3.2 for each $t \geq 1$:

    i. calculation of the shortest path between each possible customer location and each potential facility location.

    ii. random generation of set J_t^1, with $J_1^1 \neq \emptyset$, and fixed costs:
    each location j is included in J_t^1 with probability p_f^1;

– if $j \in J_t^1$, then the fixed cost at j is randomly generated from a Uniform distribution $\mathcal{U}[lf, uf]$, and for each $\tau > t$ the fixed cost is increased by a changing factor randomly generated;

– if $j \notin J_t^1$, then the fixed cost at j is set to $+\infty$.

iii. random generation of set I_t^1: each customer i is included in I_t^1 with probability p_c^1; in addition, for $t \geq 3$, if i was included in I_{t-2}^1 and excluded from I_{t-1}^1, then i is included in I_t^1 with probability $p_c < 0.5$.

4. For $s \neq 1$ (other scenarios):

4.1 for $t = 1$, consider the data generated for the basic scenario and $t = 1$.

4.1 for each $t \geq 2$:

i. each arc cost that was generated for time period t of the basic scenario (basic cost) changes in time period t of scenario s with probability p_a^s; if a variation occurs, then the arc cost is equal to the basic cost plus a changing factor Θ_a randomly generated.

ii. calculation of the shortest path between each possible customer location and each potential facility location.

iii. random generation of set J_t^s and fixed costs:
each location j is included in J_t^s with probability p_f^s;

– if $j \in J_t^s \cap J_t^1$, then the fixed cost at j that was generated for time period t of the basic scenario (basic cost) changes in time period t of scenario s with probability p_{cf}^s; if a variation occurs, then the fixed cost is equal to the basic cost plus a changing factor Θ_f randomly generated;

– if $j \in J_t^s$ but $j \notin J_t^1$, then the fixed cost at j is randomly generated from a Uniform distribution $\mathcal{U}[lf, uf]$, and for each $\tau > t$ the fixed cost is increased by a changing factor randomly generated;

– if $j \notin J_t^s$, then fixed cost at j is set to $+\infty$.

iv. random generation of set I_t^s: the demand state of customer i that was generated for time period t of the basic scenario changes in time period t of scenario s with probability p_c^s.

4.2 Computational results

The computational results obtained are presented in the next subsections. The algorithms were all coded in C–language and the computational experiments were carried out on a AMD Turion(tm) X2 Dual–Core Mobile RM–70 processor at 2.00 GHz with 3.00 GB of

RAM. Gap is given in percentage and the computational time in seconds. The time results do not include the time required to read the problems' data, only the time to solve them. The general solver used to make comparisons with the primal–dual heuristic is LPSolve v5.5.2.0 [9]. Afterwards, thanks to IBM Academic Initiative, the results refer to CPLEX MIP optimizer, v12.4.

4.2.1 Primal-Dual heuristic

The input values of (S, T, M, N) used in the random generation of the test problems are given in Table 4.2.1. For each combination of (S, T, M, N), with $N > M$, five instances were randomly generated. Different random seeds were used for each of the instances. We have, in total, 780 instances, that were solved by the heuristic and by LpSolve. We decided to stop the solver if its solution time exceeded 7200 seconds (s). We note that the smallest instance considered has 1025 variables with 1205 constraints but the largest has 3000750 variables with 3060050 constraints.

Table 4.2.1: Parameters used in the random generation of the test problems.

S	2	5	10	20
T	5	10	15	—
M	5	10	20	50
N	20	50	100	200

In Tables 4.2.2–4.2.5 we summarize the computational results obtained. Each table corresponds to a given number of scenarios. We report the minimum and maximum number of opened facilities (dimension of the set J^+) as well as the minimum, average and maximum gap (in percentage) on the five instances solved for each combination of (T, M, N). The following tables also show the solution times (in seconds) of the heuristic and the solver. We report the minimum, average and maximum time spent by the heuristic and by the solver to solve each group of five instances. The primal–dual heuristic was able to solve all the 780 instances. As far as the solver results are concerned, the solver could not solve some of the five instances, due to lack of memory to read the problem or the execution time has exceeded 7200 s. We report these cases and statistics refer only to those instances that were solved. Whenever the solver was not able to solve any of the five instances, the solver time is given as '*'. Only on the larger instances, with $(S, T, M, N) = (20, 15, 50, 200)$, the heuristic exceeded the time limit established a priori. In terms of solution quality, the worst gap, 4.02%, was observed with instances with 20 scenarios and with $T = 15$, $M = 50$ and $N = 100$. Within each S-scenario problems, in average, the larger gaps were observed in instances with largest M and N.

The average results for all S–scenario problems are reported in the last row of the corresponding tables. We can see that the number of scenarios considered do not result

in markedly different solution qualities. However, the execution times required by the solver are clearly higher than those required by the heuristic, especially for large sized problems. In most of the test problems with large dimensions the solver could not solve them in less than $7200\,s$. The heuristic time can vary a lot, even for problems with the same size. For example, for instances with $(S,T,M,N) = (10,15,20,200)$ the execution time ranges from 0.28 to $1231.29\,s$, in average 508.18 seconds.
The computational results show that the heuristic is capable of finding very good quality solutions in reasonable computational times, clearly outperforming the general solver.

As it is well known, when solving integer programming problems general solvers tend to reach a good admissible (sometimes optimal) solution fast, and then spend a lot of time trying to improve this solution or proving that the solution is optimal. So comparing the computational time of a dedicated heuristic to that of a general solver can be seen as unfair to the general solver. That is why we have repeated all the computational tests but now using the general solver as an heuristic procedure: for each set of instances, we have limited the maximum computational time spent by the general solver considering this maximum time equal to the maximum time spent by the heuristic and then compare the quality of the solutions found by the two approaches. When this time limit was considered, and for all test problems, the solver was not able to find any admissible solution (upper and lower bounds of the optimal primal objective function value were equal to '$+\infty$' and '$-\infty$', respectively). It should be noted that the minimum times presented by the solver (see Tables 4.2.2–4.2.5) are greater than the maximum times spent by the heuristic to compute the solution for the same problems.

Table 4.2.2: Computational results for 2–scenario problems.

T	M	N	$\|J^+\|$		gap (%)			Heur. time (s)			Solver time (s)		
			min	max	min	aver	max	min	aver	max	min	aver	max
5	5	20	2	4	0.00	0.00	0.00	0.00	0.01	0.02	0.08	0.11	0.16
5	5	50	2	4	0.00	0.00	0.00	0.00	0.00	0.00	0.42	0.64	0.83
5	5	100	4	5	0.00	0.00	0.00	0.00	0.00	0.00	2.23	2.64	3.32
5	5	200	5	5	0.00	0.00	0.00	0.00	0.04	0.09	7.85	8.62	9.91
5	10	20	3	4	0.00	0.11	0.44	0.00	0.02	0.06	0.19	0.38	0.53
5	10	50	5	7	0.00	0.13	0.36	0.00	0.06	0.17	1.48	2.33	3.42
5	10	100	5	7	0.00	0.03	0.14	0.00	0.08	0.27	5.51	7.36	8.81
5	10	200	7	9	0.00	0.00	0.00	0.00	0.37	1.81	18.24	25.13	31.51
5	20	50	5	9	0.00	0.41	1.52	0.05	0.13	0.30	4.56	6.57	9.66
5	20	100	8	10	0.00	0.05	0.15	0.03	0.83	1.51	20.65	23.25	27.02
5	20	200	10	13	0.00	0.01	0.04	0.02	3.24	12.29	74.54	101.52	121.56
5	50	100	13	16	0.19	0.64	1.85	0.48	3.23	5.13	75.04	169.58	264.31
5	50	200	18	22	0.11	0.33	0.67	6.29	13.41	19.44	391.73	471.97	620.62
10	5	20	3	4	0.00	0.00	0.00	0.00	0.00	0.02	0.30	0.37	0.45
10	5	50	4	5	0.00	0.00	0.00	0.00	0.00	0.02	1.79	2.43	3.03
10	5	100	5	5	0.00	0.00	0.00	0.00	0.03	0.11	8.14	8.53	9.24
10	5	200	5	5	0.00	0.00	0.00	0.02	0.03	0.08	24.16	31.05	43.01
10	10	20	3	6	0.00	0.03	0.12	0.00	0.01	0.02	0.86	1.34	1.89
10	10	50	6	8	0.00	0.00	0.00	0.00	0.01	0.03	4.62	5.61	7.27
10	10	100	7	10	0.00	0.00	0.00	0.00	0.01	0.02	16.91	19.99	21.40
10	10	200	9	10	0.00	0.00	0.00	0.00	0.02	0.08	72.24	87.83	109.93
10	20	50	8	12	0.00	0.06	0.32	0.08	0.58	1.25	13.43	23.38	33.29
10	20	100	11	15	0.00	0.04	0.20	0.11	0.98	2.26	71.04	82.84	101.03
10	20	200	16	19	0.00	0.01	0.06	0.09	1.72	6.77	233.77	270.59	361.19
10	50	100	19	23	0.37	1.08	2.39	1.95	6.33	11.25	398.89	546.24	746.12
10	50	200	26	30	0.19	0.35	0.61	40.17	52.61	90.46	1510.53	1737.18	1880.55
15	5	20	3	5	0.00	0.00	0.00	0.00	0.01	0.06	0.70	0.91	1.28
15	5	50	4	5	0.00	0.00	0.00	0.00	0.00	0.00	4.09	5.09	6.19
15	5	100	5	5	0.00	0.00	0.00	0.00	0.04	0.09	16.65	19.49	22.07
15	5	200	5	5	0.00	0.00	0.00	0.00	0.34	1.64	71.09	80.89	91.23
15	10	20	4	7	0.00	0.00	0.00	0.00	0.00	0.02	1.72	2.70	3.67
15	10	50	7	9	0.00	0.00	0.00	0.00	0.01	0.03	11.00	12.75	14.56
15	10	100	8	10	0.00	0.00	0.00	0.00	0.03	0.11	37.30	49.85	67.16
15	10	200	10	10	0.00	0.00	0.00	0.02	0.02	0.02	155.06	215.19	247.49
15	20	50	9	12	0.00	0.03	0.13	0.31	1.02	1.97	47.00	54.76	71.79
15	20	100	14	16	0.00	0.07	0.21	0.02	1.62	7.47	114.54	168.61	217.79
15	20	200	17	20	0.00	0.00	0.00	0.27	1.16	3.23	620.72	696.22	878.47
15	50	100	23	28	0.35	0.76	1.31	2.39	5.40	9.50	1064.62	1768.31	2946.97
15	50	200[a]	32	37	0.00	0.52	2.28	58.62	106.2	210.2	2699.81	3370.90	3957.47
	Aver				**0.03**	**0.12**	**0.33**	**2.84**	**5.12**	**9.94**	**200.09**	**258.54**	**331.95**

[a] Solver was unable to solve one of the instances with $T = 15$, $M = 50$ and $N = 200$.

Table 4.2.3: Computational results for 5–scenario problems.

T	M	N	$\|J^+\|$		gap (%)			Heur. time (s)			Solver time (s)		
			min	max	min	aver	max	min	aver	max	min	aver	max
5	5	20	1	2	0.00	0.00	0.00	0.00	0.00	0.02	0.47	0.51	0.56
5	5	50	2	3	0.00	0.00	0.00	0.00	0.01	0.02	3.42	4.29	5.54
5	5	100	4	4	0.00	0.00	0.00	0.00	0.01	0.02	10.48	14.21	18.70
5	5	200	4	5	0.00	0.00	0.00	0.00	0.01	0.03	38.05	51.95	61.87
5	10	20	2	3	0.00	0.00	0.00	0.00	0.02	0.06	1.50	2.06	3.42
5	10	50	3	5	0.00	0.07	0.36	0.00	0.19	0.55	8.80	11.93	17.44
5	10	100	5	7	0.00	0.00	0.00	0.00	2.63	10.19	35.65	41.94	53.42
5	10	200	7	8	0.00	0.00	0.00	0.00	0.24	0.66	138.92	176.28	204.44
5	20	50	5	6	0.00	0.39	1.41	0.08	0.74	1.89	33.79	51.27	66.67
5	20	100	7	8	0.00	0.19	0.56	0.02	5.38	10.78	93.54	184.33	240.07
5	20	200	9	13	0.00	0.08	0.26	2.14	34.26	52.57	602.52	840.71	1084.33
5	50	100	10	12	0.00	0.15	0.49	4.57	14.99	23.76	687.40	984.75	1292.27
5	50	200	14	18	0.16	0.24	0.34	49.97	94.49	188.82	3258.87	4243.81	5243.82
10	5	20	2	4	0.00	0.06	0.29	0.00	0.04	0.20	2.26	2.50	3.00
10	5	50	4	5	0.00	0.06	0.31	0.00	0.14	0.50	10.95	15.89	21.92
10	5	100	4	5	0.00	0.00	0.00	0.02	0.02	0.03	44.06	46.75	51.28
10	5	200	5	5	0.00	0.00	0.00	0.02	0.03	0.05	201.49	226.94	273.97
10	10	20	3	4	0.00	0.29	1.46	0.00	0.31	1.11	6.68	9.70	11.59
10	10	50	4	7	0.00	0.10	0.33	0.00	0.86	3.48	36.16	51.28	65.13
10	10	100	7	8	0.00	0.00	0.00	0.02	0.17	0.56	154.46	185.67	238.81
10	10	200	9	10	0.00	0.00	0.00	0.03	0.04	0.05	364.87	566.97	853.41
10	20	50	7	9	0.00	0.25	0.57	1.45	4.93	8.81	128.76	205.82	276.32
10	20	100	8	13	0.00	0.02	0.12	0.27	9.71	27.44	489.92	688.11	914.27
10	20	200	13	18	0.00	0.01	0.01	2.18	19.64	68.11	1766.34	2640.57	3348.20
10	50	100	15	19	0.30	0.74	1.34	11.22	50.01	82.74	3048.36	4795.48	7152.59
10	50	200	20	24	0.83	1.05	1.40	210.62	344.60	432.31	*	*	*
15	5	20	2	4	0.00	0.00	0.00	0.00	0.01	0.02	5.65	5.99	6.13
15	5	50	4	5	0.00	0.00	0.00	0.02	0.02	0.03	29.97	33.62	41.12
15	5	100	4	5	0.00	0.00	0.00	0.03	0.07	0.19	107.89	126.81	140.43
15	5	200	5	5	0.00	0.00	0.00	0.05	0.06	0.09	493.69	554.62	653.95
15	10	20	3	5	0.00	0.00	0.00	0.22	0.68	1.95	15.91	18.10	20.97
15	10	50	6	7	0.00	0.00	0.00	0.00	0.04	0.11	96.13	124.75	148.18
15	10	100	8	9	0.00	0.01	0.04	0.03	1.97	8.94	444.77	489.07	561.88
15	10	200	10	10	0.00	0.00	0.00	0.06	0.32	1.36	1187.18	1471.82	1701.38
15	20	50	7	9	0.00	0.11	0.39	2.81	10.82	25.55	316.88	353.42	404.52
15	20	100	9	15	0.00	0.13	0.41	4.99	23.75	48.55	1043.98	1300.18	1491.25
15	20	200	14	18	0.00	0.01	0.03	1.75	53.01	156.41	4576.93	5245.83	6506.93
15	50	100[a]	17	24	0.68	1.47	2.72	23.43	60.40	120.53	6564.31	6882.34	7200.37
15	50	200	24	30	0.42	1.30	1.87	20.58	338.01	639.04	*	*	*
Aver					**0.06**	**0.17**	**0.38**	**8.63**	**27.50**	**49.17**	**690.82**	**882.44**	**1091.36**

[a] Solver was unable to solve three of the instances with $T = 15$, $M = 50$ and $N = 100$.

Table 4.2.4: Computational results for 10–scenario problems.

T	M	N	$\lvert J^{+}\rvert$		gap (%)			Heur. time (s)			Solver time (s)		
			min	max	min	aver	max	min	aver	max	min	aver	max
5	5	20	2	3	0.00	0.07	0.36	0.00	0.22	0.53	2.89	3.19	3.84
5	5	50	3	3	0.00	0.00	0.00	0.00	0.12	0.58	12.62	17.11	23.32
5	5	100	4	4	0.00	0.00	0.00	0.00	0.01	0.02	58.38	65.14	73.29
5	5	200	4	5	0.00	0.00	0.00	0.02	0.03	0.03	207.31	230.73	243.10
5	10	20	1	4	0.00	0.02	0.11	0.00	0.36	0.62	6.96	9.89	12.84
5	10	50	3	5	0.00	0.03	0.14	0.00	1.93	6.94	45.43	66.04	109.22
5	10	100	4	5	0.00	0.00	0.00	0.09	5.44	23.95	148.47	255.05	351.09
5	10	200	6	8	0.00	0.02	0.10	0.05	4.33	10.64	795.18	1038.60	1442.13
5	20	50	4	6	0.00	0.14	0.46	1.89	5.10	8.74	155.02	226.56	356.30
5	20	100	6	8	0.00	0.25	1.27	2.40	8.23	17.85	541.68	796.70	909.26
5	20	200	8	12	0.00	0.03	0.10	26.57	164.40	341.20	2121.63	2988.89	4074.88
5	50	100	8	12	0.00	0.17	0.57	34.94	79.29	121.93	3436.65	4215.17	5468.21
5	50	200	14	19	1.22	2.07	3.25	418.86	634.12	946.89	*	*	*
10	5	20	2	3	0.00	0.00	0.00	0.00	0.01	0.02	7.89	11.65	16.07
10	5	50	3	5	0.00	0.00	0.00	0.02	0.03	0.08	50.67	65.89	95.52
10	5	100	4	5	0.00	0.00	0.00	0.03	0.04	0.05	197.23	221.54	247.67
10	5	200	5	5	0.00	0.00	0.00	0.06	0.09	0.14	810.00	868.02	992.83
10	10	20	2	4	0.00	0.00	0.00	0.02	0.32	1.25	27.33	32.99	43.54
10	10	50	4	6	0.00	0.06	0.30	0.03	9.70	34.91	182.36	223.63	320.38
10	10	100	6	8	0.00	0.00	0.00	0.06	1.24	3.78	696.07	877.82	961.08
10	10	200	8	10	0.00	0.00	0.00	0.11	0.12	0.13	2308.36	2687.90	3046.52
10	20	50	6	9	0.00	0.33	0.97	6.51	27.74	49.41	584.74	954.05	1261.76
10	20	100	9	11	0.00	0.19	0.67	7.22	73.58	205.44	2551.49	3135.98	3598.62
10	20	200	13	15	0.00	0.04	0.12	1.79	243.75	460.86	*	*	*
10	50	100	13	17	0.62	1.66	2.27	73.26	225.40	334.34	*	*	*
10	50	200	18	25	0.50	1.23	2.30	1091.9	1871.4	2703.6	*	*	*
15	5	20	3	4	0.00	0.00	0.00	0.02	1.33	6.54	21.09	30.78	38.45
15	5	50	4	5	0.00	0.00	0.00	0.05	0.07	0.13	149.46	168.39	188.90
15	5	100	4	5	0.00	0.00	0.00	0.08	0.20	0.41	545.28	595.69	690.44
15	5	200	5	5	0.00	0.00	0.00	0.14	0.44	1.47	1953.11	2107.77	2261.94
15	10	20	3	5	0.00	0.07	0.37	0.28	1.62	4.68	65.30	88.62	126.95
15	10	50	4	7	0.00	0.00	0.00	0.05	1.25	3.65	447.81	497.29	550.57
15	10	100	7	9	0.00	0.00	0.00	0.11	10.99	41.96	1472.00	1997.70	2838.22
15	10	200[a]	9	10	0.00	0.00	0.00	0.17	0.21	0.23	5932.49	6218.71	6353.24
15	20	50	6	8	0.00	0.18	0.35	0.78	17.82	40.72	1374.77	1757.89	2792.24
15	20	100[b]	8	12	0.00	0.23	0.69	8.19	70.07	115.46	4948.38	5251.97	5518.58
15	20	200	14	18	0.00	0.02	0.05	0.28	508.18	1231.3	*	*	*
15	50	100	17	23	1.14	1.95	2.85	187.43	427.05	785.32	*	*	*
15	50	200	22	28	0.41	1.23	1.91	526.03	1771.8	3170.6	*	*	*
Aver					**0.10**	**0.26**	**0.49**	**61.27**	**158.15**	**273.75**	**995.56**	**1178.35**	**1406.59**

[a] Solver was unable to solve one of the instances with $T = 15$, $M = 10$ and $N = 200$. [b] Solver was unable to solve two of the instances with $T = 15$, $M = 20$ and $N = 100$.

Table 4.2.5: Computational results for 20–scenario problems.

T	M	N	$\lvert J^+\rvert$		gap (%)			Heur. time (s)			Solver time (s)		
			min	max	min	aver	max	min	aver	max	min	aver	max
5	5	20	1	3	0.00	0.01	0.07	0.00	0.13	0.61	9.42	12.28	17.64
5	5	50	3	4	0.00	0.00	0.00	0.02	0.71	3.45	61.84	70.29	95.61
5	5	100	3	4	0.00	0.00	0.00	0.03	30.86	154.19	222.89	286.07	381.81
5	5	200	4	5	0.00	0.00	0.00	0.06	0.08	0.09	1001.93	1109.75	1302.44
5	10	20	2	4	0.00	0.22	1.08	0.02	2.53	4.96	37.78	45.14	57.60
5	10	50	3	4	0.00	0.08	0.39	1.26	14.10	23.99	257.40	291.35	309.33
5	10	100	5	6	0.00	0.00	0.00	3.09	71.90	245.59	877.80	1069.26	1361.01
5	10	200	6	8	0.00	0.03	0.15	0.09	145.20	638.04	2729.54	3547.23	4662.37
5	20	50	4	6	0.00	0.13	0.63	0.36	21.11	44.29	499.04	993.34	1600.09
5	20	100	6	8	0.00	0.19	0.82	19.64	101.85	222.91	2429.08	3547.42	4711.87
5	20	200	9	13	0.01	0.57	1.47	47.05	621.55	1342.97	*	*	*
5	50	100	8	14	1.28	2.01	2.75	202.60	310.82	359.07	*	*	*
5	50	200	16	20	1.85	2.47	3.33	383.79	1812.6	2803.42	*	*	*
10	5	20	2	3	0.00	0.00	0.00	0.03	0.55	1.78	34.05	47.44	67.78
10	5	50	3	4	0.00	0.00	0.00	0.06	0.07	0.09	256.78	266.85	277.40
10	5	100	4	5	0.00	0.00	0.00	0.13	0.15	0.19	1023.39	1138.34	1499.16
10	5	200	5	5	0.00	0.00	0.00	0.23	0.27	0.33	3540.65	3699.76	4046.42
10	10	20	2	4	0.00	0.23	1.16	0.05	2.61	6.44	128.87	157.65	201.02
10	10	50	4	5	0.00	0.00	0.00	0.08	23.42	45.74	791.93	949.63	1187.52
10	10	100	6	7	0.00	0.00	0.00	0.16	8.25	38.05	2891.10	3888.13	4534.98
10	10	200	9	10	0.00	0.00	0.00	0.30	1.62	3.76	*	*	*
10	20	50	5	8	0.00	0.12	0.56	34.94	140.44	225.34	2789.70	3035.98	3677.72
10	20	100	8	11	0.02	0.43	1.03	53.57	193.76	409.70	*	*	*
10	20	200	13	14	0.01	0.16	0.38	689.88	1786.9	3325.52	*	*	*
10	50	100	13	16	0.68	1.82	3.55	215.51	748.85	1289.9	*	*	*
10	50	200	18	21	0.62	1.10	2.25	1860.6	3639.3	4646.8	*	*	*
15	5	20	2	3	0.00	0.00	0.00	0.06	9.97	49.55	107.58	123.71	157.44
15	5	50	3	5	0.00	0.00	0.00	0.14	0.17	0.19	537.65	648.65	858.02
15	5	100	4	5	0.00	0.00	0.00	0.28	0.42	0.86	2195.15	2440.06	2643.19
15	5	200	5	5	0.00	0.00	0.00	0.50	0.57	0.70	*	*	*
15	10	20	2	5	0.00	0.14	0.72	0.06	12.31	43.73	296.65	414.25	614.06
15	10	50	5	6	0.00	0.00	0.00	0.17	79.82	297.60	1564.88	2319.46	2902.85
15	10	100	7	10	0.00	0.02	0.10	0.34	8.53	37.82	*	*	*
15	10	200	9	10	0.00	0.00	0.00	0.61	0.65	0.73	*	*	*
15	20	50	6	9	0.05	0.70	2.32	49.64	198.52	353.08	*	*	*
15	20	100	11	13	0.02	0.34	0.49	109.61	435.24	641.11	*	*	*
15	20	200	10	15	0.00	0.08	0.32	403.70	2561.1	5787.24	*	*	*
15	50	100	16	18	0.96	2.49	4.02	414.34	1973.1	3138.74	*	*	*
15	50	200	22	26	1.35	1.89	3.01	6442.4	13133.7	16098.22	*	*	*
Aver					**0.18**	**0.39**	**0.78**	**280.40**	**720.35**	**1084.28**	**1055.87**	**1308.78**	**1615.97**

4.2.2 Branch&Bound approach

To assess the ability of the branch&bound approach we have considered randomly generated different problem instances according to table 4.2.1 and following the same procedure already described for the primal–dual heuristic. Thus, we have also 780 instances in total, that were solved by the branch&bound approach and by CPLEX MIP optimizer, v12.4, that was used with its default settings. We have established a maximum computational time for the execution of branch&bound algorithm equal to one hour[1] (no time limit was imposed to CPLEX).

Tables 4.2.6–4.2.7 summarize the computational results obtained in terms of primal solution quality achieved in the root node and by the branch&bound algorithm. We report the minimum, average and maximum gap (in percentage) on the five instances solved for each combination of (S, T, M, N). The average results for all S–scenario problems are reported in the last row of the corresponding tables. Tables **??**–**??** show the solution times (minimum, average and maximum times, in seconds, on the five instances) of the branch&bound, CPLEX, and also the time needed to calculate the admissible solution of the root node. Due to the time limit restriction, the branch&bound was not able to calculate the optimal solution of some instances. As far as CPLEX results are concerned, the solver could not also solve to optimality some of the problems out of the five instances, due to lack of memory to proceed the calculation. We report these cases and solution gaps are provided. However, if these solution gaps exceeded 10% (*gaps excessively high* when compared with solution gaps provided by our procedure), we have decided to exclude them from the time statistics. We report these cases and CPLEX statistics refer only to those instances that were solved to optimality or presented a *reasonable* gap. Whenever CPLEX was not able to solve any of the five instances, the solver time is given as ' * ' (in such cases, due to lack of memory to read the problems).

The computational results show that the admissible primal solution calculated in the root node is of very good quality, and is obtained in reasonable computational times. The maximum time needed to compute the root node solution is, for most problems (around 60%), lower than the minimum time required by CPLEX for the same problems. The worst results in terms of gap are observed in instances with $M \in \{20, 50\}$, but still with a maximum gap of 4.01% ($(S, T, M, N) = (20, 15, 50, 100)$). Within each S-scenario problems, in average, the larger gaps are observed in instances with largest M and N. Nevertheless, the branch&bound algorithm is able to improve significantly the quality of the primal solution calculated in the root node. It should be noted that CPLEX has better computational times than branch&bound for $M \in \{20, 50\}$ and $N \in \{100, 200\}$, in general, but as the number of scenarios increases (especially for problems with 20 scenarios), CPLEX shows difficulties in providing a better solution or even to be able to

[1]This criterion is tested only at the beginning of each node, thus the final computational time may in fact be higher than the time limit established a priori.

generate a feasible solution. From our computational tests we have observed that different problem instances of the same size can make the optimization algorithms behave very differently, both in terms of the computational times and solution quality. To give an example, considering the 5 instances with size $(S,T,M,N) = (20,10,50,100)$, we have observed the following: the branch&bound algorithm was able to calculate the optimal solution of 2 out of the 5 problems using 1 (after only 215.5 sec) and 3 nodes of the tree, respectively. For the other problems, the algorithm was unable to calculate the optimal solutions due to the time limit restriction, but still improved the solution of two problems (using 6 and 7 nodes of the tree). CPLEX was able to calculate the optimum of one problem only (715 sec), and could not provide feasible solutions for any of the other problems due to memory restrictions. These different behaviors make us think that time should be spent looking at the problem's characteristics to try and delineate more efficient branching rules.

Table 4.2.6: Solution quality (in %) for problems with 2 and 5 scenarios.

			S=2						S=5					
T	M	N	Root			B&B			Root			B&B		
			min	aver	max	min	aver	max	min	aver	max	min	aver	max
5	5	20	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
5	5	50	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
5	5	100	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
5	5	200	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
5	10	20	0.00	0.11	0.44	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
5	10	50	0.00	0.13	0.36	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
5	10	100	0.00	0.03	0.14	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
5	10	200	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
5	20	50	0.00	0.41	1.52	0.00	0.00	0.00	0.00	0.39	1.41	0.00	0.00	0.00
5	20	100	0.00	0.02	0.12	0.00	0.00	0.00	0.00	0.19	0.56	0.00	0.00	0.00
5	20	200	0.00	0.01	0.04	0.00	0.00	0.00	0.00	0.08	0.26	0.00	0.00	0.00
5	50	100	0.03	0.62	1.85	0.00	0.00	0.00	0.00	0.15	0.49	0.00	0.00	0.00
5	50	200	0.00	0.30	0.58	0.00	0.00	0.00	0.16	0.23	0.34	0.00	0.00	0.00
10	5	20	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.06	0.29	0.00	0.00	0.00
10	5	50	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.06	0.31	0.00	0.00	0.00
10	5	100	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
10	5	200	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
10	10	20	0.00	0.03	0.12	0.00	0.00	0.00	0.00	0.29	1.46	0.00	0.00	0.00
10	10	50	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.10	0.33	0.00	0.00	0.00
10	10	100	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
10	10	200	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
10	20	50	0.00	0.06	0.32	0.00	0.00	0.00	0.00	0.25	0.57	0.00	0.00	0.00
10	20	100	0.00	0.04	0.20	0.00	0.00	0.00	0.00	0.02	0.12	0.00	0.00	0.00
10	20	200	0.00	0.01	0.06	0.00	0.00	0.00	0.00	0.01	0.01	0.00	0.00	0.00
10	50	100	0.37	0.68	1.17	0.00	0.00	0.00	0.08	0.46	1.25	0.00	0.02	0.10
10	50	200	0.02	0.25	0.45	0.00	0.02	0.08	0.02	0.15	0.40	0.00	0.10	0.23
15	5	20	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
15	5	50	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
15	5	100	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
15	5	200	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
15	10	20	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
15	10	50	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
15	10	100	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.01	0.04	0.00	0.00	0.00
15	10	200	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
15	20	50	0.00	0.03	0.13	0.00	0.00	0.00	0.00	0.11	0.39	0.00	0.00	0.00
15	20	100	0.00	0.05	0.21	0.00	0.00	0.00	0.00	0.05	0.13	0.00	0.00	0.00
15	20	200	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.01	0.03	0.00	0.00	0.00
15	50	100	0.26	0.52	0.90	0.00	0.00	0.00	0.29	0.61	1.20	0.00	0.48	1.20
15	50	200	0.00	0.34	1.47	0.00	0.11	0.57	0.00	0.37	1.06	0.00	0.26	0.75
			0.02	**0.09**	**0.26**	**0.00**	**0.00**	**0.02**	**0.01**	**0.09**	**0.27**	**0.00**	**0.02**	**0.06**

Table 4.2.7: Solution quality (in %) for problems with 10 and 20 scenarios.

			S=10						S=20					
T	M	N	Root			B&B			Root			B&B		
			min	aver	max	min	aver	max	min	aver	max	min	aver	max
5	5	20	0.00	0.07	0.36	0.00	0.00	0.00	0.00	0.01	0.07	0.00	0.00	0.00
5	5	50	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
5	5	100	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
5	5	200	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
5	10	20	0.00	0.03	0.11	0.00	0.00	0.00	0.00	0.22	1.08	0.00	0.00	0.00
5	10	50	0.00	0.03	0.14	0.00	0.00	0.00	0.00	0.08	0.39	0.00	0.00	0.00
5	10	100	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
5	10	200	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
5	20	50	0.00	0.14	0.46	0.00	0.00	0.00	0.00	0.13	0.63	0.00	0.00	0.00
5	20	100	0.00	0.25	1.27	0.00	0.00	0.00	0.00	0.19	0.82	0.00	0.00	0.00
5	20	200	0.00	0.03	0.10	0.00	0.00	0.00	0.00	0.19	0.61	0.00	0.03	0.15
5	50	100	0.00	0.17	0.57	0.00	0.00	0.00	0.08	0.70	1.42	0.00	0.46	1.42
5	50	200	0.31	0.84	1.63	0.24	0.60	1.02	1.85	2.47	3.33	1.62	2.10	2.87
10	5	20	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
10	5	50	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
10	5	100	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
10	5	200	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
10	10	20	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.23	1.16	0.00	0.00	0.00
10	10	50	0.00	0.06	0.30	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
10	10	100	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
10	10	200	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
10	20	50	0.00	0.48	1.43	0.00	0.00	0.00	0.00	0.12	0.56	0.00	0.00	0.00
10	20	100	0.00	0.06	0.20	0.00	0.00	0.00	0.00	0.14	0.34	0.00	0.00	0.00
10	20	200	0.00	0.01	0.03	0.00	0.00	0.00	0.00	0.07	0.22	0.00	0.04	0.13
10	50	100	0.00	0.64	0.92	0.00	0.41	0.77	0.00	1.43	3.55	0.00	1.10	2.57
10	50	200	0.13	0.39	1.05	0.00	0.31	1.05	0.56	1.08	2.25	0.56	1.08	2.25
15	5	20	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
15	5	50	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
15	5	100	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
15	5	200	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
15	10	20	0.00	0.07	0.37	0.00	0.00	0.00	0.00	0.14	0.72	0.00	0.00	0.00
15	10	50	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
15	10	100	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.02	0.08	0.00	0.00	0.00
15	10	200	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
15	20	50	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.06	0.28	0.00	0.00	0.00
15	20	100	0.00	0.06	0.24	0.00	0.00	0.00	0.00	0.02	0.07	0.00	0.01	0.05
15	20	200	0.00	0.01	0.03	0.00	0.00	0.00	0.00	0.01	0.04	0.00	0.01	0.04
15	50	100	0.05	1.05	2.25	0.00	0.42	1.14	0.95	2.30	4.01	0.65	2.17	3.81
15	50	200	0.26	1.12	1.79	0.26	1.01	1.79	1.27	1.84	3.01	1.27	1.84	3.01
			0.02	**0.14**	**0.34**	**0.01**	**0.07**	**0.15**	**0.12**	**0.29**	**0.63**	**0.11**	**0.23**	**0.42**

Table 4.2.8: Computational time (in sec.) for 2–scenario problems.

T	M	N	Root			B&B			CPLEX[(1)]		
			min	aver	max	min	aver	max	min	aver	max
5	5	20	0.00	0.00	0.02	0.00	0.01	0.02	0.06	0.07	0.11
5	5	50	0.00	0.00	0.00	0.00	0.00	0.00	0.16	0.17	0.17
5	5	100	0.00	0.00	0.00	0.00	0.00	0.00	0.34	0.51	0.73
5	5	200	0.00	0.03	0.08	0.00	0.03	0.08	0.92	1.03	1.25
5	10	20	0.00	0.01	0.03	0.00	0.02	0.06	0.11	0.17	0.27
5	10	50	0.00	0.06	0.17	0.00	0.11	0.30	0.34	0.36	0.41
5	10	100	0.00	0.08	0.23	0.00	0.12	0.31	0.76	0.88	0.95
5	10	200	0.00	0.26	1.28	0.00	0.26	1.28	2.04	2.19	2.37
5	20	50	0.03	0.13	0.30	0.03	1.60	5.57	0.72	2.38	6.93
5	20	100	0.03	0.83	1.51	0.03	1.82	5.87	1.89	3.18	5.46
5	20	200	0.02	3.24	12.29	0.02	9.29	26.86	4.77	6.32	11.67
5	50	100	0.48	3.23	5.13	0.89	76.01	184.24	5.16	27.34	67.10
5	50	200	6.29	13.41	19.44	45.24	259.14	600.12	17.53	27.47	58.70
10	5	20	0.00	0.00	0.02	0.00	0.00	0.02	0.11	0.13	0.16
10	5	50	0.00	0.00	0.00	0.00	0.00	0.00	0.36	0.38	0.41
10	5	100	0.00	0.03	0.08	0.00	0.03	0.08	0.80	0.90	1.00
10	5	200	0.00	0.02	0.08	0.00	0.05	0.22	2.12	2.24	2.31
10	10	20	0.00	0.01	0.02	0.00	0.03	0.09	0.23	0.27	0.30
10	10	50	0.00	0.01	0.03	0.00	0.02	0.08	0.73	0.86	1.09
10	10	100	0.00	0.01	0.02	0.00	0.01	0.02	2.09	2.16	2.25
10	10	200	0.00	0.02	0.08	0.00	0.02	0.08	5.05	5.23	5.66
10	20	50	0.08	0.58	1.25	0.08	1.09	3.24	1.56	2.50	5.54
10	20	100	0.09	0.89	2.26	0.09	2.47	10.16	4.32	4.57	4.79
10	20	200	0.09	1.68	6.57	0.09	1.79	6.57	12.04	12.39	12.59
10	50	100	1.95	6.33	11.25	33.68	247.76	432.53	17.83	64.68	106.77
10	50	200	40.17	52.61	90.46	434.76	1238.78	3624.30	44.73	69.14	156.31
15	5	20	0.00	0.01	0.06	0.00	0.01	0.06	0.17	0.20	0.23
15	5	50	0.00	0.00	0.00	0.00	0.00	0.00	0.58	0.66	0.70
15	5	100	0.00	0.03	0.09	0.00	0.03	0.09	1.19	1.32	1.45
15	5	200	0.00	0.32	1.56	0.00	0.32	1.56	3.20	3.53	3.90
15	10	20	0.00	0.00	0.02	0.00	0.00	0.02	0.39	0.41	0.44
15	10	50	0.00	0.01	0.02	0.00	0.01	0.02	1.06	1.27	1.44
15	10	100	0.00	0.03	0.11	0.00	0.03	0.11	2.67	3.26	3.88
15	10	200	0.02	0.02	0.02	0.02	0.02	0.02	9.11	9.50	9.67
15	20	50	0.31	0.99	1.97	0.31	1.16	2.62	2.54	3.32	5.16
15	20	100	0.02	1.55	7.27	0.02	1.56	7.27	6.44	6.98	7.74
15	20	200	0.20	0.95	2.26	0.20	1.18	2.73	23.31	23.91	24.62
15	50	100	2.39	5.40	9.50	417.53	1751.79	3604.30	72.45	185.26	314.83
15	50	200	58.62	106.15	210.16	81.59	1469.96	3617.16	59.87	90.70	183.60
			2.84	**5.10**	**9.89**	**26.01**	**129.91**	**311.23**	**7.94**	**14.56**	**25.97**

(1) $(T, M, N) = (15, 50, 200)$: solution gap of 2.54% in one instance.

Table 4.2.9: Computational time (in sec.) for 5–scenario problems.

T	M	N	Root			B&B			CPLEX(1)		
			min	aver	max	min	aver	max	min	aver	max
5	5	20	0.00	0.00	0.00	0.00	0.00	0.00	0.14	0.16	0.17
5	5	50	0.00	0.01	0.02	0.00	0.01	0.02	0.50	0.55	0.64
5	5	100	0.00	0.01	0.02	0.00	0.01	0.02	1.23	1.27	1.30
5	5	200	0.00	0.01	0.03	0.00	0.01	0.03	2.98	3.15	3.37
5	10	20	0.00	0.01	0.05	0.00	0.01	0.05	0.33	0.34	0.36
5	10	50	0.00	0.18	0.55	0.00	0.37	1.47	1.15	1.26	1.40
5	10	100	0.00	2.63	10.19	0.00	3.16	10.19	2.43	3.36	5.94
5	10	200	0.00	0.23	0.66	0.00	0.23	0.66	7.21	7.37	7.52
5	20	50	0.08	0.70	1.72	0.08	3.08	6.93	2.29	3.37	5.87
5	20	100	0.02	5.38	10.78	0.02	13.00	22.48	5.77	11.65	29.22
5	20	200	2.14	34.26	52.57	5.71	131.14	384.53	22.67	42.81	99.15
5	50	100	4.57	14.99	23.76	25.16	71.50	124.04	22.25	36.06	79.39
5	50	200	49.97	94.49	188.82	537.22	1708.13	3121.40	77.69	131.91	254.75
10	5	20	0.00	0.04	0.20	0.00	0.05	0.25	0.38	0.55	1.19
10	5	50	0.00	0.13	0.50	0.00	0.39	1.78	1.20	1.27	1.31
10	5	100	0.00	0.02	0.03	0.00	0.02	0.03	2.75	3.00	3.37
10	5	200	0.02	0.03	0.05	0.02	0.03	0.05	7.29	7.65	7.89
10	10	20	0.00	0.31	1.11	0.00	0.80	2.62	0.83	0.95	1.11
10	10	50	0.00	0.86	3.48	0.00	2.05	6.91	2.56	2.83	3.25
10	10	100	0.02	0.16	0.53	0.02	0.16	0.53	6.46	6.68	6.88
10	10	200	0.03	0.04	0.05	0.03	0.04	0.05	22.40	23.27	24.09
10	20	50	1.45	4.93	8.81	7.00	22.13	39.56	6.41	12.19	23.18
10	20	100	0.25	9.70	27.44	0.25	15.21	50.59	18.21	18.96	20.64
10	20	200	2.15	19.59	68.11	2.15	127.16	566.25	53.68	57.85	69.75
10	50	100	11.22	50.01	82.74	440.05	1672.13	3601.46	69.59	321.40	556.18
10	50	200	210.62	344.60	432.31	3672.54	3723.90	3843.65	200.18	244.33	297.48
15	5	20	0.00	0.00	0.00	0.00	0.00	0.00	0.51	0.59	0.66
15	5	50	0.00	0.02	0.03	0.00	0.02	0.03	1.86	1.98	2.03
15	5	100	0.03	0.06	0.19	0.03	0.06	0.19	4.43	4.68	4.90
15	5	200	0.05	0.06	0.09	0.05	0.06	0.09	15.29	15.60	15.91
15	10	20	0.22	0.63	1.95	0.33	1.06	3.76	1.23	1.81	3.42
15	10	50	0.00	0.03	0.06	0.00	0.03	0.06	4.51	4.60	4.65
15	10	100	0.03	1.97	8.94	0.03	2.58	12.00	12.22	14.67	23.31
15	10	200	0.06	0.32	1.36	0.06	0.32	1.36	38.41	39.64	40.73
15	20	50	2.81	10.82	25.55	2.81	14.30	32.21	9.95	11.94	14.03
15	20	100	4.99	23.75	48.55	4.99	74.52	161.87	33.23	41.48	63.80
15	20	200	19.19	64.08	156.41	19.19	95.58	282.41	91.23	93.13	95.00
15	50	100	23.43	60.40	120.53	738.02	2553.79	3768.07	111.53	594.58	1426.71
15	50	200	20.58	338.01	639.04	2986.28	3663.47	4178.27	264.97	264.97	264.97
			9.07	**27.78**	**49.16**	**216.46**	**356.42**	**518.61**	**28.92**	**52.15**	**88.86**

(1) $(T, M, N) = (10, 50, 200)$: solution gap of 5.04% in one instance; $(T, M, N) = (15, 50, 200)$: statistics refer only to one instance, as gaps on the other four were *excessively high*–ranged from 61% to 70%.

Table 4.2.10: Computational time (in sec.) for 10–scenario problems.

T	M	N	Root			B&B			CPLEX[(1)]		
			min	aver	max	min	aver	max	min	aver	max
5	5	20	0.00	0.20	0.53	0.00	0.24	0.73	0.38	0.66	1.65
5	5	50	0.00	0.11	0.50	0.00	0.11	0.50	1.12	1.23	1.34
5	5	100	0.00	0.01	0.02	0.00	0.01	0.02	2.92	3.17	3.48
5	5	200	0.02	0.03	0.03	0.02	0.03	0.03	7.64	7.95	8.38
5	10	20	0.00	0.33	0.45	0.00	0.54	1.28	0.92	1.25	2.14
5	10	50	0.00	1.91	6.94	0.00	6.77	31.25	2.81	4.90	12.78
5	10	100	0.05	5.30	23.34	0.05	6.27	23.34	7.13	7.85	8.69
5	10	200	0.05	3.47	9.61	0.05	3.47	9.61	21.72	22.82	26.07
5	20	50	1.73	5.06	8.74	1.73	7.05	12.56	6.90	7.94	10.58
5	20	100	2.40	8.13	17.85	4.49	31.27	85.94	21.17	32.71	67.05
5	20	200	26.57	164.40	341.20	26.57	221.01	437.28	58.87	64.29	72.42
5	50	100	34.94	79.29	121.93	52.39	588.56	1301.41	69.14	139.48	259.37
5	50	200	418.86	634.12	946.89	3723.94	3935.26	4273.85	524.82	959.66	1527.31
10	5	20	0.00	0.01	0.02	0.00	0.01	0.02	0.83	0.90	0.98
10	5	50	0.02	0.03	0.08	0.02	0.03	0.08	2.67	2.88	3.17
10	5	100	0.03	0.04	0.05	0.03	0.04	0.05	7.44	7.88	8.14
10	5	200	0.06	0.08	0.08	0.06	0.08	0.08	21.68	23.37	24.98
10	10	20	0.00	0.29	1.20	0.00	0.29	1.20	1.93	2.10	2.43
10	10	50	0.03	9.70	34.91	0.03	15.07	59.16	6.66	9.36	19.19
10	10	100	0.06	1.23	3.78	0.06	1.23	3.78	21.11	22.02	22.50
10	10	200	0.11	0.11	0.13	0.11	0.11	0.13	55.16	57.02	59.45
10	20	50	6.16	27.61	49.41	6.16	91.57	207.54	18.80	55.25	132.16
10	20	100	7.22	73.58	205.44	7.22	357.21	803.21	56.55	118.88	226.20
10	20	200	1.64	241.84	460.86	1.64	302.18	648.24	127.48	136.49	152.48
10	50	100	73.26	225.40	334.34	73.26	2847.19	3658.22	161.01	612.17	841.99
10	50	200	1091.86	1871.35	2703.61	3344.55	4004.48	4996.68	401.34	610.96	820.58
15	5	20	0.02	1.33	6.54	0.02	2.06	10.22	1.53	1.78	1.95
15	5	50	0.05	0.07	0.13	0.05	0.07	0.13	4.98	5.10	5.34
15	5	100	0.08	0.20	0.41	0.08	0.20	0.41	14.98	15.69	16.65
15	5	200	0.14	0.43	1.47	0.14	0.43	1.47	43.07	45.16	47.69
15	10	20	0.28	1.62	4.68	0.28	10.41	34.41	3.21	3.77	4.88
15	10	50	0.05	1.24	3.57	0.05	1.24	3.57	12.81	13.24	13.73
15	10	100	0.11	10.94	41.96	0.11	10.94	41.96	37.46	40.84	46.11
15	10	200	0.17	0.21	0.23	0.17	0.21	0.23	99.33	103.01	105.32
15	20	50	0.78	17.82	40.72	0.78	46.89	158.57	32.93	43.70	72.59
15	20	100	8.19	58.64	105.66	8.19	270.82	796.27	95.40	164.64	398.57
15	20	200	0.28	508.17	1231.29	0.28	625.82	1231.29	217.14	248.22	310.01
15	50	100	187.43	427.05	785.32	2662.76	3490.24	3750.85	313.22	361.24	409.27
15	50	200	526.03	1771.76	3170.56	3674.49	4156.30	4983.22	*	*	*
			61.25	**157.77**	**273.45**	**348.46**	**539.38**	**706.89**	**65.37**	**104.20**	**151.25**

[(1)] $(T, M, N) = (5, 50, 200)$: gap of 2.28% in one instance, and gaps *excessively high* in two instances; $(T, M, N) = (10, 50, 100)$: two *excessively high* gaps (15%; 19%); $(T, M, N) = (10, 50, 200)$: three *excessively high* gaps (around 60%). $(T, M, N) = (15, 20, 200)$: one gap of 0.29%; $(T, M, N) = (15, 50, 100)$: two *excessively high* gaps (62%; 69%) and one instance without any feasible solution.

Table 4.2.11: Computational time (in sec.) for 20–scenario problems.

T	M	N	Root			B&B			CPLEX[1]		
			min	aver	max	min	aver	max	min	aver	max
5	5	20	0.00	0.13	0.61	0.00	0.38	1.87	0.80	1.48	3.00
5	5	50	0.02	0.65	3.15	0.02	0.65	3.15	2.85	3.16	3.53
5	5	100	0.03	30.86	154.19	0.03	30.86	154.19	7.86	9.43	14.35
5	5	200	0.06	0.08	0.09	0.06	0.08	0.09	25.26	26.22	27.11
5	10	20	0.02	2.52	4.90	0.02	6.26	20.87	2.11	2.51	2.75
5	10	50	1.09	13.61	23.99	1.09	21.88	42.84	7.89	16.82	47.86
5	10	100	3.09	70.80	245.59	3.09	108.01	431.64	24.32	28.15	37.46
5	10	200	0.09	145.04	638.04	0.09	145.04	638.04	64.19	73.17	101.15
5	20	50	0.34	21.11	44.29	0.34	113.64	435.97	20.61	74.18	221.68
5	20	100	19.64	101.85	222.91	19.64	403.43	1015.67	56.89	117.68	288.10
5	20	200	47.05	621.55	1342.97	58.31	2219.18	4636.38	164.66	537.78	1190.23
5	50	100	202.60	310.82	359.07	988.31	2449.90	3983.55	264.14	468.16	883.07
5	50	200	383.79	1812.58	2803.42	3981.65	4692.20	5493.17	*	*	*
10	5	20	0.03	0.55	1.78	0.03	1.15	4.79	2.11	2.22	2.29
10	5	50	0.06	0.07	0.09	0.06	0.07	0.09	8.25	8.55	8.71
10	5	100	0.13	0.15	0.19	0.13	0.15	0.19	23.53	24.93	26.16
10	5	200	0.23	0.27	0.33	0.23	0.27	0.33	58.75	61.76	65.38
10	10	20	0.03	2.56	6.22	0.03	3.01	6.22	5.46	5.95	7.49
10	10	50	0.08	23.42	45.74	0.08	23.42	45.74	25.91	27.31	29.84
10	10	100	0.16	7.62	35.22	0.16	7.62	35.22	55.93	58.88	61.87
10	10	200	0.30	1.62	3.76	0.30	2.45	7.57	131.67	137.53	140.46
10	20	50	34.41	140.33	225.34	34.41	352.87	1008.43	62.28	75.39	97.60
10	20	100	53.57	193.76	409.70	107.17	1642.17	3957.66	141.54	228.71	470.41
10	20	200	689.88	1786.88	3325.52	689.88	2822.17	4521.18	313.94	390.26	512.89
10	50	100	215.51	748.85	1289.93	215.51	3259.06	4482.38	715.61	715.61	715.61
10	50	200	1860.63	3639.29	4646.76	4283.56	4696.01	5646.12	*	*	*
15	5	20	0.06	9.97	49.55	0.06	9.97	49.55	4.23	4.46	4.57
15	5	50	0.14	0.17	0.19	0.14	0.17	0.19	15.34	16.33	17.22
15	5	100	0.27	0.42	0.86	0.27	0.42	0.86	40.19	45.06	47.83
15	5	200	0.50	0.57	0.70	0.50	0.57	0.70	97.42	105.32	109.15
15	10	20	0.06	12.31	43.73	0.06	32.01	142.21	10.97	20.22	54.18
15	10	50	0.17	79.79	297.60	0.17	157.06	683.95	42.01	46.75	61.62
15	10	100	0.34	8.53	37.82	0.34	35.66	139.39	104.63	108.50	111.84
15	10	200	0.61	0.65	0.73	0.61	0.65	0.73	221.55	233.70	241.26
15	20	50	49.64	198.52	353.08	49.64	1109.26	3475.71	104.99	244.58	616.14
15	20	100	109.61	435.24	641.11	109.61	2080.56	3728.63	228.11	283.64	332.78
15	20	200	364.42	2457.07	5352.72	364.42	2525.55	5352.72	*	*	*
15	50	100	414.34	1973.06	3138.74	3914.32	4599.44	5882.31	*	*	*
15	50	200	6064.95	12619.44	15228.66	6064.95	12619.44	15228.66	*	*	*
			269.69	**704.43**	**1050.75**	**535.62**	**1183.91**	**1827.15**	**89.88**	**123.66**	**192.81**

[1] $(T, M, N) = (5, 50, 100)$: one *excessively high* gap (39%); $(T, M, N) = (10, 50, 100)$: CPLEX was only able to solve one of the instances (no feasible solutions were provided for the other four).

4.2.3 Lagrangean relaxation approach

In order to analyze the model α-DUFLPU and to assess the efficiency of the proposed algorithmic approach, six data sets were considered, with input values of (S, T, J, I) given in Table 4.2.12. The corresponding number of variables and constraints are also provided. For each one of these six sets, forty instances were randomly generated.

Table 4.2.12: Dimension of the test problems.

Set	S	T	J	I	num var	num const
I	10	10	20	100	200200	210030
II	10	10	20	200	400200	420030
III	10	10	40	100	400400	410050
IV	10	20	20	100	400400	420030
V	20	10	20	100	400200	420040
VI	50	5	20	100	500100	525070

We have considered $\alpha \in \{0.075, 0.10, 0.15, 0.20\}$. The stopping criteria were established after some preliminary tests. The maximum computational time for the execution of the algorithm is two hours for problems with 20 and 50 scenarios and one hour for all other problems. In addition, we have also established as stopping criterium the quality of the best solution achieved by the algorithm, measured by the gap between the best known upper and lower bounds: 2% for the problems with 20 and 50 scenarios and 1.5% for all the others. We have also imposed a maximum number of iterations which could vary from 20 to 50 (largest instances). The computational results provided in this section were obtained considering a step size coefficient $z = 1$ which gave the best results in general. Other initial values of z as well as lowering z after a few iterations of the algorithm were tested without significant improvements in results.

Table 4.2.13 summarizes the computational results obtained. For each data set and for each α, column 'feas/inf/ind' reports the number of instances for which a feasible solution was found by the algorithm, the number of instances identified as infeasible and also the number of instances for which the algorithm was unable to achieve a feasible solution (solution indeterminate). The statistics shown in the next columns refer only to the subsets of instances for which a feasible solution was found (feasible instances). For each α and for each feasible instance, the increase of the best objective function value relative to the best one achieved for $\alpha = 0.2$ was calculated. Column 'increase' depicts the average increase (in percentage) obtained for each α. The next columns report the minimum, average and maximum gap on the feasible instances, and the minimum, average and maximum time (in seconds) spent by the algorithm to solve each set of feasible instances. For each set, the last row shows the average results for gap and time.

We can see that the number of feasible instances decreases as α decreases in all sets, due to infeasibility of some instances or due to the algorithm being unable to achieve a feasible solution. The algorithm stopped with indeterminate solutions in only 7.6% of all 960 problems, due to the time limit established a priori, remaining the doubt about the feasibility of those instances. As expected, the objective function values increase as regret decreases. In terms of solution quality, the larger gaps were observed in sets V and VI, sets with larger number of scenarios, but the quality of the solutions is still very good. The worst gap equals 1.72% and was observed for instances with 50 scenarios. Apparently, the decrease of parameter α does not seem to cause a deterioration in the quality of the solutions in terms of gap, noticing however that the dimensions of the samples with problems for smaller values of α are very small. The computational time spent by the algorithm can vary a lot, even for problems within the same set (same size) and same α. The higher execution times were observed in set III, with larger number of potential facility locations, and sets V and VI with larger number of scenarios.

We have solved the same sets of problems using an exact algorithm, CPLEX MIP optimizer, v12.4, with the same stopping criteria. The results are reported in Table 4.2.14. CPLEX stopped with indeterminate solutions in 10% of all 960 problems, due to lack of memory. Considering only set VI, CPLEX was unable to find a feasible solution in 19.4% of those 160 problems as Lagrangean relaxation approach stopped with indeterminate solutions in only 8%. We noticed that within sets I to V the indeterminate instances of CPLEX were almost the same for which our algorithm was also unable to find a feasible solution, except 11 instances for which only our algorithm was able to find a feasible solution and 6 feasible instances only achieved by CPLEX. The results for these sets are very similar, reflecting that some instances are the hardest for both optimization algorithms. In terms of solution quality, CPLEX provides smaller average gaps than the Lagrangean relaxation approach, although less feasible instances were found by CPLEX, in particular in set VI with larger number of scenarios. In addition, CPLEX's maximum gap 1.97% is greater than the worst gap 1.72% achieved by the algorithm (achieved in sets V and VI, respectively, both for $\alpha = 0.2$). In terms of computational time, CPLEX can also vary a lot. We can see that for all problems, the minimum computational time was obtained by the algorithm, in same cases clearly outperforming CPLEX. In terms of average computational times, CPLEX is better than the algorithm on sets III and VI, thought less feasible solutions were achieved by the solver.

In order to gather more information about the set of indeterminate instances, the computational time of one hour was increased to two hours in some of the sets. However, the algorithms were only able to find more infeasible instances, though very few.

In brief, the computational results show that the Lagragean relaxation approach is capable of finding very good quality solutions in reasonable computational times. It should be noted that CPLEX has better average gaps and computational times for some of the problems considered. However, for problems with larger number of scenarios the solver shows more difficulties to generate feasible solutions.

Table 4.2.13: Computational results.

Set	α	feas/inf/ind	increase (%) mean	gap (%) min	gap (%) mean	gap (%) max	time (sec.) min	time (sec.) mean	time (sec.) max
I	0.2	40/0/0	0.00	0.00	0.23	1.05	9.95	119.13	873.9
	0.15	39/0/1	0.17	0.00	0.30	1.32	9.20	130.02	1621.2
	0.1	32/8/0	0.23	0.00	0.09	0.55	21.07	211.38	1567.9
	0.075	14/26/0	0.29	0.00	0.04	0.39	28.23	222.41	1031.7
				0.00	**0.17**	**0.83**	**17.12**	**170.74**	**1273.7**
II	0.2	40/0/0	0.00	0.00	0.09	0.51	1.2	327.2	1117.4
	0.15	40/0/0	0.19	0.00	0.25	1.14	1.3	352.7	1126.5
	0.1	38/0/2	0.25	0.00	0.19	0.99	1.3	449.2	1602.9
	0.075	18/11/11	0.26	0.00	0.12	0.90	30.2	585.9	3029.4
				0.00	**0.17**	**0.89**	**8.47**	**428.77**	**1719.1**
III	0.2	40/0/0	0.00	0.00	0.39	1.37	52.7	944.4	3609.0
	0.15	40/0/0	0.06	0.00	0.35	1.10	52.9	1008.3	3691.9
	0.1	25/9/6	0.14	0.00	0.25	0.83	98.2	789.1	3706.5
	0.075	8/26/6	0.18	0.00	0.17	0.64	97.9	807.6	3528.9
				0.00	**0.29**	**0.99**	**75.42**	**887.33**	**3634.1**
IV	0.2	40/0/0	0.00	0.00	0.29	1.46	5.5	303.2	2486.5
	0.15	40/0/0	0.24	0.00	0.45	1.46	5.5	367.2	1753.8
	0.1	23/1/16	0.28	0.00	0.33	1.33	5.5	586.4	3111.0
	0.075	8/21/11	0.29	0.00	0.12	0.68	5.6	289.5	742.2
				0.00	**0.30**	**1.23**	**5.53**	**386.6**	**2023.4**
V	0.2	40/0/0	0.00	0.00	0.31	1.59	88.4	480.3	2718.8
	0.15	36/0/4	0.37	0.00	0.32	1.51	88.6	630.7	1951.6
	0.1	18/20/2	0.44	0.00	0.28	1.05	127.0	769.9	3415.6
	0.075	5/34/1	0.48	0.00	0.14	0.52	128.6	505.4	1449.7
				0.00	**0.26**	**1.17**	**108.2**	**596.6**	**2383.9**
VI	0.2	40/0/0	0.00	0.00	0.24	1.72	59.4	929.8	3883.9
	0.15	40/0/0	0.03	0.00	0.19	1.24	57.6	1058.5	6631.6
	0.1	33/3/4	0.14	0.00	0.11	0.99	58.3	857.4	3124.8
	0.075	17/14/9	0.44	0.00	0.04	0.33	165.6	781.2	1608.7
				0.00	**0.15**	**1.07**	**85.2**	**906.7**	**3812.3**

Table 4.2.14: Computational results using CPLEX.

Set		feas/inf/ind	gap (%)			time (sec.)		
	α		min	mean	max	min	mean	max
I	0.2	40/0/0	0.00	0.15	1.26	54.40	130.48	1071.06
	0.15	39/0/1	0.00	0.21	1.46	54.41	150.48	475.98
	0.1	32/8/0	0.00	0.17	1.40	69.94	341.02	1711.57
	0.075	14/26/0	0.00	0.01	0.13	61.04	177.86	488.88
			0.00	**0.13**	**1**	**59.95**	**199.96**	**936.87**
II	0.2	40/0/0	0.00	0.04	0.91	160.4	232.8	404.9
	0.15	40/0/0	0.00	0.05	0.91	159.9	292.7	947.4
	0.1	38/0/2	0.00	0.10	0.77	158.9	567.3	3582.2
	0.075	18/11/11	0.00	0.04	0.38	167.2	685.7	2206.7
			0.00	**0.06**	**0.74**	**161.61**	**444.62**	**1785.28**
III	0.2	40/0/0	0.00	0.29	1.37	138.3	404.5	1193.1
	0.15	38/0/2	0.00	0.27	1.42	137.7	568.8	2218.1
	0.1	25/9/6	0.00	0.17	1.08	144.6	877.0	3502.1
	0.075	8/26/6	0.00	0.10	0.34	145.8	520.1	1596.8
			0.00	**0.21**	**1.05**	**141.59**	**592.62**	**2127.51**
IV	0.2	37/0/3	0.00	0.10	0.95	139.7	268.2	917.8
	0.15	36/0/4	0.00	0.16	1.30	149.0	425.9	1999.4
	0.1	23/1/16	0.00	0.15	0.65	161.4	793.4	3600.5
	0.075	12/21/7	0.00	0.15	0.58	200.2	1116.5	3268.7
			0.00	**0.14**	**0.87**	**162.60**	**651.02**	**2446.59**
V	0.2	40/0/0	0.00	0.22	1.97	181.1	424.2	1861.5
	0.15	38/0/2	0.00	0.32	1.51	201.7	934.7	4122.6
	0.1	16/20/4	0.00	0.18	0.73	196.6	896.2	3567.0
	0.075	5/34/1	0.00	0.00	0.00	198.8	409.3	784.4
			0.00	**0.18**	**1.05**	**194.6**	**666.1**	**2583.8**
VI	0.2	37/0/3	0.00	0.2	1.77	287.2	504.2	1106.1
	0.15	37/0/3	0.00	0.11	1.72	293.9	520.7	1092.9
	0.1	25/3/12	0.00	0.00	0.01	288.1	416.9	975.6
	0.075	13/14/13	0.00	0.00	0.00	294.3	384.9	590.9
			0.00	**0.08**	**0.88**	**290.9**	**456.7**	**941.4**

Postlude

We have been concerned with facility location problems under uncertainty, adding a humble contribution to the location research field, through the development of mathematical models and solution methods for this class of problems. We are dealing with difficult problems, but with a growing importance from a practical point of view as such problems may reflect better the uncertain world in which we live.

In this work, we have considered several discrete dynamic facility location problems under uncertainty. The uncertainty, in many of the problems' parameters, is explicitly represented in the models by a set of possible future scenarios. The classical DUFLP is addressed through several models and perspectives along Chapter 2: an extension considering uncertainty, that contains the classical deterministic static and dynamic problems as particular problems; an extension of the previous model with robust constraints related with the uncertain future; a multi–objective approach where each scenario is interpreted as one objective. We have considered several models with capacity facilities that bring additional difficulties but other interesting situations arise as well. In terms of models, we have limited ourselves to certain assumptions such as to objective functions minimizing expected total costs or total cost. Other objective functions that can better represent the attitude towards risk of different Decision Makers should be considered as well. Other extensions to these problems could consider the introduction of the possibility of closing already opened facilities to increase the range of applicability of the models. Mainly within capacitated problems there is still a considerable amount of situations to be explored. The incorporation of robust constraints into those models related with upper bounds on satisfied demand is an ongoing problem.

Efficient techniques were developed in Chapter 3 to cope with the uncapacitated problems, being an alternative to solvers that show more difficulties to find solutions for large–sized problems (Chapter 4). The effect of data to the performance of those algorithms also needs further study. We have not developed dedicated solution approaches to tackle the capacitated models yet, hence it is also a possible future work. Classical heuristics have a major drawback: changes in the problem's formulation (additional restrictions, changes in the objective function, for instance), imply changes in the procedures with high costs due to the time spent developing new dedicated procedures. Meta-heuristics, namely genetic algorithms, have the advantage of being flexible and intelligent algorithms, that can be easily customized to be applied to different problems with different specificities. The flexibility advantage comes, usually, at the cost of computational time. This is why hybrid methods will possibly have to be thought incorporating all the available information about the problem.

"*The best way to handle uncertainty, and to make decisions under uncertainty, is to accept uncertainty, make a strong effort to structure it and understand it, and finally, make it part of the decision making reasoning*" (Kouvelis and Yu [47]).

Bibliography

[1] F. B. Abdelaziz. Solution approaches for the multiobjective stochastic programming. *European Journal of Operational Research*, 216:1–16, 2012.

[2] E. Aghezzaf. Capacity planning and warehouse location in supply chains with uncertain demands. *Journal of the Operational Research Society*, 56(4):453–462, 2005.

[3] S. Ahmed and R. Garcia. Dynamic capacity acquisition and assignment under uncertainty. *Annals of Operations Research*, 124:267–283, 2004.

[4] Hassene Aissi, C. Bazgan, and D. Vanderpooten. Min-max and min-max regret versions of combinatorial optimization problems: a survey. *European Journal of Operational Research*, 197:427–438, 2009.

[5] Maria Albareda-Sambola, A. Alonso-Ayuso, Laureano F. Escudero, Elena Fernández, and Celeste Pizarro. On solving the multi-period location-assignment problem under uncertainty. *Technical Reports On Statistics and Decision Sciences, May 2009, Dpto. de Estadística e Investigación Operativa, Universidad Rey Juan Carlos*, 2009.

[6] Maria Albareda-Sambola, A. Alonso-Ayuso, Laureano F. Escudero, Elena Fernandez, and Celeste Pizarro. Fix-and-relax-coordination for a multi-period location-allocation problem under uncertainty. *Computers & Operations Research*, 40:2878–2892, 2013.

[7] Maria Albareda-Sambola, Elena Fernandez, and Francisco Saldanha-da-Gama. The facility location problem with Bernoulli demands. *Omega*, 39:335–345, 2011.

[8] Aharon Ben-Tal and Arkadi Nemirovski. Robust solutions of linear programming problems contaminated with uncertain data. *Mathematical Programming*, 88:411–424, 2000.

[9] Michel Berkelaar, Kjell Eikland, and Peter Notebaert. Open source (mixed–integer) linear programming system. http://lpsolve.sourceforge.net/5.5/.

[10] O. Berman and Z. Drezner. The p-median under uncertainty. *European Journal of Operational Research*, 189:19–30, 2008.

[11] Dimitris Bertsimas and Vineet Goyal. On the power of robust solutions in two-stage stochastic and adaptive optimization problems. *Mathematics of Operations Research*, 35(2):284–305, 2010.

[12] Dimitris Bertsimas and Melvyn Sim. The price of robustness. *Operations Research*, 52(1):35–53, 2004.

[13] O. Bilde and J. Krarup. Sharp lower bounds and efficient algorithms for the simple plant location problem. *Annals of Discrete Mathematics*, 1:79–97, 1977.

[14] J. R. Birge and F. V. Louveaux. *Introduction to Stochastic Programming*. Springer, 1997.

[15] Y. Cardona-Valdés, A. Álvarez, and D. Ozdemir. A bi–objective supply chain design problem with uncertainty. *Transportation Research Part C*, 19:821–832, 2011.

[16] G. Cornuejols, G. L. Nemhauser, and L. A. Wolsey. The uncapacitated facility location problem. In P. B. Mirchandani and R. L. Francis, editors, *Discrete Location Theory*, pages 119–171. John Wiley&Sons, 1990.

[17] J. Current, S. Ratick, and C. ReVelle. Dynamic facility location when the total number of facilities is uncertain: A decision analysis approach. *European Journal of Operational Research*, 110:597–609, 1997.

[18] M. S. Daskin. *Network and Discrete Location: Models, Algorithms and Applications*. John Wiley&Sons, 1995.

[19] Ron S. Dembo. Scenario optimization. *Annals of Operations Research*, 30:63–80, 1991.

[20] Joana Dias, M. Eugénia Captivo, and João Clímaco. An interactive procedure dedicated to a bicriteria plant location model. *Computers & Operations Research*, 30:1977–2002, 2003.

[21] Joana Dias, M. Eugénia Captivo, and João Clímaco. Efficient primal-dual heuristic for a dynamic location problem. *Computers & Operations Research*, 34(6):1800–1823, 2007.

[22] Joana Dias, M. Eugénia Captivo, and João Clímaco. A dynamic location problem with maximum decreasing capacities. *Central European Journal of Operations Research*, 16:251–280, 2008.

[23] Joana M. Dias. *Localização Dinâmica: Modelos e Técnicas*. Imprensa da Universidade de Coimbra, 2010.

[24] Joana M. Dias and M. C. Marques. A multiobjective approach for a dynamic simple plant location problem under uncertainty. In B. Murgante, S. Misra, A. M. Rocha, C. Torre, J. G. Rocha, M. I. Falcão, D. Taniar, B. Apduhan, and O. Gervasi, editors, *Computational Science and Its Applications– ICCSA 2014*, volume 8580 of *Lecture Notes in Computer Science*, pages 60–75. Springer International Publishing, 2014.

[25] Jitka Dupačová. Stochastic programming: Approximation via scenarios. *Apportaciones mathematicás*, Ser. Communicationes 24:77–95. Also available via http://www.emis.de/proceedings/index.html, 1995, 1998.

[26] Jitka Dupačová, Giorgio Consigli, and Stein W. Wallace. Scenarios for multistage stochastic programs. *Annals of Operations Research*, 100:25–53, 2000.

[27] Ian N. Durbach and Theodor J. Stewart. An experimental study of the effect of uncertainty representation on decision making. *European Journal of Operational Research*, 214:380–392, 2011.

[28] Martin Dyer and Leen Stougie. Computational complexity of stochastic programming problems. *Math. Program.*, Ser. A 106:423–432, 2006.

[29] D. Erlenkotter. A dual-based procedure for uncapacitated facility location. *Operations Research*, 26(6):992–1009, 1978.

[30] D. Erlenkotter. A comparative study of approaches to dynamic location problems. *European Journal of Operational Research*, 6(2):133–143, 1981.

[31] Monique Guignard. Lagrangean relaxation. *TOP*, 11:151–228, 2003.

[32] G. Guillén, F. D. Mele, M. J. Bagajewicz, A. Espuna, and L. Puigjaner. Multiobjective supply chain design under uncertainty. *Chemical Engineering Science*, 60:1535–1553, 2005.

[33] W. J. Gutjahr. Two metaheuristics for multiobjective stochastic combinatorial optimization. In O. Lupanov, O. M. Kasim-Zade, A. V. Chaskin, and K. Steinhöfel, editors, *Stochastic Algorithms: Foundations and Applications. LNCS*, pages 116–125. Springer–Verlag Berlin Heidelberg, 2005.

[34] S. L. Hakimi. Optimum locations of switching centers and the absolute centers and medians of a graph. *Operations Research*, 12:450–459, 1964.

[35] Holger Heitsch and Werner Romisch. Scenario tree generation for multi-stage stochastic programs. In Marida Bertocchi, Giorgio Consigli, and Michael A. H. Dempster, editors, *Stochastic Optimization Methods in Finance and Energy*, pages 313–341. Springer New York, 2011.

[36] Patricio Hernández, Antonio Alonso–Ayuso, Fernanda Bravo, Laureano F. Escudero, Monique Guignard, Vladimir Marianov, and Andrés Weintraub. A branch-and-cluster coordination scheme for selecting prison facility sites under uncertainty. *Computers & Operations Research*, 39:2232–2241, 2012.

[37] S. Hulsurkar, M. P. Biswal, and S. B. Sinha. Fuzzy programming approach to multi-objective stochastic linear programming problems. *Fuzzy Sets and Systems*, 88:173–181, 1997.

[38] D. A. Iancu and N. Trichakis. Pareto efficiency in robust optimization. *Management Science*, 60:130–147, 2014.

[39] S. K. Jacobsen. Multiperiod capacitated location models. In P. B. Mirchandani and R. L. Francis, editors, *Discrete Location Theory*, pages 173–208. John Wiley&Sons, 1990.

[40] K. Jornsten and M. Bjorndal. Dynamic location under uncertainty. *Studies in Regional and Urban Planning*, 3:163–183, 1994.

[41] J. Teghem Jr, D. Dufrane, M. Thauvoye, and P. Kunsch. Strange: an interactive method for multi–objective linear programming under uncertainty. *European Journal of Operational Research*, 26:65–82, 1986.

[42] Peter Kall and Stein W. Wallace. *Stochastic Programming.* John Wiley& Sons, 2nd edition, 1994.

[43] Michal Kaut and Stein W. Wallace. Evaluation of scenario-generation methods for stochastic programming. *Pacific Journal of Optimization*, 3(2):257–271, 2007.

[44] Farhad Kiya and Hamid Davoudpour. Stochastic programming approach to re-designing a warehouse network under uncertainty. *Transportation Research Part E*, 48:919–936, 2012.

[45] K. Klamroth, E. Köbis, A. Schöbel, and Chr. Tammer. A unified approach for different concepts of robustness and stochastic programming via non–linear scalarizing functionals. *Optimization*, 62(5):649–671, 2013.

[46] Bernhard Korte and Jens Vygen. Facility location. In Bernhard Korte and Jens Vygen, editors, *Combinatorial Optimization, volume 21 of Algorithms and Combinatorics*, pages 593–628. Springer Berlin Heidelberg, 2012.

[47] Panos Kouvelis and Gang Yu. *Robust Discrete Optimization and its Applications.* Kluwer Academic Publishers, 1997.

[48] Jakob Krarup and Peter M. Pruzan. Ingredients of locational analysis. In P. B. Mirchandani and R. L. Francis, editors, *Discrete Location Theory*, pages 1–54. John Wiley&Sons, 1990.

[49] A. A. Kuehn and M. J. Hamburger. A heuristic program for locating warehouses. *Management Science*, 9:643–666, 1963.

[50] C. Lamboray and D. Vanderpooten. Lexicographic α–robustness: An alternative to min–max criteria. *European Journal of Operational Research*, 220:722–728, 2012.

[51] G. Laporte, F. V. Louveaux, and L. Van Hamme. Exact solution to a location problem with stochastic demands. *Transportation Science*, 28(2):95–103, 1994.

[52] Gino J. Lim and Ayse Durukan Sonmez. γ-robust facility relocation problem. *European Journal of Operational Research*, 229(1):67–74, 2013.

[53] C. K. Y. Lin. Stochastic single-source capacitated facility location model with service level requirements. *International Journal of Production Economics*, 117(2):439–451, 2009.

[54] François V. Louveaux. Discrete stochastic location models. *Annals of Operations Research*, 6:23–34, 1986.

[55] François V. Louveaux. Stochastic location analysis. *Location Science*, 1:127–154, 1993.

[56] François V. Louveaux and D. Peeters. A dual-based procedure for stochastic facility location. *Operations Research*, 40(3):564–573, 1992.

[57] Maria Céu Marques and Joana Matos Dias. Dynamic location problem with uncertainty: a branch&bound approach. In José F. Oliveira and Clara B. Vaz, editors, *Proceedings XVI Congresso da Associação Portuguesa de Investigação Operacional*, pages 233–243. Instituto Politécnico de Bragança, Bragança, Portugal, 2013.

[58] Maria Céu Marques and Joana Matos Dias. Simple dynamic location problem with uncertainty: a primal-dual heuristic approach. *Optimization*, 62(10):1379–1397, 2013.

[59] Maria Céu Marques and Joana Matos Dias. Dynamic location problem under uncertainty with a regret based measure of robustness. *(submitted)*, 2014.

[60] P. B. Mirchandani and R. L. Francis. *Discrete Location Theory*. John Wiley&Sons, 1990.

[61] P. B. Mirchandani and A. R. Odoni. Location of medians on stochastic networks. *Transportation Science*, 13:85–97, 1979.

[62] Yin Mo and Terry P. Harrison. A conceptual framework for robust supply chain design under demand uncertainty. *Supply Chain Optimization*, pages 243–263, 2005.

[63] J. M. Mulvey, R. J. Vanderbei, and S. A. Zenios. Robust optimization of large-scale systems. *Operations Research*, 43(2):264–280, 1995.

[64] Stefan Nickel and Francisco Saldanha da Gama. Multi-period facility location. In Gilbert Laporte, Stefan Nickel, and Francisco Saldanha da Gama, editors, *Location Science*, pages 289–310. Springer International Publishing, 2015.

[65] Stefan Nickel, Francisco Saldanha da Gama, and Hans-Peter Ziegler. A multi-stage stochastic supply network design problem with financial decisions and risk management. *Omega*, 40:511–524, 2012.

[66] S. H. Owen and M. S. Daskin. Strategic facility location: A review. *European Journal of Operational Research*, 111:423–447, 1998.

[67] Feng Pan and Rakesh Nagi. Robust supply chain design under uncertain demand in agile manufacturing. *Computers & Operations Research*, 37:668–683, 2010.

[68] Bruno S. Pimentel, Geraldo R. Mateus, and Franklin A. Almeida. Stochastic capacity planning in a global mining supply chain. *Computational Intelligence in Production And Logistics Systems (CIPLS), 2011 IEEE Workshop On*, pages 1–8, 2011.

[69] R. Ravi and A. Sinha. Hedging uncertainty: approximation algorithms for stochastic optimization problems. *Lecture Notes in Computer Science*, 3064:101–115, 2004.

[70] Colin R. Reeves. *Modern Heuristic Techniques for Combinatorial Problems.* Blackwell Scientific Publications, Oxford, 1993.

[71] C.S. Revelle, H.A. Eiselt, and M.S. Daskin. A bibliography for some fundamental problem categories in discrete location science. *European Journal of Operational Research*, 184:817–484, 2008.

[72] R. T. Rockafellar and R. J-B Wets. Scenarios and policy aggregation in optimization under uncertainty. *Mathematics of Operations Research*, 16:119–147, 1991.

[73] M. Romauch and R. F. Hartl. Dynamic facility location with stochastic demands. *Lecture Notes in Computer Science*, 3777:180–189, 2005.

[74] Gary M. Roodman and Leroy B. Schwarz. Optimal and heuristic facility phase-out strategies. *AIIE Transactions*, 7(2):177–184, 1975.

[75] Gary M. Roodman and Leroy B. Schwarz. Extensions of the multi-period facility phase-out model: new procedures and application to a phase-in/phase-out problem. *AIIE Transactions*, 9(1):103–107, 1977.

[76] Jonathan Rosenhead, Martin Elton, and Shiv K. Gupta. Robustness and optimality as criteria for strategic decisions. *Operational Research Quarterly*, 23(4):413–431, 1972.

[77] T. Ross and R. Soland. A multicriteria approach to the location of public facilities. *European Journal of Operational Research*, 4:307–321, 1980.

[78] Bernard Roy. Robustness in operational research and decision aiding: A multi-faceted issue. *European Journal of Operational Research*, 200:629–638, 2010.

[79] Paul J. H. Schoemaker. When and how to use scenario planning: A heuristic approach with illustration. *Journal of Forecasting*, 10:549–564, 1991.

[80] D. Serra and V. Marianov. The p-median problem in a changing network: the case of Barcelona. *Location Science*, 6:383–394, 1998.

[81] Jeremy F. Shapiro. Challenges of strategic supply chain planning and modeling. *Computers and Chemical Engineering*, 28:855–861, 2004.

[82] L. V. Snyder. Facility location under uncertainty: a review. *IIE Transactions*, 38:537–554, 2006.

[83] L. V. Snyder and M. S. Daskin. Stochastic p–robust location problems. *IIE Transactions*, 38:971–985, 2006.

[84] A. D. Sonmez and G. J. Lim. A decomposition approach for facility location and relocation problem with uncertain number of future facilities. *European Journal of Operational Research*, 218:327–338, 2012.

[85] J. E. Torres-Soto and H. Uster. Dynamic-demand capacitated facility location problems with or without relocations. *International Journal of Production Research*, 1:1–27, 2010.

[86] B. Urli and R. Nadeau. Promise/scenarios: An interactive method for multiobjective stochastic linear programming under partial uncertainty. *European Journal of Operational Research*, 155:361–372, 2004.

[87] Kee Van der Heijden. *Scenarios: The art of Strategic Conversation.* John Wiley&Sons, 1996.

[88] T. J. Van Roy and D. Erlenkotter. A dual-based procedure for dynamic facility location. *Management Science*, 28(10):1091–1105, 1982.

[89] Xing Wang, Dachuan Xu, and Xinyuan Zhao. A primal-dual approximation algorithm for stochastic facility location problem with service installation costs. *Frontiers of Mathematics in China*, 6(5):957–964, 2011.

[90] J. R. Weaver and R. L. Church. Computational procedures for location problems on stochastic networks. *Transportation Science*, 17(2):168–180, 1983.

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