UNIVERSIDAD AUTÓNOMA DE NUEVO LEÓN
FACULTAD DE CIENCIAS FÍSICO MATEMÁTICAS

TESIS

AN EXACT ALGORITHM TO SOLVE A CLOUD SERVICE BILEVEL PROBLEM

PRESENTA

LILIAN LOPEZ VERA

COMO REQUISITO PARA OBTENER EL GRADO de MAESTRÍA EN CIENCIAS CON ORIENTACIÓN EN MATEMÁTICAS

AGOSTO, 2018

UNIVERSIDAD AUTÓNOMA DE NUEVO LEÓN

FACULTAD DE CIENCIAS FÍSICO MATEMÁTICAS

CENTRO DE INVESTIGACIÓN EN CIENCIAS FÍSICO MATEMÁTICAS

TESIS

AN EXACT ALGORITHM TO SOLVE A CLOUD SERVICE BILEVEL PROBLEM

PRESENTA

LILIAN LOPEZ VERA

COMO REQUISITO PARA OBTENER EL GRADO DE MAESTRÍA EN CIENCIAS CON ORIENTACIÓN EN MATEMÁTICAS

SAN NICOLÁS DE LOS GARZA, NUEVO LEÓN, AGOSTO DE 2018

Autonomous University of Nuevo Leon

Faculty of Physical-Mathematical Sciences

Center for Research in Physical-Mathematical Sciences

The committee members, hereby, certify that have read the thesis presented by Lilian Lopez Vera and that it is fully adequate in scope and quality as a requirement for the degree of Master in Science with Major in Mathematics.

Dr. José Fernando Camacho Vallejo
Principal Advisor

Dr. Omar Jorge Ibarra Rojas
Committee Member

Dr. Alvaro Eduardo Cordero Franco
Committee Member

Vo. Bo.

Dr. Francisco Javier Almaguer Martínez
Coordinator of the Graduate Programs in Science with Major in Mathematics

San Nicolás de los Garza, Nuevo León, México
August 2018

We are at our very best, and we are happiest when we are fully engaged in work we enjoy on the journey toward the goal weve established for ourselves. It gives meaning to our time off and comfort to our sleep. It makes everything else in life so wonderful, so worthwhile.

Earl Nightingale

Acknowledgements

I would like to thank the Autonomous University of Nuevo Leon (UANL), the Faculty (Department) of Physical-Mathematical Sciences (FCFM) and the Center for Research in Physical-Mathematical Sciences (CICFIM) for all the support provided during my postgraduate course. Furthermore, it is my pleasure to extend my gratitude to the the Mexican National Council for Science and Technology (CONACYT) for the supporting scholarship and the mixed scholarships for my research stay held at the French National Institute for Research in Computer Science and Control (INRIA).

Contents

Chapter 1

Introduction

Technology is a word used to describe the way humans are changing their lives by materializing and joining knowledge. Moreover, despite many people believe that technology has already changed generations and we are no longer a race that could survive without it, [7] showing significant ethical and social consequences . It is also proved that because all this materialization and join of different scientific aspects, is that we have reached a point where we have touched everything around us. We had been able to improve life quality, health, the way we understand our surroundings, facilitated traveling, and our most important, the fact that we can communicate no matter where we are.

Despite technological advances entirely run our lives, when we talk about this topic, all that comes to our mind is one specific technological promoter: computers. And it is not something to be impressed, because since their appearance, they have impulsed all advances and scientific branches in almost all the fields.

In the past 80 years, with the creation of modern computers and the invention of the internet, life as our forefathers knew it has been rapidly and continuously changing [19].
Being surrounded by this technology and the aim to improve work, communication, and life, also make us focus our attention on the future and the new challenges that arise for the operational research field, which emerged from a military context as a way of analyzing and solving specific real based problems. Operational Research with the use of mathematical logic and methods followed by a series of defined steps and based on their specific purpose can find or approach the best possible solution for a certain decision maker [20]. Operational Research is also evolving, and day by day, finding better ways to approach an optimal solution for each proposed problem.
Moreover in operational research and focusing on this exciting information technology field, we get to the center of our study: Cloud Computing.

According to [10] cloud computing may be defined as clusters of computers which provide on-demand resources and services over a network with the reliability of a data center offering two types of clouds: those that provide computing instances on demand and those that provide computing capacity on demand. Using similar machines but each one designed and used for different purposes.

Cloud computing who has had a notable boom since the 2000's, but its creation happened 40 years before, in the 1960's. Emerged as a fastest, strongest, and safest way of computing, but it was not an affordable option for the population or even the small to medium businesses (for the masses). It was a resource that just companies with a vast economic capacity could afford. And, just around the early 2000's companies as Amazon Web Services realize that they could offer the service of renting cloud space to others as a big trade-off between them. In 2006 they launched the first widely accessible cloud computing infrastructure service, which will allow small companies and individuals to rent computers on which to run their computer applications. The latter was a significant discovery that was able to find a huge and interesting gap in the market. Which was soon followed by other big companies, such as Google.

That means we are dealing with a major discovery that is changing our lives and facilitating work. However, it is not easy to set a price or a policy price for a virtual service, especially when it has to take into account that the price does not just influence customers, they get influenced in a certain proportion by a set of characteristics known as Quality of Service (QoS) [12], [21]. Which even though the price is important for the customers, these QoS characteristics are a way how they see reflected if a service is worth it or not. Moreover, even if a company offers a reasonable price, the customers might decide to contract a more expensive service but with better quality. Where we see different points that may not be tangible will make a high impact on the company's revenue.

This QoS is defined for the company as specific objectives to optimize. Which might be: the energy consumption, the response time, the robustness which is vital to know if a failure happens or the dynamism, among others. [11]

Based on this idea, how could we establish a fair price, that could lead the company to maximize its revenue, but also to guarantee a safe environment for the customers?

1.1 Problem description

With this background, the problem we address is the following.
If we analyze the situation from the Cloud companys point of view, which seeks to set a price scheme to maximize their profits based on specific characteristics such as energy consumption, different time windows in which he can offer the service, different capacity plans or levels to offer and the whole capacity of their servers. On the other hand, customers. Those who based on a specific consumption necessity, called demand, seek to get the best possible price. While this happens, they are being influenced by a pre-established fixed price scheme, which varies based on different levels that the company offers for the different time windows. This price scheme will make them individually evaluate all the possible solutions in which their demand is fully satisfied, which we will call feasible solutions. When a customer takes a decision and chooses a specific contract with the company, this will immediately impact the quality of the service of all customers as an additional weight that occurs at certain times,

which will indirectly affect current customers. Moreover, with a more significant measure, the future customers. This price influences customers, however, once they find a better price in a feasible solution, they evaluate the performance of the service of that option and decide if it is convenient to contract.

The situation is represented as an optimization problem, to be specific as a bilevel mathematical programming problem. Which is defined as an optimization problem, where in order to be solved has to satisfy a group of constraints. But, their particular characteristic is that as a constraint is allocated an additional optimization problem. Which will be an additional problem to solve, but whose decision will directly impact the main problem.

In this case, the cloud company will act as our main problem to solve, which will seek to establish a pricing scheme for its customers based on two main objectives: maximize the profit and offer the best possible quality of service. For the latter, we must set a specific weight to negatively impact the profit in order to define a global goal to improve, which will be focusing on both objectives.
Also, customers decision will be considered as a secondary problem, which will seek to minimize its cost based on the companys pre-defined pricing scheme.

1.2 Motivation and contribution

Bilevel programming since its appearance in the 30s is expanding the area of application of operations research and finding solutions to complex problems.

Inspired by [1] we found the complexity of pricing problems. As in other bilevel problems, it is essential to analyze how customers interact with a company and how their preferences and the company competitors can affect the main objective.

These pricing problems led us to cloud services, an exciting and modern problem, which its complexity is not just on the customer's interaction but in the pricing process by itself.
We soon realize that its complexity was due to the fact that it was a new multi-tenant paradigm, a not so exploded branch where the customer's perception and the performance of the service is also making an impact in the revenue.

This complex problem and taking [1] pricing idea plus the [11] quality of service propositions led us to our exciting case study.

1.3 Objective

The main goal of this thesis is to analyze the characteristics that are involved in the process of cloud computing. To propose a way of representing the importance of the Quality of Service (QoS) by taking the assumption that one of the most significant factors involved in the delay. Based on the available literature we propose a new bi-level programming model which proves that under certain assumptions it is possible to find the best solution for a specific problem.

By designing a new point of view that joins related topics with bigger literature, we expect to expand the pricing knowledge, but in particular the cloud computing comprehension.

1.4 Methodology

Our objectives to reach the thesis objectives were:

1. Studying how cloud computing works
2. A literature review of pricing in cloud computers
3. A literature review of bi-level programming
4. A literature review of bi-level pricing problems
5. Analyzing the most significant challenges and the main characteristics involved in cloud computing type of problems.
6. A literature review of quality of service problems.
7. Defining a particular case study and deciding a QoS point of view.
8. A literature review of delay functions.
9. In collaboration with Professor Luce Brotcorne, design a mathematical model for the case study.
10. Design and testing of an exact algorithm to reach an optimal solution
11. Develop instances according to the case study needs
12. Run and analyze the experiments in order to test and fix an accurate delay weight.
13. Analyze results to show advantages of the model.
14. Presentation of advances in the XXII ENOAN and the IWOBIP 18
15. Research internship in INRIA in Lille, France.

1.5 Structure of the thesis

Chapter 1 was an introduction to our problem where the importance of our case study was presented and the main goals and objectives of this thesis.

On chapter 2 a complete literature review is presented to study and analyze all related aspects of our case study.
Starting with the study of cloud computing and price settings; followed by the way we are presenting our problem by reviewing bi-level programming and bi-level pricing problems. In the end, we review bi-level cloud problems.

Chapter 3 is used to define our mathematical model by presenting a complete description of the problem followed by the formulation and the model's characteristics.

In chapter 4, a detailed description of the algorithm is shown, including a complete explanation of what motivates this algorithm and why is chosen as the best possible way to tackle the problem. The pseudocode is presented followed by an illustrative example of its functioning.

On chapter 5 computational experimentation is developed, where we describe the design of our instances sd well as the computational environment where they were developed and tested. Results are shown, followed by their graphics.

Chapter 6 contains the conclusions if the thesis and proposes ideas for a possible problem extension and future work.

Chapter 2

Literature Review

Doing a deep search in cloud computing literature, we found that the first published Cloud related work was [17] in 1999. The Condor project used the idea of managing in a more efficient way a set of resources, which were going to run certain parallel jobs. This research consisted in having a central manager that was in charge of administrating the resources, verifying the availability, and distributing the work. They used a distributed matching algorithm to compare requirements and possibilities, which will send an offer association to schedule the best schedule.

2.1 Cloud computing

Cloud computing was just an idea of a set of clusters working together over a network, to provide on demand resources. The first literature works started without a standard definition. Studies where basing their work on basic computer and internet concepts [19],[10], where they recognized the importance of computers and its future through the internet. The only concepts able to relate, were the basic ideas established in [11]. This papers gave the idea behind cloud computing, and even they where not using this term, the interest was on the on demand resources.
Analyzing the general idea and looking at a business perspective[15], understanding the industrial needs [6] [7] this on-demand network started to grow.

Once basic concepts of cloud computing where defined [16], companies started to find a new gap in the market and looking at the revenue opportunities. [15]established that the main issue with Cloud was the need of understanding the issues involved, and to be able to analyze the perspective of providers and consumers. The complexity of the topic is on the multiple characteristics involved in the cloud working process plus the idea of focusing on a business perspective.

Since the early 2000's the first pricing works started to emerge. Identifying the opportunity offered, works as [17]review the connection between industry and the factors that influence the price. In this review work is analyzed how customers are able to limit and influence the demand by doing a research in a real life environment. It was analyzed the influence of customers. By the other hand [13]literature research work, focus on the factors that have been identified in published works as important in the process of spot-price schemes or fixed-prices schemes. The latter found important benefits of using the concept of spot pricing in the cloud area such as a control power over a demand, saving infrastructure cost, having a better revenue and improving the resource utilization. As a big limitation for spot pricing is the robustness of the mechanisms that are needed, the generated loss revenue that can be caused by a price discrimination, and the untruthful and mutual cooperation that can tour unprofitable for cloud providers. The latter work, opened the idea of reviewing benefits of spot pricing and looking at the concerns, which lead to finding [1] work. This Energy work offered the idea of mixing benefits from both pricing schemes spot and fixed. The work proposed a planning scheme where based on the consumption necessities, the prices where fitted and selected in advance. But different to fixed planning, it possesses a changing planning scheme, that will plan in advance the price for a certain period of time, but gives the opportunity to modify the prices on the go.

To be able to establish different pre-defined prices and to be able to point the risks of having saturated servers, picewise functions are being used. [14], [4] show how prices can be influenced and stabilized over time to minimize peaks in consumption. Customers that posseses the flexibility to select a different working time, can be influenced by this pricing schemes so the peak loads can be minimized to create stability in the company.

2.2 Cloud computing QoS

Quality of service reviews the factors that create the biggest impact in the way service is being interpreted by customers. This influence their reaction and as a consequence, the decisions they will take [8]. The customers feeling of the service will create an indirect positive or negative impact in the revenue.

It is important to analyze QoS and be aware of the considerations models should have, to create a realistic model that companies could trust. Preventing damaged services can be identified as a important pricing strategy for the cloud providers [12].

Regarding quality of service in cloud companies, we are able to see the arising number of studies reviewing the dynamics of the service [9], the impact in the price [21], or the way market is being influenced [18].

As a big influence in quality of service, several works as [11] shown the importance of focusing on the delay customers may experience. We were not able to identify a only way of modeling this problems. [3] defined a price-based revenue management problem which goal is to stablish an optimal policy but they model it as a Márkov process. In contrary, [5] model, makes a balance between energy efficiency and QoS, having as a main focus and goal to maintain the system performance by using a Gaussian process. The last contrasting model was [9], who uses Nash equilibrium and Wardrop equilibrium. They assume the company will always have sufficient servers, so they can be able to focus on price service plus a small delay cost.

2.3 Bilevel Programming

Bi level programming as a particular branch of operational research, offers the possibility to solve complex problems. The first bilevel formulation was in 1934 by Heinrich Von Stackelberg. In Stackelberg economy problem, he found that it was possible to represent a problem by defining a hierarchical model where two different decision makers were causing an impact in the main objective.

The work consisted in setting two different decision makers, where each one of them posses its own objective and direct independence with the other decision maker. With this idea he created the concept of having a main optimization problem, with their own associated constraints. As a constraint of the main problem (Leader), is allocated a second optimization problem (Follower) with its own associated constraints. Stackelberg defined this hierarchical model where for a particular movement of the upper level, the lower level was answering its problem. This created a particular lower level reaction for each possible decision[20].

Chapter 3

Mathematical Model

In this section, the bilevel problem and the mathematical formulation are described.

3.1 Problem statement

For this problem, we consider a situation in which a cloud company who offers their computing service aims to set up a pricing scheme to satisfy a set of potential customers. Customers are characterized by a demand to guarantee and a group of specifications. Each one of them possesses different objectives whose goals conflict with them. The company will seek to maximize its revenue while the customers aim to minimize their cost.

This problem can be modeled and solved as a mathematical bilevel programming model. Which involves two decision levels, where company and customers could fit. An upper level associated with a certain leader of the problem is designed for the cloud company. Later, the set of potential customers at a lower level, associated to a follower that will react based on the leader decisions.

Also, certain considerations have to be taken into account.

On the upper level:

- The leader will set a time planning scheme.
- A certain group of capacity levels has to be defined for all the planning scheme; each one of them containing a different number of instructions.
- The leader will define in advance a set of prices associated with each capacity level during the planning scheme.
- There is a maximum server capacity that the leader posses.
- The cloud company must take care of the service environment by trying to give the best possible service.

On the lower level:

- The individual demand must be guarantee.
- Customers will have only one contract each, which means; we will only consider the cases where the work starts, and it is not interrupted until the demand is satisfied.

3.2 Mathematical model

To solve the pricing problem, we established, the leader aims to maximize its revenue but taking into account quality of service functions that might affect the performance of the service and impact the revenue.
The follower, attempt to minimize their global cost based on the leader pricing scheme and their consumption necessities.
The sets, parameters, and decision variables involved in the mathematical formulation are presented next.

Sets

I: set of all the customers.

T: time windows.

M: set of all the physical machines.

K: set of the capacity levels offered by the company.

Parameters

d_i : the demand of customer $i \in I$.

c_m: maximum capacity of the physical machine $m \in M$.

$\phi^t(x, n)$: a function depending on x and n used to define the delay.

λ : a fixed delay weight that will reflect the importance of decreasing the delay.

Decision variables

For the leader

p^{kt} : price set to use the resource of capacity level k in period t.

$$q^{kt} = \begin{cases} 1, & \text{if capacity level } k \text{ is available in period } t \\ 0, & \text{otherwise} \end{cases}$$

And, for the follower

$$x_i^{kt} = \begin{cases} 1, & \text{if customer } i \text{ uses capacity level } k \text{ in period } t. \\ 0, & \text{otherwise} \end{cases}$$

n_i^t : number of instructions the customer i is using in the period t.

Before we set a mathematical model, based on [11] idea, we stablished $\phi^t(x, n)$ function. We decided to base our QoS function on the response time, which is considered as the execution time of the service. Knowing the maximum capacity of each physical machine m in each period of time t, and the total demand of the actual customers, we are able to define:

$$\phi^t = \frac{\sum\limits_i x_i^t n_i^t}{\sum\limits_m c_m^t} \tag{3.1}$$

The mathematical model is as follows:

$$\max_{p,q} \sum_k \sum_t p^{kt} \sum_i x_i^{kt} - \lambda \sum_t \frac{\sum\limits_i x_i^t n_i^t}{\sum\limits_m c_m^t} \Big(\sum_k q^{kt} \sum_i x_i^{kt} \Big) \tag{3.2}$$

$$\text{subject to:} \quad 0 \leq p^{kt} \leq p_{max}^t \quad \forall k, t \tag{3.3}$$

$$q^{kt} \in \{0, 1\} \quad \forall k, t \tag{3.4}$$

$$x \in arg \min_x \sum_k \sum_t p^{kt} \sum_i x_i^{kt} \tag{3.5}$$

$$\text{subject to:} \quad \sum_k \sum_t q^{kt} x_i^{kt} \geq D_i \quad \forall i \tag{3.6}$$

$$\sum_k x_i^{kt} = 1 \quad \forall t, i \tag{3.7}$$

$$x_i^{kt} \in \{0, 1\} \quad \forall k, t, i \tag{3.8}$$

In the latter model, equation (3.2) represents the leader's objective function is represented by the total income minus a delay factor translated into money which is increased according to the number of customers using the service of the same capacity in the same period of time. Having this into consideration the leader's objective function aims to maximize the profit obtained by setting a pricing policy. Equation (3.3) sets an upper bound on the prices. In the follower's objective function given by equation (3.5), the customers try to minimize their global cost. Equation (3.6) guarantees total customer demand. Equation(3.7) ensures that there is only one contract per customer in each time period. Finally, equations (3.4) and (3.8) corresponds to the binary constraints for the decision variables.

Chapter 4

Algorithm

In this chapter, we present the algorithm used to solve the bilevel problem established in Chapter 3.

4.1 Algorithm

In this section, an exact algorithm is proposed.
For bilevel problems where the follower variables affect the leader's objective function, first, we should focus on solving that lower level to optimality to then complete the upper level.
The proposed algorithm consists of a specific pricing scheme associated with capacity levels through a planning period that will last a fixed number of time windows.
Once the complete set of information is known, each customer will compute the set of all the possible, feasible solutions and find the minimum one.
Because we are assuming an optimistic approach, when a customer finds multiple minimum solutions, the lower level will evaluate the delay on the leaders objective function and decide the option that minimizes that delay.

Solution Encoding:
The solution is represented as a collection of $2t$ strings, where each pair of strings represents the number of instructions associated with each customer per period of time.
The number of the pair will indicate the time window t, the position in the pair of strings will indicate the customer i. The first pair in the string will indicate the capacity level k associated to i, and the second string will indicate the selected number of units of that capacity level k.

$Figure$ 4.1 will show the algorithm developed to solve the problem, and the method will be detailed next:

Step 1 Read the parameters, find the planning scheme and construct the empty strings.

Step 2 Take a new i customer and evaluate the demand associated to it $D[i]$.

Step 3 Based on the available capacity levels and the planning scheme, compute all the possible, feasible combinations that can satisfy the demand. $\sum_k \sum_t q^{kt} x_i^{kt} \geq D_i$.

If a combination contains two capacity levels in the same time window, it will be discarded; Also, if a combination leaves empty time windows between two selected ones, the combination will be discarded.
A feasible solution must include only one capacity level associated to each period of time, and it must contain consecutive time windows working. If a work starts in t it must continue until the demand is satisfied.

Step 4 Find the associated Price to each feasible possibility and sort by Price.

Step 5 Select the solution associated to the mnimum Price.
If more than one mnimum solution is found, the options will be evaluated in the leader's objective function and select the one that generates the mnimum delay.

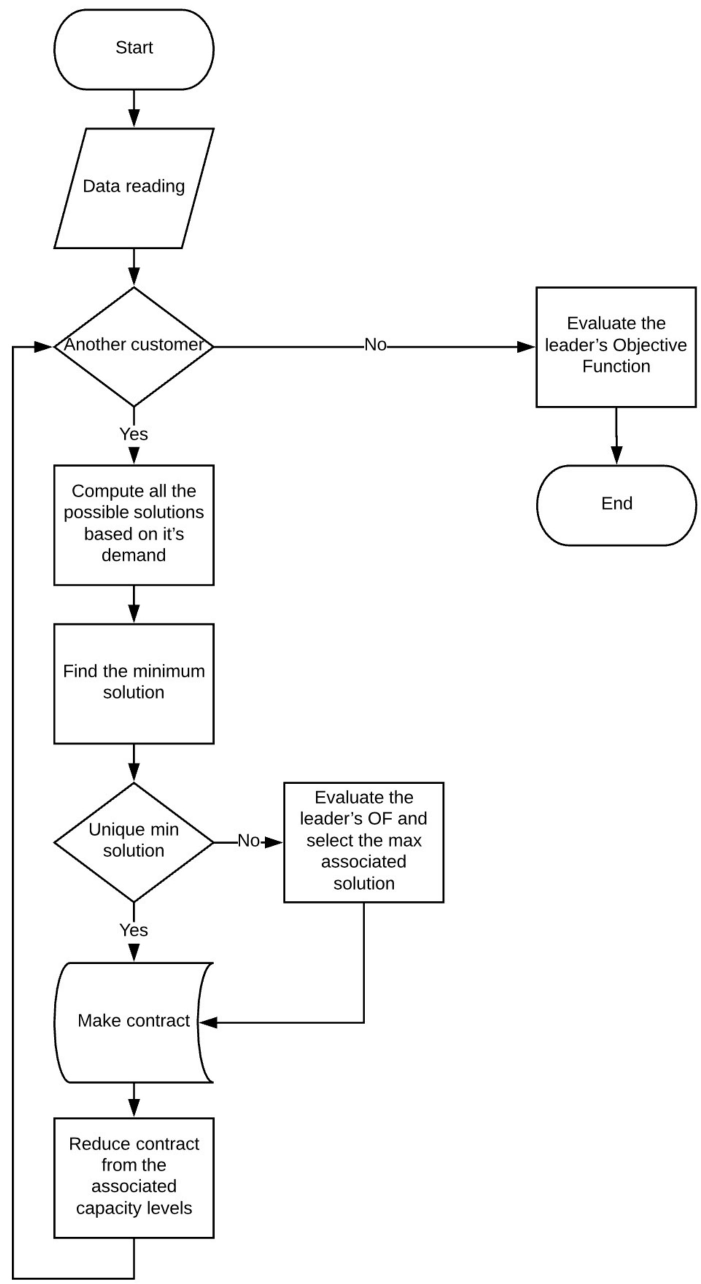

Figure 4.1: Diagram

Chapter 5

Computational Experimentation

Computational experimentation was conducted to evaluate both scenarios modeled over a set of 30 instances. The first 12 instances are obtained from the real case data presented in [?]. The remaining 18 instances are adapted from the mentioned real instances taking as a basis the existing data patterns. The size of all the instances are depicted in Table 5.1. Instances are denoted as Prob.X (R,D,P,T,S), in which X represents the instance number, R represents the number of potential retailers, D represents the number of potential distribution centers, P denotes the number of potential plants, T denotes the number of available technologies, and S represents the number of potential suppliers. Therefore, each instance's label indicates the size of the SCND under consideration.

Table 5.1: Size of the instances

Small-size	Medium-size	Large-size
Prob.1(5,3,3,2,4)	Prob.11(15,11,10,8,4)	Prob.21(24,18,15,12,24)
Prob.2(6,4,4,3,5)	Prob.12(18,12,8,12,16)	Prob.22(25,18,16,13,24)
Prob.3(7,4,4,3,6)	Prob.13(18,13,9,12,17)	Prob.23(26,18,17,13,25)
Prob.4(8,5,5,4,7)	Prob.14(19,13,9,12,18)	Prob.24(26,19,18,14,26)
Prob.5(9,5,6,4,8)	Prob.15(19,14,10,12,9)	Prob.25(27,20,19,14,26)
Prob.6(9,6,7,5,9)	Prob.16(20,15,11,12,19)	Prob.26(28,20,20,14,27)
Prob.7(10,7,7,6,9)	Prob.17(21,15,12,12,20)	Prob.27(29,21,21,14,27)
Prob.8(11,8,8,7,10)	Prob.18(21,16,13,12,21)	Prob.28(29,22,22,14,27)
Prob.9(12,9,8,7,12)	Prob.19(22,16,13,12,22)	Prob.29(30,22,23,14,28)
Prob.10(14,10,9,8,12)	Prob.20(23,17,14,12,23)	Prob.30(30,23,24,14,29)

All the instances were solved by the proposed SA-based heuristic algorithm, which is coded in GAMS-IDE on a system with configuration ASUS X450L corei5 with 8GB RAM. Due to stochasticity involved in the proposed algorithm, five independent runs were performed for each instance. The results obtained from solving scenario 1 and 2 are shown in tables 5.2 - 5.4 and 5.5 - 5.7, respectively. In these tables, the best value (BEST) of the leader objective function, the average (AVG) among the five runs, the standard deviation (STDV), and the average of required time (AVG TIME) are presented.

Table 5.2: Results for the small-size instances under scenario 1

Instance	BEST	AVG	STDV	AVG TIME (s)
Prob.1(5,3,3,2,4)	6.007E+10	6.007E+10	0.000E+00	238.21
Prob.2(6,4,4,3,5)	7.783E+10	7.874E+10	2.044E+09	247.74
Prob.3(7,4,4,3,6)	1.178E+11	1.185E+11	1.469E+09	612.14
Prob.4(8,5,5,4,7)	9.791E+10	1.057E+11	4.473E+09	549.06
Prob.5(9,5,6,4,8)	1.342E+11	1.353E+11	2.039E+09	829.98
Prob.6(9,6,7,5,9)	9.030E+10	9.142E+10	1.504E+09	478.42
Prob.7(10,7,7,6,9)	1.144E+11	1.226E+11	7.589E+09	306.40
Prob.8(11,8,8,7,10)	1.643E+11	1.676E+11	1.946E+09	677.19
Prob.9(12,9,8,7,12)	1.720E+11	1.746E+11	3.461E+09	711.78
Prob.10(14,10,9,8,12)	2.159E+11	2.162E+11	4.144E+08	838.65

Table 5.3: Results for the medium-size instances under scenario 1

Instance	BEST	AVG	STDV	AVG TIME (s)
Prob.11(15,11,10,8,4)	2.114E+11	2.116E+11	2.283E+08	698.57
Prob.12(18,12,8,12,16)	2.764E+11	2.780E+11	2.288E+09	917.18
Prob.13(18,13,9,12,17)	2.917E+11	2.948E+11	3.749E+09	897.38
Prob.14(19,13,9,12,18)	3.217E+11	3.270E+11	3.186E+09	1115.83
Prob.15(19,14,10,12,9)	2.763E+11	2.833E+11	8.435E+09	1246.42
Prob.16(20,15,11,12,19)	2.929E+11	3.028E+11	6.031E+09	784.54
Prob.17(21,15,12,12,20)	3.055E+11	3.139E+11	7.132E+09	931.63
Prob.18(21,16,13,12,21)	2.949E+11	2.986E+11	2.301E+09	1019.93
Prob.19(22,16,13,12,22)	3.594E+11	3.634E+11	4.121E+09	1098.66
Prob.20(23,17,14,12,23)	3.003E+11	3.063E+11	5.441E+09	1445.88

Table 5.4: Results for the large-size instances under scenario 1

Instance	BEST	AVG	STDV	AVG TIME (s)
Prob.21(24,18,15,12,24)	2.892E+11	2.927E+11	2.976E+09	1732.72
Prob.22(25,18,16,13,24)	3.769E+11	3.841E+11	4.813E+09	1370.77
Prob.23(26,18,17,13,25)	4.200E+11	4.268E+11	5.076E+09	1393.54
Prob.24(26,19,18,14,26)	3.906E+11	3.988E+11	6.119E+09	1426.73
Prob.25(27,20,19,14,26)	3.719E+11	3.824E+11	9.147E+09	2185.63
Prob.26(28,20,20,14,27)	3.891E+11	3.951E+11	6.003E+09	3006.15
Prob.27(29,21,21,14,27)	4.784E+11	4.910E+11	8.707E+09	3211.58
Prob.28(29,22,22,14,27)	4.040E+11	4.112E+11	8.506E+09	2332.20
Prob.29(30,22,23,14,28)	3.881E+11	3.965E+11	6.674E+09	2958.20
Prob.30(30,23,24,14,29)	4.732E+11	4.742E+11	8.080E+08	9138.54

Table 5.5: Results for the small-size instances under scenario 2

Instance	BEST	AVG	STDV	AVG TIME (s)
Prob.1(5,3,3,2,4)	6.111E+10	6.111E+10	0.000E+00	339.29
Prob.2(6,4,4,3,5)	6.905E+10	6.905E+10	0.000E+00	457.88
Prob.3(7,4,4,3,6)	1.094E+11	1.139E+11	4.772E+09	475.75
Prob.4(8,5,5,4,7)	9.742E+10	1.006E+11	4.502E+09	303.78
Prob.5(9,5,6,4,8)	1.147E+11	1.181E+11	2.679E+09	819.64
Prob.6(9,6,7,5,9)	8.826E+10	9.153E+10	2.566E+09	491.78
Prob.7(10,7,7,6,9)	1.023E+11	1.028E+11	3.022E+08	589.19
Prob.8(11,8,8,7,10)	1.223E+11	1.345E+11	8.059E+09	809.28
Prob.9(12,9,8,7,12)	1.447E+11	1.447E+11	0.000E+00	1216.77
Prob.10(14,10,9,8,12)	1.840E+11	1.927E+11	9.095E+09	1055.45

Table 5.6: Results for the medium-size instances under scenario 2

Instance	BEST	AVG	STDV	AVG TIME (s)
Prob.11(15,11,10,8,4)	1.719E+11	1.838E+11	1.191E+10	650.51
Prob.12(18,12,8,12,16)	2.097E+11	2.172E+11	5.164E+09	1065.82
Prob.13(18,13,9,12,17)	2.139E+11	2.258E+11	6.814E+09	577.60
Prob.14(19,13,9,12,18)	2.918E+11	2.972E+11	7.262E+09	923.93
Prob.15(19,14,10,12,9)	2.244E+11	2.287E+11	3.575E+09	1893.44
Prob.16(20,15,11,12,19)	2.444E+11	2.635E+11	1.227E+10	838.91
Prob.17(21,15,12,12,20)	2.401E+11	2.540E+11	8.382E+09	929.96
Prob.18(21,16,13,12,21)	2.480E+11	2.670E+11	1.442E+10	929.96
Prob.19(22,16,13,12,22)	2.860E+11	2.988E+11	9.984E+09	1019.30
Prob.20(23,17,14,12,23)	2.358E+11	2.455E+11	7.011E+09	1416.34

Table 5.7: Results for the large-size instances under scenario 2

Instance	BEST	AVG	STDV	AVG TIME (s)
Prob.21(24,18,15,12,24)	2.266E+11	2.379E+11	9.929E+09	2172.66
Prob.22(25,18,16,13,24)	3.318E+11	3.489E+11	1.406E+10	680.81
Prob.23(26,18,17,13,25)	3.995E+11	4.063E+11	3.869E+09	1313.87
Prob.24(26,19,18,14,26)	2.967E+11	3.122E+11	1.547E+10	1623.04
Prob.25(27,20,19,14,26)	2.975E+11	3.055E+11	6.389E+09	2119.33
Prob.26(28,20,20,14,27)	3.153E+11	3.226E+11	5.247E+09	2299.78
Prob.27(29,21,21,14,27)	3.903E+11	3.997E+11	9.056E+09	2717.95
Prob.28(29,22,22,14,27)	3.263E+11	3.350E+11	8.028E+09	2496.68
Prob.29(30,22,23,14,28)	3.039E+11	3.118E+11	5.390E+09	2505.21
Prob.30(30,23,24,14,29)	3.669E+11	3.742E+11	7.201E+09	4398.05

Regarding the results for scenario 1, it can be seen from table 5.2 that the SA-based algorithm obtains relatively small standard deviations, which implies that the algorithm performs steady. Moreover, for instance Prob.1(5,3,3,2,4), the same leader's objective function was obtained in all the five runs. This behavior is maintained for the medium and large-size instances. The required time was increased as it is expected, but it is still reasonable for a problem of this type.

On the other hand, under scenario 2 the results showed a similar performance of the SA-based algorithm. That is, small standard deviations were obtained for all instances, this is shown in figures 5.1-5.3. Moreover, the same leader's objective function value was obtained within the five runs for more instances than under scenario 1. The required time for scenario 2 was smallest than for scenario 1. A comparison is depicted in figure 5.7. Since supply chain network design corresponds to a strategic decision, the solution time is not very important to us, but high quality solutions are aimed.

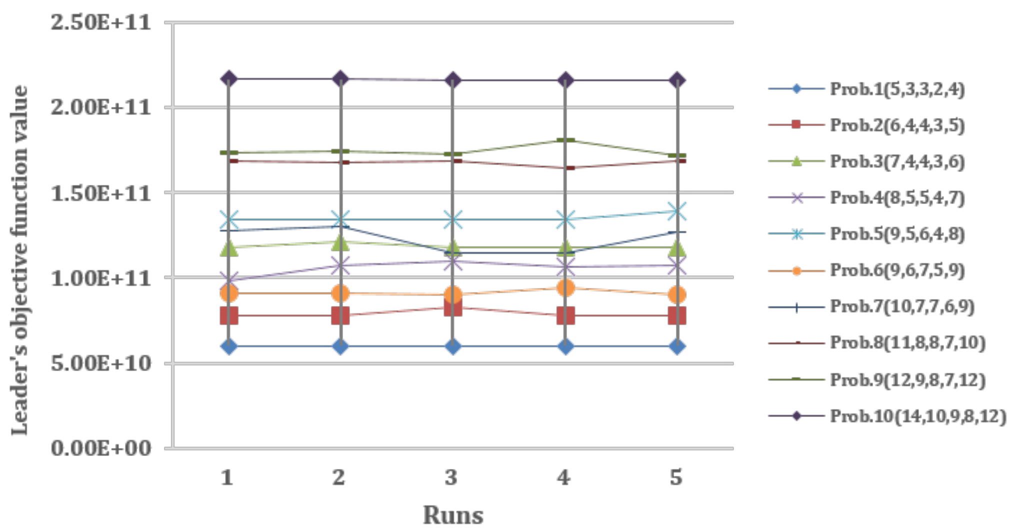

Figure 5.1: Leader's objective function value (gram CO_2) for small-size instances under scenario 1.

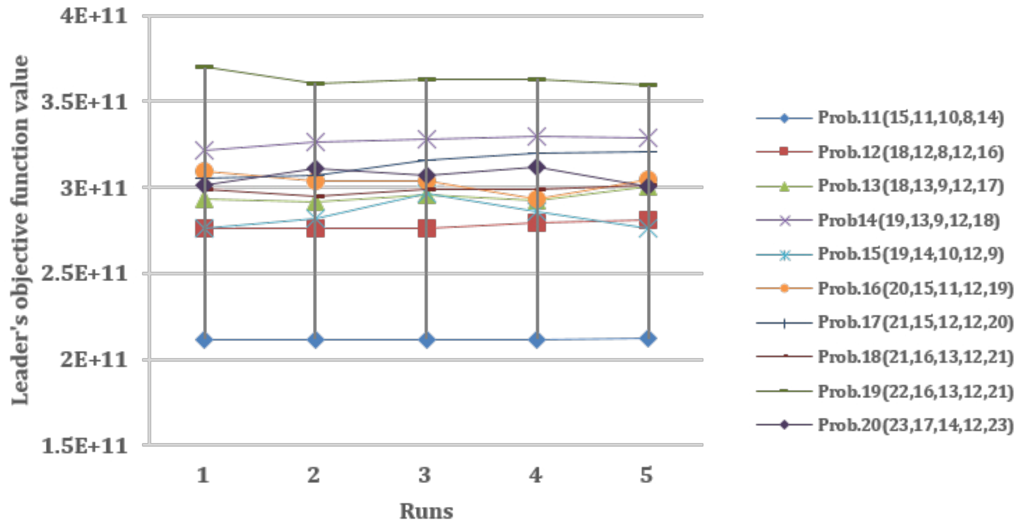

Figure 5.2: Leader's objective function value (gram CO_2) for medium-size instances under scenario 1.

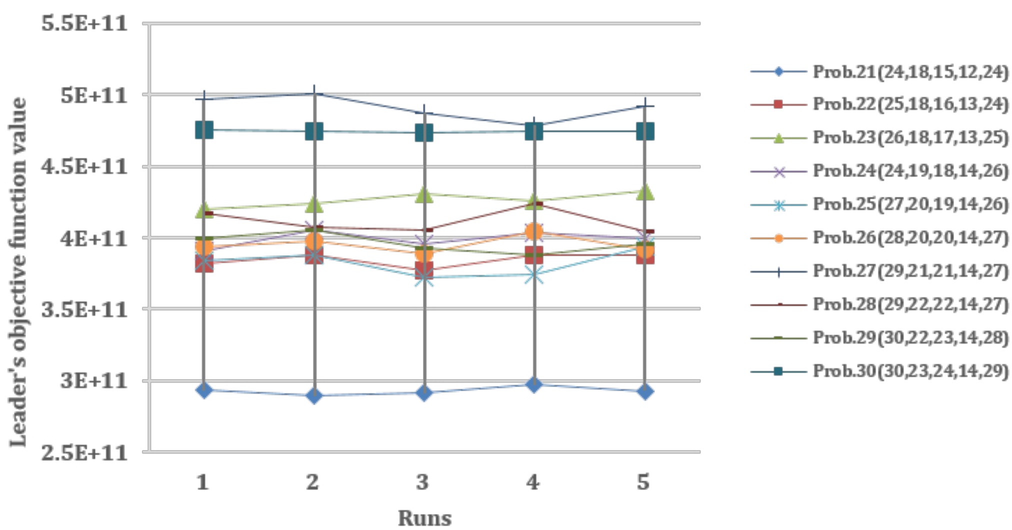

Figure 5.3: Leader's objective function value (gram CO_2) for large-size instances under scenario 1.

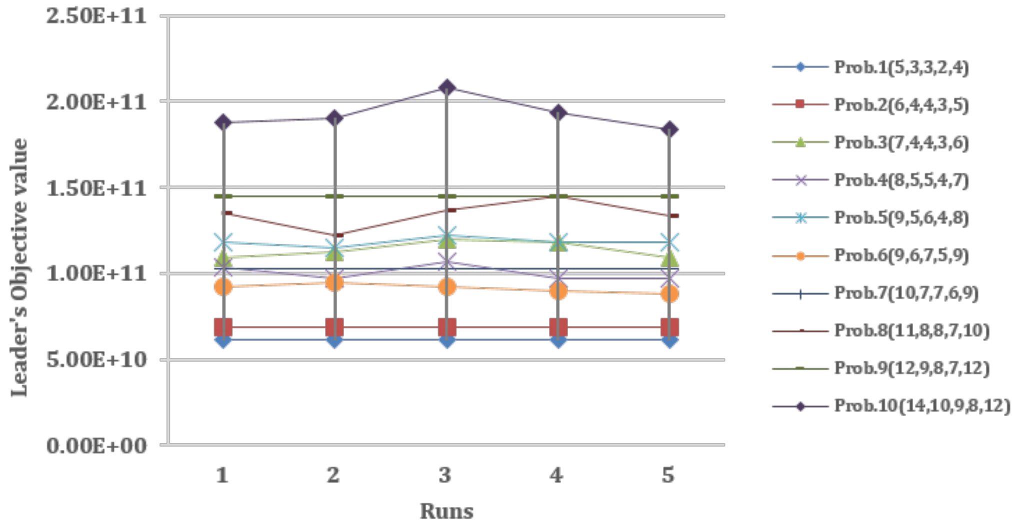

Figure 5.4: Leader's objective function value (gram CO_2) for small-size instances under scenario 2.

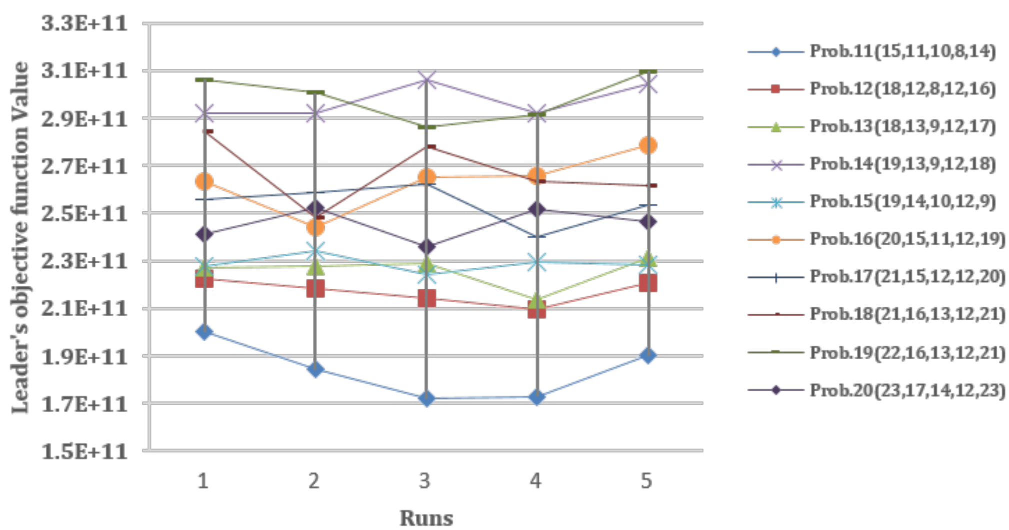

Figure 5.5: Leader's objective function value (gram CO_2) for medium-size instances under scenario 2.

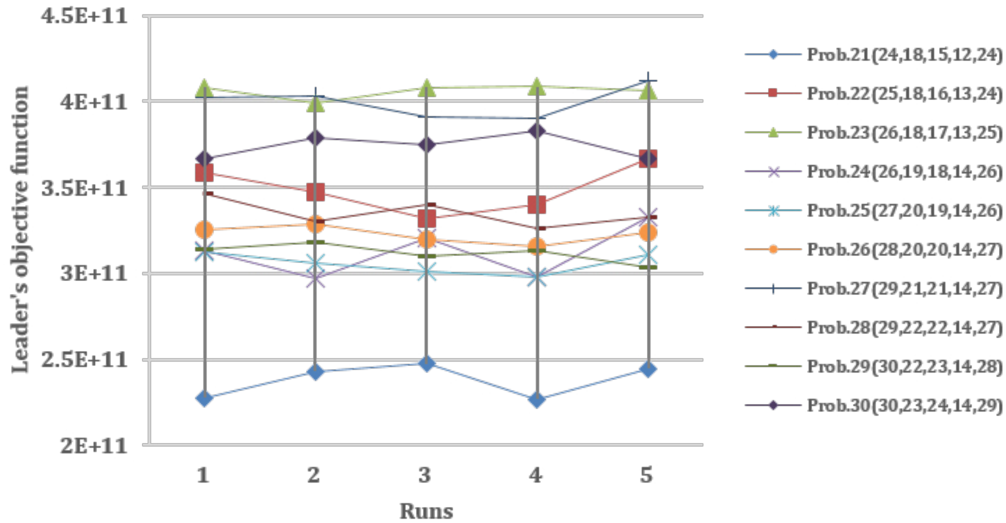

Figure 5.6: Leader's objective function value (gram CO_2) for large-size instances under scenario 2.

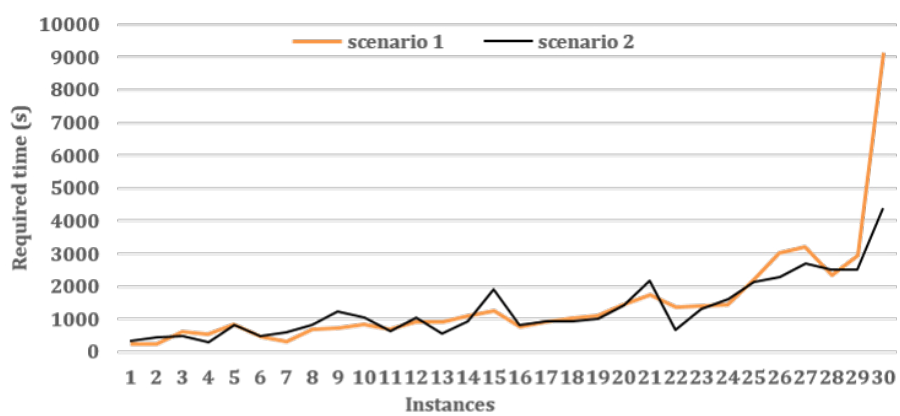

Figure 5.7: Average required time (in seconds) for both scenarios.

5.1 Validating the accuracy of the SA-based algorithm

To validate the quality of the results obtained by the proposed SA-based heuristic algorithm, a direct comparison against the results given by a full enumeration algorithm is conducted. The comparisons are depicted in tables 5.8 and 5.9 for scenarios 1 and 2, respectively. In these tables, the optimality gaps between both algorithms are computed. This comparison demonstrated that the proposed SA-based heuristic algorithm has a high efficiency and that it is capable to find optimal solutions accurately. However, this analysis is conducted only with the smallest instances due to computational limitations.

Table 5.8: Validating the accuracy under scenario 1

Instance	Full-Enumeration	SA-based algorithm	Gap (percentage)
Prob.1(5,3,3,2,4)	60,069,600,000	60,069,627,693	0.0000461
Prob.2(6,4,4,3,5)	77,826,400,000	77,826,402,996	0.00000004
Prob.3(7,4,4,3,6)	110,664,000,000	117,812,954,877	6.46

Table 5.9: Validating the accuracy under scenario 1

Instance	Full-Enumeration	SA-based algorithm	Gap (percentage)
Prob.1(5,3,3,2,4)	61,108,900,000	61,108,902,971	0.000005
Prob.2(6,4,4,3,5)	65,400,100,000	69,047,781,810	5.58
Prob.3(7,4,4,3,6)	87,231,100,000	98,048,819,437	12.401

From these two tables, it can be inferred that the obtained optimality gaps are very small and acceptable in almost all the cases. Under scenario 2, the optimality gap of the largest analyzed instance is not as small as the others, but this can be caused due to stochasticity involved in the proposed algorithm.

5.2 Analyzing the results for a particular instance

Now, let us assume an hypothetical supply chain that must be configured. Consider an instance with nine retailers, six distribution centers, seven plants, five production technologies, and nine suppliers. This instance was solved by using the proposed SA-based heuristic algorithm. The obtained solution dictates that technology 3 should be used in plant 6, technology 4 in plant 6, and technology 3 in plant 3. to receive financial incentives. Also, the designed network derived from the proposed SA-based algorithm is represented in figure 5.8. It can be seen that suppliers 1 and 3 are selected among the nine potential suppliers. Distribution centers 3, 4, and 6 are selected to satisfy the demands of all the retailers.

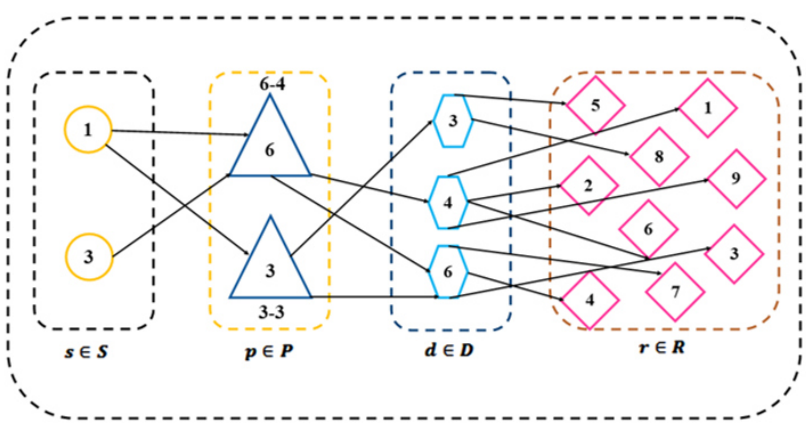

Figure 5.8: The schematic view of the optimal SCND for the aforementioned instance .

Parameters EO_{pt}, EM_{pt}, λ_{pt}, MC_{pt}, and TEC_{pt} used in the analyzed instance are displayed in tables 5.10-5.14. According to the solutions obtained by the proposed algorithm, the values of environmental and economic parameters show that these plants and technologies have less environmental impact (selected values are highlighted in bold). However, their opening and manufacturing costs are too high. On the contrary, by using the fake-knapsack problem and the ranking method explained in section 4, cleaner technologies will have higher scores leading to have more chances to receive government financial incentives. By using the proposed model, the environmental impact may be decreased in the SCND process.

Table 5.10: Parameter EO_{pt}, i.e., released emission of CO_2

	Technologies				
Plants	1	2	3	4	5
1	1,249,358	1,224,098	1,222,155	1,170,964	1,040,547
2	1,393,567	1,337,600	1,280,580	1,140,316	1,131,501
3	1,271,758	1,231,583	**1,100,082**	1,076,376	895,426
4	1,245,764	1,181,451	1,077,385	1,035,184	855,391
5	1,397,886	1,313,768	1,262,786	1,214,310	969,752
6	1,357,283	1,170,943	1,044,994	**984,291**	919,110
7	1,384,193	1,274,025	1,153,486	1,019,903	1,012,529

Table 5.11: Parameter EM_{pt}, i.e., released emission of CO_2

	Technologies				
Plants	1	2	3	4	5
1	1,222,155	1,224,098	1,170,964	1,040,547	1,249,358
2	1,393,567	1,140,316	1,131,501	1,337,600	1,280,580
3	1,271,758	1,076,376	**895,426**	1,100,082	1,231,583
4	1,035,184	1,181,451	1,245,764	1,077,385	855,391
5	1,214,310	1,262,786	1,397,886	969,752	1,313,768
6	984,291	1,170,943	1,044,994	**919,110**	1,357,283
7	1,012,529	1,153,486	1,384,193	1,019,903	1,274,025

Table 5.12: Parameter λ_{pt}, i.e., financial incentives

	Technologies				
Plants	1	2	3	4	5
1	18,381,994	23,877,173	25,861,176	26,144,822	29,402,665
2	18,413,353	19,951,340	21,323,076	23,347,034	27,092,882
3	18,554,056	19,427,972	**23,264,932**	25,755,756	26,917,590
4	19,165,581	22,578,701	22,706,724	23,980,369	26,512,378
5	25,865,735	27,056,240	27,186,202	27,881,494	29,516,928
6	20,054,240	21,312,301	22,084,629	**26,337,944**	27,542,399
7	20,242,471	21,805,194	25,023,213	26,156,432	26,472,553

Table 5.13: Parameter TEC_{pt}, i.e., installation cost of a technology in a plant

	Technologies				
Plants	1	2	3	4	5
1	79,433,098,839	81,811,528,849	93,567,857,686	100,369,460,689	102,222,139,072
2	71,788,136,663	75,969,812,389	79,587,262,773	84,374,908,653	102,380,881,509
3	74,555,673,184	78,793,720,784	**100,365,197,034**	103,950,837,705	106,921,341,999
4	82,142,479,916	92,506,884,848	98,810,469,033	101,789,335,526	106,087,901,670
5	92,936,274,682	93,674,299,648	104,168,162,165	115,871,217,103	117,258,705,656
6	77,636,060,022	80,446,746,121	83,453,079,334	**97,329,655,730**	103,097,237,595
7	87,056,230,352	91,286,442,093	105,464,085,136	108,275,000,831	108,514,275,740

Table 5.14: Parameter MC_{pt}, i.e., manufacturing cost using a technology in a plant

	Technologies				
Plants	1	2	3	4	5
1	16,635,123	17,286,695	18,907,621	19,150,934	23,092,039
2	15,641,870	16,807,377	22,486,189	23,161,401	24,210,973
3	16,201,870	17,553,867	**18,174,278**	22,673,295	22,946,579
4	15,205,357	20,250,452	20,773,942	21,712,022	23,145,398
5	18,258,336	19,400,356	22,152,125	22,890,735	24,236,757
6	17,576,137	20,464,494	21,420,608	**21,536,999**	23,522,639
7	18,988,807	19,190,483	20,056,366	22,519,464	24,326,136

Chapter 6

Conclusions and Further Research Directions

In this study a sustainable supply chain network design approached by using bi-level programming is presented for the first time. The proposed model considers the environmental impact caused by opening/establishing plants and distribution centers. Also, the emissions caused by manufacturing and transporting products are considered. Government tries to encourage supply chain's manager to use cleaner technologies by providing financial incentives. Moreover, two different scenarios were modeled. The differences between both scenarios rely in the assumption regarding the use of technologies that can be used in the plants regarding the manufacture of products.

As it is mentioned above, this problem is modeled as a bi-level mathematical program. Therefore, an environmental agency from the government plays the leader role and a supply chain's manager plays the follower role. Since the follower's problem consists in a mixed integer programming problem, KKT optimality conditions cannot be used to convert the bi-level model into an equivalent single-level model. Due to the latter, we developed a heuristic algorithm based on simulated annealing metaheuristic. Initial leader's solutions are constructed by solving a fake-knapsack subproblem adapted from the proposed model. Then, the follower's optimal response is obtained by using CPLEX optimizer. The local search phase follows the ideas from simulated annealing metaheuristic. As a result of this, an intensified and diverse search over the solution space is performed.

To evaluate the performance of the proposed algorithm, a set of 30 instances were considered. The first subset of 12 instances is taken from the literature in which a real case was resembled; the second subset of 18 instances was extended from the first subset. Results obtained from the computational experimentation indicated that utilizing financial incentives within a bi-level framework have a significant positive effect on the use of cleaner technologies in supply chain network design and to decrease the environmental impact. Hence, the proposed model could be considered as a new approach for sustainability in supply chain and has a notable effect on decreasing environmental effects.

As a future research direction, the model can be extended to incorporate other sustainable criteria, such as, local development, product risk, employment, and damage to workers by using analytic hierarchy process (AHP) techniques. Also, considering uncertainty in some parameters could be a very interesting feature to be analyzed. For example, the demand of the retailers and the amount of financial incentives required to establishing a technology in a plant.

Bibliography

[1] AFAR, S., BROTCORNE, L., MARCOTTE, P., AND SAVARD, G. Achieving an optimal trade-off between revenue and energy peak within a smart grid environment. *Renewable Energy 91* (2016), 293 – 301.

[2] ARMBRUST, M., FOX, A., GRIFFITH, R., D. JOSEPH, A., KATZ, R., KONWINSKI, A., LEE, G., PATTERSON, D., RABKIN, A., AND ZAHARIA, M. Above the clouds: A berkeley view of cloud computing.

[3] ATAR, R., CIDON, I., AND SHIFRIN, M. Mdp based optimal pricing for a cloud computing queueing model. *Performance Evaluation 78* (2014), 1 – 6.

[4] BHARDWAJ, O., ANSHELEVICH, E., AND KAR, K. Coalitionally stable pricing schemes for inter-domain forwarding. *Computer Networks 97* (2016), 128 – 146.

[5] BUI, D.-M., YOON, Y., HUH, E.-N., JUN, S., AND LEE, S. Energy efficiency for cloud computing system based on predictive optimization. *Journal of Parallel and Distributed Computing 102* (2017), 103 – 114.

[6] BUYYA, R., YEO, C. S., VENUGOPAL, S., BROBERG, J., AND BRANDIC, I. Cloud computing and emerging it platforms: Vision, hype, and reality for delivering computing as the 5th utility. *Future Generation Computer Systems 25*, 6 (2009), 599 – 616.

[7] COECKELBERGH, M. Technology and the good society: A polemical essay on social ontology, political principles, and responsibility for technology. *Technology in Society 52* (2018), 4 – 9.

[8] DING, S., XIA, C., CAI, Q., ZHOU, K., AND YANG, S. Qos-aware resource matching and recommendation for cloud computing systems. *Applied Mathematics and Computation 247* (2014), 941 – 950.

[9] DO, C. T., TRAN, N. H., HUH, E.-N., HONG, C. S., NIYATO, D., AND HAN, Z. Dynamics of service selection and provider pricing game in heterogeneous cloud market. *Journal of Network and Computer Applications 69* (2016), 152 – 165.

[10] GROSSMAN, R. L. The case for cloud computing. *IT Professional 11*, 2 (Mar. 2009), 23–27.

[11] GUROUT, T., GAOUA, Y., ARTIGUES, C., COSTA, G. D., LOPEZ, P., AND MONTEIL, T. Mixed integer linear programming for quality of service optimization in clouds. *Future Generation Computer Systems 71* (2017), 1 – 17.

[12] HUANG, J., KAUFFMAN, R. J., AND MA, D. Pricing strategy for cloud computing: A damaged services perspective. *Decision Support Systems 78* (2015), 80 – 92.

[13] LI, Z., ZHANG, H., OBRIEN, L., JIANG, S., ZHOU, Y., KIHL, M., AND RANJAN, R. Spot pricing in the cloud ecosystem: A comparative investigation. *Journal of Systems and Software 114* (2016), 1 – 19.

[14] MACAS, M., AND GUITART, J. Analysis of a trust model for sla negotiation and enforcement in cloud markets. *Future Generation Computer Systems 55* (2016), 460 – 472.

[15] MARSTON, S., LI, Z., BANDYOPADHYAY, S., ZHANG, J., AND GHALSASI, A. Cloud computing the business perspective. *Decision Support Systems 51*, 1 (2011), 176 – 189.

[16] MELL, P. M., AND GRANCE, T. Sp 800-145. the nist definition of cloud computing. Tech. rep., Gaithersburg, MD, United States, 2011.

[17] OJALA, A. Adjusting software revenue and pricing strategies in the era of cloud computing. *Journal of Systems and Software 122* (2016), 40 – 51.

[18] PAL, R., AND HUI, P. Economic models for cloud service markets: Pricing and capacity planning. *Theoretical Computer Science 496* (2013), 113 – 124. Distributed Computing and Networking (ICDCN 2012).

[19] RICO, D. F., SAYANI, H. H., AND FIELD, R. F. History of computers, electronic commerce and agile methods. In *Advances in COMPUTERS*, M. V. Zelkowitz, Ed., vol. 73 of *Advances in Computers*. Elsevier, 2008, pp. 1 – 55.

[20] RVEILLAC, J.-M. 1 - operational research. 1 – 9.

[21] SHARMA, B., THULASIRAM, R. K., THULASIRAMAN, P., GARG, S. K., AND BUYYA, R. Pricing cloud compute commodities: A novel financial economic model. In *Proceedings of the 2012 12th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing (Ccgrid 2012)* (Washington, DC, USA, 2012), CCGRID '12, IEEE Computer Society, pp. 451–457.