UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO
PROGRAMA DE MAESTRÍA Y DOCTORADO EN INGENIERÍA
Ingeniería de Sistemas – Investigación de Operaciones

A Metaheuristic based on Granular Tabu Search for the Distance Constrained Vehicle Routing Problem

TESIS

QUE PARA OPTAR POR EL GRADO DE:
MAESTRA EN INGENIERÍA

PRESENTA:
GABRIELA GARCÍA VILLEGAS

TUTORES PRINCIPALES:
MAYRA ELIZONDO CORTÉS
Facultad de Ingeniría, UNAM.

RODRIGO CARLOS EUGENIO LINFATI MEDINA
Facultad de Ingeniería, Universidad del Bío Bío, Chile.

UNIVERSIDAD DEL BÍO-BÍO

MÉXICO, D. F. MARZO 2013

JURADO ASIGNADO:

Presidenta: Angélica del Rocío Lozano Cuevas

Secretaria: María Elena Lárraga Ramírez

Vocal: Mayra Elizondo Cortés

1 er. Suplente: Daniele Vigo

2 d o. Suplente: Rodrigo Carlos Eugenio Linfati Medina

Lugares donde se realizó la tesis:

- Facultad de Ingeniería, Universidad Nacional Autónoma de México.
- Facultad de Ingeniería, Alma Mater Studiorum-Universidad de Bologna, Italia.

TUTOR DE TESIS:

Mayra Elizondo Cortés

FIRMA

Acknowledgements and Dedicatory

This thesis project is devoted to my parents, Federico García and Margarita Villegas, for their unconditional love, support and encouragement throughout my life. Thanks father for give me the rights words at the worst moment. Also, I would like to thank my brother and my sister, Luis Fernando and Beatriz because they are the best friends that the life gave me.

I want particularly mention to an important person, who gave all his love, encouragement, understanding, and patience even during hard times of this study but specially for being with me at all times, thanks Alberto Canseco.

I am extremely grateful to Daniele Vigo for gave me the opportunity to work with him in the Bologna University and provided me everything scholar that I required.

I would also like to express my gratitude to some people and organizations that, in their own manner, have made this decisive moment of my life become a reality:

- ✓ Rodrigo Linfati and Mayra Elizondo, my dissertation supervisors, for his excellent guidance and valuable support during the development of my individual project.

- ✓ The Mexican National Council for Science and Technology (CONACYT), for granting me a full scholarship, during two years, to pursue a master’s degree in Mexico and giving me the opportunity, through a mix scholarship for travel to University of Bologna with the purpose of learning efficient algorithms used in the commodities distribution.

- ✓ The Engineering Institute (IING) and Dra. Angelica Lozano Cuevas for support me in the extra time to finalize this thesis master.

- ✓ Department of Electronics, Computer Science and Systems (DEIS). University of Bologna, Italy.

Finally, I would like to thank my assessment jury, who gave me their accuracy revisions about this project.

- Angélica del Rocío Lozano Cuevas
- María Elena Lárraga Ramírez
- Mayra Elizondo Cortés
- Daniele Vigo
- Rodrigo Carlos Eugenio Linfati Medina

INDEX

ABSTRACT I
RESUMEN I
INTRODUCTION II
GENERAL OBJECTIVE IV
SCOPE AND LIMITATIONS OF RESEARCH V
CHAPTER 1 VEHICLE ROUTING PROBLEM 6
1.1 ASSOCIATED CONCEPTS 6
1.2 DISTANCE CONSTRAINED VEHICLE ROUTING PROBLEM 8
1.3 FORMULATION MODEL 10
1.4 EXACT ALGORITHM 11
1.4.1 Branch and Bound 11
1.4.2 Branch and Cut 12
1.4.3 Branch and Price...12
1.5 APPROXIMATE ALGORITHMS...12
1.5.1 Classical Heuristics for the Vehicle Routing Problem 13
1.5.2 Metaheuristics for the DCVRP...15
CHAPTER 2 METHODOLY 18
2.1 CLARK AND WRIGHT HEURISTIC 18
2.1.1 Parallel Saving Algorithm 20
2.2 ENHANCEMENTS OF THE PARALLEL CLARKE AND WRIGHT ALGORITHM 21
2.3 LOCAL SEARCH 22
2.4 TABU SEARCH...24
2.4.1 Cycle in Local Search...25
2.4.2 Tabu Search Elements 26
2.5 GRANULAR TABU SEARCH..28
2.5.1 Granular Tabu Search to Vehicle Routing Problem 29
2.5.2 Efficient Search and Diversification...31
CHAPTER 3 IMPLEMENTATION 32
3.1 GENERAL FRAMEWORK...32
3.1.1.1 Execute Mode 33
3.2 CONSTRUCTION OF INITIAL SOLUTION 34
3.2.1 Route Construction Approaches 35

3.3 IMPROVEMENT PHASE 38
3.3.1 2-*Optimal* Procedure 38
3.3.2 Relocate 40
3.4 IMPLEMENTATION OF GRANULAR TABU SEARCH 42
3.4.1. Other Strategies of Diversification 43
CHAPTER 4 ANALYSIS OF THE RESULTS 45
4.1 PROCEDURES 46
4.2 COMPUTATIONAL EXPERIMENTS 47
CONCLUSIONS AND FUTURE WORK 56
REFERENCES 59
APPENDIX A 62
APPENDIX B 63
APPENDIX C 64
APPENDIX D 65
APPENDIX E 66
APPENDIX F 67
APPENDIX G 68
APPENDIX H 69
APPENDIX I 70
APPENDIX J 71
APPENDIX K 72
APPENDIX L 73
APPENDIX M 74
APPENDIX N 75
APPENDIX Ñ 76
APPENDIX O 77
APPENDIX P 78
APPENDIX Q 79
APPENDIX R 80
APPENDIX S 81
APPENDIX T 82
APPENDIX U 83
APPENDIX V 84
APPENDIX W 85
APPENDIX X 86

APPENDIX Y 87
APPENDIX Z 88
APPENDIX A1 89
APPENDIX B1 90
APPENDIX C1 91
APPENDIX D1 92
APPENDIX E1 93
APPENDIX F1 94
APPENDIX G1 95

ABSTRACT

The objective in this thesis project was design and to implement an algorithm based on Granular Tabu Search (GTS) for the vehicle routing problem with capacity and route length restrictions. The algorithm integrated in two phases: initial solution and improvement solution. In the initial solution has been obtained by using the parameterized parallel saving method. The improvement phase includes GTS; it considers a sequence of inter-route adjacent solutions obtained by repeatedly removing a vertex from its current route and reinserting it into another route, called relocate like procedure to less the vehicles number and, 2-optimal technique use to earn another intra-route solution. Two different search strategies for selecting the next movement were implemented, the first admissible movement and best admissible move. Intensification and diversification of the search was achieved through frequency penalization. Computational results were reported for a set of the 34 instances, which are 14 classical and 20 large-scale instances with between 51 and 484 customers. In tree instances was obtained the optimal solution and significantly reduced the runtime in all instances.

Keywords: Distance Constrained Vehicle Routing Problem, Granular Tabu Search, Metaheuristic, Saving Algorithm.

RESUMEN

En esta tesis se desarrolló e implemento un algoritmo para el Problema de Ruteo de Vehículos con Restricciones de Distancia, basado en Búsqueda Tabú Granular (BTG). El algoritmo se compone de dos fases. En la primera se construye una solución inicial basada en el método de los ahorros parametrizado; mientras que en la segunda fase, de mejora, se implementa la búsqueda tabú granular; así como la generación de dos vecindarios: "relocate "para movimientos entre rutas y 2-*optimal* para movimientos dentro de la ruta. Dos estrategias de búsqueda son usadas para seleccionar un movimiento admisible y el mejor. La búsqueda se intensifica y diversifica mediante una frecuencia de penalización. Los resultados computacionales son probados en un conjunto de 34 instancias: 14 instancias clásicas y 20 de larga escala, las cuales contienen entre 51 y 484 clientes. Se obtuvo la solución óptima para tres instancias y se redujo significativamente el tiempo de ejecución en todas las instancias.

Palabras clave: Problema de Ruteo Vehicular con Restricciones de Distancia, Búsqueda Tabú Granular, Metaheurísticas, Algoritmo de Ahorros.

INTRODUCTION

The vehicle routing problem (VRP) was proposed by Dantzig and Ramser (1959); it is considerate like a combinatorial optimization and integer programming problem seeking to service a number of customers with a fleet of vehicles. The VRP forms the core of logistics planning and has been extensively studied by the operations research community. Often the context is that of delivering goods located at a central depot to customers who have placed orders for such goods. Implicit is the goal of minimizing the cost of distributing the goods. The last five decades have seen enormous improvements in the research community's ability to solve these problems, due to better algorithms as well as better computational capabilities. Toth and Vigo (2002) provide an up to date survey of problem variants, exact solution techniques, and heuristics for the vehicle routing as well important advances and new challenges techniques for modeling and solving the standard VRP and its many variants has advanced significantly (Laporte, 2009).

In this thesis project, is considering a variant of VRP called Distance Constrained Vehicle Routing Problem (DCVRP) concerning to the commodities distribution between depots and end users (customers), in which a set of vertices (customers), one depot D, homogeneous fleet K and a maximum route length L. In particular, the solution of a VRP calls for the determination of a set of routes, each performed by a single vehicle that starts and ends at its same depot, to the DCVRP has to satisfy the vehicle capacity, and maximum route length, so that all the requirements of the customers are fulfilled, all the operational constraints satisfied, and the overall transportation cost is minimized (Toth and Vigo, 2002).

For this problem has provided several techniques; therefore, an interesting question is whether the DCVRP can be solved more efficiently using new methods. The DCVRP was shown like *NP–hard* (*non-deterministic polynomial-time hard*). Hence, DCVRP be must look to means such as integer programming. Previous work on exacts methods to the VRP and its variant DCVRP; however these resolved relatively small instances, so that to implement approximate algorithms has been a hard work, because these should efficient models, which have the ability to solve large-scale problems. For the past two decades, the attention has been on approximate method called Heuristics and Metaheuristics, which are methods used to obtain imports solutions quickly(Cordeau et al., 2005).

The last ten years previous works on exact solutions have been proposed to VRP, and its variants, for the best exact algorithms for the Capacitated Vehicle Routing Problem (CVRP) have been based on either branch-and-cut or Lagrange relaxation/column generation. The resulting branch-and-cut algorithm can solve to optimality all instances from the literature with up to 135 vertices (Fukasawa et al. 2003). A branch-and-cut algorithm has described by (R. Baldacci, Hadjiconstantinou, and Mingozzi 2004), which base in two commodity network flow formulation of the CVRP. They used a variety of valid inequalities, including capacity, framed capacity, partial multistar, hypotour and classical Gomory mixed integer cuts. However, recent survey of the CVRP shows that the most promising exact algorithms for the symmetric CVRP version are based on covering both exact and heuristic algorithms. An example is the proposed algorithm by Franceschi, Fischetti and Toth (2006), which proposed a new ILP-based refinement heuristic for Vehicle Routing Problems. They considerate the DCVRP, where *k* minimum-cost routes through a central depot have to be constructed so as to cover all customers while satisfying, for each route, both a capacity and a total-distance-travelled limit. Their algorithm involves a procedure to generate a large number of new sequences through the extracted nodes, as well as a more sophisticated Integer Linear Programming (ILP) model for the reallocation of some of these sequences. An important feature of their method is that it does not rely on any specialized ILP code, as any general-purpose ILP solver can be used to solve the reallocation model. They reported computational results on a large set of CVRP instances from the literature (with symmetric/asymmetric costs and with/without distance constraints), along with an analysis of the performance of the new method and its features. Interestingly, in 13 cases the new method was able to improve the best-know solution available from the literature (Franceschi, Fischetti, and Toth 2006).

Many heuristics have been put forward. Some are purely constructive but most also include an improvement phase, these are called heuristics "classical" because they do not contain mechanisms allowing the objective function to deteriorate from one iteration to the next , these are : Saving Algorithm, Set Partitioning Heuristics and Cluster-First, Route-Second Heuristics (Laporte 2009), which will be described in the next section *1.5*. So, there are Improvement Heuristics, which can be employed to post-optimize a VRP solution. The first is called *Intra-route* moves consist of improving each route separately by means of a TSP algorithm; whereas, *inter-route*, the second, moves act on several routes simultaneously. It is common to alternate between these two schemes within the same improvement heuristic.

The Metaheuristics can be broadly classified into local search, population search, and learning mechanisms. Most Metaheuristics can be regarded as improvement methods. The best ones are rather robust and perform extremely well even if they are initiated from a low-quality solution. The number of variants of VRP Metaheuristics published in recent years; the most important for the VRP and it is variant DCVRP are: tabu search, Toth and Vigo (2003) proposed a granular tabu search, will be described in the chapter 2.So, Gendreau, Hertz and Laporte (1994) proposed a new tabu search heuristic called *TABUROUTE* for the vehicle routing problem with capacity and route length restrictions. The algorithm considers a sequence of adjacent solutions obtained by repeatedly removing a vertex from its current route and reinserting it into another route. This is done by means of a generalized insertion procedure previously developed by the authors. During the course of the algorithm, infeasible solutions are allowed (Gendreau, Hertz, and Gilbert Laporte 1994b). A variable neighborhood search (Kytöjoki et al. 2007) and adaptive large neighborhood search (Ropke and Pisinger 2006). Finally, Nagata and Bräysy (20010) proposed a Memetic algorithm upon an existing edge assembly crossover (Nagata, Bräysy, and Dullaert 2010).

As we see it, many techniques have been proposed to solve the VRP and its variant DCVRP, however in thesis project is based on tabu search (TS), which is one of the most widely used and effective heuristic approaches available for the solution of optimization problems (Toth and Vigo 2003), for this reason in this thesis project is used TS and with the suggestion of Toth and Vigo (2003) called Granular Tabu Search (GTS), which is based on the use of drastically restricted neighborhoods, not containing the moves that involve only elements that are not likely to belong to good feasible solutions. These restricted neighborhoods are called granular, and may be seen as an efficient implementation of candidate-list strategies proposed for tabu-search algorithms(Gendreau 2003).

At the difference of Toth and Vigo (2003), this thesis project has two phases: Initial solution and Improvement solution. The initial solution generated with Enhancements of the Parallel Clarke and Wright Algorithm and *TABUROUTE*, for the capacity and route length restrictions. The Improvement solution uses GTS like a local search method, what use a 2-*optimal* and *relocate* mechanisms to generate move generation and consequently create the neighboring solutions by changing one attribute or a combination of attributes of the initial solution given. The attribute is concerned to arcs connecting a pair of customers. Once a neighboring solution is identified, it is compared against the current solution. If the neighboring solution is better, it replaces the current solution, and the search continues. The results obtained are evaluated in 14 classical instances and 20 size large instances (Toth P., A. Tramontani, 2008). Subsequently will be use Local Search through GTS introduced by Toth and Vigo and has yielded excellent results on the VRP (Toth and Vigo, 2003).

GENERAL OBJECTIVE

Design, development and implement a Metaheuristic based on Granular Tabu Search to solve the Distance Constrained Vehicle Routing Problem.

The specific objectives were:

- Make an examination of state of the art of the distance constrained vehicle routing problem from the methods so commonly used to solve it.
- Search benchmarking instances and the best know solution according to state of the art.
- Evaluate strategies using different parameters to the saving algorithm to produce a good initial solution.
- Implement GTS and insert move to solve DCVRP, with the issue of proposed ideas that are necessaries to explain the algorithm.
- Make a comparison of solutions according to reports.
- Propose future writings about DCVRP.

To meet the proposed objectives, the thesis project was organized as follows. In Chapter 1 is presented the classical Vehicle Routing Problem and it is derived DCVRP, including some exact and approximate resolution methods and mathematical formulation. Chapter 2 describes the initial solution, obtained to implement parameterized saving parallel algorithm. Then, Chapter 3 introduces the ideas about Tabu Search heuristic, Granular Tabu Search, Relocate and 2-Optimal, which were implemented in this thesis project. Chapter 4 presents the Metaheuristic proposed, including through first and second phase development, where was forming the Construction and Route Improvement Metaheuristic. Here, results in terms of solution quality as well as computation time are presented and discussed. Chapter 5 shows the computational experiments and the obtained results. Finally, Chapter 6 outlines the conclusions and further research in this field. Appendices, bibliographies are included at the end.

SCOPE AND LIMITATIONS OF RESEARCH

The scope of this thesis project is limited to instances proposed from literature that are generally used as a standard benchmark for the VRP and its variant DCVRP. The 14 standard test problems are proposed by Christofides and Eilon (Christofides and Eilon 1969), which include from 201 to 484 customers (Christofides, Mingozzi, and Toth 1979) and 20 large scale instances (Feiyue Li, Bruce Golden, and Edward Wasil 2005) from 201 to 484 customers proposed, which used :

- Distance matrix calculated by Euclidian Distance.
- Deterministic Local Search

The model can be applied, with minor modifications, for solving other pickup and delivery problems, even as research project to PhD will be applied to the Single Vehicle Routing Problem with Deliveries and Selective Pickups.

CHAPTER 1 VEHICLE ROUTING PROBLEM

In this section, are describe the typical characteristics of the vehicle routing problems by considering their main components (road network, customers, depots, vehicles, and drivers), the different functional constraints that can be imposed on the construction of the routes, and the possible objectives to be achieved in the optimization process pertinent like introducing to DCVRP, after their mathematical formulation and solution methods are given.

1.1 ASSOCIATED CONCEPTS

The distribution of commodities concerns the service, in a given time period, of a set of customers by a set of vehicles, which are located in one or more depots, are operated by a set of crews (drivers), and perform their movements by using an suitable road network. In particular, the solution of a VRP calls for the determination of a set of routes, each performed by a single vehicle that starts and ends at its own depot, such that all the requirements of the customers are fulfilled, all the operational constraints are satisfied, and the overall transportation cost is minimized (Toth and Vigo, 2002)

The road network, used for the transportation of goods, is usually described through a graph, whose arcs represent the road sections, and whose vertices correspond to the road junctions and to the depot and customer locations. The arcs (and hence the corresponding graphs) can be directed or undirected, depending on whether they can be traversed in only one or in both directions, respectively. Each arc is associated with a cost, which usually represents its length, and a travel time, which is possibly based on the vehicle type or on the period during which the arc is traversed(Bramel and Simchi-Levi ,1997).

The vehicle routing problem is composed by:

(i) ***Customers:*** each customer has a demand, which should be satisfied completely. Typical characteristics of customers are:
- Vertex of the road graph in which the customer is located;
- Amount of commodities (demand), possibly of different types, which must be delivered or collected at the customer;
- Times of the day (time windows) during which the customer can be served (for instance, because of specific periods during which the customer is available or the location can be reached, due to traffic limitations);
- Times required to deliver or collect the goods at the customer location (unloading or loading times, respectively), possibly dependent on the vehicle type; and subset of the available vehicles that can be used to serve the customer (for instance, because of possible access limitations or loading and unloading requirements).

(ii) ***Depot:*** is the home depot of the vehicles and commodities, which can be one or more that one. The route should start and finish in the same depot.

(iii) ***Vehicles:*** Transportation of goods is performed by using a fleet of vehicles whose composition and size can be fixed or can be defined according to the requirements of the customers. Typical characteristics of the vehicles are:

- Capacity of the vehicle, expressed as the maximum weight, or volume, or number of pallets, the vehicle can load;
- Possible subdivision of the vehicle into compartments, each characterized by its capacity and by the types of goods that can be carried;
- Devices available for the loading and unloading operations;
- Subset of arcs of the road graph which can be traversed by the vehicle; and
- Costs associated with utilization of the vehicle (per distance unit, per time unit, per route, etc.).

- These are transport modality whereby goods are distributed to the customers, can be a vehicles set with same capacity (homogeneous fleet) or different capacity (heterogeneous fleet).

The most common side constraints include Capacity restrictions: a non-negative weight (or demand) d_i is attached to each city $i > 1$ and the sum of weights of any vehicle route may not exceed the vehicle capacity. Capacity constrained VRP will be referred to as Capacitated Vehicle Routing Problem (CVRP);

- The number of cities on any route is bounded above by q (this is a special case of (i) with $d_i = 1$ for all $i >$ land $D = q$);
- Total time restrictions: the length of any route may not exceed a prescribed bound L; this length is made up of intercity travel times c_{ij} and of stopping times δ_i at each city i on the route. Time or distance constrained VRP will be referred to as DVRP;
- Time windows: city i must be visited within the time interval $[a_i , b_i]$ and waiting is allowed at city i;
- Precedence relations between pairs of cities: city i may have to be visited before city j.

Objectives considered (Toth and Vigo, 2002)in VRP are:

1. Minimization of the global transportation cost, dependent on the global distance traveled (or on the global travel time) and on the fixed costs associated with the used vehicles (and with the corresponding drivers).

2. Minimization of the number of vehicles (or drivers) required to serve all the customers; balancing of the routes, for travel time and vehicle load.

The next picture (Figure 1.1) summarizes the VRP and their variants with. Describe the Capacitated Vehicle Routing Problem CVRP, which is the simplest and most studied member of the family, Distance-Constrained VRP, the VRP with Time Windows, the VRP with Backhauls, and the VRP with Pickup and Delivery. For each of these problems, several minor variants have been proposed and examined in the literature, and often different problems are given the same name.

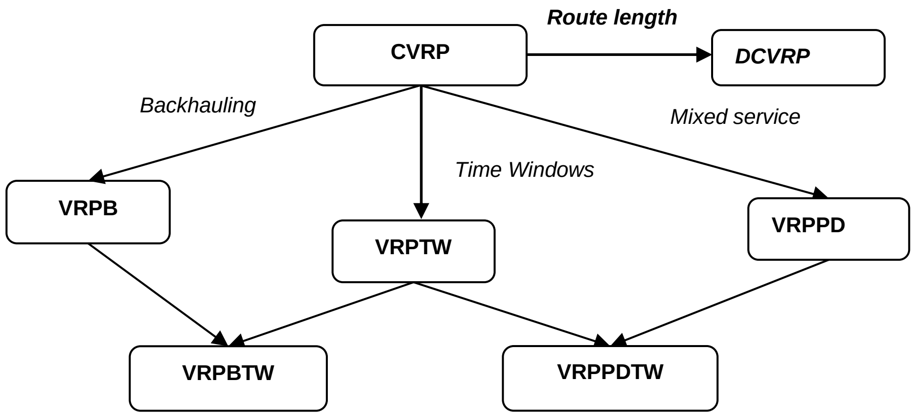

Figure 1.1 The basic problems of the VRP class and their interconnections.

The most basic Vehicle Routing Problem (VRP) with one depot is called Capacitated Vehicle Routing Problem (CVRP), which can be described as follows. A set of customers has to be served by a fleet of identical vehicles of limited capacity. The vehicles are initially located at a given depot. The objective is to find a set of routes for vehicles with minimal total length. Each route begins at the depot, visits a subset of the customers and returns to the depot without violating the capacity constraint. This thesis project considers a variation of CVRP called Distance Constrains Vehicle Routing Problem (DCVRP), as a variant of VRP, which will be described in the next section(Toth and Vigo ,2002).

1.2 DISTANCE CONSTRAINED VEHICLE ROUTING PROBLEM

The Distance Constrained vehicle Routing Problem has several minor variant of VRP, starting of CVRP may be described as the following graph theoretic problem. Let $G = (V, A)$ where $V = \{0,\ldots, n\}$ is the vertex set and A is the arc set. Vertices $i = 1,..., n$ correspond to the customers, whereas vertex 0 corresponds to the depot. Sometimes the depot is associated with vertex $n + 1$(Gilbert Laporte, Yves Nobert, and Martin Desrochers 1985).

A nonnegative cost, c_{ij}, is associated with each arc $(i, j) \in A$ and represents the travel cost spent to go from vertex i to vertex j. If G is a directed graph, the cost matrix c is symmetric, this thesis project considerate a symmetric matrix, the corresponding problem is called Asymmetric Capacitated Vehicle Routing Problem (ACVRP). Otherwise, costs matrix have a cost $c_{ij} = c_{ji}$, for all $(i, j) \in A$. In several practical cases, the cost matrix satisfies the triangle inequality $c_{ik} + c_{kj} \geq c_{ji}$ for all $i, j, k \in A$ For some instances the vertices are associated with points of the plane having given coordinates, and the cost c_{ij}, for each arc $(i, j) \in A$, is defined as the Euclidean distance between the two points corresponding to vertices i and j. In this case the cost matrix is symmetric and satisfies the triangle inequality, and the resulting problem called Euclidean SCVRP.

Each customer i (i = 1, …,n) is associated with a known nonnegative demand, d_i, to be delivered, and the depot has a fictitious demand $d_0 = 0$. Given a vertex set $S \subseteq V$, let $d(S) = \sum_{i \in S} d_i$ denote the total demand of the set.

A set of K identical vehicles, each with capacity Q, is available at the depot. To ensure feasibility we assume that $d_i <$ C for each $i = 1,\dots, n$. In the case of DCVRP $\sum_{route} d_i \leq Q$; where d_i represents d demand each customer i, so the *route* should satisfy the capacity requirement Q, $i = 1 \dots n$ .So, each vehicle based at the depot capacity D, $\{D > max(d_i)\}$.

The CVRP, as antecedent to DCVRP (See Figure 1.2), consists of finding a collection of exactly K simple circuits (each corresponding to a vehicle route) with minimum cost, defined as the sum of the costs of the arcs belonging to the circuits, and such that:

i. Each circuit visits the depot vertex;
ii. Each customer vertex is visited by exactly one circuit;
iii. The sum of the demands of the vertices visited by a circuit does not exceed the vehicle capacity, Q.

The first variant of CVRP is called Distance Constrained Vehicle Routing Problem, where for each route the capacity constraint is replaced by a maximum length constraint. In particular, a nonnegative length, is associated with each arc $(i, j) \in A$, and the total length of the arcs of each route cannot exceed the maximum route length, L.

Formally, DCVRP can be defined as follow:

Let a set of vertices (customers),one depot D, homogeneous fleet K and a maximum route length L, find a minimum cardinality set of tours originating at the depot covers all vertices, such that each tour (route) has length at most L (Ralphs et al. 2003) subject to :

- Each city, except the depot, must be visited exactly once and by a single vehicle.
- Each vehicle starts and ends its journey at the depot. The sum of demands contained on a vehicle's route may not exceed Q and the total length of the route may not exceed a pre-specified upper bound L.

- The objective is to minimize the total distance or cost traveled while satisfying all constraints.

The Figure 1.2 shows to L as maximal route length constraint (duration), which will be satisfied, as soon demand associated each customer determinate by d_i , it have to less or equal to vehicle's route Q.

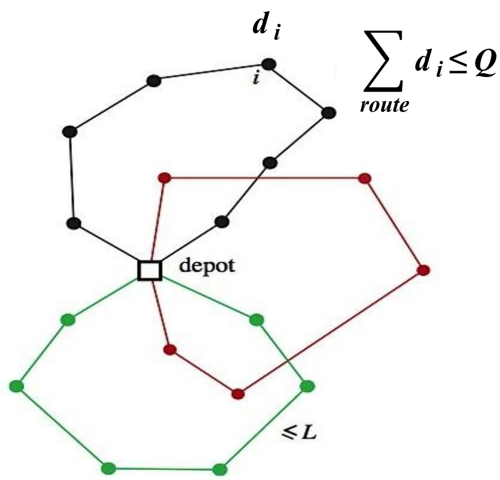

Figure 1.2 Visual representations of distance and demand constrains in DCVRP.

In following section is presented the formulation model of the DCVRP base on VRP as Integer Programming Model. In this formulation is added Distance (cost) constrain as soon all constraints above mentioned.

1.3 FORMULATION MODEL

The mathematical formulation to VRP and its variant DCVRP, use integer variables associated with each arc or edge of the graph, which count the number of times the arc is traversed by a vehicle. This mathematical formulation considers cases in which the cost of the solution can be expressed as the sum of the costs associated with the arcs, and when the most relevant constraints concern the direct transition between the customers within the route(Chung-Lun Li, Simchi-Levi, and Martin Desrochers 1992).

The Mathematical Formulation to symmetric version more general DCVRP was proposed Laporte, Norbert, and Desrochers, which assumed as usual, that the cost and the length matrices coincide, $t_{ij} = c_{ij}$, for each $i, j \in A$, $i < j$ and a maximum length constraint. In particular, a nonnegative length, t_{ij} associated with each arc i, j $\in A$ and the total length of the arcs of each route cannot exceed the maximum route length, L. In this, thesis project was considerate when arc lengths represent a service time s_i , it is associated with each customer i, denoting the time period for which the vehicle must stop at its location.

The formulation model is:

$$Min\ z(x) = \sum_{i \in V \setminus \{n\}} \sum_{j>i} c_{ij} x_{ij} \tag{1.1}$$

$$\sum_{h<i} x_{hi} + \sum_{j>i} x_{ij} = 2 \forall i \in V \setminus \{0\}, \tag{1.2}$$

$$\sum_{j \in V \setminus \{0\}} x_{0j} = 2K, \tag{1.3}$$

$$\sum_{i \in S} \sum_{\substack{j>i \\ j \in S}} x_{ij} \le |s| - r'(S) \forall S \subseteq V \setminus \{0\}, S \ne \varnothing \tag{1.4}$$

$$x_{ij} \in \{0,1\} \forall i, j \in V \setminus \{0\}, i < j, \tag{1.5}$$

$$x_{0j} \in \{0,1,2\} \forall j \in V \setminus \{0\} \tag{1.6}$$

Notation:

The constraint (1.1) minimize the objective function, the degree constraints (1.2) and (1.3) impose that exactly two arcs are incident into each vertex associated with a customer and that 2*K* arcs are incident into the depot vertex, respectively. The restriction (1.4) imposes the connectivity of the solution, the vehicle capacity, and the maximum route length requirements, by forcing that a sufficient number of arcs leave each subset of vertices. Given a subset *S* of customer vertices, the quantity *r*'(*S*) represents the minimum number of vehicles needed to serve all customers in *S*. This quantity is given by the maximum between *r*(*S*), which takes into account the capacity constraints, and the smallest value *v* satisfying:

$$v = \lceil H_v(S)/L \rceil, \ v = r(S), \dots, \min\{K, |S|\}, \tag{1.7}$$

Where is the optimal cost of a multiple TSP visiting all customers in *S* and using exactly *v* tours passing through the depot. Since the multiple TSP is an NP-hard problem, an approximation from below of the above value may be obtained by using any lower bound on the value of $H_v(S)$ (G. Laporte, M. Desrochers, and Y. Nobert 1984).

1.4 EXACT ALGORITHM

In this section are presented the most common exact algorithms used to solve the DCVRP(Roberto Baldacci et al., 2010).

1.4.1 Branch and Bound

Branch & Bound (B&B) is a general algorithm for finding optimal solutions of various optimization problems, especially in discrete and combinatorial optimization. It consists of a systematic enumeration of all candidate solutions, where large subsets of fruitless candidates are discarded in masse, by using upper and lower estimated bounds of the quantity being optimized(Barnhart et al. 1998).

It is a general algorithm for founding optimal solutions of various optimization problems, especially in discrete and combinatorial optimization. It consists of a systematic enumeration of all candidate solutions, where large subsets of fruitless candidates are discarded en masse, by using upper and lower estimated bounds of the quantity being optimized.

1.4.2 Branch and Cut

The method is a hybrid of branch & bound plus cutting plane methods. The method solves the linear program without the integer constraint using the regular simplex algorithm. When an optimal solution is obtained, and this solution has a non-integer value for a variable that is supposed to be an integer, a cutting plane algorithm is used to find further linear constraints which are satisfied by all feasible integer points but violated by the current fractional solution. If such an inequality is found, it is added to the linear program, such that resolving it will yield a different solution which is hopefully "less fractional". This process is repeated until either an integer solution is found (which is then known to be optimal) or until no more cutting planes are found (Savelsbergh, 1997).

1.4.3 Branch and Price

Branch & Price, is a method that the procedure is based on Column Generation rather than Row Generation, which sets of columns are left out of the LP relaxation of large Integer Programming because there are too many columns to handle efficiently and most of them will have their associated variables equal to zero in an optimal solution anyway. Then to check optimality, a sub problem, also called the “pricing problem” is solved to identify columns to enter the basis. If such columns are found, the LP is optimized. Branching occurs when no columns “price” out to enter the basis and the LP solution does not satisfy integrality conditions (Barnhart et al., 1998)

1.5 APPROXIMATE ALGORITHMS

Because of the VRP is so hard to solve exactly and algorithmic behavior is highly unpredictable, a great deal of effort has been invested on the design of heuristics. The classical heuristics usually consisting in construction phase followed by a relatively simple post optimization phase, and Metaheuristics based on new optimization concepts developed over the past fifteen to twenty years. Since the performance of heuristics can only be assessed experimentally it is common to make comparisons on a set of four- teen benchmark instances proposed by Christofides, Mingozzi and Toth (1979) (CMT) which range from 50 to 199 cities. The best known solution values for these instances have been obtained by Taillard (1993) and Rochat and Taillard (1995). We first describe some of the most representative classical heuristics. An extensive survey is provided by Laporte and Semet (2002).

Methods or approximation algorithms also can be derived directly from optimization algorithms, by heuristically solving different phases of the process. The approximation algorithm much of the emphasis was put on quickly obtaining a feasible solution and possibly applying to it a post optimization procedure. These methods considered the following:

- Flexibility to use good starting solutions (which the practitioner can usually provide, based on his/her knowledge of the problem, or based on a known solution of some similar problem).
- The ability to perform sensitivity analysis (resolve the problem with slightly different problem data) quickly.
- To find algorithms with probably good run times and with possibly good (optimal) solution quality.

1.5.1 Classical Heuristics for the Vehicle Routing Problem

Several families of heuristics have been proposed for the VRP. These can be broadly classified into two main classes: classical heuristics, developed mostly between 1960 and 1990, and Metaheuristics, whose growth has occurred in the last decade. Most standard construction and improvement procedures in use today belong to the first class(Bramel and Simchi-Levi 1995).

These methods perform a relatively limited exploration of the search space and typically produce good quality solutions within modest computing times. Moreover, most of them can be easily extended to account for the diversity of constraints encountered in real life contexts. Therefore, they are still widely used in commercial packages. In Metaheuristics, the emphasis is on performing a deep exploration of the most promising regions of the solution space. These methods typically combine sophisticated neighborhood search rules, data structures, and recombination of solutions (Bruce L. Golden et al. 1998)

The quality of solutions produced by these methods is much higher than that obtained by classical heuristics, but the price to pay is increased computing time. Moreover, the procedures usually are context dependent and require finely tuned parameters, which may make their extension to other difficult situations. In a sense, Metaheuristics are no more than sophisticated improvement procedures, and they can simply be viewed as natural enhancements of classical heuristics and they can be broadly classified into three categories (Toth and Vigo, 2002):

- Constructive heuristics
- Two-phase heuristics
- Improvement methods

Most of the heuristics developed for the VRP apply directly to capacity constrained problems (CVRP) and normally can be extended to the case where an upper bound is also imposed on the length of any vehicle route (DCVRP), even if this is not always explicitly mentioned in the algorithm description. Most heuristics work with an unspecified number K of vehicles, but there are some exceptions to this rule. This is clarified for each case.

- ***Constructive Methods***

Gradually this method builds a feasible solution while keeping an eye on solution cost, but they do not contain an improvement phase per se. Two main techniques are used for constructing DCVRP solutions: merging existing routes using a savings criterion, and gradually assigning vertices to vehicle routes using an insertion cost.

- ***Clarke and Wright Savings Algorithm***

This algorithm is one of the earliest and most widely used heuristics due to its speed, simplicity, and ease of adjustment to handle various constraints in real-life applications. It is based on the feasible merging of sub tours using a savings criterion, which refers to the cost saving achieved by combining two routes and using one vehicle rather than two. The Clarke and Wright algorithm can also be time consuming since all savings must be computed, stored, and sorted. Various enhancements have been proposed by a number of authors to speed up computations and to reduce memory requirements(Clarke and Wright ,1964).

- ***Sequential Insertion Heuristic***

The algorithms are based on sequential to problems with an unspecified number of vehicles. The first expands one route at a time. The second applies in turn sequential and parallel route construction procedures. Both methods contain a 3-opt improvement phase.

- ***Two-Phase Method***

The problem is decomposed into its two natural components, clustering of vertices into feasible routes and route construction, with possible feedback between the two stages. Two-phase heuristics are divided into two classes: cluster-first, route-second methods and route-first, cluster-second methods. In the first case, vertices are first organized into feasible clusters, and a vehicle route is constructed for each of them. In the second case, a tour is first built on all vertices and is then segmented into feasible vehicle routes. For example the sweep algorithm applies to planar instances of the VRP. Feasible clusters are initially formed by rotating a ray centered at the depot. A vehicle route is then obtained for each cluster by solving a TSP. Some implementations include a post optimization phase in which vertices are exchanged between adjacent clusters, and routes are optimized(Toth and Vigo, 2002).

- ***Route-First, Cluster-Second Methods***

This method attempts to upgrade any feasible solution by performing a sequence of edge or vertex exchanges within or between vehicle routes. Route-first, cluster-second methods construct in a first phase a giant TSP tour, disregarding side constraints, and decompose this tour into feasible vehicle routes in a second phase. This idea applies to problems with a free number of vehicles.

1.5.2 Metaheuristics for the DCVRP

In recent years several Metaheuristics have been proposed for the VRP. These are general solution procedures that explore the solution space to identify good solutions and often embed some of the standard route construction and improvement heuristics described previously. In a major departure from classical approaches, Metaheuristics allow deteriorating and even infeasible intermediary solutions in the course of the search process. The best-known Metaheuristics developed for the VRP typically identify better local optima than earlier heuristics, but they also tend to be more time consuming(Vega, Batista, and Pérez 2003).

Different types of Metaheuristic that have been applied to solve the VRP: Simulated Annealing (SA), Deterministic Annealing (DA), Tabu Search (TS), Genetic Algorithms (GA), Ant Systems (AS), and Neural Networks (NN). The first three algorithms start from an initial solution x_1 and move at each iteration t from to x_t a solution x_{t+1} in the neighborhood $N(x_t)$ of x_t until a stopping condition is satisfied. If $f(x)$ denotes the cost of x, then $f(x_{t+1})$ is not necessarily less than $f(x_t)$. As a result, care must be taken to avoid cycling. GA examines at each step a population of solutions(Bruce L. Golden et al. 1998). Each population is derived from the preceding one by combining its best elements and discarding the worst. In the following sections are described the common Metaheuristics useful in this thesis project.

Simulated Annealing

Simulated annealing was first proposed by Kirkpatrick, Gelatt, and Vecchi (1983). It is a randomized local search procedure where a modification to the current solution leading to an increase in solution cost can be accepted with some probability. This algorithm is motivated from an analogy with the physical annealing process used to find low-energy states of solids. In condensed matter physics, annealing denotes a process in which a solid is first melted by increasing its temperature; this is followed by a progressive temperature reduction aimed at recovering a solid state of lower energy. If the cooling is done too fast, widespread irregularities emerge in the structure of the solid, thus leading to relatively high energy states. Conversely, a careful annealing through a series of levels, where the temperature is held long enough at each level to reach equilibrium, leads to more regular structures associated with low-energy states. Basically, the process is less likely to get trapped in a high-energy state when the temperature is prevented from getting too far from the current energy level.
Their strategic belongs to a class of the local search algorithms that are known as threshold algorithms. These algorithms play a special role with local search for two reasons. First, they appear to be quite successful when applied to a broad range of practical problems. Second, some threshold algorithms such a simulated annealing have a stochastic component, which facilitates a theoretical analysis of their asymptotic convergence (Gendreau and Potvin, 2005).

Tabu Search

Tabu Search is basically a deterministic local search strategy where at each iteration, the best solution in the neighborhood of the current solution is selected as the new current solution, even if it leads to an increase in solution cost(Glover 1989). As opposed to a pure local descent, the method will thus escape from a local optimum. A short-term memory, known as the tabu list, stores recently visited solutions (or attributes of recently visited solutions) to avoid short-term cycling. Typically, the search stops after a fixed number of iterations or a maximum number of consecutive iterations without any improvement to the incumbent (best known) solution. The principle of the method originates from the work of Glover (1986).

Two Early Tabu Search Algorithms

In this algorithm, the solution is first transformed into a giant tour by replication of the depot, and neighborhoods are defined as all feasible solutions that can be reached from the current solution by means of 2-opt or 3-opt exchanges. The next solution is determined by the best non tabu move. The best non tabu feasible move is selected at each iteration. While better than Willard's algorithm, this implementation did not produce especially good results. Further research has shown that more sophisticated search mechanisms are required to make tabu search work (Toth and Vigo, 2002)

Taburoute

The Taburoute algorithm of Gendreau (1994) is rather involved and contains several innovative features. The neighborhood structure is defined by all solutions that can be reached from the current solution by removing a vertex from its current route, and inserting it into another route containing one of its p nearest neighbours using GENI, a Generalized Insertion procedure developed for the TSP. This may result in eliminating an existing route or in creating a new one. A second important feature of Taburoute is that the search process examines solutions that may be infeasible with respect to the capacity or maximum route length constraints. More precisely, the objective function contains two penalty terms, one measuring overcapacity, the other measuring over duration, each weighted by a self-adjusting parameter: every 10 iterations, each parameter is divided by 2 if all 10 previous solutions were feasible, or multiplied by 2 if they were all infeasible. This way of proceeding produces a mix of feasible and infeasible solutions and lessens the likelihood of being trapped in a local minimum (Laporte, 2000).

Taillard's algorithm

The Taillard did an implementation, which contains some of the features of Taburoute, namely random tabu durations and diversification. It defines neighborhood using the λ-interchange generation mechanism. Rather than executing the insertions with GENI, the algorithm uses standard insertions, thus enabling each insertion to be carried out in less time, and feasibility is always maintained.

Xu and Kelly's algorithm

This algorithm uses a more sophisticated neighborhood structure. They consider swaps of vertices between two routes, a global repositioning of some vertices into other routes, and local route improvements. The global repositioning strategy solves a network flow model to optimally relocate given numbers of vertices into different routes. Approximations are developed to compute the ejection and insertion costs, taking vehicle capacity into account. Route optimizations are performed by means of 3-opt exchanges and a Tabu Search improvement routine. The algorithm is governed by several parameters, which are dynamically adjusted through the search. A pool of best solutions is memorized and periodically used to reinitiate the search with new parameter values. Overall, this algorithm has produced several best-known solutions on benchmark instances, but it is fair to say that it is not as effective as some other Tabu Search implementations (Gilbert Laporte et al., 2000).

The following section presents fundamental theoretical principles, which provides support for this thesis project.

CHAPTER 2 METHODOLY

This chapter deals with basic theory needed in the remainder of the thesis project. The Clarke and Write heuristic, Tabu Search Metaheuristic, Granular Tabu Search, 2-optimal and relocate techniques are mentioned. This chapter describes the basic theory of Tabu Search (TS) and Granular Tabu Search (GTS), local search algorithm both. So, is presented the concepts about 2-optimal and relocate procedure.

2.1 CLARK AND WRIGHT HEURISTIC

The Clarke and Wright algorithm also called Saving Algorithm is one of the first originally developed heuristics for CVRP and it is frequently used, since this algorithm has been one of the earliest and most widely used heuristics due to its speed, simplicity, and ease of adjustment to handle various constraints in real life applications. It is based on the feasible merging of sub-tours using a savings criterion, which refers to the cost saving achieved by combining two routes and using one vehicle rather than two (Doyuran and Catay, 2010).

The basic savings concept expresses the cost savings obtained by joining two routes into one route as illustrated in Figure 2.1, where point 0 represents the depot.

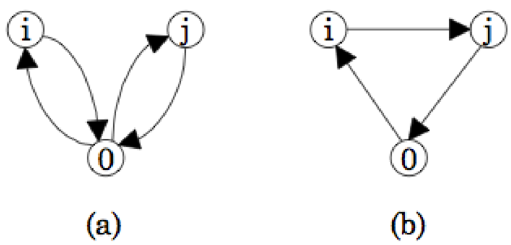

Figure 2.1 Illustration of Saving Concept (Lysgaard, 1997)

The algorithm starts from the initial solution where each route has only one customer and a corresponding vehicle. At the start, the number of vehicles is equal to the number of customers. New iteration each should reduce the number of vehicles unifying two routes that give maximal savings, e.g. reduction of the overall distance or time. There are two variants of algorithm: one with sequential and other with parallel construction of routes (Clarke and Wright, 1964).

Initially in Figure 2.1 (a) customers *i* and *j* are visited on separate routes. An alternative to this is to visit the two customers on the same route, for example in the sequence *i-j* as illustrated in (b). Because the transportation costs are given, the savings that result from driving the route in (b) instead of the two routes in Figure 2.1 (a) can be calculated.

By denoting the transportation cost between two given points i and j by c_{ij} , the total transportation cost D_a in (2.1) is:

$$D_a = c_{0i} + c_{i0} + c_{0j} + c_{j0} \qquad (2.1)$$

Equivalently, the transportation cost D_b in Figure 2.1, which corresponds to follow equation:

$$D_b = c_{0i} + c_{ij} + c_{j0} \qquad (2.2)$$

By combining the two routes one obtains the savings obtained is $S_{ij,}$:

$$S_{ij} = D_a - D_b = c_{i0} + c_{0j} - c_{ij} \qquad (2.3)$$

Where:

c_{i0} : distance between node i and 0.
c_{0j} : distance between node 0 and j.
c_{ij} : distance between node i and j.

Relatively large values of S_{ij} indicate that it is attractive, with regard to costs, to visit points i and j on the same route, such that point j is visited immediately after point i.
When two routes $(0,...,i,0)$ and $(0,j,..., 0)$ can feasibly be merged in to a single route $(0,...,i,j,...,0)$,a distance savings is generated. This version is known like version sequential.

Steps in Sequential Saving Algorithm

Step 1. Savings Computation. Compute the savings $S_{ij} = c_{i0}+ c_{0j}- c_{ij}$ for all $i , j =1, ..., n$ and $i \neq j$ .Create n vehicles routes $(0,i,0)$ for $i =1, ...,n$. Order the savings in descending order.

Step 2. Best Feasible Merge. Starting from the top of the savings list, execute the following. Given a saving S_{ij} , determine whether there exist two routes, one containing arc or edge $(0, j)$, the other containing arc or edge $(0,i)$ that can feasibly be merged (i, j) . If so, combine these two routes by deleting $(0, j)$ and $(i,0)$ and introducing

Step 3. Route Extension. Consider in turn each route $\left(0,i,...j,0\right)$ Determine the first saving $S_{k\,i}$ or $S_{j\iota}$ that can feasibly be used to merge the current route with another route containing arc or edge $(k, 0)$ or containing arc or edge $(0,\iota)$. Implement the merge and repeat this operation to the current route. If no feasible merge exists, considerer the next route and reapply the same operations. Stop when no route merge is feasible.

The Algorithm 2.1 shows the description of the operating principle of saving algorithm sequential.

Algorithm 2.1 Pseud-code of Sequential Algorithm (Batarra and Vigo, 2007)

For $i,j,(j>i) \leftarrow (i=1,j=2)$ **to** $(i=n-1,j=n)$
    do $S_{i,j} \leftarrow c_{0,i} + c_{j,0} - c_{i,j}$ **!Fill Matrix M**
Sort *Matrix M,* **filling** list ***L***
$S_{h,k} \leftarrow$Firs saving in ***L***
Nroutes ← n
While ((List L not void) and $(S_{h,k} > 0))$
do
    $S_{h,k} \leftarrow$ *First* $S_{i,j} \in$ **L** *not yet considered*
    if (Merge Feasibility *(h, k)* ==**YES***)*
      Merge$(Route_h, Route_k)$
      N_{routes}
return Nroutes

The Algorithm 2.1 is able to quickly found high-quality solutions to standard benchmarking problems; in this context benchmarking is defined as a point of reference by which the results can be measured in terms of execution time and quality solution. However, there is a version a parallel of saving algorithm, which provides a better solution and it is described in the following section (M.Battarra, 2007).

2.1.1 Parallel Saving Algorithm

Two versions of the Clarke & Wright algorithm are proposed in the literature: parallel and sequential. The best feasible merges of sub-tours are performed in the parallel approach, whereas the route extension is considered in the sequential approach. Therefore, the parallel version dominates the sequential saving method (Laporte and Semet, 2001). In this method *m* routes at a time are built, which are built simultaneously, choosing at each iteration the “best unrouted customer”, and inserting him in the “best position of the best route” among the *m* current routes, when the *m* current routes are completed, the procedure is iterated by considering *m* new routes.

General Specifications

- *Internal customers:* A customer who is neither the first nor the last at a route cannot be involved in merge operations.

- *Customers in the same route:* If the customers suggested by the saving $S_{i,j}$ are the extremes of the same route (the first or the last) the merge operation cannot be performed (no sub-tour are allowed)

Steps in Parallel Saving Algorithm

In this section is described the parallels saving algorithm proposed by Paessens 1988, which has two important phases, the first phase of this algorithm is: all pairs of customers are calculated, and all pairs of customer points are sorted in descending order of the savings. Second phase, from the top of the sorted list of point pairs one pair of points is considered at a time. When a pair of points *i-j* is considered, the two routes that visit *i* and *j* are combined (such that *j* is visited immediately after *i* on the resulting route), if this can be done without deleting a previously established direct connection between two customer points, and if the total demand on the resulting route does not exceed the vehicle capacity. In this case was only required one pass through the list.

Parallel Saving Algorithm (Paessens, 1988)

Step 1. Initialization. Initial route for each client *i*, through route construction $(0, i, 0)$.

Step 2. Saving Calculate. Calculate S_{ij} for each pair of clients *i* and *j*.

Step 3. Ordering. Sort the pairs *(i, j)* according to non-increasing values of S_{ij}.

Step 4. Next pair customers. Consider the next pair (i, j) with increase cost.

Step 5. Merge. If i^* and j^* are extreme vertices of two partial routes and these two routes can be merged: insert arc (i^*, j^*) in the solution.

Step 6. Best union. If $S_{i^*j^*}$ is the maximum value S_{ij} and it is not considered yet and r_{i^*} and r_{j^*} are respectively the routes containing the clients i^* and j^*. If i^* is the last client of r_{i^*} and j^* is the first client of r_{j^*} and the combination of r_{i^*} and r_{j^*} is feasible so, combine both routes. Delete $S_{i^*j^*}$ considerations future, if yet there are saving to be examined go to *step 3*, the other way finish.

Step 7. Not considerate. If not yet considered pairs exist: repat *step 5*.

Step 8. Completed. Complete the routes by connecting the corresponding extreme vertices with the depot.

2.2 ENHANCEMENTS OF THE PARALLEL CLARKE AND WRIGHT ALGORITHM

Several authors proposed developments of the algorithm savings. These developments may be categorized as adaptations to the savings formula, methods to speed up computation time and improvements to the route merging process. In this section is introducing the parameters (λ and μ) proposed by Gaskell and Yellow.

The enhancements proposed by (Gaskel, 1967) and (Yellow, 1970) show that the formula becomes higher when the distance between customers *i* and *j* is smaller relative to their distances to the depot. As a consequence, the saving method tends to produce good routes at the beginning. In the case when the distances of customers *i* and *j* to the depot are long whereas the distance between them is short the corresponding savings value will be large, placing it at the top of the savings list. In other words, the outermost customers (by example the customers with shorter distance between relative to their distances to the depot) are forced to be placing in the same route at the early stages. Eventually, the algorithm constructs circular-shaped routes beginning from the outermost customers and proceeds towards the inner customers. Having noticed this weakness of CW method, which prevents the merging of possible less expensive routes, (Gaskel, 1967) and (Yellow, 1970) parameterized the savings formulation as follows:

$$S_{ij}= c_{i0}+ c_{0j}-\lambda\, c_{ij} \qquad (2.4)$$

As the parameter λ increases from zero, more emphasis is placed on the distance between the customers rather than their distances to the depot.

Alternatively Paessens (1988) introduced a second term to the Gaskell and Yellow’s formula in an attempt to collect more information about the distribution. To find better solutions it is a better approach to make use of the following savings function:

$$S_{ij}=[\, c_{i0}+ c_{0j}- \lambda c_{ij}]+[\mu\left|c_{i0}+ c_{0j}\right|] \qquad (2.5)$$

Where μ is in the second term, which is a positive constant. The inclusion of the new term in (3) may exploit the asymmetry information between customers *i* and *j* regarding their distances to the depot. Nevertheless, this information adds an unfair savings to the certain customer pairs in many cases, a customer extremely close to the depot and another one very distant from the depot as such.

The next section describes the basic theory about Tabu Search and Granular Tabu Search as a part of development the improvement phase.

2.3 LOCAL SEARCH

Local search is a method for solving computationally hard optimization problems; it can be used on problems that can be formulated as finding a solution maximizing a criterion among a number of candidate solutions. Local search algorithms move from solution to another solution in the space of candidate solutions (search space) by applying local changes until a solution considered optimal is found or a time bound is elapsed (Arts and Lenstra, (2003) .

The shows the method of local search, which is an iterative method that start to initial solution S_0, generally feasible, which generate a sequence of solutions S_i choosing new iteration each the best neighbouring solution.

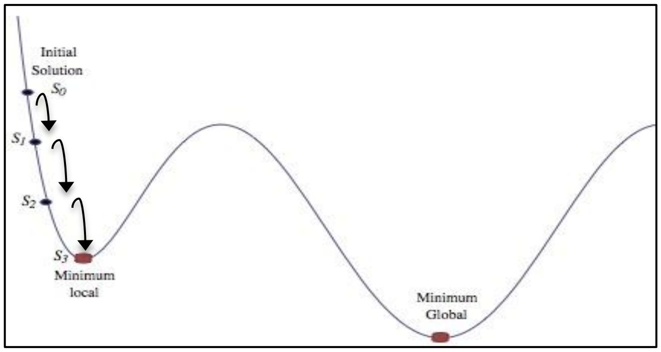

Figure 2.2 Graphic Representation of Local Search Method

Each *S* solution in the solutions space has an associated cost $z(S)$, where S_1 is better than other S_2 if $z(S_1) < z(S_2)$ .Therefore, each $z(S_i) < z(S_{i-i})$ $\forall\ i$, each new solution is better than before. In some iteration of this procedure find it a local minimum and cannot find a better solution, here procedures of search stop, obtained the best solution find it (Aarts & Lenstra, 2003).

The neighborhood solution is a set of others solutions that can find it from the initial solution, through of some movement, generally simple. The cardinality of a neighborhood is the number of moves that are neighbours of a generic solution. At each iteration, the best solution of the neighborhood that improves the current one is selected as the new current solution and the process is iterated until no improving move exists, for example until the current solution is a local optimum with respect to the current neighborhood. The time required by iteration each of a local search algorithm depends on both the cardinality of the neighborhood and on the time needed to generate each solution, check it is feasibility, and evaluate its cost. In most cases, the time per iteration is bounded by a polynomial function of the instance size. The number of iterations to be performed to reach the local optimum may be large and, in the worst case, generally grows exponentially with the instance size (Toth and Vigo, 2003).

Therefore, local search fall into the error of find a local minimum, which can be so far of the global minimum. A way to forbidden it is include movements which do not improve and change the selection condition of neighbouring solutions, allowing worse solutions. So, the next iteration the method will choose the solution with a local minimum in that it will have the minimum cost, generating a continuous cycle (Glover,1995). A way to forbidden this behaviour is including a memory mechanism to remember the visited solutions and forbidden it a definite quantity iterations, by the way the search carry out an exploration by other zones of solutions space, such can be execute tabu search (See Figure 2.2).

Algorithm 2.2 Pseudo code of Local Search

```
S0 = Initial Solution
S* = S0
 While Not Stop Criterion
   S_bar = Best move of neighborhood (S*)
    if (S_bar < S*)
       S* = S_bar
   End if
 End While
```

2.4 TABU SEARCH

According to Oklobdzija (2002) TS is a Metaheuristics that can be superimposed on any algorithmic method if this method constructs new solutions from already existing solutions by applying a sequence of moves. Such moves can be of a different nature: adding or removing a vertex from a route, adding or removing an item from a knapsack, etc. What will be considered as a move depends on a particular problem instance and on a context in which it is used. To avoid cycling TS uses tabu tenure restriction of some moves for a number of algorithm iterations. Then one says that these moves are declared tabu. Tabu tenure can be either fixed or dynamic. A fixed tabu tenure mean that moves are always penalized for a predefined number of iterations and dynamic tabu tenure means that this number of iterations changes while algorithm runs. According to Glover and Laguna (1997) dynamic tabu tenure can be changed every selected number of iterations during which it remains unchanged. It can also be changed each time some attribute becomes tabu.

The method performs an exploration of the solution space by moving from a solution S_i identified at iteration i to the best solution S_{i+1} in a subset of the neighborhood $N(S_i)$ of S_i. Since S_{i+1} does not necessarily improve upon S_i, a tabu mechanism is put in place to prevent the process from cycling over a sequence of solutions. The Figure 2.3 shows the cycle originated by allows a not improve move.

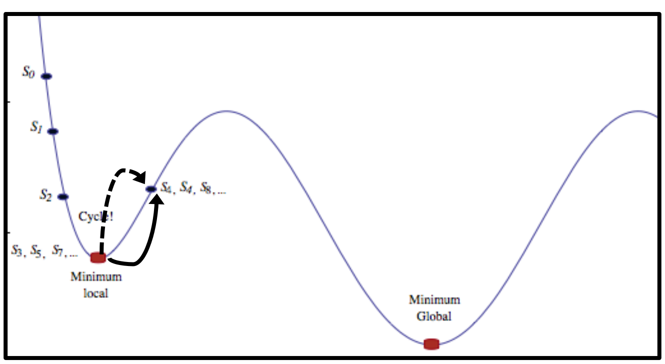

Figure 2.3 Representation of a not-improvement move.

2.4.1 Cycle in local Search

Therefore, S_{i+1} would not the best solution, will be implement a mechanism to prevent cycles is forbidden the process from going back to previously encountered solutions, but doing so would typically require excessive bookkeeping.

Instead, some attributes of past solutions are registered and any solution possessing these attributes may not be considered for θ iterations, to that effect tabu search maintains a memory structure called tabu list with the solutions historical visited or realised movements in the past. Thus, store all historical visited is so expensive, in terms memory and time, to choose just store the attributes that identified the movements that originated these solutions(Glover, 1990).

Moreover, the algorithm try to carry out a movement that belongs to tabu list (tabu movement), this movement is force to algorithm to exploit other solutions (See Figure 2.4). These prohibitions do not be definitive; just stay tuned by a quantity of iterations (tenure).

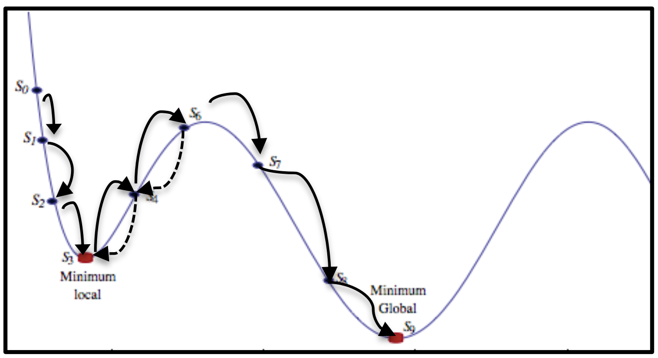

Figure 2.4 Local Search iterate until finds the best solution

It is basically a deterministic local search strategy where, at each iteration, the best solution in the neighborhood of the current solution is selected as the new current solution, even if it leads to an increase in solution cost. To avoid cycling, solutions possessing some attributes of recently visited solutions are declared forbidden or tabu for a given number of iterations, called the tabu tenure. The algorithm stops whenever a present stop criterion is satisfied. As opposed to a pure local descent, the method will thus escape from a local optimum. A short-term memory, known as the tabu list, stores recently visited solutions (or attributes of recently visited solutions) to avoid short term cycling. Typically, the search stops after a fixed number of iterations or a maximum number of consecutive iterations without any improvement to the incumbent (best known) solution (Glover, 1989).

2.4.2 Tabu Search Elements

(i) *Search space and neighborhood structure*

Tabu Search can be seen as simply the combination local search with short-term memories. It follows that two first basic elements of any Tabu Search heuristic are the definition of its search space and its neighborhood structure(Glover , 1995).

The search space of Local Search or Tabu Search heuristic is simply the space of all possible solutions that can be considered (visited) during the search. At each iteration of Local Search or Tabu Search, the local transformations that can be applied to the current solution, denoted *S*, define a set of neighboring solutions in the search space, denoted *N (S)* (the neighborhood of *S*). Formally, *N (S)* is a subset of the search space defined by (Glover, 1995):

$$N(S) = \{Solutions\ obtained\ by\ applying\ a\ \sin gle\ local\ transformation\ to\ S\}$$

.

(ii) *Tabu List*

Tabu list is one of the distinctive elements of Tabu Search when compared to Local Search. As we already mentioned, movements are used to prevent cycling when moving away from local optimal through non-improving moves. The key realization here is that when this situation occurs, something needs to be done to prevent the search from tracing back its steps to where it came from. This is achieved by declaring tabu (disallowing) moves that reverse the effect of recent moves.

Tabu List stored in a short-term memory of the search (the tabu list) and usually only a fixed and fairly limited quantity of information is recorded. In any given context, there are several possibilities regarding the specific information that is recorded. One could record complete solutions, but this requires a lot of storage and makes it expensive to check whether a potential move is tabu or not; it is therefore seldom used. The most commonly used movements involve recording the last few transformations performed on the current solution and prohibiting reverse transformations; others are based on key characteristics of the solutions themselves or of the moves.

(iii) *Memory*

An important distinction in Tabu Search arises by differentiating between short-term memory and longer-term memory. Each type of memory is accompanied by its own special strategies. The most commonly used short-term memory keeps track of solution attributes that have changed during the recent past, and is called regency-based memory.

(iv) *Tabu Tenure*

Managing Regency-Based Memory: The process is managed by creating one or several tabu lists, which record the tabu attributes and implicitly or explicitly identify their current status. The duration that an attribute remains tabu (measured in numbers of iterations) is called its tabu tenure. Tabu tenure can vary for different types or combinations of attributes, and can also vary over different intervals of time or stages of search.

(v) *Aspiration Levels and Aspiration Criteria*

Expanding the issue of defining tabu conditions at various levels of restrictiveness, an important element of flexibility in tabu search is introduced by means of aspiration criteria. The tabu status of a solution is not an absolute, but can be overruled if certain conditions are met, expressed in the form of aspiration levels. In effect, these aspiration levels provide thresholds of attractiveness that govern whether the solutions may be considered admissible in spite of being classified tabu. Clearly a solution better than any previously seen deserves to be considered admissible.

So, Tabu List is sometimes too powerful: they may prohibit attractive moves, even when there is no danger of cycling, or they may lead to an overall stagnation of the searching process. It is thus necessary to use algorithmic devices that will allow one to revoke (cancel) tabu. These are called aspiration criteria. The simplest and most commonly used aspiration criterion (found in almost all TS implementations) consists in allowing a move, even if it is tabu, if it results in a solution with an objective value better than that of the current best-known solution (since the new solution has obviously not been previously visited). The key rule in this respect is that if cycling cannot occur, tabu can be disregarded. Therefore, aspiration criteria can be defined over subsets of solutions that belong to common regions or that share specified features (such as a particular functional value or level of infeasibility).

(vi) *Candidate List Strategies*

The aggressive aspect of TS is reinforced by seeking the best available move that can be determined with an appropriate amount of effort. It should be kept in mind that the meaning of best is not limited to the objective function evaluation. (As already noted, tabu evaluations are affected by penalties and inducements determined by the search history. They are also affected by considerations of influence as subsequently characterized.) For situations where $N^*(x)$ is large or its elements are expensive to evaluate, candidate list strategies are used to restrict the number of solutions examined on a given iteration.

(vii) *Termination Criteria*

The following termination criteria can be used with TS. So, TS stops when:

- the iterations limit is reached
- the specified objective function value is reached
- Solution is not improved for a particular number of iterations
- The algorithm running time limit is reached

TS described above, sometimes can successfully solve difficult problems, but in most cases, additional elements have to be included in the search strategy to make it fully effective. The most important of these are:

(viii) *Intensification*

The idea behind the concept of search intensification is that, as an intelligent human being would probably do, one should explore more thoroughly the portions of the search space that

seem "promising" in order to make sure that the best solutions in these areas are indeed found. From time to time, one would thus stop the normal searching process to perform an intensification phase. In general, intensification is based on some intermediate-term memory, such as a regency memory, in which one records the number of consecutive iterations, that various "solution components" have been present in the current solution without interruption.
Intensification is used in many Tabu Search implementations, but it is not always necessary. This is because there are many situations where the search performed by the normal searching process is thorough enough. There is thus no need to spend time exploring more carefully the portions of the search space that have already been visited, and this time can be used more effectively as we shall see right now (Gendreau, 2002).

(ix) *Diversification*

Diversification technique makes TS extremely powerful. It diversifies the search process and helps it to move to the new regions where possibly better solutions can be found. If a solution space is highly volatile – has a lot of local optimums, then diversification is especially helpful because it helps to overcome peaks and troughs in the solution space. So, this algorithmic mechanism that tries to alleviate this problem by forcing the search into previously unexplored areas of the search space. It is usually based on some form of long-term memory of the search, such as a frequency memory, in which one records the total number of iterations (since the beginning of the search) that various "solution components" have been present in the current solution or have been involved in the selected moves.

There are several approaches for implementation of the diversification. The first approach is to use a restart method. The method consists in applying rarely use attributes to the current or best known solution and restarting the search process. The second approach is to use a continuous diversification. This method diversifies the search process when the algorithm runs. As Klein (2000) states one can use a frequency based memory to continuously diversify the search. Using information provided by this frequency based memory one can ban attributes that were frequently used during the search process for a number of iterations. This will lead to diversification, for rarely used attributes will be used more often, and thus new solutions will be explored.

A large part of the recent research in TS deals with various techniques for making the search more effective. These include methods for exploiting better the information that becomes available during search and creating better starting points. One of them techniques is Granular Tabu Search proposed by (Toth and Vigo, 2003).

2.5 GRANULAR TABU SEARCH

In the last fifteen years, several Metaheuristics have been put forward for the solution of the VRP; many tabu search heuristics have been proposed to the vehicle routing problem and theirs variants. Therefore, typically perform a thorough exploration of the solution space is necessary carry out thousand iterations to obtain high-quality solutions which demand a large computing time (Toth and Vigo, 2003). Each iteration generally consists of the exploration neighborhoods and exchange neighboors.

Tabu search clearly stands out as the best heuristic for the VRP like above was mentioned. Over the last ten years, several implementations have been developed and tested. It is fair to say they have been highly successful in tackling this difficult problem. This success is due in part to a number of key ideas contained in several implementations: the allowance of infeasible solutions during the search, the use of self-adjusting parameters, continuous diversification, adaptive memory and granularity.

Toth and Vigo (2003) define an effective implementation, which belongs to candidate-list family called Granular Tabu Search (GTS), which is defined like a mechanism, which is able to reduce the computational effort, especially for large instances by not considering some of the unpromising solution components (in their case, the long edges). It is to select a set of the nearest neighboors (plus the depot) for each customer, and at each iteration, only moves involving one member of the nearest neighboors set will be considered where the size of the set of the nearest neighboors can be selected by considering the instance characteristics and the requirements of the solution quality (or the time available for computation).

This method allows drastic reduction in the computational time requested in each iteration of Tabu Search since the list of possible moves in the neighborhood is restricted, removing elements that have no real chance of belonging to the optimal solution. This can be seen as an intensification mechanism, by reason of is because Granular Tabu Search searches a smaller neighborhood, which it is faster than the original Tabu Search.

The objective to implement Granular Tabu Search is reached by using neighborhoods that can be examined in much less time than the traditional ones but without considerably affecting the quality of the solutions found. This method proposes to derive granular neighborhoods as restrictions of other known neighborhoods, by discarding a large quantity of unpromising moves and actually exploring only a small subset of them, containing the most promising ones (Toth and Vigo, 2003).

The advantages of this method in firs to fall, it is found to be one of the least “intrusive” ways of modifying a successful solution approach while keeping its main features intact, and particularly the basic structure of the neighborhoods used within the search. Moreover, it increases applicability of the proposed method and simplifies its extension to other problems for which tabu search and other Metaheuristics proved to be effective, but not efficient in terms of time requirements. Last but not least, it allows one to evaluate in a direct way the benefits of the proposed method with respect to a tabu search algorithm that uses the same neighborhoods.

2.5.1 Granular Tabu Search to Vehicle Routing Problem

When, as is generally the case, the VRP is defined on a complete graph, it may be observed that “long” (in others word high cost) arcs have a small probability of being part of high quality solutions. The Shows test problem classical of CVRP (Toth and Vigo, 2003), through of this Table 2.1 the authors give the key of implement granular neighborhood. This Table 2.1 contains the instance name, the number of customers n, the number of available vehicles K, and the value of the best know solution z^*. In addition, the average cost of arcs in the best know solution, $\bar{z}^*=z^*/(n+k)$, is compared with the minimum, maximum, and average arc cost in the complete graph.

Table 2.1 Comparison of the Average Arc Cost in the Best-Known Solution with Respect to the Minimum, Maximum, and Average Arc Cost in the Graph for Some Classic Euclidean VRP Instances.

					Arc Cost		
Problem	*n*	*k*	*z**	$\bar{z}^*$	Min	Max	Average
E051-05e	50	5	524.61	9.54	2.24	85.63	33.75
E076-10e	75	10	835.26	9.83	2.24	85.28	34.13
E101-08e	100	8	826.14	7.65	1.41	91.83	34.64
E151-12c	150	12	1028.42	6.35	0.00	91.83	33.92
E200-17c	199	17	1291.29	5.98	0.00	91.83	33.24
E101-10c	100	10	819.56	7.45	1.00	96.18	40.27
E121-07c	120	7	1042.11	8.21	0.00	114.98	54.52

The Table 2.1 considers some classic Euclidean VRP test problems. By considering the distribution of the arc costs associated with this type of test problem it may be easily seen that the majority of the large arcs have cost larger than $\bar{z}^*$. A similar behavior can be observed in almost all known test problems for VRP and DCVRP. Therefore, a possible way to speed up the search of a neighborhood is to limit as much as possible the evaluation of moves that try to insert "long" arcs in the current solution.

z'

The procedure to generate granular neighborhoods starts from the original complete graph $G=(V,A)$, later define a new sparse graph, $G'=(V,A')$ with $|A| \ll n^2$, where n^2 is the problem algorithm complexity. This sparse graph includes all the arcs that should be considered for inclusion in the current solution: for example, all the "short" arcs and relevant subset of other important arcs and a set I of other important arcs, such as those incident to the depot or belonging to high-quality solutions founded until the moment. This is:

$$A' = \{(i,j) \in A : c_{ij} \le \varphi\} \cup I \qquad (2.6)$$

By the way, the search in granular neighborhood considers just the moves can be generated by arcs belonging to G', that is the moves that implicate at least some "*short*" arc. These arcs of A' are directly used as move generators to determine the other arcs involved in a particular move of the original neighborhood.

An arc is "short"; hence it belongs to the sparse graphic in A' if its cost is not greater than the granularity threshold value, defined as:

$$\varphi = \beta * \frac{z'}{(n+k)'} \qquad (2.7)$$

Where β is a suitable positive sparsification parameter, and z' is the value of a heuristic solution, for example, determined by the heuristic Clarke and Wright (1964).

The computational experience carry out by Toth and Vigo (2003) confirmed a basic tabu search algorithm that uses the granular version of basic neighborhoods was able to determine high quality solutions within running times comparable to those of constructive heuristics.
In terms of quality of the solution obtained based on basic Tabu Search algorithm using granular neighborhoods defined by different values of the sparsification parameter β ranging from 0.5 to 5.0. Perhaps surprisingly, the solution quality is not a monotone increasing function of the sparsification parameter, in other words, of the overall computational effort.

2.5.2 Efficient Search and Diversification

The key factor in the granular paradigm is that the search of granular neighborhoods may be efficiently implemented, in quite a natural way, by explicitly taking into account the sparse graph G' associated with the neighborhood.

Another crucial issue connected with the granular paradigm is that a dynamic modification of the structure of the sparse graph associated with the granular neighborhood provides a easy way of including intensification and diversification during the search. For example, by modifying the sparsification parameter "β", the number of arcs currently included in the sparse graph is altered; hence a possibly different new solution is obtained at the end of the search. To this end, the Tabu search algorithm may alternate between long intensification steps, associated with a small "β" value, and short diversification steps in which "β" is considerably increased, and evaluation of the moves is possibly modified to favor the possible inclusion of new (longer) arcs in the solution evaluated.

The next chapter describes the implementation of the metaheuristic development in this thesis project.

CHAPTER 3 IMPLEMENTATION

This chapter describes in detail the design and execution of the proposed algorithm in this thesis project, which includes solution initial, where employment Parameterized Saving Algorithm and improvement phase was as well the routes building. The second phase or improvement carried out the implementation of GTS and neighboors buildings defined by moves 2-optimal and relocate procedures. The platform and tools used for development of the code C++ and the test instances evaluated on 34 instances with 51 to 484 customers.

3.1 GENERAL FRAMEWORK

Toth and Vigo (2003) introduced GTS, which thesis project was based on. The general framework describes all procedure carried out for the construction of algorithm Metaheuristic proposed. It is composed of two phases, the first one initial solution and second one improve phase; chapter 4 describes the theoretical foundations, which provide an easy to understand its description. The algorithm 5.1 shows the pseudo code used to solve the proposed problem DCRVP; it be tested in 34 benchmark instances.

Algorithm 3.1 Pseudo code of proposed algorithm for the DCVRP

```
Read instance
Read specific parameters
Generate initial solution S_0 by ECW
 S*=S_0
Initialization of tabu list
Initialization Candidate List
 i = 0
  While non-accur criteria do
   Generate neighborhood of S_i based on Candidate List
   Choose best move no Tabu S_i+1
   Apply best move generate
   Update Candidate List
   Update penalization
   Update Tabu List
    If z (S_i+1) < z ( S*S*) then
       S* = S_i+1
    End If
 i= i+1
   End While
return  S*
```

The Algorithm 3.1 starts reading instances to Capacitated Vehicle Routing Problem and Distance Constrains Vehicle Routing Problem, in the code implementation detect the problem type through of one instruction; by the way start to read the parameters. In the case of CVRP, the service time was equal to zero and length route equal to a big number (infinity).

This Algorithm 3.1 can be implemented a particular instance, or instances set, which reads the necessaries parameter to execute the code; this is instance name, type (DCVRP or CVRP), capacity, maximum length, service time, vehicles and nodes. The initial solution will be explained in the section 3.2.

The next step is the implementation of tabu search, which explore the solution space attempting in each step improve the current solution through of the move generation mechanism creates the neighboring solutions by changing one attribute or a combination of attributes of the initial solution.

The Tabu List attribute could refer, for example, to arcs connecting a pair of customers. Once a neighboring solution is identified, it is compared against the current solution. If the neighboring solution is better, it replaces the current solution, and the search continues.

The Algorithm 3.1 generates a neighborhood of the current solution S_i through generation and evaluation of neighborhoods obtained to implement Two-opt and relocate. In this thesis project, just the local search was carried out in the neighborhood composed by 2-optimal and Relocate procedure, which will be explained in the next section. The neighborhood obtained will contain valid solution of S_i with the objective of found a new solution S_{i+1} that replace to S_i . The solution S_{i+1} may be lower cost and it not is tabu. The current solution S_{i+1} will be marked like tabu by tenure iterations to forbidden that came back in the last solution S_i by tenure iterations. Finally, decrease the value of tenure in the tabu list and if the current solution is the better found it up to the present time; it will be update of S^* by S_{i+1}. This iterative procedure carries out until found a stop criterion, which can be a maximum number of iterations without obtained improvement solution.

More detailed, the next sections explain the above. So, is included the implemented code in C++, to explain overall design and implementation of this Metaheuristics proposed like thesis project.

3.1.1.1 Execute mode

In this thesis project, the proposed algorithm starts with execution of instances set downloaded from: http://www.or.deis.unibo.it/research_pages/ORinstances/VRPLIB/VRPLIB.html web site, which belongs, DEIS (Dipartimento di Elettronica, Informatica e Sistemistica, Bologna Italy) - Operations Research Group, which 14 classical instances were proposed by Christofides, Mingozzi and Toth (1979) and 20 large-scale instances (with $|V|$ varying from 201 to 484) proposed by Golden, Wasil, Kelly and Chao (GWKC instances) (Bruce, Edward, James, and Ming 1998).

For execute mode, the program reads by one-instruction the file with txt extension. The first line reads instance name the next instruction decides if it is CVRP or DCVRP type, if it is DCVRP the parameters to execute are:

- NAME: D051-06c (instance name)
- COMMENT: Christofides, Mingozzi and Toth, 1979 (Who proposed the instance)
- TYPE: DCVRP (vehicle routing problem variant)

- DIMENSION: 51 (number of customers plus one depot)
- EDGE_WEIGHT_TYPE: EUC_2D (Euclidean Distances)
- CAPACITY: 160 (vehicle capacity)
- MAX. LENGTH: 200 (maximum length route)
- SERV. TIME: 10 (service time of each customer)
- VEHICLES: 6 (numbers of vehicles available)
- NODE_COORD_SECTION (for each node there are id, x, y)
- DEMAND (for each node are id, demand)

Parameters to CVRP are:

- NAME: D051-05e (instance name)
- COMMENT: Christofides, Mingozzi and Toth, 1979 (Who proposed the instance)
- TYPE: CVRP (vehicle routing problem variant)
- DIMENSION: 51 (customers quantity)
- EDGE_WEIGHT_TYPE: EUC_2D (Euclidean Distances)
- CAPACITY: 160 (vehicle capacity)
- VEHICLES: 5 (numbers of vehicles available)
- NODE_COORD_SECTION (for each node there are id, x, y)
- DEMAND (for each node are id, demand)

3.2 CONSTRUCTION OF INITIAL SOLUTION

Initial solution development in this thesis project was obtained implementing Enhancement of Parallel Clarke and Wright Algorithm proposed by Passens (1988), who adding two parameters to classical saving formula λ and *μ.* In this thesis project was used it, *λ* like route shape parameter, which avoid circumference formation of routes that are usually produced by the original saving algorithm parallel. So, their motivation in using the positive parameter *λ* is to avoid circumference formation of routes that are usually produced by the original CW algorithm. In other words, this parameter helps to reshape the routes by taking only non-negative values in order to find better quality solutions (Doyuran and Catay, 2010).
Consequently, with the aim of expanding the exploration ability of the algorithm is added a second parameter, *μ*, to Gaskell and Yellow formula in an attempt to collect more information about the distribution of the customers.

McDonald (1972) shown that with any fixed λ, results, which are far from optimum, may be obtained and that there was no value of λ, which was significantly, better than any other value. Paessens (1988) proposed ranges $0 < \lambda \leq 3$ and $0 \leq \mu \leq 1$, respectively.

First, was used Euclidean Distance to calculate the distance matrix base on coordinates (x_i, y_i) belong to *n* customers and the depot $\left(x_0, y_0\right)$, which is a square matrix order *n*+1, with distance *d* (*i, j*) between two points $\left(x_i, y_i\right)$ and $\left(x_j, y_j\right)$, the Euclidean Distance formula is:

$$\sqrt{(x_i - x_j)^2 + (y_i - y_j)^2} \qquad (3.1)$$

Algorithm 3.2 Pseudocode Saving Algorithm Parallel with λ and μ parameters

Build N_{routes}
$r_i = (0, i, 0)$
For $i, j, (j > i) \leftarrow (i = 1, j = 2)$ **to** $(i = n-1, j = n)$
do $S_{i,j} \leftarrow [c_{i0} + c_{0j} - \lambda c_{ij}] + [\mu |c_{i0} + c_{0j}|]$ **!Fill Matrix M**
Sort Matrix M descendent order, filling list L
$S_{h,k} \leftarrow$ **First saving in** L
$Nroutes \leftarrow n$
while *((List* L **not void**) && $(S_{h,k} > 0)$) && $((r_{i*}, r_{j*} \leq Q)$&&$(r_{i*}, r_{j*} \leq$ **TimeService**)
&&$(r_{i*}, r_{j*} \leq$ **RouteLarge**))
do
if ($(0,i,...,0)$ && $(0,j,..., 0)$ || $(0,i,...,0)$ && $(0,...,j,0)$ || $(0,...,i,0)$&&$(0,j,...,0)$ ||
$(0,..., i, 0)$ && $(0,...,j,0)$)
$S_{i*,j*} =$ **max** $S_{i,j}$
Let r_{i*} **be the route containing** h
Let r_{j*} **be the route containing** k
If (($i*$ **is the last in** r_{i*} **and** $j*$ **is the first shop in** r_{j*}) &&
(the combination is feasible))
then combine r_{i*} **and** r_{j*}
$S_{i*,j*} \leftarrow$ **First** $S_{i*,j*} \in L$ **not yet considered**
do
Merge$(Route_{i*}, Route_{j*})$
if (**Merge Feasibility** $(i*, j*) ==$ ***YES***)
Merge Feasibility $= N_{routes}$
End **While**
End **For**

McDonald (1972) shown that with any fixed λ, results, which are far from optimum, may be obtained and that there was no value of λ, which was significantly, better than any other value, therefore by using factorial analysis was found the parameters values λ and μ, which is base on ranges $0 < \lambda \leq 3$ and $0 \leq \mu \leq 1$, respectively proposed by Paessens (1988).

The values parameters were found it with base on values proposed by Passens (1988), and increasing his interval. For this reason was used a factorial analysis above mentioned with the [0.0, 2.0] interval.

3.2.1 ROUTE CONSTRUCTION APPROACHES

In the first step of the savings algorithm the savings for all pairs of customers are calculated, and all pairs of customer points are sorted in descending order of the savings. Subsequently, from the top of the sorted list of point pairs one pair of points is considered at a time. The route construction select arcs simultaneously until solution have been created if these can be done without deleting a previously established direct connection between two customer points, and if the total demand on the resulting route does not exceed the vehicle capacity.

In this, thesis project was found the values parameter established by Passens (1988), Enhancement of Parallel Clarke and Wright Algorithm, for almost all instances. So, do not create a violation of vehicle capacity, maximum length, and vehicles number. The vehicles number restriction was established by instances set proposed by Toth and Vigo (2003), which was used like benchmark in this thesis project. Others requirements constraints are:

One route is valid if start and finish in the same depot and satisfy the conditions before mentioned and considering the follow:

a) Each customer has a service time, and it is considering on maximum route length.
b) At each iteration, the "best feasible route merge" is chosen, and the route is joined.
c) The current route is ended when no more customers can be inserted.
d) The approach does not consider if it is better to insert a customer in the current route or to wait, and to insert the customer in one of the following routes.

Cases considered were:

Case 1. When two routes $(0,i,...,0)$ and $(0,j,..., 0)$ can feasibly be merged in to a single route $(0,...,i,j,...,0)$, a distance savings is generated.

Case 2. When two routes $(0,i,...,0)$ and $(0,...,j,0)$ can feasibly be merged in to a single route $(0,...,i,j,...,0)$, a distance savings is generated.

Case 3. When two routes $(0,...,i,0)$ and $(0,j,...,0)$ can feasibly be merged in to a single route $(0,...,i,j,...,0)$, a distance savings is generated.

Case 4. When two routes $(0,...,i,0)$ and $(0,...,j,0)$ can feasibly be merged in to a single route $(0,...,i,j,...,0)$, a distance savings is generated.

Considering the previous analysis, the obtained results in the initial solution are presented in the Table 3.1. It table contains in the column one instances set, which was used in this thesis project, column 2 shows the best solution based on review literature, 3 and 4 columns has the fixed parameters values found so factorial analysis (λ, μ), respectively. The constrain routes number is in the column 4, column 5 contains the objective function values in this first phase; the difference in percentage respect to best known solution is in the column 6.

Table 3.1 Results obtained in Initial Solution to the Distance Constrained Vehicle Routing Problem.

Instance	*Best Know Solution*	λ	μ	Initial Solution obtained by Clark & Wright Algorithm		
				Routes C&W	*Distance C&W*	*%GAP C&W*
D051-06c	553.43	1.3	0.3	6	595.31	7.57%
D076-11c	909.68	0.9	0.4		**Infeasible**	
D101-09c	865.94	1.7	0.3	9	942.70	8.86%
D101-11c	866.37	1.2	0.2	11	869.62	0.37%
D121-11c	1541.14	0.7	0.1	11	1,583.25	2.73%
D151-14c	1162.55	1.3	0.0	14	1,222.06	5.12%
D200-18c	1395.85	1.0	0.0		**Infeasible**	
D201-05k	6460.98	1.7	0.5	5	6,691.04	3.56%
D241-10k	5627.54	0.2	0.6	10	5,807.07	3.19%
D281-08k	8412.8	1.2	0.4	7	8,665.56	3.00%
D321-10k	8447.92	0.8	0.1		**Infeasible**	
D361-09k	10181.75	1.8	0.9	9	10,614.61	4.25%
D401-10k	11036.22	0.7	0.9	10	11,414.50	3.43%
D441-11K	11663.55	1.3	0.8	11	12,409.47	6.40%
D481-12k	13624.52	1.9	1.1	11	14,109.85	3.56%
E051-05e	524.61	0.8	0.9	5	563.90	7.49%
E076-10e	835.26	1.0	0.1	10	866.30	3.72%
E101-08e	826.14	1.6	0.3	8	865.60	4.78%
E101-10c	819.56	1.2	0.4	10	826.00	0.79%
E121-07c	1042.11	1.6	0.6	7	1,065.08	2.20%
E151-12c	1028.42	2.0	0.7	12	1,101.82	7.14%
E200-17c	1291.29	1.4	0.2	17	1,370.05	6.10%
E241-22k	707.79	1.8	0.9	22	746.22	5.43%
E253-27k	859.11	1.3	0.9	26	896.56	4.36%
E256-14k	583.39	0.8	1.3	14	610.39	4.63%
E301-28k	997.52	1.5	1.2	28	1,051.51	5.41%
E321-30k	1081.31	1.2	0.4	30	1,144.23	5.82%
E324-16k	741.56	0.6	1.2	16	763.31	2.93%
E361-33k	1366.86	1.4	0.3	33	1,441.40	5.45%
E397-34k	1345.23	1.8	1.1	34	1,411.05	4.89%
E400-18k	918.42	1.0	1.1	18	966.93	5.28%
E421-41k	1820.09	1.4	1.0	38	1,917.83	5.37%
E481-38k	1622.69	1.2	1.2	38	1,709.72	5.36%
E484-19k	1107.19	0.4	1.5	19	7,599.06	586.34%

However, the results found it on the first phase were not able feasible for all the instances. Since, D076-11c, D200-18c, and D321-10k (indicated in grey colour in the Table 3.1) these instances were by vehicles number, vehicle capacity and maximum length. For this reason in the second phase are applied, section 3.3, capacity and maximum length penalties proposed by Gendreau, Hertz and Laporte (1994) and relocate procedure to vehicle number constraint.

The next chapter describes the improvement phase, it was introduced GTS, building neighbourhoods and implementation of as soon a constraint to eliminate excess routes as well the implementation of capacity and maximum length penalties.

3.3 IMPROVEMENT PHASE

In the improvement phase of the developed Metaheuristic proposed in this thesis project, the TS was applied for enhancement the initial solution and to find a feasible solution for all instances set. The base of this TS solution was the granularity proposed by Toth and Vigo (2003). So, in this phase, was implemented 2-*Optimal* and *Relocate* as neighbourhoods to local search. 2-*Optimal* is a procedure carries out exchanges between arcs and *Relocate* like procedure to transfers a customer from one route to another route. Iteratively the customers are removed from infeasible circuits and connected with the best in the best position of a feasible circuit (route) considering they do not overload the target route.

The neighborhood that results consists only of all the solutions that can be obtained by transferring one node from its current position to another one and in routes different.

From the initial solution S_0 the next step is to execute the implementation algorithm based on Tabu Search and implementing granularity. Each step of this general procedure attempts improving the current solution according to cost visiting neighbors solution defined by the operations set mentioned in the section 4.2.

The neighborhoods structures η are used within the proposed implementation are based on traditional arc-exchange local moves, namely inter and intra route 2-*Optimal* and *Relocate*. The selection of neighborhood structures is dynamic. At each iteration, given the allowed set of neighbors $\phi_\eta(S_0)$, the best admissible neighbor s' replaces the current solution (s_0), while the forward and reversal attributes of the corresponding local move are stored within the tabu list. During the exploration of the neighboring space, the typical aspiration criterion is followed, that is, higher evaluation of the neighbour compared to the current solution found during the exploration of the solution space. Finally, the termination condition *maxiter* bounds the maximum number of TS iterations without observing any further improvement.

3.3.1 2-*Optimal* procedure

The 2-*Optimal* neighborhood consists of all feasible solutions that can be obtained by removing two connections between customers, and connecting them in another way. These customers can be in two different routes or within the same route (Helsgaun 2000). It is considered like an edge exchange heuristics and is widely used to improve vehicle routing solutions.
The 2-*Optimal* eliminates two edges and reconnected the paths to obtain a new cycle. This procedure tries to improve the tour replacing two of it is edges by other two edges, finally iterates until do not find a possible improvement. Basically, it works as follow:

1. *Start with an initial solution and define this solution to be the current solution.*

2. *Generate all solutions in the neighborhood of the current solution by applying all modifications associated with the method under consideration.*
3. *Select the best solution in this neighborhood and define this solution to be the new current solution.*
4. *Go back to step 2.*

The time required to solve the problem using this procedure can be increased faster as the size of the problem grows; the neighborhood generated in the step 2 is typically polynomial. Dramatically to implement Granular Tabu Search and diversification and intensification strategies reduce the computational time.

In this thesis project was considerate moves in the same tour, called intra- route and inter-routes in two different tours.

The intra –route exchanges arcs in the same route that is to say if a route *k* has the costumers *i*-*i*+1 and *j* - *j*+1, the arc that marge to *i* with *j*+1 is replacing by the arc that merge to *i* and *i*+1. So, the arc that merges *j* and *i*+1 is exchanged by the arc *j* and *j*+1 (See Figure 3.1).

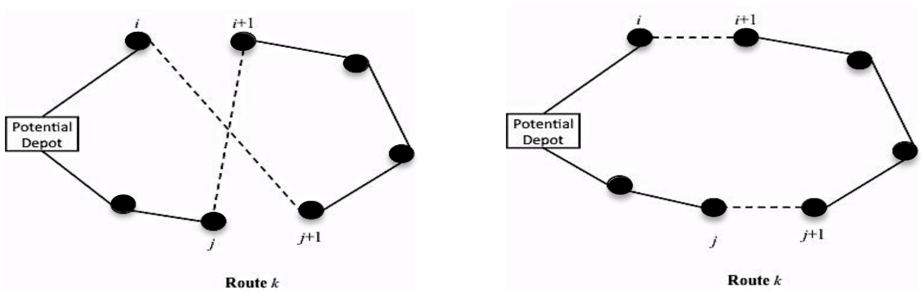

Figure 3.1 A 2-Optimal move intra-route.

The inter-route was applied in two different routes, which the arcs from one route to another. It can be implemented if exist a route *k* with *i*-*i*+1 arcs and another route *m*, which has *j*-*j*+1 arcs (See Figure 3.2).

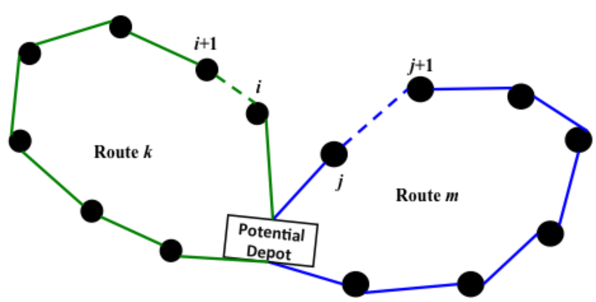

Figure 3.2 Two routes *k, m* with *i*-*i*+1 and *j*-*j*+1, respectively.

In the route *k* the arc *i*+1 is merged with *j* (*i*+1-*j*) and route m the arc *j*+1 is merged with *i* (*j*+1-*i*). See Figure 3.3.

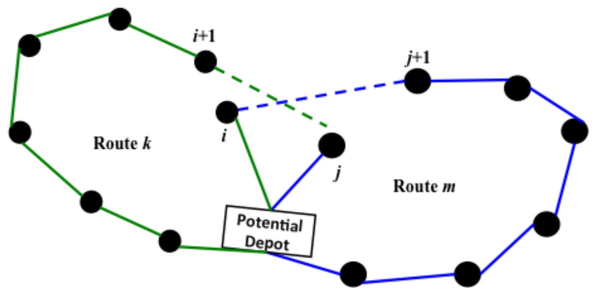

Figure 3.3 *2-Optimal* moves inter route.

The pseudo code shows in the Algorithm 3.3 the general heuristic algorithm 2-*Optimal*. It consists in analyze all the vertices; in each step choose the best move 2-*Optimal* associate each arc.

Algorithm 3.3 Pseudo code of general procedure 2-Optimal

```
Initialization
Consider a Hamiltonian initial cycle
move =1
 While (move=1)
   move = 0. Labeled all vertex as non-explored
     While (exist vertex non explored)
       Select the vertex i non-explored
     Explore all movements 2-opt that included the arc
     from the node i to the next node
      If some movement explored reduce the length of cycle
         execute the best move
          move = 1
         Else labeled i like explored
       End If
      End While
    End While
```

In the next section is implemented *Relocate* like to place customers in a strategic position with the aim of decrease the vehicle number.

3.3.2 Relocate

Relocate is a procedure used commonly, to minimizing the number of routes or, equivalently, the number of vehicles used. It insertion procedure was executed to constraint vehicles number, *Relocate*, because in the initial solution tree instances were infeasible. It tries to build a feasible one.

In this phase customers are iteratively removed from infeasible circuits and inserted in the best position of a feasible route. Possible insertions are considered only if they do not overload the target route.

In this, thesis project considerate a solution as a set S of v routes s a set like a solution $R_1,...,R_v$ where $v \in [1,\bar{v}]$, $R_v = (v_0, v_{r_1}, v_{r_2}, ..., v_0)$, and each vertex v_i ($i \geq 1$) belongs to exactly one route. These routes may be feasible or infeasible with respect to the capacity and length constraints. For convenience, is write $v_i \in R_r$ if v_i is a component of R_r , and $(v_i, v_j) \in R_r$ if v_i and v_j are two consecutive vertices of R_r. With any feasible solution S, is associated the objective function (Gendreau, Hertz, and Gilbert Laporte 1994a):

$$F_1(S) = \sum_r \sum_{(v_i, v_j) \in R_r} c_{ij} \quad (3.2)$$

Also, with any solution S (feasible or not), is associated the objective:

$$F_2(S) = F_1(S) + \alpha \sum_r \left[\left(\sum_{v_i \; R_r} q_i \right) - Q \right]^+ + \beta \sum_r \left[\left(\sum_{(v_i, v_j) \; R_r} c_{ij} + \sum_{v_i \; R_r} \delta_i \right) - L \right]^+ , \quad (3.3)$$

Where $[x]^+ = \max(0, x)$ and α and β are two positive parameters. In the solution is feasible $F_2(S)$ and $F_1(S)$ coincides; otherwise, $F_2(S)$ incorporates two penalties terms for the excess vehicle capacity and excess route duration. At any step of the algorithm, F_1^* and F_2^* denote respectively the lowest value of $F_1(S)$ and $F_2(S)$ so far encountered. Also, S^* is the best know feasible solution and S^* , the best know solution (feasible or not).

Where $r = \{R_1,...,R_v\}$ are the set of routes. Let S be the set of routes for which exist an optimal solution. Each solution $\sigma \in S$ has an associated set of neighbors, $N(\sigma) \in S$, called the neighborhood de σ. Each solution $\sigma' \in N(\sigma)$ can be reached directly from σ by a move. A move is a transition from σ to σ' by means of a move operator.

In this algorithm proposed was considerate the mentioned before as soon the rule based on smallest cardinality route, which force to eliminate customers smallest cardinality route to be inserted another route, of this way eliminate routes to converge to a feasible solution by the number of routes.

The penalty used was:

$$\gamma(S) = \begin{cases} 0, & r \leq v \\ Min_{i \in r}, \; |r_i| & r > v \end{cases} \quad (3.4)$$

Where r = number of routes, v = number of vehicle, $Min_{i \in r}$ is the smallest cardinality route k and $\gamma(S)$ route number feasible.

The penalty used $\boldsymbol{\gamma}(\boldsymbol{S})$ was planted as follow:

If a route *k*, exists, it moves a customer or customer another routes. More specifically, the vehicle routes with the smallest cardinality are removed iteratively fro the routing plans and all corresponding customers are placed into a waiting list. At iteration of TS, the possibility of the feasible relocation of the waiting listed customer is examined to the vehicle routes modified by the local search. If one or more feasible relocation moves (insertion positions) are found, these are immediately executed, and corresponding solution is denoted as new TS local optimum solution. The above procedure is repeated until all waiting listed customers are served. The major advantage of this classic route elimination procedure is that TS deal only with feasible solutions whether the best solution found upon termination is sub-optimal or optimal in terms of fleet size.

3.4 IMPLEMENTATION OF GRANULAR TABU SEARCH

Actually is one of the principal ideas in Metaheuristics, Tabu Search specifically, because of the simplicity of its structure.

In this section is explained of Granular Tabu Search (GTS) applied to CVRP and the DCVRP. The algorithm Granular Tabu Search was initialized with the heuristic solution obtained to use Clarke and Wright (1964) with enhancement, above mentioned in the section 3.2.

In the initial solution was infeasible to tree instances by vehicle constraint, for this reason in the improvement phase was applied *Relocate* to less route, which was executed violating the vehicle capacity or the maximum route length constraints, for this reason was implemented the penalties proposed by Gendreau, Hertz and Laporte (1994). So, *2-optimal* like neighborhood of GTS, which contains all arcs exchanges selecting the best according to this heuristic.

By visiting of infeasible solutions is allowed during the search and, as usual, their costs are modified by adding to the routing cost a penalty α_C multiplied by the sum of the excess loads of the overloaded routes, plus a penalty α_D multiplied by the sum of the excess lengths of infeasible routes with respect to the maximum length constraint (Gendreau, Hertz, & Laporte, 1994) and a new penalty for excess routes α_R .

The values of the penalties dynamically, α_C , α_D and α_R were updated during the search in the range $[\alpha_{\min}, \alpha_{\max}]$. In particular, every iterations, if the previous visited solution was feasible with respect to the capacity constraints, then α_C is set to $\max\{\alpha_{\min}, \alpha_C / 1.1\}$, whereas if all were infeasible it is set to $\{\alpha_{max}, 1.1 * \alpha_c\}$. The updating rules for α_D is analogous. In our computational testing, α_{min}= 1 and α_{max} = *10 * value of the solution Clarke & Wright*; α_R is set to α_R = *100 * value of the solution Clarke & Wright.*

The excess of capacity is calculated like as:

$$\sum_{r \in Routes} Max\left(0.0, Maximum\ Capacity\ Route\ \text{-}\ Route\ Demand_r\right) \quad (3.5)$$

The excess of length of route is calculated like as:

$$\sum_{r \in Routes} Max\left(0.0, Maximum\ Length\ \text{-}\ Time\ Service\ Route_r - Length\ Route_r\right) \quad (3.6)$$

The excess of the number of routes is calculated like:

$$\gamma = \begin{cases} 0,\ RoutesNumber \leq Vehicles\ Number \\ Min_{r \in Routes} Customers_r, RoutesNumber \geq Vehicles\ Number \end{cases} \quad (3.7)$$

3.4.1.1 Other strategies of diversification

A strategy implement proposed algorithm was: starting of the current solution, if many iterations with granular tabu search were carried out, the algorithm is not able to come back to be feasible found or it is infeasible yet, the current solution is update to the best feasible solution found it in all iterations and the list tabu come back real size, magic number 7, by default.

This algorithm proposed uses a multiple granular neighborhood based on neighborhoods 2-*Optimal* and *Relocate*. The granular neighborhood is obtained by considering the arcs in the sparse graph defined by all the arcs below **the current granularity threshold, plus all arcs incident to the depot, and those belonging to the best solution found and to the current one**.

After each iteration, the arcs inserted by the performed move are added to the sparse graph. The sparse graph is rebuilt from scratch every time the current solutions change between feasible to infeasible solutions. I used a sparsification factor $\beta = 1.25$, which computationally gave the best performance. However, for Euclidean VRP and DVRP instances, β values between 1.25 and 2.50 are generally appropriate.

A move is considered Tabu if it tries to reinsert an arc removed in one of the previous moves. The Tabu tenure t for each move performed is an integer uniformly distributed random variable in the interval $\left[t_{\min}, t_{\max}\right]$. I used t_{min}= 7 when the solution is feasible or infeasible; but t_{max}= 7 for the feasible solution and t_{max}= 50 for infeasible solution.

Finally, I used the granularity-based diversification described in the previous section. Whenever the current best solution is not improved after n_d iterations, the sparsification factor is increased to β_d, a new sparse graph is determined and n_h iterations are performed starting from the best solution found. Then, the sparsification factor is resetting to the original value and the search continues. We used n_d = *15 * number of customers*, β_d= *1.75*, and n_h = *number of customers*.

This algorithm is a dynamically updated, so infeasibility penalties and the tabu tenure definition. The values of the remaining parameters were experimentally determined as those allowing for the best compromise between solution quality and computational effort. Clearly, the Figure 3.4 and Figure 3.5 show efficient algorithm design, so time computational less used.

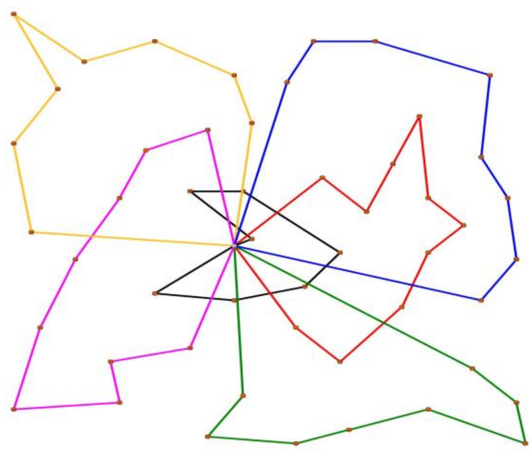

Figure 3.5 Initial solution to implement saving algorithm enhancement (λ = 1.3, μ = 0.3) to instance D051-06c with Z*=593.31, 0.001 seconds.

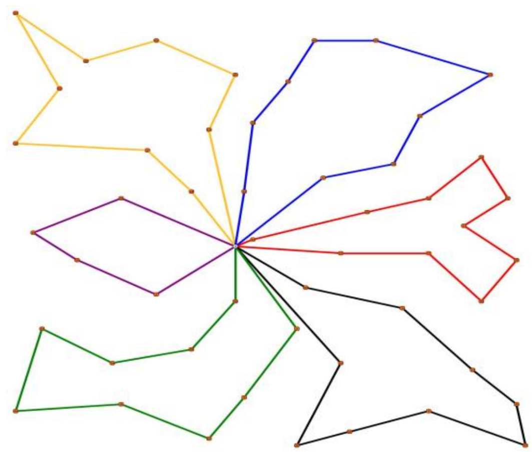

Figure 3.4 Optimal solution obtained by to implement granular tabu search in instance D051-06c with Z* = 560.24, 0.01 seconds.

CHAPTER 4 ANALYSIS OF THE RESULTS

The initial solution was used values lambda (λ) and mu (μ), both were evaluated in the interval [0.0, 2.0]. Beta (β) is the parameter of granularity and *maxiter * number of customers* is the number of iterations that execute Tabu Search.

In this thesis project, the Tabu Search Elements applied were:

Tenure: contains Tabu search tenure, iterations quantity, which a movement is considered Tabu, in my case was called magic, this implementation iteration quantity was used the magic number seven, which can grow slowly when explore infeasible regions to maximum size of 50. As a general principle, Tabu restrictions that are more stringent, as measured by the degree to which they limit the range of admissible moves, lead to somewhat smaller values for best Tabu list sizes than restrictions that are less stringent (Glover, 1990).

Quantity iterations: denote the maximum operations quantity of principal cycle of Tabu Search. This algorithm considered ***maxiter * number of clients***, maxiter is configured with 10 value.

Value ($\boldsymbol{\beta}$)

My parameter called beta, is used to calculate the maximum value of the distances between customers that will belong in neighborhood granular.

This value includes all the “short” arcs and a relevant subset of other important arcs, such as those incidents to the depot and those belonging to the best solutions encountered so far. Therefore, this does not mean “long” arcs are never inserted in the current solution, but that moves involving only “long” arcs are not considered.

Increasing factor $\boldsymbol{\beta}$

The maximum distance was calculated with the factor β**,** which increases continuously. It factor β is actualized dynamically, this way modify the structure of the sparse graph associated with the granular neighborhood provides a simple way of including intensification and diversification during the search. For example, by modifying the sparsification parameter β, the number of arcs currently included in the sparse graph is altered; hence a possibly different new solution is obtained at the end of the search. To this end, the tabu-search algorithm may alternate between long intensification steps, associated with a small value β, and short diversification steps in which β is considerably increased and the evaluation of the moves is possibly modified to favour the possible inclusion of new (longer) arcs in the solution.

The results show a typical behaviour, in terms of quality of the solution obtained within *maxiter*number of clients, mentioned before*, of the basic tabu search algorithm using granular neighborhoods defined by different values of the sparsification parameter β ranging from 0.5 to 5.0. Perhaps surprisingly, the solution quality is not a monotone increasing function of the sparsification parameter. In fact, with the benchmark problems in the literature considered in, the best results are typically obtained with β values between 1.0 and 3.5, which select about 10– 20% of the arcs of the complete graph.

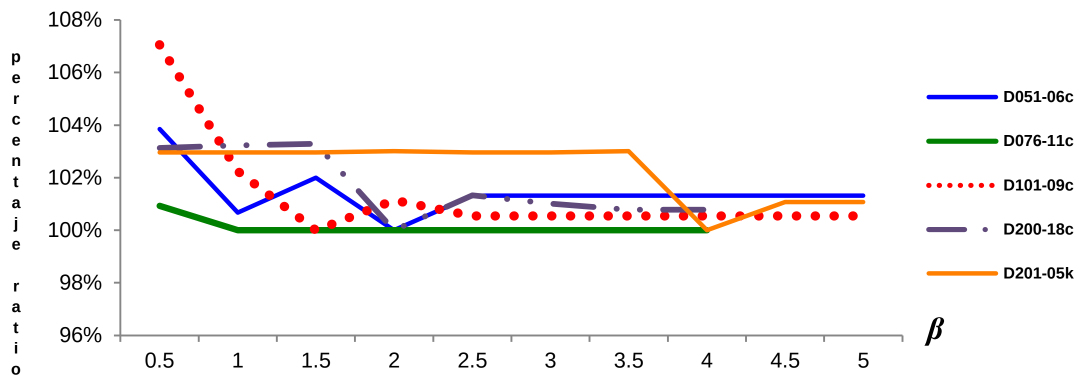

Figure 4.1 Performance of Tabu-Search applied Granular Neighborhoods and different values of the sparsification parameter β to Distance Constraints Vehicle Routing Problem (DCVRP).

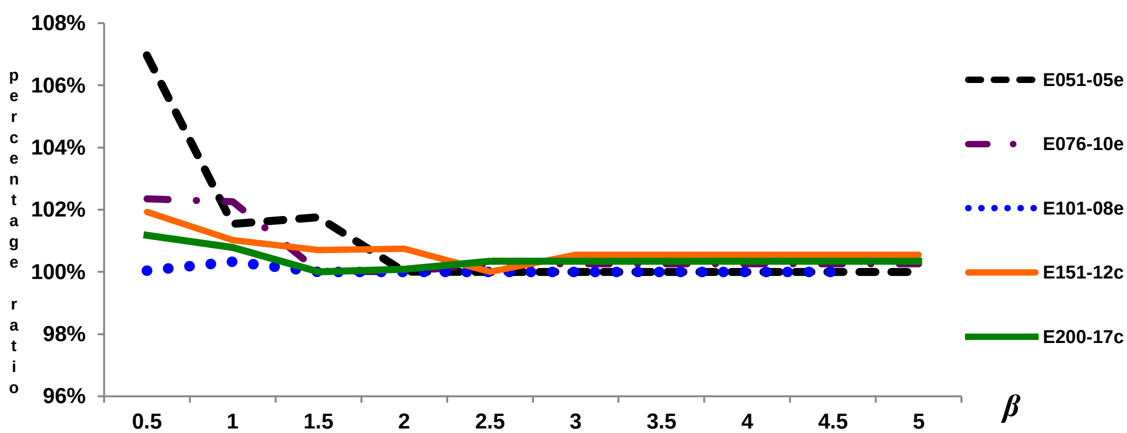

Figure 4.2 Performance of Tabu-Search applied Granular Neighborhoods and different values of the sparsification parameterβ, to Capacitated Vehicle Routing Problem (CVRP).

4.1 PROCEDURES

The Granular Tabu Search algorithm proposed by Toth and Vigo (2003) uses a multiple granular neighborhood based on four basic neighbourhoods. In this project was implemented a multiple granular neighborhood based in two operations 2–opt in inter routes (between 2 routes) and intra route (TSP). Relocate as inter route considering penalty proposed and mentioned before.

So, the granular neighborhood is obtained by considering the arcs in the sparse graph defined by all the arcs below the current granularity threshold, plus all arcs incident to the depot, and those belonging to the best solution found and to the current one. After each iteration, the arcs inserted by the performed move are added to the sparse graph. The sparse graph is rebuilt from scratch every *maxiter*, defined before. I used a sparsification proposed by Toth and Vigo factor β =1.25, which computationally gave the best performance. To my case DCVRP instances, β values used values of 1.25 and 2.50, arcs "short" 1.25 and arcs "long" 2.50.

4.2 COMPUTATIONAL EXPERIMENTS

Algorithm development described in the previous sections, based on Metaheuristic proposed by (Toth & Vigo, 2003), was implemented in C++ and compiled in the IDE **Qt** creator 2.4.1 and the GNU GCC compiler of Xcode 4.3 on Mac OS X Lion 10.7.4 compiler. All experiments were performed on MacBook Pro 8.1 with name machine Intel Core i5 2.4 GHz clock and 4 MB RAM memory.

The graphic representation of results was generated so instructions in general code and making a link with Graph editor ***yEd*** version 3.9.1, which generate a file (.gml), that contains corresponding graphs each instance proposed.

The computational testing considered several Euclidean CVRP and DCVRP instances from the literature with up to about 484 customers, which are generally used as a standard benchmark for VRP algorithms. The first set consists of the 14 instances (with $|V|$ varying from 51 to 200) proposed by Christofides, Mingozzi and Toth (CMT instances) (Christofides, 1979). The second set consists of the 20 large-scale instances (with $|V|$ varying from 201 to 484) proposed by Golden, Wasil, Kelly and Chao (GWKC instances) (Bruce et al., 1998)

Toth and Vigo (2003) show the test instances are denoted by a name that allows one to determine their characteristics quickly. The instances are divided in two groups the first one is the group *'E'* for Euclidean, which corresponding to VRP, without constraints length distance, and time service; in this case when the code is executed the length service is equal to biggest value and the service time is equal to zero. The second one group '*D*' for Euclidean DCVRP with constraints of length distance and service time your values are reading from corresponding instance. The set of instances *'E'* belongs to CVRP and 'D' belongs to DCVRP; n is the number of vertices of the corresponding graph (including the depot vertex); k is the number of available vehicles, the vehicle capacity (C) and, for DVRP instances, the route maximum length (L), the service time ($\bar{q}$) for each customer; data source identifies the paper where the instance data were first described. For example, E051-05e denotes the classic 50-customer instance proposed by Christofides and Eilon (1969).

Table 4.1 Summary of the data to VRP and DVRP instances used for computational testing

Name Instance	n	k	C	L	$\bar{q}$	Data Source	Prev. Best Solution	Solution
D051-06c	50	6	160	200	10	Christofides et al. (1979)	553.43	Taillard (1993), Gendrau et al. (1994)
D076-11c	75	11	140	160	10	Christofides et al. (1979)	909.68	Taillard (1993)
D101-09c	100	9	200	230	10	Christofides et al. (1979)	865.94	Taillard (1993), Gendrau et al. (1994)
D101-11c	100	11	200	1040	90	Christofides et al. (1979)	**866.37**	Osman (1993)
D121-11c	120	11	200	720	50	Christofides et al. (1979)	1541.14	Taillard (1993)
D151-14c	150	14	200	200	10	Christofides et al. (1979)	1162.55	Taillard (1993)
D200-18c	199	18	200	200	10	Christofides et al. (1979)	1395.85	Rochat and Taillard (1995)
D201-05k	200	5	900	1800	0	Golden et al. (1998)	6460.98	Golden, Wasil Kelly and Chao (1998)
D241-10k	240	10	550	650	0	Golden et al. (1998)	5627.54	Golden, Wasil Kelly and Chao (1998)
D281-08k	280	8	900	1500	0	Golden et al. (1998)	8412.8	Golden, Wasil Kelly and Chao (1998)
D321-10k	320	10	700	900	0	Golden et al. (1998)	8447.92	Golden, Wasil Kelly and Chao (1998)
D361-09k	360	9	900	1300	0	Golden et al. (1998)	10181.75	Golden, Wasil Kelly and Chao (1998)
D401-10k	400	10	900	1200	0	Golden et al. (1998)	11036.22	Golden, Wasil Kelly and Chao (1998)
D441-11K	440	11	900	1200	0	Golden et al. (1998)	11663.55	Golden, Wasil Kelly and Chao (1998)
D481-12k	480	12	1000	1600	0	Golden et al. (1998)	13624.52	Golden, Wasil Kelly and Chao (1998)
E051-05e	50	5	160	-		Christofides and Eilon (1969)	**524.61**	Gendreau et al. (1994)
E076-10e	75	10	140	-		Christofides and Eilon (1969)	835.26	Taillard (1993)
E101-08e	100	8	200	-		Christofides and Eilon (1969)	826.14	Taillard (1993), Gendreau et al. (1994)
E101-10c	100	10	200	-		Christofides et al. (1979)	819.56	Taillard (1993)
E121-07c	120	7	200	-		Christofides et al. (1979)	1042.11	Taillard (1993)
E151-12c	150	12	200	-		Christofides et al. (1979)	1028.42	Taillard (1993)
E200-17c	199	17	200	-		Christofides et al. (1979)	1291.29	Rochat and Taillard (1995)
E241-22k	240	22	200	-		Golden et al. (1998)	707.79	Golden, Wasil Kelly and Chao (1998)
E253-27k	252	27	1000	-		Golden et al. (1998)	859.11	Golden, Wasil Kelly and Chao (1998)
E256-14k	255	14	1000	-		Golden et al. (1998)	583.39	Golden, Wasil Kelly and Chao (1998)
E301-28k	300	28	200	-		Golden et al. (1998)	997.52	Golden, Wasil Kelly and Chao (1998)
E324-16k	323	16	1000	-		Golden et al. (1998)	741.56	Golden, Wasil Kelly and Chao (1998)
E361-33k	360	33	200	-		Golden et al. (1998)	1366.86	Golden, Wasil Kelly and Chao (1998)
E397-34k	396	34	1000	-		Golden et al. (1998)	1345.23	Golden, Wasil Kelly and Chao (1998)
E400-18k	399	18	1000	-		Golden et al. (1998)	918.42	Golden, Wasil Kelly and Chao (1998)
E421-41k	420	41	200	-		Golden et al. (1998)	1820.09	Golden, Wasil Kelly and Chao (1998)
E481-38k	480	38	1000	-		Golden et al. (1998)	1622.69	Golden, Wasil Kelly and Chao (1998)
E484-19k	483	19	1000	-		Golden et al. (1998)	1107.19	Golden, Wasil Kelly and Chao (1998)

The best value solution known found reported in the literature, actually. The last column of the table contains the values of the new best solution found during our overall testing activity. All test instance data, as well the most of the best known solution data, had been attained from the authors.

In the last file can look the solution values marked by (*) in were obtained values optimums to instances D101-11c and E051-05e and E051-05c according to results obtained by (Toth and Tramontani, 2008)

The objective of computational testing was to evaluate the behaviour of the MXGTS algorithm designed and implemented. Table 4.2 compares the results obtained on the fourteen classic CVRP and DCVRP to classical test instances 14 and 20 large-scale instances proposed by Paolo Toth and Tramontani (2008).

The implemented algorithm in this thesis project was called MXGTS (columns marked MXGTS), which was compared with Granular Tabu Search proposed by Paolo Toth and Daniele Vigo (2003), Xu and Kelly (KH) (Xu & Kelly, 1996) and Rego and Roucairol (Rego and Rouncairol, 1996), the columns marked GTS, XK, and RR, respectively.

In this thesis project is possible look the time of MXGTS is usually small compared with the other algorithms; however the quality solution is less competitive compared with the others algorithms, because in this thesis project just was implemented 2-*optimal* and relocate procedures.

For each instance of Table 4.2,Table 4.3 and Table 4.4 are reported the percentage ratio of the solution value obtained by each algorithm with respect to the best-known solution value, as well the computing time, which was expressed in seconds.

Table 4.2 Comparison of the Results of 14 instances classical on the test proposed by Golden et al. (1998)

Intance	Best Sol.	MXGTS			GTS		GHL		XK		RR	
		Sol.	%	Time	%	Time	%	Time	%	Time	%	Time
E051-05e	524.61	524.61	100.00	0.40	100.00	48.60	100.00	360.00	100.00	1795.20	100.00	51.00
E076-10e	835.26	865.49	103.62	0.20	100.40	132.60	100.06	3228.00	100.00	2928.00	100.27	1008.00
E101-08e	826.14	861.02	104.22	0.27	100.29	143.40	100.40	1104.00	100.00	4315.80	100.17	2034.00
E101-10c	819.56	822.78	100.39	0.30	100.00	66.00	100.00	960.00	100.00	3396.60	100.00	73.20
E121-07c	1042.11	1049.24	100.68	0.55	100.07	190.80	103.01	1332.00	100.00	5473.80	100.14	378.00
E151-12c	1028.42	1091.08	106.09	0.83	100.47	270.60	100.75	3528.00	100.11	8994.00	102.52	1632.00
E200-17c	1291.29	1357.65	105.14	2.56	102.08	450.00	102.42	5454.00	100.55	16351.20	103.64	975.00
VRP average			102.88	0.73	100.47	186.00	100.95	2280.86	100.09	6179.23	100.96	878.74
D051-06c	553.43	560.24	101.23	0.10	100.00	51.60	100.00	810.00	100.00	1840.20	100.00	190.20
D076-11c	909.68	972.98	106.96	1.75	101.21	165.00	100.39	3276.00	106.15	6127.80	100.00	1386.00
D101-09c	865.94	889.01	102.66	0.62	100.41	174.00	100.00	1536.00	101.78	5889.00	100.27	516.00
D101-11c	866.37	866.86	100.06	0.58	100.00	84.60	100.00	3942.00	105.64	9178.80	100.02	565.20
D121-11c	1541.14	1,573.22	102.08	1.14	100.28	560.40	102.12	3552.00	105.02	12105.00	100.59	120.00
D151-14c	1162.55	1,222.06	105.12	5.60	100.91	340.20	101.31	4260.00	-	10084.80	101.40	933.00
D200-18c	1395.85	1,472.27	105.47	6.38	102.86	546.60	101.62	5988.00	103.11	22102.20	101.79	3121.20
DVRP average			103.37	2.31	100.81	274.63	100.78	3337.71	103.62	9618.26	100.58	975.94
Overall average			**103.12**	**1.52**	**100.64**	**230.31**	**100.86**	**2809.29**	**101.86**	**7898.74**	**100.77**	**927.34**

Table 4.3 Comparison of the Results on the Very Large Instances.

Instances	Best Sol.	MXGTS			GTS		XK		RTR	
		Sol.	%	Time	%	Time	%	Time	%	Time
E241-22k	707.79	740.76	104.66	4.79	98.70	857.40	103.72	138840.00	100.00	341.40
E253-27k	859.11	889.38	103.52	4.71	98.61	685.80	100.00	87946.20	100.00	360.60
E256-14k	583.39	604.69	103.65	3.71	101.07	700.20	100.34	20412.00	100.00	1380.60
E301-28k	997.52	1,049.45	105.21	10.07	98.80	1287.00	103.63	246061.20	100.00	489.00
E321-30k	1081.31	1,141.22	105.54	9.03	99.32	870.60	101.30	94638.00	100.00	1309.80
E324-16k	741.56	758.05	102.22	7.31	100.68	949.80	100.00	30109.20	100.35	1889.40
E361-33k	1366.86	1,438.43	105.24	19.45	99.85	1803.60	102.34	343102.80	100.00	745.20
E397-34k	1345.23	1,400.28	104.09	17.01	100.38	1107.00	100.99	260404.20	100.00	1957.20
E400-18k	918.42	957.91	104.30	13.40	100.36	1987.20	100.00	51163.20	100.18	4151.40
E421-41k	1820.09	1,911.37	105.01	32.71	102.17	2583.00	103.19	650383.80	100.00	1863.00
E481-38k	1622.69	1,687.34	103.98	25.38	99.74	1384.20	100.00	536607.00	100.08	2853.00
E484-19k	1107.19	1,218.47	110.05	524.92	100.88	2574.00	100.31	69066.00	100.00	6065.40
VRP average			104.79	56.04	100.05	1399.15	101.32	210727.80	100.05	1950.50
D201-05k	6460.98	6,676.19	103.33	1.97	99.92	142.80	–	35484.00	100.00	674.40
D241-10k	5627.54	5,725.26	101.74	5.41	98.31	298.80	–	48172.20	100.00	220.80
D281-08k	8412.8	8,653.76	102.86	3.15	99.41	279.00	–	54822.00	100.00	1127.40
D321-10k	8447.92	8,913.70	105.51	12.28	99.85	496.80	–	53911.80	105.09	1359.60
D361-09k	10181.75	10,471.08	102.84	15.94	95.47	699.60	–	63763.80	101.50	1353.00
D401-10k	11036.22	11,351.06	102.85	9.74	97.89	776.40	–	104956.20	101.98	2402.40
D441-11K	11663.55	12,312.63	105.57	27.89	98.25	664.80	–	95172.00	102.16	6682.20
D481-12k	13624.52	14,083.65	103.37	16.02	101.85	907.80	–	145945.20	100.00	7356.60
DVRP average			103.51	11.55	98.87	533.25	–	75278.40	101.34	2647.05
Overall average			**104.15**	**33.80**	**99.46**	**966.20**	**101.32**	**143003.10**	**100.70**	**2298.78**

Table 4.4 Comparison general results

Instance Name	Best Know Solution	λ	μ	Initial Solution obtained by Clark & Wright Algorithm			Improvement Phase obtained Tabu Search and Granular Tabu Search							
				Routes C&W	Cost C&W	&GAP C&W	Routes GTS	Time GTS	Cost GTS	%GAP GTS	Routes TS	Time TS	Distance TS	%GAP TS
D051-06c	553.43	1.3	0.3	6	595.31	7.57%	6	0.10	560.24	1.23%	6	0.38	568.39	2.70%
D076-11c	909.68	0.9	0.4	–	–	–	11	1.75	972.98	6.96%	11	4.10	958.60	5.38%
D101-09c	865.94	1.7	0.3	9	942.70	8.86%	9	0.62	889.01	2.66%	9	4.20	880.67	1.70%
D101-11c	**866.37**	**1.2**	**0.2**	**11**	**869.62**	**0.37%**	**11**	**0.58**	**866.86**	**0.06%**	**11**	**4.36**	**866.37**	**0.00%**
D121-11c	1541.14	0.7	0.1	11	1,583.25	2.73%	11	1.14	1,573.22	2.08%	11	8.13	1,556.33	0.99%
D151-14c	1162.55	1.3	0.0	14	1,222.06	5.12%	14	5.60	1,222.06	5.12%	14	27.69	1,222.06	5.12%
D200-18c	1395.85	1.0	0.0	–	–	–	18	6.38	1,472.27	5.47%	18	49.28	1,452.44	4.05%
D201-05k	6460.98	1.7	0.5	5	6,691.04	3.56%	5	1.97	6,676.19	3.33%	5	43.65	6,554.33	1.44%
D241-10k	5627.54	0.2	0.6	10	5,807.07	3.19%	10	5.41	5,725.26	1.74%	10	73.80	5,725.87	1.75%
D281-08k	8412.8	1.2	0.4	7	8,665.56	3.00%	7	3.15	8,653.76	2.86%	7	115.38	8,649.98	2.82%
D321-10k	8447.92	0.8	0.1	–	–	–	10	12.28	8,913.70	5.51%	10	187.76	8,917.81	5.56%
D361-09k	10181.75	1.8	0.9	9	10,614.61	4.25%	9	15.94	10,471.08	2.84%	9	378.82	10,422.68	2.37%
D401-10k	11036.22	0.7	0.9	10	11,414.50	3.43%	10	9.74	11,351.06	2.85%	10	384.11	11,312.36	2.50%
D441-11K	11663.55	1.3	0.8	11	12,409.47	6.40%	11	27.89	12,312.63	5.57%	11	677.48	12,308.27	5.53%
D481-12k	13624.52	1.9	1.1	11	14,109.85	3.56%	11	16.02	14,083.65	3.37%	11	746.43	13,986.34	2.66%
E051-05e	**524.61**	**0.8**	**0.9**	**5**	**563.90**	**7.49%**	**5**	**0.08**	**535.12**	**2.00%**	**5**	**0.40**	**524.61**	**0.00%**
E076-10e	835.26	1.0	0.1	10	866.30	3.72%	10	0.20	865.49	3.62%	10	1.78	848.14	1.54%
E101-08e	826.14	1.6	0.3	8	865.60	4.78%	8	0.27	861.02	4.22%	8	3.52	856.21	3.64%
E101-10c	819.56	1.2	0.4	10	826.00	0.79%	10	0.30	822.78	0.39%	10	3.62	822.78	0.39%
E121-07c	1042.11	1.6	0.6	7	1,065.08	2.20%	7	0.55	1,049.24	0.68%	7	8.17	1,043.89	0.17%
E151-12c	1028.42	2.0	0.7	12	1,101.82	7.14%	12	0.83	1,091.08	6.09%	12	13.69	1,086.46	5.64%
E200-17c	1291.29	1.4	0.2	17	1,370.05	6.10%	17	2.56	1,357.65	5.14%	17	39.60	1,358.59	5.21%
E241-22k	707.79	1.8	0.9	22	746.22	5.43%	22	4.79	740.76	4.66%	22	67.28	740.76	4.66%
E253-27k	859.11	1.3	0.9	26	896.56	4.36%	26	4.71	889.38	3.52%	26	85.29	888.08	3.37%
E256-14k	583.39	0.8	1.3	14	610.39	4.63%	14	3.71	604.69	3.65%	14	82.38	602.62	3.30%
E301-28k	997.52	1.5	1.2	28	1,051.51	5.41%	28	10.07	1,049.45	5.21%	28	149.52	1,049.45	5.21%
E321-30k	1081.31	1.2	0.4	30	1,144.23	5.82%	30	9.03	1,141.22	5.54%	30	194.19	1,139.58	5.39%
E324-16k	741.56	0.6	1.2	16	763.31	2.93%	16	7.31	758.05	2.22%	16	173.24	759.04	2.36%
E361-33k	1366.86	1.4	0.3	33	1,441.40	5.45%	33	19.45	1,438.43	5.24%	33	284.79	1,438.43	5.24%
E397-34k	1345.23	1.8	1.1	34	1,411.05	4.89%	34	17.01	1,400.28	4.09%	34	466.61	1,392.58	3.52%
E400-18k	918.42	1.0	1.1	18	966.93	5.28%	18	13.40	957.91	4.30%	18	349.33	958.79	4.40%
E421-41k	1820.09	1.4	1.0	38	1,917.83	5.37%	38	32.71	1,911.37	5.01%	38	484.68	1,911.56	5.03%
E481-38k	1622.69	1.2	1.2	38	1,709.72	5.36%	38	25.38	1,687.34	3.98%	38	755.74	1,678.11	3.42%
E484-19k	1107.19	0.4	1.5	19	7,599.06	586.34%	19	524.92	1,218.47	10.05%	19	816.35	1,218.47	10.05%
Average						**23.40%**		**23.11**		**3.86%**		**196.64**		**3.44%**

The instances D101-11c and E051-05e were founded optimal solution equal to the optimal solution by so exact methods reported in Toth and Tramontani (2008).

The Table 4.4 with the instances D076-11c, D200-18c and D321-10k shows tree important phases, the first one is which start with an initial solution parameterized (λ, β). This initial solution was infeasible by number of routes, however to implement tabu search it accomplish the feasibility considering the penalty proposed as soon strategies of diversification. However, the computational time to tabu search is highly, here is the importance of introduce granularity with granular neighborhoods, which were implemented this way the results respect to time was excellents compared with just use tabu search (See Figure 4.3).

In the instance E484-19k start with a feasible initial solution but with a GAP of 586.45 %; homever to apply Granular Tabu Search there was obtained a GAP of 10.05%.

The computational times of Clarke and Wright do not presented because is very closed to cero. The time of Granular Tabu Search is so less compared with Tabu Search because the number or arcs is reduced by the candidate list in or the Granular neighborhoods Tabu Search. The quality of the solution of Granular Tabu Search is competitive with high quality paper. The results of the XK algorithm for the DVRP instances are those reported in Golden et al. (1998), since Xu and Kelly (1996) had not tested their algorithm on DCVRP instances.

Granular Tabu Search designed in this project is proved able to determine, in a quite short computing time, good solutions, but do not better to solutions reported by Granular Tabu Search (Toth and Vigo, 2003), whose the quality was compared with the best solutions obtained by other tabu search approaches from the literature.

In particular, the Granular Tabu Search proposed solution is worse than the GHL only in one of the seven VRP instances, and the GAP with respect to the best known solutions of Granular Tabu Search is half that of GHL. For the DCVRP instances, GHL is strictly better than Granular Tabu Search in three out of seven cases, and the average percentage ratios for the two algorithms are almost the same. The computing times of the Granular Tabu Search algorithm is on average about one fifth the equivalent computing times of GHL.

The comparison between Granular Tabu Search and XK on the seven classic VRP instances, XK generally obtains better solutions, whereas Granular Tabu Search is considerably more effective than XK on the DVRP and the Fisher's VRP instances. The computing times of the Granular Tabu Search algorithm implemented are on average **80–83** times smaller than the equivalent computing times of Paolo Toth and Daniele Vigo (Toth & Vigo, 2003).

Finally, the performance of Granular Tabu Search designed in this project is quite similar to that of algorithm RR both in terms of solution quality and overall computing times. In particular, Granular Tabu Search performs slightly better on CVRP instances and slightly worse on DCVRP ones and the average equivalent computing times are almost the same. When it runs with the complete neighborhoods, the tabu search used as a basis for the Granular Tabu Search algorithm obtained solutions on average slightly better than those of MXGTS, but required computing times of the same order of magnitude as the XK and GHL algorithms. This clearly illustrates the positive impact of candidate-list strategies within the local search methods. On the one hand, granular neighborhoods may be easily introduced into any existing local search algorithm and lead to drastic reductions of the computational effort: The Granular Tabu Search algorithm is several orders of magnitude faster than a basic tabu search using complete neighborhoods, and requires on large-scale instances about the same amount of computing times as effective constructive heuristics. On the other hand, the introduction of granular neighborhoods does not substantially affect the overall efficacy of the approach: the quality of the solutions obtained by Granular Tabu Search is comparable to those of the best available algorithms.

So, the Figure 4.3 shows correlation coefficients of 0.98, which denote that 98% dates, are adapted to third grade polynomial. Because of execution of general tabu of *n* times with computational time of $0(n)$ and execution of neighborhoods (2-opt and relocate) with computational time of $0(n^2)$ *so* $0(n) * 0(n^2) = 0(n^3)$.

The importance to implement granularity concept in vehicle routing problem, in this case with constraint distance is that the time necessary is lees that used in Tabu Search like can be seen in Figure 4.3.

General Computational Time

Instance	Time GTS	Time TS
D051-06c	0.10	0.38
D076-11c	1.75	4.10
D101-09c	0.62	4.20
D101-11c	0.58	4.36
D121-11c	1.14	8.13
D151-14c	5.60	27.69
D200-18c	6.38	49.28
D201-05k	1.97	43.65
D241-10k	5.41	73.80
D281-08k	3.15	115.38
D321-10k	12.28	187.76
D361-09k	15.94	378.82
D401-10k	9.74	384.11
D441-11K	27.89	677.48
D481-12k	16.02	746.43
E051-05e	0.08	0.40
E076-10e	0.20	1.78
E101-08e	0.27	3.52
E101-10c	0.30	3.62
E121-07c	0.55	8.17
E151-12c	0.83	13.69
E200-17c	2.56	39.60
E241-22k	4.79	67.28
E253-27k	4.71	85.29
E256-14k	3.71	82.38
E301-28k	10.07	149.52
E321-30k	9.03	194.19
E324-16k	7.31	173.24
E361-33k	19.45	284.79
E397-34k	17.01	466.61
E400-18k	13.40	349.33
E421-41k	32.71	484.68
E481-38k	25.38	755.74
E484-19k	524.92	816.35
Average	**23.11**	**196.64**

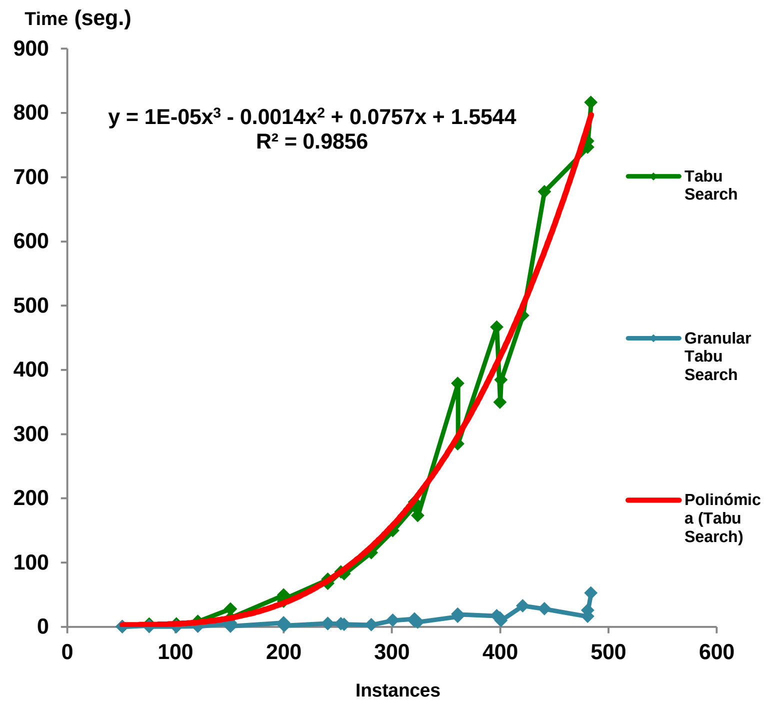

Figure 4.3 Computational time Granular Tabu Search versus Tabu Search

CONCLUSIONS AND FUTURE WORK

In this final section presents the coclusions and future work about this project, which was the result of the implemented algorithm.

Conclusions

For the purpose of selecting an efficient method so solve the Distance Constrained Vehicle Routing Problem was make a review of state art was made it. So, this review helped in the search benchmarking instances and best know solution according. This way was evaluated strategies as factorial analysis to find the saving algorithm parameters, which produced a good initial solution.

In this thesis project was implement GTS and insert move 2-*Optimal* to solve DCVRP, in this case was necessary to implement relocate by the results obtained in the initial solution, with the issue to obtain good solution in all instances, which ware compared with the best solutions according to reports of literature.

In this thesis project was presented and tested an effective implementation of candidate-list strategies to be used within Tabu-search algorithms for a wide class of graph-theoretic and combinatorial optimization problems.

The proposed approach, called granular Tabu search, was applied to the well-known symmetric capacitated and distance constrained vehicle routing problem for which several Tabu search algorithms have been presented in the literature.

These algorithms are able to obtain high-quality solutions but often require a large amount of computing time to solve large instances. Granular Tabu search is based on the use of granular neighborhoods, which include a small number of “promising” moves. Which use a simple strategy to obtain granular neighborhoods from standard ones and discussed their efficient search.

The computational testing shows the importance of using appropriate candidate-list strategies and their impact in creating better methods: on standard test instances from the literature, granular Tabu search is able to determine very good solutions within short computing times.

Attainment of initial solution competitive was obtained by a search exhaustive of a set of right parameters to find the majority of initial feasible solutions, considering value best objective function (minimize distance) and using a right and efficient implementation on C++. At difference Professor Toth and Professor Vigo, algorithm in initial solution, they used Clarke & Wright plus Patching, I use a Clarke & Wright parameterized, which use parameters λ and μ, these used to expanding the exploration ability of the algorithm, λ to design shape elliptic routes and μ to attempt to collect more information about the distribution of the customers as to whether depot.

In this algorithm was design and implemented a new strategy and penalty appropriate to eliminate automatically so Tabu Search and procedure relocate the routes of less cardinality. In the initial solution some instances were infeasible by routes numbers, whereby was designed an appropriate penalty to eliminate less routes loaded automatically by Tabu Search and the procedure Relocate so of added a penalty competent, when routes number major vehicles number eliminated.

Penalties proposed by Gendrau (Gendreau, Hertz, & Laporte, 1994) when routes that may feasible or infeasible with respect to the capacity and length constraints. This penalties implemented were using by way light with small updates in each iteration.

Implementation efficient neighborhoods with evaluation time *0(1)* in each movement.
Update strategy automatically of tabu list in Tabu Search in an interval [7,50] when it moves by infeasible regions tabu list can grow until 50, when search move by feasible solution tabu list size use “magic number “ seven. Like diversification strategy, if Tabu Search does not accomplish come back to feasible solution it returns the last best solution feasible find it.

This methodology can be implement whichever variant belongs to general VRP, these are:

- Capacitated Vehicle Routing Problem (CVRP)
- Distance Constraints Vehicle Routing Problem (CVRP)
- Vehicle Routing Problem with Backhauls (VRPB)
- Vehicle Routing Problem with Time Windows (VRPTW)
- Vehicle Routing Problem with Pick up and Deliveries (VRPPD)
- Vehicle Routing Problem with Backhauls and Time Windows (VRPBTW)
- Vehicle Routing Problem with Pick up and Deliveries and Time Windows (VRPPDTW).

Future Work

In this project just was implemented like neighborhoods 2-opt (TSP cross, inter route parallel and cross) and relocate between routes. However, the quality solution is bad compared with results obtained by Paolo Toth and Daniele Vigo whereby I propose implemented in future work Cross exchange, Flip, One point move (OPM), Or-opt, Swap, Sweep and 3-opt.

So, use other Metaheuristics like Simulated Annealing, Neighborhood Variable and Scatter Search.

Otherwise, through excellent results obtained in computational time a good idea will be implement in other new variant of vehicle routing problem like the single vehicle routing problem with deliveries and selective pickups (SVRPDSP), which is defined on a graph in which pickup and delivery demands are associated with customer vertices. The difference between this problem and the single vehicle routing problem with pickups and deliveries (SVRPPD) lies in the fact that it is no longer necessary to satisfy all pickup demands. In the SVRPDSP pickup revenue is associated with each vertex, and the pickup demand at that vertex will be collected only if it is profitable to do so. The net cost of a route is equal to the sum of routing costs, minus the total collected revenue. The aim is to design a vehicle route of minimum net cost, visiting each customer, performing all deliveries, and a subset of the pickups. A mixed integer linear programming formulation is proposed for the SVRPDSP. Classical construction and improvement heuristics, as well as a tabu search heuristic (TS), are developed and tested on a number of instances derived from VRPLIB.

REFERENCES

[1] Aarts, E., and Lenstra, K. (2003). *Local Search in Combinatorial Optimization.* Jonh Wiley.

[2] Battarra Maria and Daniele Vigo (2007). *Clarke and Wright Algorithm.* Università di Bologna, Dipartimento di Elettronica, Informatica e Sistemistica. Laboratorio di Simulazione e Ottimizzazione.

[3] Bräysy and Gendreau. (2005). *Vehicle Routing Problem with Time Windows, Part I: Route Construction and Local Search Algorithms*. Tranportation Science, 39, 104-118.

[4] Christofides, Nicos (1979). The vehicle routing problem in Combinatorial Optimization. 315–338.

[5] Baldacci Roberto, Hadjiconstantinou, and Mingozzi (2004). *An Exact Algorithm for the Capacitated Vehicle Routing Problem Based on a Two-Commodity Network Flow Formulation*. Operations Research 52(5): 723–738.

[6] Baldacci Roberto, DanieleVigo and PaoloToth (2010). *Exact Solution of the Capacitated Vehicle Routing Problem*. *In* Wiley Encyclopedia of Operations Research and Management Science.

[7] Barnhart Cynthia, Ellis L. Johnson, George L. Nemhauser, Martin W. P. Savelsbergh, and Pamela H. Vance (1998). *Branch-and-Price: Column Generation for Solving Huge Integer Programs*. Operations Research 46(3): 316–329.

[8] Bramel Julien and David Simchi-Levi (1997). *The Logic of Logistics: Theory, Algorithms, and Applications for Logistics*. Management Science. Springer.

[9] Bramel, Julien, and David Simchi-Levi (1995). *A Location Based Heuristic for General Routing Problems*. Operations Research 43(4): 649–660.

[10] Christofides Nicos and Samuel Eilon (1969). *Expected Distances in Distribution Problems*. OR 20(4): 437–443.

[11] Christofides, Mingozzi, and Toth (1979). *Combinatorial Optimization*. Wiley.

[12] Clarke, G., and J. W. Wright (1964). *Scheduling of Vehicles from a Central Depot to a Number of Delivery Points*. Operations Research 12(4): 568–58.

[13] Cordeau, Jean-Francois, Michel Gendreau, Alain Hertz, Gilbert Laporte, and Jean-Sylvain Sormany (2005). *New Heuristics for the Vehicle Routing Problem in Logistics Systems: Design and Optimization*. p. 279–297. Springer US.

[14] Doyuran T., B. Catay. (2010). *A robust enhancement to the Clarke–Wright savings algorithm*. Journal of Operations Research, 223–231.

[15] Franceschi Roberto, Matteo Fischetti, and Paolo Toth (2006). *A New ILP-based Refinement Heuristic for Vehicle Routing Problems. Mathematical Programming* 105(2/3): 471–499.

[16] Fukasawa Ricardo, Marcus Poggi Aragão, Marcelo Reis, and Eduardo Uchoa (2003). *Robust Branch-and-cut-and-price for the Capacitated Vehicle Routing Problem.* IN PROCEEDINGS OF THE INTERNATIONAL NETWORK OPTIMIZATION CONFERENCE Pp. 231–236.

[17] Gaskel, T. (1967). Bases for Vehicle Fleet Scheduling. *Journal of the Operational Research Society, 18*, 281–295.

[18] Gendreau, Michel (2003). *An Introduction to Tabu Search*. In Handbook of Metaheuristics. Fred Glover and Gary A. Kochenberger, Pp.37–54. International Series in Operations Research & Management Science, 57.Springer US.

[19] Gendreau, Michel, Alain Hertz, and Gilbert Laporte (1994). *A Tabu Search Heuristic for the Vehicle Routing Problem*. Manage. Sci. 40(10): 1276–1290.

[20] Glover, Fred (1989). *Tabu Search—Part I*. ORSA Journal on Computing 1(3): 190–206.

[21] Glover, Fred (1990). *Tabu Search*—Part II. ORSA Journal on Computing 2(1): 4–32.

[22] Glover, Fred (1995). *Tabu Search Fundamentals and Uses*.

[23] Golden, Bruce L., Edward A. Wasil, James P. Kelly, and I.-Ming Chao (1998). *The Impact of Metaheuristics on Solving the Vehicle Routing Problem: Algorithms, Problem Sets, and Computational Results*. In Fleet Management and Logistics, Pp. 33–56. Springer US.

[24] Helsgaun, Keld (2000). *An Effective Implementation of the Lin–Kernighan Traveling Salesman Heuristic*. European Journal of Operational Research 126(1): 106–130.

[25] Kytöjoki, Jari, Teemu Nuortio, Olli Bräysy, and Michel Gendreau (2007). *An Efficient Variable Neighborhood Search Heuristic for Very Large Scale Vehicle Routing Problems*. Comput. Oper. Res. 34(9): 2743–2757.

[26] Laporte, G., M. Desrochers, and Y. Nobert (1984). *Two Exact Algorithms for the Distance-Constrained Vehicle Routing Problem*. NETWORKS 14(1).

[27] Laporte, Gilbert (2009). *Fifty Years of Vehicle Routing*. Transportation Science 43(4): 408–416.

[28] Laporte, Gilbert, Yves Nobert, and Martin Desrochers (1985). *Optimal Routing Under Capacity and Distance Restrictions*. Operations Research 33(5): 1050–1073.

[29] Li, Chung-Lun, David Simchi-Levi, and Martin Desrochers (1992). *On the Distance Constrained Vehicle Routing Problem*. Operations Research 40(4): 790–799.

[30] Li, Feiyue, Bruce Golden, and Edward Wasil (2005). *Very Large-scale Vehicle Routing: New Test Problems, Algorithms, and Results*. Computers & Operations Research 32(5): 1165–1179.

[31] Nagata, Yuichi, Olli Bräysy, and Wout Dullaert (2010). *A Penalty-based Edge Assembly Memetic Algorithm for the Vehicle Routing Problem with Time Windows*. Comput. Oper. Res. 37(4): 724–737.

[32] Paessens. (1988). *The savings algorithm for the vehicle routing problem*. EuropeanJournalof OperationalResearch, 336-344.

[33] Ralphs T. K., L. Kopman, W. R. Pulley, and L. E. Trotter (2003). *On the Capacitated Vehicle Routing Problem*. Mathematical Programming 94(2-3): 343–359.

[34] Rego, C., & Rouncairol, C. (1996). *A parallel tabu search algorithm using ejection chains for the vehicle routing problem*. Meta-heuristics: Theory and Applications.

[35] Ropke, Stefan, and David Pisinger (2006). *A Unified Heuristic for a Large Class of Vehicle Routing Problems with Backhauls*. European Journal of Operational Research 171(3): 750–775.

[36] Toth Paolo and Daniele Vigo (2002). *The Vehicle Routing Problem*. SIAM.

[37] Toth Paolo and Daniele Vigo (2003). *The Granular Tabu Search and Its Application to the Vehicle-Routing Problem*. INFORMS Journal on Computing 15(4): 333–346.

[38] Yellow, P. (1970). *A Computational Modification to the Savings Method of Vehicle Scheduling*. Operational Research Quarterly *, 21*, 281-283.

[39] Vega José Marcos Moreno, María Belén Melián Batista, and José Andrés Moreno Pérez (2003). *Metaheuristicas: Una Visión Global*. Inteligencia Artificial: Revista Iberoamericana De Inteligencia Artificial 7(19): 7–28.

APPENDIX A

Objective Function	*Mu (μ)*																				
Lambda (*λ*)	**0.0**	**0.1**	**0.2**	**0.3**	**0.4**	**0.5**	**0.6**	**0.7**	**0.8**	**0.9**	**1.0**	**1.1**	**1.2**	**1.3**	**1.4**	**1.5**	**1.6**	**1.7**	**1.8**	**1.9**	**2.0**
D051-06c																					
0.0	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf
0.1	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	638.12	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf
0.2	661.57	653.43	653.34	Inf	Inf	Inf	Inf	Inf	643.20	618.45	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf
0.3	661.57	636.93	632.09	631.58	622.27	622.27	615.21	618.67	605.97	625.82	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf
0.4	632.09	632.09	622.79	622.79	615.72	615.72	621.50	605.97	613.33	613.33	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf
0.5	622.79	622.79	622.79	615.72	615.72	621.50	605.97	Inf	613.33	613.33	633.22	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf
0.6	615.72	615.72	615.72	615.72	621.50	619.18	605.97	613.33	613.33	613.33	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf
0.7	618.39	615.72	615.72	619.71	619.18	605.97	642.03	613.33	613.33	613.33	623.45	620.75	630.32	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf
0.8	618.39	618.39	619.71	619.71	605.97	605.97	642.03	630.54	619.43	624.52	621.79	618.34	616.41	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf
0.9	618.39	618.39	619.71	605.97	605.97	635.60	626.28	619.43	624.52	621.11	621.79	621.79	619.44	622.06	Inf	Inf	Inf	Inf	Inf	Inf	Inf
1.0	618.39	618.39	600.38	602.60	635.60	635.60	613.85	Inf	Inf	637.74	Inf	Inf	634.29	645.68	Inf	Inf	Inf	Inf	Inf	Inf	Inf
1.1	618.39	618.39	599.45	599.80	602.07	613.85	630.64	Inf	Inf	637.33	637.33	Inf	639.20	634.67	Inf	Inf	Inf	Inf	Inf	Inf	Inf
1.2	622.63	599.45	599.80	599.84	599.84	599.92	630.64	630.64	Inf	637.33	637.33	637.33	644.24	626.31	643.97	Inf	Inf	Inf	Inf	Inf	Inf
1.3	600.77	599.45	599.80	**595.31**	599.84	603.41	615.50	630.64	Inf	629.39	629.39	629.39	630.14	643.88	Inf	Inf	Inf	Inf	Inf	Inf	Inf
1.4	601.12	601.12	595.31	595.31	595.31	603.41	603.41	615.50	630.64	629.39	629.39	629.39	629.39	630.14	640.86	Inf	Inf	Inf	Inf	Inf	Inf
1.5	601.99	601.76	601.76	595.31	603.41	603.41	603.41	603.41	614.83	627.73	629.39	629.39	629.39	630.14	640.86	Inf	Inf	Inf	Inf	Inf	Inf
1.6	Inf	602.63	601.76	601.76	603.41	603.41	603.41	603.41	608.72	621.05	627.73	629.39	629.39	630.14	626.76	Inf	Inf	Inf	Inf	Inf	Inf
1.7	601.55	616.50	604.67	616.51	603.41	603.41	603.41	603.41	614.94	621.05	621.05	626.40	629.39	630.14	629.78	649.89	Inf	Inf	Inf	Inf	Inf
1.8	601.55	602.79	616.50	616.51	616.51	615.20	615.20	615.20	603.97	603.01	608.48	617.06	625.59	630.14	629.78	649.89	Inf	Inf	Inf	Inf	Inf
1.9	622.14	601.55	605.85	625.61	616.51	616.51	615.20	615.20	620.51	603.01	603.01	608.48	608.48	631.28	630.14	649.89	Inf	Inf	Inf	Inf	Inf
2.0	622.14	601.55	605.85	605.85	619.60	616.51	616.51	615.20	619.55	619.55	603.01	603.01	608.48	608.48	608.48	649.89	Inf	Inf	Inf	Inf	Inf

Figure 4.4 Parameterized of instance D056-06c, feasible to Initial Solution with of λ=1.3 and *μ=0.3* values.

APPENDIX B

Objective Function	*Mu (μ)*																				
Lambda (λ) D076-11c	**0.0**	**0.1**	**0.2**	**0.3**	**0.4**	**0.5**	**0.6**	**0.7**	**0.8**	**0.9**	**1.0**	**1.1**	**1.2**	**1.3**	**1.4**	**1.5**	**1.6**	**1.7**	**1.8**	**1.9**	**2.0**
0.0	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf
0.1	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf
0.2	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf
0.3	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf
0.4	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf
0.5	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf
0.6	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf
0.7	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf
0.8	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf
0.9	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf
1.0	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf
1.1	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf
1.2	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf
1.3	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf
1.4	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf
1.5	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf
1.6	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf
1.7	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf
1.8	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf
1.9	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf
2.0	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf

Figure 4.5 Parameterized of instance D076-11c, infeasible for the Initial Solution.

.

APPENDIX C

Objective Function	*Mu (μ)*																				
Lambda (*λ*) D101-09c	**0.0**	**0.1**	**0.2**	**0.3**	**0.4**	**0.5**	**0.6**	**0.7**	**0.8**	**0.9**	**1.0**	**1.1**	**1.2**	**1.3**	**1.4**	**1.5**	**1.6**	**1.7**	**1.8**	**1.9**	**2.0**
0.0	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf
0.1	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	1017.82	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf
0.2	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	1025.67	975.72	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf
0.3	Inf	Inf	1038.78	Inf	Inf	Inf	Inf	1008.99	971.11	975.76	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf
0.4	Inf	Inf	Inf	993.19	Inf	1024.0	1008.9	991.83	983.08	963.19	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf
0.5	Inf	Inf	Inf	Inf	Inf	1019.8	1000.2	962.67	970.96	974.67	Inf	Inf	987.57	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf
0.6	1014.82	984.47	1007.60	1019.23	1021.51	987.05	974.91	986.78	Inf	993.28	989.31	Inf	975.62	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf
0.7	970.97	Inf	Inf	990.48	987.56	983.00	986.23	976.67	983.47	974.17	981.05	988.22	Inf	992.56	Inf	Inf	Inf	Inf	Inf	Inf	Inf
0.8	Inf	Inf	997.13	985.86	Inf	986.23	986.78	976.67	983.47	Inf	1002.48	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf
0.9	Inf	Inf	Inf	981.22	967.98	968.53	981.24	981.24	981.13	977.59	1002.48	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf
1.0	973.94	Inf	975.62	971.10	967.98	968.53	973.26	973.26	975.67	974.99	968.77	997.43	979.32	978.41	Inf	Inf	Inf	Inf	Inf	Inf	Inf
1.1	969.64	975.62	971.10	982.05	965.95	973.26	Inf	Inf	Inf	980.37	988.99	997.43	973.12	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf
1.2	976.47	974.07	965.84	965.54	965.95	Inf	Inf	Inf	Inf	977.65	967.47	999.01	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf
1.3	976.47	969.73	969.55	970.16	969.86	Inf	Inf	Inf	972.33	977.65	972.50	999.01	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf
1.4	976.47	969.73	969.73	971.34	Inf	Inf	Inf	Inf	972.33	972.33	984.11	994.14	999.01	979.32	Inf	Inf	Inf	Inf	Inf	Inf	Inf
1.5	976.47	976.47	973.66	973.66	Inf	Inf	Inf	Inf	976.74	973.06	971.58	981.40	985.49	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf
1.6	992.36	992.36	973.66	973.66	947.67	Inf	Inf	972.95	976.74	968.64	965.88	971.66	985.49	972.44	Inf	Inf	Inf	Inf	Inf	Inf	Inf
1.7	992.36	977.50	956.27	942.70	944.10	947.67	Inf	966.65	976.74	968.64	959.62	971.66	981.38	966.54	Inf	Inf	Inf	Inf	Inf	Inf	Inf
1.8	967.97	956.27	956.27	953.48	944.10	944.10	Inf	973.15	976.74	976.74	968.66	975.74	977.30	Inf	Inf	Inf	974.95	Inf	Inf	Inf	Inf
1.9	952.75	952.75	952.75	953.48	953.48	945.97	952.26	968.60	976.74	976.74	959.62	971.66	981.38	Inf	Inf	Inf	983.47	Inf	Inf	Inf	Inf
2.0	952.72	952.75	952.75	Inf	953.48	956.02	948.69	985.68	984.52	983.78	958.93	984.53	977.30	Inf	Inf	Inf	Inf	Inf	1011.42	Inf	Inf

Figure 4.6 Parameterized of instance D101-09c, feasible to Initial Solution with of λ=1.3 and *μ=0.3* values.

APPENDIX D

Objective Function	*Mu (μ)*																				
Lambda (λ) D101-11c	0.0	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1.0	1.1	1.2	1.3	1.4	1.5	1.6	1.7	1.8	1.9	2.0
0.0	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf
0.1	1374.72	1202.7	1202.40	1144.7	1063.98	985.82	1028.99	1021.27	1020.58	981.78	952.31	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf
0.2	963.89	999.22	1012.41	1003.9	1004.69	1004.69	1004.00	982.78	964.64	956.40	939.51	986.88	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf
0.3	948.07	964.27	964.27	956.62	981.44	977.94	958.47	921.65	935.36	939.89	932.07	939.20	999.81	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf
0.4	930.03	955.0	943.77	963.07	959.42	948.29	941.03	932.24	912.55	901.73	907.22	914.85	985.57	1080.08	Inf	Inf	Inf	Inf	Inf	Inf	Inf
0.5	922.78	930.8	941.30	928.23	920.76	923.96	916.50	927.80	926.06	900.95	900.33	912.99	929.52	987.61	Inf	Inf	Inf	Inf	Inf	Inf	Inf
0.6	903.38	927.4	898.82	910.01	907.87	906.43	906.82	914.02	926.06	900.63	908.16	905.67	915.49	973.81	987.61	Inf	Inf	Inf	Inf	Inf	Inf
0.7	891.25	892.47	897.92	899.04	899.81	905.71	908.27	913.61	916.76	929.35	901.98	905.67	909.42	953.54	974.82	1004.58	1130.46	Inf	Inf	Inf	Inf
0.8	876.86	892.35	893.45	893.45	900.18	889.31	897.72	910.37	921.24	921.59	901.98	898.21	901.65	907.97	973.81	987.50	1038.44	1117.71	Inf	Inf	Inf
0.9	878.09	877.84	889.14	899.52	889.31	889.31	896.55	896.55	909.62	918.85	907.70	898.21	901.65	908.56	959.19	974.71	987.50	1058.23	1183.07	Inf	Inf
1.0	875.75	877.98	879.30	908.59	878.23	891.36	896.55	897.32	904.96	904.88	907.70	909.14	898.21	902.49	942.15	968.37	979.23	992.81	1068.35	1187.23	Inf
1.1	877.26	877.06	879.30	873.84	873.84	884.81	885.52	897.46	905.09	904.88	912.24	908.37	898.21	902.19	908.89	937.49	952.69	961.81	0.00	1090.63	1186.52
1.2	907.98	878.57	869.62	873.84	876.55	880.42	880.42	887.21	895.14	905.02	906.27	908.37	908.37	900.88	905.60	930.42	946.06	961.81	968.04	1024.50	1090.63
1.3	912.75	885.79	871.67	869.62	878.25	880.42	881.19	882.83	887.05	893.05	894.08	897.43	908.37	911.81	904.83	908.12	944.32	952.69	961.81	973.45	1050.10
1.4	915.89	907.17	880.33	880.33	871.64	876.15	876.92	882.83	882.66	887.05	886.78	896.18	896.18	897.43	901.16	910.85	936.72	946.06	952.69	961.81	985.01
1.5	910.44	909.01	907.17	880.33	884.60	871.64	876.66	878.55	882.66	878.55	888.48	894.96	894.96	896.18	900.88	904.27	931.38	940.57	951.91	961.81	968.04
1.6	910.44	910.44	907.94	907.33	884.60	885.37	874.05	878.30	878.55	878.55	882.39	890.58	894.96	894.96	899.62	903.01	907.56	933.74	945.29	949.36	961.04
1.7	910.44	910.44	910.44	907.33	886.44	885.37	887.01	874.05	878.30	878.55	888.48	890.58	890.58	894.96	898.41	898.02	909.59	926.67	945.29	949.36	958.48
1.8	910.44	910.44	909.17	909.17	907.33	885.37	887.01	887.01	874.55	877.63	882.39	890.58	890.58	890.58	894.02	898.95	907.98	906.30	933.74	945.29	949.36
1.9	910.44	910.44	909.17	909.17	909.94	908.10	887.01	887.01	887.01	874.55	887.72	890.58	890.58	892.61	892.61	896.05	901.40	904.69	933.74	945.29	949.36
2.0	911.83	909.17	909.17	909.17	909.94	910.51	909.74	887.01	887.01	887.01	882.49	891.68	892.61	892.61	892.61	896.05	893.64	907.98	930.44	932.79	945.29

Figure 4.7 Parameterized of instance D101-11c, feasible to Initial Solution with of λ=1.3 and *μ=0.3* values.

APPENDIX E

Objective Function										*Mu (μ)*											
Lambda (λ) D121-11c	**0.0**	**0.1**	**0.2**	**0.3**	**0.4**	**0.5**	**0.6**	**0.7**	**0.8**	**0.9**	**1.0**	**1.1**	**1.2**	**1.3**	**1.4**	**1.5**	**1.6**	**1.7**	**1.8**	**1.9**	**2.0**
0.0	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf
0.1	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf
0.2	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf
0.3	1664.08	1664.08	1662.71	0.00	0.00	0.00	0.00	1617.15	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf
0.4	1644.15	1645.06	1642.25	1600.93	1622.97	1609.26	1609.84	1612.81	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf
0.5	1596.04	1597.81	1599.23	1599.85	1587.79	1593.87	1590.42	0.00	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf
0.6	1598.38	1599.07	1584.20	1584.20	1592.75	1589.67	0.00	0.00	1619.79	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf
0.7	1598.27	1583.25	1583.77	1587.96	1585.86	1585.86	0.00	1673.25	1622.80	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf
0.8	1592.94	1594.84	1601.38	1591.49	1591.49	0.00	0.00	1616.76	1623.94	Inf	Inf	Inf	1649.31	Inf	1650.59	1664.70	Inf	Inf	Inf	Inf	Inf
0.9	1591.24	1596.72	1602.17	1602.17	1625.01	0.00	0.00	0.00	1622.79	Inf	Inf	Inf	Inf	Inf	Inf	1653.82	Inf	Inf	Inf	Inf	Inf
1.0	1596.72	1592.64	1591.02	1625.29	1617.35	1627.59	0.00	0.00	0.00	1651.27	1675.05	Inf	Inf	Inf	Inf	1649.59	1658.62	Inf	Inf	Inf	Inf
1.1	1593.78	1591.02	1591.26	1597.15	1619.49	1622.63	0.00	0.00	0.00	1639.33	0.00	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf
1.2	1593.09	1589.72	1617.77	1607.62	0.00	0.00	0.00	0.00	0.00	1629.95	1634.65	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf
1.3	Inf	Inf	1600.51	1604.16	1607.56	0.00	0.00	0.00	1632.53	1626.80	1643.87	1643.87	1630.22	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf
1.4	1629.33	1611.24	0.00	1605.05	1606.59	1610.05	0.00	0.00	1621.28	1625.30	1629.55	1643.87	1634.10	1628.02	1632.22	Inf	Inf	Inf	Inf	Inf	Inf
1.5	Inf	Inf	1618.82	0.00	1603.01	1605.30	1612.68	0.00	1617.23	1617.23	1625.98	1640.71	1630.68	1632.55	1631.11	1635.30	Inf	Inf	Inf	Inf	Inf
1.6	Inf	Inf	Inf	1623.26	0.00	1603.01	1607.93	1613.77	1620.42	1618.18	1621.49	1633.40	1637.53	1630.68	1631.11	1632.22	1636.87	1644.55	Inf	Inf	Inf
1.7	Inf	Inf	Inf	Inf	1624.33	0.00	1604.54	1607.26	1612.74	1617.94	1616.64	1625.97	1629.29	1622.30	1623.82	1628.11	1631.19	Inf	1640.00	1644.47	Inf
1.8	Inf	Inf	Inf	Inf	Inf	1624.33	1641.06	1610.23	1609.15	1612.74	0.00	1625.97	1625.97	1622.30	1620.74	1624.93	1628.01	1632.76	1638.02	1644.47	1647.71
1.9	Inf	Inf	Inf	Inf	Inf	Inf	Inf	1626.96	1612.12	1614.92	1612.91	1626.74	0.00	1617.80	1621.15	1623.82	1624.93	1629.58	1634.84	1640.00	1644.47
2.0	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	1626.96	1608.54	Inf	Inf	1626.74	1621.88	1620.33	1624.23	1624.93	1628.01	Inf	1634.84	1644.47

Figure 4.8 Parameterized of instance D121-11c, feasible to Initial Solution with of λ=0.6 and ***μ=0.3*** **values.**

APPENDIX F

Objective Function	***Mu (μ)***																				
Lambda (*λ*) D151-14c	**0.0**	**0.1**	**0.2**	**0.3**	**0.4**	**0.5**	**0.6**	**0.7**	**0.8**	**0.9**	**1.0**	**1.1**	**1.2**	**1.3**	**1.4**	**1.5**	**1.6**	**1.7**	**1.8**	**1.9**	**2.0**
0.0	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf
0.1	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf
0.2	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf
0.3	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf
0.4	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf
0.5	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf
0.6	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf
0.7	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf
0.8	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf
0.9	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf
1.0	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf
1.1	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf
1.2	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf
1.3	1222.06	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf
1.4	Inf	1223.51	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf
1.5	1234.45	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf
1.6	1236.24	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf
1.7	1243.73	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf
1.8	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf
1.9	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf
2.0	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf

Figure 4.9 Parameterized of instance D151-14c, feasible to Initial Solution with of λ=0.6 and *μ=0.0* values.

APPENDIX G

Objective Function	Mu (μ)																				
Lambda (λ)	0.0	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1.0	1.1	1.2	1.3	1.4	1.5	1.6	1.7	1.8	1.9	2.0
D200-18c																					
0.0	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf
0.1	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf
0.2	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf
0.3	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf
0.4	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf
0.5	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf
0.6	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf
0.7	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf
0.8	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf
0.9	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf
1.0	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf
1.1	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf
1.2	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf
1.3	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf
1.4	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf
1.5	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf
1.6	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf
1.7	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf
1.8	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf
1.9	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf
2.0	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf

Figure 4.10 Parameterized of instance D200-18c, infeasible for the Initial Solution.

APPENDIX H

Objective Function	*Mu (μ)*																				
Lambda (*λ*) E051-05e	**0.0**	**0.1**	**0.2**	**0.3**	**0.4**	**0.5**	**0.6**	**0.7**	**0.8**	**0.9**	**1.0**	**1.1**	**1.2**	**1.3**	**1.4**	**1.5**	**1.6**	**1.7**	**1.8**	**1.9**	**2.0**
0.0	1620.91	1604.67	1642.37	1606.6	1662.10	1572.56	1592.9	1572.46	1609.5	1586.65	Inf	1647.25	1719.70	1719.87	1731.40	1730.26	1660.40	1546.72	1575.41	1693.32	1607.51
0.1	904.76	904.76	976.74	874.77	858.38	775.69	681.94	655.89	649.20	635.25	600.52	0.00	Inf	1217.24	1241.23	1225.10	1249.33	1281.72	1281.74	1283.81	1308.43
0.2	721.57	707.06	702.51	681.11	661.07	629.13	629.13	622.21	606.47	Inf	589.90	628.56	Inf	1127.54	Inf	1146.13	1139.99	1151.07	1210.12	1210.12	Inf
0.3	641.24	628.22	628.91	628.91	617.25	601.07	581.45	581.45	596.26	Inf	589.90	580.17	694.00	Inf	Inf	Inf	Inf	1146.13	1145.38	1148.42	1148.42
0.4	616.91	616.91	620.80	Inf	594.14	581.45	581.45	588.89	Inf	581.42	589.90	585.08	628.56	696.46	Inf	Inf	1127.54	Inf	Inf	1137.60	1146.13
0.5	598.35	598.35	594.14	594.14	581.45	581.45	588.89	570.48	Inf	Inf	584.47	582.14	602.82	676.11	696.46	Inf	Inf	1127.54	Inf	Inf	Inf
0.6	Inf	Inf	Inf	Inf	581.45	575.58	588.89	570.48	Inf	Inf	584.47	582.14	580.17	628.56	Inf	712.74	Inf	Inf	Inf	1128.90	Inf
0.7	Inf	Inf	Inf	Inf	Inf	568.81	570.48	Inf	Inf	Inf	584.47	582.14	Inf	602.82	Inf	Inf	712.74	Inf	Inf	Inf	1128.90
0.8	Inf	Inf	Inf	Inf	568.81	568.81	Inf	Inf	Inf	563.90	583.12	582.14	Inf	0.00	628.56	Inf	710.19	Inf	Inf	Inf	Inf
0.9	Inf	Inf	Inf	568.83	568.81	Inf	Inf	Inf	Inf	583.12	583.12	Inf	Inf	592.92	604.11	652.26	Inf	710.19	Inf	Inf	Inf
1.0	Inf	Inf	577.09	568.83	Inf	Inf	Inf	Inf	Inf	572.93	587.35	Inf	Inf	592.92	Inf	628.56	676.11	Inf	710.19	Inf	Inf
1.1	Inf	Inf	577.09	568.83	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	615.86	652.26	Inf	Inf	Inf	Inf
1.2	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	592.92	Inf	628.56	Inf	Inf	710.19	Inf
1.3	577.09	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	577.38	Inf	618.62	638.94	Inf	Inf	710.19
1.4	578.69	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	582.09	Inf	628.56	Inf	Inf	Inf
1.5	579.56	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	577.38	603.48	628.22	638.94	Inf	Inf
1.6	587.08	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	577.38	582.09	603.48	638.08	675.28	Inf
1.7	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	582.09	603.48	628.22	Inf	675.28
1.8	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	577.38	582.09	628.22	Inf	Inf
1.9	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	577.38	582.09	592.03	Inf	Inf
2.0	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	577.38	582.09	592.03	Inf	Inf

Figure 4.11 Parameterized of instance E051-05e, feasible to Initial Solution with of λ=0.8 and μ=0.9 values.

APPENDIX I

Objective Function	***Mu (μ)***																				
Lambda *(λ)* E076-10e	**0.0**	**0.1**	**0.2**	**0.3**	**0.4**	**0.5**	**0.6**	**0.7**	**0.8**	**0.9**	**1.0**	**1.1**	**1.2**	**1.3**	**1.4**	**1.5**	**1.6**	**1.7**	**1.8**	**1.9**	**2.0**
0.0	2542.16	2505.74	2550.47	2540.90	2477.99	2469.94	2495.87	2475.59	2512.01	2447.71	Inf	2699.31	2719.04	2763.25	2846.19	2782.62	2906.93	2758.60	2918.89	2816.29	2786.94
0.1	1425.52	1354.71	1393.32	1258.74	1185.60	1149.76	1112.16	Inf	Inf	936.24	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	2220.83	2233.59
0.2	1101.94	1074.67	1075.53	1032.42	Inf	968.59	Inf	Inf	936.24	938.61	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf
0.3	993.52	Inf	984.98	984.98	Inf	946.36	943.68	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	1930.43
0.4	948.16	951.95	930.13	930.13	942.92	Inf	Inf	Inf	869.62	890.89	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf
0.5	Inf	Inf	Inf	Inf	Inf	897.34	888.01	869.62	905.37	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf
0.6	Inf	Inf	Inf	913.70	897.34	873.02	876.61	882.65	905.37	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf
0.7	Inf	Inf	Inf	886.76	876.11	876.61	892.84	900.21	924.24	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf
0.8	Inf	893.70	886.76	879.69	879.69	876.61	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf
0.9	896.46	890.35	869.52	869.52	869.52	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf
1.0	907.39	866.30	869.52	880.40	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf
1.1	866.32	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf
1.2	893.81	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf
1.3	888.65	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf
1.4	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf
1.5	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf
1.6	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf
1.7	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf
1.8	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf
1.9	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf
2.0	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf

Figure 4.12 Parameterized of instance E076-10c, feasible to Initial Solution with of λ=1.0 and μ=0.1 values.

APPENDIX J

Objective Function	Mu (μ)																				
Lambda (λ) E101-08e	0.0	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1.0	1.1	1.2	1.3	1.4	1.5	1.6	1.7	1.8	1.9	2.0
0.0	3556.88	3383.36	3306.07	3454.43	3287.51	3452.43	3231.18	3414.31	3431.07	3437.59	3367.60	3953.42	3846.25	3765.15	3890.33	3830.42	3774.57	3746.35	3709.05	3764.70	3755.97
0.1	1508.80	1471.53	1452.14	1450.40	1355.80	1172.06	1169.84	1113.65	1001.50	988.44	944.76	1706.07	2090.99	2166.41	2194.03	2310.58	2353.06	2434.08	2500.94	2546.28	2548.82
0.2	1186.96	1134.74	1087.59	1080.18	1081.47	1023.99	987.15	974.54	985.38	946.04	908.03	972.83	Inf	1966.52	2051.03	2092.69	2148.57	2187.46	2198.94	2279.97	2282.57
0.3	1034.13	991.59	1015.43	982.00	935.04	978.85	978.85	981.39	952.57	918.81	923.38	945.53	980.69	1712.64	2011.44	2014.58	2009.88	2047.90	2210.12	2220.20	2211.93
0.4	991.87	930.26	935.04	935.04	912.03	931.33	919.72	948.72	939.90	917.22	900.38	934.25	989.65	1058.82	1675.59	1907.73	1973.28	1975.20	2000.09	2007.49	2080.29
0.5	957.94	957.94	954.54	963.56	921.97	908.94	941.34	920.28	911.10	918.01	905.97	937.21	950.74	969.21	Inf	Inf	Inf	1951.67	1984.67	1974.09	2000.09
0.6	940.14	936.12	949.46	924.66	908.94	927.99	912.56	883.68	880.23	928.36	899.12	916.52	927.06	973.27	995.29	Inf	Inf	1917.46	1970.87	1957.50	1981.42
0.7	944.56	918.84	912.32	902.27	889.93	887.27	903.10	892.35	888.64	900.10	914.51	916.52	930.53	936.81	976.83	1064.20	Inf	1675.59	1915.47	1907.21	1958.89
0.8	912.33	909.34	894.15	889.93	896.13	903.89	883.37	880.23	889.10	905.14	927.10	914.21	924.75	934.52	975.88	966.52	1060.62	Inf	Inf	1929.06	1893.51
0.9	903.49	902.09	902.09	905.57	899.68	878.86	880.23	894.90	892.68	908.68	927.10	914.21	922.79	920.43	941.56	976.67	1003.20	1069.04	1200.59	1675.59	1920.06
1.0	889.00	889.00	898.31	888.73	888.66	896.75	886.15	889.60	892.67	908.61	914.57	927.10	922.56	918.47	925.31	952.40	971.42	1031.39	Inf	Inf	Inf
1.1	886.95	878.87	885.21	890.59	891.14	885.21	877.25	889.60	910.92	912.30	926.01	922.87	914.03	912.05	920.43	951.98	970.41	954.42	1054.21	Inf	Inf
1.2	883.60	876.98	887.63	890.59	891.14	877.54	885.21	904.01	910.92	908.55	922.28	921.94	922.87	914.66	931.68	933.20	959.84	954.14	1003.20	1060.01	Inf
1.3	879.37	895.77	885.75	888.98	891.94	877.54	883.38	886.81	904.98	897.28	909.48	921.94	927.89	912.05	919.51	922.45	951.06	965.24	966.52	1013.33	1068.43
1.4	886.41	891.90	891.90	880.05	877.65	877.54	883.38	889.37	897.28	904.98	903.95	913.57	921.50	918.88	915.90	919.51	937.59	949.79	951.92	956.45	1054.21
1.5	890.11	887.55	880.05	880.05	865.60	886.16	889.61	895.60	897.28	897.28	909.48	913.57	903.95	904.42	912.09	0.00	937.59	943.62	964.84	956.25	Inf
1.6	890.11	890.11	880.05	865.60	886.95	892.29	892.13	897.28	897.28	897.28	909.48	903.95	903.95	905.54	912.99	919.51	Inf	933.20	949.79	958.87	965.60
1.7	892.13	887.99	879.26	881.74	883.98	892.29	893.04	905.63	899.69	897.28	909.48	903.95	903.95	903.95	912.99	910.29	923.45	936.52	950.52	956.83	956.25
1.8	896.95	885.76	879.03	884.00	883.98	883.15	893.04	905.63	901.15	901.88	909.48	903.95	903.95	903.95	905.76	912.99	926.29	934.26	940.91	956.83	957.96
1.9	898.77	893.68	893.68	875.53	875.53	883.15	890.20	903.83	903.83	901.15	909.48	903.95	903.95	903.95	905.18	905.18	926.29	935.66	940.11	954.17	954.17
2.0	891.97	893.68	893.68	878.39	875.53	875.53	893.25	904.00	903.83	903.83	904.70	902.60	902.60	903.95	905.18	905.18	917.47	922.35	939.40	953.98	954.17

Figure 4.13 Parameterized of instance E101-08e, feasible to Initial Solution with of λ=1.6 and μ=0.3 values.

APPENDIX K

Objective Function	Mu (μ)																				
Lambda (λ) E101-10c	0.0	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1.0	1.1	1.2	1.3	1.4	1.5	1.6	1.7	1.8	1.9	2.0
0.0	3,245.98	3,403.66	3,124.42	3,450.46	3,301.50	3,364.55	3,151.64	3,260.53	3,203.55	3,373.51	3,960.29	4,575.87	4,632.95	4,417.53	4,458.22	4,391.32	4,416.52	4,506.32	4,592.09	4,570.22	4,609.51
0.1	1,229.82	1,200.55	1,305.58	1,267.80	1,197.50	1,130.89	1,129.79	1,071.93	987.28	942.01	922.09	1,459.47	2,858.38	3,018.77	3,066.34	3,196.11	3,206.00	3,371.37	3,309.44	3,338.70	3,475.80
0.2	943.68	954.55	983.48	984.28	984.28	984.28	968.79	956.79	916.39	900.68	898.02	947.82	1,467.85	2,884.56	2,854.06	2,862.81	3,019.22	2,923.62	3,076.24	3,107.15	3,202.13
0.3	934.67	932.54	941.53	928.90	931.27	932.26	917.90	902.34	908.80	895.75	899.88	913.87	945.72	1,459.87	2,707.18	2,759.60	2,853.82	2,821.25	2,981.51	2,953.07	2,965.24
0.4	901.56	907.19	910.03	908.86	916.46	906.40	899.14	896.36	892.30	897.27	904.58	878.29	936.65	1,025.97	Inf	2,660.28	2,854.59	2,788.82	2,842.07	2,821.21	2,862.78
0.5	874.53	887.66	888.12	894.66	892.09	895.28	890.60	890.60	893.31	899.55	890.04	874.85	913.96	937.36	1,040.73	Inf	2,524.42	2,722.84	2,724.14	2,871.57	2,844.77
0.6	842.73	859.86	860.19	877.69	875.26	873.81	883.10	884.80	893.31	899.69	892.21	889.35	895.10	931.52	937.36	1,073.87	Inf	2,395.29	2,699.38	2,808.70	2,729.30
0.7	840.13	842.73	861.73	862.13	861.18	873.39	873.39	883.43	886.58	899.69	899.08	889.35	889.03	922.79	930.22	950.75	1,063.27	Inf	2,321.14	2,659.91	2,697.99
0.8	835.75	841.23	843.71	843.71	863.99	852.40	859.08	874.79	874.06	874.83	886.63	875.85	885.30	888.72	932.67	936.91	984.11	1,051.13	Inf	Inf	2,549.47
0.9	836.40	836.73	836.73	846.36	839.57	839.57	847.49	855.46	865.67	874.40	862.43	862.75	866.19	868.58	928.24	903.37	927.25	1,010.72	1,055.08	Inf	2,269.17
1.0	833.51	836.29	837.60	844.84	837.11	841.62	842.63	848.26	852.12	860.44	856.12	862.75	862.75	862.52	892.78	898.47	917.03	923.33	1,013.00	1,058.86	Inf
1.1	830.25	834.82	833.59	832.14	832.73	840.84	841.56	843.54	852.26	852.47	857.51	857.39	856.44	866.73	868.92	902.16	905.03	914.16	937.80	1,033.51	1,126.91
1.2	833.13	831.56	827.38	828.13	**826.00**	836.45	836.45	843.24	847.40	852.04	851.18	853.28	857.39	865.42	859.32	893.83	899.85	914.16	912.05	970.18	1,033.51
1.3	837.89	839.10	828.49	827.38	827.01	827.01	836.29	838.86	843.08	845.30	851.18	853.28	853.28	856.72	860.27	868.15	900.77	905.03	914.16	919.32	994.56
1.4	844.02	834.49	833.64	826.06	826.00	827.01	827.78	830.00	842.41	843.08	845.93	853.28	853.28	853.28	856.59	866.29	895.54	893.54	898.72	907.85	930.89
1.5	838.56	838.56	834.49	833.64	835.71	827.07	827.53	829.42	829.42	835.89	851.36	854.12	852.06	853.28	856.72	859.71	879.98	900.63	899.67	907.85	912.05
1.6	838.56	838.56	835.27	835.51	835.76	836.48	829.47	829.16	829.42	829.42	835.93	853.45	857.84	852.06	856.72	860.37	863.00	894.46	893.93	899.67	908.80
1.7	838.56	838.56	838.56	836.25	837.23	836.16	837.80	829.16	828.85	829.42	835.55	844.12	844.12	857.84	861.28	861.82	866.29	887.39	893.23	899.67	908.80
1.8	838.56	838.56	839.21	839.21	836.25	836.16	837.80	837.80	829.16	828.85	829.46	837.65	841.75	844.12	856.90	861.82	864.68	863.00	894.86	893.23	899.67
1.9	838.56	838.56	839.21	839.21	839.98	837.02	837.80	837.80	837.80	829.16	835.46	837.65	837.65	843.78	846.15	857.17	865.20	861.39	894.86	893.23	899.67
2.0	839.96	839.21	839.21	839.21	839.98	839.43	838.65	837.80	837.80	837.80	836.04	839.43	839.68	839.68	843.20	849.59	850.33	864.68	879.04	893.91	893.23

Figure 4.14 Parameterized of instance E101-10c, feasible to Initial Solution with of λ=1.2 and μ=0.4 values.

APPENDIX L

Objective Function	*Mu (μ)*																				
Lambda (λ) E121-07c	**0.0**	**0.1**	**0.2**	**0.3**	**0.4**	**0.5**	**0.6**	**0.7**	**0.8**	**0.9**	**1.0**	**1.1**	**1.2**	**1.3**	**1.4**	**1.5**	**1.6**	**1.7**	**1.8**	**1.9**	**2.0**
0.0	3330.20	3270.38	3382.74	3103.71	3094.56	3193.40	3081.21	3105.71	3416.05	3120.27	Inf	8680.73	8649.37	8744.09	8735.59	8670.23	8729.13	8817.41	8799.49	8744.84	8743.11
0.1	1577.77	1573.07	1576.15	1571.07	1524.43	1309.69	1305.38	1287.26	1255.36	1177.45	Inf	5385.67	7611.63	7773.57	7741.99	7910.72	7818.09	7822.12	7849.26	7934.44	7934.54
0.2	1265.48	1268.22	1269.49	1209.45	1202.30	1190.52	1193.10	1176.97	1165.78	1149.83	1136.77	1168.11	5156.52	7146.88	7671.78	7711.52	7644.19	7704.30	7716.91	7788.67	7790.34
0.3	1159.56	1173.84	1172.13	1172.52	1172.52	1175.98	1166.30	1154.43	1144.89	1151.07	1135.44	Inf	1182.85	5344.21	7292.57	7268.20	7591.72	7650.75	Inf	7461.93	7478.49
0.4	1151.02	1136.16	1140.78	1131.83	1158.27	1153.71	1146.79	1137.92	1144.88	Inf	1133.08	Inf	1162.60	Inf	5436.22	6806.24	7098.21	7280.87	7564.36	7677.15	7717.81
0.5	1136.19	1126.82	1134.67	1130.55	1129.05	1124.43	Inf	Inf	Inf	Inf	1134.41	1145.98	Inf	1162.02	Inf	5359.94	6916.59	7172.87	7259.15	7426.28	7489.10
0.6	1105.02	1118.92	1128.26	1127.59	1127.68	Inf	Inf	Inf	1122.94	1128.03	1131.11	1144.00	Inf	1162.60	Inf	1283.83	5431.77	7036.50	7028.51	7161.32	7398.48
0.7	1101.71	1101.41	1105.91	1130.81	Inf	Inf	Inf	1112.58	1113.96	1131.11	1136.96	1144.00	Inf	Inf	1163.52	Inf	1292.69	5405.25	6793.84	Inf	7192.22
0.8	1094.87	1098.53	1114.63	1106.84	1100.16	1129.06	1128.59	Inf	Inf	1121.62	1125.16	1133.07	Inf	Inf	1164.71	1176.63	Inf	1297.00	5415.99	6993.64	6929.98
0.9	1075.42	1082.00	1091.46	1106.08	1108.20	1102.01	1102.01	Inf	Inf	1111.81	1110.82	1133.07	1137.54	Inf	Inf	1162.47	Inf	Inf	1296.97	5431.77	6826.03
1.0	1071.07	1070.08	1071.51	1093.55	1102.02	Inf	Inf	Inf	Inf	1109.20	1109.98	1121.91	1126.38	1132.55	1130.40	1157.65	1152.25	Inf	Inf	1305.28	5338.79
1.1	Inf	1068.14	1075.71	1073.38	1081.14	Inf	Inf	Inf	1115.74	1115.74	1111.33	1113.99	1127.93	1127.92	1130.41	1147.82	1149.31	1176.89	Inf	1273.08	Inf
1.2	Inf	Inf	1070.68	1072.63	1079.81	1079.12	Inf	Inf	1109.00	1115.74	1117.87	1123.01	1133.27	1127.92	1132.55	1149.26	1160.06	1162.72	Inf	Inf	1283.12
1.3	Inf	Inf	Inf	1066.43	1075.94	1079.12	1079.01	1082.92	1106.68	1107.60	1111.14	1123.01	1132.29	1141.29	1147.46	1138.92	1156.33	1160.06	1163.55	1155.84	Inf
1.4	Inf	Inf	Inf	Inf	1066.59	1071.52	1079.01	1079.01	1082.43	1094.49	1119.68	1130.36	1132.29	1140.98	1147.15	1143.13	1151.84	1163.71	1157.82	1176.89	1154.57
1.5	Inf	Inf	Inf	Inf	Inf	1065.08	1071.88	1078.82	1079.01	1082.43	1102.21	1130.36	1125.55	1143.26	1142.52	1142.82	1142.82	1163.71	1163.71	1173.39	1155.84
1.6	Inf	Inf	Inf	Inf	Inf	Inf	1065.08	1072.62	1079.55	1079.74	1081.68	1106.70	1125.55	1125.55	1144.16	1144.90	1142.82	1158.68	1165.11	1161.47	1175.50
1.7	Inf	Inf	Inf	Inf	Inf	Inf	Inf	1070.13	1076.72	1079.55	1081.11	1091.45	1123.33	1117.17	1145.38	1148.79	1144.46	1129.29	1151.68	1165.11	1165.85
1.8	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	1074.02	1076.72	1082.86	1091.45	1113.42	1106.30	1124.91	1148.79	1144.46	1144.46	1158.06	1152.36	1162.86
1.9	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	1074.02	1078.09	1087.31	1112.99	1113.42	1107.51	1123.68	1131.91	1144.46	1153.61	1160.03	1152.36
2.0	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	1075.95	1082.89	1101.03	1112.99	1111.44	1111.87	1128.32	1123.86	1129.77	1158.06	1160.03

Figure 4.15 Parameterized of instance E121-07c, feasible to Initial Solution with of λ=1.6 and μ=0.6 values.

APPENDIX M

Objective Function	*Mu (μ)*																				
Lambda (λ) E151-12c	0.0	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1.0	1.1	1.2	1.3	1.4	1.5	1.6	1.7	1.8	1.9	2.0
0.0	4665.66	4700.65	4578.43	4615.78	4786.26	4570.62	4573.87	4580.11	4606.84	4590.66	Inf	5357.08	5234.77	5164.89	5371.69	5210.16	5455.73	5489.95	5463.01	5371.75	5505.74
0.1	1730.97	1719.29	1685.62	1687.31	1681.66	1646.24	1540.43	1438.29	1314.84	1302.08	1164.39	Inf	3078.97	3186.95	3216.84	3243.40	3390.92	3458.54	3485.45	3557.05	3516.11
0.2	1530.67	1481.62	1441.01	1385.65	1407.76	1309.87	1286.54	1254.21	1238.95	1253.17	1162.49	1235.58	Inf	3012.92	3077.89	3098.51	3221.77	3281.35	3313.47	3385.04	3424.37
0.3	1316.66	1312.76	1283.57	1341.11	1331.34	1272.63	1231.26	1246.46	1212.14	1217.86	1153.58	1168.70	1314.79	Inf	2884.59	2981.24	2976.55	3029.42	3084.80	3277.80	3282.84
0.4	1261.05	1224.83	1252.20	1256.68	1198.55	1206.04	1234.97	1154.09	1200.59	1213.41	1155.31	1162.69	1226.29	1384.10	Inf	2962.88	3004.24	3022.16	3021.26	3014.75	3091.34
0.5	1187.36	1180.20	1181.64	1209.69	1214.43	1184.81	1151.66	1163.16	1159.27	1151.79	1147.14	1169.96	1154.48	1313.28	Inf	Inf	Inf	2921.90	2927.46	3000.71	3046.87
0.6	1162.07	1191.99	1200.90	1186.40	1179.55	1162.87	1160.54	1166.25	1165.85	1151.79	1129.06	1136.20	1158.07	1208.17	1296.10	0.00	Inf	Inf	Inf	3025.74	2926.17
0.7	1159.77	1161.37	1193.12	1165.28	1162.48	1155.65	1121.79	1153.96	1134.17	1125.93	1133.25	1153.32	1157.94	1187.51	1257.55	1308.03	Inf	Inf	Inf	Inf	2933.91
0.8	1147.66	1187.39	1151.95	1158.05	1165.41	1147.27	1125.98	1123.26	1108.10	1120.00	1127.74	1130.34	1158.45	1152.95	Inf	1300.65	1381.12	Inf	Inf	2881.24	2924.30
0.9	1157.43	1147.10	1129.35	1153.16	1153.66	1158.58	1135.65	1136.32	1116.96	1117.08	1117.08	1121.24	1146.66	1155.44	Inf	1249.75	1289.77	1381.13	1493.76	Inf	Inf
1.0	1133.43	1128.57	1138.42	1138.42	1131.04	1158.58	1143.69	1146.72	1127.54	1126.93	1117.69	1117.69	1145.03	1166.59	1175.58	0.00	1288.89	1303.18	0.00	Inf	Inf
1.1	1115.67	1116.33	1131.88	1126.95	1129.02	1131.99	1129.52	1124.71	1142.46	1123.72	1126.93	1141.51	1141.77	1165.40	1175.58	0.00	Inf	1309.78	1341.32	Inf	Inf
1.2	1123.02	1110.32	1127.36	1122.73	1122.73	1129.02	1103.85	1143.83	1141.73	1153.11	Inf	1140.45	1145.98	1169.24	1186.32	1185.63	Inf	1313.54	1298.91	1365.32	Inf
1.3	1132.48	1119.04	1109.72	1121.92	1111.38	1111.93	1117.48	1131.75	1141.49	1153.08	1148.31	1155.08	1171.96	1157.15	1176.78	1182.92	Inf	1234.78	1313.54	1322.60	1370.81
1.4	1133.68	1113.15	1108.43	1121.92	1121.92	1111.11	1137.32	1132.93	1129.27	1140.96	1146.78	1148.46	1166.48	1182.82	1183.37	1183.04	1191.56	0.00	1264.99	1304.20	1322.60
1.5	1133.59	1120.19	1113.60	1122.48	1112.67	1116.79	1131.39	1124.72	1138.74	1122.85	1148.14	1146.78	1144.27	1181.37	1187.21	1196.74	1178.35	0.00	1234.78	1313.27	1301.41
1.6	1114.53	1131.11	1119.90	1113.60	1119.18	1128.36	1131.67	1118.40	1115.78	1115.95	1126.82	1148.14	1145.66	1159.14	Inf	1184.55	1192.77	0.00	1240.14	1254.74	1304.33
1.7	1147.13	1134.13	1128.06	1120.15	1113.58	1121.46	1131.72	1131.16	1106.08	1108.39	1128.89	1142.16	1146.78	1156.94	Inf	1189.54	1186.10	1201.42	Inf	1234.78	1313.27
1.8	1147.70	1134.78	1136.22	1124.18	1128.36	1117.47	1108.65	1130.32	1123.72	1119.58	1121.33	1142.16	1157.74	1139.17	1171.61	1187.05	1187.99	1190.76	Inf	1240.14	1260.21
1.9	1135.36	1134.78	1135.03	1132.99	1127.76	1131.30	1102.34	1107.37	1116.47	1111.33	1105.25	1121.89	1159.01	Inf	1182.47	1164.96	1185.77	1182.90	1193.51	1233.12	1259.82
2.0	1134.02	1135.36	1124.00	1118.07	1130.78	1135.42	1129.88	1101.82	1107.93	1125.98	1108.33	1105.25	1148.19	1142.90	1158.41	1187.57	1186.54	1181.80	Inf	Inf	Inf

Figure 4.16 Parameterized of instance E151-12c, feasible to Initial Solution with of λ=2.0 and μ=0.7 values.

APPENDIX N

Objective Function	Mu (μ)																				
Lambda (λ) E200-17c	0.0	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1.0	1.1	1.2	1.3	1.4	1.5	1.6	1.7	1.8	1.9	2.0
0.0	5919.27	6123.63	5924.46	5981.57	6029.12	5774.48	5703.22	5936.09	6012.48	6009.96	6327.47	7011.19	6935.79	7210.91	7156.24	7296.26	7003.84	6932.02	6852.16	6885.50	7123.85
0.1	2273.58	2183.15	2139.70	2022.76	1968.46	1938.49	1896.37	1824.56	1640.37	1632.92	1425.38	Inf	3971.71	4058.22	4247.70	4263.70	4389.17	4414.07	4477.95	4533.42	4511.75
0.2	1836.47	1815.39	1811.67	1745.49	1680.98	1602.10	1637.83	1652.00	1568.10	1530.84	1442.51	1550.24	Inf	3915.03	3921.44	3935.32	4083.85	4074.97	4228.37	4229.75	4238.49
0.3	1636.64	1565.51	1639.80	1629.76	1533.23	1534.73	1571.83	1555.81	1521.39	1477.96	1440.54	1499.65	1596.67	Inf	3868.04	Inf	3908.58	3926.88	3996.12	4006.57	4094.61
0.4	1575.41	1593.51	1547.39	1526.61	1576.63	1560.43	1534.89	1483.15	1496.56	1445.27	1420.02	1486.67	1528.51	1639.00	Inf	3772.38	3850.14	3884.76	3900.25	Inf	3955.02
0.5	1549.84	1474.26	1481.85	1469.65	1481.72	1499.79	1499.62	1458.03	1462.94	1445.16	1413.94	1455.01	1512.29	1587.58	Inf	Inf	3772.69	Inf	Inf	3946.57	3940.64
0.6	1471.78	1458.75	1448.40	1455.10	1484.81	1459.87	1461.76	1466.13	1467.85	1442.92	1415.22	1454.56	1475.29	1563.06	1616.39	1703.41	Inf	3659.02	3772.04	Inf	3922.29
0.7	1421.85	1442.42	1469.76	1468.94	1438.16	1449.55	1460.99	1437.93	1474.60	1469.66	1416.75	1451.89	1482.90	1515.50	1580.13	1592.13	Inf	Inf	3721.64	3716.64	3782.93
0.8	1420.85	1436.18	1444.47	1418.76	1423.11	1427.10	1447.38	1451.69	1457.16	1447.06	1411.13	1447.78	1474.71	1502.30	1559.91	1564.80	1634.47	Inf	Inf	Inf	Inf
0.9	1429.93	1433.88	1433.23	1420.47	1433.40	1441.08	1436.57	1440.13	1424.87	1426.13	1428.22	1434.99	1448.37	1485.71	1542.97	1557.30	1590.74	1639.63	Inf	Inf	Inf
1.0	1395.74	1409.47	1423.67	1429.29	1444.36	1432.33	1429.45	1435.45	1421.45	1418.91	1415.75	1433.99	1448.46	1493.66	1540.22	1547.41	1544.34	1591.62	Inf	Inf	Inf
1.1	1381.03	1379.05	1391.97	1408.53	1412.83	1427.34	1428.74	1409.39	1428.73	1432.75	1405.01	1445.11	1443.50	1436.48	1511.20	1544.80	1556.43	1588.85	1616.63	Inf	Inf
1.2	1392.50	1381.03	1371.57	1403.95	1408.69	1412.30	1418.99	1416.03	1427.48	1434.80	1439.02	1443.33	1467.37	1444.25	1484.43	1543.85	1550.11	1599.81	1602.19	1627.10	1704.95
1.3	1402.11	1376.30	1389.74	1389.55	1386.08	1393.34	1398.52	1403.14	1426.04	1420.26	1409.43	1455.56	1452.78	1462.03	1485.44	1516.68	1531.93	1552.10	1540.39	1564.01	1638.20
1.4	1386.87	1384.23	1370.05	1380.02	1383.78	1399.37	1415.33	1402.60	1396.85	1409.54	1427.80	1449.37	1449.19	1464.64	Inf	1497.01	1521.83	1539.30	1540.09	1567.64	1560.01
1.5	1387.76	1381.78	1405.86	1393.58	1406.70	1407.35	1400.30	1404.56	1405.88	1397.24	1398.18	1449.37	1449.37	1457.35	1460.28	1478.20	1532.47	1550.48	1531.27	1559.86	1590.97
1.6	1408.80	1387.22	1390.98	1391.57	1408.20	1407.19	1394.95	1392.25	1402.63	1396.28	1387.51	1424.07	1449.45	1455.40	1448.88	1471.65	1519.17	1521.83	1551.66	1550.73	1559.65
1.7	1400.04	1386.00	1379.19	Inf	1383.77	1405.21	1394.66	1401.74	1398.82	1395.96	1400.82	1422.01	1423.80	1441.11	1475.18	1455.27	1475.12	1537.17	1557.64	1541.90	1591.22
1.8	1412.36	1396.49	1387.85	1379.19	Inf	1404.04	1390.27	1398.45	1394.59	1390.75	1389.57	1421.47	1425.76	1436.08	1440.84	1452.04	1461.05	1510.04	1526.98	1551.67	1539.97
1.9	1404.23	1405.59	1395.68	1388.13	1380.42	1387.55	1389.14	1392.31	1400.45	1385.50	1383.79	1395.01	1418.48	1431.45	1442.01	1456.56	1457.74	1529.66	1548.46	1553.43	1533.85
2.0	1424.31	1401.12	1405.54	1377.89	1381.91	1382.81	1389.59	1390.14	1376.03	1395.05	1388.63	1435.30	1411.18	1416.45	1443.96	1452.84	1450.11	1471.47	1537.95	1536.88	1533.74

Figure 4.17 Parameterized of instance E200-17c, feasible to Initial Solution with of λ=1.4 and μ=0.2 values.

APPENDIX Ñ

Objective Function	*Mu (μ)*																				
Lambda *(λ)* D201-05k	0.0	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1.0	1.1	1.2	1.3	1.4	1.5	1.6	1.7	1.8	1.9	2.0
0.0	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf
0.1	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	8683.68	7400.31	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf
0.2	Inf	Inf	Inf	Inf	Inf	Inf	8617.23	7542.61	7384.99	7200.15	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf
0.3	Inf	Inf	Inf	Inf	8632.01	7813.13	7352.07	7384.99	7015.20	7705.34	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf
0.4	Inf	Inf	Inf	7940.09	7462.61	7361.22	7244.67	6874.99	7252.69	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf
0.5	Inf	7968.37	8159.75	7640.60	7440.51	7128.63	7153.12	7000.47	7723.65	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf
0.6	7964.51	7661.57	7610.46	7291.14	7336.06	7051.51	6776.90	7172.45	7915.10	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf
0.7	7650.34	7504.03	7291.14	7250.29	7056.97	7062.33	6940.06	7172.45	7916.23	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf
0.8	7452.19	7392.48	7367.20	7144.46	6954.14	7166.72	7270.54	7885.80	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf
0.9	7217.01	7161.72	7160.40	7043.70	6977.62	7187.51	7202.77	7916.23	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf
1.0	7296.89	7402.35	6954.14	6990.20	6975.44	7221.08	7970.74	7913.67	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf
1.1	7218.03	7112.11	7209.23	7180.48	7172.45	7221.08	7858.79	7886.21	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf
1.2	6913.05	7129.17	7081.19	7231.89	7172.45	7235.18	7904.06	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf
1.3	6928.61	6919.04	7075.79	Inf	7234.61	7919.30	8015.60	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf
1.4	7139.43	6960.01	7085.29	6702.49	Inf	7862.23	7920.61	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf
1.5	7386.02	7118.33	6949.72	6936.08	6984.53	7483.00	7962.24	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf
1.6	7223.21	7013.68	7316.42	7023.02	6936.08	6733.99	7504.07	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf
1.7	Inf	Inf	7394.04	7013.68	6722.10	**6691.04**	6871.27	7521.65	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf
1.8	Inf	Inf	Inf	7403.43	7428.00	7316.42	Inf	6715.07	7539.51	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf
1.9	Inf	Inf	Inf	Inf	6958.05	6954.60	7375.89	6937.21	6700.08	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf
2.0	Inf	7000.13	7440.68	Inf	7013.68	7310.37	7310.37	6942.22	6700.08	6729.73	7584.78	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf

Figure 4.18 Parameterized of instance E201-05k, feasible to Initial Solution with of λ=1.7 and μ=0.5 values.

APPENDIX O

Objective Function	*Mu (μ)*																				
Lambda (λ) D241-10k	0.0	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1.0	1.1	1.2	1.3	1.4	1.5	1.6	1.7	1.8	1.9	2.0
0.0	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf
0.1	5837.15	5842.42	5825.10	5820.39	5882.23	5830.30	5830.30	5876.76	5882.23	5932.09	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf
0.2	6066.95	5901.97	6360.34	5864.38	5818.97	5881.18	5807.07	5830.30	5865.40	5910.40	5924.09	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf
0.3	5853.02	6056.42	6236.34	6182.32	6022.29	6109.04	5983.35	5953.00	5981.13	Inf	5877.20	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf
0.4	6159.01	6163.43	Inf	5955.12	5967.86	5926.87	5903.68	5957.89	Inf	6093.81	6101.44	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf
0.5	5989.53	6022.78	5995.55	5957.39	5972.05	Inf	5992.22	5987.83	5939.19	5954.88	6008.01	6142.05	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf
0.6	6058.96	6108.32	5981.41	6116.49	5951.01	5973.41	5966.60	6098.43	Inf	Inf	6009.85	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf
0.7	6084.04	6084.04	5941.45	6085.67	6078.39	5976.79	5918.87	5950.63	5975.19	5941.67	6100.60	6245.84	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf
0.8	6057.31	6081.25	6109.31	5953.00	5977.12	6087.50	6121.09	6086.80	5982.73	6125.77	6091.55	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf
0.9	5952.77	5972.05	5981.04	5973.41	6087.64	5939.99	6087.35	6093.68	6078.11	5941.93	5991.51	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf
1.0	6027.56	6075.34	Inf	5952.77	6083.69	6100.60	6129.32	5976.79	6001.33	6100.85	6100.85	Inf	6164.3	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf
1.1	6079.94	6000.45	5967.21	Inf	5981.41	6125.77	5980.67	6096.65	5997.05	6125.77	5991.51	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf
1.2	Inf	6089.51	5972.17	Inf	5976.65	6089.73	6149.18	6087.61	6072.76	6144.50	6140.56	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf
1.3	Inf	6121.31	6125.84	6104.21	6001.31	6090.74	6144.50	6068.81	6129.72	5954.88	6125.77	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf
1.4	Inf	Inf	5979.55	5955.78	6172.76	5931.82	5990.94	5982.73	Inf	6125.77	5941.93	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf
1.5	Inf	Inf	Inf	6084.92	6084.89	5964.85	6145.23	6069.06	5950.46	6149.18	5995.93	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf
1.6	Inf	Inf	Inf	Inf	5961.91	6171.88	5948.09	5986.25	6016.03	6100.85	6073.01	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf
1.7	Inf	Inf	Inf	Inf	Inf	5960.32	6001.00	6000.46	5991.51	6124.07	6091.55	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf
1.8	Inf	Inf	Inf	Inf	Inf	Inf	6026.28	6084.15	5983.10	5964.48	6004.77	5991.51	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf
1.9	Inf	Inf	Inf	Inf	Inf	Inf	Inf	6033.59	6016.50	6005.74	6002.92	6002.92	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf
2.0	Inf	Inf	Inf	Inf	Inf	Inf	Inf	6026.21	5972.78	6001.04	6042.36	5965.12	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf

Figure 4.19: Parameterized of instance D241-10k, feasible to Initial Solution with of λ=0.2 and μ=0.6 values.

APPENDIX P

Objective Function	Mu (μ)																				
Lambda (λ) D281-08k	0.0	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1.0	1.1	1.2	1.3	1.4	1.5	1.6	1.7	1.8	1.9	2.0
0.0	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf
0.1	11645.05	11645.05	11746.34	11113.72	11099.10	11010.12	11074.12	11026.67	10832.56	9388.62	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf
0.2	11036.65	11039.56	11064.57	11039.77	11124.16	10908.15	10820.21	10770.62	9440.89	8764.39	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf
0.3	10833.31	10787.55	10833.31	10770.62	10770.62	10770.62	10770.62	9374.08	9086.45	8842.57	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf
0.4	10770.62	10770.62	10770.62	10770.62	10770.62	10770.62	9351.85	8899.53	8707.03	8969.93	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf
0.5	10770.62	10770.62	10770.62	10770.62	9786.18	9358.89	9514.13	9070.78	8879.93	8871.28	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf
0.6	10770.62	10770.62	10770.62	9906.24	9444.91	9391.92	8996.62	8835.15	8890.61	8911.13	10167.64	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf
0.7	10770.62	10770.62	9894.34	9449.17	9395.85	9077.49	8956.72	8832.48	8817.13	9006.88	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf
0.8	10770.62	9942.30	9358.33	9391.92	8996.62	9071.29	8716.07	8923.56	8739.54	9006.88	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf
0.9	9902.07	9351.86	9401.10	9014.97	9062.10	8751.98	8716.07	8776.73	8817.13	9651.58	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf
1.0	9351.86	9403.80	9449.17	9062.10	9071.29	8716.07	8867.34	8897.03	8897.03	9651.58	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf
1.1	9449.17	9442.40	8968.83	8996.62	8716.95	8724.94	8840.81	8817.13	8810.41	9651.58	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf
1.2	9358.89	8968.83	8996.62	9071.29	8665.56	8848.54	8869.10	8836.99	8770.10	9651.58	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf
1.3	9127.81	9071.29	8968.83	8688.00	8688.00	8841.77	8775.03	8809.61	8817.13	9651.58	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf
1.4	9070.89	8968.83	8996.62	8766.21	8665.56	8840.35	8917.60	8884.38	8927.19	9651.58	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf
1.5	9135.90	9154.87	8858.95	8903.46	8997.15	8700.51	8908.41	8917.60	8966.87	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf
1.6	9162.10	8990.22	8956.55	8892.16	9064.14	8797.72	8811.15	8817.13	9018.52	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf
1.7	9446.72	9076.32	8941.93	9072.09	8997.15	8717.63	8752.42	8813.85	9651.58	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf
1.8	9567.41	9102.52	9463.19	9143.48	8765.14	8752.42	8691.01	8719.38	9312.92	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf
1.9	Inf	9541.21	9574.27	9561.12	9390.78	8804.35	8752.42	8792.14	8844.94	9236.01	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf
2.0	9238.49	Inf	9608.06	9650.82	9443.43	9478.19	8788.03	8752.42	8812.12	8787.38	9282.25	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf

Figure 4.20: Parameterized of instance D281-08k, feasible to Initial Solution with of λ=1.2 and μ=0.4 values.

APPENDIX Q

Objective Function	*Mu (μ)*																				
Lambda (λ) D321-10k	**0.0**	**0.1**	**0.2**	**0.3**	**0.4**	**0.5**	**0.6**	**0.7**	**0.8**	**0.9**	**1.0**	**1.1**	**1.2**	**1.3**	**1.4**	**1.5**	**1.6**	**1.7**	**1.8**	**1.9**	**2.0**
0.0	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf
0.1	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf
0.2	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf
0.3	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf
0.4	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf
0.5	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf
0.6	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf
0.7	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf
0.8	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf
0.9	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf
1.0	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf
1.1	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf
1.2	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf
1.3	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf
1.4	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf
1.5	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf
1.6	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf
1.7	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf
1.8	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf
1.9	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf
2.0	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf

Figure 4.21: Parameterized of instance D321-10k, infeasible for the Initial Solution.

APPENDIX R

Objective Function	*Mu (μ)*																				
Lambda (λ) D361-09k	**0.0**	**0.1**	**0.2**	**0.3**	**0.4**	**0.5**	**0.6**	**0.7**	**0.8**	**0.9**	**1.0**	**1.1**	**1.2**	**1.3**	**1.4**	**1.5**	**1.6**	**1.7**	**1.8**	**1.9**	**2.0**
0.0	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf
0.1	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	10792.75	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf
0.2	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	10692.53	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf
0.3	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	10779.54	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf
0.4	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	10926.34	Inf	10696.86	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf
0.5	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	10813.79	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf
0.6	Inf	Inf	Inf	Inf	Inf	Inf	Inf	11015.30	Inf	Inf	10741.21	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf
0.7	Inf	Inf	Inf	Inf	Inf	Inf	11267.24	0.00	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf
0.8	Inf	Inf	Inf	Inf	Inf	Inf	Inf	10734.29	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf
0.9	Inf	Inf	Inf	Inf	11194.09	Inf	Inf	Inf	Inf	Inf	10707.24	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf
1.0	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	10675.16	10825.28	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf
1.1	Inf	Inf	Inf	Inf	10991.57	Inf	Inf	Inf	Inf	10644.41	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf
1.2	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	10646.15	Inf	11110.01	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf
1.3	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	11107.99	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf
1.4	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	10677.86	10929.14	11107.99	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf
1.5	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	10624.91	10925.72	10936.02	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf
1.6	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	10643.51	Inf	10864.89	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf
1.7	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	10624.09	Inf	10864.89	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf
1.8	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	10614.61	Inf	0.00	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf
1.9	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	10649.11	10649.11	Inf	10875.23	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf
2.0	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	10625.35	10783.56	Inf	10949.74	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf

Figure 4.22: Parameterized of instance D361-09k, feasible to Initial Solution with of λ=1.8 and μ=0.9 values.

APPENDIX S

Objective Function	*Mu (μ)*																				
Lambda (λ) D401-10k	**0.0**	**0.1**	**0.2**	**0.3**	**0.4**	**0.5**	**0.6**	**0.7**	**0.8**	**0.9**	**1.0**	**1.1**	**1.2**	**1.3**	**1.4**	**1.5**	**1.6**	**1.7**	**1.8**	**1.9**	**2.0**
0.0	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf
0.1	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf
0.2	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf
0.3	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf
0.4	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf
0.5	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	11414.50	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf
0.6	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	11559.33	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf
0.7	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	11414.50	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf
0.8	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf
0.9	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	11416.27	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf
1.0	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	11416.27	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf
1.1	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf
1.2	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf
1.3	Inf	Inf	Inf	Inf	Inf	Inf	11479.60	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf
1.4	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf
1.5	Inf	Inf	Inf	Inf	Inf	Inf	11416.16	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf
1.6	Inf	Inf	Inf	Inf	11617.16	11540.67	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf
1.7	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf
1.8	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf
1.9	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf
2.0	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf

Figure 4.23: Parameterized of instance D401-10k, feasible to Initial Solution with of λ=0.7 and μ=0.9 values.

APPENDIX T

Objective Function	***Mu (μ)***																				
Lambda (*λ*) D441-11K	**0.0**	**0.1**	**0.2**	**0.3**	**0.4**	**0.5**	**0.6**	**0.7**	**0.8**	**0.9**	**1.0**	**1.1**	**1.2**	**1.3**	**1.4**	**1.5**	**1.6**	**1.7**	**1.8**	**1.9**	**2.0**
0.0	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf
0.1	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf
0.2	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf
0.3	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf
0.4	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf
0.5	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	12553.56	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf
0.6	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	12419.71	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf
0.7	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	12566.17	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf
0.8	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	12483.22	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf
0.9	Inf	Inf	Inf	Inf	Inf	Inf	Inf	12603.01	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf
1.0	Inf	Inf	Inf	Inf	Inf	Inf	12585.70	12573.27	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf
1.1	Inf	Inf	Inf	Inf	Inf	Inf	12546.28	12470.63	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf
1.2	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf
1.3	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	12409.47	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf
1.4	Inf	Inf	Inf	Inf	Inf	Inf	12559.68	Inf	12426.97	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf
1.5	Inf	Inf	Inf	Inf	12476.65	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf
1.6	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf
1.7	Inf	Inf	Inf	Inf	Inf	Inf	12454.29	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf
1.8	Inf	Inf	Inf	Inf	Inf	Inf	12455.09	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf
1.9	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf
2.0	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf

Figure 4.24: Parameterized of instance D441-11k, feasible to Initial Solution with of λ=1.3 and μ=0.8 values.

APPENDIX U

Objective Function	*Mu (μ)*																				
Lambda (λ) D481-12k	0.0	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1.0	1.1	1.2	1.3	1.4	1.5	1.6	1.7	1.8	1.9	2.0
0.0	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf
0.1	17783.92	18503.76	17576.05	17578.99	17546.12	17981.66	16859.08	17188.45	16606.36	16325.68	14375.44	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf
0.2	17386.35	17612.22	17126.33	17514.82	16953.82	16795.39	16398.02	16136.13	16343.79	14503.81	14227.74	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf
0.3	16962.64	16990.10	16917.92	16723.31	16561.98	16470.29	16271.10	16015.70	14933.33	14344.28	14278.77	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf
0.4	16811.52	17022.47	16638.18	16151.55	16717.26	16271.23	16479.26	15379.84	14597.00	14252.50	14224.30	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf
0.5	16342.34	16398.74	16414.31	16348.26	16076.65	16173.76	15359.23	15049.35	14429.69	14155.77	14235.81	16384.17	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf
0.6	16491.25	16236.35	16330.44	16292.47	16178.93	15189.16	14921.03	14363.93	14222.75	14363.17	14393.30	16384.17	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf
0.7	16184.89	16515.00	16240.95	16355.51	15448.42	14683.19	14817.21	14724.50	14438.92	14343.49	14111.68	16374.82	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf
0.8	16086.59	16350.62	16325.19	15448.42	15696.34	14744.20	14566.07	14257.85	14503.79	14310.14	14353.17	15177.17	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf
0.9	16052.22	16020.12	15953.63	15400.91	14986.26	14612.63	14567.70	14265.09	14162.61	14457.12	14237.74	15177.17	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf
1.0	16181.31	16223.66	15342.14	15014.02	14959.40	14741.51	14368.93	14387.63	14458.43	14228.58	14251.81	15177.17	16365.46	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf
1.1	16498.34	15342.14	15535.52	15032.67	14526.17	14785.93	14297.98	14497.31	14328.96	14428.94	14338.58	15177.17	16365.46	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf
1.2	15512.09	15440.50	15056.84	14927.91	14719.83	14550.34	14440.05	14430.88	14385.16	14352.90	14306.91	15177.17	16365.46	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf
1.3	15495.97	15038.26	14985.79	14834.97	14710.12	14426.16	14396.64	14468.68	14465.94	14148.96	14300.55	15177.17	16365.46	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf
1.4	14920.05	15164.13	14855.92	14601.23	14493.55	14387.51	14398.25	14230.14	14565.99	14331.10	14304.77	15177.17	Inf	16384.96	Inf	Inf	Inf	Inf	Inf	Inf	Inf
1.5	15186.34	15001.16	14548.50	14393.12	14485.77	14502.68	14478.71	14356.84	14395.02	14435.36	14310.31	15177.17	15177.17	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf
1.6	15162.10	14608.30	14650.33	14594.00	14638.86	14321.46	14307.39	14384.76	14395.02	14128.15	14259.43	15177.17	15177.17	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf
1.7	15253.00	14850.76	14427.59	14309.20	14467.44	14325.60	14279.09	14132.35	14329.04	14252.20	14206.03	15177.17	15177.17	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf
1.8	14886.69	14821.16	14838.56	14637.60	14424.40	14469.58	14352.84	14394.65	14394.25	14395.41	14356.93	14153.42	15177.17	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf
1.9	Inf	Inf	14896.32	14682.90	14693.73	14520.34	14279.72	14229.97	14323.30	14400.33	14322.27	14109.85	15177.17	Inf	16384.96	Inf	Inf	Inf	Inf	Inf	Inf
2.0	Inf	Inf	Inf	14616.92	14366.04	14484.26	14496.35	14333.65	14277.04	14260.85	14283.77	14244.96	15177.17	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf

Figure 4.25: Parameterized of instance D481-12k, feasible to Initial Solution with of λ=1.9 and μ=1.1values.

APPENDIX V

Objective Function	*Mu (μ)*																				
Lambda (λ) E241-22k	0.0	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1.0	1.1	1.2	1.3	1.4	1.5	1.6	1.7	1.8	1.9	2.0
0.0	3785.34	3794.05	3723.51	3910.19	3722.96	3692.01	3690.81	3703.40	3909.71	3597.15	3355.59	3596.09	3654.47	3500.79	3721.99	3552.29	3609.75	3704.83	3668.05	3619.51	3655.40
0.1	1097.97	1109.52	1100.73	979.70	955.26	869.36	862.20	0.00	810.29	Inf	784.17	Inf	Inf	Inf	1805.31	0.00	1842.74	1846.58	1873.49	1858.95	1894.00
0.2	873.45	868.46	861.31	831.19	Inf	Inf	806.16	0.00	784.20	Inf	Inf	Inf	Inf	Inf	Inf	1791.26	0.00	1797.19	0.00	1834.15	1853.00
0.3	831.72	816.44	809.62	818.76	770.31	Inf	789.36	0.00	774.81	763.38	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	1811.62	Inf	Inf
0.4	795.27	797.64	Inf	Inf	Inf	782.41	773.51	774.19	769.10	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf
0.5	Inf	Inf	780.19	773.69	Inf	Inf	Inf	Inf	Inf	Inf	773.64	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf
0.6	Inf	782.18	781.76	0.00	0.00	778.76	771.23	Inf	763.47	Inf	757.59	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf
0.7	Inf	773.69	0.00	768.98	765.72	765.72	767.98	Inf	0.00	Inf	Inf	Inf	771.21	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf
0.8	785.49	Inf	770.53	765.72	765.72	770.75	0.00	Inf	754.08	759.78	749.78	Inf	771.21	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf
0.9	Inf	763.76	768.85	768.85	768.85	Inf	Inf	755.22	754.86	Inf	758.28	750.69	0.00	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf
1.0	771.18	770.00	Inf	768.22	Inf	Inf	Inf	Inf	751.94	750.95	755.47	751.94	768.36	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf
1.1	770.17	771.70	Inf	Inf	748.24	750.35	755.58	756.45	756.45	752.75	754.57	754.47	772.82	768.42	Inf	Inf	Inf	Inf	Inf	Inf	Inf
1.2	771.70	770.00	771.18	754.42	750.36	750.03	750.30	752.32	750.95	753.59	755.93	756.42	750.64	768.48	Inf	Inf	Inf	Inf	Inf	Inf	Inf
1.3	771.18	765.65	771.70	748.55	749.42	754.04	752.23	752.90	750.03	752.18	751.94	756.42	752.87	769.42	Inf	Inf	Inf	Inf	Inf	Inf	Inf
1.4	770.00	770.17	747.27	Inf	751.76	754.39	749.76	756.13	749.82	751.78	760.19	752.58	0.00	766.72	768.61	Inf	Inf	Inf	Inf	Inf	Inf
1.5	768.60	766.19	759.44	Inf	Inf	752.97	753.86	749.78	753.64	757.21	755.89	758.62	755.37	Inf	Inf	779.84	Inf	Inf	Inf	Inf	Inf
1.6	770.17	761.57	776.34	Inf	Inf	Inf	757.19	749.89	750.09	748.63	756.31	753.37	0.00	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf
1.7	752.33	761.57	782.84	Inf	Inf	Inf	756.30	756.30	759.41	757.78	758.80	0.00	753.82	Inf	Inf	750.11	Inf	Inf	Inf	Inf	Inf
1.8	757.17	771.53	773.19	784.15	Inf	Inf	Inf	753.41	750.93	746.22	756.75	757.97	750.11	760.74	Inf	Inf	Inf	Inf	Inf	Inf	Inf
1.9	Inf	773.19	775.62	777.64	Inf	Inf	Inf	Inf	Inf	756.30	753.64	752.92	753.02	755.07	Inf	Inf	Inf	Inf	Inf	Inf	Inf
2.0	771.53	779.84	775.28	777.71	Inf	Inf	Inf	Inf	Inf	Inf	756.77	752.84	752.84	759.63	Inf	Inf	Inf	Inf	Inf	Inf	Inf

Figure 4.26: Parameterized of instance E241-22k, feasible to Initial Solution with of λ=1.8 and μ=0.9 values.

APPENDIX W

Objective Function	*Mu (μ)*																				
Lambda (λ) E253-27-k	**0.0**	**0.1**	**0.2**	**0.3**	**0.4**	**0.5**	**0.6**	**0.7**	**0.8**	**0.9**	**1.0**	**1.1**	**1.2**	**1.3**	**1.4**	**1.5**	**1.6**	**1.7**	**1.8**	**1.9**	**2.0**
0.0	4504.61	4417.25	4294.71	4388.81	4449.75	4514.16	4703.56	4411.98	4516.61	445 7.5	4482.02	4610.98	4551.79	4625.26	4527.74	4679.57	4587.06	4636.79	4548.63	4593.91	4528.75
0.1	Inf	1222.16	1226.41	1192.34	1162.99	1121.67	Inf	1076.64	988.14	962.49	Inf	Inf	Inf	2364.42	2406.94	2462.55	2443.45	2513.61	2543.49	2565.41	2594.10
0.2	1127.70	1080.18	1141.39	1053.50	1069.71	Inf	998.61	958.32	955.07	901.82	Inf	1015.25	Inf	2334.07	2467.61	2375.19	2400.62	2438.66	2486.72	2481.71	2508.37
0.3	1043.51	1055.25	1023.17	1015.43	1001.43	976.07	980.91	968.27	902.11	901.79	932.11	Inf	Inf	Inf	2362.70	2368.22	Inf	2410.29	2385.35	2393.49	2410.15
0.4	Inf	992.36	1015.93	990.69	963.71	Inf	959.03	938.59	899.03	902.05	919.75	Inf	Inf	Inf	Inf	Inf	2320.56	2368.99	Inf	2415.44	2394.83
0.5	997.55	978.39	970.23	977.71	963.72	968.27	930.67	903.63	899.03	904.39	Inf	Inf	Inf	Inf	Inf	Inf	2360.50	2338.30	2349.13	2363.97	2455.34
0.6	972.56	973.98	963.11	950.39	957.77	930.67	900.27	Inf	901.63	Inf	918.66	Inf	Inf	Inf	Inf	Inf	Inf	2350.39	2350.07	2351.52	2365.96
0.7	967.03	963.29	958.49	963.44	929.72	903.54	902.94	903.73	901.48	Inf	907.70	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	2367.11	2359.56
0.8	971.68	Inf	Inf	937.15	930.67	Inf	Inf	899.03	Inf	906.69	Inf	Inf	908.69	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf
0.9	967.33	956.66	930.99	930.67	902.30	902.37	899.03	Inf	901.32	901.79	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf
1.0	963.86	939.78	935.12	900.75	900.96	903.15	902.28	Inf	903.73	899.03	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf
1.1	941.34	Inf	Inf	Inf	903.54	899.15	901.63	904.30	903.69	904.49	909.70	908.69	Inf	Inf	Inf	949.65	Inf	Inf	Inf	Inf	Inf
1.2	933.16	930.67	Inf	Inf	901.92	Inf	902.05	899.07	899.03	902.96	Inf	Inf	Inf	Inf	Inf	929.56	Inf	Inf	Inf	Inf	Inf
1.3	939.91	Inf	905.86	Inf	Inf	906.24	899.10	904.39	901.82	896.56	910.46	Inf	Inf	Inf	Inf	Inf	958.86	Inf	Inf	Inf	Inf
1.4	907.52	909.77	904.75	911.05	905.07	Inf	899.03	902.28	899.25	Inf	907.31	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf
1.5	910.19	910.50	911.43	911.05	Inf	Inf	Inf	902.28	900.49	Inf	Inf	Inf	Inf	Inf	Inf	Inf	908.69	Inf	Inf	Inf	Inf
1.6	Inf	Inf	Inf	Inf	Inf	909.39	Inf	Inf	905.59	Inf	Inf	Inf	Inf	Inf	Inf	Inf	908.69	Inf	Inf	Inf	Inf
1.7	Inf	Inf	910.18	Inf	908.49	909.39	909.80	Inf	Inf	899.00	917.94	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf
1.8	908.54	908.20	Inf	Inf	Inf	906.21	Inf	906.40	902.91	Inf	Inf	914.72	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf
1.9	909.43	910.19	Inf	907.58	909.77	911.43	908.14	Inf	905.86	Inf	921.92	Inf	Inf	Inf	Inf	Inf	Inf	Inf	923.36	Inf	Inf
2.0	908.93	Inf	912.20	Inf	Inf	Inf	911.10	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	945.69	Inf

Figure 4.27: Parameterized of instance E253-27k, feasible to Initial Solution with of λ=1.3 and μ=0.9 values.

APPENDIX X

Objective Function	*Mu (μ)*																				
Lambda (λ) E256-14k	**0.0**	**0.1**	**0.2**	**0.3**	**0.4**	**0.5**	**0.6**	**0.7**	**0.8**	**0.9**	**1.0**	**1.1**	**1.2**	**1.3**	**1.4**	**1.5**	**1.6**	**1.7**	**1.8**	**1.9**	**2.0**
0.0	2102.90	2171.21	2046.77	2283.07	2341.39	2187.15	2268.52	2185.87	2190.44	2377.49	3068.27	3940.87	3973.38	3932.17	3901.52	3974.27	3939.78	3923.98	4040.23	3952.26	3954.70
0.1	938.65	941.51	925.88	914.87	892.71	816.78	777.47	751.06	700.53	661.17	634.58	Inf	3033.17	3113.93	3131.45	3105.87	3153.89	3201.41	3217.35	3209.59	3209.10
0.2	793.88	783.10	774.96	751.64	738.44	698.59	702.95	682.78	659.72	625.68	Inf	687.13	Inf	3015.12	3035.28	3059.07	3105.16	3070.04	3086.14	3114.02	3117.08
0.3	722.32	707.83	689.84	696.48	704.73	692.55	686.80	663.93	623.96	624.53	Inf	Inf	734.94	Inf	3012.96	3041.90	3055.14	3087.25	3053.87	3047.30	3077.91
0.4	Inf	696.94	704.17	Inf	698.15	672.23	655.88	647.55	619.23	623.14	Inf	612.47	687.13	Inf	Inf	3015.67	Inf	3071.21	3052.89	3067.89	3079.50
0.5	694.51	686.86	Inf	677.58	676.50	661.49	656.40	620.03	628.04	624.19	Inf	Inf	612.46	708.87	Inf	Inf	3022.61	2997.61	3032.20	3048.24	3059.66
0.6	Inf	685.17	686.93	675.47	655.95	659.82	623.99	620.03	624.19	Inf	635.60	612.47	Inf	687.13	733.41	Inf	Inf	2970.45	3015.38	3050.43	3013.54
0.7	677.62	674.63	659.56	656.64	Inf	626.86	620.03	620.90	623.96	624.89	Inf	Inf	612.46	631.76	Inf	736.84	Inf	Inf	3033.26	2985.00	2975.55
0.8	675.62	665.65	666.05	Inf	647.04	624.53	620.03	624.89	624.53	Inf	Inf	610.39	612.46	610.39	Inf	Inf	743.55	Inf	Inf	3024.68	3043.91
0.9	666.57	663.96	664.98	651.59	620.82	624.65	624.69	624.19	624.19	Inf	Inf	Inf	612.22	Inf	655.73	714.22	731.31	Inf	851.46	2332.62	3030.32
1.0	Inf	665.19	652.63	624.30	620.10	627.12	Inf	621.18	619.23	Inf	Inf	613.48	Inf	613.22	612.22	690.29	722.84	748.02	Inf	853.28	Inf
1.1	657.66	652.98	626.86	623.88	620.03	623.97	620.03	619.23	624.19	619.23	630.94	Inf	Inf	Inf	Inf	664.10	Inf	724.19	745.89	750.23	Inf
1.2	636.29	Inf	626.69	620.03	625.04	624.19	624.19	624.19	624.19	Inf	Inf	612.47	Inf	Inf	613.22	Inf	688.45	Inf	Inf	739.44	Inf
1.3	Inf	632.93	624.19	625.37	620.82	624.19	624.19	622.89	620.10	619.23	633.56	Inf	Inf	Inf	613.46	Inf	671.13	706.92	728.51	747.92	Inf
1.4	636.69	624.19	623.80	622.89	625.04	Inf	Inf	617.52	623.97	624.53	Inf	Inf	Inf	Inf	612.54	613.22	Inf	688.16	Inf	Inf	Inf
1.5	624.19	623.07	625.37	Inf	625.53	Inf	Inf	628.35	619.82	Inf	636.24	Inf	612.46	Inf	Inf	613.22	Inf	678.04	689.44	717.18	Inf
1.6	624.19	623.80	628.35	Inf	623.96	Inf	Inf	Inf	619.23	Inf	Inf	612.47	Inf	Inf	Inf	Inf	Inf	Inf	Inf	711.33	728.51
1.7	Inf	623.35	622.89	628.35	Inf	Inf	Inf	625.53	Inf	621.42	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	672.97	Inf	Inf
1.8	620.82	624.19	Inf	Inf	Inf	624.19	619.82	624.19	620.90	Inf	Inf	611.51	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	697.32
1.9	Inf	Inf	624.19	620.82	628.39	620.82	619.23	624.39	619.23	Inf	Inf	Inf	Inf	611.50	Inf	Inf	Inf	Inf	Inf	681.25	Inf
2.0	623.97	624.19	Inf	624.19	624.19	Inf	Inf	Inf	Inf	Inf	637.00	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf

Figure 4.28: Parameterized of instance E256-14k, feasible to Initial Solution with of λ=0.8 and μ=1.3 values.

APPENDIX Y

Objective Function	Mu (μ)																				
Lambda (λ) E301-28k	0.0	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1.0	1.1	1.2	1.3	1.4	1.5	1.6	1.7	1.8	1.9	2.0
0.0	5259.57	5555.25	5330.05	5401.32	5433.11	5353.61	5468.80	5200.95	5325.45	5392.82	5061.17	5265.20	5257.24	5333.88	5201.92	5013.79	5479.49	5316.15	5223.98	5253.00	5470.44
0.1	1538.22	1603.68	1553.12	1375.33	1307.01	1274.69	1235.56	1162.36	1132.92	1083.09	1099.22	2313.77	2801.71	2831.85	2803.01	2820.01	2852.32	2895.11	2865.95	2892.32	2942.55
0.2	1245.28	1253.79	1244.56	1179.44	1161.67	1142.13	1128.26	1114.14	1082.19	1082.70	1099.68	1121.93	2283.96	2759.78	2790.38	2776.63	2822.95	2809.96	2806.88	2807.32	2840.62
0.3	1159.52	1109.53	1101.89	1118.62	1115.79	1137.09	1123.75	1082.19	1104.30	Inf	1103.47	1124.96	1118.56	2296.16	2745.34	2767.48	2789.68	2778.32	2810.38	2839.53	Inf
0.4	1118.79	1109.69	1118.03	1100.99	1101.73	1091.85	1079.53	1093.23	1077.68	1087.12	1092.45	1104.95	1105.61	1207.24	2296.16	Inf	Inf	2792.13	2790.38	2784.75	2786.64
0.5	1099.20	1102.24	1090.80	1090.55	1096.27	1080.89	1079.53	1072.05	1086.03	Inf	1087.34	1113.57	1132.01	1127.39	1207.24	2283.91	2750.40	2779.14	2788.62	2773.21	2791.37
0.6	1120.03	1089.82	1087.73	1079.04	1077.73	1079.53	1077.55	1081.43	1094.91	1095.77	Inf	1095.35	Inf	1111.42	1120.36	Inf	2287.76	Inf	Inf	2759.95	2777.41
0.7	1108.84	1110.10	1079.04	1076.91	1071.80	1074.49	1073.92	1094.80	1092.13	1087.05	1097.60	1094.70	1104.24	1114.27	1116.97	1189.76	1321.48	2278.50	Inf	Inf	2767.27
0.8	1095.49	1079.04	1071.80	1071.80	1071.80	1071.80	1065.00	Inf	1092.89	1087.58	1084.15	Inf	1106.76	1113.04	1110.53	1124.77	1207.24	Inf	2282.10	Inf	Inf
0.9	1078.50	1073.78	1071.80	1078.50	1072.50	1067.17	1086.75	1091.62	1091.03	1085.71	1088.41	1098.51	Inf	1119.32	1109.82	1125.54	1126.37	1207.24	0.00	2287.17	Inf
1.0	1069.29	1067.77	1068.73	1070.76	1066.78	1079.76	1096.54	1095.00	1098.51	1098.37	1089.94	1079.07	1109.45	1114.18	1127.29	1112.65	1116.97	1151.70	1207.24	Inf	2317.62
1.1	1067.93	1071.57	1071.87	1068.83	1076.31	1070.09	1088.98	1082.67	1097.32	1091.55	1098.37	1099.31	1103.97	1097.33	1137.58	1109.45	1120.56	1126.56	1187.14	Inf	Inf
1.2	1069.61	1067.77	1073.09	1066.52	1072.07	1070.42	Inf	1066.00	1080.74	1093.33	1077.58	1093.28	1075.81	1106.00	1105.64	1103.79	1110.62	1118.76	1118.56	1207.24	0.00
1.3	1067.93	1073.59	1067.77	1063.11	1064.01	Inf	1061.35	1076.37	1067.08	1068.55	1096.05	1094.40	1083.91	1107.51	Inf	1112.32	1106.19	1116.97	1125.11	1122.16	1203.94
1.4	1071.90	1069.61	1063.26	1064.96	Inf	1063.68	1062.99	1063.04	1068.53	Inf	1055.87	Inf	1084.56	1090.33	1092.45	1139.61	1107.28	1103.51	1124.28	1127.07	1180.64
1.5	1069.61	1064.55	1064.12	Inf	1063.02	Inf	1062.16	1066.68	1064.01	1072.15	1051.58	Inf	1051.51	1076.54	1093.71	1088.92	1107.41	1099.94	1126.11	1122.71	1127.82
1.6	1069.61	1063.47	1097.63	Inf	Inf	1062.74	1065.01	Inf	1062.25	1065.61	1056.59	Inf	1057.23	1080.15	1083.62	Inf	1105.24	1110.43	1105.53	1124.51	1125.70
1.7	1081.60	1071.90	Inf	1097.69	Inf	Inf	Inf	1057.33	1063.80	1065.99	1058.50	Inf	Inf	1074.31	1081.91	1077.13	Inf	1106.47	1108.71	1115.88	1124.28
1.8	1074.35	1104.68	Inf	Inf	Inf	Inf	Inf	Inf	0.00	1062.30	1062.02	1060.80	Inf	1060.87	1074.31	1074.76	Inf	1105.53	1100.47	1113.16	1122.47
1.9	1075.17	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	1061.38	1061.31	1060.80	1061.07	0.00	1080.27	Inf	Inf	1115.22	1110.57	1123.57
2.0	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	1055.88	1053.15	1054.34	1067.26	1079.27	1074.76	1094.41	Inf	1113.55	1108.66	1103.85

Figure 4.29: Parameterized of instance E301-28k, feasible to Initial Solution with of λ=1.5 and μ=1.2 values.

APPENDIX Z

Objective Function	*Mu (μ)*																				
Lambda (λ) E321-30k	**0.0**	**0.1**	**0.2**	**0.3**	**0.4**	**0.5**	**0.6**	**0.7**	**0.8**	**0.9**	**1.0**	**1.1**	**1.2**	**1.3**	**1.4**	**1.5**	**1.6**	**1.7**	**1.8**	**1.9**	**2.0**
0.0	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	6223.71	6602.82	6677.79	6579.72	6290.44	6605.21	6847.26	6639.16	6454.59	6515.95	6601.23
0.1	Inf	Inf	Inf	Inf	Inf	Inf	1451.30	Inf	Inf	Inf	Inf	Inf	3277.53	3230.94	3442.10	3425.32	3450.49	3553.40	3603.50	3583.49	3657.06
0.2	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	1243.79	Inf	Inf	3271.09	0.00	3369.27	Inf	0.00	0.00	3431.85
0.3	Inf	Inf	Inf	Inf	1287.32	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	3252.39	3343.39	Inf	0.00	3438.29	3460.54
0.4	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	1246.62	Inf	Inf	Inf	Inf	Inf	3314.22	3367.71	Inf
0.5	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	1294.99	Inf	Inf	3276.80	0.00	3295.82	Inf	Inf
0.6	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	1243.79	1307.71	Inf	Inf	3216.53	0.00	3312.67	3273.97
0.7	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	1284.61	1300.24	Inf	Inf	Inf	Inf	3322.39
0.8	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	1243.79	Inf	Inf	Inf	Inf	Inf	Inf
0.9	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf
1.0	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	1243.79	1284.04	1302.71	Inf	Inf	Inf
1.1	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	1267.45	Inf	Inf	Inf	1534.20
1.2	Inf	Inf	Inf	Inf	1144.23	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	1252.27	1281.07	Inf	Inf	Inf
1.3	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	1238.97	1272.75	Inf	Inf	Inf
1.4	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	1252.27	Inf	Inf	Inf
1.5	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	1144.66	0.00	Inf	Inf	Inf
1.6	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	1252.27	Inf	Inf
1.7	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf
1.8	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	1246.86	Inf
1.9	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf
2.0	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	1246.94

Figure 4.30: Parameterized of instance E321-30k, feasible to Initial Solution with of λ=0.8 and μ=1.3 values.

APPENDIX A1

Objective Function	*Mu (μ)*																				
Lambda (λ) E324-16k	**0.0**	**0.1**	**0.2**	**0.3**	**0.4**	**0.5**	**0.6**	**0.7**	**0.8**	**0.9**	**1.0**	**1.1**	**1.2**	**1.3**	**1.4**	**1.5**	**1.6**	**1.7**	**1.8**	**1.9**	**2.0**
0.0	3235.39	3064.51	2955.86	3029.09	3087.58	3101.73	3097.93	3150.33	3231.95	2964.08	Inf	5474.70	5690.80	5519.55	5536.62	5557.88	5676.29	5531.86	5610.06	5535.13	5628.03
0.1	1169.01	1184.26	1196.16	1137.23	1100.22	1029.23	996.20	955.20	872.09	860.75	798.83	Inf	4345.15	4406.69	4385.75	4439.38	4457.47	4436.48	4491.97	4508.38	4487.86
0.2	1005.04	998.94	996.53	953.95	929.22	911.35	878.92	869.35	841.36	787.92	807.93	877.87	Inf	4267.85	4353.83	4384.60	4391.85	4430.36	4380.09	4454.49	4446.10
0.3	936.09	924.03	892.33	896.33	871.03	880.06	876.66	843.60	794.05	788.05	817.11	767.97	939.53	Inf	4301.19	4254.46	4362.45	4332.02	4390.81	4383.90	4414.32
0.4	912.27	891.76	874.54	872.05	868.83	868.66	846.39	823.39	790.48	785.07	813.96	764.61	878.22	953.04	Inf	4221.15	4275.05	4326.78	4322.62	4360.96	4289.01
0.5	884.12	870.27	873.62	865.81	850.79	841.87	830.33	803.95	794.54	788.77	807.88	765.74	764.14	908.79	964.70	Inf	4214.34	4328.12	4292.04	4394.97	4267.71
0.6	873.62	853.64	866.52	863.27	855.08	822.46	802.48	792.09	788.05	784.44	808.59	765.74	763.31	875.03	940.06	959.29	Inf	4231.22	4311.57	4257.18	4261.06
0.7	863.98	861.37	848.59	838.82	832.55	813.99	800.67	803.95	787.01	786.03	797.05	764.14	764.14	798.58	Inf	947.52	984.57	Inf	4277.56	4280.42	4337.90
0.8	863.88	856.73	845.59	842.64	805.08	798.58	798.55	800.92	794.54	788.77	809.50	766.71	765.11	766.35	874.82	918.98	953.04	1007.65	Inf	4349.38	4315.74
0.9	856.48	851.79	833.90	827.61	803.17	800.67	798.55	798.74	786.03	794.54	802.98	770.25	772.81	765.11	831.42	882.08	940.06	961.12	1029.83	Inf	4341.48
1.0	826.15	831.57	817.54	800.83	797.65	798.49	790.66	798.55	782.10	788.00	806.85	767.17	771.78	775.29	770.98	875.01	903.98	944.71	Inf	1094.74	Inf
1.1	842.98	821.25	816.77	803.51	797.65	798.49	800.95	798.49	788.52	785.07	806.88	772.59	771.78	771.46	771.78	830.09	870.80	920.38	946.99	Inf	Inf
1.2	825.01	819.72	802.48	800.67	791.59	800.92	798.49	798.49	785.03	782.10	811.32	772.59	772.59	769.28	771.78	777.78	0.00	901.02	937.51	955.19	Inf
1.3	820.50	815.85	801.49	803.95	791.59	803.95	800.95	800.92	786.76	782.94	805.41	770.35	769.27	769.28	769.28	773.31	840.54	879.55	918.71	940.34	962.45
1.4	818.72	798.58	800.67	798.22	797.05	798.49	789.96	803.95	782.94	794.54	810.54	767.54	769.27	769.27	769.28	769.28	801.14	886.60	896.01	Inf	947.52
1.5	802.48	800.46	799.91	789.73	800.92	800.95	798.49	803.95	794.54	794.54	800.01	763.62	767.58	767.58	771.48	769.27	770.41	840.66	888.24	901.02	937.51
1.6	799.81	797.05	798.22	790.87	792.21	800.95	800.16	799.08	794.54	794.54	0.00	763.75	766.80	767.58	766.45	771.48	769.27	816.20	Inf	894.81	912.80
1.7	799.05	797.05	800.16	797.05	791.59	798.49	791.17	800.16	784.70	784.70	808.16	766.64	763.75	766.80	763.62	767.58	763.62	769.27	837.07	895.01	901.02
1.8	799.05	790.81	800.16	798.55	798.22	789.73	803.95	800.16	782.94	784.70	808.97	766.20	769.48	763.75	763.75	763.75	767.58	767.58	832.76	Inf	895.56
1.9	800.50	794.46	798.55	804.21	798.55	804.21	800.16	803.95	794.54	784.70	820.89	767.81	766.16	766.16	766.80	766.80	766.80	765.12	784.93	861.47	884.42
2.0	795.47	798.55	798.55	804.21	804.21	801.20	790.00	790.66	786.79	784.70	815.10	765.13	767.04	766.20	766.20	766.64	766.80	765.25	767.94	829.16	Inf

Figure 4.31: Parameterized of instance E324-16k, feasible to Initial Solution with of λ=0.6 and μ=1.2 values.

APPENDIX B1

Objective Function	*Mu (μ)*																				
Lambda (λ) E361-33k	**0.0**	**0.1**	**0.2**	**0.3**	**0.4**	**0.5**	**0.6**	**0.7**	**0.8**	**0.9**	**1.0**	**1.1**	**1.2**	**1.3**	**1.4**	**1.5**	**1.6**	**1.7**	**1.8**	**1.9**	**2.0**
0.0	7108.16	7237.96	7129.38	7474.79	7319.61	7358.97	7338.98	7561.57	7253.54	7020.63	6935.71	7207.33	7485.54	7453.74	7214.62	7291.21	7482.18	7571.09	7428.10	7223.14	7482.58
0.1	2119.44	2067.08	2076.59	1901.10	1819.32	1699.13	1692.65	1567.38	1537.47	1482.78	1496.06	Inf	Inf	4225.23	4161.57	4230.16	4275.39	4272.18	4347.70	4326.63	4348.93
0.2	1716.97	1717.95	1681.94	1603.72	1592.66	1546.89	1552.85	1529.14	1486.81	1472.93	Inf	Inf	Inf	Inf	Inf	4219.90	4239.08	4194.76	4191.68	4215.89	4218.42
0.3	1553.95	1519.19	1513.84	1492.43	1520.47	1542.87	1504.76	Inf	1508.56	1492.58	1507.49	1512.86	1520.75	Inf	Inf	4190.10	4192.59	Inf	Inf	4235.79	4200.12
0.4	Inf	1505.37	1506.63	1507.52	1492.28	Inf	1477.62	Inf	1482.14	1487.29	1501.39	Inf	Inf	Inf	Inf	Inf	Inf	Inf	4188.82	4185.21	4212.45
0.5	1503.72	Inf	1490.60	1513.74	Inf	1466.25	1475.83	1482.96	1479.87	Inf	1480.31	Inf	1518.80	1515.15	Inf	Inf	Inf	Inf	Inf	4181.99	4190.50
0.6	1512.91	1479.35	1482.55	1482.10	1484.20	1468.35	1472.67	1461.76	1492.12	Inf	1491.76	1501.99	1510.24	Inf	1523.48	Inf	Inf	Inf	Inf	Inf	Inf
0.7	1507.11	Inf	1482.46	1482.65	1474.58	1471.33	1472.05	1463.66	1465.45	1467.59	Inf	1498.07	1488.60	1516.28	1515.15	1607.94	Inf	Inf	Inf	Inf	Inf
0.8	1494.36	1470.04	1468.40	1463.49	1467.55	Inf	1461.18	1459.28	Inf	Inf	Inf	Inf	Inf	1509.85	Inf	1524.49	Inf	Inf	Inf	Inf	Inf
0.9	Inf	Inf	1463.49	Inf	Inf	1462.14	Inf	1462.12	1464.61	1463.49	1467.37	Inf	1503.99	1531.00	1502.11	1518.74	1518.01	Inf	Inf	Inf	Inf
1.0	1469.38	Inf	Inf	Inf	1457.02	Inf	Inf	Inf	1460.39	Inf	1455.63	1475.36	1510.98	1496.77	1503.76	Inf	1520.89	1566.97	Inf	Inf	Inf
1.1	1467.47	1464.13	1458.21	1457.11	1468.26	Inf	1459.71	1466.80	1463.03	1456.00	Inf	1471.97	1504.52	1495.42	1530.19	Inf	1524.49	1524.49	1608.10	Inf	Inf
1.2	1461.79	Inf	1460.53	Inf	Inf	1453.69	1461.10	1467.53	1462.11	1470.27	Inf	Inf	1461.98	1496.01	Inf	Inf	Inf	1521.37	1523.81	Inf	Inf
1.3	Inf	1466.53	Inf	Inf	1465.29	1460.66	1455.90	1461.02	1461.07	1462.64	1455.26	Inf	1466.98	Inf	1527.31	Inf	Inf	1523.19	1520.55	1522.51	Inf
1.4	1464.00	Inf	1449.49	1441.40	1454.69	1453.48	1453.99	1458.89	1458.82	1458.06	0.00	Inf	Inf	1489.54	Inf	1527.34	Inf	Inf	1523.19	1520.32	1611.06
1.5	1468.89	1463.68	1450.66	1443.31	Inf	1460.66	1463.98	1460.48	1459.30	1460.58	1450.76	1459.59	1455.00	1451.77	1487.41	Inf	Inf	Inf	1517.44	1517.44	1528.26
1.6	Inf	1480.23	Inf	Inf	Inf	1444.13	1461.30	1472.44	1455.55	1463.74	1463.17	Inf	Inf	Inf	Inf	1457.88	Inf	Inf	Inf	1515.15	1523.19
1.7	1480.68	Inf	Inf	Inf	1489.96	1481.02	Inf	Inf	1453.94	1454.29	Inf	1464.78	Inf	Inf	1460.99	Inf	1461.11	Inf	Inf	1524.80	1515.15
1.8	1477.23	1498.63	Inf	Inf	Inf	1489.13	1485.87	1452.45	Inf	1452.98	1461.96	1453.85	1451.22	Inf	Inf	1457.87	Inf	Inf	Inf	Inf	1523.19
1.9	1480.66	Inf	Inf	Inf	Inf	1477.75	1485.87	1482.60	Inf	1462.54	1451.06	Inf	Inf	1463.95	1456.46	1461.73	Inf	1462.73	Inf	Inf	1523.19
2.0	Inf	Inf	Inf	Inf	Inf	Inf	1481.36	1481.02	Inf	Inf	Inf	Inf	Inf	Inf	Inf	1458.81	1449.36	1460.81	1498.95	Inf	Inf

Figure 4.32: Parameterized of instance E361-33k, feasible to Initial Solution with of λ=1.4 and μ=0.3 values.

APPENDIX C1

Objective Function	*Mu (μ)*																				
Lambda *(λ)* E397-34k	0.0	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1.0	1.1	1.2	1.3	1.4	1.5	1.6	1.7	1.8	1.9	2.0
0.0	8408.87	8673.77	8861.33	8787.54	8321.22	8519.62	8423.31	8807.25	8656.44	8286.69	8517.18	8972.65	8879.40	9303.00	8703.97	8583.45	9024.89	9077.77	9158.63	8681.35	8933.15
0.1	Inf	Inf	1988.80	1944.64	1930.07	1822.10	Inf	1647.62	Inf	Inf	1469.89	Inf	4495.43	4490.65	4524.01	4701.08	4647.53	4660.64	4708.49	4859.47	4728.35
0.2	Inf	1796.66	Inf	Inf	Inf	1644.00	1563.42	Inf	Inf	1441.51	Inf	Inf	Inf	4397.88	4443.17	4474.68	4551.90	4555.78	4573.82	4529.53	4651.49
0.3	Inf	Inf	Inf	1613.44	1569.33	1550.34	Inf	Inf	1446.45	1442.38	Inf	Inf	Inf	Inf	4465.16	4437.93	4500.26	4537.50	4573.64	4507.16	4545.57
0.4	1613.50	1588.91	Inf	1569.94	Inf	Inf	1520.77	1440.23	1444.72	Inf	1432.87	Inf	Inf	Inf	Inf	4491.49	4406.49	4523.31	4491.14	4587.69	4515.24
0.5	Inf	Inf	1540.43	Inf	Inf	Inf	1450.93	Inf	1444.16	Inf	Inf	1418.32	Inf	Inf	Inf	Inf	4466.14	4438.89	4483.87	4467.98	4525.91
0.6	1554.81	Inf	Inf	1505.31	1503.29	Inf	1438.37	Inf	1444.16	1436.85	Inf	Inf	Inf	Inf	Inf	Inf	Inf	4486.70	4446.75	4463.05	4481.77
0.7	Inf	Inf	Inf	1512.44	Inf	Inf	1448.19	Inf	Inf	1438.03	Inf	Inf	1433.31	1460.12	Inf	Inf	Inf	Inf	4452.40	Inf	4449.64
0.8	Inf	1504.00	1519.72	Inf	1446.62	1444.18	Inf	1442.19	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	4445.91	4449.16
0.9	Inf	Inf	Inf	1442.17	1443.78	Inf	1440.33	Inf	Inf	Inf	Inf	1433.28	1419.70	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf
1.0	Inf	Inf	Inf	1446.61	1442.48	1442.99	Inf	1438.05	Inf	Inf	Inf	Inf	1423.37	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf
1.1	Inf	Inf	1441.94	1443.75	Inf	1442.47	Inf	Inf	Inf	1441.00	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf
1.2	Inf	1446.82	Inf	Inf	Inf	Inf	1444.62	Inf	1444.62	Inf	Inf	Inf	Inf	Inf	Inf	1449.06	Inf	Inf	Inf	Inf	Inf
1.3	Inf	1442.48	Inf	1439.77	Inf	Inf	Inf	Inf	1440.42	Inf	Inf	Inf	1433.66	Inf	1426.17	Inf	Inf	Inf	Inf	Inf	Inf
1.4	1440.98	1439.79	1444.12	1442.92	Inf	1438.59	Inf	Inf	Inf	1440.42	1438.90	Inf	1419.68	Inf	Inf	Inf	1466.82	Inf	Inf	Inf	Inf
1.5	Inf	Inf	Inf	Inf	Inf	Inf	1436.18	Inf	1440.34	Inf	Inf	Inf	Inf	1426.53	Inf	Inf	Inf	Inf	Inf	Inf	Inf
1.6	Inf	Inf	Inf	1444.59	Inf	1442.99	1437.03	1434.79	1443.46	1438.63	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf
1.7	Inf	Inf	1441.62	1442.48	1441.37	Inf	1434.99	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf
1.8	Inf	Inf	1439.23	Inf	Inf	1438.54	Inf	Inf	1438.83	1441.42	Inf	1411.05	Inf	Inf	Inf	1429.23	1426.17	Inf	Inf	Inf	Inf
1.9	Inf	Inf	Inf	Inf	1439.23	Inf	Inf	Inf	Inf	1440.56	Inf	Inf	Inf	Inf	Inf	Inf	1422.75	Inf	Inf	Inf	Inf
2.0	Inf	1441.62	Inf	Inf	Inf	1432.50	Inf	Inf	Inf	1440.56	1429.91	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	1496.40	Inf

Figure 4.33: Parameterized of instance E397-34k, feasible to Initial Solution with of λ=1.8 and μ=1.1 values.

APPENDIX D1

Objective Function	*Mu (μ)*																				
Lambda *(λ)* E400-18k	0.0	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1.0	1.1	1.2	1.3	1.4	1.5	1.6	1.7	1.8	1.9	2.0
0.0	4179.05	3941.35	4374.18	4191.76	4376.95	4299.33	4061.59	4162.48	4325.91	4078.79	5669.71	7615.50	7576.15	7621.91	7631.42	7774.91	7663.35	7765.37	7626.79	7631.80	7682.67
0.1	1523.70	1507.15	1452.96	1436.40	1353.34	1311.77	1232.51	1151.71	1089.37	1061.33	1011.58	Inf	5807.88	5901.64	5978.24	5997.54	6042.51	6105.35	6119.95	6145.66	6182.43
0.2	1264.22	1272.18	1215.48	1167.74	1163.43	1136.62	1103.03	1077.02	1049.12	987.78	1003.51	1080.09	Inf	5716.37	5820.16	5861.52	5962.92	6019.56	6069.45	6069.23	6019.79
0.3	1180.45	1135.67	1105.18	1121.15	1102.67	1088.86	1074.09	1062.77	1002.95	987.78	1002.33	970.13	Inf	Inf	5729.89	5742.06	5801.02	5812.98	5960.93	5946.87	6002.17
0.4	1118.87	1112.67	1102.89	1082.39	1090.23	1061.67	1047.41	1022.84	988.32	989.19	1011.34	971.47	1081.15	1190.84	4793.08	5812.13	5688.37	5768.23	5820.61	5808.25	5897.25
0.5	1093.14	1082.17	1102.37	1071.32	1062.48	1056.02	1021.72	993.07	987.78	988.72	992.72	970.45	970.45	1126.47	1194.32	Inf	5848.34	5772.00	5810.60	5811.60	5796.27
0.6	1078.46	1078.99	1078.63	1071.47	1049.02	1025.89	998.87	993.07	987.78	989.19	1010.13	968.35	968.35	1078.99	Inf	1203.36	Inf	5745.14	5771.64	5793.13	5777.99
0.7	1075.00	1061.57	1061.91	1050.55	1026.34	1005.05	1001.22	999.66	989.19	988.72	989.43	968.35	968.35	997.07	1123.48	Inf	1210.37	Inf	5771.02	5852.86	5786.56
0.8	1051.92	1053.41	1050.29	1038.06	1022.13	987.43	992.45	989.34	989.19	985.90	1004.33	968.35	968.35	969.59	1077.26	1145.88	1192.13	Inf	Inf	5796.75	5807.74
0.9	1061.19	1064.74	1033.16	1020.12	989.34	987.43	992.45	986.31	987.56	982.22	996.92	968.35	968.35	969.59	1059.03	1090.08	Inf	1187.90	1336.66	Inf	5804.65
1.0	1064.74	1024.24	1025.07	1015.87	987.43	988.53	989.34	986.31	987.38	982.22	997.79	966.93	966.93	969.59	969.59	1076.60	1126.81	1167.97	1201.55	1323.23	Inf
1.1	1039.81	1021.89	1004.32	990.95	988.53	988.53	989.34	989.34	981.41	989.86	997.14	966.93	966.93	968.17	968.17	1048.75	1105.86	1145.09	1182.70	1202.47	1359.55
1.2	1017.49	1013.65	990.63	987.43	988.53	988.53	989.34	989.34	982.22	987.38	994.27	966.93	966.93	968.17	968.17	981.10	1057.68	1126.09	1164.06	1190.71	1203.17
1.3	1019.37	1012.33	989.34	988.53	989.34	989.34	992.19	989.34	981.41	982.22	998.58	966.93	966.93	968.17	968.17	968.17	1053.25	1116.17	1142.28	Inf	1182.50
1.4	1015.72	990.05	987.43	989.34	988.53	989.34	986.31	989.34	997.07	989.86	995.35	966.93	966.93	968.17	968.17	968.17	997.07	1070.27	1096.15	1149.65	Inf
1.5	992.54	989.59	989.59	988.53	989.34	988.53	989.34	989.34	989.18	982.22	1012.94	966.93	966.93	968.17	968.17	968.17	968.17	1053.25	1092.04	1101.69	Inf
1.6	987.43	989.59	989.59	989.59	988.53	988.53	986.31	989.34	990.93	982.22	997.69	Inf	968.41	968.17	968.17	968.17	968.17	1042.93	1072.01	1100.24	1143.17
1.7	989.59	988.53	989.59	989.59	989.34	989.34	986.31	989.34	988.17	981.41	996.96	Inf	969.49	Inf	968.17	968.17	968.17	967.35	1046.02	1065.62	1101.69
1.8	989.59	988.53	989.59	988.53	988.53	989.34	986.63	989.34	984.80	980.85	984.36	Inf	Inf	Inf	970.47	Inf	968.17	967.35	1059.38	1072.01	1098.05
1.9	988.53	988.53	989.59	989.59	988.53	989.59	992.19	986.31	997.07	986.80	1009.81	Inf	Inf	971.55	Inf	Inf	Inf	967.35	985.40	1042.23	1056.83
2.0	989.59	989.59	989.59	988.53	988.53	988.53	986.63	989.34	987.56	981.41	994.84	969.49	Inf	Inf	970.47	970.47	Inf	Inf	967.35	1073.16	1071.54

Figure 4.34: Parameterized of instance E400-18k, feasible to Initial Solution with of λ=1.0 and μ=1.1 values.

APPENDIX E1

Objective Function	Mu (μ)																				
Lambda (λ) E421-41k	0.0	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1.0	1.1	1.2	1.3	1.4	1.5	1.6	1.7	1.8	1.9	2.0
0.0	9798.40	9807.69	9375.47	9673.82	9871.35	9817.12	9629.85	9521.69	9672.74	9908.73	9371.60	10066.83	10159.25	10099.46	10032.71	10170.38	10178.87	9813.91	10241.46	10101.91	9972.63
0.1	2882.04	2807.54	2814.30	2419.98	2402.38	2334.74	2335.73	2053.47	2082.19	1972.16	1960.23	Inf	6110.58	6126.34	6085.67	6164.48	6164.74	6224.04	6200.24	6343.03	6325.26
0.2	2311.89	2292.82	2232.06	2156.88	2063.98	2060.62	2071.90	2024.43	1973.43	1961.83	1966.61	2000.24	4820.34	6116.72	6097.15	6117.79	6123.34	6046.05	6089.35	6110.27	6130.19
0.3	2088.50	2009.77	2013.46	2007.91	2033.21	2067.56	2035.07	1971.24	2023.82	1956.89	1960.45	2014.66	2035.33	4748.20	6030.40	6097.55	6120.52	6140.01	6145.31	6143.84	6097.75
0.4	2008.89	2004.92	2007.96	1985.76	1996.07	1980.87	1978.25	2022.10	1954.38	1973.74	1965.97	1992.13	2000.72	Inf	4765.42	6019.31	6127.16	6094.21	6094.77	6154.52	6140.97
0.5	2023.49	2007.86	2014.36	1988.31	2004.99	1970.86	1970.16	1972.25	1962.88	1957.94	1969.10	1995.25	2007.85	2010.46	Inf	4768.48	6097.38	6040.66	6081.10	6104.06	6112.08
0.6	2011.42	1987.69	2003.04	1959.91	1970.88	1956.70	1966.10	1954.33	1968.38	1960.70	1962.99	1978.18	2015.73	2009.04	2019.71	Inf	4839.15	5981.01	6030.45	6105.63	6101.11
0.7	1987.68	1976.21	1977.63	1952.78	1959.40	1968.37	1954.41	1953.79	1950.94	1967.99	1959.27	1952.39	1994.93	2003.97	2016.35	2177.28	Inf	4790.35	6001.30	6011.75	6070.03
0.8	2003.61	1964.84	1963.72	1960.24	1967.08	1961.65	1935.28	1947.55	1961.42	1953.27	1959.15	1970.88	1990.28	2014.68	1990.83	2009.16	Inf	Inf	Inf	6066.23	6016.51
0.9	1957.12	1951.19	1960.44	1959.54	1970.93	1946.55	1951.89	1942.43	1957.54	1946.04	1963.33	1959.04	1989.41	2003.27	2001.32	2016.51	2027.61	Inf	Inf	4744.39	6012.85
1.0	1957.87	1964.82	1954.30	1960.13	1949.47	1945.82	1941.25	1941.59	1948.79	1959.28	1953.52	1946.72	1994.61	1988.18	2014.49	1992.70	2009.16	2111.46	Inf	Inf	4713.15
1.1	1961.97	1956.48	1953.50	1958.09	1938.67	1928.13	1956.79	1939.60	1946.30	1961.06	1959.53	1964.09	1990.97	1974.00	2053.08	1994.84	2016.18	2009.68	2170.53	Inf	Inf
1.2	1951.88	1956.86	1954.85	1940.91	1933.57	1933.63	1924.86	1942.21	1958.30	1951.77	1957.41	1956.90	1959.15	1973.72	1977.75	2012.83	1993.55	2009.95	2027.45	Inf	Inf
1.3	1956.68	1960.21	1952.41	1924.65	1939.28	1936.48	1935.48	1930.87	1940.66	1931.93	1951.51	1953.23	1963.25	1959.60	2014.46	1997.06	1997.71	2022.06	2022.68	2027.61	Inf
1.4	1960.87	1963.44	1920.51	1929.85	1931.43	1928.67	1934.60	1939.58	1936.74	1938.18	1917.83	1945.06	1965.27	1952.50	1977.25	2060.77	2010.25	1989.58	2016.51	2027.45	2177.59
1.5	1965.89	1951.20	1939.63	1926.36	1930.24	1920.59	1935.06	1936.00	1942.09	1918.77	1927.65	1938.71	1930.79	1943.80	1975.57	1964.56	2008.88	1984.07	2015.18	2022.68	2019.71
1.6	1960.61	1978.87	1969.11	1981.93	1930.47	1936.54	1921.48	1931.09	1930.95	1929.17	1952.76	1932.74	1932.57	1940.59	1935.88	1940.90	1991.30	1992.69	1994.82	2010.44	2014.65
1.7	1942.07	1948.04	1993.07	1983.86	1985.15	1988.54	1920.06	1929.66	1920.82	1936.26	1940.34	1931.04	1937.98	1938.99	1942.48	1943.51	1953.47	2005.30	2007.17	2015.02	2014.65
1.8	1946.83	2018.13	2013.44	1993.42	1992.66	1979.08	1984.75	1927.53	1934.93	1929.35	1932.49	1931.36	1934.17	1940.12	1945.09	1942.65	1952.20	2007.66	2002.33	1999.57	2015.02
1.9	1929.64	2009.52	2012.83	1993.41	1998.70	1978.88	1985.48	1987.03	2005.36	1931.31	1927.53	1941.12	1932.05	1949.51	1939.97	1941.54	1943.12	1953.28	1997.81	1984.85	2010.44
2.0	2006.30	2005.76	2009.52	2014.57	1999.87	1992.89	1977.29	1984.06	2002.33	2009.90	1928.20	1937.61	1926.05	1941.40	1941.63	1949.96	1938.37	1951.26	1994.58	1995.35	1985.31

Figure 4.35: Parameterized of instance E421-41k, feasible to Initial Solution with of λ=1.4 and μ=1.0 values.

APPENDIX F1

Objective Function	*Mu (μ)*																				
Lambda (λ) E481-38k	**0.0**	**0.1**	**0.2**	**0.3**	**0.4**	**0.5**	**0.6**	**0.7**	**0.8**	**0.9**	**1.0**	**1.1**	**1.2**	**1.3**	**1.4**	**1.5**	**1.6**	**1.7**	**1.8**	**1.9**	**2.0**
0.0	11503.83	11878.76	11799.74	11022.72	11104.18	11306.15	11712.31	11292.39	11432.83	11138.79	11428.71	11877.79	11854.58	11993.70	12116.44	11539.56	11428.68	12125.90	12003.44	12008.13	11787.06
0.1	2447.13	2361.95	Inf	2241.98	2215.04	Inf	Inf	2030.60	1923.40	1820.20	1775.19	Inf	5789.60	5872.79	5860.33	6071.75	6073.13	6246.49	6292.33	6295.31	6197.85
0.2	2160.19	2109.12	2076.42	2040.56	2000.81	1990.96	1915.15	1851.07	1823.60	1743.75	1760.62	1942.37	Inf	5773.20	5812.80	5895.73	5839.51	5878.22	6099.37	6166.01	6141.57
0.3	2014.57	2030.68	1961.06	1959.05	1944.24	1913.28	1842.55	Inf	Inf	1744.01	1754.88	Inf	Inf	Inf	5748.66	5858.41	5811.90	5845.81	5816.95	6067.15	5984.53
0.4	Inf	1974.34	1906.83	1899.62	1857.35	1867.31	1792.39	1722.43	1745.78	Inf	Inf	Inf	1940.97	Inf	Inf	5784.60	5841.86	5822.49	Inf	5875.31	5871.31
0.5	1936.19	1906.95	1857.49	1859.27	1852.47	Inf	1781.43	1741.75	1744.81	Inf	1759.52	Inf	Inf	1937.03	2033.95	Inf	5850.94	5819.45	5873.82	Inf	Inf
0.6	1913.24	1871.69	1863.51	1864.32	1794.10	1782.37	1742.71	Inf	1750.98	Inf	1747.57	Inf	Inf	1953.62	Inf	2045.07	Inf	Inf	Inf	5841.23	5870.82
0.7	1859.75	1849.62	1834.53	1812.16	1792.15	1744.53	Inf	Inf	1744.18	Inf	Inf	Inf	Inf	Inf	Inf	2022.73	2165.69	Inf	5787.65	5783.79	5862.65
0.8	1831.90	1819.23	1803.94	1801.79	1745.94	1742.24	1742.36	1744.54	1746.26	1745.77	Inf	Inf	Inf	Inf	1910.30	1934.46	Inf	Inf	Inf	5770.93	5786.18
0.9	1849.18	1806.29	1809.21	1737.89	Inf	Inf	Inf	Inf	1738.18	Inf	Inf	Inf	Inf	Inf	Inf	1937.53	Inf	Inf	Inf	Inf	5799.82
1.0	Inf	1810.84	1781.07	1738.54	Inf	1752.95	1748.19	1744.63	1745.78	Inf	1758.14	1711.31	Inf	Inf	Inf	1913.80	1933.80	Inf	Inf	Inf	Inf
1.1	1803.09	Inf	1734.65	1745.47	Inf	1744.30	Inf	Inf	1740.13	Inf	1720.16	Inf	Inf	Inf	Inf	Inf	Inf	1933.80	Inf	2026.55	Inf
1.2	1776.29	1735.13	1738.94	1744.03	1745.60	1744.23	1740.67	Inf	1745.15	Inf	Inf	Inf	1709.72	Inf	Inf	Inf	1910.30	1951.89	Inf	Inf	Inf
1.3	1729.73	1736.00	1749.29	Inf	1743.64	Inf	1744.23	Inf	Inf	1746.17	Inf	Inf	Inf	Inf	1709.72	1711.31	Inf	Inf	1933.80	Inf	Inf
1.4	Inf	1722.89	1753.09	Inf	1735.89	1748.93	1743.26	Inf	Inf	Inf	1758.95	1709.72	Inf	Inf	Inf	Inf	Inf	1929.75	Inf	Inf	Inf
1.5	1737.99	Inf	1742.36	1745.28	1742.30	1739.71	1742.29	Inf	1744.66	1742.60	1759.38	Inf	Inf	Inf	Inf	Inf	1709.72	Inf	Inf	1948.99	Inf
1.6	Inf	Inf	1746.78	1742.23	1742.03	Inf	1730.11	Inf	1743.24	Inf	1758.90	Inf	Inf	Inf	Inf	Inf	Inf	Inf	1930.78	Inf	Inf
1.7	1739.07	1736.97	1750.82	Inf	1748.72	1746.74	1745.69	1750.99	1745.40	1736.46	Inf	1748.87	Inf	Inf	Inf	Inf	Inf	Inf	Inf	1925.96	Inf
1.8	Inf	Inf	1741.08	Inf	1747.82	1737.15	1748.46	Inf	Inf	1740.79	1761.16	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	1927.38	Inf
1.9	1751.61	1740.54	1735.63	Inf	Inf	Inf	Inf	1740.58	1740.39	1743.15	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	1923.58
2.0	1740.27	Inf	1737.55	1743.40	1742.15	1745.08	1743.68	1735.32	1739.30	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	1925.75

Figure 4.36: Parameterized of instance E481-38k, feasible to Initial Solution with of λ=1.2 and μ=1.2 values.

APPENDIX G1

Objective Function	*Mu (μ)*																				
Lambda (*λ*) E484-19k	0.0	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1.0	1.1	1.2	1.3	1.4	1.5	1.6	1.7	1.8	1.9	2.0
0.0	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	10310.93	10222.10	10031.38	10288.51	10345.80	10275.61	10186.28	10207.10	10386.44	10210.87
0.1	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	7670.60	Inf	7890.65	7928.29	8110.75	8092.86	8076.24	8078.44	8126.92
0.2	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	7805.53	7818.81	7810.82	7980.41	7988.17	Inf
0.3	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	7642.57	7783.06	7784.30	7880.29	Inf	7880.33
0.4	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	7616.60	Inf	Inf	7868.69	7859.60
0.5	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	7627.81	7617.26	7715.78	7790.71
0.6	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	7663.83	Inf	Inf
0.7	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	7647.95	Inf	7721.32
0.8	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf
0.9	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf
1.0	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf
1.1	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf
1.2	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf
1.3	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf
1.4	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf
1.5	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf
1.6	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf
1.7	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf
1.8	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf
1.9	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf
2.0	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf

Figure 4.37 Parameterized of instance E484-19k, feasible to Initial Solution with of λ=0.4 and μ=1.6 values.