UNIVERSITY KASDI-MERBAH OUARGLA
Faculty of New Technologies of Information and Communication
Department of Computer Science and Information Technologies

Master Thesis

Specialty: Computer Science Industrial

To obtain diploma
Master of Computer Science

Thesis subject:

Smart waste collection vehicles planning for Ouargla city using a metaheuristic

Submitted by :
Mammar GHARBI

Mr.Abdelhakim CHERIET **as Supervisor**
Mr.Amin KHALDI(P) **as Examiner**
Mr.Fouad BEKKARI **as Examiner**

University Year: 2019/2020

Abstract

In order to solve the optimization problem of municipal waste collection and transportation in Ouargla city, this thesis constructs a capacitated vehicle routing problem(CVRP) model and applies a meta-heuristic to solve the model under the scenario of the application of smart bins. we chose two multiobjective evolutionary algorithms (MOEAs) the Fast Elitist Non-dominated Sorting Genetic Algorithm 2, and strength Pareto evolutionary algorithm 2, due to their global optimization capability. The effectiveness of the two algorithms is verified by applying the case of waste collection and transportation in a proposed model for acquiring reliable conclusions, and the application of the model is tested by setting different waste fill levels. The results show that total costs reduce when applying smart waste bins, especially if the prior knowledge of the quantity of waste is well exploited in the optimization process.

Keywords : Multiobjective optimization,VRP, evolutionary algorithm,MOEA,NSGA-II, SPEA2

Résumé:

Afin de résoudre le problème d'optimisation de la collecte et du transport des déchets municipaux dans la ville de Ouargla, cette thèse construit un modèle de problème de routage de véhicule capacitif (CVRP) et applique une méta-heuristique pour résoudre le modèle sous le scénario de l'application de smart bins. nous avons choisi deux algorithmes évolutionnaires multi-objectifs, l'algorithme génétique élitiste de tri non- dominé 2 et l'algorithme évolutionnaire de force Pareto 2 , en raison de leur capacité d'optimisation globale. L'efficacité des deux algorithmes est vérifiée en appliquant le cas de la collecte et du transport des déchets dans un modèle proposé pour acquérir des conclusions fiables, et l'application du modèle est testée en définissant différents niveaux de remplissage des déchets. Les résultats montrent que les coûts totaux diminuent lors de l'application de poubelles intelligentes, surtout si la connaissance préalable de la quantité de déchets est bien exploitée dans le processus d'optimisation.

Mots-clé : Optimisation multiobjectif, problème de tournées de véhicules, algorithme évolutionnaire, AEOM, NSGA-II, SPEA2

Acknowledgements

First of all, I thank Allah, the Almighty, who has given me the strength, patience, and will to do this modest work. This thesis would not have been possible without the conscious intervention of some people. I would like to thank Dr.Abdelhakim Cheriet for giving me the wonderful opportunity to complete my master thesis under his supervision, it is truly an honor. Thank you for all the advice, ideas, moral support, and patience in guiding me through this project. Our thanks also go to all of my teachers during the school years. Finally, I would like to thank all those who, from near or far, have contributed to the achievement of this work.

Dedication

To my father ;

To my wife and my children

Wasim and Aslan ;

To my mother, may her soul

rest in peace.

MAMMAR GHARBI

Contents

General introduction **1**

1 Waste collection and vehicle routing problems **4**

1.1 Preliminaries 5

1.1.1 The complexity of an algorithm 5

1.1.2 Combinatorial Optimization 6

1.1.3 Multi-Objective Optimization Problem (MOP) 8

1.2 The vehicle routing problem (VRP) 11

1.2.1 The traveling salesman problem (TSP) 11

1.2.2 Definition of Vehicle Routing Problem 11

1.2.3 Vehicle Routing Problem's Formulation 12

1.2.4 Real-life Routing Problems 14

1.2.5 VRP Variants 15

1.3 Waste collection as a vehicle routing problem 16

1.3.1 VRP for collection 16

1.3.2 VRP for waste collection 17

1.3.3 Waste collection vehicle routing problem state of the art 18

2 A literature review on methods of solving the vehicles routing problem 19

2.1 Exacts Methods 19

2.1.1 Dynamic Programming 20

2.1.2 Branch and bound 20
2.1.3 Advantages and disadvantages of exact methods 21
2.2 Heuristic and Meta-heuristic methods 21
2.2.1 Heuristic Methods 21
2.2.2 Meta-heuristics Methods 23

3 Evolutionary Algorithms for Multi-Objective Optimization 28
3.1 Basic Principles of Evolutionary Algorithms 28
3.1.1 Definition 28
3.1.2 Basic Terminology 29
3.1.3 Basic Structure 30
3.2 Key Issues in Multi-Objective Search 34
3.2.1 Fitness Assignment and Selection 34
3.2.2 Population Diversity 35
3.2.3 Elitism 37
3.3 Parallel Evolutionary Algorithms 37
3.3.1 The Island Model 37
3.3.2 The Master-Slave Model 38
3.3.3 The cellular model 39
3.4 Structures of Various MOEAs 40
3.4.1 Multi-Objective Genetic Algorithm (MOGA) 40
3.4.2 Niched Pareto Genetic Algorithm (NPGA): 40
3.4.3 Nondominated Sorting Genetic Algorithm (NSGA) 40
3.4.4 Strength Pareto Evolutionary Algorithm (SPEA): 41
3.5 Strength Pareto Evolutionary Algorithm (SPEA2) 41
3.6 A fast and elitist multiobjective genetic algorithm (NSGA-II) 42
3.6.1 Procedure of NSGA-II 42
3.6.2 Crowding distance calculating 43

4 Implementation and experimental results **45**

4.1 Municipal Waste collection problem in Ouargla city 45

4.1.1 Definition 45

4.1.2 Problem description 46

4.2 Implementation of NSGA-II and SPEA2 49

4.2.1 Implementation Environment 49

4.2.2 Solution Representation 50

4.3 Analysis of results 52

4.3.1 Data presentation 52

4.3.2 Parameters setting 53

4.3.3 Results and analysis 54

General Conclusion **59**

List of Figures

1.1 Relations between complexity classes. 6

1.2 Representation of the decision variable space and the corresponding objective space.[1] 9

1.3 The Pareto front of a set of solutions in a two objective space. 10

1.4 An instance of a VRP (left) and its solution (right). 12

1.5 Examples of two vehicle routes for a waste collection problem[2]. 17

3.1 The classic crossover operators. 32

3.2 Example of mutation operators. 32

3.3 The general scheme of an Evolutionary Algorithm as a flow-chart. 33

3.4 Illustration of island EAs[3]. 38

3.5 Illustration of master-slave EAs[3]. 39

3.6 Illustration of cellular EAs[3]. 39

3.7 Schematic representation of NSGA-II working principle. 43

3.8 Crowding distance calculation. 44

4.1 Simplified routing diagram of municipal waste collection. 47

4.2 Location of waste bins in map. 52

4.3 Plotting the sub-routes in map. 55

4.4 Distance evolution versus iteration increment for NSGA-II. 56

4.5 Distance evolution versus iteration increment for SPEA2. 56

List of Tables

4.1 The corresponding notation 48
4.2 Parameters of NSGA-II and SPEA2 53
4.3 Best solution obtained with iteration=100 54
4.4 Result of NSGA-II and SPEA2 with various iterations 55
4.5 Result of NSGA-II and SPEA2 with various P_c and P_m 57

General introduction

Solid waste collection is one of the most complex logistical problems facing any municipality. In recent years, increased fuel prices, operational costs, environmental and health concerns, and a growing regulatory burden have led to waste collection companies, municipal and private alike, to improve their collection methods. According to various studies, in some cases, transportation costs may be reach more than 70% of the total cost of waste collection. Besides, the trucks picking up and carrying the collected waste are considered to be very notorious in CO 2 emissions to the atmosphere. Moreover, several municipalities do not take into consideration that some particular kinds of waste must they collected and transported with the least delay possible such as medical or hazardous waste. Therefore, any small improvements in the collection routes will play a significant role in saving municipal expenditure and the companies' bottom lines. Waste management problem is a multi-faceted problem composed of many stages including generation, collection and transportation, transformation, treatment, and final disposal. From the viewpoints of citizens or decision-makers in municipalities, the collection stage of waste receives relatively greater attention to be seriously taken into consideration as the ignorance of such stage results in disastrous impacts.

Waste collection has been addressed by many researchers around the world and is well-known in optimization literature as a vehicle routing problem (VRP). All studies focus on identifying the efficient and optimal with minimum distance possible, and reduce emission, with the fewest possible vehicles (homogeneous or not homogeneous), maximizing

productivity. Many studies used different variants of VRP according to the case studied or corporate needs such as VRPTW(VRP with time windows), GVRP (green VRP), EVRP (Emissions Vehicle Routing Problem), PDVRP (Pick-Up and Delivery Vehicle Routing Problem), WCVRP (Waste Collection Vehicle Routing Problem), etc. On the other hand, many approaches and algorithms used for solving VRP for MSW collection, but as a problem classed under NP-Hard, meta-heuristics approaches the most popular in this field such as ant colony optimization (ACO), genetic algorithms (GAs), and particle swarm optimization(PSO), Tabu search (TS), etc. In addition, a number of software packages used in MSW optimization. The very commonly used is ArcGIS software which uses geospatial information to improve real-time collection routes. Recently, smart waste bins are gradually being introduced. Akhtar et al[4], established a waste collection model considering the application of smart bins. The research results show that the improved model and algorithm perform better in path optimization. Some researches apply smart waste bins in real cases, including glass bins in Geneva, Switzerland [5], and residential waste bins in the UAE[6]. Maurizio et al[7], regarded the amount of waste generated as a random variable. Real-time data were acquired through modern traceable devices such as RFID, GPRS, and GPS, and corresponding rules are set to determine whether waste bins should be collected.

This thesis focuses on operational decisions at the stage of collection, and transportation. The main purpose is to obtain an optimal collection plan by combining optimization methods and background of smart bins to know the level of bins in real-time. As known, VRP subject to several types of constraints. The most studied are capacity constraints and time constraints, so a capacitated vehicle routing problem (CVRP) model is developed for MSW collection in Ouargla city by taking into consideration the bin level. The model has developed using a multi-objective evolutionary algorithm (MOEA) called NSGA-II, this meta-heuristic method gave good results in reducing the distance traveled and the number of vehicles used.

The study project will be structured as follows: The first chapter presents the basic concepts of algorithm complexity, combinatorial optimization, and multi-objective optimization, in particular VRP, next, we present a state of the art of waste collection VRPs. The second chapter is an overview of the literature review on methods of solving the vehicle routing problem. The third chapter is devoted to multi-objective evolutionary algorithm (MOEAs). In the fourth chapter, we describe the implementation environment and review the obtained results, Finally, the general conclusion.

Chapter 1

Waste collection and vehicle routing problems

Introduction

In this chapter, we will address the vehicle routing problem (VRP), The Vehicle Routing Problem (VRP) aims the optimization of routes in order to deliver or pick up certain products from defined stations or costumers. Dantzig and Ramser introduce the problem in 1959 by optimizing the routes of a petrol delivery company. Later on, the two mathematicians Clarke and Wright developed the algorithm and introduced an improved model in 1964. During the following years, multiple variants of VRP was developed in order to meet the demanding needs of the costumers includes the waste collection problem wich is the main object of this thesis, but before talking about VRP and waste collection problem in this chapter we will define some fundamental such as algorithm complexity, combinatorial optimization (CO), and Multi-objective optimization problems (MOPs).

1.1 Preliminaries

1.1.1 The complexity of an algorithm

In the context of operational research to be able to compare the effectiveness of the different algorithms that can be used in the resolution process. Two main criteria are generally considered when the operational researcher studies the efficiency of an algorithm: the computation time, and the memory space necessary for algorithm execution. In the majority of problems, computation time is the main reference for measuring the performance of an algorithm. In the following, we will explain the algorithm complexity theory:

Definition 1.1. An algorithm is called polynomial If the number of elementary operations necessary to solve the size of the problem is a polynomial function, then the algorithm is considered to be efficient if, and only if, it is polynomial.

In the previous definition, the term "elementary operations" it looks vague. It should be understood as additions, multiplications, comparisons, etc. generally, everything that the processor can do in a fixed time.

Definition 1.2. A decision problem is in class **P** (**P**olynomial) if its resolution on a machine is deterministic in polynomial time compared to the size of the data, it is a problem of complexity $O(n^k)$.

–A decision problem is in class **NP** (**N**on-deterministic **P**olynomial) if its resolution on a machine is Non-deterministic in polynomial time.

–A problem is in class **NP-complete** if its resolution in polynomial time requires the resolution in polynomial time of any problem NP.

So NP-complete problems are more complicated than NP problems.

Any problem that can be solved with an algorithm of polynomial complexity is considered “easy”. Any problem that can be solved with an algorithm of exponential (Non-polynomial) complexity, or worse than exponential is considered “hard”, This kind of problem known as NP-Hard problems.Figure 1.1 illustrate the relations between complexity classes.

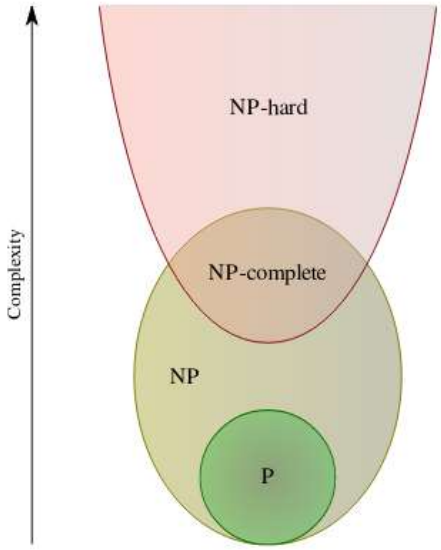


Figure 1.1: Relations between complexity classes.

1.1.2 Combinatorial Optimization

Solving the VRP and especially its base form the TSP is a typical combinatorial optimization task. In practice, combinatorial optimization is one of the more difficult forms of mathematical optimization. It is characterized by a finite but often huge set of elements, with the goal being to find an optimal element regarding a cost function. Formally a combinatorial optimization problem.

Definition 1.3. Combinatorial Optimization is defined from a finite set S and an application $f : S \rightarrow \mathbf{R}$. that assigns to every solution point $\hat{s} \in S$ such that: $f(\hat{s}) = Min_{s \in S}\,[f(s)]$.

Combinatorial optimization problems can be written mathematically as:

$$(P)\begin{cases} optimize \;\; f(x) \\ Subject \;\; to \;\; x \in S \end{cases} \tag{1.1}$$

where $f : \mathbb{R}^n \rightarrow \mathbb{R}$ is the objective function,The subset $S \subseteq \mathbb{R}^n$ (it can be also $S \subseteq \{0,1\}^n$)represents feasible set or all allowable choices for x, such that $x = (x_1, x_2, ..., x_n)^T$ is the vector of variables of dimension n.

1.1.2.1 Basic concepts of optimization

Objective function : objective function is a mathematical formula that describes the goal of optimization to be achieved based on a set of constraints, those constraints can be capacity, resources, availability, etc. The objective function takes the form of *F(x),*generally *F(x)* is a vector*: F(x) =[f 1 (x), f 2 (x), ..., f k (x)].* It called also cost function *(minimization),* and fitness function.

Decision vector : Is the vector which belongs to a feasible set $S \subseteq R^n$, it is noted $x = [x_1, x_2, ..., x_n]^T$ with n is the number of variables or dimension of the problem and x_k the variable on dimension K

Constraints : The constraints of a problem determine limits on how to maximize or to minimize some variables, those constraints can be equality or inequality relations often noted $g_i(x)$,with $i = 1, ..., q,$ number of constraints.

Research space : represents all possible values of the variables that satisfy the problem's constraints.

Objective space : is the images set of the search space, determined by all possible values of objective functions.

Global minimum (point) : A point $\bar{x} \in S$ is a global minimum if and only if: $f(\bar{x}) \leq f(x)$ for all $x \in S$

Local minimum (point) : A point $\bar{x} \in S$ is a local minimum if and only if, there exists

a neighborhood N of $\bar{x}$ such that*:* $f(\bar{x}) \leq f(x)$ for all $x \in N \cap S$ [8].

1.1.3 Multi-Objective Optimization Problem (MOP)

1.1.3.1 Defintion of multi-objective optimization problem

The multi-objective nature of the problem is given by the existence of contradictory objectives signifying that the improvement of one objective leads to the deterioration of another. This conflict between objectives is often encountered when seeking to obtain the best possible performance for a low cost. The more complex and efficient a structure, the higher the cost. Multi-objective optimization then consists of finding all the solutions which correspond to the best compromises between different objectives, for example, maximize profit and minimize cost.

1.1.3.2 General formulation of a multi-objective optimization

A multi-objective optimization problem characterized by a set of objective functions to maximize or minimize a set of numeric values by a certain number of constraints to be satisfied. Unlike the single-objective case for which a single objective function to be optimized, multi-objective optimization consists of finding the best compromises different conflicting goals.Multi-objective optimization problems can be formulated Mathematically as follows:

$$(P) \begin{cases} min/max \;\; f_1(x), f_2(x), \ldots, f_n(x) \\ subject\ to \;\; x \in \Omega \end{cases} \tag{1.2}$$

Where x is solution, n is the number of objective functions, Ω is the set of feasible solutions,$fn(x)$ is the nth objective function, and min/max is combined object operations.

The MOP's evaluation function, $f : \Omega \to \Lambda$, maps decision variables $(x = x_1, \ldots, x_n)$ to vectors $(y = a_1, \ldots, a_k)$. This situation is represented in Figure 1.2 for the case $n = 2$,$m =$

0, and $k = 3$. This mapping may or may not be onto some region of objective function space, dependent upon the functions and constraints composing the particular MOP[1].

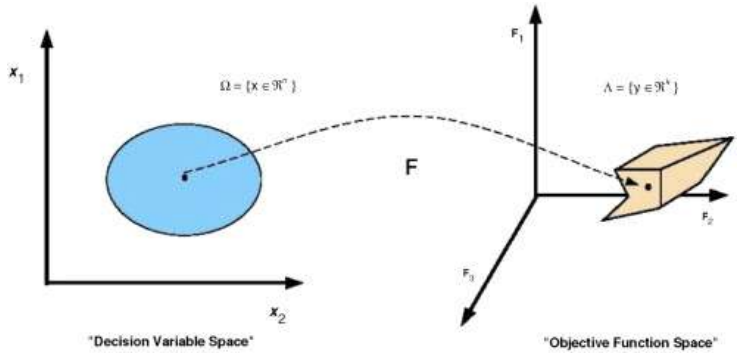


Figure 1.2: Representation of the decision variable space and the corresponding objective space.[1]

1.1.3.3 Pareto optimization

Pareto optimum:

The Italian mathematician, Vilfredo Pareto, formulated the concept of the theory of the elites, This concept applies to the case of MOP, such that it cannot improve a criterion without deteriorating at least one of the other criteria. This equilibrium called Pareto optimum. a point x is said to be Pareto-optimal if it is not dominated by any other point belonging to S. These points are also called non-inferior or non-dominated solutions[9].

The concept of Dominance:

A solution $x^{(i)}$ dominates another solution $x^{(j)}$ if the following conditions are satisfied:

1) $z_k(x^{(i)}) \leq z_k(x^{(j)})\ \forall k \in \{1, \ldots, K\}$;

2) $\exists k \in \{1, \ldots, K\}$ such that $z_k(x^{(i)}) < z_k(x^{(j)})$.

3) If $x^{(i)}$ dominates $x^{(j)}$, $x^{(i)} \prec x^{(j)}$

Figure 1.3 shows a particular case of the Pareto front in the presence of two objective functions.

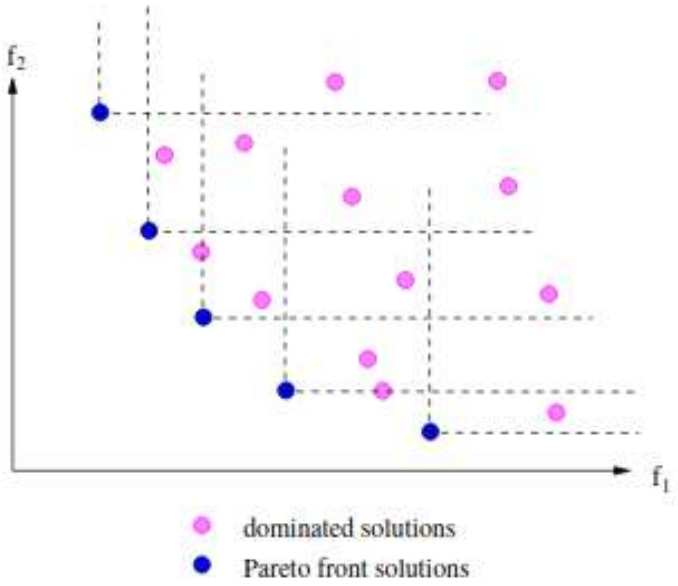


Figure 1.3: The Pareto front of a set of solutions in a two objective space.

Strict and weak dominance :

Definition 1.4. A solution $x^{(i)}$ dominates a solution $x^{(j)}$ strictly if and only if:

$z_k(x^{(i)}) < z_k(x^{(j)}) \; \forall$ k $\in \{1, \ldots, K\}$, This relation is noted $x^{(i)} \prec x^{(j)}$

Definition 1.5. A solution $x^{(i)}$ dominates a solution weakly $x^{(j)}$ if and only if:

$z_k(x^{(i)}) < z_k(x^{(j)}) \; \forall$ k $\in \{1, \ldots, K\}$, This relation is noted $x^{(i)} \preceq x^{(j)}$

Pareto frontier :

For a finite set of solutions, all solutions can be compared two by two according to the principle of dominance, and we can deduce which solution dominates the other. at the end, we obtain a set where none of the solutions dominates other, this set is called the set of non-dominated solutions or else the set of Pareto-optimal solutions[9].

If the set P represents the whole of the search space S, the set of non-dominated solutions P' is called pareto-optimal set in the decision space or pareto front in the objective space.

Definition 1.6. Pareto-optimal set is the set of non-dominated solutions of the feasible search space S.

We say that a solution x ∈ S is efficient, if there is no other solution y ∈ S such that the vector $f(y)$ dominates the vector $f(x)$.

1.2 The vehicle routing problem (VRP)

1.2.1 The traveling salesman problem (TSP)

The traveling salesman problem is the most famous and studied problem in combinatorial optimization field,in this problem, a traveling salesman must visit several cities or customers (nodes) passing once and only once each of them, and minimizing the total distance traveled. More formally, a TSP is modeled in the form of a graph where the vertices represent the cities to visit, [Dhaenensand al., 2002]. The weight associated with each edge represents the cost of the connection between the two cities and generally corresponds to the distance which separates them. The objective is to find a Hamiltonian cycle, e.g. a passing cycle once and only once by all the vertices of the graph, and of minimum length. As an optimization problem, the TSP is an NP-Hard problem. Indeed, in its symmetrical version, e.g. in the case where the associated graph is not oriented, the total number of possible solutions is $\frac{(n-1)!}{2}$ where n is the number of cities. With factorial complexity, the efficient resolution of the TSP, therefore, requires the use of specialized heuristics or even meta-heuristics. Indeed, the exact methods remain limited to the problems of small size.

1.2.2 Definition of Vehicle Routing Problem

The vehicle routing problem (VRP) naturally appears as a central problem in the fields of transportation, is very widespread in distribution and logistics, As we saw above it is a generalization of the TSP, which means that it is a problem classified as an NP-Hard problem, it is obtained when K vehicles of capacity Q are available at the node-depot to satisfy the demands of the nodes-client, Each vehicle must, therefore, carry out a feasible tour, e.g. leave the depot, visit once customers whose sum of requests

does not exceed the capacity Q, before returning to the depot. Each customer must be served by a single vehicle that fully satisfies their request. The objective of VRP is to serve the customers by minimizing one or more criteria related to the cost of delivery of goods, for example, minimize the total distance of tour or total traveling times to visit all customers, while respecting the constraint of vehicle capacity, In other words, the capacity of goods delivered in one tour should not exceed The capacity of the vehicle which ensures it.Practical applications, illustrated among others in Golden et al (2008), are often large and currently need to be heuristically resolved[10]. Figure 1.4 illustrates a solution of classical VRP with 5 vehicles and 25 costumer.

The Application of vehicle routing problem is suitable and useful in many sectors such as:Product Delivery and Pickup, Transportation, Trip planning, Waste Collection, Goods distribution, Food distribution, Mail and Package delivery, etc.

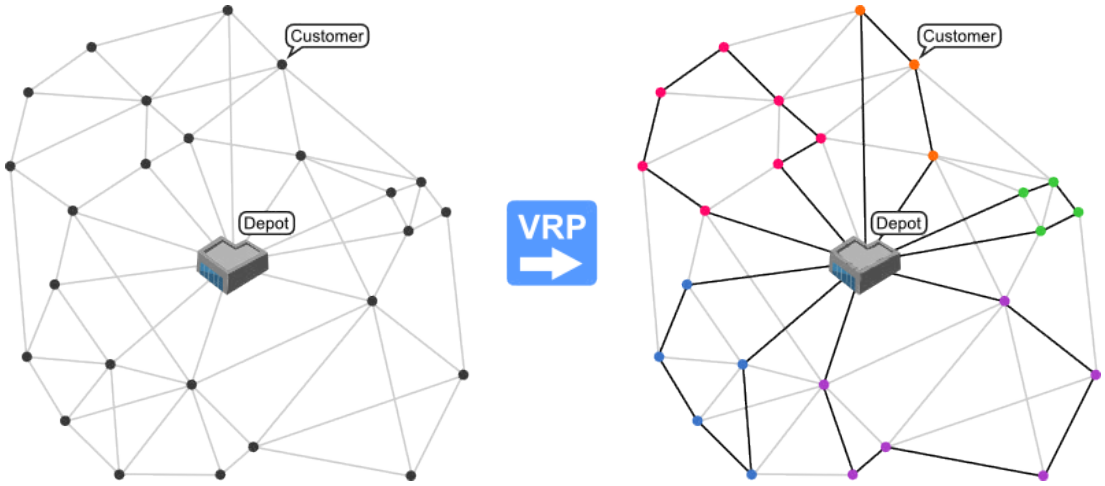


Figure 1.4: An instance of a VRP (left) and its solution (right).

1.2.3 Vehicle Routing Problem's Formulation

The VRP is defined on a graph $G = (V, A)$, where $V = \{u_0, u_1, u_2, \ldots, u_n\}$ represents the set of vertices, that is to say, depot and customers to visit, and $A = \{u_i, u_j\}$ such that $i \neq j$ and v_i, $v_j \in V$ represents the possible set of arcs. The point v_0 represents the depot which is the point of starting and arrival of all routes, a distance d_{ij} is associated with

each arc $(i, j) \in A$, these distances are symmetrical, that is to say, that $d_{ij} = d_{ji} \forall i, j \in A$. The other constants of the problem :

- n number of customer (or summits).
- m number of vehicles.
- Q vehicle capacity.
- q_i customer request i.
- c_{ij} the cost of the edge between the vertices i and j (distance or travel time).

The decision variables of the problem x_{ijk} defined as follow:

$$x_{ijk} \begin{cases} 1 & if\ (i,j)\ is\ visited\ by\ the\ vehicle\ k. \\ 0 & otherwise \end{cases}$$

$$Minimize \sum_{i=1}^{n} \sum_{j=1}^{n} c_{ij} \sum_{k=1}^{m} x_{ijk} \tag{1.3}$$

subject to the following constraints:

$$\sum_{i=1}^{n} \sum_{k=1}^{m} x_{ijk}\ \forall 1 \leq j \leq n \tag{1.4}$$

$$\sum_{j=1}^{n} \sum_{k=1}^{m} x_{ijk}\ \forall 1 \leq i \leq n \tag{1.5}$$

$$\sum_{i=1}^{n} \sum_{l=1}^{n} x_{ijk} = \sum_{l=1}^{n} \sum_{j=1}^{n} x_{ijk} \tag{1.6}$$

$$\sum_{j=1}^{n} x_{0jk}\ \forall 1 \leq k \leq m \tag{1.7}$$

$$\sum_{i=1}^{n} x_{i0k}\ \forall 1 \leq k \leq m \tag{1.8}$$

$$\sum_{i=1}^{n}\sum_{j=1}^{n} x_{ijk} \leq Q \; \forall 1 \leq k \leq m \tag{1.9}$$

$$x_{ijk} \in 0,1 \; \forall 0 \leq i,j \leq n; 1 \leq k \leq m \tag{1.10}$$

The equation (1.2):the objective of the optimization problem must minimize the sum of the costs of all the routes.Constraint (1.3) and (1.4) require each customer to be served only once. Constraint (1.5) ensure that flow conservation. Constraint (1.6) ensure that each tour begins and ends at the depot. Constraint (1.7) the capacity constraints.Constraint (3.8) binary constraints on decision variables x_{ijk}.

1.2.4 Real-life Routing Problems

Routing problems are of concern in real life whenever things need to be transported from one place to another. For example, garbage collection companies need to plan the routes for collecting garbage in the urban. Bus companies need to plan the time and routes for buses and drivers. These real-life rout-ing problems usually include complications that are not considered by the basic CVRP :

- Planning horizon : In real life, routes are planed for a given planning horizon. This planning horizon can consist of multiple periods.

- Customer : In the basic VRP, each customer has a demand, In more complicated problems, the customers may have requirements on the service time and/or the vehicle type. There could also be different types of services, for example, pickup service, delivery service, or pickup-and-delivery service.

- Depot: There can be multiple depots in a large distribution network.These depots may serve different purposes, such as warehousing or cross-docking, to reduce the total cost in the supply chain.

- Vehicle : The vehicles used for delivery can be different in capacities and sizes. There are usually a limited number of vehicles available in real-life planning. A vehicle may be used in multiple tours instead of a single trip in a routing plan. In the problem with

multiple depots, each vehicle maybe associated to a base depot. The vehicle must start from and end at its base depot

- Uncertainty : There can be uncertainties in the route planning. For example, the locations and/or the demands of customers are unknown at the beginning but revealed over time when the vehicles have already been sent out to carry out tasks[11].

1.2.5 VRP Variants

The most studied and common VRP variants are the capacitated VRP (CVRP), where capacity constraints exist. The objective is to minimize the total cost (reducing the number of routes, their length or travel time), serving all customers. Capacity is, in fact, an issue that all real routing problems have to deal; therefore, the following variations are also CVRP variants:

1.2.5.1 VRP with Time Windows (VRPTW)

the VRP with Time Windows (VRPTW), assumes that deliveries to a given customer (node) $v \in V$ must occur in a certain time interval $[w_i^a, w_i^b]$, which varies from customer to customer[12].

1.2.5.2 Multiple Depot VRP (MDVRP)

A company may have several depots from which it can serve its customers. If the customers are clustered around depots, then the distribution problem should be modeled as a set of independent VRPs. However, if the customers and the depots are intermingled then a Multi-Depot Vehicle Routing Problem should be solved. An MDVRP requires the assignment of customers to depots. A fleet of vehicles is based at each depot. Each vehicle originates from one depot, service the customers assigned to that depot, and return to the same depot[13].

1.2.5.3 VRP with Pick-Up and Delivering

The Vehicle Routing Problem with Pick-up and Delivering (VRPPD) is a VRP in which the possibility that customers return some commodities is contemplated. So in VRPPD it's needed to take into account that the goods that customers return to the deliver vehicle must fit into it. This restriction make the planning problem more difficult and can lead to bad utilization of the vehicles capacities, increased travel distances or a need for more vehicles[13].

1.2.5.4 Periodic Vehicle Routing Problem (PVRP)

This problem takes into account several planning days, unlike the CVRP, with customers that require service on multiple days during the planning period. In order to find the set of minimum routes for each day.

There are many other variants of VRP, but until now, all variants considered a homogeneous fleet, however, in real life, a company may have a fleet with different vehicle types, with distinct capacities and costs. So far, it has also been considered that in one route products would either be collected or delivered, yet some customer can be associated with two quantities, representing the demand of goods to be delivered and other the ones to be picked up at its location, adding more challenges to the problem and to capacity management.

1.3 Waste collection as a vehicle routing problem

1.3.1 VRP for collection

Essentially, the VRP for collection is dealing with the same type of constraints as in a delivery problem when constructing vehicle routes. Thus, this problem also attempts to determine the number of vehicles needed to serve the customers as well as the routes

that will minimize the total distance traveled by the vehicles. However, the vehicle for the collection problem is empty when it starts from the depot, whereas the vehicle for the delivery problem begins its route loaded with customers' goods that need to be delivered. In the collection, problem vehicles will collect goods from a set of customers and return to the depot at the end of the working day[14].

1.3.2 VRP for waste collection

Dealing with a waste collection problem is different from the collection problem. There is an additional constraint that needs to be considered in solving this problem. Instead of returning to the depot to unload the collected goods(or waste), in a waste collection problem vehicles need to be emptied at a disposal facility before continuing collecting waste from other customers. Thus, multiple tours to the disposal facility occur in this problem before the vehicles return to the depot empty, with zero waste. A complication in the problem arises when more than one disposal facility is involved. Here we need to determine the right time to empty the vehicles as well as to choose the best disposal facility that respects distance constraints. For example, When the vehicle reaches its maximum size, it must visit a disposal facility..Figure 3.1 illustrates a VRP for waste collection with two routes, one depot, and three disposals facility[14]

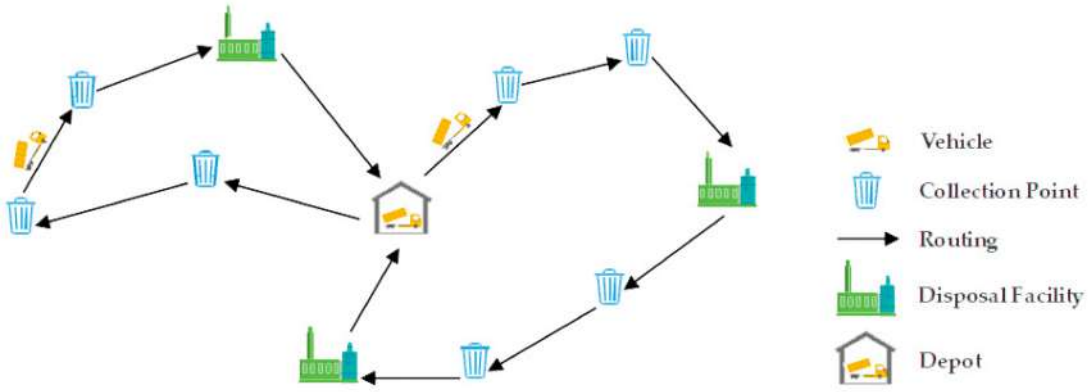


Figure 1.5: Examples of two vehicle routes for a waste collection problem[2].

1.3.3 Waste collection vehicle routing problem state of the art

Various models designed for solving the waste collection vehicle routing problem are based on the classic VRP, with some special considerations: Green Vehicle Routing Problem (GVRP), Pollution Routing Problem (PRP), Emissions Vehicle Routing Problem (EVRP), and Routing and Scheduling in Time-Dependent Environment (RS-TDE), where they look for reduction of polluting emissions to the environment, especially carbon dioxide, as a consequence of the minimization of the total distance, fuel consumption, operative costs, or the selection of uncongested routes. Another model is the Routing of Hazardous Materials (RHM) that minimizes the risk to the population and environment caused by hazardous material transportation. Other models that have been created are the Waste Collection Vehicle Routing Problem (WCVRP), the Multi-Modal Vehicle Routing Problem (MMVRP), the Pick-Up and Delivery Vehicle Routing Problem (PDVRP), and the Energy Routing Problems (ERP). In many of these models, the objectives presented are not in conflict, because by minimizing the total distance or routing time the fuel consumption is also reduced and with this the amount of polluting emissions[15].

Conclusion

The vehicle routing problem several cases is a multi-objective optimization problem also classed under NP-Hard combinatorial problems. Waste collection can be dealing as VRP for collect or pick up, there are various models of VRP designed for solving waste collection problem according to the case studied and need of corporate, in the last chapter we will explain in detail our case studied. Chapter 2 will be custom to literature review on methods of solving the vehicle routing problem.

Chapter 2

A literature review on methods of solving the vehicles routing problem

Introduction

In this chapter, we provide an overview of the methods of solving the vehicle routing problem. Our goal is not to detail how these methods work, but rather to gain an overview of their basic principle and their classifications.Like other combinatorial optimization problems, the vehicle routing problem has been studied and solved by exact methods, specific heuristics, and meta-heuristics. These three families make up the rational general classification of the methods of resolution of VRP.

2.1 Exacts Methods

The main principle of the exact methods consists generally of enumerates all candidate solutions in the search space, and eliminate the impossible solutions. Among the exact methods. A wide variety of exact methods have been devised for solving MOPs in general and, in particular, the VRP problems, which continue to attract the attention of researchers. Among the most recurrent achievements we cite: Linear Programming,

Dynamic Programming, Branch and Bound Approaches.

2.1.1 Dynamic Programming

It's a bottom-up method, We usually start with the smallest sub-problems and work our way up to the increasingly difficult sub-problems. It is used for problems that satisfy Bellman's principle of optimality: "In an optimal sequence (of decisions or choices), each subsequence must also be optimal". An example of this type of problem is the shortest path between two vertices of a graph. The basic idea is to avoid calculating the same thing twice, usually using an already calculated results table, filled up while solving the sub-problems, Christofides (1971) applied this principle for the resolution of VRPS.

The strong point of this method is the fact that it avoids evaluating the same function twice, generally by using a table of results obtained, filled in as we solve the subproblems and this is how we save a considerable amount of time

2.1.2 Branch and bound

It is a technique that performs an in-depth search of the research tree in order to provide one or more optimal solutions from a set of potential solutions. At each step of the search corresponding to a node in the search tree, the algorithm uses a Bound function to calculate bound for the solutions set of the sub-tree taking its root at this node. In the first part of the resolution, this bound is initialized to a maximum value (minimization case). If this assessment is worse than the best solution found up to this level of research, the entire sub-tree can be sterilized. It is important to emphasize that the efficiency of the Branch and Bound algorithm depends closely on the calculation of the bound used. Christofides, Mingozzi, and Toth applied the separation and evaluation method to the VRVPTW-large at varying vehicle capacity constraints. The best-resolved instant is 53 customers and 08 vehicles. After solving the problem, the maximum number of vehicles

on a route is 15.

2.1.3 Advantages and disadvantages of exact methods

The exact methods guarantee the optimality of the solutions they provide. Indeed, the determination of the optimal solution is done by an exhaustive journey of the domain of feasible solutions to the problem. On the other hand, for CO problems, the size of the domain of feasible solutions to a problem increases exponentially. with its size, therefore the exhaustive path to this domain of feasible solutions is impossible from a certain size of the problem.

2.2 Heuristic and Meta-heuristic methods

The majority of Multi-objective optimization problems are part of the NP-Hard class, including the VRP discussed in the previous chapter. The complexity of the problems makes exact resolution methods inefficient, and given the frequency and recurrence of these problems in practice, researchers have been led to develop approximate methods to solve these problems. complex ones. Indeed, the approximate methods give appreciable solutions to problems of important complexity at a reasonable time.

2.2.1 Heuristic Methods

The heuristic term comes from the Greek verb heuristic that means ”to find”. A heuristic is an algorithm which aims to find a feasible solution, without guarantee of optimality, unlike exact methods, but those methods have exponential complexity, it may be better to use heuristics to calculate an approximate solution to a problem or also to speed up the exact resolution process, such as Branch and Bound. Generally, a heuristic is designed for a particular problem.We mainly distinguish three types of heuristics:

2.2.1.1 Route construction heuristics

Route construction heuristics were among the first experimental techniques of CVRP and still form the core of many software applications for various routing applications. These algorithms typically start from an empty solution and iteratively build route by inserting one or more customers into each iteration, until all customers are routed. The construction algorithms are also divided into sequential and parallel, depending on the number of routers eligible for customer input. Sequential methods extend only one path at a time, while parallel methods take into account more than one route simultaneously. The route construction algorithms are fully defined by their three main components, namely the configuration standard, the selection criterion that defines which customers are selected for inclusion in the current iteration, and the insertion criterion for determining where to locate the chosen clients in the current routes[16].

2.2.1.2 Route improvement heuristics

Unlike the constructive heuristics, the improvement heuristics deal with complete solutions and attempt to improve them iteratively by applying a sequence of modifications to the solutions. These modifications are also called operators or moves and they are usually very simple. Since the improvement heuristics only accepts the modifications that improve the solution.

2.2.1.3 Two phases Heuristics

The two-phase methods are based on the separating of the VRP solution process into two separate sub problems: 1) clustering: determining a partition of customers into subsets, each corresponding to a route, and 2) routing: determining the sequence customers on each route. In a cluster-first-route-second method, customers are first grouped into clusters, and routes are then determined by appropriately sequencing customers in each

cluster. Different techniques have been proposed for the clustering phase, while the routing phase amounts to solving a TSP[16].

2.2.2 Meta-heuristics Methods

Meta-heuristic algorithms optimization methods designed according to the strategies laid out in a meta-heuristic framework, are always heuristic in nature. This fact distinguishes them from exact methods, that do come with a proof that the optimal solution will be found in a finite (although often prohibitively large) amount of time. Meta-heuristics are therefore developed specifically to find a solution that is "good enough" in a computing time that is "small enough". As a result, they are not subject to combinatorial explosion – the phenomenon where the computing time required to find the optimal solution of NP-Hard problems increases as an exponential function of the problem size[17].

During the last three decades, many meta-heuristics have been proposed for the VRP. These meta-heuristics include Tabu Search (TS), Simulated Annealing (SA),Iterated Local Search (ILS), Large Neighborhood Search (LNS),Ant colony optimization (ACO), Genetic algorithms (GA), Scatter Search (SS), and so on. These methods are categorized into two main groups:Single solution based meta-heuristics, and population based meta-heuristics.

2.2.2.1 Single solution based meta-heuristics

Among the most popular single solution based meta-heuristics, we cite the following algorithms :

- Tabu Search

Tabu search (TS) is a local search method introduced by Glover[18]. The principle of this method is to avoid being trapped in minimum local by performing movements from initial solution s (even infeasible solutions) in a set of local solutions S to subsets solution N(s)

belonging to the neighborhood S is generated by means of the evaluation function we retain the solution which improves the value of f, chosen from the set of neighboring solutions $N(s)$, If this neighborhood is too large, only a part $V(s) \subset N(s)$ will be examined. At each iteration, the tabu algorithm chooses the best non-tabu neighbor, even if it degrades the cost function. For this reason, it is said that the tabu search is an aggressive method[9].

- Simulated Annealing

Simulated Annealing (SA) is a global optimization algorithm inspired by the metallurgy, in this physical process, the material is heated and slowly cooled repeatedly until getting homogeneous shape, this increases the size of the molecules of the material. The heat excites the energy of the atoms, which allows them to move freely, and the slow cooling allows Reshaping the material and reduces their defects at low energy. The simulated annealing makes it possible to leave a local minimum by accepting with a certain probability a degradation of the function. Thus, the probability that a physical system goes from an energy level E1 to a level E2 is given by:

$$P = e^{\frac{-\Delta E}{k_b T}} \tag{2.1}$$

k_b is the Boltzmann constant, T is the temperature of the system.

As this formula shows, increasing the temperature leads to the possibility of observing an increase in energy, Therefore at the level of the simulated annealing: A decrease in the function will always be accepted, and An increase in the function will be accepted with a probability defined according to a formula of the previous [9].

2.2.2.2 Population based meta-heuristics

Two population based meta-heuristics approaches,have been widely used in literature for solving VRP and its variants:

- Ant Colony Optimization (ACO)

The phenomenon of pheromone communication has been the drive behind ant colony optimization (ACO) meta-heuristics algorithms. Artificial ants simulate the real-life ants which communicate their experience while optimizing their search for food in nature through trails of pheromone.

Candidate solutions to the optimization problem are constructed by individual ants by interactively adding solution components to initialized empty solution. A complete solution is generated by the ants using the two components: pheromone information which is the accumulated experience and heuristic information which is problem specific data. Which ants are allowed to modify the pheromone information and how they modify is governed by the update strategy. Usually, better solution components will receive higher amount of pheromone and will have a higher probability of being used by other ants in the subsequent iterations of algorithm[19].

- Genetic Algorithm (GA):

Genetic algorithm is part of Evolutionary Algorithms that we will mention with more detail in chapter 3, developed by John Holland and his collaborators in the 1960s and 1970s, are a model or abstraction of biological evolution based on Charles Darwin's theory of natural selection. Holland was the first to use crossover, recombination, mutation and selection in the study of adaptive and artificial systems. These genetic operators are the essential components of genetic algorithms as a problem-solving strategy. Since then, many variants of genetic algorithms have been developed and applied to a wide range of optimization problems, from graph coloring to pattern recognition, from discrete systems (such as the traveling salesman problem) to continuous systems (e.g., the efficient design of airfoil in aerospace engineering), and from financial markets to multi-objective engineering optimization.[20]

The process begins with creating a set of possible solutions randomly, this set is called the population, each population consists of individuals, and each individual represents a solution, An individual consists of a set of characteristics (or variables to be determined) are then used in gene sequences which will be combined with other genes to form chromosomes and afterwards individuals, To evaluate the individuals of a population we use an evaluation function (fitness function) this function is often a transformation of the objective function, The result provided by this function will make it possible to select or refuse an individual in order to keep only the individuals having the best cost according to the current population, The selection is followed by crossover then procreation. such that, every two individuals (parents) transmit part of their genetic heritage to their offspring. The child's genotype makes it more or less fit for the environment. If it has better fitness (or best cost), it will have a greater chance of procreating the future generation. This process keeps on iterating until to achieve the stop criterion, and at the end, a generation that has fittest individuals according to objective function $f(x)$, where x is the solution that the individual represents.

Finally, there is another classification Includes the learning mechanisms, which include artificial neural networks (ANN). ANN is inspired by the neurons in the brain and gradually adjusts a set of link weights until an acceptable solution is reached, the elastic net and the self-organizing map are examples of models used in VRP to generate a feasible solution[16]. ACO also can be classed under these types of methods.

Conclusion

We have seen, in this chapter, The different methods for solving the vehicle routing problem. We have exposed several exact and approximate resolution methods. The exact methods are only intended for small problems, and heuristic methods type, the drawback is that one cannot theoretically guarantee the good quality of the results (minimization of costs, time) for a well-defined study case. This proves to be slow and numerically painful. The genetic algorithm method has proven its robustness and high efficiency to solve large complicated problems such as VRP and its variants, as we noted the strength of GA in finding the best approximate solution in a reasonable time. Because GA is a part of the evolutionary algorithm, so in the next chapter, we will present the Multi-objective evolutionary algorithms (MOEAs) in more detail, and explain its mechanism.

Chapter 3

Evolutionary Algorithms for Multi-Objective Optimization

Introduction

Evolutionary algorithms prove to be general, powerful, and robust search mechanisms. In addition, AEs seem to be especially useful in multi-objective optimization because they are able to determine several optimal Pareto solutions in a single execution and can exploit the similarities of solutions by crossing. Some researchers suggest that multi-objective research and optimization may be problems where AEs do better than other blind search methods. These algorithms are called evolutionary multi-objective optimization algorithms(MOEAs).

3.1 Basic Principles of Evolutionary Algorithms

3.1.1 Definition

The term evolutionary algorithm (EA) stands for a class of stochastic optimization methods that simulate the process of natural evolution. The origins of EAs can be

traced back to the late 1950s, and since the 1970s several evolutionary methodologies have been proposed, mainly genetic algorithms, evolutionary programming, and evolution strategies[21]. All these approaches work on a set of candidate solutions. The concept of the evolutionary algorithms is quite simple given that it uses the two basic principles of Darwinism: selection and variation. The selection represents the competition for resources among living things. Some are better than others and are better able to survive and transmit their genetic material. In evolutionary algorithms, natural selection is simulated by a stochastic selection process. Each solution has a chance to recur a number of times, depending on their quality (fitness). So quality is assessed by evaluating individuals and assigning them a fitness value. The other principle, variation, imitates the natural ability to create new offspring and new hereditary traits through crossover and mutation.

3.1.2 Basic Terminology

In the following we present the initial principles of evolutionary algorithms, which are inspired by the mechanism of natural evolution:

1. A finite set of individuals is called population;
2. The objective function to optimize is called performance function or fitness function;
3. The calculation of an individual's performance is called evaluation;
4. Generation corresponds to a population in a certain iteration;
5. Evolution is an iterative process of finding optimal individuals;
6. The operators of variations are used to generate new individuals and are most often categorized into two types of operators: the crossing which consists in exchanging component parts (genes) between two or more individuals, and the mutation which consists in the modification of one or more genes d 'an individual ;

7. Selection is the process of choosing individuals, based on their performance;

8. Replacement is the process of forming a new population from parents and children[22].

3.1.3 Basic Structure

The design of evolutionary algorithm can be divided into follwing components:

3.1.3.1 Representation

One of the most important steps in the evolutionary algorithm is determining the representation that we will use to represent our solutions. Therefore, choosing a proper representation, having a proper definition of the mappings between the phenotype and genotype spaces is essential for the success of an EA[23].

3.1.3.2 Initialization

The evolutionary optimization process begins with creating an initial population P_0 that contains a random number of possible solutions to the problem in search space S.

3.1.3.3 Fitness function

The fitness function takes in the characteristics of an individual, and evaluates (calculates) its performance, then gives us a numerical result of how viable of a solution it is.

3.1.3.4 Parent Selection

The individuals who have the best fit are chosen for reproduction. However, care must be taken to avoid that one of the suitable solutions takes over the entire population within a few generations, as this leads to solutions that are close to each other in the solution space, thus leading to a loss of diversity. Maintaining good diversity in the population is extremely crucial for the success of an EA. This taking of the entire population by

an extremely suitable solution is known as premature convergence and is an undesirable condition in EA.in the following we briefly mention some selection methods :

- Tournament Selection
- Random Selection
- Truncation Selection
- Fitness Proportionate Selection (Roulette Wheel, Stochastic universal sampling).
- Ranking Selection(Linear Ranking,Non-linear Ranking)

3.1.3.5 Crossover

crossover operators resembles the reproductive mechanism in nature. Thus, along with mutation operators, they are collectively referred to as reproductive operators. In general, a crossover operator aims to combine two individuals to form a new individual. It tries to divide an individual into parts and then put those parts together into a new individual[23].in addition to the classic crossover operators shown in Figure 2.1. There exist a lot of other crossovers like Partially Mapped Crossover (PMX), Order based crossover (OX2), Shuffle Crossover, Ring Crossover, etc.

3.1.3.6 Mutation

mutation operators simulates the mechanism of mutation in which parts of a genome undergo changes of a random nature. Thus, as a typical modeling practice, a mutation operator changes parts of an individual's genome. On the other hand, mutations can be thought of as an exploration mechanism to balance the exploitative power of crossover operators[23]. Among the most commonly used mutation operators :Bit-flip Mutation(used for binary encoded), swap mutation, and displacement mutation.

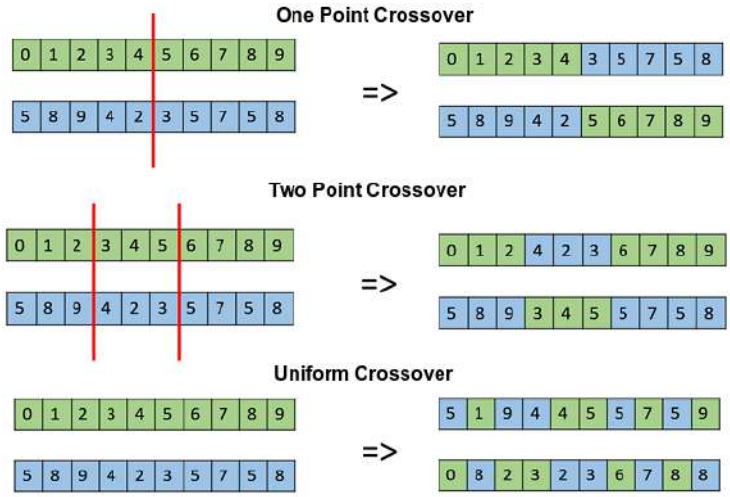


Figure 3.1: The classic crossover operators.

Figure 3.2: Example of mutation operators.

3.1.3.7 Survival Selection

Survival selection aims to select a subset of good individuals from a population, where the goodness of the individual is proportional to its fitness in most cases. Thus, the mechanism of survival selection is somehow similar to the mechanism of parent selection, most of the parent selection mechanisms can be reapplied in survival selection. For example, a proportional selection of fitness can be applied as survival [23].

3.1.3.8 Termination condition

The termination condition refers to the condition at which an evolutionary algorithm should end. The number of generations is often chosen as the termination measurement: an evolutionary algorithm terminates when a certain number of generations has been reached (e.g. 1000 generations). The number of fitness function evaluations is also adopted in

some cases. CPU time is also adopted. Nonetheless, convergence is not guaranteed. Thus, the fitness improvement of each generation has adopted as another condition for termination[23].

Figure 3.1 and Algorithm 1 describe the basic structure of an MOEA.

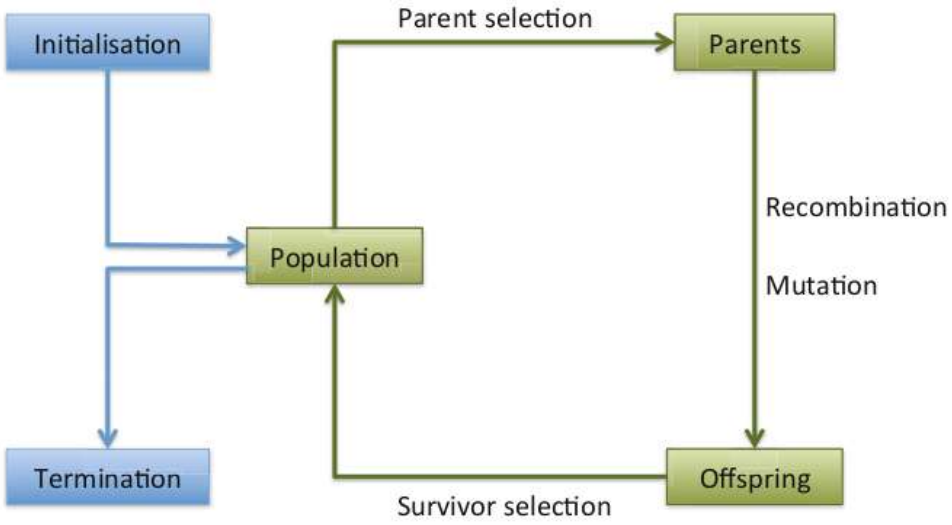


Figure 3.3: The general scheme of an Evolutionary Algorithm as a flow-chart.

Algorithm 1 Pseudo-code of Evolutionary Algorithm

1: $t \leftarrow 0$
2: Initialize $P(t)$;
3: **while** not termination condition **do**
4: Evaluate the objective vector F for each individual of $P(t)$;
5: Assign a fitness for each individual of $P(t)$
6: Select from $P(t)$ a set of parents $P'(t)$ preferring the best
7: Recombine the individuals of $P'(t)$ to obtain the offspring $P''(t)$
8: Mutate individuals into $P(t)$
9: Combine $P'(t)$ and $P''(t)$ and select the best individuals to have $P(t+1)$
10: $t \leftarrow t+1$
11: **end while**

3.2 Key Issues in Multi-Objective Search

As mentioned earlier in this chapter, EA manipulates a population of solutions instead of a single solution. This property makes it possible to be well suited to solving multi-objective optimization problems where the optimum is represented by a set of points (Pareto set) and not by a single solution, but two major problems posed when using an EA for solving multi-objective optimization problem:

1. How to achieve Fitness value assignment and selection, respectively, to guide process research towards the Pareto-optimal set.

2. How to keep a diverse population to avoid premature convergence, and achieve a non-dominated and well-distributed set[24].

In the following, a categorization of general techniques which deal with these issues is presented :

3.2.1 Fitness Assignment and Selection

In contrast to single-objective optimization, where objective function and fitness function are often identical, both fitness assignment and selection must allow for several objectives with MOPs. In general, one can distinguish MOEAs where the objectives are considered separately, approaches that are based on the classical aggregation techniques, and methods which make direct use of the concept of Pareto. dominance[24].

3.2.1.1 Selection by Switching Objectives

Instead of combining the objectives into a single fitness value, this class of MOEAs exchanges the objectives during the selection phase. Each time an individual is selected for reproduction A different objective allows us to decide which member of the population will be copied into the mating pool. As a result, steps 2 and 3 of the previous Algorithm (General EA) are often integrated or performed alternately[24].

3.2.1.2 Aggregation Selection with Parameter Variation

Other MOEA implementations based on traditional techniques for generating Pareto surface compromises. With those methods, the objectives are aggregated into a single parameterized objective function, however, there is no change in the parameters of this function for different EA but varies during the same run[24].

3.2.1.3 Pareto-based Selection

The concept of calculating an individual's fitness on the basis of Pareto dominance was first suggested by Goldberg in 1989. He presented a "revolutionary10-line sketch" of an iterative ranking procedure: First, all non-dominated individuals are assigned rank one and temporarily removed from the population. Then, the next nondominated individuals are assigned rank two and so forth. Finally, the rank of an individual determines its fitness value. Remarkable here is the fact that fitness is related to the whole population, while with other aggregation techniques an individual's raw fitness value is calculated independently of other individuals[24].

3.2.2 Population Diversity

In order to approximate the Pareto-optimal set in a single optimization run, evolutionary optimizers have to perform a multimodal search where multiple, widely different solutions are to be found. Therefore, maintaining a diverse population is crucial for the efficacy of an MOEA. Unfortunately, a simple (elitist) EA tends to converge towards a single solution and often loses solutions[24]. To overcome this drawback, several techniques aimed at maintaining diversity in the population have been proposed, the most frequently used are briefly summarized here:

3.2.2.1 Fitness sharing

Fitness sharing(Goldberg and Richardson 1987), aims at promoting the formulation and maintenance of stable sub-populations (niches).It is based on the idea that individuals in a particular niche have to share the available resources. The more individuals are located in the neighborhood of a certain individual, the more its fitness value is degraded[24]. The adaptation function of each individual (x_i) is degraded by a niche counter $m(x_i)$ calculated for this same individual. This niche counter calculates the degree of similarity that an individual has with the rest of the population:

$$m(x_i) = \sum_{i \in pop} sh(d(x_i, x_j)) \tag{3.1}$$

- $d(x_i, x_j)$: distance between i and j

- sh_d : the decreasing function of $d(i, j)$, such that : $sh(0) = 1$, and $sh(d \geq \sigma_{share})$. The most commonly used function $sh(d)$ is the triangular function defined as follows :

$$sh(d) = \begin{cases} 1 - \frac{d}{\sigma_{share}} \; if \; d < \sigma_{share} \\ 0 \quad otherwise \end{cases} \tag{3.2}$$

- σ_{share}: niche radius, fixed in most cases by the user according to the minimum separation distance desired between the different peaks[9]

3.2.2.2 Crowding distance

The crowding distance factor gives us an idea of how crowded are the closest neighbors of a given individual, in objective function space. This measure estimates the perimeter of the cuboid formed by using the nearest neighbors as the vertices.

3.2.3 Elitism

Elitism is the mechanism designed to prevent the loss of the best solutions found in research due to stochastic effects. This concept plays a major role in MOEAs. In multi-objective optimization, the implementation of elitism is more complex than in single-objective optimization. As a result of limited memory resources, if more non-dominated solutions arise than can be stored, then which solutions should be discarded? Therefore, the elitist strategy adopted determines whether fitness is broadly convergent or not. Currently, we can distinguish mainly two approaches to implement elitism. One of them is to combine the old and the new population, and then use a selection that preserves the best solutions in the next generation. The other approach is to maintain an external set of individuals called archives that store the non-dominated solutions found during the research process.

3.3 Parallel Evolutionary Algorithms

3.3.1 The Island Model

The most straight forward implementation of island MOEAs runs a number of MOEA populations independently, each trying to obtain the complete Pareto-front and every rate generations migration takes place. Beyond this simple strategy, many researchers believe that a 'divide and conquer' approach on multi-objective optimization problems could be more successful because individual sub-populations could specialize in certain areas of the Pareto-front and thus be more efficient[25].The first work in this area dates back to the 1960s with Bossert who was probably the first to propose an evolutionary algorithm with multiple populations in order to improve the quality of solutions in an optimization problem. Other works that were subsequently proposed differ according to the choices made on the many parameters to be managed with this model, for example, the number and size of islands, the frequency of migration, the number, and destination of migrants,

and finally the method used to select which individual will migrate. Moreover, this is the reason why this type of parallelization remains difficult to control[22].

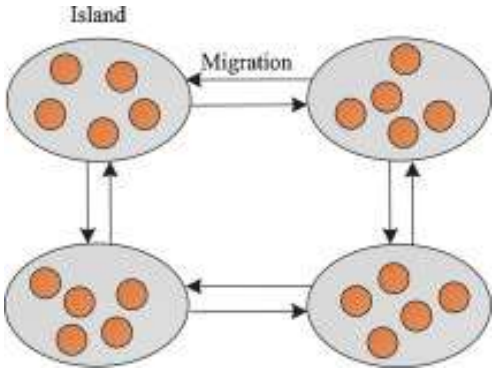


Figure 3.4: Illustration of island EAs[3].

3.3.2 The Master-Slave Model

In the "master-slave" model, which is the simplest model to implement, the master process distributes the evaluation of the objective function on the various slave nodes and performs all the steps of the evolutionary algorithm (selection, kingship, mutation). Communication between individuals only takes place after slave disputes return the evaluation assigned to them. It is therefore algorithmically identical to a sequential evolutionary algorithm. The master-slave model has been widely used in the literature. The earliest work dates back to the 1970s with Bethke who was the first to describe a parallel implementation of an AE. Subsequently, Grefenstette proposed several prototypes of the parallel AEs representing several variants of the master-slave models, knowing that the overall optimization cost with this model involves two stages, that of the evaluation of the different children and the communication time between the different slave nodes and the master process, more recent work in the literature has focused on the problem of the high communication cost which can affect the efficiency of the algorithm. In this model, the acceleration in computing time is proven to be linear relative to the number of processors, up to a certain limit (when communication times become more important than evaluation

time). Note that in the case of real applications where the evaluation of the objective function is very expensive, communication times are often neglected[22].

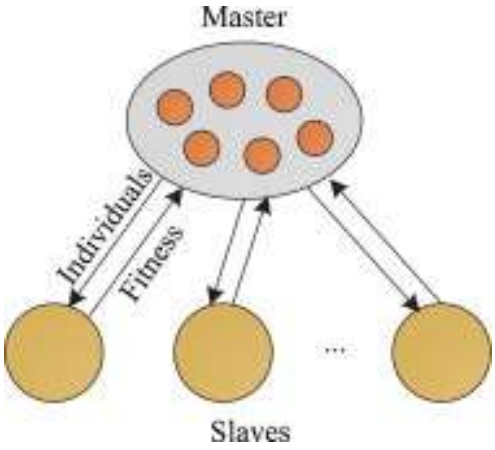


Figure 3.5: Illustration of master-slave EAs[3].

3.3.3 The cellular model

Was developed for massively parallel computers, which provide numerous processors with a local but fast communication network. It uses both properties, an individual is evaluated and mutated on a single processor, and selection and crossover is limited to few neighbors often given by the network topology[25].However, this last model remains little studied in the literature by comparing it with the first two models.

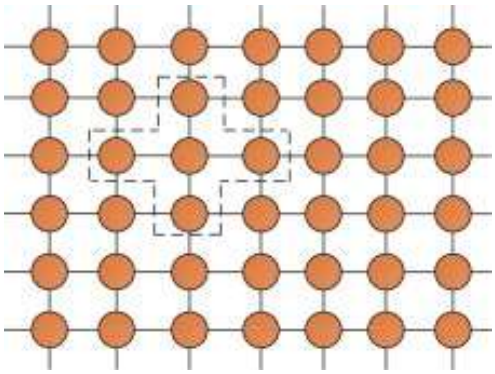


Figure 3.6: Illustration of cellular EAs[3].

3.4 Structures of Various MOEAs

3.4.1 Multi-Objective Genetic Algorithm (MOGA)

Carlos M. Fonseca and Peter J[26]. Fleming proposed a variation of Goldberg's technique called "Multi-Objective Genetic Algorithm" (MOGA), in which the rank of a certain individual corresponds to the number of chromosomes in the current population by which it is dominated. Consider, for example, an individual xi at generation is dominated by $p(t)i$ individuals in the current generation; thus, an individual is assigned a rank by the following rule: rank(xi,t)=1+p(t)i[27].

3.4.2 Niched Pareto Genetic Algorithm (NPGA):

Horn and Nafpliotis [28] combined tournament selection and Pareto dominance. In this method, two individuals are selected at random in order to compare them to a subset of the population. The non-dominated individual is chosen as the parent. On the other hand, if the two individuals are dominated or non-dominated, then the winner is chosen using a fitness sharing feature[27].

3.4.3 Nondominated Sorting Genetic Algorithm (NSGA)

Srinivas and Deb implemented Goldberg's idea in a simpler way. Non-dominated Sorting Genetic Algorithm (NSGA) classifies the population in different layers or non-dominated fronts compared to non-dominance. The first front (the best ranking) is made up of the non-dominated individuals of the current population. The second front is the set of non-dominated individuals to the exclusion of the first rank. In general, each front is calculated only with unclassified individuals in the population. Deb et al.[29] later proposed a new version of the algorithm called NSGA-II. This algorithm improves the efficiency of the original NSGA by reducing the number of times the population needs to be ranked and incorporates an elitist selection system, as well as a Crowding comparison

operator[27].

3.4.4 Strength Pareto Evolutionary Algorithm (SPEA):

Strength Pareto Evolutionary Algorithm (SPEA) was developed by Zitzler and Thiele[30], as a means to combine the most successful techniques of different MOEAs. SPEA uses the individuals stored in the archives to classify individuals in the current population. For each individual, in the archive, a value s is calculated called strength which is equal to the number of individuals in the population who are dominated by the individual in the corresponding archive. The fitness of each individual x is calculated by adding s (strength) of all members of the archive that dominate x. This scheme attempts to guide research towards the Pareto front and, at the same time, preserves the diversity of non-dominated solutions[27].

3.5 Strength Pareto Evolutionary Algorithm (SPEA2)

The improved Strength Pareto Evolutionary Approach (SPEA2) is chosen to perform the control system optimization resulting in the final analysis and comparison. SPEA is an extension of the Genetic Algorithm for multiple objective optimization problems. SPEA2 has an external archive consisting of the previously found non-dominated solutions. It is updated after each generation and for each solution a strength value is computed. An archive of the non-dominated set is maintained separately from the population of candidate solutions used in the evolutionary process, providing a form of elitism. Due to potential weaknesses of SPEA, the improved version SPEA2 has better fitness assignment scheme, more precise guidance of the search and a new archive truncation methods. To avoid situations where population members dominated by the same members of the archive have the same fitness value, SPEA2 takes into account both the number of dominating and dominated solutions in computing the raw fitness of a solution. The objective of the

algorithm is to locate and maintain a front of non-dominated solutions (set of Pareto optimal solutions). This is achieved by using evolutionary process to explore the search space, and a selection process that uses a combination of the degree to which a candidate solution is dominated and an estimation of density of the Pareto front as an assigned fitness. An archive of the non-dominated set iskept separate from the population of candidate solutions used in the evolutionary process, which represents a kind of elitism[30].

3.6 A fast and elitist multiobjective genetic algorithm (NSGA-II)

In this section, we will present the different characteristics of NSGA-II, that we will use in the rest of this thesis. The NSGA-II algorithm was proposed by Deb et al[29]. It incorporates a selection operator, based on a calculation of the distance from overcrowding (or crowding) detailed in Chapter II, this calculation estimates the density of each individual in the population. Compared to NSGA, NSGA-II obtains better results on all the instances presented in the work of K. Deb. The following points make this algorithm one of the most used today :

- It uses an elitist principle ,e.g. the elites of a population are given the opportunity to be carried to the next generation.
- It uses an explicit diversity preserving mechanism (Crowding distance).
- It emphasizes the non-dominated solutions.[9].

3.6.1 Procedure of NSGA-II

1. perform a non-dominated sorting in the combination (R_t) of parent and offspring populations, P_t and Q_t, respectively. Classify them by fronts, e.g.they are sorted according to an ascending level of non-domination (F1,F2, F3. . .).

2. Fill the new population P_{t+1} with fronts according to the front raking in which solutions belonging to the least dominated fronts are selected first. If there are more solutions in a front that can fit in the new population, then sort the solution in the front with respect to the crowding distance (a distance related to the density of solutions around each solution). The greater crowding distances are preferred.

3. Create an offspring population Q_{t+1} from P_{t+1} using crowded tournament selection (comparing by front-ranking, if equal then by crowding distance), crossover, and mutation operators[31].

A schema of the procedure can be seen in Figure 3.5:

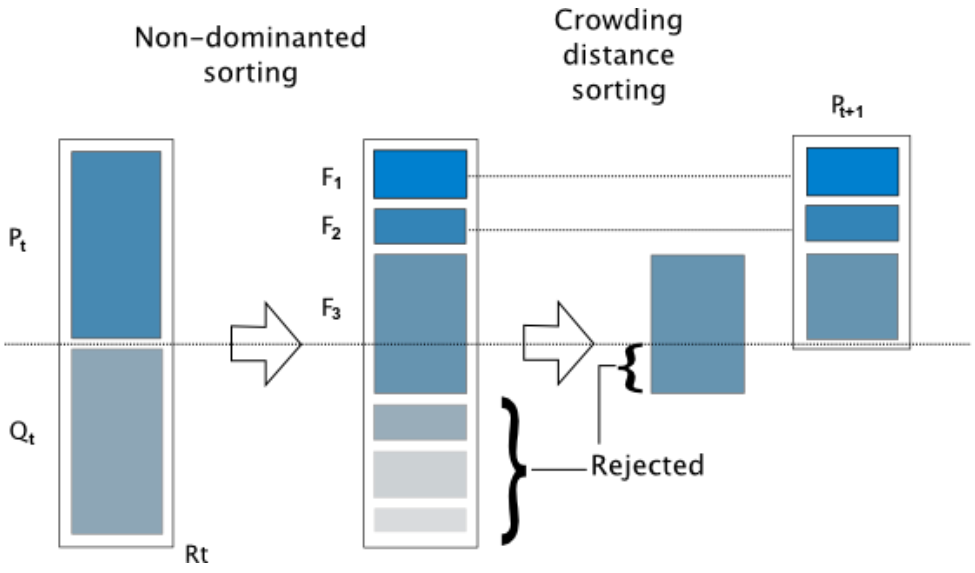


Figure 3.7: Schematic representation of NSGA-II working principle.

3.6.2 Crowding distance calculating

The Crowding sort of the last front points that could be fully accommodated is achieved in descending order of their Crowding distance values, and the points at the top of the ordered list are chosen. The Crowding distance d_i from point i is a measure of the objective space around i which is not occupied by any other solution in the population. Here we simply calculate this quantity d_i by estimating the perimeter of the cuboid (Figure

3.6) formed using the nearest neighbors in objective space (We call this the Crowding. distance)[27].

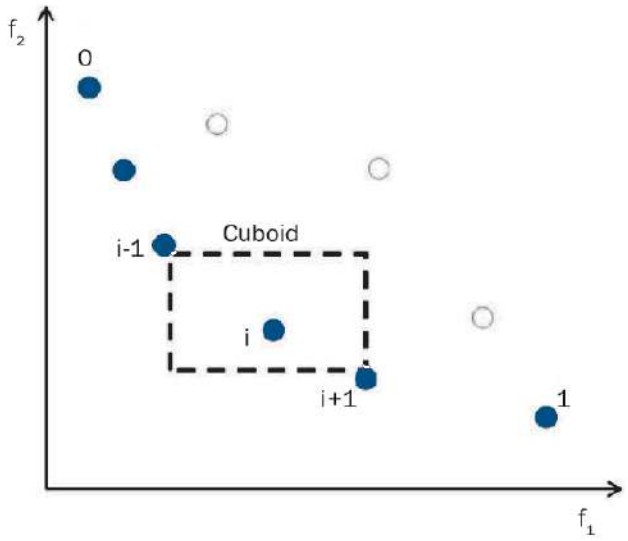


Figure 3.8: Crowding distance calculation.

Conclusion

Multi-objective evolutionary algorithms (MOEAs) the most general and efficient techniques for determining the optimal Pareto set with a good diversity of points, but the performance estimation of any MOEA approach cannot be made before its implementation and analysis of the results obtained. The following chapter is dedicated to the implementation of the NSGA-II and SPEA2 algorithm designed for the solution of our problem which is also will describe in the same chapter.

Chapter 4

Implementation and experimental results

Introduction

The aim of this chapter is to apply the NSGA-II and SPEA2 algorithms which defined in chapter 3 to solve a model of CVRP that we will present in this chapter, we will model the problem of CVRP of our case studied by specifying the set of parameters, then we will explain the approach, NSGA-II for the resolution of the CVRP, by detailing the different stages which constitute it, then a brief description of the programming environment which is Linux as well as the description of the programming language used which is Python.

4.1 Municipal Waste collection problem in Ouargla city

4.1.1 Definition

In this study, we consider the optimization VRP of MSWs collection problem in Ouargla city the capital of Ouargla Province located in the southern est of Algeria. Ouargla one of

the largest municipalities in Algeria with a total area of $2,887\ km^2$. In addition to other services, the municipality provides MSWs collection service to all regions in the city (old city, new city, and villages).

According to the information given by the municipal of Ouargla and Public establishment for the management of technical landfill centers: The daily amount of solid wastes (residential) collected in all regions of Ouargla city is about 150 tons per day.In order to increase the quality of waste collection service, the municipality divides Ouargla city into 16 regions and provides 41 vehicles with maximum capacity 34 quintals (3400 kg) and maximum volume 7 m^3, each vehicle carrying between 18 to 28 quintals of waste in one tour, as each truck makes two tours per day, the total tours for the whole fleet equal 110 tours per day. The total number available of garbage containers reaches 240 containers with 1.1 m^3 of volume, distributed more in urban areas, and in the new city.The citizens can also request a waste collection service on some special events or when the waste is accumulated in an irregular manner.

In addition to the municipal vehicles, there are private-sector vehicles (no information available about their number). as well as the number of small garbage bins. Moreover, no information about some particular kinds of waste, and how they transporting such as chemical waste, hospital waste, hazardous waste, and waste close to gas stations and fuel stations, etc.So we will take into consideration household (domestic) waste only.

4.1.2 Problem description

The current collection system relies only on the experience of vehicle drivers who have a good knowledge of the geography of the area, which has resulted in high operating costs, and frequent visits to some garbage containers while skipping some others. The problem involves how to obtain optimum routes for each vehicle in order to reduce the total distance and total overall costs including the number of vehicles.

4.1.2.1 Constraints

Waste bins that have a high level loaded characterized as high priority bins which should be collected as soon as possible.As shown in Figure 4.1, the vehicles are located at the depot and start their tours toward the allocated waste bins. When the waste collection vehicle is fully loaded, it must go to the disposal facility to unload the collected waste, before starting the second tour, when the collection task completed the vehicle must go back to the depot at the end of the workday.

From the above description, we make the following assumptions :

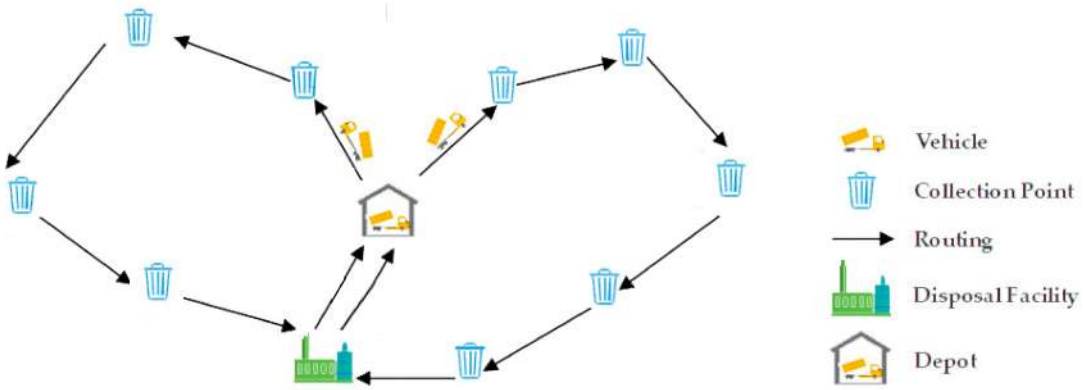


Figure 4.1: Simplified routing diagram of municipal waste collection.

1. Each waste bin is only collected by one vehicle once.
2. There is one disposal center.
3. There is one depot center.
4. The vehicles must depart from the depot and go to the disposal center twice before go back to the depot when the task ends.
5. Vehicle fleet is homogenous The location of the disposal center and each waste bin are known.

6. The information (including location) sent by the citizens considered as waste bin that has a high level.

4.1.2.2 Parameters and Variables

The problem formulation considers the following elements:

Table 4.1: The corresponding notation

Notation	Units	Description
B		Set of waste bins ($B = b_1, b_2, \ldots, b_i, \ldots, b_n$)
V		Set of vehicle ($V = v_1, v_2, \ldots, v_k, \ldots, v_m$)
Q_p	kg	Maximum load capacity of vehicle
L		Waste landfill center
TD	km	Total distance of all vehicles
q_i	kg	Collected waste at waste bin j
d_{ij}	kg	Distance between waste bin i and j
δ_i		The level of bin i sent by sensor
x_{ij}^k		If the vehicle k visits bin j from i, x_{ij}^k is 1. Otherwise, x_{ij}^k is 0

4.1.2.3 Model formulation

The objective function can be written as follows:

$$Minimize \sum_{i=1}^{n} \sum_{j=1}^{n} \sum_{k=1}^{m} d_{ij} x_{ij}^k \tag{4.1}$$

subject to the following constraints:

$$\sum_{i=1}^{n} \sum_{k=1}^{m} x_{ij}^k = 1 \; \forall 1 \leq j \leq n \tag{4.2}$$

$$\sum_{j=1}^{n} \sum_{k=1}^{m} x_{ij}^k = 1 \; \forall 1 \leq i \leq n \tag{4.3}$$

$$\sum_{j=1}^{n+1} x_{0jk} \; \forall 1 \leq k \leq m \tag{4.4}$$

$$\sum_{i=1}^{n+1}\sum_{k=1}^{m} x_{i0}^{k} \; \forall 1 \leq k \leq m \tag{4.5}$$

$$\sum_{i=1}^{n}\sum_{j=1}^{n} x_{ijk} \leq Q \; \forall 1 \leq k \leq m \tag{4.6}$$

$$x_{ij}^{k} \in \{0,1\} \; \forall 0 \leq i,j \leq n; 1 \leq k \leq m \tag{4.7}$$

Equation (4.1) is the objective function that calculates the total minimum distance traveled by all municipal vehicles. The constraints in equations (4.2) and (4.3), respectively, ensure that each bin must be visited once by one vehicle only and leave it After get lifted its load by that vehicle. The constraint (4.4) ensures that all vehicles start from the depot with an empty load. The constraint (4.5) ensures that all vehicles return to the depot eventually. The constraint (4.6) ensures that the load of each vehicle must not exceed the maximum capacity. The constraint (4.7) is the binary constraint on the decision variables.

4.2 Implementation of NSGA-II and SPEA2

4.2.1 Implementation Environment

The implementation was done using Python language. Python is an interpreted, high-level, and oriented-object programming language designed to be easy to read and simple to implement. We have chosen version Python 3.7 one of the most stable versions, It works with most Python libraries without bugs. The script has written in Jupiter notebook, Jupiter is a browser-based tool for interactive authoring of documents, as well as is part of the Anaconda environment which include the most popular packages (libraries) of scientific-computing.

Among packages,we used on our implementation :

- **Distributed Evolutionary Algorithms in Python (DEAP):** It is the most important package in our work, It was chosen for the following reasons. It contains

most of the basic functions required by evolutionary algorithms so that it can easily construct various of both single and multi-objective evolutionary algorithms and execute them using multiple processors. It can be used with an abundance of other Python packages for data processing as well as other machine learning techniques.

- **openrouteservice :**openrouteservice is a routing API library, it is a free alternative to Google APIs services
- **Geopy :** to calculate the geodesic distance between two points, that allow as to create the distance matrix.
- **Folium:** to visualize data that's been manipulated in Python on an interactive map.
- **Pandas:** for data manipulation and analysis, it offers data structures and array manipulation operations.
- **NumPy:** intended to handle multidimensional matrices or arrays as well as mathematical functions operating on these arrays.
- **Matplotlib :** is a Python library for plotting and visualizing data in the form of graphs.

All computational experiments were carried out on a Lenovo ThinkPad x250 PC with Intel Core i5-5300U (2.30 GHz x 4) CPU, and 8 GB of memory.

4.2.2 Solution Representation

4.2.2.1 Encoding and Decoding

The solution uses an array in the form of chromosomes. A chromosome is a sequence of nodes, each chromosome contains a set of Nodes (bins), to be visited by an associated vehicle. This type of encoding is called permutation list encoding.
For example, there is a depot (0), 10 smart waste bins (1 to 10), and landfill(11):

Node(i)	0	4	10	3	9	7	2	5	6	1	8	11	0

After decoding obtain an initial solution with three sub-path, each sub-path associated to a vehicle (v_1 to v_3):

Sub-path1	v_1	0	4	10	3	9	11	0	
Sub-path2	v_2	0	7	2	5	11	0		
Sub-path3	v_3	0	6	1	8	11	0		

4.2.2.2 Crossover operator

When the initial population generated, we must proceed to the crossover phase which ensures the recombination of parental genes to obtain a new generation (offspring). To do this, we choose a two-point crossover.

4.2.2.3 Mutation operator

With mutation probability, the offspring mutated at specific positions in the chromosome, which makes them different from its parents (previous generation). This phase also promotes the idea of population diversity.

4.2.2.4 Fitness evaluation

Since the path has satisfied the vehicle capacity constraint in the course of individual decoding the fitness function in this implementation has expressed as :

$$f(x) = \sum_{i=1}^{n} \sum_{j=1}^{n} d_{ij} \tag{4.8}$$

The individual that achieves the lowest distance, is the individual which has the best fitness.

4.2.2.5 Terminating Condition

Finally, the stop criterion achieved when the number of generations reaches to maximum

4.3 Analysis of results

4.3.1 Data presentation

In our work, we have collected previously the coordinates of 97 waste bins, as well as the public landfill and municipal depot. The waste bins located in six popular neighborhoods, four urban neighborhoods, and downtown. In addition to geospatial information, we have added estimated levels of bins. The data represented in an interactive map(Figure 4.2).

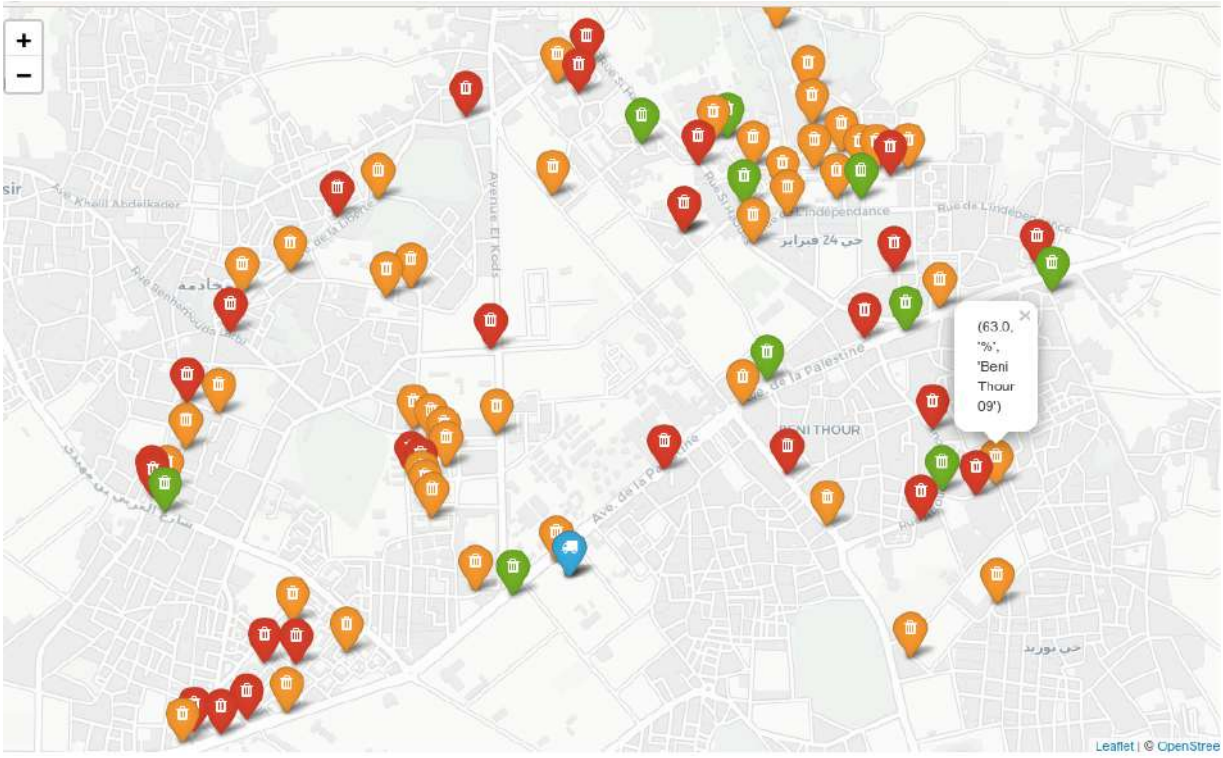


Figure 4.2: Location of waste bins in map.

The waste bins colored according to their level (green for level< 35, orange if level in range [35,75], orange if level > 75).

4.3.2 Parameters setting

We will implement the algorithm in two ways. First, without taking into consideration the levels of waste bins. Second, we will implement the algorithm with smart waste bins approach.

4.3.2.1 Parameters of the algorithms

The parameters of NSGA-II and SPEA2 have represented in Table 4.2

Table 4.2: Parameters of NSGA-II and SPEA2

Parameter	NSGA-II and SPEA2
Population size	100
Max generation	50/100/200/250
Crossover operator	Two-point crossover
Crossover probability (P_c)	0.6/0.7/0.8
Mutation probability (P_m)	0.4/0.3/0.2

The parameters are examined separately, e.g. changing the value for the given parameter while keeping the values of the rest of parameters at a constant level. During the investigation of the effect of each parameter is examined separately. First we have test the NSGA-II and SPEA2 with ordinary waste bins, then we have fixed the threshold of collection on 35%, in other words, If the waste bin less than 35%, the vehicle skips the bin. Here we wish to emphasize that the bins level that was chosen randomly does not affect the process of the first approach, because trucks weigh waste bins when collecting. The difference between the two approaches only lies in the prior knowledge of the bins level and chose some level for taking a decision if the bin visit or note.

4.3.2.2 Capacity constraints

Due to the difference in the density of the materials (plastic, cardboard, glass, organic materials, etc.), we cannot calculate the weight of waste in a container by volume. Some

studies show that each container uses only one type of waste. We'll consider the volume of waste as its weight, and the maximum volume of the truck is the maximum capacity.We have calculates the capacity of waste in each bin through following relation:

$$q_i = \frac{\delta_i * CB}{100} \tag{4.9}$$

CB is the volume of waste bin($CB = 1.1m^3$),q_i and δ_i, mentioned in Table 4.1, $Q_p = 7m^3$.

4.3.3 Results and analysis

Table 4.3 and Figure 4.3,represent one of the best solutions we have obtained with minimum distance 134.16 km.

Table 4.3: Best solution obtained with iteration=100

Vehicle	Route
v_1	$0 \to 89 \to 43 \to 10 \to 17 \to 83 \to 1 \to 36 \to 18 \to 60 \to 11 \to 13 \to 98 \to 0$
v_2	$0 \to 56 \to 28 \to 3 \to 5 \to 62 \to 69 \to 22 \to 88 \to 92 \to 37 \to 98 \to 0$
v_3	$0 \to 23 \to 25 \to 4 \to 40 \to 95 \to 80 \to 63 \to 74 \to 98 \to 0$
v_4	$0 \to 87 \to 85 \to 84 \to 29 \to 48 \to 66 \to 94 \to 38 \to 76 \to 90 \to 86 \to 98 \to 0$
v_5	$0 \to 68 \to 57 \to 39 \to 52 \to 70 \to 64 \to 79 \to 35 \to 98 \to 0$
v_6	$0 \to 19 \to 16 \to 96 \to 59 \to 42 \to 41 \to 55 \to 67 \to 98 \to 0$
v_7	$0 \to 77 \to 58 \to 61 \to 91 \to 30 \to 20 \to 47 \to 14 \to 6 \to 98 \to 0$
v_8	$0 \to 9 \to 49 \to 34 \to 8 \to 53 \to 26 \to 12 \to 33 \to 15 \to 98 \to 0$
v_9	$0 \to 46 \to 51 \to 31 \to 54 \to 32 \to 27 \to 7 \to 2 \to 72 \to 98 \to 0$
v_{10}	$0 \to 75 \to 21 \to 73 \to 24 \to 93 \to 98 \to 0$
Distance(km)	134.16

As shown in Fig. 4.3, the right side represents the obtained sub-routes network plotted by colors, the left side shows that all vehicles visit the landfill before returning to the depots.

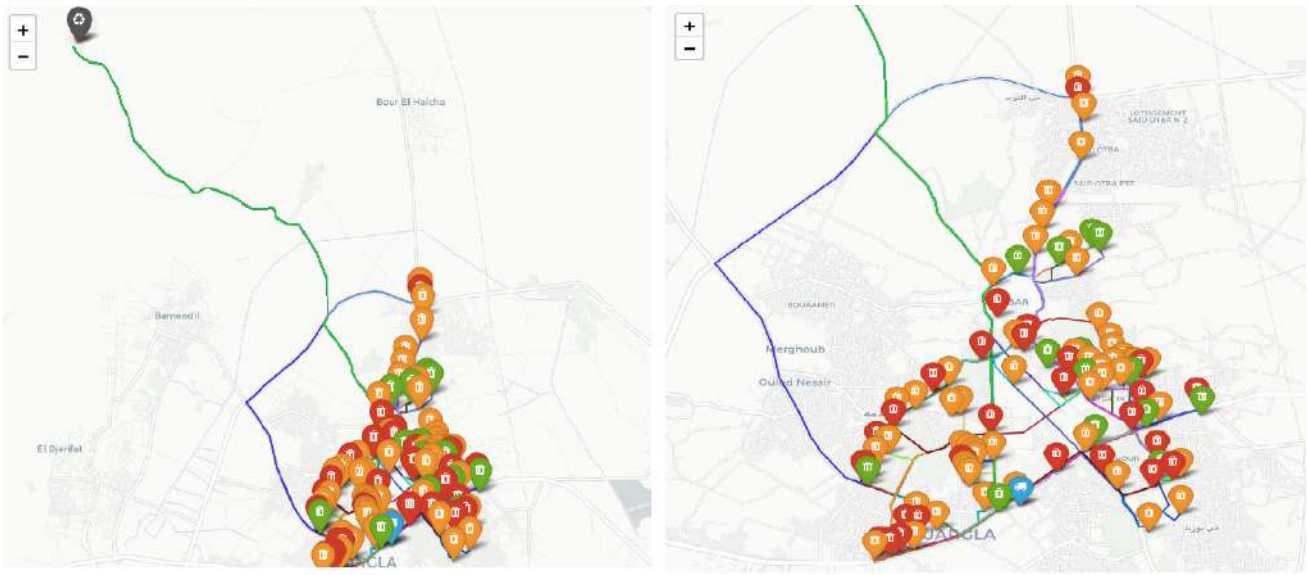


Figure 4.3: Plotting the sub-routes in map.

4.3.3.1 Generation number influence

We have fixed the crossover probability (P_c) in 0.7 and mutation probability (P_m) in 0.3 , and we implemented the two algorithms with various iterations.The results represented in Table 4.5 :

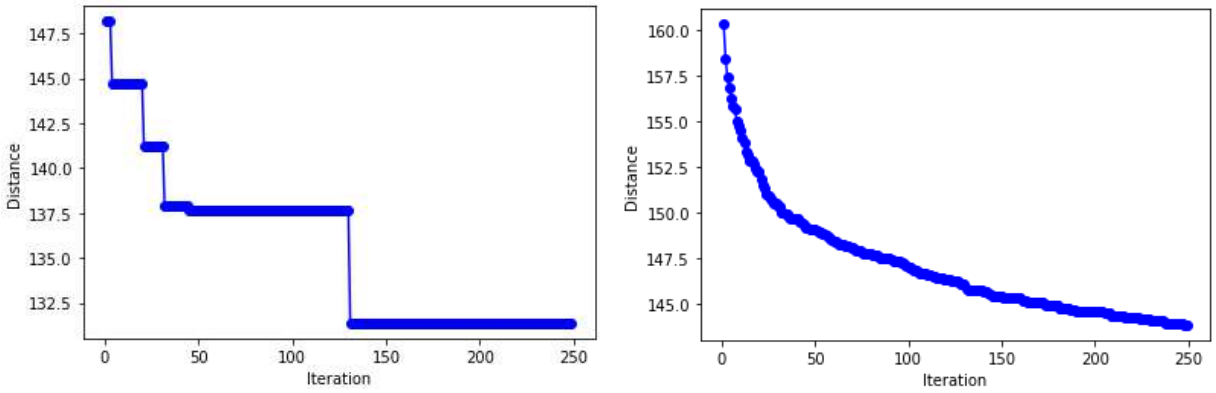
Table 4.4: Result of NSGA-II and SPEA2 with various iterations

		NSGA-II				SPEA2			
	Iteration	Best	Worst	Average	CPU times	Best	Worst	Average	CPU times
Ordinary bins	50	140.44	151.96	148.82	5min 49s	138.54	151.73	148.63	6min 35s
	100	134.16	149.15	146.24	11min 31s	138.36	148.64	145.28	12min 8s
	200	134.25	148.04	145.32	18min 57s	137.90	147.33	145.39	24min 37s
	250	131.36	146.22	143.82	28min 36s	131.46	145.72	143.32	28min 52s
Smart bins	50	119.76	130.87	127.65	8min 58s	117.68	131.24	128.13	9min 56s
	100	114.78	128.51	125.7	19min 33s	115.78	127.98	125.43	17min 56s
	200	114.06	126.81	124.11	36min 40s	112.40	126.38	123.87	35min 53s
	250	112.19	121.48	121.22	47min 4s	113.51	124.87	122.92	44min 28s

Table 4.5 shows the evolution of the optimization process while increasing the generation number. Both studied cases need more than 150 generations to find feasible solutions. We can see also in the same table that the execution time is increased when we implement the

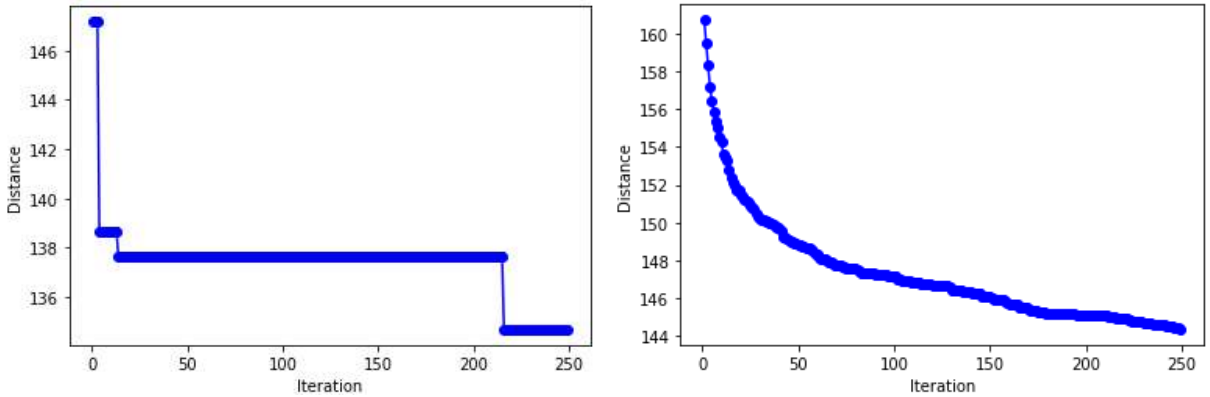
NSGA-II and SPEA2 with smart waste bins, for example for iteration 250 the time is 28 min for implementation NSGA-II only, and 47 min for NSGA-II with smart waste bins. This time increase is due to the fact that we changed the objective function by adding the waste bin level condition.

Figure 4.4, Figure 4.5 illustrate the evolution of minimize the distance according the increase of iteration number for NSGA-II and SPEA2 with number of generation=250 :



(a) Minimum distance (km). (b) Average distance (km).

Figure 4.4: Distance evolution versus iteration increment for NSGA-II.



(a) Minimum distance (km). (b) Average distance (km).

Figure 4.5: Distance evolution versus iteration increment for SPEA2.

4.3.3.2 Crossover probability and mutation probability Influence

we have fixed the number of generations in 200, and we implemented the two algorithms with different P_c and P_m.The results represented in Table 4.6 :

Table 4.5: Result of NSGA-II and SPEA2 with various P_c and P_m

			NSGA-II				SPEA2			
	P_c	P_m	Best	Worst	Average	CPU times	Best	Worst	Average	CPU times
Ordinary	0.8	0.2	132.11	143.80	142.82	25min 18s	135.14	146.08	144.78	30min 17s
bins	0.7	0.3	134.25	148.04	145.32	18min 57s	137.90	147.33	145.39	24min 37s
	0.6	0.4	129.60	145.24	143.38	23min 24s	133.48	144.05	142.49	27min 55s
Smart	0.8	0.2	115.15	127.34	124.86	43min 9s	116.11	127.64	124.42	47min 14s
bins	0.7	0.3	114.06	126.81	124.11	36min 40s	112.40	126.38	123.87	35min 53s
	0.6	0.4	117.76	125.79	123.14	36min 2s	117.23	125.68	122.94	34min 1s

Table 4.5 shows that best result we have obtained was with Pc=0.6 and Pm=0.4 for the two algorithms, but with smart bins the best result was with Pc=0.7 and Pm=0.3. We also note the convergence of NSGA-II and SPEA2 with preference for NSGA-II in execution time. In the case of smart bins, SPEA2 has little precedence, but in general we can say NSGA-II faster and efficient than SPEA2.

The lowest average distance is found with the number of generations 250. Therefore, increasing the number of generations improves the solution quality. Nevertheless, the execution time will increase. Mostly, NSGA-II and SPEA can obtain an optimal solution in a reasonable amount of time. For instance, when varying the crossover operators and fixing the other parameters, several times, for our model, a moderate population size (100) is enough to converge. Moreover, increasing the amount of crossover probability to its maximum value is not necessary to converge into optimal solutions. The same assertion is true for the mutation probability.

Conclusion

In this chapter, the general problem studied in this note is described and then we proposed a mathematical model for CVRP to solve the problem of municipal waste collection. In the second section, we have carried out a series of experiments in order to evaluate the performance of the variants of the MOEAs proposed and to judge the efficiency of the NSGA-II and SPEA2, and we end with a comparison the results of the NSGA-II and SPEA2 (with ordinary waste bins and smart waste bins) . We found changes in the number of generations have an evident impact on the obtained results for the two algorithms, also that the optimization in the total traveled distance and the number of vehicles is more efficient with smart waste bins.

General Conclusion

Planning a series of vehicle routes well is a challenging task in an effort to decrease collection and transportation costs, the negative effects of waste, and to ensure that all inhabitants live in a comfortable and healthful environment. Thus, this thesis proposed a model for waste collection and transportation with the minimized total comprehensive costs including total traveled distance and number of vehicles used. A CVRP model is proposed to solve the municipal waste collection in Ouargla city.

A meta-heuristic has chosen by applying the NSGA-II and SPEA2 algorithms, the efficiency of the NSGA-II is proved by combining it with the smart waste bins idea. The experimental results proved that smart waste bins can raise the efficiency of waste, as the distance was reduced obviously when we set certain conditions such as the condition of skipping the low-level bins.

We used real data such as coordinates, vehicle capacity,bins capacity, etc. except for waste bins level which created based on information according to the information given by the municipal of Ouargla based on the truck drivers experience. Although the results were encouraging we were unable to carry field trials, it would therefore be interesting to an extension of this work by developing a full smart system for municipal waste management.

Bibliography

[1] Carlos Coello, David Veldhuizen, and Gary Lamont. *Evolutionary Algorithms for Solving Multi-Objective Problems Second Edition*. 01 2007.

[2] Qingqing Qiao, Fengming Tao, Hailin Wu, Xuewei Yu, and Mengjun Zhang. Optimization of a capacitated vehicle routing problem for sustainable municipal solid waste collection management using the pso-ts algorithm. *International Journal of Environmental Research and Public Health*, 17:2163, 03 2020.

[3] Yue-Jiao Gong, Wei-neng Chen, Zhi-Hui Zhan, Jun Zhang, Yun Li, and Qingfu Zhang. Distributed evolutionary algorithms and their models: A survey of the state-of-the-art. *Applied Soft Computing*, 34, 05 2015.

[4] Mahmuda Akhtar, M. A. Hannan, Rawshan Begum, Hassan Basri, and Edgar Scavino. Backtracking search algorithm in cvrp models for efficient solid waste collection and route optimization. *Waste Management*, 61:117–128, 01 2017.

[5] Iliya Markov, Yousef Maknoon, Jean-François Cordeau, Sacha Varone, and Michel Bierlaire. Waste collection inventory routing with non-stationary stochastic demands. *Computers & Operations Research*, 01 2016.

[6] Mohamed Abdallah, Mohamad Adghim, Munjed Maraqa, and Elkhalifa Aldahab. Simulation and optimization of dynamic waste collection routes. *Waste Management & Research*, 37:0734242X1983315, 03 2019.

[7] Maurizio Faccio, Alessandro Persona, and Giorgia Zanin. Waste collection multi objective model with real time traceability data. *Waste management (New York, N.Y.)*, 31:2391–405, 08 2011.

[8] M.Houam Yahia. *Commande Multi-Objectifs En Utilisant Les Inégalités Matricielles Linéaires (LMIs) Et Les Algorithmes Génétiques*. Université Mohamed Khider – Biskra, 2013.

[9] BAAZIZ ADLANE. *Optimisation Multi-objectif des Chaînes Logistiques par les Algorithmes Génétiques*. Université des Sciences et de la Technologie Houari Boumediene, 2015.

[10] Sahbi ISMAIL, Francois Legras, and Gilles Coppin. Synthèse du problème de routage de véhicules. 01 2011.

[11] Min Wen. *Rich Vehicle Routing Problems and Applications*. PhD thesis, DTU Management Engineering, 2010.

[12] Kris Braekers, Katrien Ramaekers, and Inneke Nieuwenhuyse. The vehicle routing problem: State of the art classification and review. *Computers & Industrial Engineering*, 99:300–313, 12 2015.

[13] http://www.bernabe.dorronsoro.es/vrp/index.html?vrp-intro.html.

[14] Aida Benjamin and John Beasley. Metaheuristics for the waste collection vehicle routing problem with time windows, driver rest period and multiple disposal facilities. *Computers & OR*, 37:2270–2280, 12 2010.

[15] Daniela Arango González, Elias Olivares-Benitez, and Pablo Miranda. Insular biobjective routing with environmental considerations for a solid waste collection system in southern chile. *Advances in Operations Research*, 2017:1–11, 08 2017.

[16] Jean-François Cordeau, Gilbert Laporte, Martin Savelsbergh, and Daniele Vigo. *Vehicle Routing*, volume 14, pages 195–224. 01 2007.

[17] Kenneth Sorensen: University of Colorad. Metaheuristics, http://www.scholarpedia.org/article/metaheuristics.

[18] Fred Glover. Tabu search - part i. *INFORMS Journal on Computing*, 2:4–32, 01 1990.

[19] Prasanna Kumar, Mervin Herbert, and Srikanth Rao. Population based metaheuristic algorithm approach for analysis of multi-item multi-period procurement lot sizing problem. *Advances in Operations Research*, 2017:3601217, Dec 2017.

[20] Xin-She Yang. *Nature-Inspired Optimization Algorithms*. 03 2014.

[21] H-P Schwefel Thomas Back, Ulrich Hammel. Evolutionary computation: Comments on the history and current state. *Evolutionary computation, IEEE Transactions on*, 1(1):3–17, 1997.

[22] Mouadh Yagoubi. *Optimisation évolutionnaire multi-objectif parallèle : application à la combustion Diesel*. Université Paris Sud - Paris XI, 2012.

[23] Ka-Chun Wong. Evolutionary algorithms: Concepts, designs, and applications in bioinformatics. *Nature-Inspired Computing: Concepts, Methodologies, Tools, and Applications*, 08 2015.

[24] Eckart Zitzler. *Evolutionary Algorithms for Multiobjective Optimization:Methods and Applications*. Swiss Federal Institute of Technology, 1999.

[25] Felix Streichert, Holger Ulmer, and Andreas Zell. Parallelization of multi-objective evolutionary algorithms using clustering algorithms. volume 3410, pages 92–107, 03 2005.

[26] Tadahiko Murata and H. Ishibuchi. Moga: Multi-objective genetic algorithms. page 289, 01 1995.

[27] Abdelhakim Cheriet. *Métaheuristique hybride pour l'optimisation multi-objectif*. PhD thesis, Université Mohamed Khider – Biskra, 2014.

[28] Jeffrey Horn, Nicholas Nafpliotis, and David Goldberg. A niched pareto genetic algorithm for multiobjective optimization. *IEEE Conference on Evolutionary Computation - Proceedings*, 1, 03 1997.

[29] Kalyan Deb, Amrit Pratap, Sameer Agarwal, and T. Meyarivan. A fast and elitist multiobjective genetic algorithm: Nsga-ii. *Evolutionary Computation, IEEE Transactions on*, 6:182 – 197, 05 2002.

[30] Eckart Zitzler, Marco Laumanns, and Lothar Thiele. Spea2: Improving the strength pareto evolutionary algorithm for multiobjective optimization. volume 3242, 01 2001.

[31] K. Deb. *Multi-Objective Optimization Using Evolutionary Algorithms*, volume 16. 01 2001.