UNIVERSITA' DEGLI STUDI DI PADOVA
Dipartimento di Ingegneria Civile, Edile e Ambientale

FACOLTA' DI INGEGNERIA

CORSO DI LAUREA MAGISTRALE IN ENVIRONMENTAL ENGINEERING

TESI DI LAUREA

VEHICLE ROUTING PROBLEM SOLVED THROUGH GENETIC ALGORITHMS IN THE FRAMEWORK OF MUNICIPAL SOLID WASTE COLLECTION AND TRANSPORT

IL VEHICLE ROUTING PROBLEM RISOLTO MEDIANTE ALGORITMI GENETICI NELLA RACCOLTA E TRASPORTO DEI RIFIUTI SOLIDI URBANI

RELATORE:
Prof. Giuseppe Salemi

LAUREANDO:
Giacomo Titti
1082725

Anno Accademico 2015/2016

Contents

List of Figures

List of Tables

Abstract

The route planning of a waste collection truck is a daily operative problem, which follow the optimization and the reduction of costs. It can be summed up by the Vehicle Routing Problem (VRP). The VRP is one of the well known problem studied by the Graph theory.

An operative solution of the Vehicle Routing Problem in the framework of the municipal solid waste drop-off collection and transport is exposed below. Two different approaches to the Problem have been presented, the Nearest Neighbor Algorithm (NNA) and an algorithm based on Genetic Algorithm (GA), that has been developed in this work. This algorithm, called Three-phase Algorithm, has been compared with the results of the Nearest Neighbor Algorithm and simulated on the drop-off waste collection system in the city of Padova (Italy). The case study is just partial due to the lack of availability by the Local Authorities to exchange data.

The GA based algorithm is composed of two parts, one aimed to improve the construction of the cluster of the waste collection points that can be collected with one route, and the other to choose the shortest route that reach each point just one time in a single cluster. Both processes are based on the Genetic Algorithm.

Introduction

The municipal solid waste management, is a process which is not considered a priority in a large part of the world. For the most industrialized cities it is often an income but, to reach this aim, it requires a depth research about where the costs should be reduced and the benefits increased. In the developing countries, the lack of one of the steps of the waste management, or the lack of the waste management itself, creates unsanitary conditions and epidemics.

The modern waste management moves from the waste generation, to the final destinations, through the waste collection, transport and treatment. Each of these steps can be optimized to reduce the costs and increase the benefits. The idea explained below is concentrated in the optimization of the collection and transport.

The aim is to improve the collection and transport of the municipal solid waste (MSW) considering just one cost: the distance, that is probably the most expensive aspect of this step in the MSW management, and a limit: the volume capacity of the collection trucks.

Euler solved the first problem using graph theory in 1735 and thereby led the foundation of a very vast and important field of graph theory. He has created the first graph to simulate a real time place and situation to solve a problem which was then considered one of the toughest problems [33] (the seven bridges of Konigsberg). The study of cycles on polyhedra by the Thomas P. Kirkman (1806 - 95) and William R. Hamilton (1805-65) led to the concept of a Hamiltonian graph [58]. Thus the graph theory has ever solved the route planning.

The Vehicle Routing Problem (VRP) is the evolution of the Hamiltonian graph and more recent than the Traveling Salesman Problem. Briefly, the VRP is the problem of the Traveling Salesman but with the add of costs. The evolution of the graph theory has inspired the application of heuristic and meta-heuristic algorithms like the Nearest Neighbor Algorithm and the Genetic Algorithm to solve problems of this discipline. This paper is an example of that adaptation.

This work was began with the idea on the base of the Nearest Neighbor Algorithm developed by Sas Wahid HamzahAll [56], to improve the waste collection in the city of Padova and the positioning of the collection bins in the drop-off collection system of the city. But, after the finding of the Genetic Algorithm wrote by Joseph Kirk [57] on Matlab platform, the goal of the research has been change, and a GA based algorithm has been developed.

The main goal of the paper is to study the behavior of the Three-phase Algorithm (applied in the drop-off waste collection system in a part of the city of Padova) and to make a comparison with the NNA. The case study is just partial due to the lack of availability by the Local Authorities to exchange data. For this reason, a data collection campaign have been done in the city.

The study is organized as follow: the chapter 1 gives an overview of the world of the waste management. It describes the structure and the single moments of the waste management, from the segregation to the final destination, using and specifying the global standardized terms of the sector. The chapter gives also an overview of the waste collection in some cities of the world, from the richest to the poorest.

In the second chapter, the evolution, the definitions and the most important theorems of the graph theory in the routing problem have been presented. Moreover, an introduction and a classification of the modern Routing Problem has been done. The chapter 3 describes the difference between the heuristic algorithms and the evolution in the meta-heuristic algorithms. Then, it introduces the Genetic Algorithm, the Nearest Neighbor Algorithm and their applications in the VRP and in the Traveling Salesman Problem.

The chapter four is the core of this work. The methodology and the results of the simulations have been reported.

Finally the last two chapters are the discussion, where the results have been analyzed, and the conclusions.

The methodology describes all the instruments used. Then the route plans developed by the simulations have been shown. At first, the results of the Nearest Neighbor Algorithm and after the results of the Three-phase Algorithm have been presented. The large number of simulations done give a complete description and evaluation of the algorithm developed in this work, in comparison to the NNA, that is considered one of the best algorithm to solve the Vehicle Routing Problem. A best and unique result doesn't exist in the combinatorial problems, as confirmed by the conclusions of this work.

Chapter 1

Waste management in the world

The Waste management is the group of procedures that come from the production of waste to the final destinations of them.

The terminology is important for understanding the waste management system and for communication, but a generally accepted terminology does not exist within the solid waste community.

1.1 Definitions

Some definitions from ASTM and US EPA of terms that are used below, have been shown in the following paragraph:

- Waste: a material that is unwanted at its present location; that is no longer useful for its original purpose; that has been disposed, or any combination thereof. Waste composition of a solid waste: characterization of multi-constituent waste by a breakdown into specified waste components on the basis of mass or volume fraction or percentage.(Syn. solid waste composition).
- Unprocessed municipal solid waste: municipal solid waste in its as-discarded form and that has not been size-reduced, separated, or otherwise processed.
- Incineration: controlled burning of waste products or other combustible material;
- Incinerator: a device constructed for the purpose of containing a material for thermal oxidation [3].
- Biodegradable material (putrescible): materials that can be broken down by microorganisms into simple, stable compounds such as carbon dioxide and water. Most organic materials, such as food scraps and paper, are biodegradable.
- Bottle bill: a law requiring deposits on beverage containers (see Container Deposit Legislation).
- Bulky items: large items of refuse including, but not limited to,appliances, furniture, large auto parts, nonhazardous construction and demolition materials, trees, branches, and stumps that cannot be handled by normal solid waste processing, collection, or disposal methods.
- Commercial waste: waste materials originating in wholesale, retail, institutional, or service establishments, such as office buildings, stores, markets, theaters, hotels, and warehouses.

- Composting: the controlled biological decomposition of organic solid materials under aerobic conditions.
- Construction and demolition waste: materials resulting from the construction, remodeling, repair, or demolition of buildings, bridges, pavements, and other structures.
- Curbside collection: programs in which recyclable materials are collected at the curb, often from special containers, and then taken to various processing facilities.
- Domestic Waste: means the waste produced in the course of a domestic activity.
- Drop-off collection: a method of collecting recyclable or compostable materials in which the materials are taken by individuals to collection sites, where they deposit the materials into designated containers.
- Ferrous metals: metals derived from iron. They can be removed from commingled materials using large magnets at separation facilities.
- Gate volume: the amount of waste, measured by volume, that enters a landfill.
- Generation rate: the amount of waste that is produced over a given amount of time. For example, a district may have a generation rate of 100 tons per day.
- Hazardous waste: waste material that exhibits a characteristic of hazardous waste as defined in RCRA (ignitability, corrosivity, reactivity, or toxicity), is listed specifically in RCRA 261.3 Subpart D, is a mixture of either, or is designated locally or by the state as hazardous or undesirable for handling as part of the municipal solid waste and would have to be treated as regulated hazardous waste if not from a household.
- Inorganic waste: waste composed of matter other than plant or animal (i.e., contains no carbon).
- Institutional waste: waste materials originating in schools, hospitals, prisons, research institutions, and other public buildings.
- Integrated solid waste management: a practice using several alternative waste management techniques to manage and dispose of specific components of the municipal solid waste stream. Waste management alternatives include source reduction, recycling, composting, energy recovery, and landfilling.
- Mechanical separation: the separation of waste into components using mechanical means, such as cyclones, trommels, and screens.
- Municipal solid waste (MSW): MSW means household waste, commercial solid waste, nonhazardous sludge, conditionally exempt small quantity hazardous waste, and industrial solid waste.
- NIMBY: acronym for "not in my back yard." An expression frequently used by residents whose opposition to siting a waste management facility is based on the facility's proposed location.
- Organic material (organic waste): materials containing carbon. The organic fraction of MSW includes paper, wood, food scraps, plastics, and yard trimmings.

- Recycling: the process by which materials otherwise destined for disposal are collected, reprocessed, or remanufactured, and are reused.
- Reuse: the use of a product more than once in its same form for the same purpose; e.g., a soft drink bottle is reused when it is returned to the bottling company for refilling.
- Roll-off container: a large waste container that fits onto a tractor trailer that can be dropped off and picked up hydraulically.
- Scavenging: at a landfill or material recovery facility, scavenging is the uncontrolled separation of recyclable and reusable materials. Uncontrolled means that the operator does not monitor the removal of materials, and in many cases prohibits it. Material scavenging of recyclables may also occur at the curb or at drop-off centers.
- Scavenger: one who illegally removes materials at any point in the solid waste management system.
- Solid waste: any garbage, or refuse, sludge from a wastewater treatment plant, water supply treatment plant, or air pollution control facility and other discarded material, including solid, liquid, semi-solid, or contained gaseous material resulting from industrial, commercial, mining, and agricultural operations, and from community activities, but does not include solid or dissolved materials in domestic sewage, or solid or dissolved materials in irrigation return flows or industrial discharges that are point sources subject to permit under 33 U.S.C. 1342, or source, special nuclear, or by-product materials as defined by the Atomic Energy Act of 1954, as amended (68 Stat. 923). (Definition from 40CFR 258.2.).
- Source reduction: the design, manufacture, acquisition, and reuse of materials so as to minimize the quantity and/or toxicity of waste produced. Source reduction prevents waste either by redesigning products or by otherwise changing societal patterns of consumption, use, and waste generation. (See also, "waste reduction.")
- Source separation: the segregation of specific materials at the point of generation for separate collection. Residential generators source separate recyclables as part of curbside recycling programs.
- Special waste: refers to items that require special or separate handling, such as household hazardous wastes, bulky wastes, tires, and used oil.
- Transfer station: a permanent facility where waste materials are taken from smaller collection vehicles and placed in larger vehicles for transport, including truck trailers, railroad cars, or barges. Recycling and some processing may also take place at transfer stations.
- Waste reduction: waste reduction is a broad term encompassing all waste management methods source reduction, recycling, composting that result in reduction of waste going to a combustion facility or landfill.
- Waste stream: a term describing the total flow of solid waste from homes, businesses, institutions and manufacturing plants that must be recycled, burned, or disposed of in landfills; or any segment thereof, such as the "residential waste stream" or the "recyclable waste stream."

- Wastewater: water that is generated, usually as a by-product of a process, that cannot be released into the environment without some type of treatment [5].

1.2 Waste management introduction

Waste management system is a very complex process that can be divided into 4 phases according to Christensen (2011):

- waste generation: waste categories, waste types, waste quantities and composition;
- collection and transport: source separation, waste collection stations and centers (eventually called recycling centers), collection, transport and bulk transfer;
- treatment: separation of waste in material recovery facilities, incineration, biological treatment and other operations or processes changing the characteristics of the waste;
- recycling, utilization and landfilling (RUL): here the waste leaves the waste system permanently and is recycled (e.g. paper and glass for remanufacturing), utilized on land (e.g. compost) or in construction (e.g. bottom ash from incineration), or is disposed of in a landfill [1].

1.2.1 Waste generation

Waste generation is the starting point of the waste system and defines the waste in terms of waste categories, waste types, quantities, material fractions and substances. Proper knowledge about waste generation is a prerequisite for planning and designing a good waste management system. The following terms characterize the waste:

- Waste categories are broad classes of waste coming from sources with common characteristics. Residential waste, commercial and institutional waste, industrial waste, and construction and demolition (C & D) waste are the main categories.
- Waste types are subclasses of waste categories and have common characteristics with respect to source and composition potentially resulting in separate collection and handling. For example, residential waste includes the waste types: household waste, garden waste, bulky waste and household hazardous waste. Industrial waste holds several types according to industrial branch.
- Waste quantities are often reported as wet weight, since this is easily measured. Occasionally quantities are given as volume. The unit generation rate is a key parameter. For residential waste, the unit generation rate is often kg/year/person, or kg/week/household. For commercial waste, the unit generation rate could be kg/year/employee, or kg/year/m^2 of store, or kg/1000 Euros of sales. By multiplying together the number of characteristic units and the timeframe, the total amount of waste is determined. The unit generation rate is also a convenient parameter for estimating future waste quantities in areas with a growing population.
- Material fractions are visually identifiable fractions in the waste with common features: paper, plastic, glass, organic kitchen waste, etc. Each material fraction may be divided into subfractions such as, for example in the case of paper: newspaper, advertisements, magazines, paper towels, etc.
- Substances are individual chemical substances in the waste, which typically require analytical techniques to identify. This could be water, protein, ash content, nitrogen, cadmium, etc. [1].

1.2.2 Collection and transport

Waste collection is the organized storage of waste at source and its collection and transport to a waste treatment or RUL (recycling, utilization, landfilling) facility. The purpose of waste collection and transport is to remove the waste from the source of generation and collect sufficient quantities for a rational management system. The following terms define waste collection and transport:

- Source segregation: the art of segregating the waste at source into different waste types, material fractions and/or material sub-fractions for separate collection.
- In-house collection: the technical system for storage of waste at the source prior to collection (e.g. bins, bags, containers). This may include bins in the individual households as well as common bins in apartment buildings; the latter may actually be located outside.
- Waste collection stations or centers: public facilities, where individuals can bring waste. These centers are often referred to as amenity centers, recycling centers or recycling stations since they receive recyclables, but most centers also receive other types of waste.
- Collection: the organized collection of waste at the pick-up location (at the house, at the curb, at central location points, etc.), routing of the vehicle to different pick-up locations, any weighting and control of the waste, until the vehicle is full or has completed its task.
- Transport: the transport of the waste from the point where the collection was completed and until unloading of the waste at a treatment or RUL facility. Transport may also involve transport of treated waste or treatment residues.
- Transfer stations: for the transfer of waste, e.g. from small vehicles to large units (trailer systems, trains, barges) suited to long-distance transport. Transfer stations may provide some kind of mechanical treatment (e.g. compaction, shredding, baling) [1].

Waste collection and removal are serious health issues. We can summarize the main collection systems [40]:

- Shared collection
- Individual collection

The first category could be divided into:

- Dumping at designated location: residents and other generators are required to dump their waste at a specified location or in a masonry enclosure.
- Shared container (Drop-off) Fig. 1.1 residents and other generators put their waste inside a container which is emptied or removed.

The second class could be divided into:

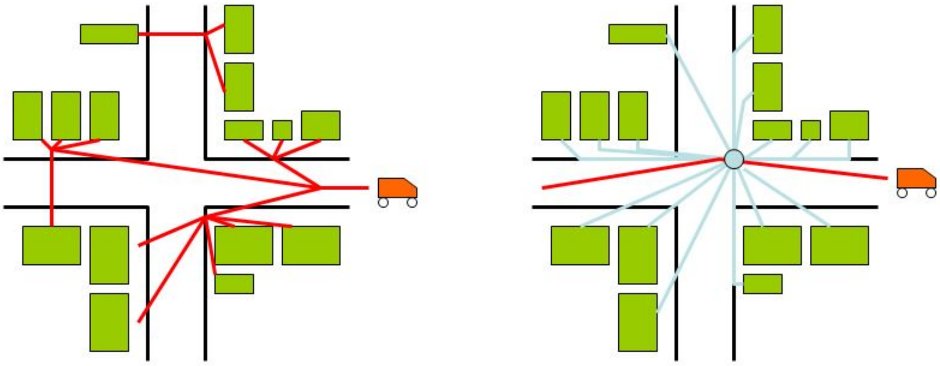


Figure 1.1: A schematic description of two waste collection systems. On the left curbside collection, on the right drop-off collection. Red line: collection route, ble line: delivery route [6].

- Block collection: collector sounds horn or rings bell and waits at specified locations for residents to bring waste to the collection vehicle.
- Kerbside collection (curbside collection) Fig. 1.1 waste is left outside property in a container and picked up by passing vehicle, or swept up and collected by sweeper.
- Door to door collection: waste collector knocks on each door or rings doorbell and waits for waste to be brought out by resident.
- Yard collection: collection labourer enters property to remove waste [40].

1.2.3 Treatment

Treatment may involve mechanical, thermal or biological treatment or combinations hereof. The purpose of waste treatment is to recover recyclables, extract energy from the waste or improve the characteristics of waste before further handling (remove impurities, degrade putrescible and odorous waste, reduce volume, etc). The treatment processes are:

- Mechanical treatment involves size reduction, sorting and compaction. They may appear as separate facilities or in combination with thermal or biological waste treatment as pre- or post-processing units. If the mechanical treatment is focused on separating or upgrading recyclables the facility may be referred to as a material recovery facility (MRF). Mechanical treatment basically consists of unit operations that alter the physical but not the chemical characteristic of the waste.
- Thermal treatment involves incineration and pyrolysis/gasification. Incineration is combustion with excess of air yielding nearly complete oxidation of organic carbon to carbon dioxide. Pyrolysis is partial oxidation that increases the temperature (internal combustion) subsequently resulting in the generation of pyrolysis gas. Gasification is a high-temperature process with external heating of the waste resulting in the release of reduced gasses with high calorific value.

  Thermal processes produce gas/flue gas that must be cleaned and a solid residue often referred to as bottom ash or slag.

- Biological treatment involves composting, anaerobic digestion and a combination thereof. Composting is a biological aerobic process converting the easily degradable organic waste into carbon dioxide and stable organic matter. The solid residues are compost and a reject. The latter requires further treatment. Anaerobic digestion is the degradation of organic waste in the absence of oxygen. This leads to methane and carbon dioxide. The methane content makes the gas useable as an energy source. The residues are liquid or solid. The off-gases from composting as well as from anaerobic digestion must be controlled [1].

1.2.4 Final destinations

Recycling, utilization and landfilling (RUL) is the final step in the waste management system. The purpose of RUL is to recover and utilize materials or return the materials to the cradle (landfill). The terms are:

- Recycling is the use of the materials in the production of the same or similar products that were the origin of the waste material. Recycling uses the original material characteristics of the waste. In recycling, waste substitutes for virgin production of the same material. Typical materials are paper, glass, plastic, iron, aluminium and asphalt.
- Utilization is the use of waste fractions or treated waste in away that is different from the origin of the waste. Utilization is often driven by secondary characteristics of the waste material: it covers both material utilization and energy utilization. Utilization may be: the use of compost on land as a fertilizer, etc...
- Landfilling is the dedicated use of land for disposing waste in an engineered facility. Landfilling may also have the form of land reclamation where filling takes place at the shore of water bodies with the intention to reclaim new land. Since a landfill has a lifetime of many decades and environmental controls must be maintained and operated for extended periods, the landfill after filling has been completed also forms part of the waste management system [1].

1.2.5 Municipal solid waste

A simple definition of waste according to Christensen (2011) could be: "Waste is a left-over, a redundant product or material of no or marginal value for the owner and which the owner wants to discard." An important characteristic is that being "waste" is not an intrinsic property of an item but depends on the situation in which the item appears as defined by its owner or in other words how the owner values the item. This means that becoming "waste" may depend on many factors, for example:

- Location: farming communities may easily make use of food waste for animal feeding, while this is less feasible in a highrise in an urban area.
- State: the item may be repairable depending on its state (price, age, type of damage) and thereby avoid being discarded.
- Income level: the higher your income the more food you may discard or the more items you may discard because they no longer are in fashion or up to date.

- Personal preferences: certain types of items may be collectors items or possess veneration for some individuals.

This also suggests that what is waste to one person may not be waste to another person and there may be a potential for trading if the cost for transferring the item does not exceed the value of the item as preceived by the new owner [1].

The term municipal solid waste, can be studied dividing it into single words. With "municipal", waste ranges from households waste to commercial, non hazardous industrial waste (the latter two have to be similar to domestic waste in quantity and quality), demolition waste and sewage sludge [4]. So mainly wastes that came from the living area of the city.

The definition of "solid waste" would be anticipated to be "a waste in a solid state". However, solid waste may be solid, or liquid as a sludge or as a free chemical phase. This originates from defining solid waste as waste that is not water (wastewater) or air borne (flue gasses) [1]. Therefore "solid" leave out the sewage sludge that "municipal" considers.

It is often convenient to distinguish between non-hazardous waste and hazardous waste. This may apply to practical waste management as well as to the regulatory aspects of waste management. Hazardous waste is more dangerous to the environment and to those handling the waste and must be technically managed with more strict controls than non-hazardous waste. The hazardousness of a waste is assessed according to criteria as (simplified after CEC, 2008):

- Explosive under the effect of flame, shock or friction.
- Oxidizing in contact with other materials resulting in highly exothermic reactions.
- Flammable in contact with air having flashpoint less than 55 ℃(highly flammable, with a flashpoint less than 21 ℃).
- Irritant: causing inflammation through contact with skin or mucous membrane.
- Harmful: causing limited health risks through inhalation, ingestion or penetration of skin.
- Toxic: causing serious, acute or chronic health risks and even death through inhalation, ingestion or penetration of skin.
- Carcinogenic: inducing cancer or increasing cancer incidence through inhalation, ingestion or penetration of skin.
- Corrosive by destroying living tissue on contacts.
- Infectious due to viable microorganism or their toxins known or reliably believed to cause disease in man or other living organisms.
- Toxic for reproduction: substances and preparations which, if they are inhaled or ingested or if they penetrate the skin, may induce nonhereditary congenital malformations or increase their incidence.
- Mutagenic: inducing hereditary genetic defects or increasing their incidence through inhalation, ingestion or penetration of skin.
- Releasing toxic gases in contact with water, air or an acid.

- Sensitizing: substances and preparations which, if they are inhaled or if they penetrate the skin, are capable of eliciting a reaction of hypersensitization such that on further exposure to the substance or preparation, characteristic adverse effects are produced.
- Ecotoxic: presenting any immediate or delayed risks for any sector of the environment.
- Substances capable by any means after disposal of yielding another substance which possesses any of the characteristics listed above.

These criteria are for practical assessments supplemented with quantitative limits as well as methods for their determination (for example, see CEC, 2008, and in particular CEC, 2000) [1].

1.3 The european's waste management policy

EU waste policy has evolved over the last 30 years through a series of environmental action plans and a framework of legislation that aims to reduce negative environmental and health impacts and create an energy and resource-efficient economy.

The EUs Sixth Environment Action Programme (2002-2012) identify waste prevention and management as one of four top priorities. Its primary objective is to ensure that economic growth does not lead to more and more waste. This led to the development of a long-term strategy on waste. The 2005 Thematic Strategy on Waste Prevention and Recycling resulted in the revision of the Waste Framework Directive, the cornerstone of EU waste policy. The revision brings a modernised approach to waste management (Integrated Waste Management), marking a shift away from thinking about waste as an unwanted burden to seeing it as a valued resource. The Directive focuses on waste prevention and puts in place new targets which will help the EU move towards its goal of becoming a recycling society. It includes targets for EU Member States to recycle 50% of their municipal waste and 70% of construction waste by 2020.

The Directive introduces a five-step waste hierarchy where prevention is the best option (Fig. 1.2), followed by re-use, recycling and other forms of recovery, with disposal such as landfill as the last resort. EU waste legislation aims to move [4].

Figure 1.2: The European Waste Framework Directive and its revisions introduces a five-step waste hierarchy where prevention is the best option in the waste management, followed by re-use, recycling and other forms of recovery, with disposal such as landfill as the last resort [4].

1.4 Waste Management in Padova

Padova is a beautiful, historical city, in the north-east of Italy (Fig. 1.3).

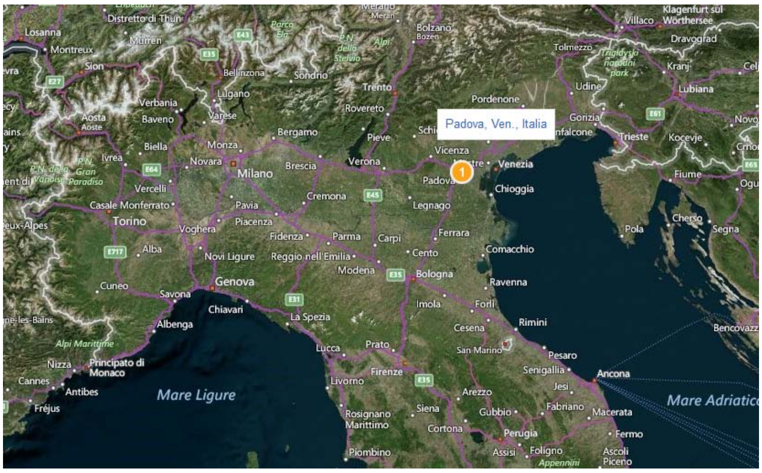


Figure 1.3: Padova is situated in the north-east of Italy, in the Veneto region, near to Venice. The image, is a capture from Bing map [61].

The city is divided into 6 districts as shown in the Fig. 1.4:

1. Centre (yellow);
2. North (blue);
3. East (gray);
4. South-East (violet);
5. South-West (red);
6. West (green).

The management of the MSW is done by a single company from the collection to the final destination.

The segregation and collection of the MSW change according to the district and the zone into the district (Fig. 1.5). The service, is done in different way in an Orange (Centre), Green (Centre) and Yellow area.

The Orange includes the area around the core of the Old town, in particular around Piazza delle Erbe, Piazza della Frutta and Piazza dei Signori (Fig. 1.6).

The second one includes the area inside the historical town walls (Fig. 1.6).

The last one includes the rest of the city, except some areas of the suburb (they are the non-yellow regions in the outer border of the Fig. 1.5, like: Altichiero, Pontevigodarzere, etc.) which have a specific waste collection [34].

The MSW in the Orange zone, is collected with a curbside system, and before segregated into sacs according to following classes:

- unsorted waste,
- glass and metals,
- plastic,

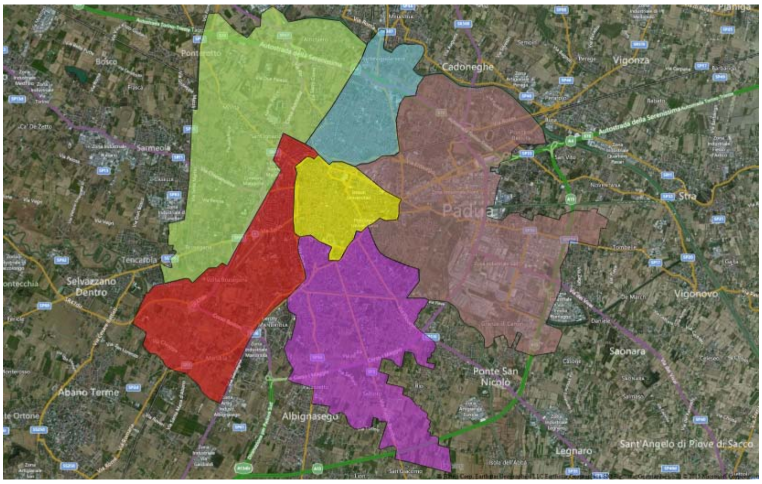


Figure 1.4: This zones division of Padova, is for a better management of the services in the city. Zones: 1-Centre (yellow), 2-North (blue), 3-East (gray), 4-South East (violet), 5-South West (red), 6-West (green). Colors are not relevant

Figure 1.5: MSW collection in Padova according to the single distric and zone of district. The service, is done in different way in an Orange (Centre), Green (Centre) and Yellow area, the other colors rapresent specific waste collection systems, which are not significance for this study [62].

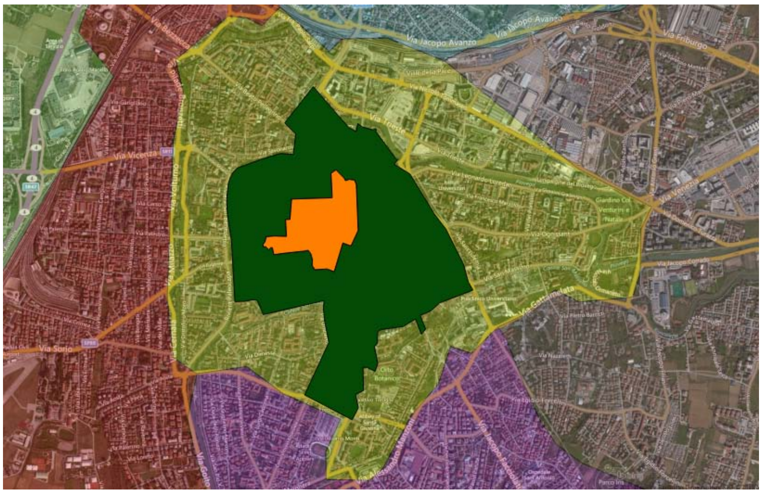


Figure 1.6: Centre district(the yellow ones in Fig. 1.4) is divided into two different zones, which represent different MSW collection systems: Green and Orange areas.

Figure 1.7: Small truck for the Orange zone [34].

- putrescible material,
- paper,
- cardboard (dedicated areas).

Due to the limited width of the roads of the town all, small trucks are used in the Orange zone (Fig 1.7).

The MSW in the Green zone, is collected with a drop-off system, and before segregated into bins of 1000l, 360l, 240l, 120l according to following classes:

- unsorted waste,
- metals and plastic,
- glass,
- putrescible material,
- paper and cardboard.

Figure 1.8: Truck for the Green zone (10 m^3) [34].

In this case, compactor trucks bigger than the first type (capacity of 10 m^3) are used in the Green zone (Fig 1.8). In some areas of the Green zone, the plastic, metals and glass are segregated in the same bin (multi-material), for example in the university campus.

The MSW in the Yellow zone, is collected with a drop-off system, and before segregated into bins of 2400l, 3300l, 240l according to following classes:

- unsorted waste,
- metals, plastic and glass,
- putrescible material,
- paper and cardboard.

In this case, the compactor trucks bigger than the first two types (capacity of 26 m^3) are used in the Yellow zone (Fig 1.9) [34].

Figure 1.9: Truck for the Yellow zone (10 m^3) [34].

The final destinations available are:

- an inceneration plant,
- two landfills(Ponte San Nicoló, Via Vasco de Gama),
- recicling companies [34].

1.5 Some world's cities Waste Management

Waste generation is linked to economical activities and flow of materials in society. The schematic diagram in Fig. 1.10 illustrates the flow of materials from the environment through society and back to the environment. The diagram pictures the fact that resources are not consumed but merely transformed in the process of extraction from the environment, production and use before ending up as waste. This waste may be returned back into the production-use cycle in society or disposed of into the environment. The material flow is driven by a significant use of energy, and emissions to air, water and soil are associated with all activities within the flow system. Also the extraction of resources and the disposal of waste into the environment may have associated environmental burdens. A large extraction of resources may also lead to the depletion of resources, for example certain metals; and disposal activities may damage resources by contamination, for example groundwater resources at a landfill. Modern society is characterized with a very large extraction of resources from the environment and a very large disposal of waste in the environment. As the economy expands, the material flow traditionally also expands, leading to increased environmental burdens and resource consumption. Schematically it may seem possible to link the disposal with the extraction of resources and thereby close the loop, but the resources mined and the biomasses grown (inputs) are so different from the waste disposed (outputs) that this is neither technical not economical feasible [1].

Figure 1.10: Materials life cycle: Lavoisier law. It illustrates the flow of materials from the environment through society and back to the environment. And their consumption and transformation, until come back to the land (closure of the loop) [2].

The direct material input per person per year (including energy resources) on a country basis is somewhat larger than the input to a city, since mining, industry and agriculture also have large inputs. Fig. 1.11 shows a plot of the direct material input as a function of gross domestic product (GDP: is the standard measure of the value of final goods and services produced by a country during a period minus the value of imports [39].) for a number of countries. In general it appears that a high GDP is associated with a high material input, in the order of 2050 t/year/person. The link is supposed to be strongest for countries with a basic developing economy, while some countries seem to have decoupled the economic growth from the direct material input. This may be due to economical development in high-tech areas (e.g. telecommunications) or the

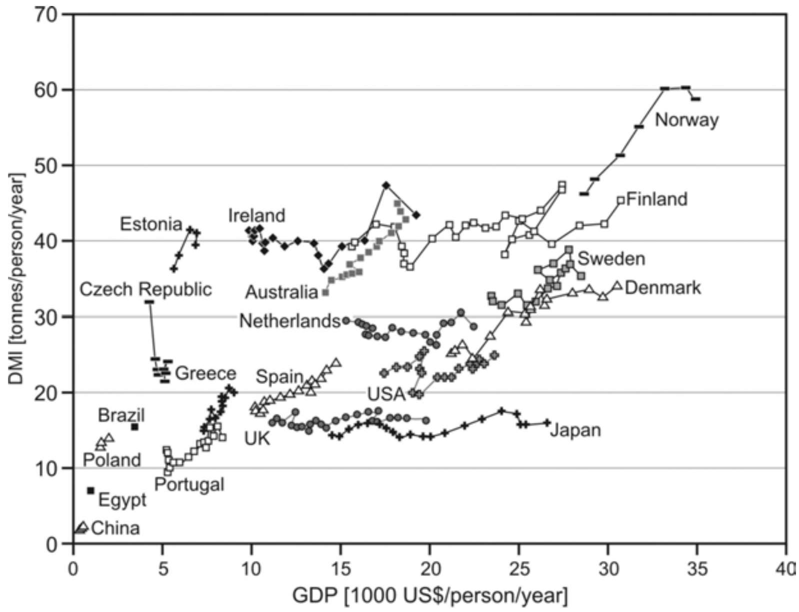


Figure 1.11: The relationship between the direct material input (DMI) as a function of the gross domestic product (GDP) of some countries [1].

service sector (e.g. financial sector) or the combination of a decline in heavy industry and mining and growth in sectors with less material consumption. When heavy industry closes down in one country, this may lead to growth in other countries of the same type of industry, thus only moving the material use to another country unless the moving involves introduction of new and resource-saving technologies [1].

Now we are looking for an overview of the waste management strategies developed in some cities in the world in a range from cities with a small to an enormous number of people and from low to great economic development, summarized in Tab. 1.1 and below some other general informations.

Munich

The waste of 1,32 Million inhabitans in 741000 households (curbside collection: residual waste, paper waste, organic waste, bulky waste and refrigerators per order) is manages with 1000 container (Duales System Deutschland: DSD Duales System Holding GmbH & Co. KG synergizes the entrepreneurial activities of Der Grune Punkt for a sustainable economy, one that keeps recyclables in closed circuits. The aim is to avoid wasting raw materials, and to minimize the impact on the climate and the natural environment [52]. This dual system picks up household packaging in parallel to the existing municipal waste-collection systems. DSD only collects packaging material from manufacturers who pay a license fee to DSD.) (in the street about 200 m: glass, metal, plastic); 52 stops of the hazardous waste collection truck, every 4 weeks for 1 hour (in the street about 2000 m: hazardous waste); 12 recycling yards, open 60 hours per week (enclosure about 3000 m: hazardous waste, recyclable material, bulky waste, garden waste);

One waste incinerator (waste for thermal treatment, waste for Energy recovery), one landfill, one Dry Fermentation Plant Recycling Center for: large volumes of bulky waste, recyclables, asbestos and mineral fibres), are the destinations of the waste collected in the city of Munich [7].

Table 1.1: Some example of the world's cities' waste collection. DF=Drop-off collection, CU=Curbside collection.

City	Segregation bins	Collection system
Munich [7]	• residual waste, paper, organic waste (3-bins system) • glass • cans • plastic, packaging material and composite material waste (DSD) • hazardous waste	• CU for residual; organic; paper • DF for glass; cans, plastic, packaging and composite
London [41]	• one mixture recycling sack (paper,glass,plastic bottles, card, plastic containers and other plastic, houshold metal packaging) • food waste • small electrical and electronical items • large appliances • textile • batteries and low energy light bulb • unsorted waste	• see note
Dar es Salaam (Tanzania) [8]	• unsorted waste and recycling by pikers	• CU high density population • handcart to collection sites for low density population
Beijing [17]	• glass, metal, paper, plastic (system and pickers) • unsorted waste • kitchen waste	• ...

City	Segregation bins	Collection system
Madrid [12]	• paper and paperboard • glass • plastic and metal containers, cartons, plastic bags and wooden boxes • food waste, unsorted waste • hazardous, bulky waste, debris and other	• DF for paper and paperboard; glass; plastic and metal containers, cartons, plastic bags and wooden boxes; food waste, unsorted waste • Clean Point for hazardous, bulky waste, debris and other
Prague [43]	• unsorted waste • plastic • glass • green waste • bulky waste • metal • electrical waste • kitchen waste	• DF for unsorted waste; plastic; glass; paper, carton, cardboard; beverage carton; metal; bulky waste; electrical waste; green waste; kitchen waste
Oslo [44]	• paper, cardboard, glass, metal • plastic, kitchen waste, unsorted waste (blue and green sack)	• CU
Shibuya (Japan) [46] [13]	• combustible (biodegradable waste, clothes, wood scraps, paper scraps, plastic items, rubber items, leather items) • incombustible (elettrical appliances, metal items, glass products, ceramics, pottery, fuorescent lights, batteries) • pet bottles • glass bottles • paper • cans • spray cans • bulky waste	• DF each family has a proper specific site

City	Segregation bins	Collection system
Tokyo [14] [15] [47]	• combustible (biodegradable waste, wood and grass, paper, plastic from 2008) • incombustible (ceramic, etc..) • pet bottles • glass bottles • paper • cans • bulky waste	• DF each family has a proper specific site
Istabul [18]	• unsorted waste and recycling by pikers	• DF
Saint-Lauren (Canada) [16] [48]	• unsorted waste • leaves and green waste (paper bags) • paper, cardboard, carton, metal, glass, plastic • batteries, CDs, mobile phones (specific bins) • bulky waste • kitchen waste	• CU for unsorted waste; leaves and green waste (paper bags); paper, cardboard, carton, metal, glass, plastic; kitchen waste (also possible have private composting bins) • DF for batteries, CDs, mobile phones—call for bulky waste
Cape Town [42]	• paper and cardboard • cans • glass • plastic • bulky waste • electronic waste • garden waste • unsorted waste	• DF for paper and cardboard; cans; glass; plastic; bulky waste; electronic waste; garden waste • CU for unsorted waste

City	Segregation bins	Collection system
Moscow (USA) [49]	• paper • metal • glass • plastic • unsorted waste • green waste • bulky waste • kitchen waste	• CU for unsorted waste; kitchen waste • DF for paper; metal; glass; plastic; (CU for household low than 4 units)—DF for bulky and green waste
Sydney [50] [51]	• paper, cardboard • plastic bottles, glass, cans, tins • plastic film, packaging • kitchen waste • batteries	• DF
Santiago (Chile) [9] [10]	• unsorted waste and recycling by pikers	• DF
New Berlin (USA) [19]	• residual waste, organic waste • paper, glass, cans, plastic, packaging material and composite material waste (single sort recycling) • hazardous waste	• CU for residual waste, organic waste; paper, glass, cans, plastic, packaging material and composite material waste (single sort recycling)
Berlin [45] [11]	• residual waste, paper, organic waste (3-bins system) • glass • cans • plastic, packaging material and composite material waste (Duales System Deutschland) • hazardous waste	• CU for residual; organic; paper • DF for glass; cans; plastic, packaging and composite

London

The waste management of London can be described for each district and kind of buildings. In the Barbican district, every family has a specific refuse cupboard for the collection daily of mixed recycling, food waste, unsorted waste (CU). Other bins are also located in your nearest car park (DF).

In Golden Lane Estate, Mansell Street Estate, Middlesex Street Estate the sacks of mixed recycling is collected in front of the house's door (CU). Food waste can be placed in the caddy containing the tied liner inside the refuse cupboard of each family that will be collected (CU). Additionally, recycling bins and food waste bins are located in your nearest car park (DF). For unsorted waste, a chute is provided for the household waste. Ground floor flats will have access to the waste cupboard (CU) [41].

In addition, according to the type of building the waste collection can differ:

- Private blocks of flats with a bin store are provided with recycling, food waste, unsorted waste bins and/or clear sacks (DF).
- For private blocks of flats without bin store: mixed recycling, food waste, unsorted waste: a kerbside collection is provided to those residents who do not have access to a bin store or refuse chamber (CU).

In any case, small electrical and electronic items can be recycled in the bin areas at the specific addresses (DF), large electrical appliances are collected through the Bulky Waste collection service and sent for reuse or re cycling, textiles, clothes and shoes are collected in the Salvation Army collection bank at the specific addresses (DF) and low energy light bulbs and batteries can be recycled in the bins at the specific locations (DF) [41].

Dar es Salaam (Tanzania)

In more affluent, planned areas of the city, wastes are generally collected at curbside from households, commercial establishments, institutions and industry and taken directly to the Pugu dump. Where access by collection vehicle is impractical, collected wastes from these areas are taken initially to neighbourhood collection sites by handcart for bulking and informal resource recovery before transportation to Pugu. In planned and unplanned areas of the city where the populations are less affluent and the neighbourhoods more congested, waste is picked up by handcart for delivery to neighbourhood collection sites or taken directly to these sites by householders. The collector agency subsequently pick up the accumulated waste from the neighbourhood collection sites for transportation to the Pugu. In areas of the city where collection service is poor, individuals commonly dump their waste into drainage ditches, streams and by the roadside. This has been estimated to total upwards of 60% of the overall waste stream. Industry and commercial establishments are responsible for managing their solid wastes using private sector solid waste collection contractors. On the other hand, these neighbourhood collection sites can become mini dump sites with overflowing skips and burning wastes spread across a wide area. These sites currently increase the risk of animal borne diseases and respiratory illness.

It has been estimated that less than 40% of the solid waste generated in the city is collected for recovery or disposal. The remainder is illegally dumped in drains, rivers or by the road side. This results in increased flooding during the rainy season, risk of malaria carrying mosquitoes and greenhouse gas emissions [8].

Santiago

Presently, it is not compulsory to separate trash in Chile. As a consequence, there is little recycling consciousness among the citizens. In a 2001 survey, close to 70% of Chileans said they never or almost never separate their trash. Where recycling exists, it is minimal, sporadic and accomplished in an informal and voluntary way. It is estimated that 9% of the total amount of MSW generate in Santiago is recycled.

Most waste recuperation in Chile is done through rudimentary methods. The recovery, accumulation and commercialization of recyclable material is done manually. This informal economic sector is made up of street cardboard collectors (”cartoneros”) and scavengers (”cachureros”) who as individuals recover small volumes of paper, glass and aluminum cans from homes and businesses. Another informal commercial sector buys the collected material and sells it to a handful of recycling companies. Still, some government authorities are trying to raise recycling consciousness through the use of collecting containers, household compost projects, encouraging recycling in public offices and universities, educational programs in schools, and training courses. However, as long as trash separation is not compulsory, recycling will continue to be very limited [9].

Cape Town

Single residential properties and group housing schemes are issued with one 240 l container per household for unsorted waste. This container is collected once a week during weekdays (including public holidays). Residents may place these containers on the pavement outside their homes on waste collection days. Residents who need additional containers may apply for an enhanced service at an additional cost per container. Curbside collection for the other kind of waste [42].

Prague

Sorting out of waste has become an ever increasing challenge. Distinguishing between mixed domestic waste, which is incinerated or buried, and sorted waste, which can be recycled, is of ever greater importance. Different types of collecting bins have been produced for each type of waste; they vary in color and labeling symbols to distinguish what goes where:

- grey or black skips for mixed waste
- green skips for glass
- blue skips for paper, newspapers, cardboards, cartons
- black skips with orange lids for beverages cartons
- yellow skips for plastics (including plastic bags and containers)

Prague Town Hall arranges skips for organic waste (leaves from trees, branches), also skips for large waste from households (old furniture, textiles), as well as skips for hazardous waste (paints, varnishes, refrigerators, oil, fluorescent lamps, televisions). All this type of waste can be disposed of at the collection facilities; there is a limit of one cubic meter per a person. You can get rid of used batteries into special skips at collection facilities at some Town Halls [43].

Oslo

The city aims to recycle 50% of the household waste by the year 2018. Currently, it is 37%, so the goal is well within reach. A total of 85.5 kg of food waste per person was generated in 2014. And 40% of this was separated into green bags. Glass, metal, paper and cardboard are handled outside of the Optibag system (green bag). The food waste that ends up in the green bags is the right type of material [44].

Berlin

The Green Dot system (DSD) has been one of the most successful recycling initiatives, which has literally put packaging on a diet. The crux is that manufacturers and retailers have to pay for a "Green Dot" on products: the more packaging there is, the higher the fee. This clever system has led to less paper, thinner glass and less metal being used, thus creating less garbage to be recycled. The net result: a drastic decline of about one million tons less garbage than normal every year.

Any kind of bottle or glass jar that is non-returnable and on which you did not pay a deposit or "Pfand", belongs in the designated glass bins. Glass is sorted by color. There are different slots for depositing green, brown and clear glass. These bins are dotted over every neighborhood. The other bins are at your doorstep, and are color coded: green, blue, yellow, brown and gray. All packaging made of paper and cardboard, newspapers, magazines, waste paper, paper bags, etc, belong in the blue bins. Tissues, however, do not belong here. If the blue bin isn't at your home, it will be certainly somewhere in your neighborhood. Cans, plastic, polystyrene, aluminum, tinplate and "composite" materials like beverage cartons made of a mixture of materials belong in the yellow bin or should be put in the yellow bags. Empty spray cans are also allowed here. Bio stuff (biodegradable material) is anything destined for the compost heap in a good gardener's back yard. This includes kitchen scraps, peels, leftover food, coffee filters, tea bags and garden waste. For the private houses there is separate brown bin for this. Gray bin is the destiny of, finally, "almost the rest". Everything in the gray bins will be incinerated. The hazardous waste includes: fluorescent tubes, batteries and acids, cans of paint still containing paint, thinners, adhesives, corrosives, disinfectants, insecticides, and so forth, has to be treated as hazardous waste. They should be bring to the site for them to dispose of it in the proper manner. Batteries are disposed of separately in a small bin at the local shopping area [45].

Chapter 2

Graph Theory

Using the same terms of Frank Harary with some specifications introduced by Melissa DeLeon, a graph G consists of a finite nonempty set V of p points (nodes) together with a prescribed set X of q unordered pairs of distinct points (nodes) of V. Each pair $x = (u, v)$ of points in X is a line (edge) of G, and x is said to join u and v. We say that u and v are adjacent points; point u and line x are incident with each other, as are v and x. If two distinct lines x and y are incident with a common point, then they are adjacent lines. A graph with p points and q lines is called a (p, q) graph. The $(1, 0)$ graph is trivial. Thus, in the graph G of Fig. 2.1, the points u and v are adjacent but u and w are not; lines x and y are adjacent but x and z are not. Although the lines x and z intersect in the diagram, their intersection is not a point of the graph [21] [23].

2.1 Story of Graph Theory

Euler solved the first problem using graph theory in 1735 and thereby led the foundation of very vast and important field of graph theory. He created first graph to simulate a real time place and situation to solve a problem which was then considered one of the toughest problems [33] (the seven bridges of Konigsberg). The study of cycles on polyhedra by the Thomas P. Kirkman (1806 - 95) and William R. Hamilton (1805-65) led to the concept of a Hamiltonian graph. The concept of a tree, a connected graph without cycles, appeared implicitly in the work of Gustav Kirchhoff (1824-87), who employed graph-theoretical ideas in the calculation of currents in electrical networks or circuits. Later, Arthur Cayley (1821-95), James J. Sylvester(1806-97), George Polya(1887-1985),

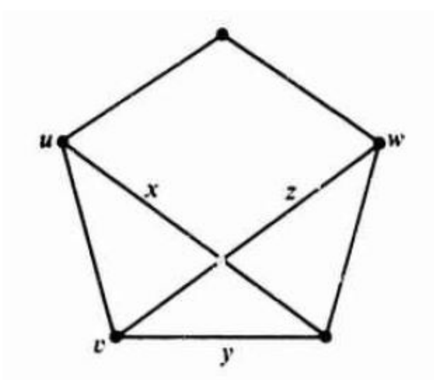


Figure 2.1: Graph G: u, v are a pair x, the line which joins them, x, y are not adjacent lines and also the points w, u are not adjacent [21].

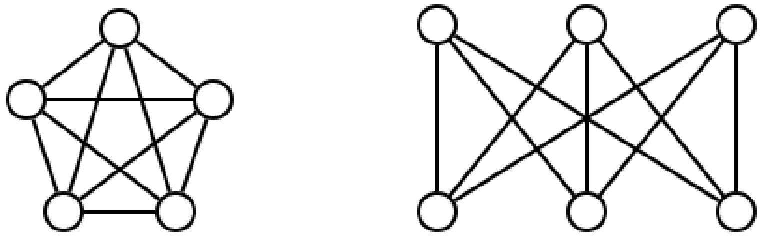


Figure 2.2: Graph $K5$ (left) and $K3,3$ (right) [59].

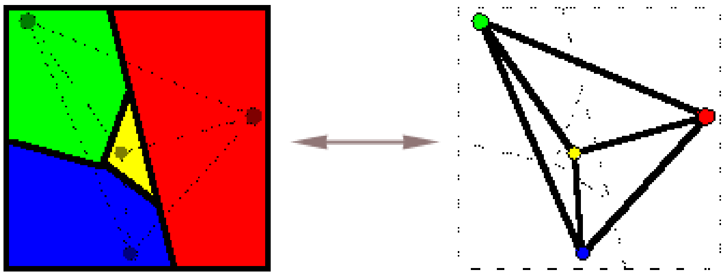


Figure 2.3: Four colors problem, posed by Francis Guthrie in 1852 [60].

and others use ”tree” to enumerate chemical molecules.

The study of planar graphs originated in two recreational problems involving the complete graph $K5$ (Fig. 2.2) and the complete bipartite graph $K3,3$ (Fig. 2.2). These graphs proved to be planarity, as was subsequently demonstrated by Kuratowski. First problem was presented by A. F. Mobius around the year 1840 as follows:

> Once upon a time, there was a king with five sons. In his will he stated that after his death the sons should divide the kingdom into five provinces so that the boundary of each province should have frontiers line in common with each of the other four provinces.

Here the problem is whether one can draw five mutually neighboring regions in the plane.

> The king further stated that all five brothers should join the provincial capital by roads so that no two roads intersect.

Here the problem is that deciding whether the graph $K5$ is planar. The origin of second problem is unknown but it is first mentioned by H. Dudeney in 1913 in its present form:

> The puzzle is to lay a water, gas, and electricity to each of the three houses without any pipe crossing another.

This problem is that of deciding whether the graph $K3,3$ is planar.

The celebrated four-color problem (no two adjacent regions have the same color) (Fig. 2.3) was first posed by Francis Guthrie in 1852. and a celebrated incorrect ”proof” by appeared in 1879 by Alfred B. Kempe. It was proved by Kenneth Appel and Wolfgang Haken in 1976 and a simpler and more systematic proof was produced by Neil Roberton, Daniel Sanders, Paul Seymour, and Robin Thomas in 1994 [58].

2.2 The seven bridges of Konigsberg

The term "combinatorial mathematics" is understood differently by different authors. Since the ancient Greeks, the principal mathematical abstraction has been infinity (which appears in the infinite series of positive integers, the unbounded divisibility of a line interval, the unboundedness of space, etc.). Unlike almost all other domains of mathematics, combinatorial calculus studies problems related to properties of objects finite in all senses [20].

One of the most famous combinatorial problem is the seven bridges of Konigsberg. Its negative resolution by Leonhard Euler in 1736 laid the foundations of graph theory. The "Konigsberg bridge" problem originated in the city of Konigsberg, formerly in Germany but, now known as Kaliningrad and part of Russia, located on the river Preger. The city had seven bridges, which connected two islands with the main-land via seven bridges Fig. 2.4. People staying there always wondered whether was there any way to walk over all the bridges once and only once [33]. The difficulty was the development of a technique of analysis and of subsequent tests that established this assertion with mathematical rigor.

Eulers trick is now sometimes called "double count" (the same quantity is evaluated in two different ways). First, Euler considers dryland regions connected by bridges, rather than the bridges themselves, as basic objects. If the dryland regions are denoted by capital Latin letters, then the route through the bridges can be represented by a sequence $ABCAD...$ of such letters. (An expression of this form does not reflect the order in which bridges connecting the same pair of regions are crossed, if there are several such bridges, but it turns out that the order does not matter!) Next, the number of symbols in a route once crossing each bridge is larger by 1 than the number of bridges crossed by this route, because each pair of neighboring symbols corresponds to crossing one bridge. Euler applies the double-count trick once more, when he notes that each bridge has two ends [20].

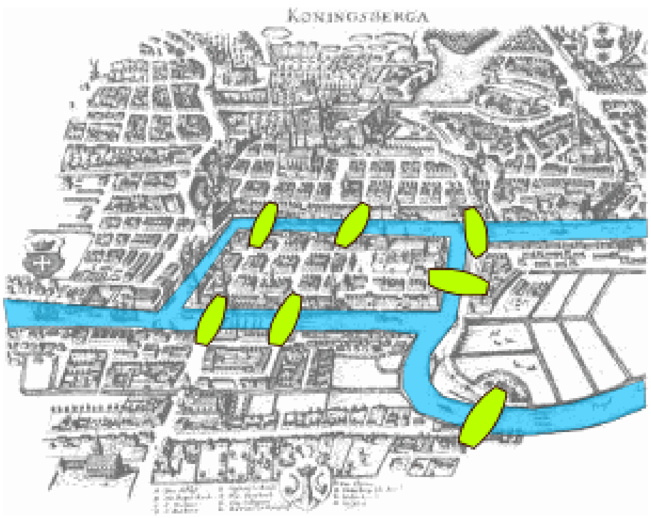


Figure 2.4: The seven bridges of Konigsberg. It is located on the river Preger. The city had seven bridges, which connected two islands with the main-land via seven bridges. People staying there always wondered whether was there any way to walk over all the bridges once and only once [33].

Rather than treating this specific situation, Euler generalized the problem and developed a criterion for a given graph to be so traversable; namely, that it is connected

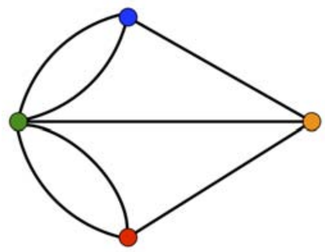


Figure 2.5: The graph that give a schematization of the problem of the seven bridges of Konigsberg. It introduces the Graph Theory [53].

and every point is incident with an even number of lines. While the graph in Fig. 2.5 is connected, not every point is incident with an even number of lines [21].

From all these considerations Euler concludes that *if the number of bridges is M and the number of regions at which an odd number of bridges end equals $2m$, then the expression for any route crossing all bridges must contain at least $M + m$ symbols, which is larger that $M + 1$ if $m > 1$.* This condition implies the nonexistence of a route crossing each of the seven bridges of Konigsberg precisely once. Euler did not give a detailed description of the construction of a route in the situations where $m \leqslant 1$; he confined himself to mentioning that, for $m = 1$, the required route must start and end at regions carrying an odd number of bridges, and that if two regions are joined by some bridges, then we can mentally remove pairs of such bridges and obtain a simpler scheme; after a traversal of the simplified scheme is constructed, the passages of the removed bridges are added [20].

In modern graph theory, a path traversing each edge precisely once is called an Eulerian path, and a closed Eulerian path, an Eulerian circuit. In the modern terminology, the criterion found by Euler is as follows: *in a connected unoriented graph, an Eulerian path (circuit) exists if and only if the number of vertices of odd degree (i.e the degree of a node v, in a graph G (Fig. 2.1), is defined to be the number of edges incident with v* [23]*) is at most 2 (there are no vertices of odd degree)* (see the Fig. 2.5).

A more general criterion covers the case in which the graph under consideration contains both oriented and unoriented edges: *in a connected graph, an Euler path (circuit) exists if and only if such a path (circuit) exists in the graph obtained by neglecting the orientation of edges and, for each vertex, the difference between the numbers of incoming and outgoing edges does not exceed the number of unoriented edges incident to this vertex* [20].

The previous conclusions of Euler, can be summarized and proved by the follow theorem.

Theorem: The following statements are equivalent for a connected graph G:

1. G is eulerian.
2. Every point of G has even degree.
3. The set of lines of G can be partitioned into cycles.

By the previous Theorem it follows that if a connected graph G has no points of odd degree, then G has a closed trail containing all the points and lines of G. There is an analogous result for connected graphs with some odd points.

Corollaries:

1. Let G be a connected graph with exactly $2n$ odd points, $n > 1$. Then the set of lines of G can be partitioned into n open trails.
2. Let G be a connected graph with exactly two odd points. Then G has an open trail containing all the points and lines of G (which begins at one of the odd points and ends at the other) [21].

2.3 NP-complete Hamiltonian path

A Hamiltonian path, also called a Hamilton path, is a graph path between two vertices of a graph that visits each vertex exactly once. If a Hamiltonian path exists whose endpoints are adjacent, then the resulting graph cycle is called a Hamiltonian cycle (or Hamiltonian cycle). A graph that possesses a Hamiltonian path is called a traceable graph [54]. In the general case, the problem of the existence of a Hamiltonian path in a graph is NP-complete [20].

Diracs theorem is a corollary of Ores result, but rather obtain them both as corollaries of a more general result, the Bondy-Chvatal Theorem. First have been to define the closure of a graph. Given a graph G with n vertices, the closure $cl(G)$ is uniquely constructed from G by repeatedly adding a new edge uv connecting a nonadjacent pair of vertices u and v with $degree(v) + degree(u) \geqslant n$ until no more pairs with this property can be found.

Theorem (BondyChvtal): A graph is Hamiltonian if and only if its closure is Hamiltonian [55].

As complete graphs are Hamiltonian, all graphs whose closure is complete are Hamiltonian, which is the content of the following earlier Ore and Dirac corollaries of the Posa theorem.

Theorem (Posa): Let G have $p > 3$ points. If for every n, $1 < n < \frac{p-1}{2}$, the number of points of degree not exceeding r. It is less than n and if, for odd p, the number of points of degree $\frac{p-1}{2}$ does not exceed $\frac{p-1}{2}$, then G is hamiltonian [21].

Corollaries:

1. (Dirac) A simple graph with n vertices ($n \geqslant 3$) is Hamiltonian if every vertex has degree $\frac{n}{2}$ or greater.

2. (Ore) A graph with n vertices ($n \geqslant 3$) is Hamiltonian if, for every pair of non-adjacent vertices, the sum of their degrees is n or greater [55].

The toughness, measure how tightly various pieces of a graph hold together. Every hamiltonian graph is necessarily 1-tough. Chvatal confirm the results of Fleischner, every graph that is more than $\frac{3}{2}$-tough is necessarily hamiltonian [22].

2.4 The Routing Problem

The VRP concern the vehicle connection from one ore more depots, to a set of destinations. The principal aim is determination of the m tours with minimum costs (i.e. distance, moneys, etc...) to reach from the deposit, the arcs ($R \subseteq X$) and vertices required ($U \subseteq V$) [25]. The road network can be described using a graph $G = (V, X)$ where the arcs (group X) are roads and vertices (group V) are junctions between them. The arcs may be directed or undirected due to the possible presence of one way streets or different costs in each direction.

The Vehicle Routing Problem (also known in literature as the "vehicle scheduling", "vehicle dispatching", or simply as the "vehicle problem" appears very frequently in practical situations not directly related to the delivery of goods) is a generic name given to a whole class of problems [24]:

- if $R = 0$, the VRP is called Node Routing Problem (NRP);
- if $U = 0$, the VRP is called Arc Routing Problem (ARP).

Moreover if $m = 1$, VRP become the Traveling Salesman Problem (TSP), that consists of find just one tour of all the vertices of G that are reached just one time. It is an NP-hard problem in combinatorial optimization [26]. Even though the problem is computationally difficult, a large number of heuristics and exact methods are known. Again, the ARP become Rural Postman Problem (RPP), whose problem is to reach each arcs required with a single tour. RPP is called Chinese Postman Problem (CPP) if postman tour or route inspection problem is to find a shortest closed path or circuit that visits every edge of a (connected) unoriented graph [25].

Eksioglu et al. propose a detailed list of subcategories for the generalized routing problem as:

1. Shortest path problem,
2. Chinese postman problem,
3. Rural postman problem,
4. Dial-a-ride service route problem,
5. Arc routing problem,
6. TSP,
7. VRP [27].

Chapter 3

Algorithms

Researchers on the VRP have proved that the VRP is a NP-complete combinatorial optimization problem [36].

Optimization problems arise in various disciplines such as engineering design, manufacturing system, economics etc. thus in view of the practical utility of optimization problems there is a need for efficient and robust computational algorithms which can solve optimization problems arising in different fields.

In most optimization problems there is more than one local solution. Therefore, it becomes very important to choose a good optimization method that will not be greedy and look only in the neighborhood of the best solution; because this will mislead the search process and leave it stuck at a local solution. However, the optimization algorithm should have a mechanism to balance between local and global search. There are multiple methods used to solve optimization problems of both the mathematical and combinatorial types. In fact, if the optimization problem is difficult or if the search space is large, it will become difficult to solve the optimization problem by using conventional mathematics.

Combinatorial generally means that the state space is discrete. Combinatorial optimization is widely applied in a number of areas nowadays. Combinatorial optimization problems (COP) are those problems that have a finite set of possible solutions. The best way to solve a combinatorial optimization problem is to check all the feasible solutions in the search space. However, checking all the feasible solutions is not always possible, especially when the search space is large. Thus, many heuristic and then meta-heuristic algorithms have been devised and modified to solve these problems. The meta-heuristic approaches are not guaranteed to find the optimal solution since they evaluate only a subset of the feasible solutions, but they try to explore different areas in the search space in a smart way to get a near-optimal solution in less cost and time [35].

3.1 Heuristic and Metaheuristic Algorithms introduction

The term heuristic is used for algorithms which find solutions among all possible ones, but they do not guarantee that the best will be found,therefore they may be considered as approximately and not accurate algorithms. These algorithms, usually find a solution close to the best one and they find it fast and easily. Sometimes these algorithms can be accurate, that is they actually find the best solution, but the algorithm is still called heuristic until this best solution is proven to be the best [64].

Some well know heuristic algorithms:

- the Nearest neighbour algorithm,

- the Cheapest insertion [25],
- the Clarke and Wright algorithm,
- the Sweep algorithm,
- the Christofides-Mingozzi-Toth two-phase algorithm [38].

The words of "meta" and "heuristic" are Greek where, "meta" is "higher level" or "beyond" and heuristics means "to nd", "to know", "to guide an investigation" or "to discover". Heuristics are methods to find good (near-) optimal solutions in a reasonable computational cost without guaranteeing feasibility or optimality. In other words, meta-heuristics are a set of intelligent strategies to enhance the efficiency of heuristic procedures.

Laporte and Osman defined a meta-heuristic as: "An iterative generation process which guides a subordinateheuristic by combining intelligently different concepts for exploring andexploiting the search space, learning strategies are used to structure information inorder to find efficiently near-optimal solutions." [37]

Some well know meta-heuristic algorithms:

- the Genetic algorithm,
- the Simulated Annealing,
- the Tabu Search,
- the Ant Colony algorithm,
- the Cuckoo Search [37].

For this study, the Nearest Neighbor Algorithm has been used and an other meta-heuristic algorithm has been developed, based on the Genetic Algorithm.

3.2 Nearest Neighbor Algorithm (NNA)

The Nearest Neighbour is heuristic algorithm applied on number of combinatorial optimization problems as: pattern recognition and computational geometry, interpolation for interpolating data, probability theory, solving the traveling salesman problem. Nearest Neighbor Algorithm is taken as one of the efficient computing method for vehicle routing problem.

3.2.1 Nearest Neighbor Algorithm Introduction

Let $X_1, X_2, ..., X_\mathrm{n}$, be a set of $NL - dimensional$ vectors called samples, where the X_is take values in a metric space upon which is defined a metric c (i.e. in geographical applications could be distance, etc...). Let X_i be the m^{th} nearest neighbor of X_j, and X_j be the n^{th} nearest neighbor of X_i [32]. Thus starting from an initial vector, the algorithm establishes a ranking of the rest of vectors, according to the value of the metric c measured for each pair (X_0, X_i).

3.2.2 Nearest Neighbor Algorithm for the Vehicle Routing Problem

The nearest-neighbor heuristic starts every route by finding the unrouted customer "closest" (in terms of a measure to be described later) to the depot. At every subsequent iteration, the heuristic searches for the customer "closest" to the last customer added to the route. This search is performed among all the customers who can feasibly (with respect to time windows, vehicle arrival time at the depot, and capacity constraints) be added to the end of the emerging route. A new route is started any time the search fails, unless there are no more customers to schedule [31].

3.3 Genetic Algorithm (GA)

A Genetic Algorithm applies the theory of the evolution developed by Darwin in the modellistic field. In simple terms, a GA is an algorithm that produce "children" results from the combination of "parents" data, and from some perturbations produced in the latter, in the same way of the combination of chromosomes from parents to children.

3.3.1 Genetic Algorithm introduction

Biological evolution is an appealing source of inspiration for addressing these problems. The fitness criteria continually change as creatures evolve, so evolution is searching a constantly changing set of possibilities. Searching for solutions in the face of changing conditions is precisely what is required for adaptive computer programs. Furthermore, evolution is a massively parallel search method: rather than working on one species at a time, evolution tests and changes millions of species in parallel. Finally, viewed from a high level the "rules" of evolution are remarkably simple: species evolve by means of random variation (via mutation, recombination, and other operators), followed by natural selection in which the fittest tend to survive and reproduce, thus propagating their genetic material to future generations. Yet these simple rules are responsible, in large part, for the marvelous complexity could be in the biosphere, including human intelligence [29].

The chromosomes in a GA population most often take the form of bit strings (i.e. strings of 1s and 0s); each bit position ("locus") in the chromosome has two possible values ("alleles"), 0 and 1. These biological terms are used in the spirit of analogy with real biology. The search takes place by processing populations of chromosomes, changing from one such population to another. The GA most often requires a "fitness function" that assigns a score (fitness) to each chromosome in the current population. The fitness of the chromosome depends on how well that chromosome solves the problem at hand.

As a simple example, one might want to maximize the real-valued one dimensional function

$$f(x) = x + |sin(32x)|$$

over all values of x between 0 and Π (Riolo, 1992). Here the candidate solutions are values of x, which can be encoded as bit strings representing real numbers. The fitness calculation translates a given bit string into a real number x and then evaluates the function at that value. The fitness of a string is the function value at that point [29].

The simplest form of genetic algorithm involves three types of operators: selection, crossover (single point), and mutation. The first operator selects chromosomes in the population for reproduction.

Crossover randomly chooses a locus and exchanges the subsequences before and after that locus between two chromosomes to create two offspring. For example, the strings 10000100 and 11111111 could be crossed over after the third locus in each to produce the two offspring 10011111 and 11100100. The crossover operator mimics biological recombination between two single chromosome.

The last operator randomly flips some of the bits in a chromosome. For example, the string 00000100 might be mutated in its second position to yield 01000100 [30].

3.3.2 A functional scheme

Given a clearly defined problem to be solved and a bit-string representation for candidate solutions, the simple GA works as follows:

1. Start with a randomly generated population of N L-bit chromosomes (candidate solutions to a problem).
2. Calculate the fitness $F(x)$ of each chromosome x in the population.
3. Repeat the following steps (a)-(c) until N offspring have been created:
   (a) Select a pair of parent chromosomes from the current population, with the probability of selection being an increasing function of fitness. Selection is done "with replacement," meaning that the same chromosome can be selected more than once to become a parent.
   (b) With probability pc (the crossover probability), cross over the pair at a randomly chosen point (chosen with uniform probability) to form two offspring. If no crossover takes place, form two offspring that are exact copies of their respective parents.
   (c) Mutate the two offspring at each locus with probability pm (the mutation probability), and place the resulting chromosomes in the new population.
4. Replace the current population with the new population.
5. Go to step 2.

(This assumes that N is even; if N is odd, one offspring can be discarded at random.) Each iteration of this process is called a "generation." A GA is typically iterated for anywhere from 50 to 500 or more generations, which is called a "run." At the end of a run, there are often one or more highly fit chromosomes in the population. Since randomness plays a large role in each run, two runs with different random number seeds will generally produce different detailed behavior. GA researchers often report statistics (such as the best fitness found and generation at which best fitness was found) averaged over many different runs of the GA on the same problem [29].

3.3.3 Genetic Algorithm for the Traveling Salesman Problem

Chatterjee et al., describe the operations of crossover, inversion and mutation on a ten-city problem. Consider two sequences of 10 random numbers, of $magnitude < N$ (say 100) and their corresponding ranks, ranks representing legal tours.

$$SolutionA : 34 - 43 - 39 - 32 - 28 - 47 - 25 - 33 - 52 - 31$$
$$RanksA : 6 - 8 - 7 - 4 - 2 - 9 - 1 - 5 - 10 - 3$$

$$SolutionB : 66 - 58 - 61 - 54 - 69 - 41 - 46 - 51 - 64 - 40$$
$$RanksB : 9 - 6 - 7 - 5 - 10 - 2 - 3 - 4 - 8 - 1$$

Suppose the crossover is applied at position number 5 (chosen at random). Thus the two children of this crossover are given by:

$$A^{\mathrm{C}} : 34 - 43 - 39 - 32 - 28 - 41 - 46 - 51 - 64 - 40$$

$$RanksA^{\mathrm{C}} : 3-7-4-2-1-6-8-9-10-5$$

$$B^{\mathrm{C}} : 66-58-61-54-69-47-25-33-52-31$$
$$RanksB^{\mathrm{C}} : 9-7-8-6-10-4-1-3-5-2$$

Ties are broken randomly so that the paths chosen are legal. The value of N must be much larger than the number of cities to reduce ties during cross-over. In this scheme, the cities have a fixed parameter (or gene) positions in the String and the order in which they are visited is determined by sorting on the parameter values. For inversion and mutation, we can directly work with the ranks. For example, suppose an inversion takes place for solution A between the third and the sixth positions. The inverted route A^{I} (for solution A) is given as: $A^{\mathrm{I}} : 6-8-9-2-4-7-1-5-10-3$. A mutation in this scheme is defined as an exchange of ranks between a pair of cities. Thus if solution A is to undergo a mutation between the third and the eighth locations, the mutated solution A^{M} is given by $A^{\mathrm{M}} : 6-8-5-4-2-9-1-7-10-3$. We note that a mutation of this type is arbitrary and can be defined in many other ways [28].

Chapter 4

VRP in the city of Padova

The goal of this work is the optimization of the vehicle routing for the drop-off MSW collection system, to solve the VRP. In detail the aim is minimized the total distance covered by the collection truck, taking into account its total capacity. For this reason an algorithm based on the Genetic Algorithm have been developed, and it has been compared with the algorithm for VRP created by Sas Wahid HamzahAll [56] based on the Nearest Neighbor Algorithm. The study have been applied in some zones of the city of Padova.

4.1 Methodology

In this paragraph a rapid overview has been done on the instruments used in the study. The data have been collected with the My Tracks app installed on a Samsung GT-S6500 with Android version 2.3.6. My Tracks app is a GPS tracker (Fig. 4.2) that allows to add indicators and photos. Therefore each waste collection point has been sampled by photos and position (Fig. 4.3).

Then the data have been elaborated with QuantumGIS to extract the coordinates of the waste collection points. QGIS is an Open Source Geographic Information System, currently runs on most Unix platforms, Windows, and OS X. QGIS is developed using the Qt toolkit and $C++$. This means that it feels snappy and has a pleasing, easy-to-use graphical user interface (GUI). QGIS aims to be a user-friendly GIS (Geographic Information System), providing common functions and features. The initial goal of the project was to provide a GIS data viewer and has reached the point in its evolution where it is being used by many for their daily GIS data-viewing needs [65].

Finally the data have been elaborated by algorithms implemented on Matlab software. MATLAB ("MATrix LABoratory") is a tool for numerical computation and visualization. The basic data element is a matrix, so if you need a program that manipulates array-based data it is generally fast to write and run in MATLAB [66].

The simulations have been ran on a Dell Inspiron-1564 with a processor Intel Core i3 CPU, 2.13 GHz that support an OS Windows 7.

4.2 Data collection

In the city of Padova 65 waste collection points have been collected in the Centre zone and in the North zone. The surveys have been done in the areas of:

- Arcella district,
- Piazza Eremitani,
- university campus (via Loredan, via Marzolo),
- Piazza delle Erbe,
- Piazza della Frutta,
- Prato della Valle,
- Piazza del Santo,
- Via San Francesco,
- Via del Risorgimento,
- Via Roma.

The type of MSW has been characterized for each collection point. The categories collected include:

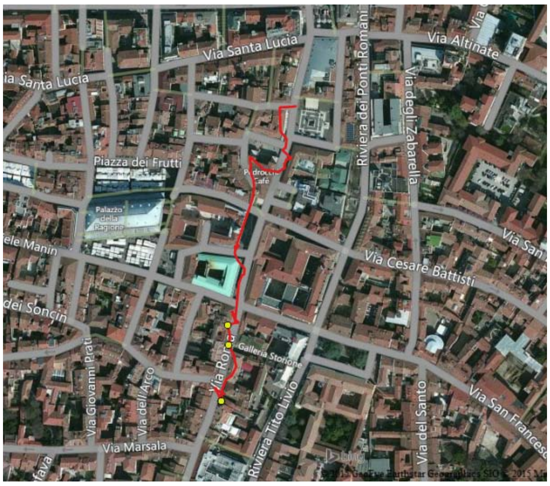
- unsorted waste,
- multi-material(plastic, metal, glass),
- putrescible material,
- paper and carton,
- plastic and metal,
- glass.

Figure 4.1: An example of complete waste collection point. It is the 17th indicator, geo-referenced in Prato della Valle with planar coordinates 255446.949283, 5032080.543377 WGS84/UTM zone 33N

(a) GPS track in Arcella district.

(b) GPS track in Piazza degli Eremitani and university campus.

(c) GPS track in Via Risorgimento.



(d) GPS track in Via Roma.

(e) GPS track in Prato della Valle, Piazza del Santo, Via San Francesco.

(f) GPS track in Piazza delle Erbe and Piazza della Frutta.

Figure 4.2: GPS tracks collected with the My Tracks app installed on a Samsung GT-S6500 with Android version 2.3.6. The app uses the WGS 84 ellipsoidal reference system. The data have been surveyed in the period between September 25th and October 5th, 2015.

Figure 4.3: 65 waste collection points surveyed and elaborated with QGIS. Each waste collection point has been sampled by photos and position.

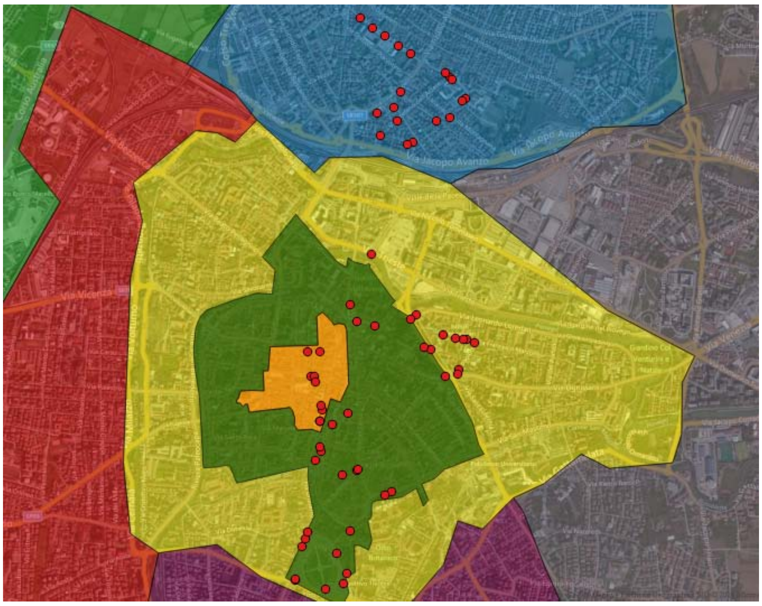


Figure 4.4: Waste collection points classified according to single area of relevance. In particular the 65 collection points, are surveyed in the Centre (yellow) and North (blue) areas.

4.3 Algorithm construction and simulation

At the beginning of the entire work, the idea was to implement the NNA based algorithm wrote by Sas Wahid HamzahAll, and use this for applications and innovations in the city of Padova. Then the finding of the algorithm wrote by Joseph Kirk to solve the TSP, has inspired the construction of a double application of GA based algorithm.

4.3.1 Nearest Neighbor Algorithm

The first method used to solve the VRP, has been implemented on Matlab by Sas Wahid HamzahAll [56]. It can be described by the pseudo-code Algorithm 4.3.2 and Algorithm 4.3.1.

Algorithm 4.3.1: NEAREST NEIGHBOR ALGORITHM *part 2 Route selection*(**Input:**$Problem, Route, Capacity$.**Output:**$NumberOfVehicles, TotalDistance, RouteSets$.)

$$
\begin{array}{l}
Column_1(Problem) = OrderedIndex \\
Column_2(Problem) = Xcoordinates \\
Column_3(Problem) = Ycoordinates \\
Column_4(Problem) = Demand \\
Capacity = 20 \\
Route \equiv [[RouteNow]_1, ..., [RouteNow]_{\mathrm{k\ -\ 1}}, CandidateToInsert, 1] \\
Route(aa \times bb) \\
\textbf{for } hh = 1 \textbf{ to } aa \\
\quad \textbf{do} \left\{ \begin{array}{l}
ee = 1 \textbf{ and } cc = 2 \\
\textbf{while } hh \leqslant aa \textbf{ and } cc < bb \\
\quad \textbf{do} \left\{ \begin{array}{l}
ff = 2 \\
Routes \equiv [v_1, ..., v_{\mathrm{f}}] \textbf{ and } f \in N \\
[v_1]_{\mathrm{hh,\ ee}} \\
[Load]_{\mathrm{hh,\ ee}} = 0 \\
\textbf{while } [Load]_{\mathrm{hh,\ ee}} \leqslant Capacity \textbf{ and } cc < bb \\
\quad \textbf{do} \left\{ \begin{array}{l}
\textbf{if } [Load]_{\mathrm{hh,\ ee}} + [Demand]_{[\mathrm{Route}]_{\mathrm{hh,\ cc}}} > Capacity \\
\quad \textbf{then } break \\
[v_{\mathrm{ff}}]_{\mathrm{hh,\ ee}} \equiv [Route]_{\mathrm{hh,\ cc}} \\
[Load]_{\mathrm{hh,\ ee}} \equiv [Load]_{\mathrm{hh,\ ee}} + [Demand]_{[\mathrm{Route}]_{\mathrm{hh,\ cc}}} \\
cc \equiv cc + 1 \\
ff \equiv ff + 1
\end{array} \right. \\
[v_{\mathrm{ff}}]_{\mathrm{hh,\ ee}} = 1 \\
ee = ee + 1
\end{array} \right. \\
hh \equiv hh + 1
\end{array} \right. \\
Routes(NumberOfVehicles \times m) \\
RouteSet \equiv Routes \\
nn \equiv NumberOfVehicles \\
\textbf{for } RouteNumber = 1 \textbf{ to } nn \\
\quad \textbf{do} \left\{ \begin{array}{l}
Route\_VRP(1 \times r) \equiv v_{\mathrm{RouteNumber}} \\
jum = 0 \\
\textbf{for } t = 1 \textbf{ to } (r-1) \\
\quad \textbf{do} \left\{ \begin{array}{l}
subrute \equiv jum + [d]_{[\mathrm{Route\_VRP}]_{\mathrm{t}},\ [\mathrm{Route\_VRP}]_{\mathrm{t\ +\ 1}}} \\
jum \equiv subrute
\end{array} \right. \\
[DistanceSets]_{\mathrm{RouteNumber}} \equiv jum \\
TotalDistance \equiv \sum_{RouteNumber} [DistanceSets]_{\mathrm{RouteNumber}}
\end{array} \right. \\
TotalDistance \equiv Totaldistance
\end{array}
$$

Algorithm 4.3.2: NEAREST NEIGHBOR ALGORITHM *part 1*(**Input:**$Problem, Capacity, xy00, xy00UTM$.**Output:**$Routes, NumberOfRoutes, TotalDistance.$)

$Column_1(Problem) = OrderedIndex$
$Column_2(Problem) = Xcoordinates$
$Column_3(Problem) = Ycoordinates$
$Column_4(Problem) = Demand$
$Capacity = 20$
$xy00 = [0, 0]$
$xy00UTM = [259051.864391, 5033215.737004]$
$RouteNow(w \times k) = [1, 1]$
$UnservedOutlet(1 \times n) = [2, ..., 26]$
while $n > 0$
do
- **for** $NumberOfOutletNotYetInserted = 1$ **to** n
  do
  - $CandidateToInsert \equiv [UnservedOutlet]_{\mathrm{N}}umberOfOutletNotYetInserted$
  - $Route \equiv [[RouteNow]_1, ..., [RouteNow]_{\mathrm{k\text{ - }1}}, CandidateToInsert, 1]$
  - ALGORITHM 4.3.1(**Input:**$Problem, Route, Capacity.$ **Output:**$NumberOfVehicles, TotalDistance, RouteSets.$)
  - **if** $NumberOfVehivles > 1$
    then $\begin{cases} FeasibleOutletToInsert \equiv FeasibleOutletToInsert \\ Distance \equiv Distance \end{cases}$
    else $\begin{cases} FeasibleOutletToInsert \equiv \\ \equiv [FeasibleOutletToInsert, CandidateToInsert] \\ Distance \equiv [Distance, TotalDistance] \end{cases}$
- $FeasibleOutletToInsert(mm, nn)$
- **if** $mm \geqslant 1$
  then
  - $ShortestDist \equiv min\{[Distance]_{\mathrm{j}}\} \equiv [Distance]_{\mathrm{i}}$
  - $best\_FeasibleOutletToInsert \equiv$
    $\equiv [FeasibleOutletToInsert]_{\mathrm{i}}$
  - **if** $[best\_FeasibleOutletToInsert]_{\mathrm{j}} \equiv [UnservedOutlet]_{\mathrm{z}}$
    then $\begin{cases} [UnservedOutlet]_{\mathrm{z}} = 0 \\ UnservedOutlet \equiv [a_{\mathrm{w}}] \textbf{ and } 2 \leqslant a_{\mathrm{w}} \leqslant 26 \end{cases}$
  - $RouteNow \equiv$
    $\equiv [[RoutNow_{\mathrm{k}}], ..., [RoutNow_{\mathrm{k\text{ - }1}}], best\_FeasibleOutletToInsert, 1]$
- **if** $mm = 0$ **or** $n = 0$
  then $\begin{cases} RouteFixed \equiv RouteNow \\ v_{\mathrm{f\,+\,1}} \equiv RouteFixed \\ RouteNow \equiv [1, 1] \end{cases}$

$NumberOfRoutes \equiv nn$
for $RoutesIndex = 1$ **to** $NumberOfRoutes$
do $\begin{cases} v_{\mathrm{RoutesIndex}}(1 \times kk)NumberOfOutletsInRoute \equiv kk \\ SetsOfNumberOfOutletsInRoute \equiv \\ \equiv [SetsOfNumberOfOutletsInRoute, NumberOfOutletsInRoute] \end{cases}$
if $[SetsOfNumberOfOutletsInRoute]_{\mathrm{ttt}} \leqslant 2$
then $\{[Routes]_{\mathrm{ttt}}(n \times m) = 0$
for $RouteIndex = 1$ **to** n
do
- $Route\_VRP(1 \times r) \equiv v_{\mathrm{RouteIndex}}$
- $jum = 0$**for** $t = 1$ **to** $(r - 1)$
  do $\begin{cases} subrute \equiv jum + d_{[\mathrm{Route\_VRP}]_{\mathrm{t}},\ [\mathrm{Route\_VRP}]_{\mathrm{t\,+\,1}}} \\ jum = subrute \end{cases}$
- $Row_{\mathrm{RouteIndex}}(DistanceSets) \equiv \sum_{RouteIndex} [DistanceSets]_{\mathrm{RouteIndex}}$

4.3.2 Three-phase Algorithm

The second method has been developed with a double application of the Genetic Algorithm (GA). The algorithm (Algorithm 4.3.4) wrote with Matlab software, can be simplified in three parts. For this reason the algorithm has been called Three-phase Algorithm. One step to improve the chose of the cluster of the waste collection points that can be collected with one route (Algorithm 4.3.6), and the other to choose the shortest route that reach each point just one time in a single cluster (Algorithm 4.3.3). Both are based on the Genetic Algorithm. The first application is in the Route Selection (Algorithm 4.3.5), to choose the clusters.

The part of the code related to the GA that has been used in the script, has been recovered and then adapted, from the code wrote by Joseph Kirk [57]. The pseudo-code of the latter is the Algorithm 4.3.3 to solve the TSP.

The GA uses operations (crossover, inversion, mutation and others) to generate a new population with the "DNA" from the parents population, as described above in the Algorithms chapter. In particular have been used:

- inversion,
- mutation,
- slide.

The pseudo-code of the Three-phase Algorithm that has been generated, is the Algorithm 4.3.4.

Algorithm 4.3.3: GENETIC ALGORITHM(**Input:**$xyz, SingleCluster.$ **Output:**$optRoute, minDist.$)

$Row_{\mathrm{i}}(Problem(nnx4)) \equiv [i, x_{\mathrm{i}}, y_{\mathrm{i}}, [Demand]_{\mathrm{i}}]$
$1 \leqslant i, j \leqslant n$
$popSize \equiv n$
if $i = 1$
  then $Row_1(POP) = (1, 2, ..., nn)$
  else if $i > 1$
  then $\begin{cases} 1 \leqslant [POP]_{\mathrm{i,j}} \leqslant nn \\ [POP]_{\mathrm{i,j}} \neq [POP]_{\mathrm{i,k}} \Leftrightarrow j \neq k \end{cases}$
$GlobalMin = \infty$
for $iter = 1$ **to** 10000
do
  for $p = 1$ **to** n
  do
    $g \equiv [dmat]_{[\mathrm{POP}]_{(\mathrm{p,popSize})}, [\mathrm{POP}]_{(\mathrm{p,1})}}$
    for $k = 2$ **to** nn
    do
      if $k = 2$
        then $d_{\mathrm{k}} \equiv d_2 \equiv g + [dmat]_{[\mathrm{POP}]_{(\mathrm{p,k-1})}, [\mathrm{POP}]_{(\mathrm{p,k})}}$
        else if $k > 2$
        then $d_{\mathrm{k}}(k) \equiv d(k-1) + [dmat]_{[\mathrm{POP}]_{(\mathrm{p,k-1})}, [\mathrm{POP}]_{(\mathrm{p,k})}}$
    $totalDist \equiv [d_{\mathrm{k}}(p = 1), ..., d_{\mathrm{k}}(p = nn)]$
  $[minDist]_{\mathrm{e}} \equiv min\{[totalDist]_{\mathrm{e}}\}$
  $1 \leqslant e \leqslant popSize$
  if $[minDist]_{\mathrm{e}} < globalMin$
    then $\begin{cases} globalMin \equiv [minDist]_{\mathrm{e}} \\ optRoute \equiv Row_{\mathrm{e}}(POP) \end{cases}$
  $1 \leqslant [randomOrder]_{\mathrm{pp,zz}} \leqslant popSize$ **and** $[randomOrder]_{\mathrm{pp,zz}} \neq$
  $\neq [randomOrder]_{\mathrm{pp,kk}} \Leftrightarrow pp \neq zz$
  for $p = 4$ **to** $popSize$, **for each** 4
  do
    for $x = 1$ **to** 4
      do $Row_{\mathrm{x}}(rtes) \equiv Row_{[\mathrm{randomOrder}]_{\mathrm{p\ -\ (4\ -\ x)}}}(POP)$
    for $v = 1$ **to** 4
      do $\begin{cases} [dists]_{\mathrm{v}} \equiv [totalDist]_{[\mathrm{randomOrder}]_{\mathrm{p-(4-v)}}} \\ [ignore] \equiv min[dists] \equiv [dists]_{\mathrm{h}} \\ bestOf4Route \equiv Row_{\mathrm{h}}(rtes) \end{cases}$
    $\begin{cases} routeInsertionPoints \equiv [c, d] \\ 0 \leqslant c, d \leqslant nn \\ c \neq d \text{ and } c < d \end{cases}$
    $tmpPOP \equiv$
      $Row_1(tmpPOP) \equiv bestOf4Route$
      for $l \equiv c$ **to** d
        do $[tmpPOP]_{2,\mathrm{l}} \equiv [tmpPOP]_{2,(\mathrm{d\ +\ 1)\ -\ l}}$
      $[tmpPOP]_{3,\ \mathrm{c}} \equiv [tmpPOP]_{3,\ \mathrm{d}}$ **and** $[tmpPOP]_{3,\ \mathrm{d}} \equiv$
      $\equiv [tmpPOP]_{3,\ \mathrm{c}}$
        $[tmpPOP]_{4,\ \mathrm{c}} \equiv [tmpPOP]_{4,\ \mathrm{d}}$
        for $ll \equiv c + 1$ **to** d
          do $[tmpPOP]_{4,\ \mathrm{ll}} \equiv [tmpPOP]_{4,\ \mathrm{ll\ -\ 1}}$
    for $lll = 1$ **to** 4
      do $Row_{\mathrm{p\ -\ (4\ -\ lll)}}(newPOP) \equiv Row_{\mathrm{l}}ll(tmpPOP)$
  $POP \equiv newPOP$

Algorithm 4.3.4: Three-phase Algorithm *part 1*(**Input:**$Problem, Capacity, NumOfIterations, xy00, xy00UTM$.**Output:**$TotalMin, TotalRoute$.)

$$
\begin{aligned}
&Column_1(Problem) = OrderedIndex\\
&Column_2(Problem) = Xcoordinates\\
&Column_3(Problem) = Ycoordinates\\
&Column_4(Problem) = Demand\\
&Capacity = 20\\
&xy00 = [0,0]\\
&xy00UTM = [259051.864391, 5033215.737004]\\
&TotalMin = \inf\\
&NumOfIterations = X\\
&PopulationDensity = N\\
&popSize \equiv nRow_{\mathrm{i}}(Problem(nnx4)) \equiv [i, x_{\mathrm{i}}, y_{\mathrm{i}}, [Demand]_{\mathrm{i}}], 1 \leqslant i \leqslant nn\\
&\textbf{for } iter = 1 \textbf{ to } NumOfIterations
\end{aligned}
$$

$$
\textbf{do}\begin{cases}
\textbf{for } f = 1 \textbf{ to } PopulationDensity\\
\textbf{do}\begin{cases}
\textbf{if } iter = 1\\
\textbf{then}\begin{cases}
\textbf{if } f = 1\\
\quad\textbf{then}\begin{cases} Row_{\mathrm{g}}(IndexDemand) \equiv [g, [Demand]_{\mathrm{g}}]\\ NewNewIndex \equiv [1, ..., nn]\end{cases}\\
\quad\textbf{else}\begin{cases} NewNewIndex \equiv [a_1, ..., a_{\mathrm{nn}}]\\ 1 \leqslant a_{\mathrm{m}} \leqslant nn\\ a_{\mathrm{m}} \neq a_{\mathrm{j}}\\ 1 \leqslant m, j \leqslant nn \Leftrightarrow m \neq j\\ Row_{\mathrm{i}}(IndexDemand) \equiv\\ \equiv [NewNewIndex_{\mathrm{i}}, [Problem]_{[\mathrm{NewNewIndex}]_{\mathrm{i}},\, 4}]\end{cases}\\
NewIndex = TotalIndex
\end{cases}\\
\textbf{else}\begin{cases} NewNewIndex \equiv Row_{\mathrm{f}}(TotalIndex)\\ Row_{\mathrm{jj}}(IndexDemand) \equiv\\ \equiv [[NewNewIndex]_{\mathrm{jj}}, [Problem]_{[\mathrm{NewNewIndex}]_{\mathrm{jj}},\, 4}]\\ 1 \leqslant jj \leqslant nn\end{cases}\\
\text{ALGORITHM 4.3.5}(\textbf{Input:}Problem, Capacity, IndexDemand, Demand.\\
\textbf{Output:}NewCluster.)\\
NewCluster(kxmm)\\
\textbf{for } e = 1 \textbf{ to } k\\
\quad\textbf{do}\begin{cases} RowCluster(kxs) \equiv Row_{\mathrm{e}}(NewCluster) \equiv [a_1, ..., a_{\mathrm{s}}]\\ \textbf{if } [RowCluster]_{\mathrm{e,\, s}} = 0\\ \quad\textbf{then } RowCluster(ssxkk) \equiv [a_1, ..., a_{\mathrm{s\text{-}1}}]\end{cases}\\
\textbf{for } ee = 1 \textbf{ to } kk\\
\quad\textbf{do}\begin{cases} SingleCluster(1xff) \equiv [RowCluster]_{1,\, \mathrm{ee}} \equiv [a_1, ..., a_f f]\\ SingleCluster \equiv [a_{\mathrm{gg}}], 2 \leqslant gg \leqslant ff\\ Row_{\mathrm{b}}(IndexAndSingleCluster) \equiv [ff - 1, a_{\mathrm{b\text{-}1}}]\\ Row_1(xyz) \equiv [0,0]\\ \textbf{for } vv = 2 \textbf{ to } ff - 1\\ \quad\textbf{do}\begin{cases} Row_{\mathrm{vv}}(xyz) \equiv\\ \equiv [[Problem]_{[\mathrm{SingleCluster}]_{1,\, \mathrm{vv}},\, 2}, [Problem]_{[\mathrm{SingleCluster}]_{(1,\, \mathrm{vv})},\, 3}]\end{cases}\\ \text{ALGORITHM 4.3.3}(\textbf{Input:}xyz, SingleCluster.\\ \textbf{Output:}optRoute, minDist.)\\ [RowDist]_{\mathrm{ee},\, 1} \equiv minDist\\ optRoute(1, bb)\\ \textbf{for } y = 1 \textbf{ to } bb\\ \quad\textbf{do}\ \{Column_{\mathrm{y}}(realRoute) \equiv [SingleCluster]_{[\mathrm{optRoute}]_{\mathrm{y}}}\\ [RouteOfnn]_{\mathrm{e,\, ee}} \equiv realRoute\end{cases}\\
[RowSumDist]_{\mathrm{e}} \equiv \sum_w [RowDist]_{\mathrm{w}}\\
minRowDist = min(RowSumDist) = [RowSumDist]_{\mathrm{z}}\\
\textbf{if } minRowDist < TotalMin\\
\quad\textbf{then}\begin{cases} TotalMin \equiv minRowDist\\ TotalRoute \equiv Row_{\mathrm{z}}(RouteOfnn)\end{cases}
\end{cases}\\
\text{ALGORITHM 4.3.6}(\textbf{Input:}bestRowDist, NewIndex, t.\\
\textbf{Output:}TotalIndex)
\end{cases}
$$

Algorithm 4.3.5: THREE-PHASE ALGORITHM *part 2 Route selection*(**Input:**$Problem, Capacity$ $, IndexDemand, Demand.$**Output:**$NewCluster.$)

$IndexDemand \equiv [Column_1(Problem), Column_4(Problem)]$
$Column_1(IndexDemand(nixmi)) \equiv NotYetUsed$
$RouteNow = [0, 0]$
$i = 1$
$bbi = 1$
$ffi = 1$
while $CountCurrent \leqslant ni$

$$\textbf{do} \begin{cases} Route \equiv [RouteNow, [NotYetUsed]_{\mathrm{ij}}, 0] \\ Load \equiv Load + [Demand]_{[\mathrm{NotYetUsed}]_{\mathrm{ij}}} \\ \textbf{if } Load \leqslant Capacity \\ \textbf{then} \begin{cases} RouteNow \equiv Route \equiv [c_{\mathrm{ji}}] \equiv [c_1, ..., c_{\mathrm{vi}}] \\ Used \equiv [c_2, ..., c_{(\mathrm{vi - 1})}] \\ NotYetUsed(1xhi) \equiv [d_{\mathrm{iij}}, ..., d_{\mathrm{ij - 1}}, d_{\mathrm{ij}}, d_{\mathrm{ij + 1}}, ..., d_{\mathrm{jji}}] \\ NotYetUsed \equiv [d_{\mathrm{iij}}, ..., d_{\mathrm{ij - 1}}, d_{\mathrm{ij + 1}}, ..., d_{\mathrm{jji}}] \\ ij = 1 \textbf{if } hi = 0 \\ \textbf{then} \begin{cases} [Routes(gixwi)]_{\mathrm{CountCurrent,bbi}} \equiv Route \\ RouteNow \equiv StartRoute \\ NotYetUsed \equiv IndexDemand \\ Load = 0 \\ ffi \equiv fi + 1 \\ CountCurrent \equiv CountCurrent + 1 \\ bbi = 1 \\ ij \equiv CountCurrent \end{cases} \\ \textbf{else if} \begin{cases} Route \equiv [[c_1, ..., c_{\mathrm{vi - 1}}], 0] \\ [Routes]_{\mathrm{CountCurrent,bbi}} \equiv Route \\ RouteNow \equiv StartRoute \\ Load = 0 \\ ffi = 1 \\ ij = 1 \\ bbi \equiv bbi + 1 \end{cases} \end{cases} \end{cases}$$

for $hhi = 1$ **to** gi

$$\textbf{do} \begin{cases} \textbf{for } ggi = 1 \textbf{ to } wi \\ \textbf{do} \begin{cases} \textbf{if } [Cluster]_{\mathrm{hhi,ggi}} \equiv [Cluster]_{\mathrm{nni,mmi}} \\ \textbf{then} \begin{cases} 1 \leqslant nni \leqslant gi \textbf{ and } 1 \leqslant mmi \leqslant wi \\ Row_{\mathrm{hhi}}(Cluster) \equiv [\varnothing] \end{cases} \end{cases} \end{cases}$$

Algorithm 4.3.6: THREE-PHASE ALGORITHM *part 3 GA operations*(**Input:**$bestRowDist$, $NewIndex, t$.**Output:**$TotalIndex$)

$$
\begin{array}{l}
\textbf{for } t = 1 \textbf{ to } PopulationDensity/4\\
\textbf{do} \begin{cases}
[best4RowDist] \equiv min[bestRowDist] \equiv [bestRowDist]_{\mathrm{v}}\\
[best4RowDist]_{\mathrm{t}} \equiv best4RowDistRow_{\mathrm{t}}(LastIndex) \equiv Row_{\mathrm{v}}(NewIndex)\\
\textbf{if } \begin{cases} 0 \leqslant c, d \leqslant nn \\ c \neq d \textbf{ and } c < d \end{cases}\\
\quad \textbf{then } routeInsertionPoints \equiv [c, d]\\
RowIndex \equiv \begin{cases}
Row_1(RowIndex) \equiv bestOf4Route\\
\textbf{for } l \equiv c \textbf{ to } d\\
\quad \textbf{do } [RowIndex]_{2,\mathrm{l}} \equiv [RowIndex]_{2,(\mathrm{d}\,+\,1)\,-\,\mathrm{l}}\\
[RowIndex]_{3,\ \mathrm{c}} \equiv [RowIndex]_{3,\ \mathrm{d}} \textbf{ and } [RowIndex]_{3,\ \mathrm{d}} \equiv\\
\equiv [RowIndex]_{3,\ \mathrm{c}}\\
\begin{cases}
[RowIndex]_{4,\ \mathrm{c}} \equiv [RowIndex]_{4,\ \mathrm{d}}\\
\textbf{for } ll \equiv c+1 \textbf{ to } d\\
\quad \textbf{do } [RowIndex]_{4,\ \mathrm{ll}} \equiv [RowIndex]_{4,\ \mathrm{ll}\,-\,1}
\end{cases}\\
first \equiv t * 4 - 3\\
second \equiv t * 4\\
\textbf{for } h = first \textbf{ to } second\\
\quad \textbf{do } Row_{\mathrm{h}}(TotalIndex) \equiv RowIndex
\end{cases}
\end{cases}
\end{array}
$$

4.3.3 Hypothesis

The simulations of the NNA and Three-phase Algorithm, have been done at first fixing the following assumptions:

1. Only the collection points collected with the drop-off system have been considered. It means that the Orange zone of the Centre is excluded.
2. Only the bins for collection of the unsorted waste have been considered. Therefore from the 65 collection points that have been surveyed, only 43 have been chosen.
3. All of the bins have been considered at the moment of the pick-up in the truck, full for 3/4 of their total volume.
4. The trucks that have been used for the waste collection and transport, are not compactor, and they have a volume capacity of 20 m^3.
5. In the simulations, finally the waste are located in a deposit situated in the industrial area of Padova.
6. WGS84/UTM zone 33N is the global reference system used to have the cartesian coordinates that have been used in the simulations.

The first and second hypothesis have been done to better homogenize the data surveyed. According to the districts, the dimensions of the trucks used, are related to the width of the roads. But for a single simulation, can be considered, just one kind of vehicle. Otherwise more than one simulations have been done, dividing the points collected in subcategories (areas of the city: Orange, Yellow, Green). 20 m^3 is a real, intermediate dimension between 10 and 22 m^3, that is the volume of the truck of the collection company [34]. The fourth hypothesis have been done because in this work, the final destinations of the waste, are not important. Therefore a deposit that could be everything(transfer station, landfill, private company, incenerator, composting plant, etc...), is the best solution for these simulations. The last assumption is a solution to limit the errors of planar coordinates, from a standard spheroidal reference surface (WGS84) where the GPS coordinates have been collected.

4.3.4 Nearest Neighbor Algorithm simulation and results

The simulation of the waste collection and transport based on the Nearest Neighbor Algorithm, has been processed from the input data shown in Tab. 4.1 and from the previous hypothesis.

The Tab. 4.2 shows the characterization of the deposit.

All the routes that result from the simulation, start and finish from the deposit (id 44). Therefore the Tab. 4.3 shown each route from the row of the id 44 to the row of the id 44.

5 routes and a total distance of 39.5 Km have been obtained.

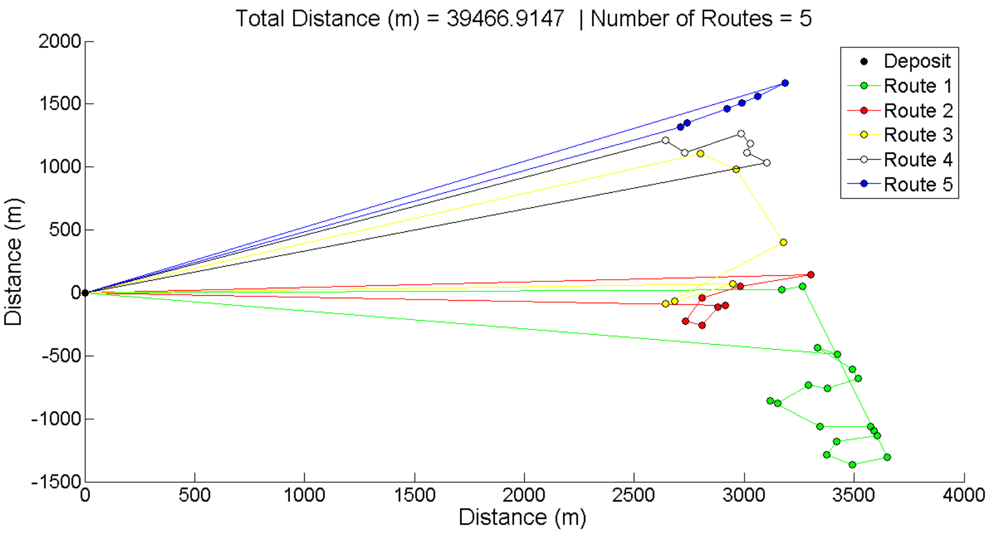


Figure 4.5: The shortest routes plan results from the Nearest Neighbor Algorithm simulation implemented on Matlab. From the deposit, five route are necessary to reach each waste collection point with a truck capacity of 20 m^3.

Figure 4.6: NNA routes plan on QuantumGIS platform.

Table 4.1: Characterization of the unsorted waste collection points. Bins positions, volume and number in the collection point have been characterized. In the areas surveyed just bins of 1 and 2.4 m^3 are located. Reference system is WGS 84/UTM 33N

id	Coordinate x	Coordinate y	Bins demand (lt)	n° bins for unsorted waste
1	255628.583800869	5032726.808474400	1000	1
2	255935.337870106	5032357.086090620	1000	2
3	255558.702458421	5031848.005041690	1000	1
4	255898.945508198	5032336.245989560	1000	1
5	255673.673792253	5032455.027433400	1000	1
6	255533.522407402	5032538.410363300	1000	2
7	256104.299103770	5033289.230643370	2400	2
8	256409.377579713	5033131.223553460	2400	2
9	256366.352106577	5033148.625915140	2400	2
10	255881.750174129	5033240.180559450	1000	1
11	255785.826530948	5033269.238715940	1000	3
12	255749.912334742	5033357.616949720	1000	1
13	255873.196458077	5033619.410464310	2400	2
14	255630.854341550	5032038.684513050	1000	1
15	255706.712233326	5032155.343681460	1000	1
16	255675.372616814	5031927.586545350	1000	1
17	256244.288081726	5033178.380723650	2400	3
18	255446.949282576	5032080.543377010	1000	1
19	255400.936652359	5031908.097526770	1000	1
20	255948.069021440	5034250.244479690	2400	2
21	256040.288863684	5034326.884370310	2400	2
22	256023.962045765	5034400.838618200	2400	2
23	256128.482257812	5034677.493612770	2400	2
24	256065.239606522	5034480.234596910	2400	2
25	255990.595664290	5034779.640914700	2400	2
26	256061.557482728	5034727.598430280	2400	1
27	255864.117118510	5034882.906177100	2400	1
28	255462.669529721	5032119.986247770	1000	1
29	255478.382061578	5032155.201535770	1000	1
30	256310.541395513	5034569.743184430	2400	1
31	256409.805148712	5034429.046381020	2400	1
32	256343.624034497	5034535.861678260	2400	3
33	256323.512120710	5034332.311882570	2400	2
34	256250.376014123	5034319.347926460	2400	1
35	256069.804336837	5033267.764357700	1000	1
36	256090.544432397	5034200.434097270	2400	2
37	256319.746563686	5032990.427849020	2400	4
38	256173.487694673	5033105.142617960	2400	2
39	256138.197891210	5033114.632764470	1000	1
40	256244.534849801	5032959.697884490	1000	1
41	255560.651586478	5032609.286239500	1000	3
42	255720.399256877	5032783.322989720	1000	2
43	255759.693089803	5032482.283847840	1000	2

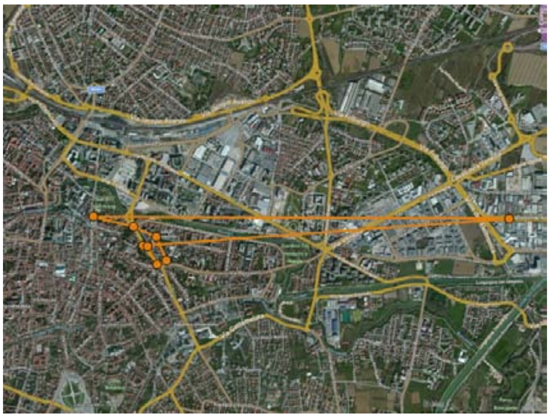
Table 4.2: Deposit characterization. Deposit has been located in the area of productivity but have not been follow other criteria due to the little importance in the aim of that study.

id	Coordinate x	Coordinate y
44	259051,864391	5033215,737004

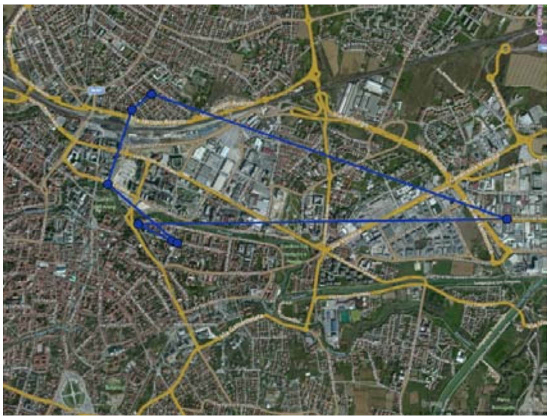
Table 4.3: Routes description from NNA. The first column shows the id of the single waste collection point, the second column shows the route of membership, the third column gives the sequence of the waste collection points in each route, the fourth and fifth the planar coordinates (WGS 84/UTM zone 33N) of the collection points.

id	n° of single route	Order in single route	Coordinate x	Coordinate y
44	1	1	259051.864391000	5033215.737003990
1	1	2	255628.583800869	5032726.808474400
42	1	3	255720.399256876	5032783.322989710
41	1	4	255560.651586477	5032609.286239500
6	1	5	255533.522407401	5032538.410363290
5	1	6	255673.673792252	5032455.027433400
43	1	7	255759.693089803	5032482.283847830
4	1	8	255898.945508198	5032336.245989560
2	1	9	255935.337870106	5032357.086090610
15	1	10	255706.712233325	5032155.343681460
29	1	11	255478.382061578	5032155.201535760
28	1	12	255462.669529721	5032119.986247760
18	1	13	255446.949282576	5032080.543377010
14	1	14	255630.854341550	5032038.684513050
16	1	15	255675.372616814	5031927.586545350
3	1	16	255558.702458421	5031848.005041680
19	1	17	255400.936652359	5031908.097526770
11	1	18	255785.826530947	5033269.238715940
10	1	19	255881.750174129	5033240.180559440
44	1	20	259051.864391000	5033215.737003990
44	2	1	259051.864391000	5033215.737003990
39	2	2	256138.197891210	5033114.632764460
38	2	3	256173.487694673	5033105.142617960
40	2	4	256244.534849801	5032959.697884480
37	2	5	256319.746563686	5032990.427849010
17	2	6	256244.288081726	5033178.380723640
35	2	7	256069.804336837	5033267.764357700
12	2	8	255749.912334742	5033357.616949720
44	2	9	259051.864391000	5033215.737003990
44	3	1	259051.864391000	5033215.737003990
7	3	2	256104.299103770	5033289.230643370
9	3	3	256366.352106576	5033148.625915130
8	3	4	256409.377579713	5033131.223553460
13	3	5	255873.196458077	5033619.410464310
36	3	6	256090.544432397	5034200.434097260
34	3	7	256250.376014123	5034319.347926460
44	3	8	259051.864391000	5033215.737003990
44	4	1	259051.864391000	5033215.737003990
20	4	2	255948.069021439	5034250.244479680
21	4	3	256040.288863683	5034326.884370310
22	4	4	256023.962045764	5034400.838618200
24	4	5	256065.239606521	5034480.234596910
33	4	6	256323.512120709	5034332.311882570
31	4	7	256409.805148712	5034429.046381020
44	4	8	259051.864391000	5033215.737003990
44	5	1	259051.864391000	5033215.737003990
32	5	2	256343.624034497	5034535.861678260
30	5	3	256310.541395512	5034569.743184420
23	5	4	256128.482257811	5034677.493612770
26	5	5	256061.557482728	5034727.598430270
25	5	6	255990.595664290	5034779.640914700
27	5	7	255864.117118510	5034882.906177090
44	5	8	259051.864391000	5033215.737003990

(a) Route 1.



(b) Route 2.



(c) Route 3.

(d) Route 4.

(e) Route 5.

Figure 4.7: Single routes from the shortest routes plan results of the Nearest Neighbor Algorithm simulation.

4.3.5 Three-phase Algorithm simulations and results

The simulations of the waste collection and transport based on the Genetic Algorithm, have been processed from the same input data of the NNA simulation, shown in Tab. 4.1 and from the previous hypothesis. The GA idea is based on the Theory of the Evolution. It means that only the strongest individual survives. Therefore after a lot of generations rest only the strongest population. The aim of this algorithm is the same. The population is composed of N strings (subjects of starting population), of M numbers representative of the rows (Tab. 4.1) of the input data. The population evolves in a second generation composed of N strings of M numbers. This generation came from a combination and mutation of the first, and is composed of the best subjects produced from the first. This process continues for X generations (iterations).

Characterization analysis

Optimization is a process that finds a best, or optimal, solution of a problem. An optimization problem is defined as: finding values of the variables that minimize or maximize the objective function while satisfying the constraints. The Optimization problems are centered on three factors: (1) an objective function which is to be minimized or maximized. (2) A set of unknowns or variables that affect the objective function. (3) A set of constraints that allow the unknowns to take on certain values but exclude others [35].

The parameters N and X should be optimize. This work, is based on real data that have been collected in the city of Padova, but they are not enough and so not representative of the waste collection in the city. Is not a complete case study. For this reason, the sensitivity analysis doesn't have been done, but it will be necessary if the aim will be find the best possible result.

Anyway, to simplify the discussion, as wrote in the paragraph 4.3.2, the Three-phase Algorithm can be divided into two levels. The first described by the Algorithm 4.3.6, and the second by the Algorithm 4.3.3. Both levels required different N and X parameters value. Therefore, a simple study have been done before the simulations to briefly choose N and X for the route optimization in each cluster (level 2). The analysis has been done in two times, at first the value of N has been fixed and the value of X changed, vice versa in the second time after the examination of the first results. At the same time in level 1 the parameter N has been fixed at 4 (must be divisible by 4) and X at 5. The results have been shown in the Fig. 4.8 and Fig. 4.9, and in Tab. 4.4. Each simulation has been done 3 times to reduce the aleatory of the results that are averaged between them.

The first analysis (Fig. 4.8) shows that the Delta value in a range of $X = 10 \div 100$, has a rapid jump of about 8%. The footprint is not diminished with significance (1.5%) if X value has been chosen equal to 50, in comparison with the time-dependent that is about halved.

3% is the biggest jump of Delta in the second analysis (Fig. 4.9). In comparison with the previous ones could be neglected and a value of $N = 4$ has been chosen, moreover the time-dependent is improved.

Simulations

The value of the input parameters and results of the TPA simulations are shown in the Tab. 4.5. Then comparisons between time of process and efficiency with $N = 4, 36, 100$ in level 1, are shown in the Fig. 4.10, Fig. 4.11 and Fig. 4.12. The aim is to improve

Table 4.4: Parameters of simulations during the characterization analysis. $N = numerosity of population$, $X = iterations$.

Level 1		Level 2		Total distance (Km)
N	X	N	X	
4	5	4	10	48.914
4	5	4	50	45.763
4	5	4	100	45.113
4	5	4	500	45.075
4	5	4	1000	44.891
4	5	4	50	45.763
4	5	36	50	45.092
4	5	100	50	44.659
4	5	400	50	44.157
4	5	1000	50	44.411

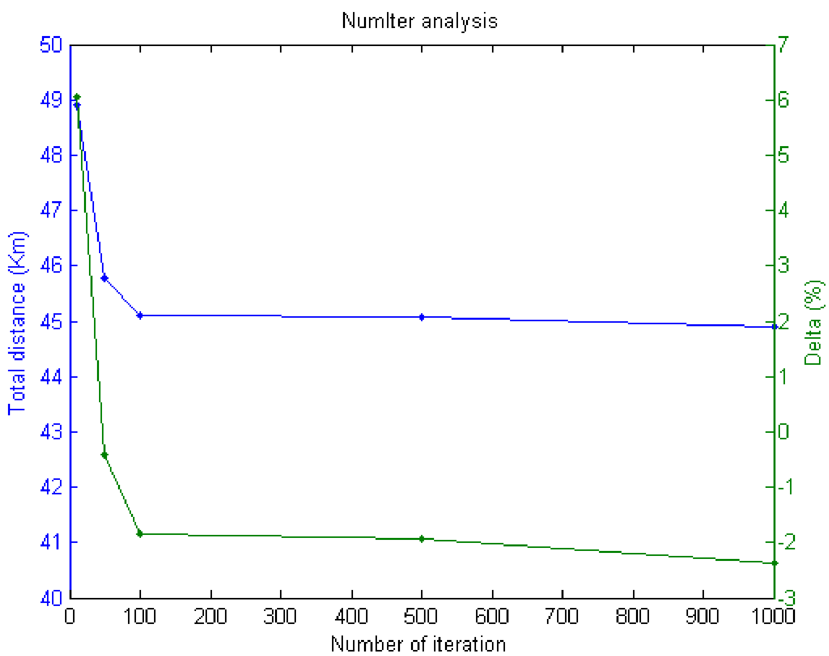


Figure 4.8: Level 2 parameters analysis. The value of $N = 4$ fixed and the value of X changed from 10 to 1000. Delta is equal to $[(Totaldistance)_{\mathrm{n}} - (\sum_n (Totaldistance)_{\mathrm{n}})/4] \cdot 100/(Totaldistance)_{\mathrm{n}}$.

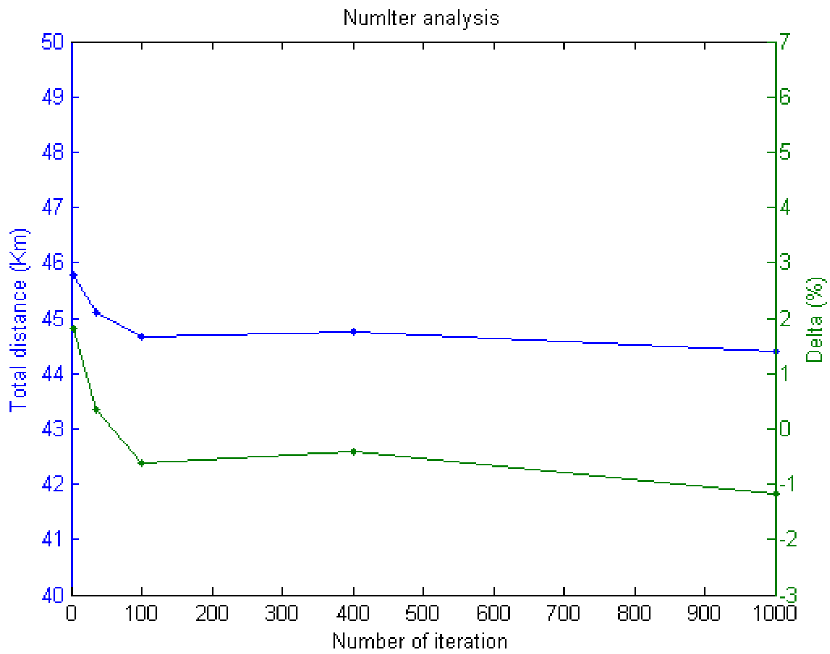


Figure 4.9: Level 2 parameters analysis. The value of $X = 50$ fixed and the value of N changed from 4 to 1000. Delta is equal to $[(Totaldistance)_{\mathrm{n}} - (\sum_{n}(Totaldistance)_{\mathrm{n}})/4] \cdot 100/(Totaldistance)_{\mathrm{n}}$.

the knowledge of the algorithm and is not to find the best result possible, it requires a complete optimization.

The results reported in Tab. 4.5, are also shown in Fig. 4.10, Fig. 4.11, Fig. 4.12 and their combination in Fig. 4.13.

Table 4.5: Simulations parameters. $N = numerosityofpopulation$, $X = iterations$.

Level 1 N	Level 1 X	Level 2 N	Level 2 X	**Total distance** (Km)	**Number of Routes**
4	10	4	50	44.542	5
4	50	4	50	44.876	5
4	100	4	50	42.058	5
4	200	4	50	43.799	5
4	500	4	50	41.568	5
36	10	4	50	42.855	5
36	50	4	50	41.900	5
36	100	4	50	39.355	5
36	200	4	50	39.436	5
36	500	4	50	39.137	5
100	10	4	50	41.208	5
100	50	4	50	40.725	5
100	100	4	50	39.366	5
100	200	4	50	37.715	5
100	500	4	50	37.525	5

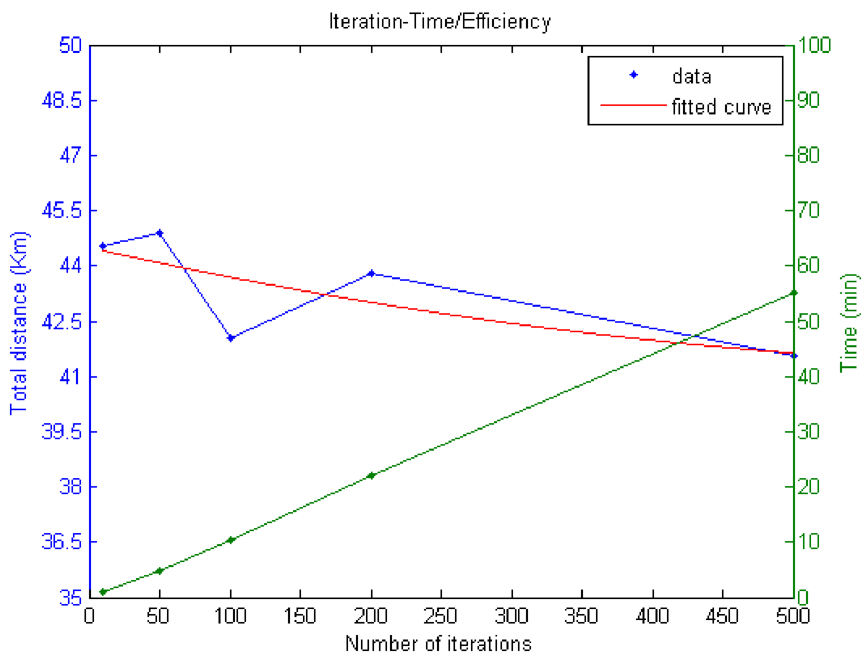


Figure 4.10: Comparison between time of process and efficiency with $N = 4$ at level 1. The blue curve, represents the results shown in Tab 4.5. The time required for each simulation, is represented by the green curve.

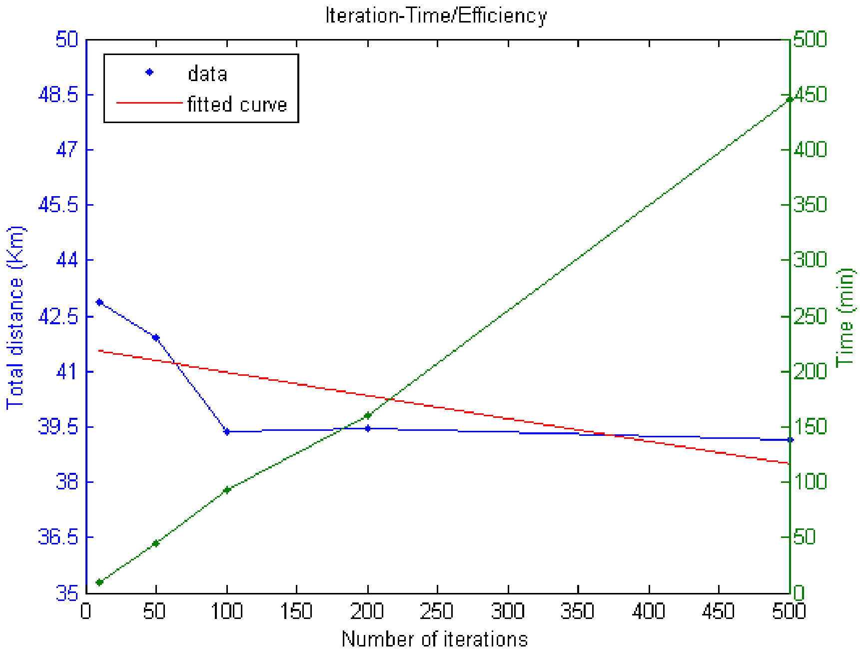


Figure 4.11: Comparison between time of process and efficiency with $N = 36$ at level 1. The blue curve, represents the results shown in Tab 4.5. The time required for each simulation, is represented by the green curve.

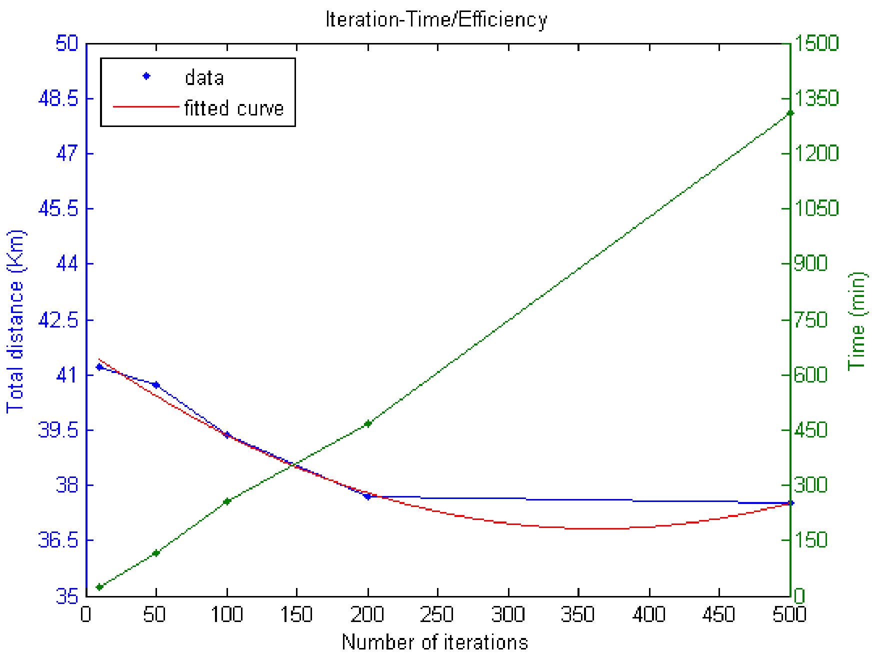


Figure 4.12: Comparison between time of process and efficiency with $N = 100$ at level 1. The blue curve, represents the results shown in Tab 4.5. The time required for each simulation, is represented by the green curve.

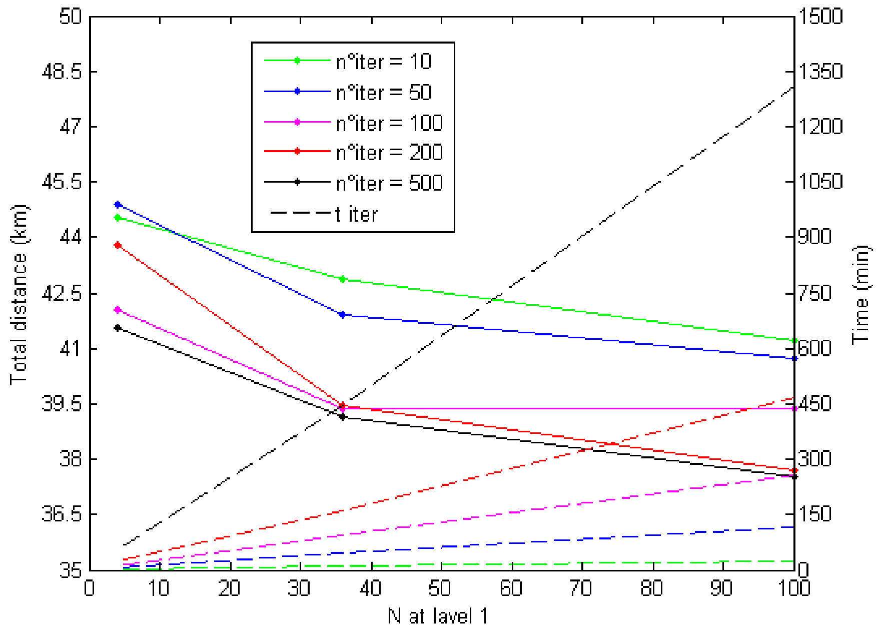


Figure 4.13: Comparison between time of processes and efficiency at different N and equal iterations. Each continuous curve represent the evolution at $N = 4, 36, 100$ at the same X. The dashed curves are their time-dependence.

Summarizing tables

The tables from Tab. I.5,to Tab. 4.10 show the numeric results of the simulations. In this paragraph only the tables of the last simulations ($X = 100$) have been reported. The others are in the Annex 1. The sum of the total volume of waste required by waste collection points in a single route, is plotted in Fig. 4.14. The scale of the axis z is increased to improve the visibility of the behavior.

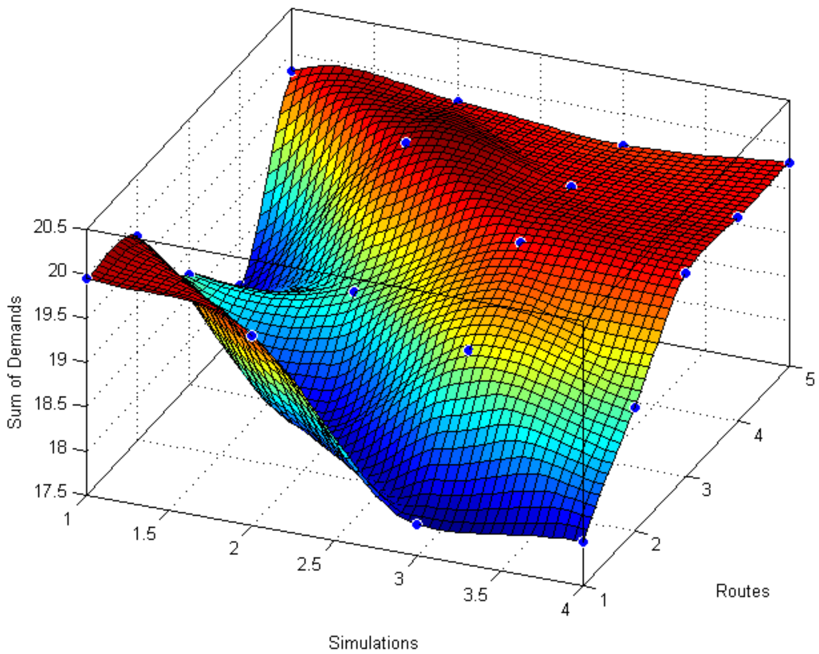


Figure 4.14: Surface produced by a cubic interpolation of the sum of the capacity of the collection truck required by the unsorted waste collection bins in each waste collection points. Therefore the total volume of waste collected by a truck in a single route. The range values is 17.85÷19.8

Table 4.6: Routes description from Three-phase Algorithm with 10 iterations ($N = 100$, level 1). The first column shows the id of the single waste collection point, the second column shows the route of membership, the third column gives the sequence of the waste collection points in each route, the fourth and fifth the planar coordinates (WGS 84/UTM zone 33N) of the collection points, and the last column shows the capacity of the collection truck required by the unsorted waste collection bins in each waste collection points and the total volume of waste collected by a truck in a single route.

id	n^o of single route	Order in single route	Coordinate x	Coordinate y	Demand (m^3)
44	1	1	259051.864391000	5033215.737003990	0
4	1	2	255898.945508198	5032336.245989560	0.75
3	1	3	255558.702458421	5031848.005041680	0.75
5	1	4	255673.673792252	5032455.027433400	0.75
1	1	5	255628.583800869	5032726.808474400	0.75
10	1	6	255881.750174129	5033240.180559440	0.75
7	1	7	256104.299103770	5033289.230643370	3.6
17	1	8	256244.288081726	5033178.380723640	5.4
9	1	9	256366.352106576	5033148.625915130	3.6
8	1	10	256409.377579713	5033131.223553460	3.6
44	1	11	259051.864391000	5033215.737003990	**19.95**
21	2	1	256040.288863683	5034326.884370310	3.6
22	2	2	256023.962045764	5034400.838618200	3.6
24	2	3	256065.239606521	5034480.234596910	3.6
26	2	4	256061.557482728	5034727.598430270	1.8
23	2	5	256128.482257811	5034677.493612770	3.6
44	2	6	259051.864391000	5033215.737003990	0
20	2	7	255948.069021439	5034250.244479680	3.6
21	2	8	256040.288863683	5034326.884370310	**19.8**
16	3	1	255675.372616814	5031927.586545350	0.75
15	3	2	255706.712233325	5032155.343681460	0.75
14	3	3	255630.854341550	5032038.684513050	0.75
44	3	4	259051.864391000	5033215.737003990	0
25	3	5	255990.595664290	5034779.640914700	3.6
27	3	6	255864.117118510	5034882.906177090	1.8
13	3	7	255873.196458077	5033619.410464310	3.6
11	3	8	255785.826530947	5033269.238715940	2.25
12	3	9	255749.912334742	5033357.616949720	0.75
6	3	10	255533.522407401	5032538.410363290	1.5
29	3	11	255478.382061578	5032155.201535760	0.75
28	3	12	255462.669529721	5032119.986247760	0.75
18	3	13	255446.949282576	5032080.543377010	0.75
19	3	14	255400.936652359	5031908.097526770	0.75
16	3	15	255675.372616814	5031927.586545350	**18.75**
30	4	1	256310.541395512	5034569.743184420	1.8
33	4	2	256323.512120709	5034332.311882570	3.6
34	4	3	256250.376014123	5034319.347926460	1.8
36	4	4	256090.544432397	5034200.434097260	3.6
44	4	5	259051.864391000	5033215.737003990	0
31	4	6	256409.805148712	5034429.046381020	1.8
32	4	7	256343.624034497	5034535.861678260	5.4
30	4	8	256310.541395512	5034569.743184420	**18**
39	5	1	256138.197891210	5033114.632764460	0.75
35	5	2	256069.804336837	5033267.764357700	0.75
42	5	3	255720.399256876	5032783.322989710	1.5
41	5	4	255560.651586477	5032609.286239500	2.25
43	5	5	255759.693089803	5032482.283847830	1.5
2	5	6	255935.337870106	5032357.086090610	1.5
44	5	7	259051.864391000	5033215.737003990	0
37	5	8	256319.746563686	5032990.427849010	7.2
40	5	9	256244.534849801	5032959.697884480	0.75
38	5	10	256173.487694673	5033105.142617960	3.6
39	5	11	256138.197891210	5033114.632764460	**19.8**

Table 4.7: Routes description from Three-phase Algorithm with 50 iterations ($N = 100$, level 1). The first column shows the id of the single waste collection point, the second column shows the route of membership, the third column gives the sequence of the waste collection points in each route, the fourth and fifth the planar coordinates (WGS 84/UTM zone 33N) of the collection points, and the last column shows the capacity of the collection truck required by the unsorted waste collection bins in each waste collection points and the total volume of waste collected by a truck in a single route.

id	n^o of single route	Order in single route	Coordinate x	Coordinate y	Demand (m^3)
29	1	1	255478.382061578	5032155.201535760	0.75
28	1	2	255462.669529721	5032119.986247760	0.75
18	1	3	255446.949282576	5032080.543377010	0.75
19	1	4	255400.936652359	5031908.097526770	0.75
3	1	5	255558.702458421	5031848.005041680	0.75
2	1	6	255935.337870106	5032357.086090610	1.5
44	1	7	259051.864391000	5033215.737003990	0
30	1	8	256310.541395512	5034569.743184420	1.8
32	1	9	256343.624034497	5034535.861678260	5.4
31	1	10	256409.805148712	5034429.046381020	1.8
33	1	11	256323.512120709	5034332.311882570	3.6
34	1	12	256250.376014123	5034319.347926460	1.8
29	1	13	255478.382061578	5032155.201535760	**19.65**
23	2	1	256128.482257811	5034677.493612770	3.6
44	2	2	259051.864391000	5033215.737003990	0
22	2	3	256023.962045764	5034400.838618200	3.6
24	2	4	256065.239606521	5034480.234596910	3.6
27	2	5	255864.117118510	5034882.906177090	1.8
25	2	6	255990.595664290	5034779.640914700	3.6
26	2	7	256061.557482728	5034727.598430270	1.8
23	2	8	256128.482257811	5034677.493612770	**18**
10	3	1	255881.750174129	5033240.180559440	0.75
11	3	2	255785.826530947	5033269.238715940	2.25
12	3	3	255749.912334742	5033357.616949720	0.75
13	3	4	255873.196458077	5033619.410464310	3.6
20	3	5	255948.069021439	5034250.244479680	3.6
21	3	6	256040.288863683	5034326.884370310	3.6
36	3	7	256090.544432397	5034200.434097260	3.6
44	3	8	259051.864391000	5033215.737003990	0
35	3	9	256069.804336837	5033267.764357700	0.75
10	3	10	255881.750174129	5033240.180559440	**18.9**
16	4	1	255675.372616814	5031927.586545350	0.75
14	4	2	255630.854341550	5032038.684513050	0.75
15	4	3	255706.712233325	5032155.343681460	0.75
4	4	4	255898.945508198	5032336.245989560	0.75
39	4	5	256138.197891210	5033114.632764460	0.75
7	4	6	256104.299103770	5033289.230643370	3.6
17	4	7	256244.288081726	5033178.380723640	5.4
9	4	8	256366.352106576	5033148.625915130	3.6
8	4	9	256409.377579713	5033131.223553460	3.6
44	4	10	259051.864391000	5033215.737003990	0
16	4	11	255675.372616814	5031927.586545350	**19.95**
40	5	1	256244.534849801	5032959.697884480	0.75
37	5	2	256319.746563686	5032990.427849010	7.2
44	5	3	259051.864391000	5033215.737003990	0
38	5	4	256173.487694673	5033105.142617960	3.6
42	5	5	255720.399256876	5032783.322989710	1.5
1	5	6	255628.583800869	5032726.808474400	0.75
41	5	7	255560.651586477	5032609.286239500	2.25
6	5	8	255533.522407401	5032538.410363290	1.5
5	5	9	255673.673792252	5032455.027433400	0.75
43	5	10	255759.693089803	5032482.283847830	1.5
40	5	11	256244.534849801	5032959.697884480	**19.8**

Table 4.8: Routes description from Three-phase Algorithm with 100 iterations ($N = 100$, level 1). The first column shows the id of the single waste collection point, the second column shows the route of membership, the third column gives the sequence of the waste collection points in each route, the fourth and fifth the planar coordinates (WGS 84/UTM zone 33N) of the collection points, and the last column shows the capacity of the collection truck required by the unsorted waste collection bins in each waste collection points and the total volume of waste collected by a truck in a single route.

id	n^o of single route	Order in single route	Coordinate x	Coordinate y	Demand (m^3)
6	1	1	255533.522407401	5032538.410363290	1.5
1	1	2	255628.583800869	5032726.808474400	0.75
35	1	3	256069.804336837	5033267.764357700	0.75
7	1	4	256104.299103770	5033289.230643370	3.6
17	1	5	256244.288081726	5033178.380723640	5.4
8	1	6	256409.377579713	5033131.223553460	3.6
44	1	7	259051.864391000	5033215.737003990	0
2	1	8	255935.337870106	5032357.086090610	1.5
5	1	9	255673.673792252	5032455.027433400	0.75
6	1	10	255533.522407401	5032538.410363290	**17.85**
3	2	1	255558.702458421	5031848.005041680	0.75
19	2	2	255400.936652359	5031908.097526770	0.75
18	2	3	255446.949282576	5032080.543377010	0.75
28	2	4	255462.669529721	5032119.986247760	0.75
15	2	5	255706.712233325	5032155.343681460	0.75
43	2	6	255759.693089803	5032482.283847830	1.5
10	2	7	255881.750174129	5033240.180559440	0.75
11	2	8	255785.826530947	5033269.238715940	2.25
12	2	9	255749.912334742	5033357.616949720	0.75
13	2	10	255873.196458077	5033619.410464310	3.6
20	2	11	255948.069021439	5034250.244479680	3.6
44	2	12	259051.864391000	5033215.737003990	0
4	2	13	255898.945508198	5032336.245989560	0.75
29	2	14	255478.382061578	5032155.201535760	0.75
14	2	15	255630.854341550	5032038.684513050	0.75
16	2	16	255675.372616814	5031927.586545350	0.75
3	2	17	255558.702458421	5031848.005041680	**19.2**
44	3	1	259051.864391000	5033215.737003990	0
34	3	2	256250.376014123	5034319.347926460	1.8
27	3	3	255864.117118510	5034882.906177090	1.8
25	3	4	255990.595664290	5034779.640914700	3.6
26	3	5	256061.557482728	5034727.598430270	1.8
23	3	6	256128.482257811	5034677.493612770	3.6
32	3	7	256343.624034497	5034535.861678260	5.4
31	3	8	256409.805148712	5034429.046381020	1.8
44	3	9	259051.864391000	5033215.737003990	**19.8**
30	4	1	256310.541395512	5034569.743184420	1.8
24	4	2	256065.239606521	5034480.234596910	3.6
22	4	3	256023.962045764	5034400.838618200	3.6
21	4	4	256040.288863683	5034326.884370310	3.6
36	4	5	256090.544432397	5034200.434097260	3.6
33	4	6	256323.512120709	5034332.311882570	3.6
44	4	7	259051.864391000	5033215.737003990	0
30	4	8	256310.541395512	5034569.743184420	**19.8**
37	5	1	256319.746563686	5032990.427849010	7.2
44	5	2	259051.864391000	5033215.737003990	0
9	5	3	256366.352106576	5033148.625915130	3.6
38	5	4	256173.487694673	5033105.142617960	3.6
39	5	5	256138.197891210	5033114.632764460	0.75
42	5	6	255720.399256876	5032783.322989710	1.5
41	5	7	255560.651586477	5032609.286239500	2.25
40	5	8	256244.534849801	5032959.697884480	0.75
37	5	9	256319.746563686	5032990.427849010	**19.65**

Table 4.9: Routes description from Three-phase Algorithm with 200 iterations (N = 100, level 1). The first column shows the id of the single waste collection point, the second column shows the route of membership, the third column gives the sequence of the waste collection points in each route, the fourth and fifth the planar coordinates (WGS 84/UTM zone 33N) of the collection points, and the last column shows the capacity of the collection truck required by the unsorted waste collection bins in each waste collection points and the total volume of waste collected by a truck in a single route.

id	n^o of single route	Order in single route	Coordinate x	Coordinate y	Demand (m^3)
16	1	1	255675.372616814	5031927.586545350	0.75
14	1	2	255630.854341550	5032038.684513050	0.75
15	1	3	255706.712233325	5032155.343681460	0.75
2	1	4	255935.337870106	5032357.086090610	1.5
44	1	5	259051.864391000	5033215.737003990	0
8	1	6	256409.377579713	5033131.223553460	3.6
17	1	7	256244.288081726	5033178.380723640	5.4
43	1	8	255759.693089803	5032482.283847830	1.5
29	1	9	255478.382061578	5032155.201535760	0.75
28	1	10	255462.669529721	5032119.986247760	0.75
18	1	11	255446.949282576	5032080.543377010	0.75
19	1	12	255400.936652359	5031908.097526770	0.75
3	1	13	255558.702458421	5031848.005041680	0.75
16	1	14	255675.372616814	5031927.586545350	**18**
36	2	1	256090.544432397	5034200.434097260	3.6
13	2	2	255873.196458077	5033619.410464310	3.6
12	2	3	255749.912334742	5033357.616949720	0.75
11	2	4	255785.826530947	5033269.238715940	2.25
10	2	5	255881.750174129	5033240.180559440	0.75
35	2	6	256069.804336837	5033267.764357700	0.75
7	2	7	256104.299103770	5033289.230643370	3.6
9	2	8	256366.352106576	5033148.625915130	3.6
44	2	9	259051.864391000	5033215.737003990	0
36	2	10	256090.544432397	5034200.434097260	**18.9**
44	3	1	259051.864391000	5033215.737003990	0
22	3	2	256023.962045764	5034400.838618200	3.6
24	3	3	256065.239606521	5034480.234596910	3.6
27	3	4	255864.117118510	5034882.906177090	1.8
25	3	5	255990.595664290	5034779.640914700	3.6
26	3	6	256061.557482728	5034727.598430270	1.8
23	3	7	256128.482257811	5034677.493612770	3.6
30	3	8	256310.541395512	5034569.743184420	1.8
44	3	9	259051.864391000	5033215.737003990	**19.8**
32	4	1	256343.624034497	5034535.861678260	5.4
21	4	2	256040.288863683	5034326.884370310	3.6
20	4	3	255948.069021439	5034250.244479680	3.6
34	4	4	256250.376014123	5034319.347926460	1.8
33	4	5	256323.512120709	5034332.311882570	3.6
44	4	6	259051.864391000	5033215.737003990	0
31	4	7	256409.805148712	5034429.046381020	1.8
32	4	8	256343.624034497	5034535.861678260	**19.8**
41	5	1	255560.651586477	5032609.286239500	2.25
1	5	2	255628.583800869	5032726.808474400	0.75
42	5	3	255720.399256876	5032783.322989710	1.5
39	5	4	256138.197891210	5033114.632764460	0.75
38	5	5	256173.487694673	5033105.142617960	3.6
40	5	6	256244.534849801	5032959.697884480	0.75
37	5	7	256319.746563686	5032990.427849010	7.2
44	5	8	259051.864391000	5033215.737003990	0
4	5	9	255898.945508198	5032336.245989560	0.75
5	5	10	255673.673792252	5032455.027433400	0.75
6	5	11	255533.522407401	5032538.410363290	1.5
41	5	12	255560.651586477	5032609.286239500	**19.8**

Table 4.10: Routes description from Three-phase Algorithm with 500 iterations ($N = 100$, level 1). The first column shows the id of the single waste collection point, the second column shows the route of membership, the third column gives the sequence of the waste collection points in each route, the fourth and fifth the planar coordinates (WGS 84/UTM zone 33N) of the collection points, and the last column shows the capacity of the collection truck required by the unsorted waste collection bins in each waste collection points and the total volume of waste collected by a truck in a single route.

id	n^o of single route	Order in single route	Coordinate x	Coordinate y	Demand (m^3)
44	1	1	259051.864391000	5033215.737003990	0
41	1	2	255560.651586477	5032609.286239500	2.25
1	1	3	255628.583800869	5032726.808474400	0.75
42	1	4	255720.399256876	5032783.322989710	1.5
11	1	5	255785.826530947	5033269.238715940	2.25
12	1	6	255749.912334742	5033357.616949720	0.75
10	1	7	255881.750174129	5033240.180559440	0.75
35	1	8	256069.804336837	5033267.764357700	0.75
39	1	9	256138.197891210	5033114.632764460	0.75
38	1	10	256173.487694673	5033105.142617960	3.6
9	1	11	256366.352106576	5033148.625915130	3.6
44	1	12	259051.864391000	5033215.737003990	**16.95**
44	2	1	259051.864391000	5033215.737003990	0
8	2	2	256409.377579713	5033131.223553460	3.6
17	2	3	256244.288081726	5033178.380723640	5.4
7	2	4	256104.299103770	5033289.230643370	3.6
13	2	5	255873.196458077	5033619.410464310	3.6
36	2	6	256090.544432397	5034200.434097260	3.6
44	2	7	259051.864391000	5033215.737003990	**19.8**
44	3	1	259051.864391000	5033215.737003990	0
32	3	2	256343.624034497	5034535.861678260	5.4
30	3	3	256310.541395512	5034569.743184420	1.8
23	3	4	256128.482257811	5034677.493612770	3.6
26	3	5	256061.557482728	5034727.598430270	1.8
25	3	6	255990.595664290	5034779.640914700	3.6
27	3	7	255864.117118510	5034882.906177090	1.8
31	3	8	256409.805148712	5034429.046381020	1.8
44	3	9	259051.864391000	5033215.737003990	**19.8**
44	4	1	259051.864391000	5033215.737003990	0
20	4	2	255948.069021439	5034250.244479680	3.6
21	4	3	256040.288863683	5034326.884370310	3.6
22	4	4	256023.962045764	5034400.838618200	3.6
24	4	5	256065.239606521	5034480.234596910	3.6
34	4	6	256250.376014123	5034319.347926460	1.8
33	4	7	256323.512120709	5034332.311882570	3.6
44	4	8	259051.864391000	5033215.737003990	**19.8**
5	5	1	255673.673792252	5032455.027433400	0.75
6	5	2	255533.522407401	5032538.410363290	1.5
29	5	3	255478.382061578	5032155.201535760	0.75
28	5	4	255462.669529721	5032119.986247760	0.75
18	5	5	255446.949282576	5032080.543377010	0.75
19	5	6	255400.936652359	5031908.097526770	0.75
3	5	7	255558.702458421	5031848.005041680	0.75
16	5	8	255675.372616814	5031927.586545350	0.75
14	5	9	255630.854341550	5032038.684513050	0.75
15	5	10	255706.712233325	5032155.343681460	0.75
4	5	11	255898.945508198	5032336.245989560	0.75
2	5	12	255935.337870106	5032357.086090610	1.5
44	5	13	259051.864391000	5033215.737003990	0
37	5	14	256319.746563686	5032990.427849010	7.2
40	5	15	256244.534849801	5032959.697884480	0.75
43	5	16	255759.693089803	5032482.283847830	1.5
5	5	17	255673.673792252	5032455.027433400	**19.95**

Evolution of iterations

The evolution during simulations have been presented in this paragraph from Fig. 4.15 to Fig. 4.44.

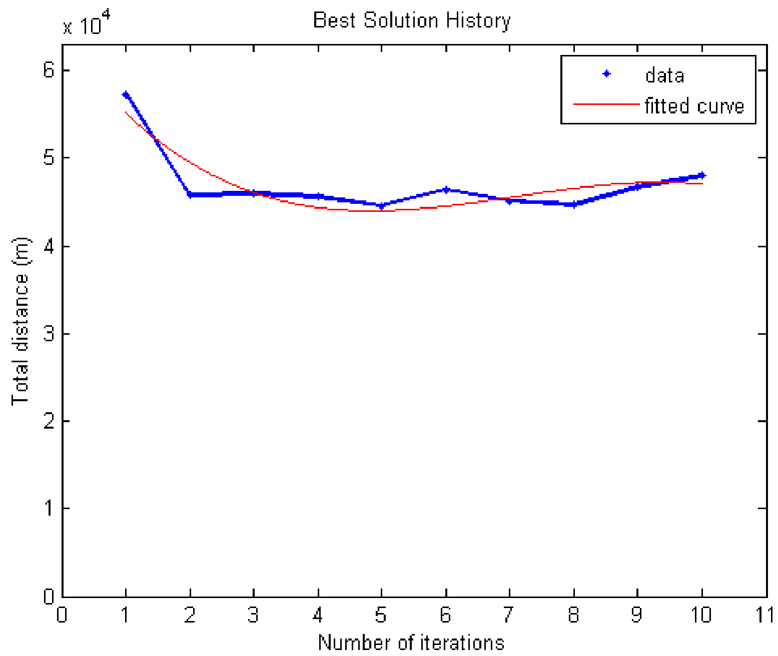


Figure 4.15: Evolution of the 10 iterations simulation ($N = 4$ at level 1). The best of the total distances of each iteration has been plotted. The aleatory based GA produces the oscillating trend.

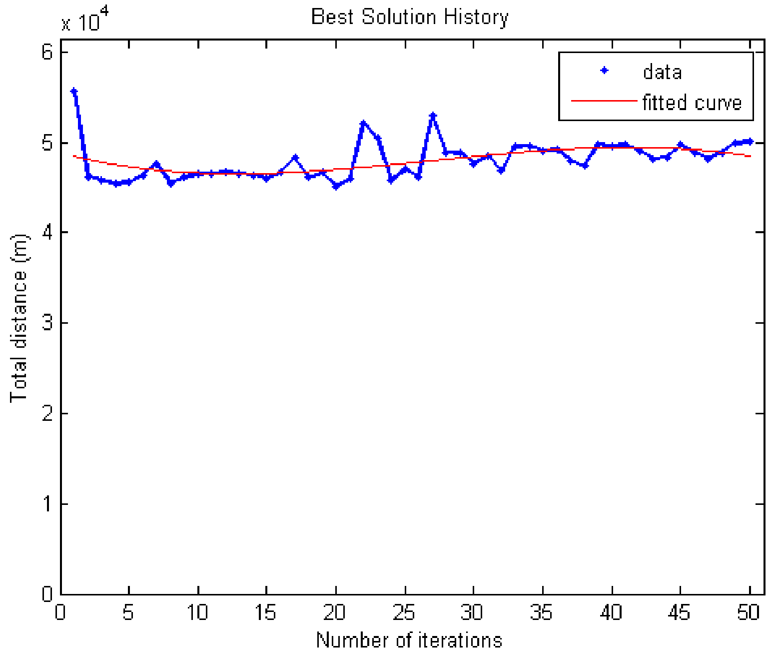


Figure 4.16: Evolution of the 50 iterations simulation ($N = 4$ at level 1). The best of the total distances of each iteration has been plotted. The aleatory based GA produces the oscillating trend.

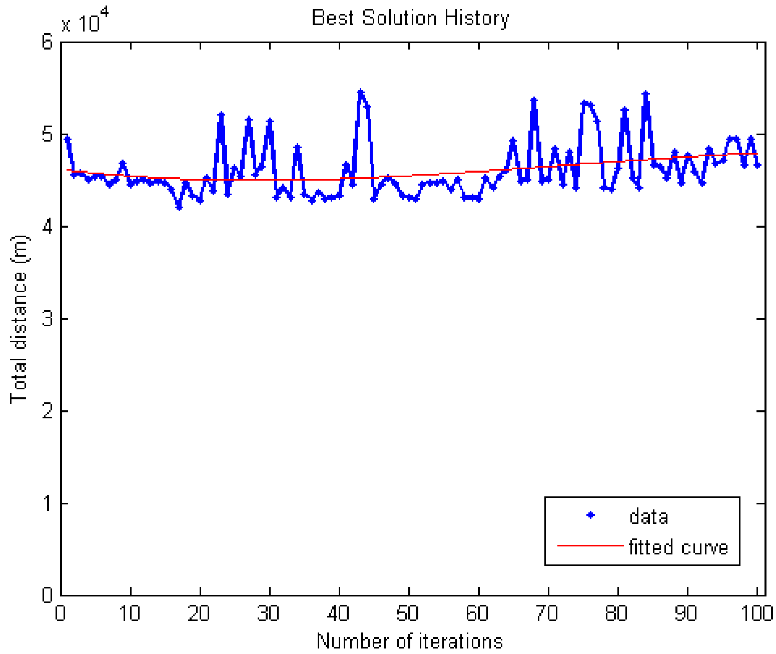


Figure 4.17: Evolution of the 100 iterations simulation ($N = 4$ at level 1). The best of the total distances of each iteration has been plotted. The aleatory based GA produces the oscillating trend.

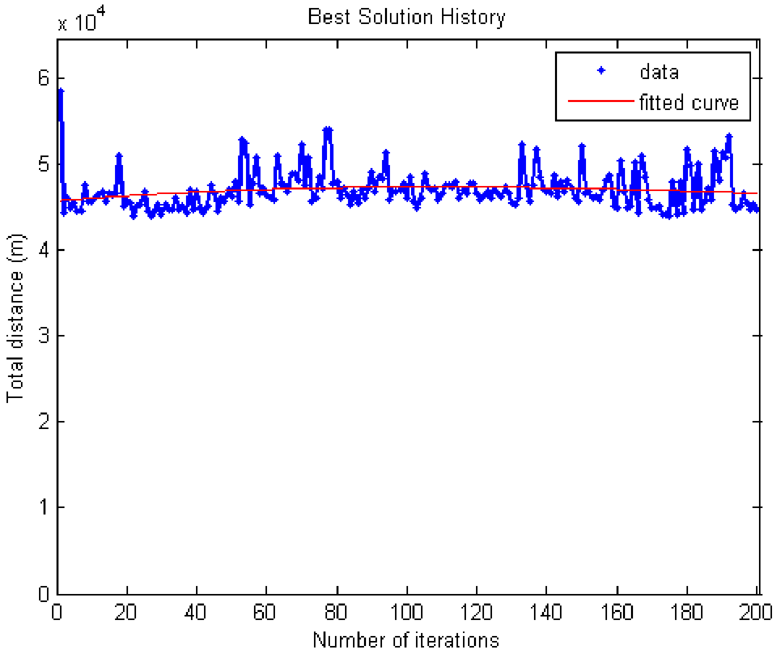


Figure 4.18: Evolution of the 200 iterations simulation ($N = 4$ at level 1). The best of the total distances of each iteration has been plotted. The aleatory based GA produces the oscillating trend.

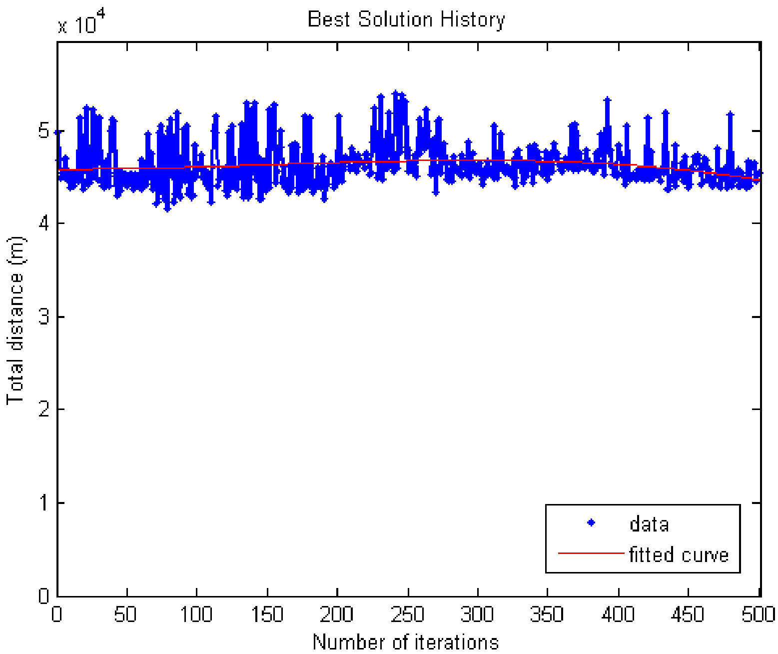


Figure 4.19: Evolution of the 500 iterations simulation ($N = 4$ at level 1). The best of the total distances of each iteration has been plotted. The aleatory based GA produces the oscillating trend.

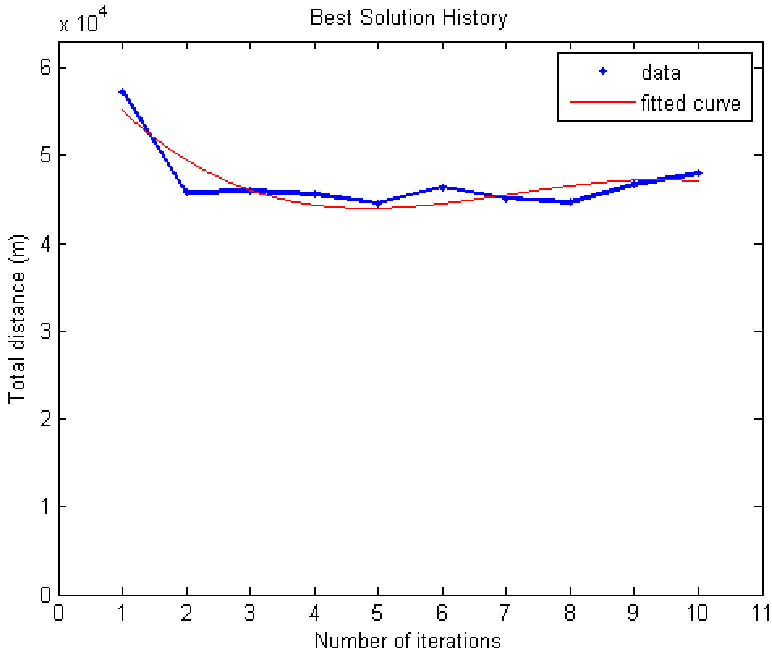


Figure 4.20: Evolution of the 10 iterations simulation ($N = 36$ at level 1). The best of the total distances of each iteration has been plotted. The aleatory based GA produces the oscillating trend.

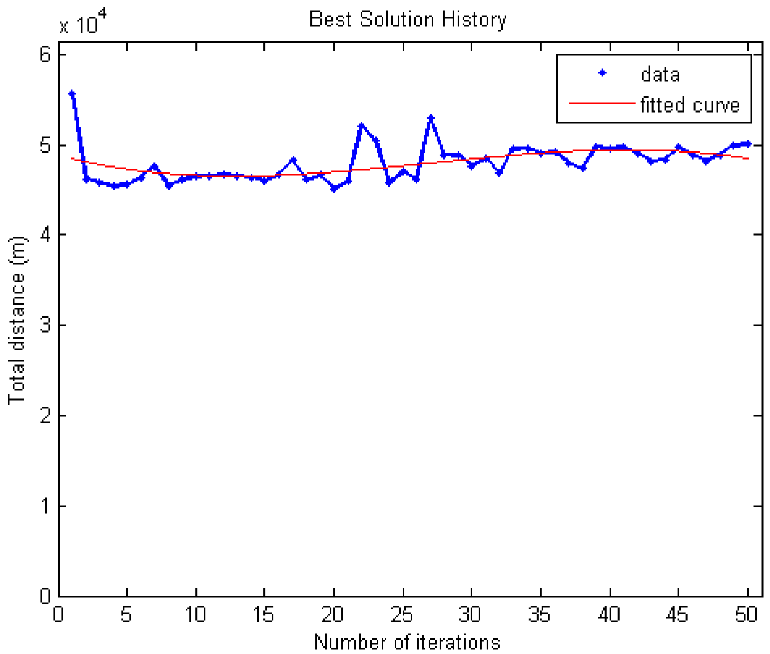


Figure 4.21: Evolution of the 50 iterations simulation ($N = 36$ at level 1). The best of the total distances of each iteration has been plotted. The aleatory based GA produces the oscillating trend.

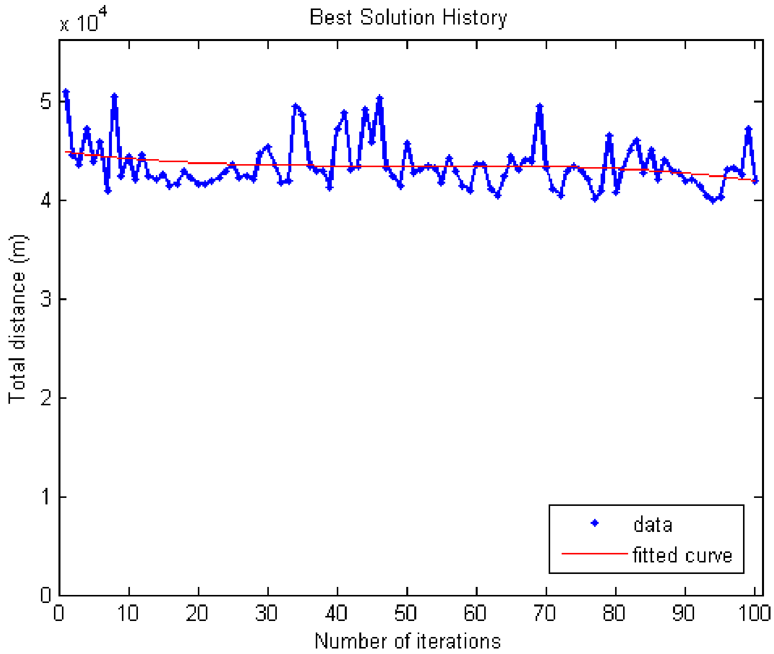


Figure 4.22: Evolution of the 100 iterations simulation ($N = 36$ at level 1). The best of the total distances of each iteration has been plotted. The aleatory based GA produces the oscillating trend.

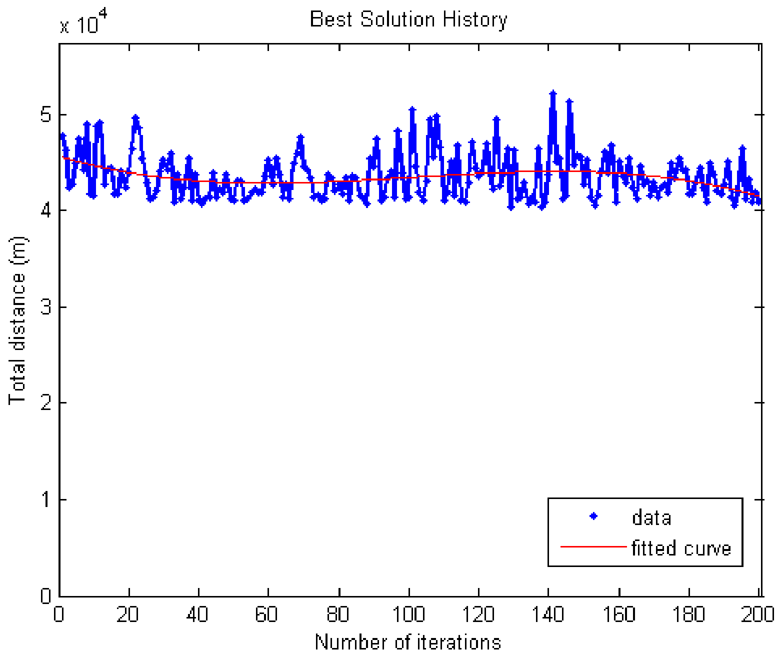


Figure 4.23: Evolution of the 200 iterations simulation ($N = 36$ at level 1). The best of the total distances of each iteration has been plotted. The aleatory based GA produces the oscillating trend.

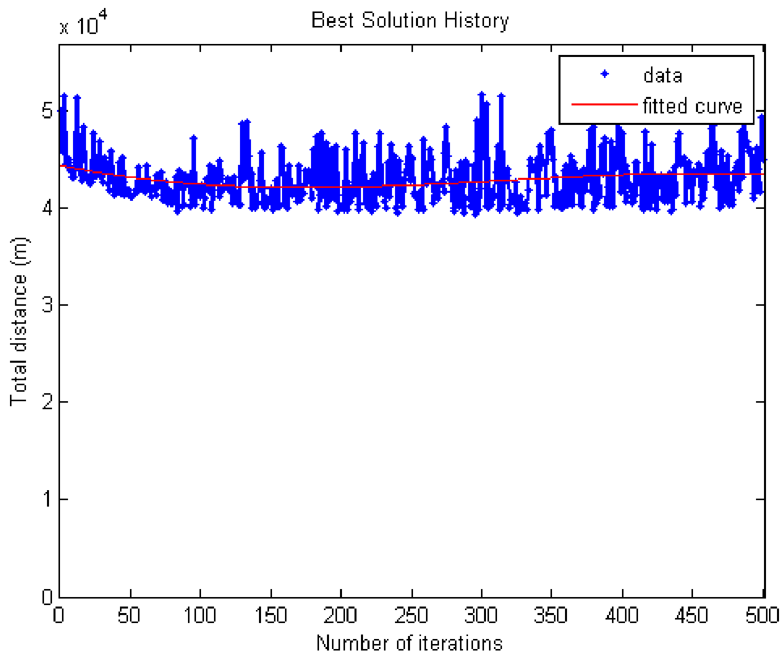


Figure 4.24: Evolution of the 500 iterations simulation ($N = 36$ at level 1). The best of the total distances of each iteration has been plotted. The aleatory based GA produces the oscillating trend.

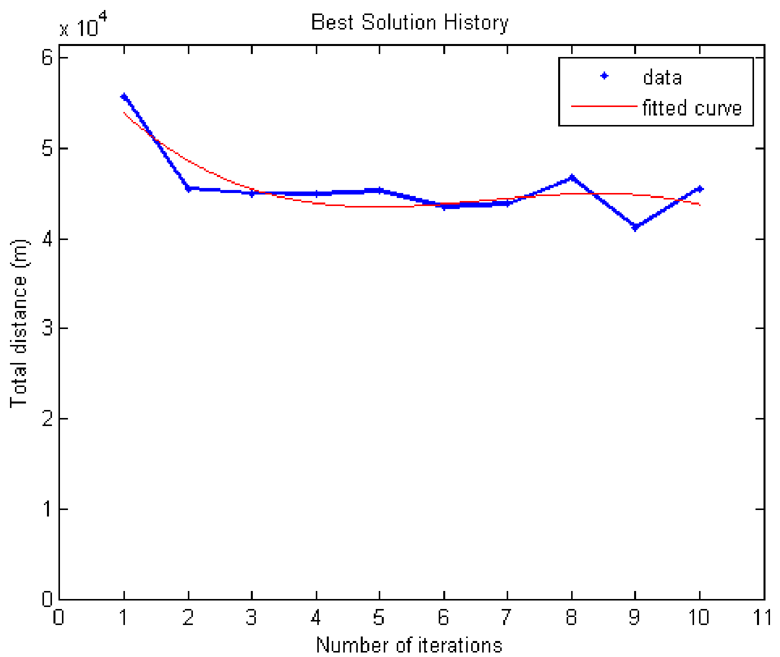


Figure 4.25: Evolution of the 10 iterations simulation ($N = 100$ at level 1). The best of the total distances of each iteration has been plotted. The aleatory based GA produces the oscillating trend.

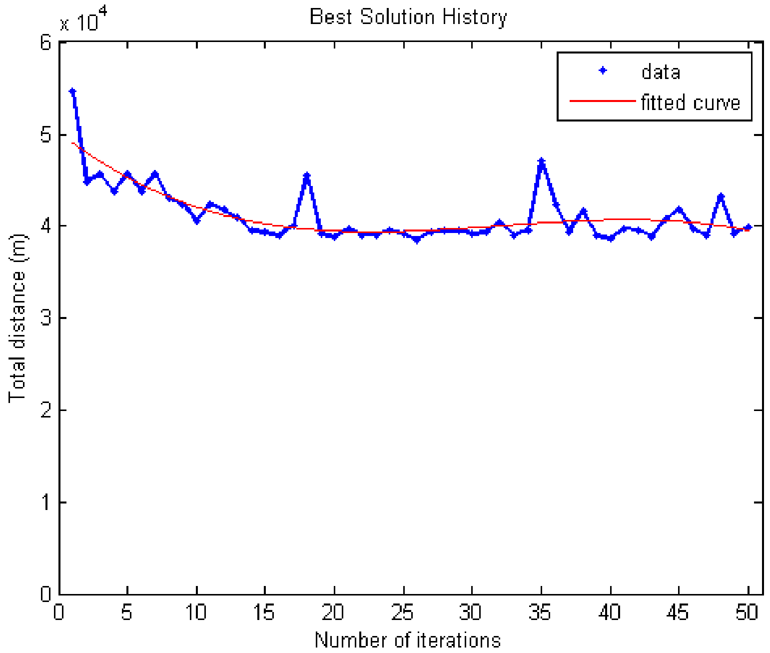


Figure 4.26: Evolution of the 50 iterations simulation ($N = 100$ at level 1). The best of the total distances of each iteration has been plotted. The aleatory based GA produces the oscillating trend.

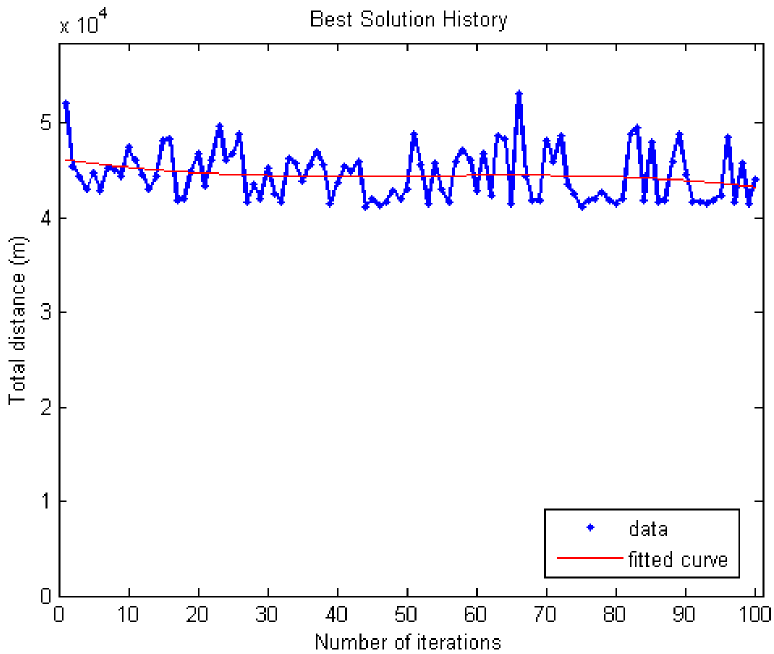


Figure 4.27: Evolution of the 100 iterations simulation ($N = 100$ at level 1). The best of the total distances of each iteration has been plotted. The aleatory based GA produces the oscillating trend.

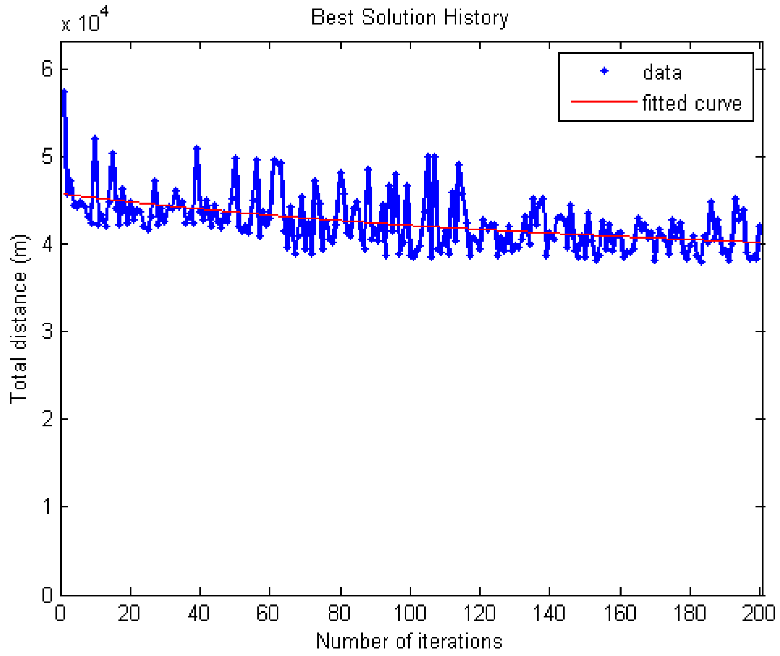


Figure 4.28: Evolution of the 200 iterations simulation ($N = 100$ at level 1). The best of the total distances of each iteration has been plotted. The aleatory based GA produces the oscillating trend.

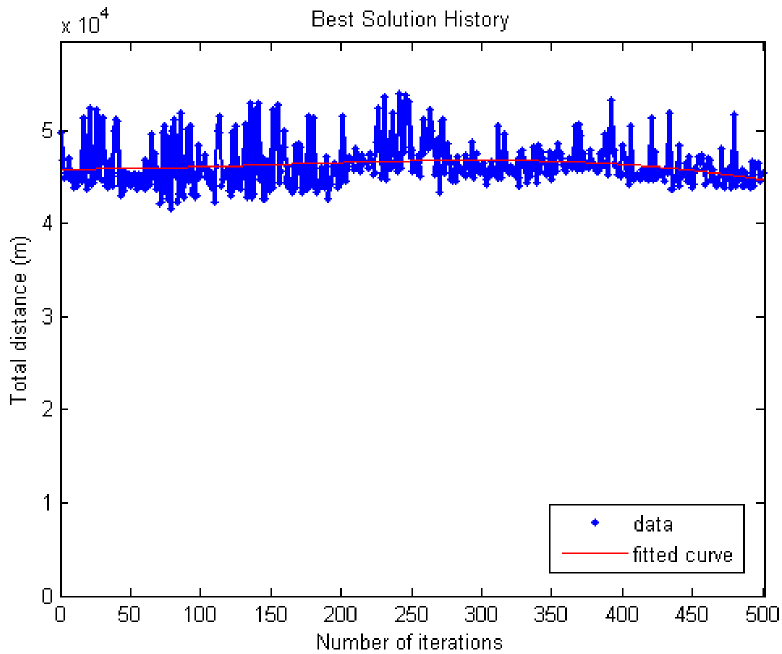


Figure 4.29: Evolution of the 500 iterations simulation ($N = 100$ at level 1). The best of the total distances of each iteration has been plotted. The aleatory based GA produces the oscillating trend.

Matlab routes planning

From Fig. 4.30 to Fig. 4.44 are the complete routes planning, from Matlab simulations. Then elaborated in QGIS (Paragraph 4.3.5).

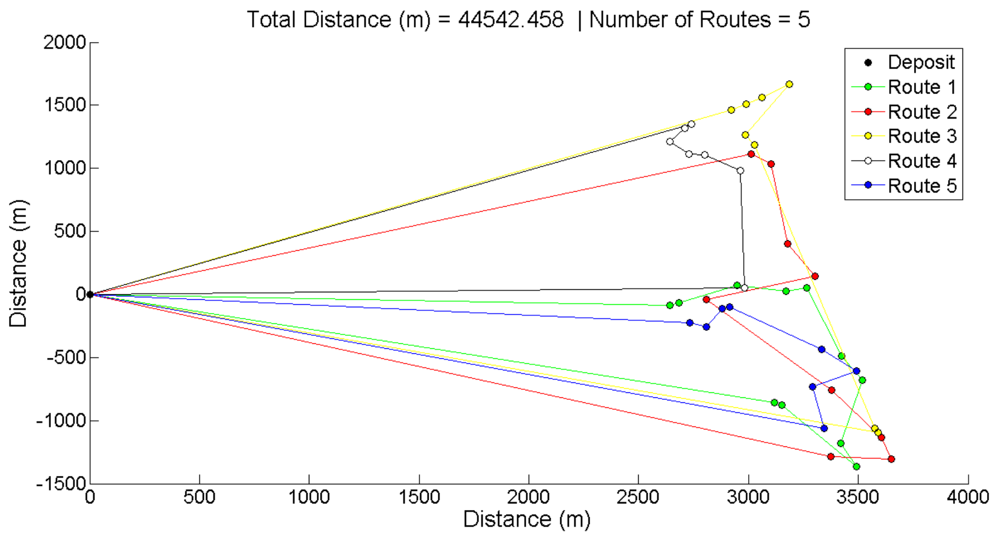


Figure 4.30: Routes results from Matlab simulation with 10 iterations ($N = 4$ at level 1). Total distance of the path is the sum of all the routes distances. The point in position (0,0) represents the deposit.

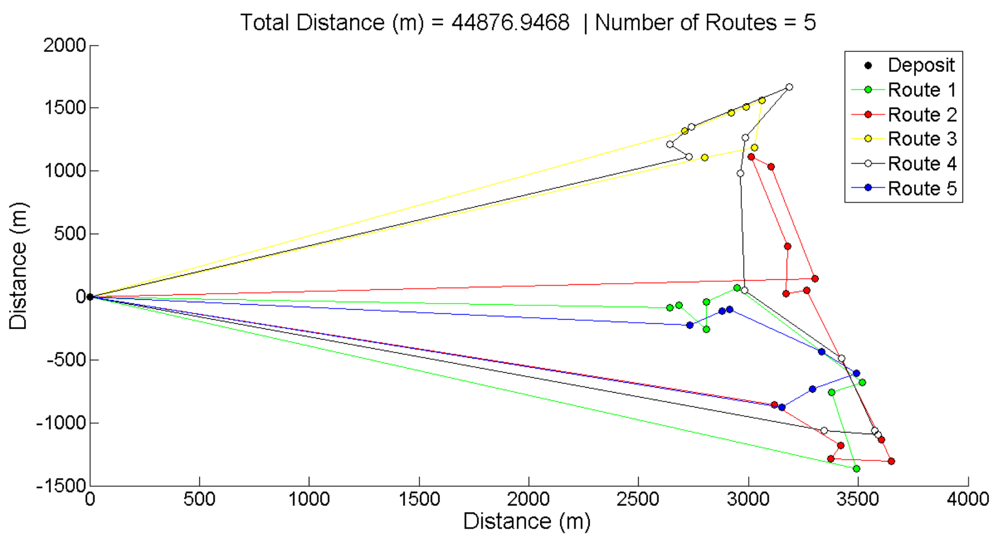


Figure 4.31: Routes results from Matlab simulation with 50 iterations ($N = 4$ at level 1). Total distance of the path is the sum of all the routes distances. The point in position (0,0) represents the deposit.

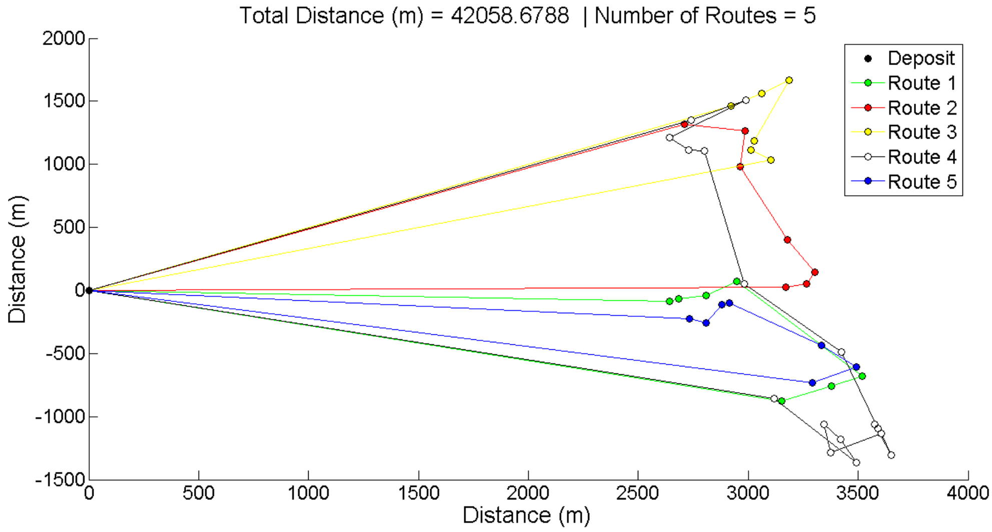


Figure 4.32: Routes results from Matlab simulation with 100 iterations ($N = 4$ at level 1). Total distance of the path is the sum of all the routes distances. The point in position (0,0) represents the deposit.

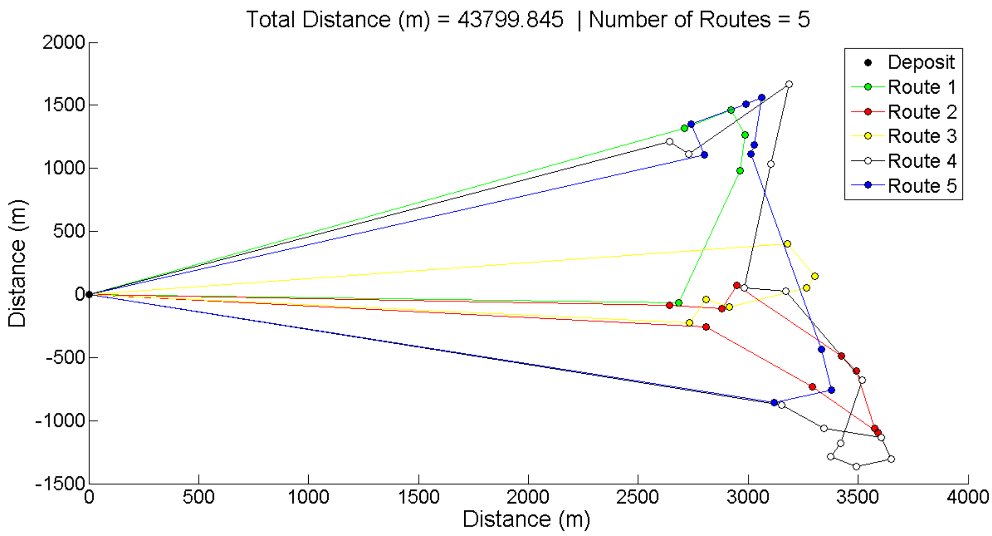


Figure 4.33: Routes results from Matlab simulation with 200 iterations ($N = 4$ at level 1). Total distance of the path is the sum of all the routes distances. The point in position (0,0) represents the deposit.

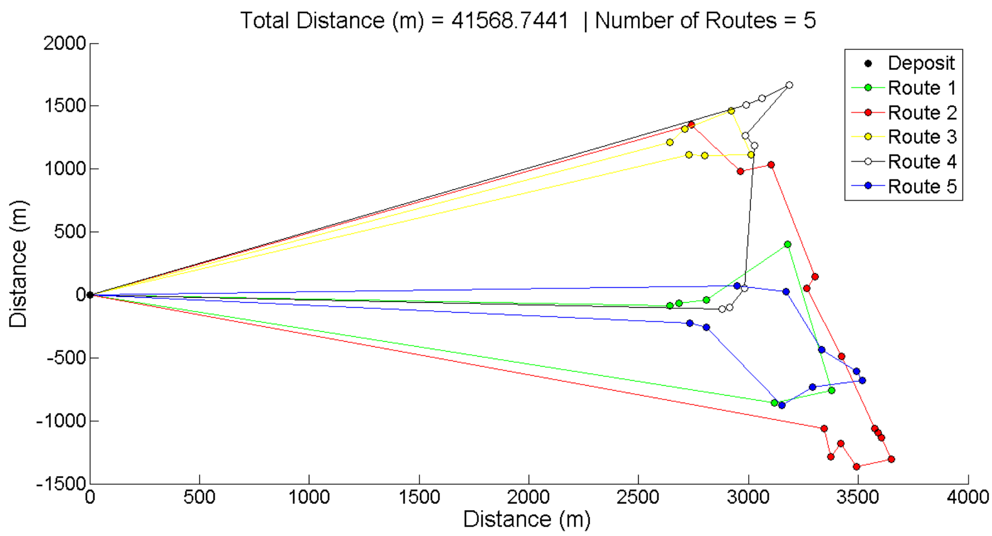


Figure 4.34: Routes results from Matlab simulation with 500 iterations ($N = 4$ at level 1). Total distance of the path is the sum of all the routes distances. The point in position (0,0) represents the deposit.

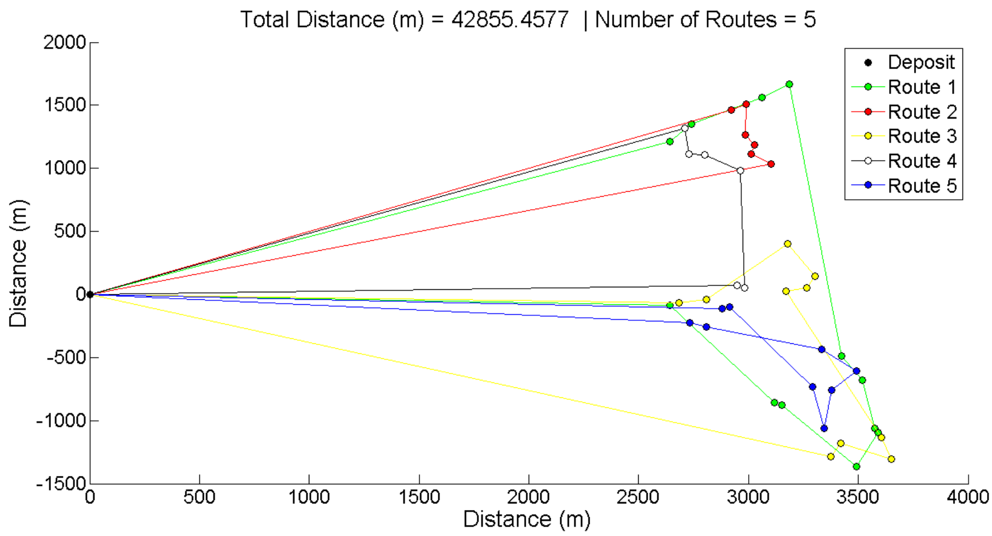


Figure 4.35: Routes results from Matlab simulation with 10 iterations ($N = 36$ at level 1). Total distance of the path is the sum of all the routes distances. The point in position (0,0) represents the deposit.

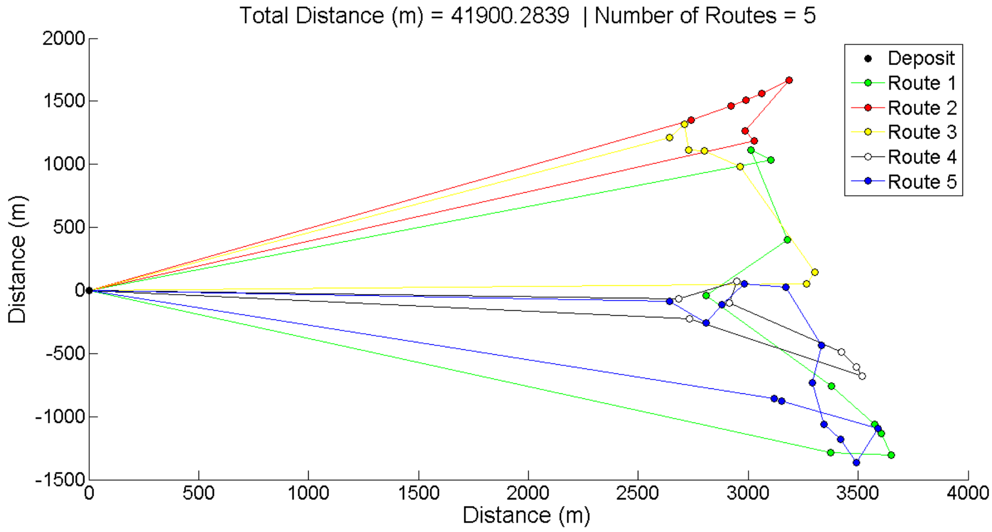


Figure 4.36: Routes results from Matlab simulation with 50 iterations ($N = 36$ at level 1). Total distance of the path is the sum of all the routes distances. The point in position (0,0) represents the deposit.

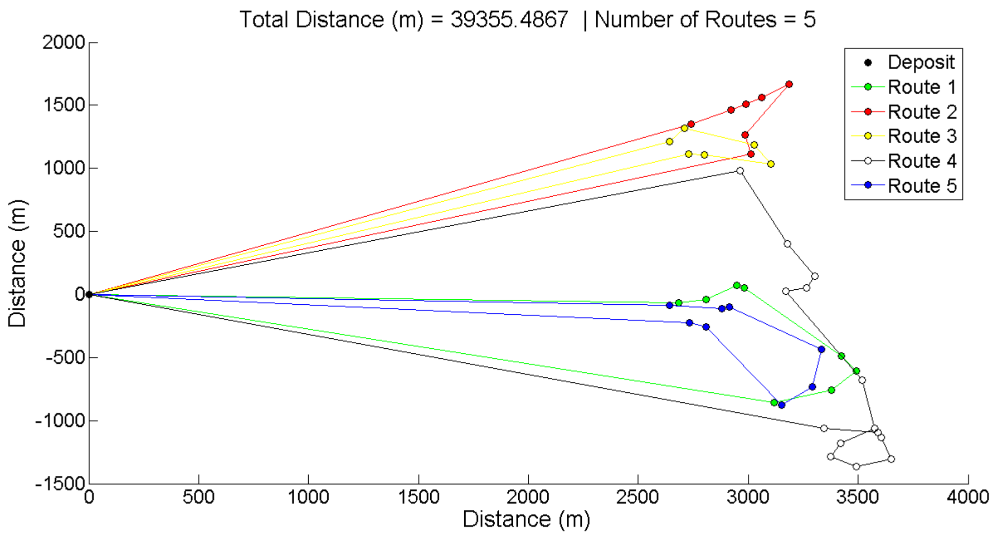


Figure 4.37: Routes results from Matlab simulation with 100 iterations ($N = 36$ at level 1). Total distance of the path is the sum of all the routes distances. The point in position (0,0) represents the deposit.

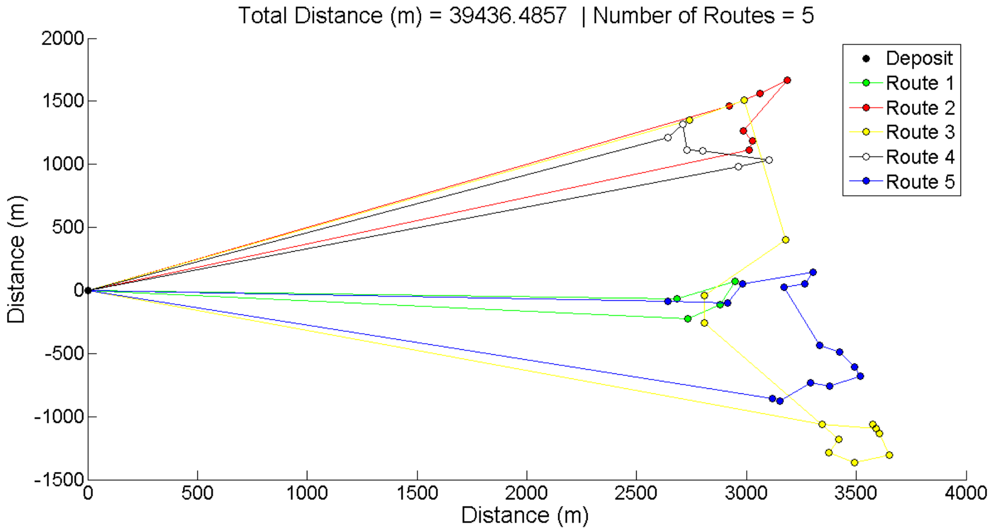


Figure 4.38: Routes results from Matlab simulation with 200 iterations ($N = 36$ at level 1). Total distance of the path is the sum of all the routes distances. The point in position (0,0) represents the deposit.

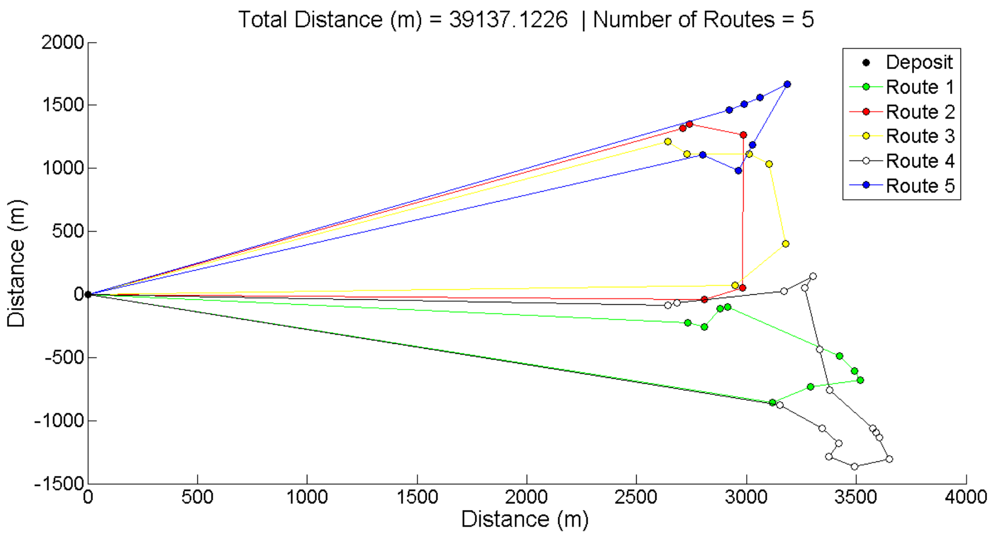


Figure 4.39: Routes results from Matlab simulation with 500 iterations ($N = 36$ at level 1). Total distance of the path is the sum of all the routes distances. The point in position (0,0) represents the deposit.

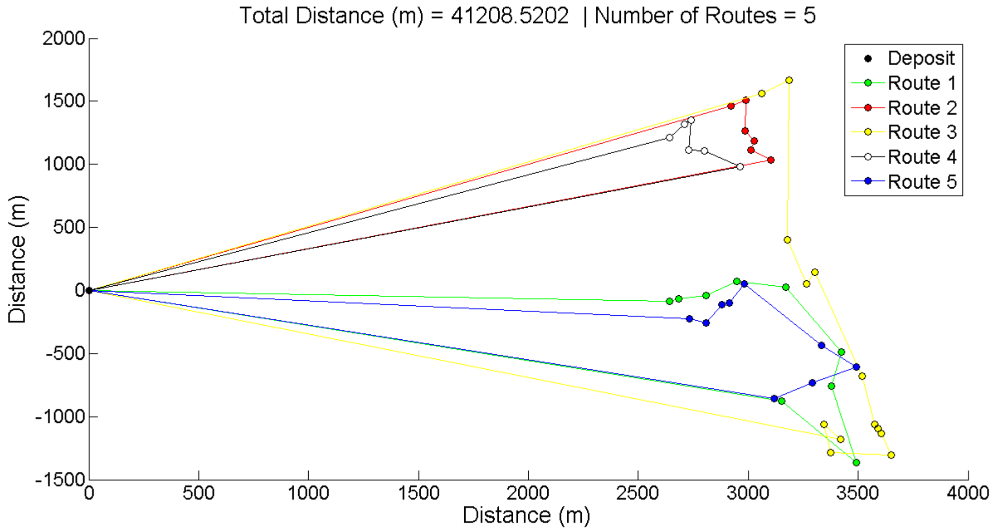


Figure 4.40: Routes results from Matlab simulation with 10 iterations ($N = 100$ at level 1). Total distance of the path is the sum of all the routes distances. The point in position (0,0) represents the deposit.

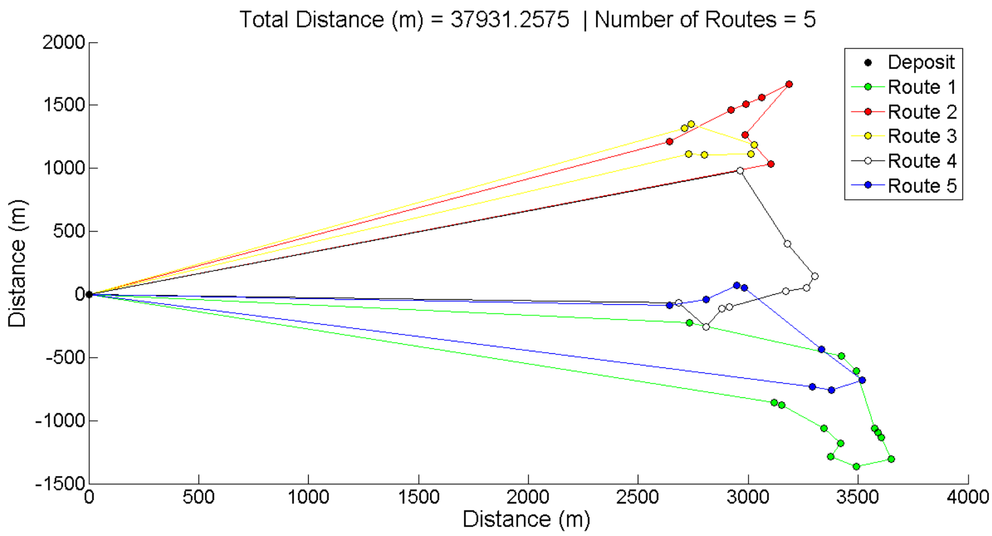


Figure 4.41: Routes results from Matlab simulation with 50 iterations ($N = 100$ at level 1). Total distance of the path is the sum of all the routes distances. The point in position (0,0) represents the deposit.

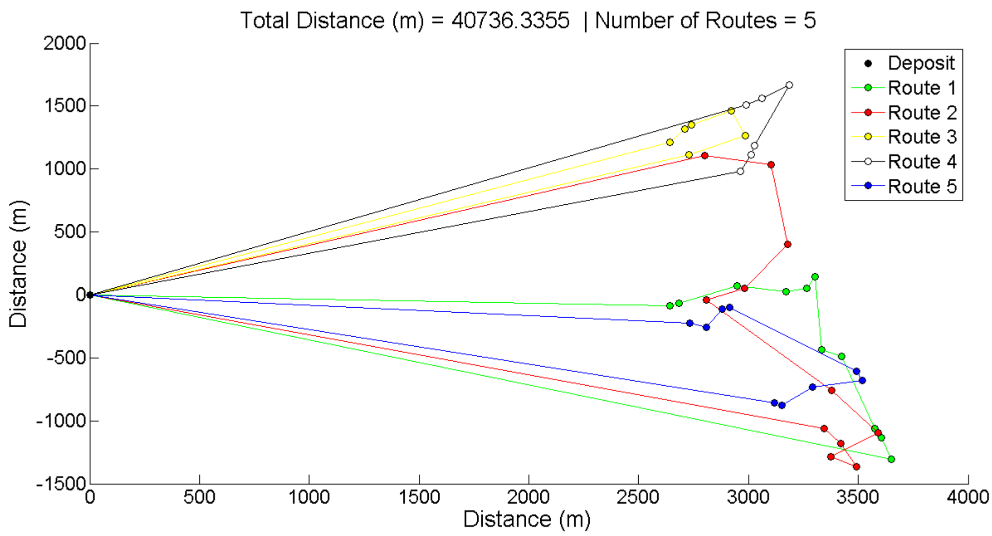


Figure 4.42: Routes results from Matlab simulation with 100 iterations ($N = 100$ at level 1). Total distance of the path is the sum of all the routes distances. The point in position (0,0) represents the deposit.

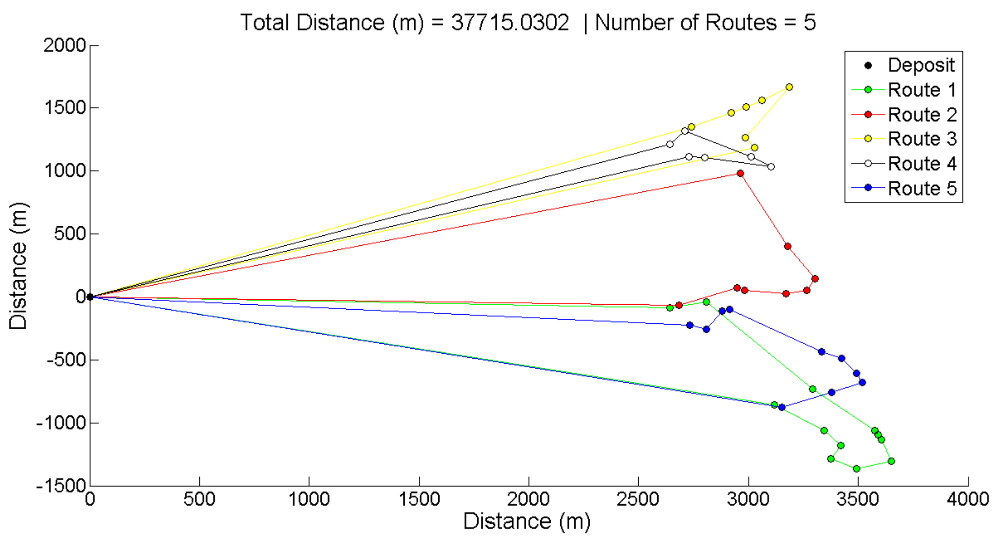


Figure 4.43: Routes results from Matlab simulation with 200 iterations ($N = 100$ at level 1). Total distance of the path is the sum of all the routes distances. The point in position (0,0) represents the deposit.

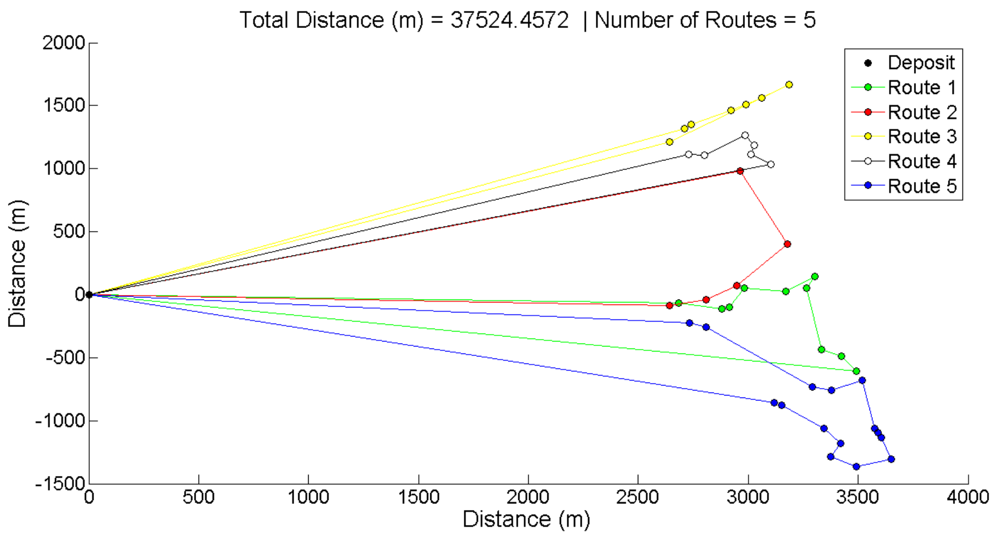


Figure 4.44: Routes results from Matlab simulation with 500 iterations ($N = 100$ at level 1). Total distance of the path is the sum of all the routes distances. The point in position (0,0) represents the deposit.

QGIS routes

Finally have been decided to show just the results of QGIS elaboration of the simulations with an input value of N at level 1 equal to 100 (from Fig. 4.45 to Fig. 4.50), that are the best results obtained from this study.

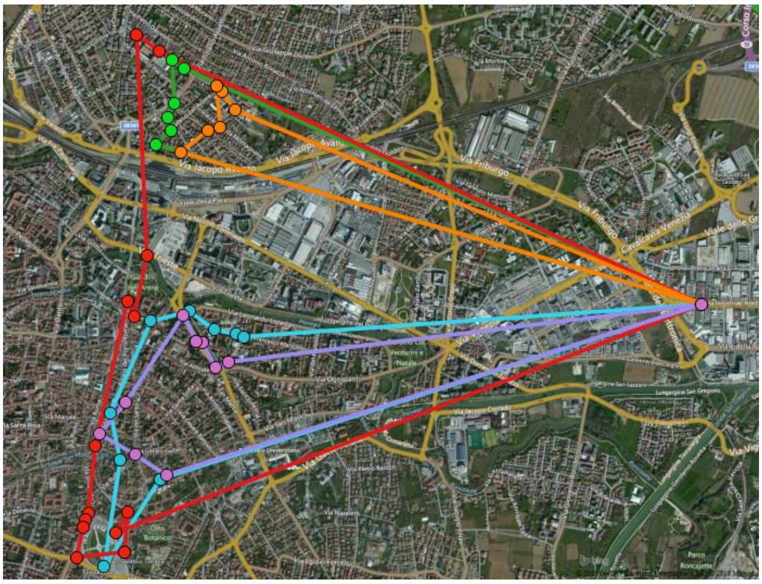


Figure 4.45: Three-phase Algorithm routes planning after 10 iterations ($N = 100$ at level 1) implemented on QGIS platform on Bing map.

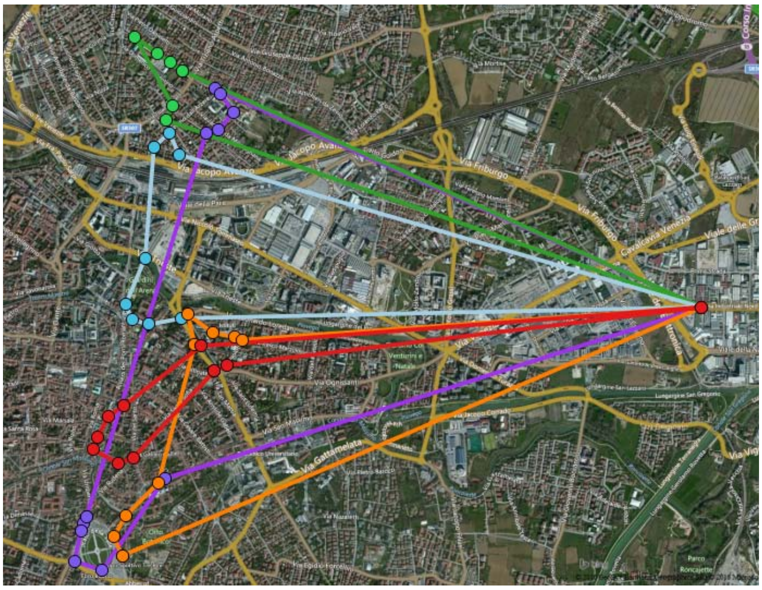


Figure 4.46: Three-phase Algorithm routes planning after 50 iterations ($N = 100$ at level 1) implemented on QGIS platform on Bing map.

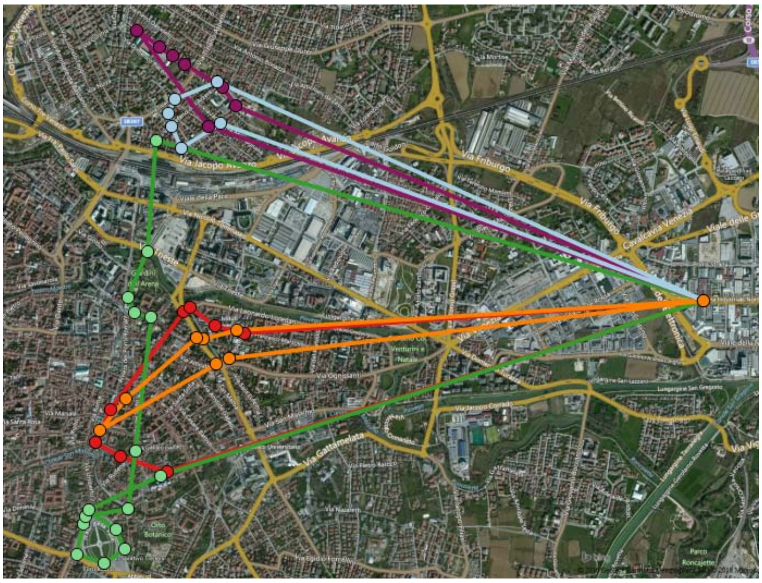


Figure 4.47: Three-phase Algorithm routes planning after 100 iterations ($N = 100$ at level 1) implemented on QGIS platform on Bing map.

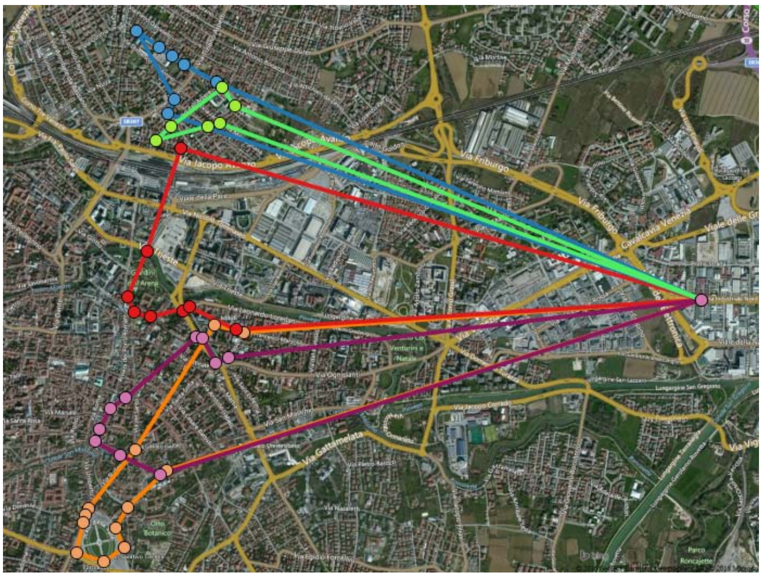
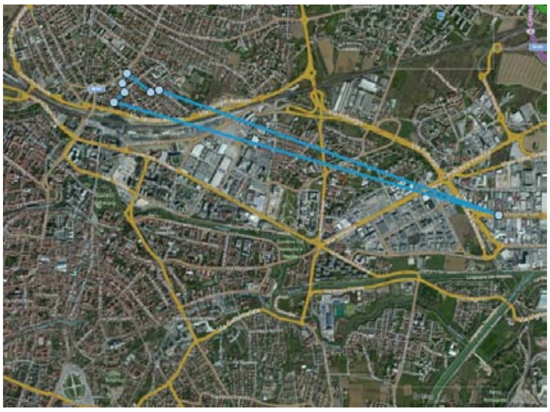


Figure 4.48: Three-phase Algorithm routes planning after 200 iterations ($N = 100$ at level 1) implemented on QGIS platform on Bing map.

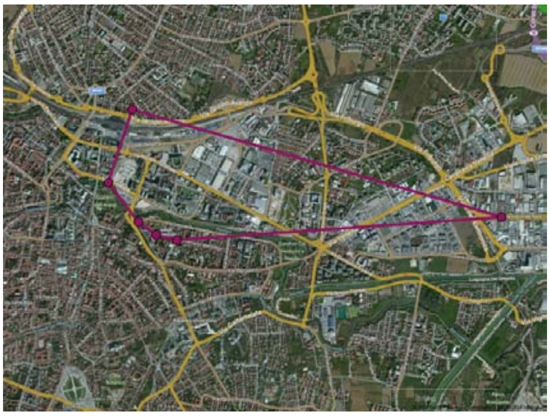
Figure 4.49: Three-phase Algorithm routes planning after 500 iterations ($N = 100$ at level 1) implemented on QGIS platform on Bing map.

(a) Route 1.



(b) Route 2.

(c) Route 3.



(d) Route 4.

(e) Route 5.

Figure 4.50: Three-phase Algorithm routes planning after 500 iterations ($N = 100$ at level 1) implemented on QGIS platform on Bing map.

Chapter 5

Discussion

The goal of this work is the optimization of the vehicle routing for the drop-off MSW collection system, to solve the VRP. In detail the aim is minimized the total distance covered by the collection truck, taking into account its total capacity.

In the city of Padova 65 waste collection points have been collected in the Centre zone and in the North zone, but just 43 satisfy all the hypothesis (in particular the unsorted waste collection, and the drop-off collection system). The first problem that has been met, was the lack of open-source lists about the characterization of the waste drop-off collection bins, in particular their position and numerosity. Therefore just 65 points have been collected, and a complete case study couldn't be done.

The second problem that has been met, was the absence of a complete open-source digitized route net. This means that was possible just a visual comparison between the results of the simulations, and the net of routes.

The algorithm created by Sas Wahid HamzahAll [56] and based on NNA, has gave a total distance of 39.5 Km traveled in 5 routes, in a time process of few seconds. The result is unique and without aleatory. In simple terms (a detailed description has been done in the Section 3.2.2) starting from each point collected and reaching the nearest next point, the algorithm chooses the shortest route plan that satisfy the volume capacity of the vehicles between the solutions that are created. In this way the numerosity of the possible routes is equal to the point reached. In this case 44 included the deposit.

The Three-phase Algorithm can be divided into two levels. One (level 1) to improve the choice of the cluster of the waste collection points that can be collected with one route, and the other (level 2) to choose the shortest route that reach each point just one time in a single cluster. Both levels required different N and X parameters value.

As said in the paragraph 4.3.5, the GA based algorithm should be studied by a sensitivity analysis and so optimized. But just a simple study have been done before the simulations to briefly choose N and X for the route optimization in each cluster (level 2). The analysis has been done in two times, at first the value of N has been fixed and the value of X changed, then vice versa.

The results of the analysis have a rapid decrease in the first part of the graph, while a stabilization in the second part, from an abscissa value of 100 (Fig. 4.8, Fig. 4.9). But the first decreasing of the analysis with X as the variable, is more significance than the second. Also the time-dependent of N is more affecting than X. In conclusion, the values that have been chosen are $X = 50$ and $N = 4$.

This work, is based on real data that have been collected in the city of Padova, but they are not enough and so not representatives of the waste collection in the city. Is not a complete case study. For this reason, the sensitivity analysis doesn't have been done, but it will be necessary if the aim will be find the best possible result.

A scheme of the simulations and the total distance resulting, are shown in Tab. 4.5.

Table 5.1: Simulations parameters. $N = numerosity of population$, $X = iterations$. The last column represents the difference between the result of the current simulation and the first of the sector.

Level 1 N	X	**Level 2** N	X	**Total distance** (Km)	**Number of Routes**	**Distance diff.**
4	10	4	50	44.542	5	0
4	50	4	50	44.876	5	-0.334
4	100	4	50	42.058	5	2.818
4	200	4	50	43.799	5	-1.741
4	500	4	50	41.568	5	2.231
36	10	4	50	42.855	5	0
36	50	4	50	41.900	5	0.955
36	100	4	50	39.355	5	2.545
36	200	4	50	39.436	5	-0.081
36	500	4	50	39.137	5	0.299
100	10	4	50	41.208	5	0
100	50	4	50	37.931	5	3.277
100	100	4	50	40.736	5	-2.805
100	200	4	50	37.715	5	3.021
100	500	4	50	37.525	5	0.190

The results reported in Tab. 5.1, are also shown in extended manner, in the tables of the Section 4.3.5. Important results are the sum of the last columns of each cluster, that represent the total volume carried by the collection truck. With a maximum capacity of 20 m^3, the load is almost always in a range between $18 \div 20m^3$. Therefore the function for the cluster selection, works very well (Algorithm 4.3.5).

The next analysis is described by the Fig. 4.10, Fig. 4.11, Fig. 4.12. They represent the results of the simulations with $N = 4, 36, 100$. The first result is the linearity of the time-dependence which is confirmed by all the simulations, and the slowness of the algorithm.

As regard the efficiency, the distance, decrease in different way in the three graphs. The best fitting curve in Fig. 4.10 and Fig. 4.12 is the quadratic curve. The total distances of the simulations with $N = 36$, is better fitted by an exponential curve almost linear. But the data points fitted, are not enough to give a definitive general rule. Anyway there are negative trends, so the total distances have continuous improvements.

The Fig. 4.13, shows the behavior of the total distance in relation with the value of N at the same X at the level 1. Noticeable is the positive trend of the angular coefficients of the time relations. The angle increase with the number of iterations. This confirm the linearity of the previous graphs.

The points fitted in the Fig. 4.10, Fig. 4.11, Fig. 4.12, are interpolated by cubic function and shown in the Fig. 5.1. The graph gives an easy comprehension of the previous observations. From the firsts simulations to the lasts, the improvement of the total distances is intuitive. A final blue table has been reached.

The best total distances result from each iteration, have been plotted in "The Best Solution History" graphs. All of them, have a characteristic behavior on the first iterations. Have be noticed that less than fluctuations, the trend tend to a plateau, after the 30-40% of total iterations, except the simulation with $N = 100$ and 200 iterations, where the plateau has not yet been reached at the end of the simulation (Fig. 5.4). The best of the "Best Solution History" in Fig. 5.2 and Fig. 5.3, underline as said.

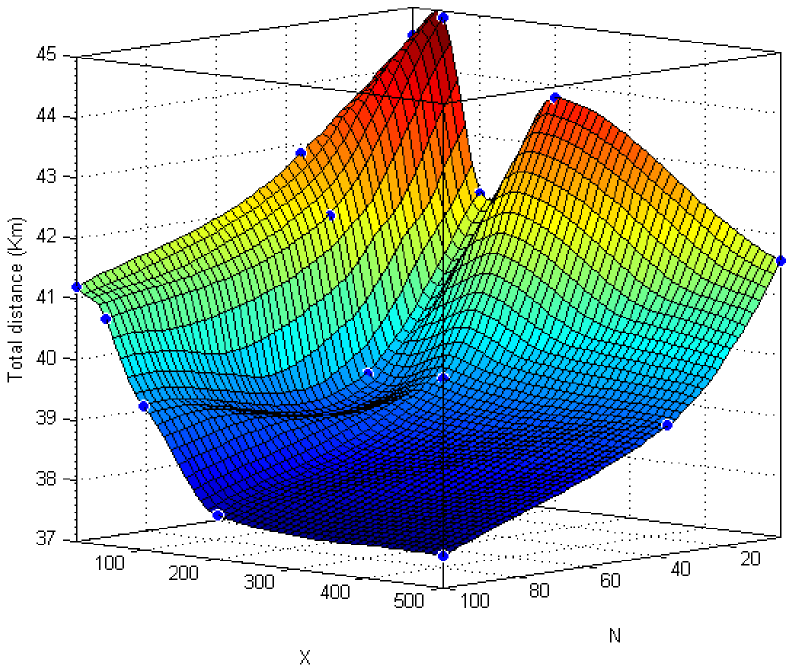


Figure 5.1: 3D view of all the simulations reported in the Tab. 4.5. The scale along z axis, has been increase to improve the visibility of the trend.

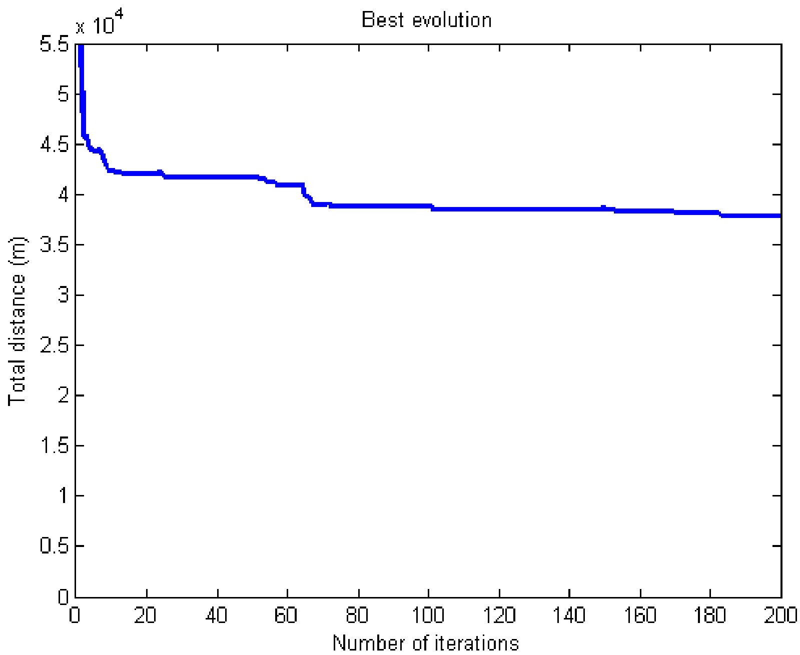


Figure 5.2: The evolution of the best of "The Best Solution History" ($N = 100$, $X = 200$, level 1).

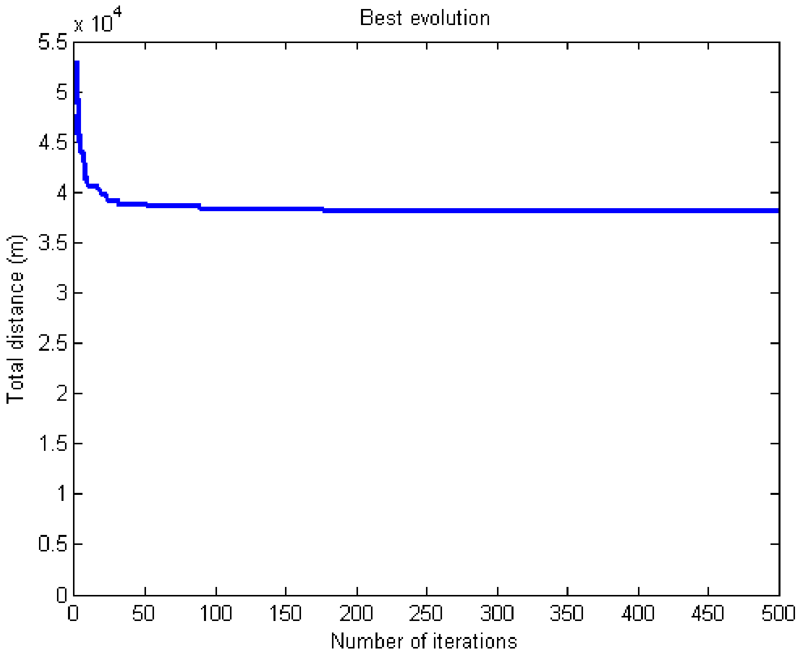


Figure 5.3: The evolution of the best of "The Best Solution History" ($N = 100$, $X = 500$, level 1).

The graphs visible in the last two sections of the chapter Simulations and results ("Matlab routes planning", "QGIS routes"), are obvious. From the depot, 5 routes began. From 10 to 500 iterations, and from $N = 4$ to $N = 100$ is noticeable the trend of the choice of the clusters. In particular, the total distance decrease, if the routes are isolated between them.

The best result reached by the Three-phase Algorithm is 37.5 Km, that is lower than the NNA result of 2000 m. Naturally the difference is not immediately significance, but in long time could develop positive benefits. The same behavior is not repeated by the time-dependence. The meta-heuristic algorithm requires more more time than the heuristic ones. So according to the application, one can be preferred than the other.

The heuristic algorithm, is based on a limited number of solutions available, because the NNA gives a number of possibilities equal to the number of starting points, that are equal to the number of collected points. The unique limitation of an algorithm based on the genetic approach, is the time. Anyway, both the paths, the clusters are well joined. Though the result of the GA based algorithm is better grouped.

An important aspect that has not been treated, is the aleatory of the GA. In fact the GA is based on a casual generation of the first population, that is then selected. The jump in the point with coordinates $X = 100$ and $N = 10$ in Fig. 5.1 is an example of the aleatory of it. Therefore each result is not unique. For the same reason each simulation should be repeated enough times to reduce the uncertainty of the paths. For the goal of this work it is not primary. The highest aim is to verify that the Three-phase Algorithm working well.

All of them give a significant margin for improvements.

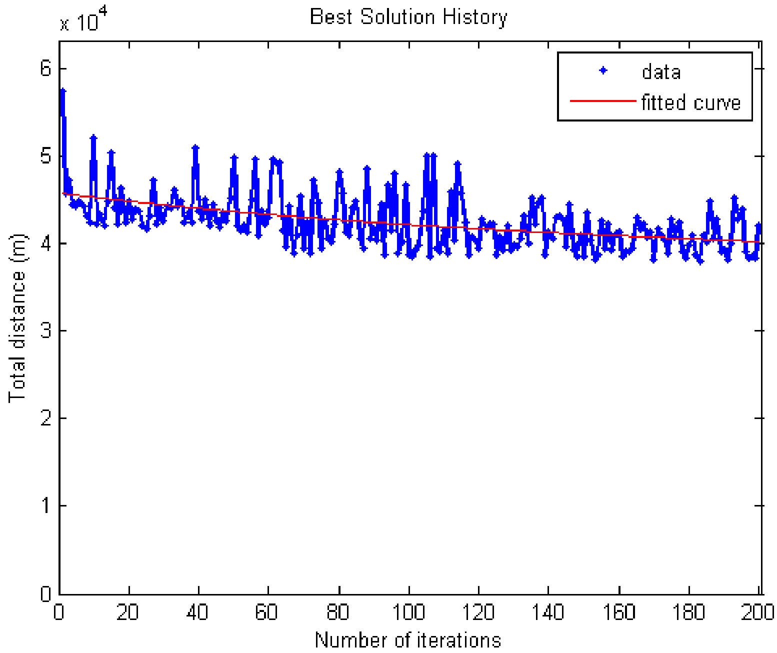


Figure 5.4: Evolution of the 200 iterations simulation ($N = 100$ at level 1). The best of the total distances of each iteration has been plotted. The aleatory based GA produces the oscillating trend.

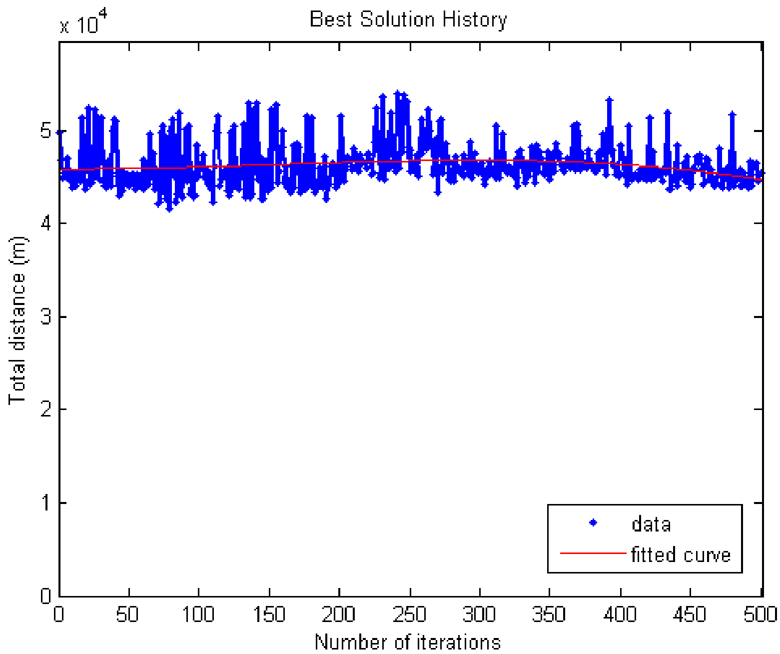


Figure 5.5: Evolution of the 500 iterations simulation ($N = 100$ at level 1). The best of the total distances of each iteration has been plotted. The aleatory based GA produces the oscillating trend.

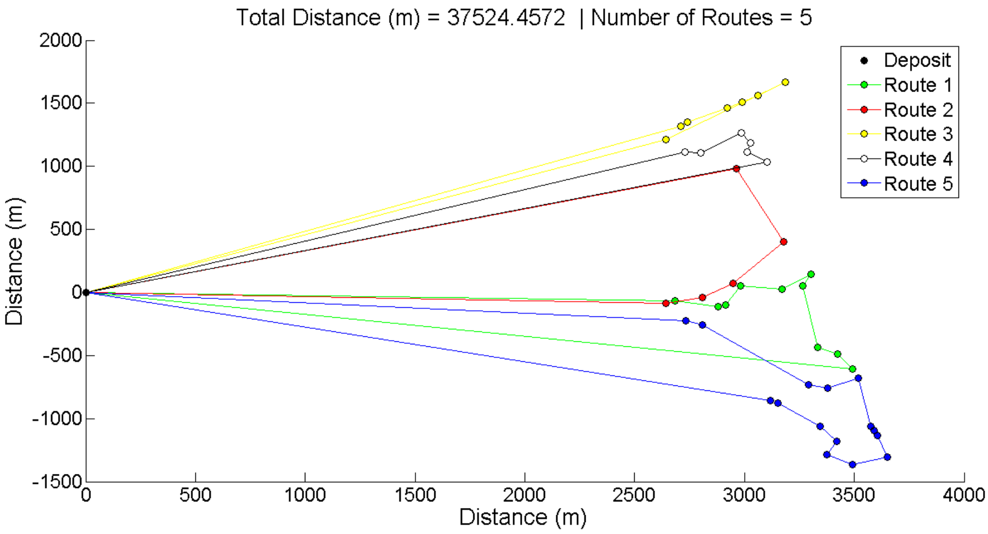


Figure 5.6: Routes results from Matlab simulation with 500 iterations ($N = 100$ at level 1). Total distance of the path is the sum of all the routes distances. The point in position (0,0) represents the deposit.

Figure 5.7: Three-phase Algorithm routes planning after 500 iterations ($N = 100$ at level 1) implemented on QGIS platform on Bing map.

Chapter 6

Conclusions

At the beginning of the entire work, the idea was to implement the NNA based algorithm wrote by Sas Wahid HamzahAll, and to use it for applications and innovations in the waste collection in the city of Padova. Then, the finding of the algorithm wrote by Joseph Kirk to solve the TSP has inspired the construction of a double application of GA based algorithm.

In a general review, the shortest path that have been obtained in this study has been done by the Three-phase Algorithm. A difference of 2000 m has been obtained between the result of the NNA and the best of TPA. It is an improvement of the 5% of the NNA total distance. As regard the time-dependence and the performance of the TPA, it can be improved by an optimization work.

For the intrinsic property of the GA, each result is not unique. Therefore, to reduce the uncertainty, the simulations should be repeated more times. This gives the possibility to choose between more than one solution of the same problem, with almost the same quality; while the result of the NNA is unique and without aleatory.

The potentiality of the GA based algorithm are not completed explored, as suggested by the trend of the Iteration-Time/Efficiency graphs. So the unique limitation of the algorithm based on the genetic approach is the time.

The absence of open-source data as regard the digitized roads net and the location of the waste collection points in the specific areas of interest has been hindered the development of the case study in the city of Padova, which could be shown the defects or the potentiality of the Algorithm. This means that it was possible to do just a visual comparison between the results of the simulations and the net of roads.

Anyway a meta-heuristic algorithm has been developed, that, after small adjustments, could be immediately use in a MSW management. In a simple example, the Final GA can be used to a complete collection of the MSW that remain permanent, while the Joseph Kirk's algorithm could be use daily, to optimize the route planning inside each cluster, and to solve the unexpected daily roads lack, even simultaneously with the collection.

A future improvement will be the implementation of the Three-phase Algorithm in Python code, to solve the VRP in the QGIS platform. The routes resulting by this work are based on Euclidean distance. The algorithm can be applied directly on the roads net, measuring the roads distances. This is possible if the digitized roads net and the location of the waste collection points will be available.

The lack of availability by the Local Authorities to exchange data give this improvement not yet possible in the city of Padova.

In conclusion, this work has presented a new application of the well known Genetic Algorithm and, in particular, the double application of the same in one model to solve the Vehicle Routing Problem of waste collection and transport, that hasn't a single perfect solution. Keeping in mind this, while a good result have been reached (5%), the Three-phase Algorithm still has potentiality that can be applied in a lot of other sectors or kinds of waste collection system (i.e. curbside waste collection system).

Bibliography

[1] Thomas H. Christensen, *Solid Waste Technology & Management, Volume 1*, Wiley, 2011.

[2] *ChemMatters, February 2014 Teacher's Guide*

[3] ASTM, *Standard Terminology for Waste and Waste Management*, D 5681-98a.

[4] Publications Office of the European Union, *Being wise with waste: the EU's approach to waste management*, 2010.

[5] US EPA, *Decision maker's guide to solid waste management, Volume 2, Glossary.*

[6] Cossu lectures, *Waste collection*, Unipd, a.y. 2013/2014.

[7] Abfallwirtschaftsbetrieb Munchen, *Waste Management In The City Of Munich*, 2009.

[8] Robert Breeze, *Municipal Solid Waste Management in Dar es Salaam, Draft baseline analysis*, 2012.

[9] Paula Estevez, *Management of Municipal Solid Waste in Santiago, Chile: Assessing Waste-to-Energy Possibilities*, 2003.

[10] Brautigam K.-R, et al., *Risk habitat megacity, Waste Management in Santiago de Chile*, Redisa, 2008.

[11] Berlin Senate Department for Urban Development and the Environment Communication, *Environment Municipal waste management in Berlin*, 2013.

[12] Ecoembes, *Gua para aprender a separar residuos domsticos en Madrid*, Madrid.

[13] Shibuya city waste collection office, *How to Sort and Dispose of Resources and Garbage Properly*, 2014.

[14] Japan Environmental Sanitation Center, *Solid Waste Management and Recycling Technology of Japan, Toward a Sustainable Socity*, 2012.

[15] *Waste Management in Tokyo 1, Promotion of 3Rs.*

[16] Saint-Laurent Montreal, *Collection from A to Z-2014*, 2014.

[17] Li Zhen-shan, et al., *Municipal solid waste management in Beijing City*, Waste Management, 2009.

[18] Gurdal Kanat, *Municipal solid-waste management in Istanbul*, Waste Management, 2010.

[19] City of New Berlin, *City-Wide New Garbage and Recycling Program, Frequently Asked Questions*, 2015.

[20] A. M. Zubkov, *Euler and Combinatorial Calculus*, conference "Leonhard Euler and Modern Mathematics", 2010.

[21] F. Harary, *Graph Theory*, Addison-Wesley, 1970, p.3, p.8-9, p.64.

[22] V. Chvatal, *Tough graphs and hamiltonian circuits*, Noth-Holland Publishing Company, 1972.

[23] M. DeLeon, *A Study of Sufficient Conditions for Hamiltonian Cycles.*

[24] N. Christofides, *The Vehicle Routing Problem*, Numdam, 1976.

[25] M. Fabbri, *Pianificazione ed ottimizzazione dei percorsi nella logistica dei magazzini industriali*, a.y. 2011-2012, p.33-35.

[26] J. Grfenstette et al., *Genetic Algorithm for the Traveling Salesman Problem*,Psychology Press, 1985, p.160-168.

[27] B. Eksioglu et al., *The vehicle routing problem: A taxonomic review*, Elsevier, 2009.

[28] S. Chatterjee et al., *Genetic algorithms and traveling salesman problems*, Elsevier, 1995.

[29] M. Mitchell, *Genetic Algorithms: An Overview*, 1995.

[30] M. Mitchell, *An Introduction to Genetic Algorithms*, The MIT Press, 1999.

[31] Marius M. Solomon, *Algorithms for the vehicle routing and scheduling problems with time window constraints*, Jstor, 1985.

[32] K. Chidananda Gowda and G. Krishna, *The Condensed Nearest Neighbor Rule Using the Concept of Mutual Nearest Neighborhood*, IEEE Transactions on Information Theory, 1979.

[33] Rhishikesh, *Graph Theory Origin and Seven Bridges of Konigsberg.*

[34] AcegasAps, *Gestione rifiuti urbani, Piano economico finanziario*, 2014.

[35] Gamal Abd El-Nasser A. Said, et al., *A Comparative Study of Meta-heuristic Algorithms for Solving Quadratic Assignment Problem*, IJACSA, 2014.

[36] R. Nallusamy et al., *Optimization of Multiple Vehicle Routing Problems using Approximation algorithms*, International Journal of Engineering Science and Technology, 2009.

[37] Zahra Beheshti et al., *A Review of Population-based Meta-HeuristicAlgorithms*, ICSRS, 2013.

[38] Gilbert Laporte, *The Vehicle Routing Problem: An overview of exact and approximate algorithms*, European Journal of Operational Research, 1992.

[39] www.data.oecd.org, September 16th, 2015.

[40] www.web.mit.edu, September 16th, 2015.

[41] www.cityoflondon.gov.uk, September 16th, 2015.

[42] www.capetown.gov.za, September 16th, 2015.

[43] www.praha.eu, September 16th, 2015.

[44] www.waste-management-world.com, September 16th, 2015.

[45] www.howtogermany.com, September 16th, 2015.

[46] www.city.shibuya.tokyo.jp, September 16th, 2015.

[47] www.kankyo.metro.tokyo.jp, September 16th, 2015.

[48] www.ville.montreal.qc.ca, September 16th, 2015.

[49] www.moscowrecycling.com, September 17th, 2015.

[50] www.sydneywaste.com, September 17th, 2015.

[51] www.cityofsydney.nsw.gov.au, September 17th, 2015.

[52] www.gruener-punkt.de, September 17th, 2015.

[53] www.rosalind.info, November 10th, 2015.

[54] www.mathworld.wolfram.com, November 10th, 2015.

[55] www.maths.manchester.ac.uk, November 10th, 2015.

[56] www.freesourcecode.net, November 19th, 2015.

[57] www.mathworks.com, October 22th, 2015.

[58] www.personal.kent.edu, November 10th, 2015.

[59] www.nomachetejuggling.com, December 28th, 2015.

[60] www.bcu.copsewood.net, December 28th, 2015.

[61] www.bing.com, January 4th, 2016.

[62] www.gruppohera.it, January 4th, 2016.

[63] www.padovanet.it, January 4th, 2016.

[64] www.students.ceid.upatras.gr January 5th, 2016.

[65] www.docs.qgis.org, January 19th, 2016.

[66] www.web.gps.caltech.edu, January 19th, 2016.

Annex I

Tables

The following tables, represent the results of the simulations of Three-phase Algorithm from 10 to 500 iterations (N at level 1 of 4 and 36). The first column shows the id of the single waste collection point, the second column shows the route of membership, the third column gives the sequence of the waste collection points in each route, the fourth and fifth the planar coordinates (WGS 84/UTM zone 33N) of the collection points, and the last column shows the capacity of the collection truck required by the unsorted waste collection bins in each waste collection points and the total volume of waste collected by a truck in a single route.

Table I.1: Routes description from Three-phase Algorithm with 10 iterations ($N = 4$, level 1).

id	n^o of single route	Order in single route	Coordinate x	Coordinate y	Demand (m^3)
6	1	1	255533.522407401	5032538.410363290	1.5
1	1	2	255628.583800869	5032726.808474400	0.75
11	1	3	255785.826530947	5033269.238715940	2.25
10	1	4	255881.750174129	5033240.180559440	0.75
7	1	5	256104.299103770	5033289.230643370	3.6
9	1	6	256366.352106576	5033148.625915130	3.6
8	1	7	256409.377579713	5033131.223553460	3.6
44	1	8	259051.864391000	5033215.737003990	0
2	1	9	255935.337870106	5032357.086090610	1.5
4	1	10	255898.945508198	5032336.245989560	0.75
3	1	11	255558.702458421	5031848.005041680	0.75
14	1	12	255630.854341550	5032038.684513050	0.75
6	1	13	255533.522407401	5032538.410363290	**19.8**
18	2	1	255446.949282576	5032080.543377010	0.75
19	2	2	255400.936652359	5031908.097526770	0.75
16	2	3	255675.372616814	5031927.586545350	0.75
44	2	4	259051.864391000	5033215.737003990	0
21	2	5	256040.288863683	5034326.884370310	3.6
20	2	6	255948.069021439	5034250.244479680	3.6
13	2	7	255873.196458077	5033619.410464310	3.6
12	2	8	255749.912334742	5033357.616949720	0.75
17	2	9	256244.288081726	5033178.380723640	5.4
5	2	10	255673.673792252	5032455.027433400	0.75
18	2	11	255446.949282576	5032080.543377010	**19.95**
23	3	1	256128.482257811	5034677.493612770	3.6
26	3	2	256061.557482728	5034727.598430270	1.8
25	3	3	255990.595664290	5034779.640914700	3.6
27	3	4	255864.117118510	5034882.906177090	1.8
24	3	5	256065.239606521	5034480.234596910	3.6
22	3	6	256023.962045764	5034400.838618200	3.6
29	3	7	255478.382061578	5032155.201535760	0.75
28	3	8	255462.669529721	5032119.986247760	0.75
44	3	9	259051.864391000	5033215.737003990	0
23	3	10	256128.482257811	5034677.493612770	**19.5**
35	4	1	256069.804336837	5033267.764357700	0.75
36	4	2	256090.544432397	5034200.434097260	3.6
34	4	3	256250.376014123	5034319.347926460	1.8
33	4	4	256323.512120709	5034332.311882570	3.6
31	4	5	256409.805148712	5034429.046381020	1.8
32	4	6	256343.624034497	5034535.861678260	5.4
30	4	7	256310.541395512	5034569.743184420	1.8
44	4	8	259051.864391000	5033215.737003990	0
35	4	9	256069.804336837	5033267.764357700	**18.75**
15	5	1	255706.712233325	5032155.343681460	0.75
43	5	2	255759.693089803	5032482.283847830	1.5
41	5	3	255560.651586477	5032609.286239500	2.25
42	5	4	255720.399256876	5032783.322989710	1.5
39	5	5	256138.197891210	5033114.632764460	0.75
38	5	6	256173.487694673	5033105.142617960	3.6
40	5	7	256244.534849801	5032959.697884480	0.75
37	5	8	256319.746563686	5032990.427849010	7.2
44	5	9	259051.864391000	5033215.737003990	0
15	5	10	255706.712233325	5032155.343681460	**18.3**

Table I.2: Routes description from Three-phase Algorithm with 50 iterations ($N = 4$, level 1).

id	n^o of single route	Order in single route	Coordinate x	Coordinate y	Demand (m^3)
5	1	1	255673.673792252	5032455.027433400	0.75
43	1	2	255759.693089803	5032482.283847830	1.5
4	1	3	255898.945508198	5032336.245989560	0.75
2	1	4	255935.337870106	5032357.086090610	1.5
44	1	5	259051.864391000	5033215.737003990	0
8	1	6	256409.377579713	5033131.223553460	3.6
17	1	7	256244.288081726	5033178.380723640	5.4
39	1	8	256138.197891210	5033114.632764460	0.75
40	1	9	256244.534849801	5032959.697884480	0.75
42	1	10	255720.399256876	5032783.322989710	1.5
1	1	11	255628.583800869	5032726.808474400	0.75
41	1	12	255560.651586477	5032609.286239500	2.25
5	1	13	255673.673792252	5032455.027433400	**19.5**
20	2	1	255948.069021439	5034250.244479680	3.6
38	2	2	256173.487694673	5033105.142617960	3.6
37	2	3	256319.746563686	5032990.427849010	7.2
28	2	4	255462.669529721	5032119.986247760	0.75
44	2	5	259051.864391000	5033215.737003990	0
33	2	6	256323.512120709	5034332.311882570	3.6
20	2	7	255948.069021439	5034250.244479680	**18.75**
27	3	1	255864.117118510	5034882.906177090	1.8
25	3	2	255990.595664290	5034779.640914700	3.6
30	3	3	256310.541395512	5034569.743184420	1.8
32	3	4	256343.624034497	5034535.861678260	5.4
31	3	5	256409.805148712	5034429.046381020	1.8
44	3	6	259051.864391000	5033215.737003990	0
34	3	7	256250.376014123	5034319.347926460	1.8
36	3	8	256090.544432397	5034200.434097260	3.6
27	3	9	255864.117118510	5034882.906177090	**19.8**
23	4	1	256128.482257811	5034677.493612770	3.6
26	4	2	256061.557482728	5034727.598430270	1.8
24	4	3	256065.239606521	5034480.234596910	3.6
22	4	4	256023.962045764	5034400.838618200	3.6
21	4	5	256040.288863683	5034326.884370310	3.6
35	4	6	256069.804336837	5033267.764357700	0.75
6	4	7	255533.522407401	5032538.410363290	1.5
29	4	8	255478.382061578	5032155.201535760	0.75
44	4	9	259051.864391000	5033215.737003990	0
23	4	10	256128.482257811	5034677.493612770	**19.2**
19	5	1	255400.936652359	5031908.097526770	0.75
18	5	2	255446.949282576	5032080.543377010	0.75
44	5	3	259051.864391000	5033215.737003990	0
9	5	4	256366.352106576	5033148.625915130	3.6
7	5	5	256104.299103770	5033289.230643370	3.6
10	5	6	255881.750174129	5033240.180559440	0.75
11	5	7	255785.826530947	5033269.238715940	2.25
12	5	8	255749.912334742	5033357.616949720	0.75
13	5	9	255873.196458077	5033619.410464310	3.6
15	5	10	255706.712233325	5032155.343681460	0.75
14	5	11	255630.854341550	5032038.684513050	0.75
16	5	12	255675.372616814	5031927.586545350	0.75
3	5	13	255558.702458421	5031848.005041680	0.75
19	5	14	255400.936652359	5031908.097526770	**19.05**

Table I.3: Routes description from Three-phase Algorithm with 100 iterations ($N = 4$, level 1).

id	n^o of single route	Order in single route	Coordinate x	Coordinate y	Demand (m^3)
8	1	1	256409.377579713	5033131.223553460	3.6
9	1	2	256366.352106576	5033148.625915130	3.6
17	1	3	256244.288081726	5033178.380723640	5.4
7	1	4	256104.299103770	5033289.230643370	3.6
6	1	5	255533.522407401	5032538.410363290	1.5
5	1	6	255673.673792252	5032455.027433400	0.75
4	1	7	255898.945508198	5032336.245989560	0.75
44	1	8	259051.864391000	5033215.737003990	0
8	1	9	256409.377579713	5033131.223553460	**19.2**
13	2	1	255873.196458077	5033619.410464310	3.6
12	2	2	255749.912334742	5033357.616949720	0.75
11	2	3	255785.826530947	5033269.238715940	2.25
10	2	4	255881.750174129	5033240.180559440	0.75
44	2	5	259051.864391000	5033215.737003990	0
32	2	6	256343.624034497	5034535.861678260	5.4
24	2	7	256065.239606521	5034480.234596910	3.6
36	2	8	256090.544432397	5034200.434097260	3.6
13	2	9	255873.196458077	5033619.410464310	**19.95**
22	3	1	256023.962045764	5034400.838618200	3.6
21	3	2	256040.288863683	5034326.884370310	3.6
20	3	3	255948.069021439	5034250.244479680	3.6
44	3	4	259051.864391000	5033215.737003990	0
23	3	5	256128.482257811	5034677.493612770	3.6
25	3	6	255990.595664290	5034779.640914700	3.6
27	3	7	255864.117118510	5034882.906177090	1.8
22	3	8	256023.962045764	5034400.838618200	**19.8**
30	4	1	256310.541395512	5034569.743184420	1.8
44	4	2	259051.864391000	5033215.737003990	0
2	4	3	255935.337870106	5032357.086090610	1.5
3	4	4	255558.702458421	5031848.005041680	0.75
14	4	5	255630.854341550	5032038.684513050	0.75
15	4	6	255706.712233325	5032155.343681460	0.75
16	4	7	255675.372616814	5031927.586545350	0.75
18	4	8	255446.949282576	5032080.543377010	0.75
19	4	9	255400.936652359	5031908.097526770	0.75
29	4	10	255478.382061578	5032155.201535760	0.75
28	4	11	255462.669529721	5032119.986247760	0.75
1	4	12	255628.583800869	5032726.808474400	0.75
35	4	13	256069.804336837	5033267.764357700	0.75
34	4	14	256250.376014123	5034319.347926460	1.8
33	4	15	256323.512120709	5034332.311882570	3.6
31	4	16	256409.805148712	5034429.046381020	1.8
26	4	17	256061.557482728	5034727.598430270	1.8
30	4	18	256310.541395512	5034569.743184420	**19.8**
37	5	1	256319.746563686	5032990.427849010	7.2
40	5	2	256244.534849801	5032959.697884480	0.75
38	5	3	256173.487694673	5033105.142617960	3.6
39	5	4	256138.197891210	5033114.632764460	0.75
42	5	5	255720.399256876	5032783.322989710	1.5
41	5	6	255560.651586477	5032609.286239500	2.25
43	5	7	255759.693089803	5032482.283847830	1.5
44	5	8	259051.864391000	5033215.737003990	0
37	5	9	256319.746563686	5032990.427849010	**17.55**

Table I.4: Routes description from Three-phase Algorithm with 200 iterations ($N = 4$, level 1).

id	n^o of single route	Order in single route	Coordinate x	Coordinate y	Demand (m^3)
44	1	1	259051.864391000	5033215.737003990	0
32	1	2	256343.624034497	5034535.861678260	5.4
23	1	3	256128.482257811	5034677.493612770	3.6
24	1	4	256065.239606521	5034480.234596910	3.6
36	1	5	256090.544432397	5034200.434097260	3.6
9	1	6	256366.352106576	5033148.625915130	3.6
44	1	7	259051.864391000	5033215.737003990	**19.8**
7	2	1	256104.299103770	5033289.230643370	3.6
38	2	2	256173.487694673	5033105.142617960	3.6
8	2	3	256409.377579713	5033131.223553460	3.6
44	2	4	259051.864391000	5033215.737003990	0
40	2	5	256244.534849801	5032959.697884480	0.75
43	2	6	255759.693089803	5032482.283847830	1.5
28	2	7	255462.669529721	5032119.986247760	0.75
29	2	8	255478.382061578	5032155.201535760	0.75
41	2	9	255560.651586477	5032609.286239500	2.25
1	2	10	255628.583800869	5032726.808474400	0.75
7	2	11	256104.299103770	5033289.230643370	**17.55**
17	3	1	256244.288081726	5033178.380723640	5.4
39	3	2	256138.197891210	5033114.632764460	0.75
11	3	3	255785.826530947	5033269.238715940	2.25
12	3	4	255749.912334742	5033357.616949720	0.75
13	3	5	255873.196458077	5033619.410464310	3.6
44	3	6	259051.864391000	5033215.737003990	0
37	3	7	256319.746563686	5032990.427849010	7.2
17	3	8	256244.288081726	5033178.380723640	**19.95**
10	4	1	255881.750174129	5033240.180559440	0.75
6	4	2	255533.522407401	5032538.410363290	1.5
14	4	3	255630.854341550	5032038.684513050	0.75
16	4	4	255675.372616814	5031927.586545350	0.75
3	4	5	255558.702458421	5031848.005041680	0.75
19	4	6	255400.936652359	5031908.097526770	0.75
18	4	7	255446.949282576	5032080.543377010	0.75
15	4	8	255706.712233325	5032155.343681460	0.75
4	4	9	255898.945508198	5032336.245989560	0.75
44	4	10	259051.864391000	5033215.737003990	0
31	4	11	256409.805148712	5034429.046381020	1.8
33	4	12	256323.512120709	5034332.311882570	3.6
27	4	13	255864.117118510	5034882.906177090	1.8
20	4	14	255948.069021439	5034250.244479680	3.6
35	4	15	256069.804336837	5033267.764357700	0.75
10	4	16	255881.750174129	5033240.180559440	**19.05**
5	5	1	255673.673792252	5032455.027433400	0.75
2	5	2	255935.337870106	5032357.086090610	1.5
44	5	3	259051.864391000	5033215.737003990	0
34	5	4	256250.376014123	5034319.347926460	1.8
30	5	5	256310.541395512	5034569.743184420	1.8
26	5	6	256061.557482728	5034727.598430270	1.8
25	5	7	255990.595664290	5034779.640914700	3.6
22	5	8	256023.962045764	5034400.838618200	3.6
21	5	9	256040.288863683	5034326.884370310	3.6
42	5	10	255720.399256876	5032783.322989710	1.5
5	5	11	255673.673792252	5032455.027433400	**19.95**

Table I.5: Routes description from Three-phase Algorithm with 500 iterations ($N = 4$, level 1).

id	n^o of single route	Order in single route	Coordinate x	Coordinate y	Demand (m^3)
17	1	1	256244.288081726	5033178.380723640	5.4
13	1	2	255873.196458077	5033619.410464310	3.6
5	1	3	255673.673792252	5032455.027433400	0.75
2	1	4	255935.337870106	5032357.086090610	1.5
44	1	5	259051.864391000	5033215.737003990	0
8	1	6	256409.377579713	5033131.223553460	3.6
9	1	7	256366.352106576	5033148.625915130	3.6
17	1	8	256244.288081726	5033178.380723640	**18.45**
11	2	1	255785.826530947	5033269.238715940	2.25
1	2	2	255628.583800869	5032726.808474400	0.75
29	2	3	255478.382061578	5032155.201535760	0.75
28	2	4	255462.669529721	5032119.986247760	0.75
18	2	5	255446.949282576	5032080.543377010	0.75
19	2	6	255400.936652359	5031908.097526770	0.75
3	2	7	255558.702458421	5031848.005041680	0.75
14	2	8	255630.854341550	5032038.684513050	0.75
16	2	9	255675.372616814	5031927.586545350	0.75
15	2	10	255706.712233325	5032155.343681460	0.75
44	2	11	259051.864391000	5033215.737003990	0
30	2	12	256310.541395512	5034569.743184420	1.8
36	2	13	256090.544432397	5034200.434097260	3.6
20	2	14	255948.069021439	5034250.244479680	3.6
12	2	15	255749.912334742	5033357.616949720	0.75
11	2	16	255785.826530947	5033269.238715940	**18.75**
34	3	1	256250.376014123	5034319.347926460	1.8
33	3	2	256323.512120709	5034332.311882570	3.6
44	3	3	259051.864391000	5033215.737003990	0
31	3	4	256409.805148712	5034429.046381020	1.8
32	3	5	256343.624034497	5034535.861678260	5.4
23	3	6	256128.482257811	5034677.493612770	3.6
21	3	7	256040.288863683	5034326.884370310	3.6
34	3	8	256250.376014123	5034319.347926460	**19.8**
35	4	1	256069.804336837	5033267.764357700	0.75
22	4	2	256023.962045764	5034400.838618200	3.6
24	4	3	256065.239606521	5034480.234596910	3.6
27	4	4	255864.117118510	5034882.906177090	1.8
25	4	5	255990.595664290	5034779.640914700	3.6
26	4	6	256061.557482728	5034727.598430270	1.8
44	4	7	259051.864391000	5033215.737003990	0
38	4	8	256173.487694673	5033105.142617960	3.6
39	4	9	256138.197891210	5033114.632764460	0.75
35	4	10	256069.804336837	5033267.764357700	**19.5**
4	5	1	255898.945508198	5032336.245989560	0.75
43	5	2	255759.693089803	5032482.283847830	1.5
6	5	3	255533.522407401	5032538.410363290	1.5
41	5	4	255560.651586477	5032609.286239500	2.25
42	5	5	255720.399256876	5032783.322989710	1.5
10	5	6	255881.750174129	5033240.180559440	0.75
7	5	7	256104.299103770	5033289.230643370	3.6
44	5	8	259051.864391000	5033215.737003990	0
37	5	9	256319.746563686	5032990.427849010	7.2
40	5	10	256244.534849801	5032959.697884480	0.75
4	5	11	255898.945508198	5032336.245989560	**19.8**

Table I.6: Routes description from Three-phase Algorithm with 10 iterations ($N = 36$, level 1).

id	n^o of single route	Order in single route	Coordinate x	Coordinate y	Demand (m^3)
4	1	1	255898.945508198	5032336.245989560	0.75
2	1	2	255935.337870106	5032357.086090610	1.5
8	1	3	256409.377579713	5033131.223553460	3.6
44	1	4	259051.864391000	5033215.737003990	0
31	1	5	256409.805148712	5034429.046381020	1.8
30	1	6	256310.541395512	5034569.743184420	1.8
25	1	7	255990.595664290	5034779.640914700	3.6
27	1	8	255864.117118510	5034882.906177090	1.8
1	1	9	255628.583800869	5032726.808474400	0.75
6	1	10	255533.522407401	5032538.410363290	1.5
29	1	11	255478.382061578	5032155.201535760	0.75
28	1	12	255462.669529721	5032119.986247760	0.75
3	1	13	255558.702458421	5031848.005041680	0.75
4	1	14	255898.945508198	5032336.245989560	**19.35**
24	2	1	256065.239606521	5034480.234596910	3.6
22	2	2	256023.962045764	5034400.838618200	3.6
21	2	3	256040.288863683	5034326.884370310	3.6
20	2	4	255948.069021439	5034250.244479680	3.6
44	2	5	259051.864391000	5033215.737003990	0
23	2	6	256128.482257811	5034677.493612770	3.6
26	2	7	256061.557482728	5034727.598430270	1.8
24	2	8	256065.239606521	5034480.234596910	**19.8**
14	3	1	255630.854341550	5032038.684513050	0.75
19	3	2	255400.936652359	5031908.097526770	0.75
18	3	3	255446.949282576	5032080.543377010	0.75
10	3	4	255881.750174129	5033240.180559440	0.75
11	3	5	255785.826530947	5033269.238715940	2.25
12	3	6	255749.912334742	5033357.616949720	0.75
13	3	7	255873.196458077	5033619.410464310	3.6
17	3	8	256244.288081726	5033178.380723640	5.4
9	3	9	256366.352106576	5033148.625915130	3.6
44	3	10	259051.864391000	5033215.737003990	0
16	3	11	255675.372616814	5031927.586545350	0.75
14	3	12	255630.854341550	5032038.684513050	**19.35**
44	4	1	259051.864391000	5033215.737003990	0
7	4	2	256104.299103770	5033289.230643370	3.6
35	4	3	256069.804336837	5033267.764357700	0.75
36	4	4	256090.544432397	5034200.434097260	3.6
34	4	5	256250.376014123	5034319.347926460	1.8
33	4	6	256323.512120709	5034332.311882570	3.6
32	4	7	256343.624034497	5034535.861678260	5.4
44	4	8	259051.864391000	5033215.737003990	**18.75**
42	5	1	255720.399256876	5032783.322989710	1.5
41	5	2	255560.651586477	5032609.286239500	2.25
5	5	3	255673.673792252	5032455.027433400	0.75
15	5	4	255706.712233325	5032155.343681460	0.75
43	5	5	255759.693089803	5032482.283847830	1.5
39	5	6	256138.197891210	5033114.632764460	0.75
38	5	7	256173.487694673	5033105.142617960	3.6
44	5	8	259051.864391000	5033215.737003990	0
37	5	9	256319.746563686	5032990.427849010	7.2
40	5	10	256244.534849801	5032959.697884480	0.75
42	5	11	255720.399256876	5032783.322989710	**19.05**

Table I.7: Routes description from Three-phase Algorithm with 50 iterations ($N = 36$, level 1).

id	n^o of single route	Order in single route	Coordinate x	Coordinate y	Demand (m^3)
4	1	1	255898.945508198	5032336.245989560	0.75
2	1	2	255935.337870106	5032357.086090610	1.5
8	1	3	256409.377579713	5033131.223553460	3.6
44	1	4	259051.864391000	5033215.737003990	0
31	1	5	256409.805148712	5034429.046381020	1.8
30	1	6	256310.541395512	5034569.743184420	1.8
25	1	7	255990.595664290	5034779.640914700	3.6
27	1	8	255864.117118510	5034882.906177090	1.8
1	1	9	255628.583800869	5032726.808474400	0.75
6	1	10	255533.522407401	5032538.410363290	1.5
29	1	11	255478.382061578	5032155.201535760	0.75
28	1	12	255462.669529721	5032119.986247760	0.75
3	1	13	255558.702458421	5031848.005041680	0.75
4	1	14	255898.945508198	5032336.245989560	**19.35**
24	2	1	256065.239606521	5034480.234596910	3.6
22	2	2	256023.962045764	5034400.838618200	3.6
21	2	3	256040.288863683	5034326.884370310	3.6
20	2	4	255948.069021439	5034250.244479680	3.6
44	2	5	259051.864391000	5033215.737003990	0
23	2	6	256128.482257811	5034677.493612770	3.6
26	2	7	256061.557482728	5034727.598430270	1.8
24	2	8	256065.239606521	5034480.234596910	**19.8**
14	3	1	255630.854341550	5032038.684513050	0.75
19	3	2	255400.936652359	5031908.097526770	0.75
18	3	3	255446.949282576	5032080.543377010	0.75
10	3	4	255881.750174129	5033240.180559440	0.75
11	3	5	255785.826530947	5033269.238715940	2.25
12	3	6	255749.912334742	5033357.616949720	0.75
13	3	7	255873.196458077	5033619.410464310	3.6
17	3	8	256244.288081726	5033178.380723640	5.4
9	3	9	256366.352106576	5033148.625915130	3.6
44	3	10	259051.864391000	5033215.737003990	0
16	3	11	255675.372616814	5031927.586545350	0.75
14	3	12	255630.854341550	5032038.684513050	**19.35**
44	4	1	259051.864391000	5033215.737003990	0
7	4	2	256104.299103770	5033289.230643370	3.6
35	4	3	256069.804336837	5033267.764357700	0.75
36	4	4	256090.544432397	5034200.434097260	3.6
34	4	5	256250.376014123	5034319.347926460	1.8
33	4	6	256323.512120709	5034332.311882570	3.6
32	4	7	256343.624034497	5034535.861678260	5.4
44	4	8	259051.864391000	5033215.737003990	**18.75**
42	5	1	255720.399256876	5032783.322989710	1.5
41	5	2	255560.651586477	5032609.286239500	2.25
5	5	3	255673.673792252	5032455.027433400	0.75
15	5	4	255706.712233325	5032155.343681460	0.75
43	5	5	255759.693089803	5032482.283847830	1.5
39	5	6	256138.197891210	5033114.632764460	0.75
38	5	7	256173.487694673	5033105.142617960	3.6
44	5	8	259051.864391000	5033215.737003990	0
37	5	9	256319.746563686	5032990.427849010	7.2
40	5	10	256244.534849801	5032959.697884480	0.75
42	5	11	255720.399256876	5032783.322989710	**19.05**

Table I.8: Routes description from Three-phase Algorithm with 100 iterations ($N = 36$, level 1).

id	n^o of single route	Order in single route	Coordinate x	Coordinate y	Demand (m^3)
41	1	1	255560.651586477	5032609.286239500	2.25
5	1	2	255673.673792252	5032455.027433400	0.75
2	1	3	255935.337870106	5032357.086090610	1.5
44	1	4	259051.864391000	5033215.737003990	0
9	1	5	256366.352106576	5033148.625915130	3.6
17	1	6	256244.288081726	5033178.380723640	5.4
7	1	7	256104.299103770	5033289.230643370	3.6
35	1	8	256069.804336837	5033267.764357700	0.75
1	1	9	255628.583800869	5032726.808474400	0.75
41	1	10	255560.651586477	5032609.286239500	**18.6**
25	2	1	255990.595664290	5034779.640914700	3.6
27	2	2	255864.117118510	5034882.906177090	1.8
24	2	3	256065.239606521	5034480.234596910	3.6
21	2	4	256040.288863683	5034326.884370310	3.6
44	2	5	259051.864391000	5033215.737003990	0
30	2	6	256310.541395512	5034569.743184420	1.8
23	2	7	256128.482257811	5034677.493612770	3.6
26	2	8	256061.557482728	5034727.598430270	1.8
25	2	9	255990.595664290	5034779.640914700	**19.8**
32	3	1	256343.624034497	5034535.861678260	5.4
31	3	2	256409.805148712	5034429.046381020	1.8
44	3	3	259051.864391000	5033215.737003990	0
33	3	4	256323.512120709	5034332.311882570	3.6
34	3	5	256250.376014123	5034319.347926460	1.8
20	3	6	255948.069021439	5034250.244479680	3.6
22	3	7	256023.962045764	5034400.838618200	3.6
32	3	8	256343.624034497	5034535.861678260	**19.8**
18	4	1	255446.949282576	5032080.543377010	0.75
28	4	2	255462.669529721	5032119.986247760	0.75
15	4	3	255706.712233325	5032155.343681460	0.75
44	4	4	259051.864391000	5033215.737003990	0
36	4	5	256090.544432397	5034200.434097260	3.6
13	4	6	255873.196458077	5033619.410464310	3.6
12	4	7	255749.912334742	5033357.616949720	0.75
11	4	8	255785.826530947	5033269.238715940	2.25
10	4	9	255881.750174129	5033240.180559440	0.75
6	4	10	255533.522407401	5032538.410363290	1.5
29	4	11	255478.382061578	5032155.201535760	0.75
14	4	12	255630.854341550	5032038.684513050	0.75
16	4	13	255675.372616814	5031927.586545350	0.75
3	4	14	255558.702458421	5031848.005041680	0.75
19	4	15	255400.936652359	5031908.097526770	0.75
18	4	16	255446.949282576	5032080.543377010	**18.45**
44	5	1	259051.864391000	5033215.737003990	0
37	5	2	256319.746563686	5032990.427849010	7.2
40	5	3	256244.534849801	5032959.697884480	0.75
4	5	4	255898.945508198	5032336.245989560	0.75
43	5	5	255759.693089803	5032482.283847830	1.5
42	5	6	255720.399256876	5032783.322989710	1.5
39	5	7	256138.197891210	5033114.632764460	0.75
38	5	8	256173.487694673	5033105.142617960	3.6
8	5	9	256409.377579713	5033131.223553460	3.6
44	5	10	259051.864391000	5033215.737003990	**19.65**

Table I.9: Routes description from Three-phase Algorithm with 200 iterations ($N = 36$, level 1).

id	n^o of single route	Order in single route	Coordinate x	Coordinate y	Demand (m^3)
9	1	1	256366.352106576	5033148.625915130	3.6
7	1	2	256104.299103770	5033289.230643370	3.6
38	1	3	256173.487694673	5033105.142617960	3.6
37	1	4	256319.746563686	5032990.427849010	7.2
44	1	5	259051.864391000	5033215.737003990	0
9	1	6	256366.352106576	5033148.625915130	**18**
25	2	1	255990.595664290	5034779.640914700	3.6
23	2	2	256128.482257811	5034677.493612770	3.6
44	2	3	259051.864391000	5033215.737003990	0
21	2	4	256040.288863683	5034326.884370310	3.6
22	2	5	256023.962045764	5034400.838618200	3.6
24	2	6	256065.239606521	5034480.234596910	3.6
27	2	7	255864.117118510	5034882.906177090	1.8
25	2	8	255990.595664290	5034779.640914700	**19.8**
29	3	1	255478.382061578	5032155.201535760	0.75
28	3	2	255462.669529721	5032119.986247760	0.75
15	3	3	255706.712233325	5032155.343681460	0.75
44	3	4	259051.864391000	5033215.737003990	0
30	3	5	256310.541395512	5034569.743184420	1.8
26	3	6	256061.557482728	5034727.598430270	1.8
13	3	7	255873.196458077	5033619.410464310	3.6
17	3	8	256244.288081726	5033178.380723640	5.4
40	3	9	256244.534849801	5032959.697884480	0.75
14	3	10	255630.854341550	5032038.684513050	0.75
16	3	11	255675.372616814	5031927.586545350	0.75
3	3	12	255558.702458421	5031848.005041680	0.75
19	3	13	255400.936652359	5031908.097526770	0.75
18	3	14	255446.949282576	5032080.543377010	0.75
29	3	15	255478.382061578	5032155.201535760	**19.35**
34	4	1	256250.376014123	5034319.347926460	1.8
33	4	2	256323.512120709	5034332.311882570	3.6
32	4	3	256343.624034497	5034535.861678260	5.4
31	4	4	256409.805148712	5034429.046381020	1.8
44	4	5	259051.864391000	5033215.737003990	0
36	4	6	256090.544432397	5034200.434097260	3.6
20	4	7	255948.069021439	5034250.244479680	3.6
34	4	8	256250.376014123	5034319.347926460	**19.8**
43	5	1	255759.693089803	5032482.283847830	1.5
5	5	2	255673.673792252	5032455.027433400	0.75
6	5	3	255533.522407401	5032538.410363290	1.5
41	5	4	255560.651586477	5032609.286239500	2.25
1	5	5	255628.583800869	5032726.808474400	0.75
42	5	6	255720.399256876	5032783.322989710	1.5
10	5	7	255881.750174129	5033240.180559440	0.75
11	5	8	255785.826530947	5033269.238715940	2.25
12	5	9	255749.912334742	5033357.616949720	0.75
35	5	10	256069.804336837	5033267.764357700	0.75
39	5	11	256138.197891210	5033114.632764460	0.75
8	5	12	256409.377579713	5033131.223553460	3.6
44	5	13	259051.864391000	5033215.737003990	0
2	5	14	255935.337870106	5032357.086090610	1.5
4	5	15	255898.945508198	5032336.245989560	0.75
43	5	16	255759.693089803	5032482.283847830	**19.35**

Table I.10: Routes description from Three-phase Algorithm with 500 iterations ($N = 36$, level 1).

id	n^o of single route	Order in single route	Coordinate x	Coordinate y	Demand (m^3)
40	1	1	256244.534849801	5032959.697884480	0.75
37	1	2	256319.746563686	5032990.427849010	7.2
44	1	3	259051.864391000	5033215.737003990	0
2	1	4	255935.337870106	5032357.086090610	1.5
43	1	5	255759.693089803	5032482.283847830	1.5
6	1	6	255533.522407401	5032538.410363290	1.5
41	1	7	255560.651586477	5032609.286239500	2.25
1	1	8	255628.583800869	5032726.808474400	0.75
39	1	9	256138.197891210	5033114.632764460	0.75
38	1	10	256173.487694673	5033105.142617960	3.6
40	1	11	256244.534849801	5032959.697884480	**19.8**
24	2	1	256065.239606521	5034480.234596910	3.6
30	2	2	256310.541395512	5034569.743184420	1.8
32	2	3	256343.624034497	5034535.861678260	5.4
44	2	4	259051.864391000	5033215.737003990	0
17	2	5	256244.288081726	5033178.380723640	5.4
35	2	6	256069.804336837	5033267.764357700	0.75
24	2	7	256065.239606521	5034480.234596910	**16.95**
33	3	1	256323.512120709	5034332.311882570	3.6
21	3	2	256040.288863683	5034326.884370310	3.6
20	3	3	255948.069021439	5034250.244479680	3.6
13	3	4	255873.196458077	5033619.410464310	3.6
7	3	5	256104.299103770	5033289.230643370	3.6
44	3	6	259051.864391000	5033215.737003990	0
31	3	7	256409.805148712	5034429.046381020	1.8
33	3	8	256323.512120709	5034332.311882570	**19.8**
11	4	1	255785.826530947	5033269.238715940	2.25
12	4	2	255749.912334742	5033357.616949720	0.75
10	4	3	255881.750174129	5033240.180559440	0.75
9	4	4	256366.352106576	5033148.625915130	3.6
8	4	5	256409.377579713	5033131.223553460	3.6
44	4	6	259051.864391000	5033215.737003990	0
4	4	7	255898.945508198	5032336.245989560	0.75
15	4	8	255706.712233325	5032155.343681460	0.75
14	4	9	255630.854341550	5032038.684513050	0.75
16	4	10	255675.372616814	5031927.586545350	0.75
3	4	11	255558.702458421	5031848.005041680	0.75
19	4	12	255400.936652359	5031908.097526770	0.75
28	4	13	255462.669529721	5032119.986247760	0.75
18	4	14	255446.949282576	5032080.543377010	0.75
29	4	15	255478.382061578	5032155.201535760	0.75
5	4	16	255673.673792252	5032455.027433400	0.75
42	4	17	255720.399256876	5032783.322989710	1.5
11	4	18	255785.826530947	5033269.238715940	**19.95**
23	5	1	256128.482257811	5034677.493612770	3.6
26	5	2	256061.557482728	5034727.598430270	1.8
25	5	3	255990.595664290	5034779.640914700	3.6
27	5	4	255864.117118510	5034882.906177090	1.8
22	5	5	256023.962045764	5034400.838618200	3.6
36	5	6	256090.544432397	5034200.434097260	3.6
34	5	7	256250.376014123	5034319.347926460	1.8
44	5	8	259051.864391000	5033215.737003990	0
23	5	9	256128.482257811	5034677.493612770	**19.8**

Annex II

Scripts of algorithms

II.1 Nearest Neighbor Algorithm

```
function [varargout]=NIAlgVRP(Problem,Capacity,xy00,xy00UTM)
%Formulating distance matrix===============================================
[aaaa,bbbb]=size(Problem)
Problem(aaaa+1,:)=[aaaa+1 xy00(1,1) xy00(1,2) 0];
xy=[Problem(:,2) Problem(:,3)];
[n m]=size (Problem);
a = meshgrid(1:n);
d = reshape(sqrt(sum((xy(a,:)-xy(a',:)).^2,2)),n,n);
%==========================================================================
%Nearest Neighbour Algorithm Details
%Step 1 : Begin With [1  1], 1 is Depo
%Step 2 :Examine all outlet that not yet served, there will be feasible outlets,
    %and infeasible outlets
%Step 3 :For feasible outlets, choose the best outlet i.e the outlet that has
    %minimum distance, insert it to the route and position it after the last outlet in
that route,
    %for example if outlet no. 6 is the best, then [1 6 1]
%Step 4 :Repeat Step 2 and 3 until there is no more feasible outlet
%Step 5 :If there is no more feasible outlet, then Create New Route [1 1] and Repeat
Step 1 to Step 4
%Step 6 :Do All the steps until all outlets are served
 Demand=Problem(:,4);
%==========================================================================
if Capacity<max(Demand)
    fprintf('The Capacity of The Vehicle Is Not Feasible\n')
    return
end
%Nearest Insertion Algorithm===============================================
RouteNow=[aaaa+1 aaaa+1]; %This is the beginning of Route Construction%%%%%%%%%
UnservedOutlet=1:length(d);
Routes={[]};
while   length(UnservedOutlet)>0
        FeasibleOutletToInsert=[]; %set consists of outlets that can be inserted to
existing route
        Distance=[];
        for NumberOfOutletNotYetInserted=1:length(UnservedOutlet);
            CandidateToInsert=UnservedOutlet(1,NumberOfOutletNotYetInserted);
            Route=[RouteNow(1:end-1) CandidateToInsert aaaa+1];%%%%%%%%%%%%%%
            [NumberOfVehicles, TotalDistance, RouteSets]=ConstEvalVRP(Problem,Route,
Capacity); %VRP Constrain Examination
            if  NumberOfVehicles>1
                FeasibleOutletToInsert=FeasibleOutletToInsert;
                Distance=Distance;
            else
%Sets of feasible outlet to insert
                FeasibleOutletToInsert=[FeasibleOutletToInsert CandidateToInsert];
                Distance=[Distance TotalDistance];
            end
%Information about distances after we examine all feasible outlet
        end
        %Choose an outlet that has minimum distance to insert
        if length(FeasibleOutletToInsert)>=1;
           [ShortestDistance idx]=min(Distance);
           best_FeasibleOutletToInsert=FeasibleOutletToInsert(idx);
           g=find(UnservedOutlet(:)==best_FeasibleOutletToInsert);
```

```
            UnservedOutlet(g)=[];
            RouteNow=[RouteNow(1:end-1) best_FeasibleOutletToInsert↙
aaaa+1];%%%%%%%%%%%%%
        end
%If there is no more feasible outlet, then Create New Route [1 1]
        if  length(FeasibleOutletToInsert)==0 | length(UnservedOutlet)==0;
             RouteFixed=RouteNow;
             Routes{end+1}=[RouteFixed];
             RouteNow=[aaaa+1 aaaa+1];%%%%%%%%%%%
        end
end
NumberOfRoutes=length(Routes);
SetsOfNumberOfOutletsInRoute=[];
for RoutesIndex=1:NumberOfRoutes
    NumberOfOutletsInRoute=length(Routes{1,RoutesIndex});
    SetsOfNumberOfOutletsInRoute=[SetsOfNumberOfOutletsInRoute NumberOfOutletsInRoute];
end
ttt=find(SetsOfNumberOfOutletsInRoute(:)<=2);
Routes(ttt)=[];
N=length(Routes);
for RouteIndex=1:N
    Route_VRP=Routes{1,RouteIndex}
    r=length(Route_VRP);
    jum=0;
    for t=1:r-1,
        subrute=jum+(d(Route_VRP(t),Route_VRP(t+1)));
        jum=subrute;
    end
    DistanceSets(RouteIndex,:)=[jum];
    TotalDistance=sum(DistanceSets);
end
AA=[];
q=1;
vectorStart=[];
Num1=[];
for X=1:N
    a=length(Routes{1,X});
    vectors(1,1:a)=q;
    Num=[1:a];
    Number=[Num1,Num];
    V(1,1:length(Routes{1,X}))=X;
    C=[AA(1:end),Routes{1,X}(1:end)];
    CC=[vectorStart vectors];
    vectorStart=CC;
    AA=C(1,:);
    q=q+1;
    vectors=[];
    Num1=Number;
end
NRoute=[C;CC;Number];
for Q=1:length(C)
    OrderedXY(:,Q)=[Problem(NRoute(1,Q),2)+xy00UTM(1,1);Problem(NRoute(1,Q),3)+xy00UTM↙
(1,2)];
end
%
```

```
NRoutes=[NRoute(1,:);NRoute(2,:);NRoute(3,:);OrderedXY(1,:);OrderedXY(2,:)];
NRoutes=NRoutes';
%dlmwrite('RoutesNNA.txt',[NRoutes],'delimiter', '\t','precision',10);
fnam='RoutesNNA.txt';
hdr={'id','numRoute','Order','x','y'};
ID=NRoutes;
txt=sprintf('%s\t',hdr{:});
txt(end)='';
dlmwrite(fnam,txt,'');
dlmwrite(fnam,ID,'-append','delimiter','\t','precision',25);
NumberOfRoutes=N;
%
TotalDistance=TotalDistance;
img = imread ('map.jpg'); %<==File name of your map
min_x = min(abs(Problem(:,2)));
max_x= max(abs(Problem(:,2)))+500;
%min_y = (min(Problem(:,3)))-500;
min_y = -2000;
%max_y = (max(Problem(:,3)))+500;
max_y = 2000;
x=abs(Problem(:,2));
y=Problem(:,3);
figure('Name','VRP-NNA|Routes','Numbertitle','off');
x_min = min_x;
x_max = max_x;
y_min = min_y;
y_max = max_y;
imagesc ([x_min x_max ], [y_min y_max], img);
set(gca,'YDir','normal','FontSize',18)
scatter(xy00(1,1),xy00(1,2),50,'k','filled','o');
%Colouring Line
for tyt=1:N
    hold on
    shortestPath=Routes{1,tyt};
    colour=mod(tyt,8);
    xd=[x(shortestPath)];
    yd=[y(shortestPath)];
%     for i=2:length(shortestPath)-1
%             text(xd(i),yd(i),['  Outlet ',num2str(shortestPath(i))]);
%     end
    %text(xy(aaaa+1,1),xy(aaaa+1,2),↙
['Depo'],'HorizontalAlignment','right','VerticalAlignment','bottom');%%%%%%%%%%
    if colour==1
    plot(xd,yd,'-go','LineWidth',1.45,'MarkerEdgeColor','k',...
                'MarkerFaceColor','g',...
                'MarkerSize',7.3)
    hold on
    elseif colour==2
    plot(xd,yd,'-ro','LineWidth',1.45,'MarkerEdgeColor','k',...
                'MarkerFaceColor','r',...
                'MarkerSize',7.3)
    hold on
    elseif colour==3
    plot(xd,yd,'-yo','LineWidth',1.45,'MarkerEdgeColor','k',...
                'MarkerFaceColor','y',...
```

```
                'MarkerSize',7.3)
    hold on
    elseif colour==4
    plot(xd,yd,'-ko','LineWidth',1.45,'MarkerEdgeColor','k',...
                'MarkerFaceColor','w',...
                'MarkerSize',7.3)
    hold on
    elseif colour==5
    plot(xd,yd,'-bo','LineWidth',1.45,'MarkerEdgeColor','k',...
                'MarkerFaceColor','b',...
                'MarkerSize',7.3)
    hold on
    elseif colour==6
    plot(xd,yd,'-mo','LineWidth',1.45,'MarkerEdgeColor','k',...
                'MarkerFaceColor','m',...
                'MarkerSize',7.3)
    hold on
    elseif colour==7
    plot(xd,yd,'-co','LineWidth',1.45,'MarkerEdgeColor','k',...
                'MarkerFaceColor','c',...
                'MarkerSize',7.3)
    hold on
    elseif colour==8
    plot(xd,yd,'-ko','LineWidth',1.45,'MarkerEdgeColor','k',...
                'MarkerFaceColor','k',...
                'MarkerSize',7.3)
    hold on
    end
   if nargout
        resultStructNNA = struct( ...
            'Routes',          Routes, ...
            'NumberOfRoutes',  NumberOfRoutes, ...
            'NRoutes',         NRoutes, ...
            'TotalDistance',   TotalDistance);

        varargout = {resultStructNNA};
   end
end
title(['Total Distance (m) = ',num2str(TotalDistance),'  | Number of Routes = ',num2str↙
(NumberOfRoutes)],'FontSize',20);
xlabel('Distance (m)','FontSize',20);
ylabel('Distance (m)','FontSize',20);
legend('Deposit','Route 1','Route 2','Route 3','Route 4','Route 5','Route 6', ...
    'Route 7','Route 8','Route 9','Route 10','Route 11','Location','northeast');
scatter(xy00(1,1),xy00(1,2),50,'k','filled','o');
```

```
function [NumberOfVehicles, TotalDistance, RouteSets]=ConstEvalVRP(Problem,Route,↙
Capacity);
%Formulating distance matrix=============================================
xy=[Problem(:,2) Problem(:,3)];
[n m]=size (Problem);
a = meshgrid(1:n);
d = reshape(sqrt(sum((xy(a,:)-xy(a',:)).^2,2)),n,n);
%========================================================================
Demand=Problem(:,4);
%========================================================================
[aa,bb]=size(Route);
for hh=1:aa
    ee=1;
    cc=2;
    while   hh<=aa && cc<bb
            ff=2;
            Routes{hh,ee}(1)=1;
            Load(hh,ee)=0;
            while   Load(hh,ee)<=Capacity && cc<bb
                if  (Load(hh,ee)+Demand(Route(hh,cc)))>Capacity
                    break
                end
                Routes{hh,ee}(ff)=Route(hh,cc);
                Load(hh,ee)=Load(hh,ee)+Demand(Route(hh,cc));
                cc=cc+1;
                ff=ff+1;
            end
            Routes{hh,ee}(ff)=1;
            ee=ee+1;
    end
    hh=hh+1;
end
%========================================================================
NumberOfVehicles=length(Routes);
RouteSets=Routes;
n=length(Routes);
for RouteNumber=1:n
    Route_VRP=Routes{1,RouteNumber};
    r=length(Route_VRP);
    jum=0;
    for t=1:r-1,
        subrute=jum+(d(Route_VRP(t),Route_VRP(t+1)));
        jum=subrute;
    end
    DistanceSets(RouteNumber,:)=[jum];
    TotalDistance=sum(DistanceSets);
end
TotalDistance=TotalDistance;
```

II.2 Three-phase Algorithm

```
function [varargout]=GaForVRP(Problem,Capacity,NumOfIterations,xy00,xy00UTM)
[aaaa,bbbb]=size(Problem)
x_y=[Problem(:,2) Problem(:,3)];
Points=[Problem(:,1) Problem(:,4)];
[nn mm]=size(Points);
TotalMin=Inf;
Demand=Points(:,2)
Demand(end+1,1)=0

% Iterations beginning
for iter=1:NumOfIterations

% Creation of the population
    for f=1:100% Population size %%%%%%%%%%%%
        if iter==1% Not yet improved by the GA operations application
            if f==1% First element of the population
                IndexDemand=[Problem(:,1) Problem(:,4)];
                NewNewIndex=Problem(:,1);
                NewNewIndex=NewNewIndex';
            else
                NewNewIndex=randperm(nn);
                NewNewIndex=NewNewIndex';
                for cc=1:nn
                    IndexDemand(cc,:)=[NewNewIndex(cc) Problem(NewNewIndex(cc),4)];
                end
            end
            NewIndex(f,:)=NewNewIndex';
        else
            NewNewIndex=TotalIndex;
            NewNewIndex=NewNewIndex(f,:);
            NewNewIndex=NewNewIndex';
            for cc=1:nn
                IndexDemand(cc,:)=[NewNewIndex(cc) Problem(NewNewIndex(cc),4)];
            end
            NewIndex=TotalIndex;
        end
        VerifyCluster=[]%
        NewCluster=[]
        [NewCluster]=RouteSelection(Problem,Capacity,IndexDemand,Demand);% Selection of↙
the clusters
        VerifyCluster=NewCluster
        [k,mm]=size(NewCluster);
        for e=1:k
            RowCluster=VerifyCluster(e,1:end);
            s=isempty(RowCluster{1,end});
            if s==1
                RowCluster=RowCluster(1,1:end-1);
                abc=e
            end
            [ss,kk]=size(RowCluster);
            for ee=1:kk
                SingleCluster=RowCluster{1,ee};
                SingleCluster=SingleCluster(1,2:end);
                IndexCluster=[1:length(SingleCluster)];
                IndexAndSingleCluster=[IndexCluster' SingleCluster'];
```

```
                %verify
                totalCapacity3=sum(Demand(SingleCluster(2:end),1))
                if totalCapacity3>Capacity
                    fprintf('seventh');
                    return
                end

            xyz=[];
            xyz(1,:)=xy00;
            x=length(SingleCluster);
            for vv=2:x
                xyz(vv,:)=x_y(SingleCluster(1,vv),:);
            end
            [optRoute,minDist]= vrp_ga(xyz,SingleCluster);
            RowDist(ee,1)=minDist;
            realRoute=[];
            for y=1:length(optRoute)
                realRoute(1,y)=SingleCluster(optRoute(y));
            end
            SortedRealRoute=sort(realRoute)
            SortedRouteOfnn{e,ee}=SortedRealRoute
            RouteOfnn{e,ee}=realRoute;
            optRoute=[];
            SingleCluster={[]};
        end
        RowSumDist(e,1)=sum(RowDist);
        [minRowDist,index]=min(RowSumDist);
        distHistory(iter) = minRowDist;
        if minRowDist < TotalMin% Selection of the best route
            TotalMin=minRowDist;
            TotalRoute=[]
            TotalRoute=RouteOfnn(index,:);
            minIter=iter;
        end
        RowCluster={[]}
        RowDist=[]%
    end
    bestRowDist(f,1)=min(minRowDist);
    RouteOfnn=[]%
    RowSumDist=[]%
end

% GA application to improve the cluster selection
% Genetic Algorithm Operators
for t=1:25%%%%%%%%%%%%
    [best4RowDist,index]=min(bestRowDist);
    best4RowDist(t,1)=best4RowDist;
    LastIndex(t,:)=NewIndex(index,:);
    routeInsertionPoints = sort(ceil(nn*rand(1,2)));
    I = routeInsertionPoints(1);
    J = routeInsertionPoints(2);
    for r = 1:4 % Mutate the population with GA operations
        RowIndex(r,:) = LastIndex(t,:);
        switch r
```

```
                case 2 % Inversion
                    RowIndex(r,I:J) = RowIndex(r,J:-1:I);
                case 3 % Mutation
                    RowIndex(r,[I J]) = RowIndex(r,[J I]);
                case 4 % Slide
                    RowIndex(r,I:J) = RowIndex(r,[I+1:J I]);
                otherwise % Do nothing
            end
        end
        first=t*4-3;
        second=t*4;
        TotalIndex(first:second,:)=RowIndex;
    end

end
ss=isempty(TotalRoute{1,end});
if ss==1
   TotalRoute=TotalRoute(1,1:end-1);
end
% Plot of the results
[nnn,mmm]=size(TotalRoute);
NumberOfRoutes=mmm;
TotalDistance=TotalMin;
img = imread ('map.jpg'); %<==File name of your map
min_x = min(abs(Problem(:,2)));
max_x= max(abs(Problem(:,2)))+500;
%min_y = (min(Problem(:,3)))-500;
min_y = -2000;
%max_y = (max(Problem(:,3)))+500;
max_y = 2000;
for z=1:mmm
    xxx=find(TotalRoute{1,z}==0)
    TotalRoute{1,z}(xxx)=aaaa+1%%%%%%%%%%%%

    %verify capacity
    totalCapacity7=sum(Demand(TotalRoute{1,z}(2:end),1))
    if totalCapacity7>Capacity
       fprintf('sixth');
       return
    end
end
for ab=1:mmm
    first_point=TotalRoute{1,ab}(1)
    TotalRoute{1,ab}(end+1)=first_point
end
Problem(aaaa+1,:)=[aaaa+1 xy00(1,1) xy00(1,2) 0];%%%%%%%%
x=abs(Problem(:,2));
y=Problem(:,3);

% Plote of the routes
figure('Name','VRP-GA|Routes','Numbertitle','off');
x_min = min_x;
x_max = max_x;
y_min = min_y;
y_max = max_y;
```

```
imagesc ([x_min x_max ], [y_min y_max],img);
set(gca,'YDir','normal','FontSize',18)
scatter(xy00(1,1),xy00(1,2),50,'k','filled','o');
for tyt=1:mmm
    hold on
    shortestPath=TotalRoute{1,tyt};
    colour=mod(tyt,8);
    xd=[x(shortestPath)];
    yd=[y(shortestPath)];
    if colour==1
    plot(xd,yd,'-go','LineWidth',1.45,'MarkerEdgeColor','k',...
                'MarkerFaceColor','g',...
                'MarkerSize',7.3)
    hold on
    elseif colour==2
    plot(xd,yd,'-ro','LineWidth',1.45,'MarkerEdgeColor','k',...
                'MarkerFaceColor','r',...
                'MarkerSize',7.3)
    hold on
    elseif colour==3
    plot(xd,yd,'-yo','LineWidth',1.45,'MarkerEdgeColor','k',...
                'MarkerFaceColor','y',...
                'MarkerSize',7.3)
    hold on
    elseif colour==4
    plot(xd,yd,'-ko','LineWidth',1.45,'MarkerEdgeColor','k',...
                'MarkerFaceColor','w',...
                'MarkerSize',7.3)
    hold on
    elseif colour==5
    plot(xd,yd,'-bo','LineWidth',1.45,'MarkerEdgeColor','k',...
                'MarkerFaceColor','b',...
                'MarkerSize',7.3)
    hold on
    elseif colour==6
    plot(xd,yd,'-mo','LineWidth',1.45,'MarkerEdgeColor','k',...
                'MarkerFaceColor','m',...
                'MarkerSize',7.3)
    hold on
    elseif colour==7
    plot(xd,yd,'-co','LineWidth',1.45,'MarkerEdgeColor','k',...
                'MarkerFaceColor','c',...
                'MarkerSize',7.3)
    hold on
    elseif colour==8
    plot(xd,yd,'-ko','LineWidth',1.45,'MarkerEdgeColor','k',...
                'MarkerFaceColor','k',...
                'MarkerSize',7.3)
    hold on
    end
end
title(['Total Distance (m) = ',num2str(TotalMin),'  | Number of Routes = ',num2str↙
(mmm)],'FontSize',20);
xlabel('Distance (m)','FontSize',20)
ylabel('Distance (m)','FontSize',20)
```

```
legend('Deposit','Route 1','Route 2','Route 3','Route 4','Route 5','Route 6', ...
    'Route 7','Route 8','Route 9','Route 10','Route 11','Location','northeast');
scatter(xy00(1,1),xy00(1,2),50,'k','filled','o');

% Results extraction
AA=[];
q=1;
vectorStart=[];
Num1=[];
for X=1:length(TotalRoute)
    a=length(TotalRoute{1,X});
    vectors(1,1:a)=q;
    Num=[1:a];
    Number=[Num1,Num];
    V(1,1:length(TotalRoute{1,X}))=X;
    C=[AA(1:end),TotalRoute{1,X}(1:end)];
    CC=[vectorStart vectors];
    vectorStart=CC;
    AA=C(1,:);
    q=q+1;
    vectors=[];
    Num1=Number;
end
NRoute=[C;CC;Number];
for Q=1:length(C)
    OrderedXY(:,Q)=[Problem(NRoute(1,Q),2)+xy00UTM(1,1);Problem(NRoute(1,Q),3)+xy00UTM↙
(1,2)];
    OrderedDemand(1,Q)=Demand(NRoute(1,Q),1)
end
NRoutes=[NRoute(1,:);NRoute(2,:);NRoute(3,:);OrderedXY(1,:);OrderedXY(2,:);↙
OrderedDemand];
NRoutes=NRoutes';

% dlmwrite('RoutesGA.txt',[NRoutes],'delimiter', '\t','precision',10);
fnam='RoutesGA.txt';
hdr={'id','numRoute','Order','x','y','Demand'};
ID=NRoutes;
txt=sprintf('%s\t',hdr{:});
txt(end)='';
dlmwrite(fnam,txt,'');
dlmwrite(fnam,ID,'-append','delimiter','\t','precision',25);

% Plot of the evolution
N=[1:NumOfIterations]
[fittest,gof]=fit(N',distHistory',  'poly3')
figure('Name','VRP-GA|Iterations','Numbertitle','off');
plot(distHistory,'b','LineWidth',2);
hold on
plot(fittest,N',distHistory)
title('Best Solution History');
set(gca,'XLim',[0 NumOfIterations+1],'YLim',[0 1.1*max([1 distHistory])]);
xlabel('Number of iterations');
ylabel('Total distance (m)');

% Results
```

```
if nargout
        resultStructGA = struct( ...
            'TotalMin',       TotalMin, ...
            'TotalRoute',     TotalRoute, ...
            'NRoutes',        NRoutes, ...
            'minIter',        minIter);

        varargout = {resultStructGA};
end
end
```

```
function [NewCluster]=Route_Selection(Problem,Capacity,IndexDemand,Demand)
[n m]=size(IndexDemand)
Demand=IndexDemand(:,2);

% Verify the Demand required by each collection point
if Capacity<max(Demand)
    fprintf('The Capacity of The Vehicle Is Not Feasible\n');
    return
end
NewCluster=[]
Used=[];
NotYetUsed=IndexDemand(1:end,1);
Route=[];
Routes={};
[a,c]=size(Routes);
StartRoute=[0 0];
RouteNow=[0 0];
i=1;
d=1;
ff=1;
bb=1;
Load=0;
countCurrent=1;

% Start with the global cycle
while countCurrent<=n
    Route=[RouteNow(1:end-1) NotYetUsed(i) 0];
    Load=Load+Demand(NotYetUsed(i));
    if Load<=Capacity
        RouteNow=Route;
        Used=RouteNow(2:end-1);
        g=find(NotYetUsed(:)==NotYetUsed(i));
        NotYetUsed(g)=[];
        h=length(NotYetUsed);
        i=1;
        if h==0
            Routes{countCurrent,bb}=Route;
            RouteNow=StartRoute;
            Used=[];
            NotYetUsed=IndexDemand(1:end,1);
            Load=0;
            ff=ff+1;
            countCurrent=countCurrent+1;
            bb=1;
            i=countCurrent;
        end
    else
        Route=[RouteNow(1:end-1) 0];
        Routes{countCurrent,bb}=Route;
        RouteNow=StartRoute;
        Load=0;
        ff=1;
        i=1;
        bb=bb+1;
    end
```

```
end
[v,w]=size(Routes);
for hh=1:v
    for gg=1:w
Cluster{hh,gg}=sort(Routes{hh,gg});
str{hh,gg}=mat2str(Cluster{hh,gg});

%verify capacity
totalCapacity(hh,gg)=sum(Demand(Cluster{hh,gg}(3:end),1))
       if totalCapacity(hh,gg)>Capacity
          fprintf('first');
          return
       end

    end
end
NewCluster=Cluster;
aa=length(str);
for ii=1:aa
    count=ii+1;
    while count<=aa
        arraysAreEqual = isempty(setxor(str(ii,:),str(count,:)));
        if arraysAreEqual==1
            NewCluster(count,:)=[];
            str(count,:)=[];
        else
            count=count+1;
        end
        aa=length(str);
    end
end
end
```

```
% Author: Joseph Kirk
% Email: jdkirk630@gmail.com
% Release: 3.0
% Release Date: 05/01/2014

function [optRoute,minDist]= Ga_VRP(xyz,SingleCluster)
dmat= [];% Distance matrix
popSize= 4;% Population size
numIter= 50;% Number of iterations
xy=xyz
Single_Cluster=SingleCluster
n=length(Single_Cluster)
optRoute=[]
minDist=[]

% Distancce matrix construction
    if isempty(dmat)
        nPoints = size(xy,1);
        a = meshgrid(1:nPoints);
        dmat = reshape(sqrt(sum((xy(a,:)-xy(a',:)).^2,2)),nPoints,nPoints);
    end

    % Verify Inputs
    [N,dims] = size(xy);
    [nr,nc] = size(dmat);
    if N ~= nr || N ~= nc
        error('Invalid XY or DMAT inputs!')
    end
    n = N;

    % Sanity Checks
    popSize     = 4*ceil(popSize/4);
    numIter     = max(1,round(real(numIter(1))));
pop = zeros(popSize,n);
    pop(1,:) = (1:n);
    for k = 2:popSize
        pop(k,:) = randperm(n);
    end

    % Run the GA
    globalMin = Inf;
    totalDist = zeros(1,popSize);
    distHistory = zeros(1,numIter);
    tmpPop = zeros(4,n);
    newPop = zeros(popSize,n);
    for iter = 1:numIter

        % Evaluate Each Population Member (Calculate Total Distance)
        for p = 1:popSize
            d = dmat(pop(p,n),pop(p,1)); % Closed Path
            for k = 2:n
                d = d + dmat(pop(p,k-1),pop(p,k));
            end
            totalDist(p) = d;
        end
```

```
        % Find the Best Route in the Population
        [minDist,index] = min(totalDist);
        distHistory(iter) = minDist;
        if minDist < globalMin
            globalMin = minDist;
            optRoute = pop(index,:);
        end

         % Genetic Algorithm Operators
        randomOrder = randperm(popSize);
        for p = 4:4:popSize
            rtes = pop(randomOrder(p-3:p),:);
            dists = totalDist(randomOrder(p-3:p));
            [ignore,idx] = min(dists); %#ok
            bestOf4Route = rtes(idx,:);
            routeInsertionPoints = sort(ceil(n*rand(1,2)));
            I = routeInsertionPoints(1);
            J = routeInsertionPoints(2);
            for k = 1:4 % % Mutate the population with GA operations
                tmpPop(k,:) = bestOf4Route;
                switch k
                    case 2 % Inversion
                        tmpPop(k,I:J) = tmpPop(k,J:-1:I);
                    case 3 % Mutation
                        tmpPop(k,[I J]) = tmpPop(k,[J I]);
                    case 4 % Slide
                        tmpPop(k,I:J) = tmpPop(k,[I+1:J I]);
                    otherwise % Do nothing
                end
            end
            newPop(p-3:p,:) = tmpPop;
        end
        pop = newPop;
    end

    %verify the cluster
    OrderedOptRoute=sort(optRoute)
    if OrderedOptRoute==Single_Cluster
       fprintf('second');
       return
    end
end
```

Ringraziamenti

Non sono mai stato un bravo scrittore e ció non mi ha di certo aiutato nella stesura di questo elaborato, tanto meno nella sua stesura in inglese. Ma sono comunque riuscito ad ottenere dei risultati e a raggiungere delle conoscenze lontane dal mio background di partenza. Sono perció molto soddisfatto e per questo devo ringraziare il Professor Salemi, che fin dall'inizio di questo progetto ha saputo guidarmi e stimolarmi, e che è davvero stato sempre presente e disponibile.
Un ringraziamento va ai miei genitori, che hanno sempre sostenuto ogni mia libera scelta, cosí come stanno ancora facendo.
Inoltre un forte abbraccio va ai Carissimi di Padova, gli amici che per primi mi hanno accolto, e sopratutto a Valentina senza la quale tutto ció non sarebbe mai stato cosí felice.