Master degree course in Computer Engineering

Master Degree Thesis

Data-driven PSO optimization of a smart-city metro line

Supervisors
Prof. Luca Vassio
Dr. Indaco Biazzo

Candidate
Zeda Zhu

July 2021

Summary

The object of this paper is the data-driven optimization of a new public transport lines in smart-cities. This paper provides a solution to improve urban public transport line planning based on public transport data, urban street data, and population data. Two scores for computing the performance of public transport network (PTN) are used: the velocity score and sociality score. They are indicators of city transport accessibility and public transport operation capacity based on population distribution respectively. Also, a scoring model based on the principle of isochron is used to evaluate and calculate the sociality score of the city PTN. Then, the particle swarm optimization algorithm(PSO) is introduced and what kind of problems can be solved by PSO algorithm. A greedy strategy is proposed to optimize the line by changing the position on the stops in line, and a improved version of the PSO is proposed. According to the sociality score of PTN, greedy strategy is adopted, and improved PSO algorithm is used to optimize the longitude and latitude position of the stations of the new line to seek the maximum sociality score of city PTN after adding this line, so as to have the best operation ability based on population distribution. This paper optimizes the current version of the metro D line under construction in Rome. By adjusting the location of the stops of metro D in the planning, the sociality score is improved. By comparing the urban sociality score before and after optimization, our new metro D have achieved 8.0% improvement of the city sociality score. This paper confirms the role of improved PSO as an optimization algorithm in solving the line planning problem and gives a complete strategy that tends to do the line planning according to the population density distribution and PTN. The research significance of this paper is that the line planning scheme combined with scoring model and optimization algorithm can help city line planning to be data-driven to provide more schemes for engineers and planners reference, for the final line construction site selection to provide a data-driven scientific basis.
Key words: *data-driven, line planning, line optimization, particle swarm*

optimization, accessibility.

Acknowledgements

I would like to give my sincere gratitude to Prof. Luca Vassio and Dr. Indaco Biazzo, my supervisors who extraordinary patience and consistent encouragement. They gave me great help by providing me with necessary materials, advice of great value and inspiration of new ideas. I benefited a lot in the process of finishing the paper because of their support.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

1.1 Smart-city theory

In today's world, there seems to be an obvious trend to use the prefix "smart". For example, cities all over the world are branding themselves as "smart cities" or striving to become "smart cities". Planners and decision-makers support "smart growth". Infrastructure planning involves "smart grid" of energy, "smart network" of information and communication technology (ICT), and "smart mobility" of transportation. "Smart" or "smarter" may be seen as the next frontier in urban planning, decision-making, and management. A common basic theme is to apply technology to urban planning and management, so that time and resources can be more optimized, so as to improve efficiency. In the context of modern technology, "intelligence" means following the process of computer programming or guidance, involving a certain degree of intelligent autonomy or automation [1].

The smart city aspects involve information and communication technology to gather, analyze and integrate the key information of the core system of urban operation, so as to make an intelligent response to various needs including people's livelihood, environmental protection, public safety, urban services, industrial and commercial activities. Its essence is to use advanced information technology to realize the intelligent management and operation of the city, so as to create a better life for people in the city and promote the harmonious and sustainable growth of the city. Google adopted the GTFS standard file (https://developers.google.com/transit/gtfs/) to encourage public transport companies to release their data in a uniform way in order to be included in its map platform. It is nowadays possible to find hundreds of companies having released their data, and there are portals where this data is collected and exposed [2]. Using big data sets released by public transport companies, we can analyze and have a new understanding of the operation of the city. Using these, new urban planning can be carried out. Nowadays, the data generated by mobile phone applications and sensors in our daily life continue to supplement the traditional data sets. The massive and accurate data enriches our experience of the operation mode of cities.

There are many scenarios of smart cities. For example, in terms of "smart transportation", the population and area of some large cities and mega cities are often very large. A huge transportation network will be built to serve people's daily life. For example, Rome has a population of 4.3 million and an area of 1285.31 square kilometers. Cities like Rome have a high demand for urban transportation. Rome has currently over 350 bus lines, three metro lines over 37 miles (60 km) and a new metro line D under planning, also there are several regional railways complement the three metro lines in Roma (see figure 1.1). An efficient and reasonable transportation network will bring great traffic benefits, which is very important for the citizens and the municipal government.

1.2 Optimization of public transportation

Public transport plays an important role in metropolitan areas. In the suburban environment, the planning and operation of public transport is particularly important because of the low population density and the distance to the city center. An excellent public transport planning can greatly improve the income of public transport companies and the convenience of citizens, and from the perspective of environmental protection, citizens using public transport can reduce exhaust emissions and protect the air. Public transportation planning mainly includes two levels [9]: strategic level and operational level as it is reported in figure 1.2 In the strategic level, it involves the design of transportation network planning, line planning, timetabling and tariff planning, refer to the public transportation network planing, the new line planing, the public transportation timetable design, and the plan for the ticket price. In the operation stage, it includes delay-management and rescheduling, works during the operation of public transport. The optimization of public transport can start from the above transportation planning. Here we now briefly introduce strategic level's planning. **Network planning** includes the design of the transportation network. The outcome of the process of network planning is the public transportation network (PTN). However, in

Figure 1.1. The routes and stations of the regional railway and metro in Rome[3]

real life a PTN is usually not designed from scratch, but only modifications of an existing PTN are considered, such as finding new stations in a metro or bus network, closing existing stations, or finding a subnetwork for opening rapid transit lines.

Line planning is to plan the route of bus or metro line in PTN on which service should be offered. Line planning has been well studied in the literature. In [10, 11], the goal is to maximize the number of directly connected passengers under the constraint that all passengers can be transported. The advanced integer programming technique is used in the proposed method. Under similar constraints, the goal of [12] is to minimize the cost of public transport companies. The paper [13, 14] studies the route planning problem considering different vehicles simultaneously. Various models and algorithms are discussed in [24]. In addition, there are also studies on new lines. Recently, in [15, 16, 17, 18, 19, 22], it is considered to find the route plan and the

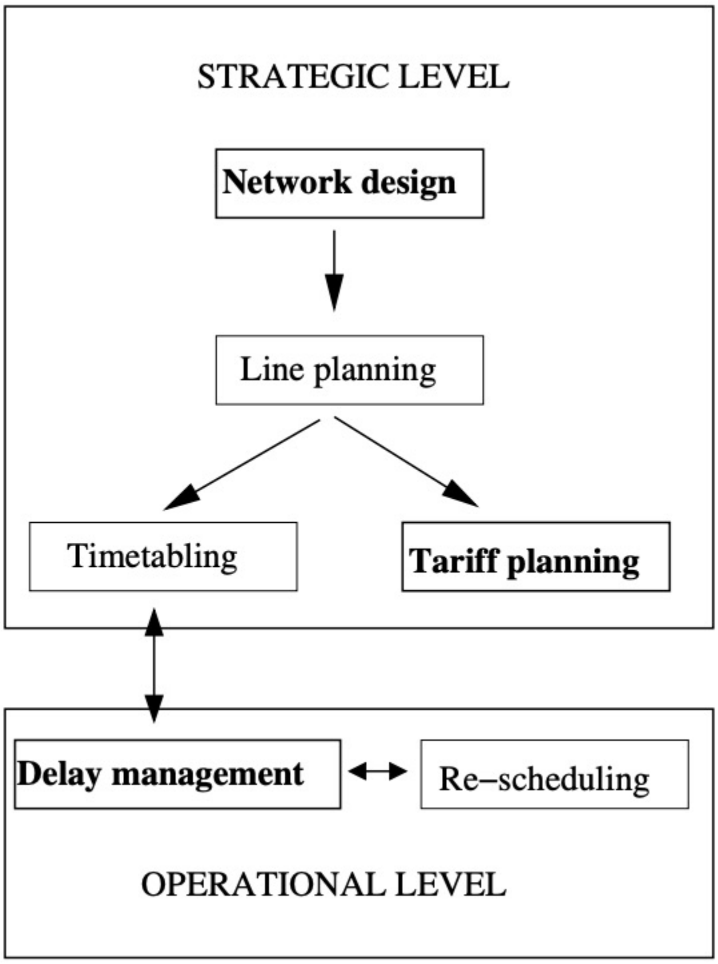


Figure 1.2. Planning process in public transportation[9] p.2

best route for customers. In these methods, the goal is to design the route in such a way that the customer's travel time is minimized. In addition, some research use the number of customers to transfer as a target, [23] deals with the special case of locating a transport line to maximize the number of passengers.

It can be seen that the optimization of public transport is more focused on line planning at the strategic level, and the optimization for line planning is closer to real life, which is conducive to solving practical problems. Therefore, according to the actual problems, the design of the optimal optimization scheme can greatly improve the reliability of public transport at the strategic level.

1.3 Choosing new public transport routes

Now some cities are actually facing the need to optimize PTN, Take Rome as an example, with a population of 4.3 million and an area of 1285.31 square kilometers. Under the pressure of population growth, according to the previous paragraph, planning the new traffic line is the most direct scheme to optimize the existing PTN. This brings new problems, such as how to choose the location of the new stations? How to maximize the benefits of the new public transport route? The quality of the public transport route includes many factors, such as the speed to the destination, the carrying capacity of urban people, and the cost of the traffic line construction. These problems will have new solutions based on big data obtained from smart city. Using the massive data and the existing PTN distribution, we can use a scoring model for new line planning. Scoring model is used to analyze the existing public transport routes and the distribution of residents, evaluate the public transport score. According to the research of Biazzo et al[21], we will be able to use public data to analyze and evaluate the major cities in world. Biazzo et al also provide the model which can analyze the public transport score of the whole city through the city data, we call it scoring model in this paper. The model could evaluate urban PTN and give two types of indicators velocity score: quantifying the overall speed of entering a specific area of the city, and sociality score: quantifies how many people you can reach from a specific area. With these two indicators we can evaluated the urban PTN and calculate the score of the whole city by adding simulation stations.The detailed introduction of the model and the concept and calculation of the public transport score will be explained in Chapter 2. By adjusting the location of the new line and comparing the relationship between the location of each station and the score of the new line, the optimal line is selected as the construction goal. This paper will focus on what kind of method and algorithms can effectively select the optimal route. In this paper we focus on social significance of public transport, quantifies how many people one person can meet in a typical daily trip. Our experiment give an option about how to choose a new route on the basis of the existing network, so that it can get the maximum **sociality score**. Our goal is to enable PTN to let the population reach as many people as possible in a specific time window. It also means that PTN can connect more people and regions.

1.4 Outline of methodology and results

The research problem of this paper is the optimization of urban public transport lines. This paper uses some experience on some data-driven optimization researches, such as optimization of charging infrastructure placement for shared vehicles [4, 5]. What kind of execution strategy and optimization algorithm to choose for line optimization is the main research content of this paper. This paper takes Rome as the experimental object. Firstly, the public transportation and urban population distribution of Rome are analyzed by using the scoring model. Then, according to the demand of new public transport lines in the city, the solution based on real population distribution and public transport network is provided for line planning. Based on the scoring model, this paper evaluates the simulation line. According to the evaluation results of the scoring model, the PSO algorithm and local optimal greedy strategy are used to optimize the simulation line to maximize the commuting efficiency of the whole city. At the end of the paper, based on the existing public transport network in Rome, we evaluate the Metro D line that Rome is planning to build based on the score model. With the location of metro D stations planned by the government, we use PSO and greedy strategy to optimize it. After optimize the metro D, the sociality score of the whole city (with new metro D) is **8.0%** higher than that of the original Metro D.

Chapter 2

Scoring model and computational improvements

The most critical points for finding the optimal solution of line planning based on known PTN are:

1. How to quantify and evaluate the PTN added to the new line?
2. What kind of algorithm can be used to select the station location of the new line?

In this chapter we will introduce a way to quantify and evaluate the PTN.

The work developed in this thesis aim at providing optimization scheme for the construction of new urban traffic lines. So, how to compare the effect of adding new lines on existing PTN? First of all, we need to evaluate and score the existing PTN, and then calculate the score of the PTN with a new line after line planning. We will see the impact of a new line on the whole city PTN. On this basis, we can adjust the new line, adjust the station position in the line, and use the scoring model to evaluate after each adjustment to get a new score. According to this score, we can judge whether the adjusted line is better than the previous one. Of course, for the sake of fair comparison, the modification of the line only includes the adjustment of the station position, not the change of the number of stations. Because lines with more stations have more coverage capacity than lines with fewer stations. According to the process of route selection, we can see that the scoring model is very important for line planning. The scoring model will be directly used as the optimization function of optimization algorithms, and the scoring will be used as the direct standard to judge the optimization results. Therefore, in this chapter, we will introduce the model used in this thesis to evaluate and score the situation of urban PTN.

2.1 Scoring model

First of all, we need define what kind of score can evaluate the city PTN. There are two kinds of scores that will be explained in this thesis: velocity score and sociality score.

1. The velocity score of a city represents the different places that urban residents can reach through public transport.

2. The sociality score of a city is the number of people that a person living in the city can potentially meet within a typical daily trip.

The two scores represent the quality of PTN without population distribution and with population distribution influence respectively.

We need scoring model to evaluate these two score of PTN. This thesis uses the urban accessibility assessment method based on public transport data proposed in [21]. This method uses the concept of isochrones as a metric for accessibility and to measure the performance of urban transportation system connecting people and places.

It is important to be able to quantify accessibility in a way that closely represents the experience of citizens. Following the general approach in accessibility research [25], the model focuses on travel time between geographical regions, which better represents the mindset adopted by citizens in planning their mobility.

The key mathematical concept used to quantify travel time in this method is isochronic maps, which showing areas related to isochrones between different points. Considering a geographical point, its isochronic map will be composed of isochronic contours, which are marked with different transportation systems to mark the areas that can be reached within a given time span. The concept of isochronic maps has existed since 1881 [26], which shows the travel time from London to all over the world. Different from 1881, at present, it is very accurate to draw isochronal map of area and transportation system based on open source data. More precise calculation and analysis can be done.

The model proposed by Biazzo et al[21] focuses on public transportation, and uses multiple routing methods to calculate travel time and isochrones. There are many modes of routing methods, for example, the best route from point a to point B in the city can be realized by a combination of many modes of transportation, such as walking, bus, metro and train. Generally, PTN is analyzed as static graph in public transport research, points and edges represent the stops and the connection between stations respectively [27, 28, 29, 30, 34]. Few studies have considered the "time" factor of these transportation systems. Model proposed by Biazzo et al pays more attention to the dynamical aspect of mobility. Its mainly produces two indicators: Velocity score, quantifying the overall speed of entering a specific area of the city. Sociality score quantifies how many people you can meet from a specific area. In the horizontal comparison between cities, the model will also reduce the dependence of social score on the total urban population. This is to reduce the impact of urban population on scores. Since our line planning is mainly for a certain city, it does not involve horizontal comparison with other cities, so this thesis will not describe it here.

In addition, because the definition of accessibility is very broad, which may refer to the ability to provide services for the disabled [35], or the ability to make ordinary people reach the workplace [36], the significance of this method is to provide a general, efficient and easily visualized on maps to make the scoring and display of urban PTN easier to use and more general. This thesis will also show the visualization of PTN score change (impact on city) after line planning in Chapter 5. In addition, the score calculation time of the model for the whole city is short (Roma, for example, takes 7 seconds on average run the code [33] on Intel(R) Xeon(R) Gold 6140 CPU @ 2.30GHz with 16 CPU Threads and 64 GB mem) The calculation of isochron is based on the multi-mode method to calculate the travel time between any pair of places in the city. As mentioned above, the method includes a variety of routing methods, taking into account the walking, bus, subway and train. In order to keep the computing time low, the model uses hexagonal subdivision of urban areas. Biazzo et al. constructed a hexagonal grid with a side length of 0.2 km. Not the whole area of a city is covered by hexagons. Hexagon covers the city areas which the locations containing at least one public service stops, that is, if there is no public service stops in the area, it will not be divided into hexagon area. Moreover, as well as all areas from public service stops, the walking path should not exceed 15 minutes, which is convenient to calculate the walking path between public service stops and the total walking time.

After dividing the city into hexagonal grid, it is necessary to calculate the walking path between the public service stops and the hexagonal grid. The model uses the open source routing machine (OSRM) [37] OSRM, which allows to use the corresponding OpenStreetMap [38]to calculate the shortest walking path in PTN. Except for the walking path, the public transport schedule data uses the data released by 100 companies incorporated by Google [2] . In this way, the model has the basic data to analyze multiple routing modes. The model also uses the urban population density data. In order to match the population density data with the hexagon region, the population density is divided into hexagon regions according to the proportion of overlapping surfaces. Data about population densities in urban areas have been gathered through the Eurostat Population Grid [31] for the European cities [40] and the Gridded Population of the world made by the Center for International Earth Science Information Network [41]. The model adopted a modified version of the connection scan algorithm (CSA) [42], that we call the intransitive connection scan algorithm (ICSA) Finally, combined with the city hexagon area division, pedestrian routing and bus information. Using ICSA algorithm, the travel time between any pair of hexagons can be calculated. And because the model combines the hexagon area with the population density data, the model also has the ability to quantify the performance of public transport in connecting people.

2.2 Computation of sociality and velocity score

The definition of city sociality and velocity score has been given above. This section introduces the calculation method of two scores. The **city scores** are the average scores we get as the final score we use to evaluate the PTN of the city.

Velocity score: The velocity score aims at giving a synthetic representation of the information encoded in all the isochronic maps computed from all the points of a city. Therefore, the isochronic map is regarded as an expansion process starting from a starting point. Consider the isochrone centered on a hexagon λ. At time t0 corresponding to the travel time τ, $I(\tau,(\lambda,t0))$. The covered area $A(\tau,(\lambda,t0))$ of the isochrone at time τ will thus be the area contained within $I(\tau,(\lambda,t0))$. By approximating the perimeter of the isochrone with a circle, the average travelled distance $\bar{r}$ taking a random direction from the starting point p0 is given by

$$\overline{r}(\tau,(\lambda,t_0)) = \sqrt{\frac{A(\tau,(\lambda,t_0))}{\pi}} \tag{2.1}$$

and dividing by the time τ we obtain a quantity that has the dimension of a speed:

$$\overline{v}(\tau,(\lambda,t_0)) = \frac{\overline{r}(\tau,(\lambda,t_0))}{\tau} \tag{2.2}$$

The interpretation of $\overline{v}(\tau,(\lambda,t0))$ is the average speed of expansion, at time τ, of a circular isochrone with the same area as the real one in figure 2.1

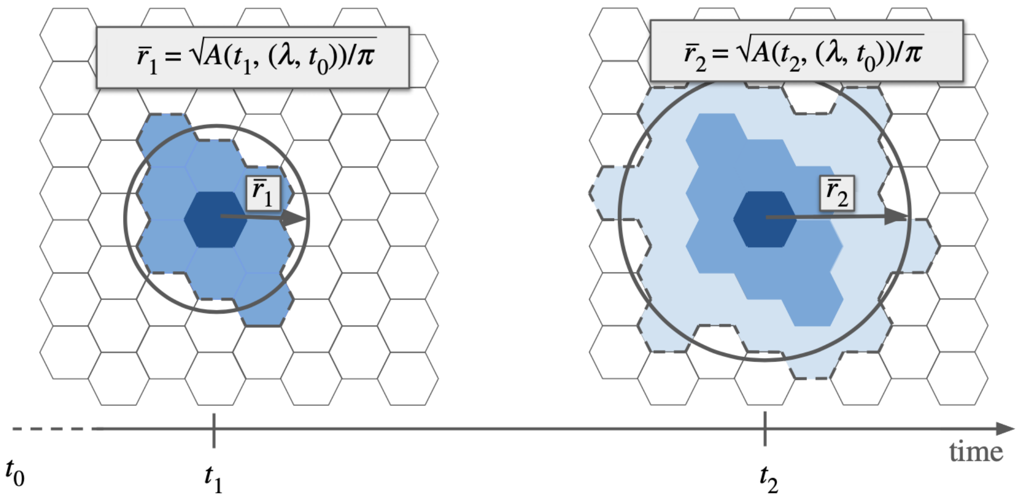


Figure 2.1. Isochrone area. Isochrones with hexagonal tessellation at different times. The circles in the figure have the same area as the area contained by the isochrones.[32]

Sociality score: Compared with the average velocity approximation a of a person leaving the hexagon in a random direction provided by the velocity score, the sociality score is used to measure the number of people that may be contacted in a trip. The Sociality score considers the distribution of the population density, because the distribution of population density has relevant demand for transport services, such as strengthening transport services in densely populated areas. the sociality score quantifies the performance of public transit in connecting people. Model define $P(\tau,(\lambda,t0))$ as the amount of population living within the isochrone $I(\tau,(\lambda,t0))$. Similarly to what we did for the velocity score, we can average $P(\tau,(\lambda,t0))$ over the travel time

τ (with the same distribution of daily budget times $f(\tau)$) and over different starting times t0, obtaining the sociality score as(t_0 is the operation time of public transport):

$$v(\lambda) = \frac{\sum_{t_0=6.00}^{22.00} \int_0^\infty v(\tau, (\lambda, t_0)) f(\tau) \mathrm{d}\tau}{\sum_{t_0=6.00}^{22.00} \int_0^\infty f(\tau) \mathrm{d}\tau} \tag{2.3}$$

City scores: For each hexagon, λ, we have both the number of people living there, $pop(\lambda)$, as well as the average velocity of their trips with public transport starting from the considered hexagon. Through the formula, the average velocity per person of the whole city can be calculated to evaluate velocity property of the whole city, which represents the average number of different places that urban residents can easily reach through public transport and called city velocity score.

$$v_{city} = \frac{\sum_{\lambda \in city} v(\lambda) * pop(\lambda)}{pop(city)} \tag{2.4}$$

The calculation of sociality score is close to that of velocity score, and the city sociality score is defined as:

$$v_{city} = \frac{\sum_{\lambda \in city} s(\lambda) * pop(\lambda)}{pop(city)} \tag{2.5}$$

the areas of the city not served by public transport are considered to have zero sociality score. The city sociality is the typical number of people that a person living in the city can potentially meet within a typical daily trip.[21]

In this thesis, based on this scoring model, the line planning will strive to find a new line to make the PTN with new line could get the maximum city sociality score.

2.3 Performance improvement

As we mentioned above, the score calculation of Roma takes 7 seconds on average to run the code [33] on Intel(R) Xeon(R) Gold 6140 CPU @ 2.30GHz with 16 CPU Threads and 64 GB mem. This section describes how we increase the computational efficiency.

First, let's look at how the city is divided into hexagon area. Figure 2.2 shows the population distribution of a region in Rome. Figure 2.3 shows the isochrones that the selected hexagon can reach the other hexagon through public transport. Through these two figures, we can see how the hexagon divides the city into multiple regions. Through the divided regions, the calculation of isochrone from one hexagon region to other hexagon regions could be done.

Figure 2.2. The distribution of population in a certain area of Rome is divided into hexagons[39]

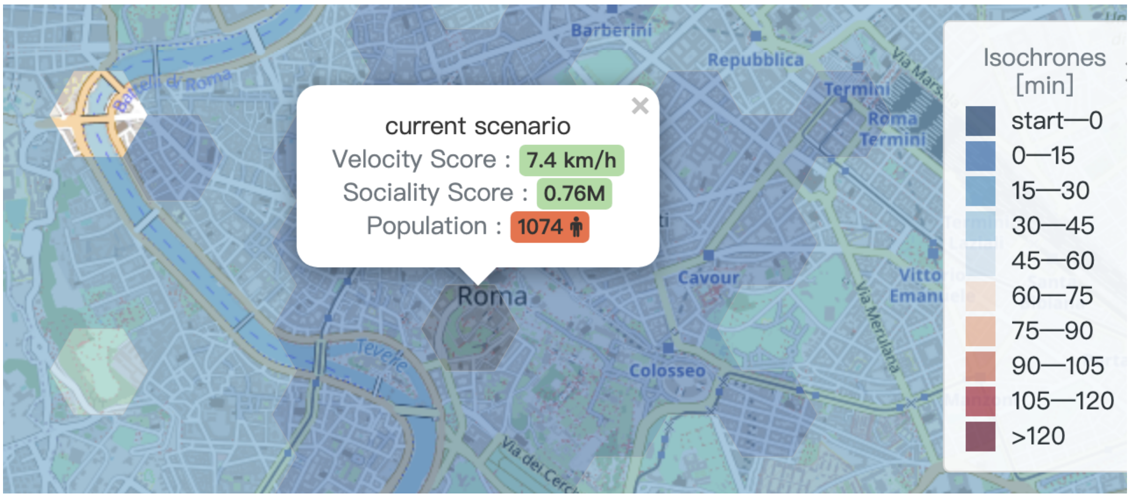


Figure 2.3. The isochrones of the map divided by the hexagon area. The isochrones is calculated with the selected hexagon as starting point[39]

When we calculate the city score, we need accessibility information and population information to calculate the sociality score and velocity score, as shown in Figure 2.1. Because we need to calculate the score frequently when we add and adjust new line. Every change of the line will mean the change of score. We need to recalculate the score and make comparison again and again. the efficiency of the code to calculate the score is particularly important. So in the experimental stage, we tried to improve the performance of the scoring model. We think about whether there is a way to improve the computational efficiency in the experimental stage. First of all, we thought about redesign the hexagon, because the population data information is divided by the hexagon also when calculating the isochrones, whether the region can be reached at time t is determined by hexagon. When calculating the isochrone starting from a hexagon, we need to calculate the reachable time t from this starting point to all other hexagons. Therefore, the more the number of city hexagons, the more areas the city divides, and the higher the complexity of calculating the isochrone, because each hexagon needs to calculate its time to all other hexagon areas. Therefore, we expand the side length of the hexagon from 0.2km to 0.5km. The number of hexagons in the whole city has declined. After such modification, the average calculation time of the scores decreases from 7 seconds to 4 seconds, nearly 50% improvement in efficiency.

Chapter 3

Meta-heuristic optimization algorithms and PSO

In this chapter, we will explain what is our optimization problem. Then we will introduce the class of meta-heuristic algorithms, and then the specific PSO category that we will use for line planning.

3.1 Optimization problem

This paper aims to solve the problem of how to select a new route in urban line planning so that the whole PTN can get a higher sociality score. In short, we hope to find a new line, which can make the city PTN get the maximum sociality score within certain limits (such as the fixed number of stations, the fixed starting and ending stations, etc.). The new line is composed of stops and the path between stops. The most important attribute of the stop in this paper is the geographical location, and the coordinates can be calculated by longitude and latitude on the map. This paper does not focus on the attributes of the line between stations, but can be directly considered as a straight line between two stops. A line composed of five stops can be expressed as $line = (S_1, S_2, S_3, S_4, S_5)$. What we want to do is to adjust the location of the five stations to maximize the sociality score calculated after the line is added to the PTN of the city. From the previous introduction, it can be seen that the calculation of the city's sociality score is very complex. First of all, we need the urban population distribution data and public transport data, follow the concept of accessibility, and use the routing method ICSA algorithm to calculate the sociality score, and then average the entire urban area to get the sociality score of the whole city. The complexity of the formula is beyond the scope of the general mathematical model. So how to choose the route efficiently and reliably? What kind of problem does our line planning problem belong to?

Some calculation problems are deterministic, such as addition, subtraction, multiplication and division. As long as you deduce according to the formula, step by step, you can get the results. However, there are some problems that cannot be directly calculated step by step. For example, the problem of finding large prime numbers. Is there an algorithm that you can work out step by step with a set of formulas? What's the next prime number? There is no such algorithm. Another example is the problem of factoring prime factors with large composite numbers. Is there a formula to directly calculate the respective factors by substituting the composite numbers? There is no such formula.

The answer to this question can not be calculated directly, but can only be obtained through indirect "Guessing". This is the problem with uncertainty. These problems usually have an algorithm, which can not directly tell you what the answer is, but can tell you whether a possible result is correct or wrong. This algorithm can tell you whether the answer to "guess" is correct or not. If it can be worked out in polynomial time, it is called polynomial non deterministic problem **NP**. However, there is a set of problems in NP that are proved to be more or at least as difficult to solve than all other problems in NP. These problems are called **NP-complete(NPC)**[53].

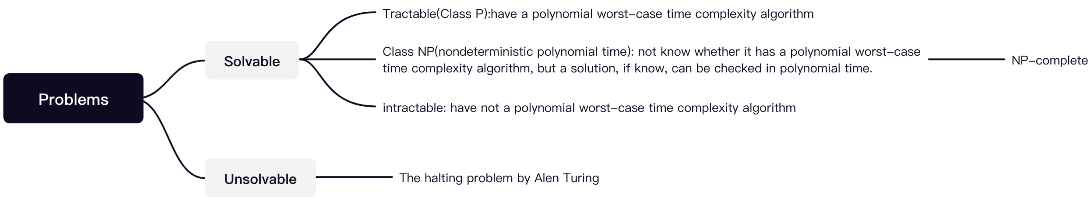


Figure 3.1. Shows a brief description of NP and NP complete and their categories

In general there is two kinds of NPC problems called decision and optimization problems, where the first consists of finding if it is true or not and the other represents a problem where an optimal solution needs to be found.[54] Our line planning problem is a NP-complete problem. When looking for a new line, the location of each station can be placed in any coordinates of the city map. Suppose that each station has 1000 locations. Then, for the combination of N stations line, there will have the possibility of $1000!/(1000-N)!$ combinations(Suppose that two sites cannot be in the same location). So far when we use scoring model to evaluate the PTN after line planning, even if someone gives us a new "optimal line", it is difficult for us to verify whether the "optimal line" is optimal. But, as long as we try all the possibilities, we can get the optimal solution, and the score calculation for each combination can be completed in a short time. The problem is that the combinations are often more than 1000^N, and even if each calculation can be completed in a short time, the whole process of violent verification will be extremely time-consuming.

3.2 Heuristic and Meta-Heuristic algorithms

What method or algorithms is suitable for solving NPC problem? Heuristics and meta heuristics are good methods to solve NPC problems. Heuristic strategy is a general term for a kind of strategies that can give a solution to a specific problem in an acceptable time and space, but do not guarantee the optimal solution (and the deviation between the feasible solution and the optimal solution). Many heuristic algorithms are quite specific and depend on a specific problem. In the process of seeking the optimal solution, heuristic strategy can change its search path according to individual or global experience. When the optimal solution of the problem becomes impossible or difficult to complete (NP-complete problem), heuristic strategy is an efficient way to obtain feasible solution. In short, the heuristic strategy in a limited search space, greatly reduces the number of attempts and can quickly solve the problem. Many important discoveries made by scientists are often based on very simple heuristic rules.(need to cite more)

Meta-Heuristic Algorithm is an improvement of heuristic algorithm. Meta-heuristic strategy usually do not rely on the specific conditions of a problem, so they can be applied to a wider range of aspects. It is the combination of random algorithm and local search algorithm. Meta-heuristic algorithm is proposed relative to the optimization algorithm. The optimization algorithm of a problem can obtain the optimal solution of the problem, while meta heuristic algorithm is an algorithm based on intuition or experience, which can give a feasible solution of the problem at an acceptable cost, And the deviation degree between the feasible solution and the optimal solution cannot be predicted in advance. Considering that our line planning problem is a kind of NPC problem, meta-heuristic algorithm is very suitable for this kind of problem. This paper uses meta-heuristic algorithm to solve the line planning problem. Meta-Heuristic Algorithm includes simulated annealing algorithm, genetic algorithm, ant colony optimization algorithm, particle swarm optimization algorithm, artificial fish swarm algorithm, artificial bee colony algorithm, artificial neural network algorithm and so on. These meta-Heuristic algorithms have their own advantages and disadvantages. As shown in the table 3.1, the advantages and disadvantages of GA algorithm, ant colony algorithm and PSO algorithm are compared.

	Genetic algorithm (GA)	Ant colony algorithm (ACO)	Particle swarm optimization algorithm (PSO)
Advantages	Fast convergence and good versatility	It has strong global search ability, strong robustness and easy to combine with other algorithms	Fast convergence, few parameters, simple and easy to operate, easy to combine with other algorithms
Disadvantages	Easy to converge to local optimal	It is easy to fall into local optimal solution and weak in solving continuous problems	The search ability is not strong in the late iteration, and it is easy to fall into the local optimal solution
Applications	Combinatorial optimization problems , continuous optimization problems	Discrete optimization problem, combinatorial optimization problem	Continuous optimization problem

Table 3.1. Advantages and disadvantages of three meta-heuristic algorithms

So, which meta-heuristic algorithm should we choose for line planning? After the introduction of the optimization problem in the first section of this chapter, we can know that what we want to optimize is the transport line, more specifically, the longitude and latitude position of each stop of the line. Therefore, the longitude and latitude position of each stop of the line is our variable to be optimized. Through the introduction of the scoring model, we can know that expect public transport schedule and route data, the scoring model also includes the walking time of citizens from one stop to another and the population distribution. Because the stop location reflects the value of longitude and latitude, even the slight change of address will lead to the change of walking time and social score of the city. In this way, the station location problem can be solved as a continuous problem, and the station location can be discretized to make the line problem as a discrete problem. In this paper, the problem is regarded as a continuous problem. The stop address is regarded as a continuous variable.

As a meta-heuristic algorithm, the implementation of PSO is relatively simple and suitable for solving continuous problems. Many known modules of general and meta-heuristic algorithms can be added to the PSO algorithm, such as local search, simulated annealing, etc., which can improve the efficiency of the algorithm when necessary. Therefore, This paper will use particle swarm optimization(PSO) algorithm as the algorithm to find the optimal solution of line planning.

3.3 Particle Swarm Optimization algorithm - PSO

Particle swarm optimization algorithm was developed by J. Kennedy and R. C. Eberhart et al [6]. The idea of Particle swarm optimization algorithm is based on the bird predatory behavior. It simulates the behavior of birds flying and foraging in groups, and makes the group reach the optimal goal through collective cooperation among birds. The basic idea of particle swarm optimization: Imagine a group of birds randomly searching for food. It is known that there is only one piece of food in this area; all the birds do not know where the food is; all birds can communicate with their peers. When a bird finds that it is closest to the food, it will inform other birds of its location. So the birds will know the position of the bird which nearest to the food. So what's the best strategy for finding food?

1. Search the area around the bird that closest to the food so far.

2. Judge where the food is according to your own flying experience.

The idea of PSO algorithm is just like the example of birds foraging above. The main idea is to combine the two kinds of search experience of individual (particle) and swarm (swarm) (the so-called experience for birds is the position closest to the food so far. For the equation $f(x, y)$, experience is the position of x and y which lead the best $f(x, y)$). Each particle in the group needs to constantly search for new positions, accumulate their own experience and record their optimal positions, and select the optimal position of the whole swarm after each round of particle search and calculation. Particles will constantly change their position according to their own best experience(optimal position) and the best experience of the swarm(optimal position of the whole swarm), in order to seek closer to the optimal results, and constantly search for the optimal possible position.

PSO algorithm is a kind of evolutionary algorithm, It is an optimization method based on swarm intelligence. It seeks the global optimum by following the current optimal value. PSO and evolutionary algorithm have a lot in common. Both of them initialize the population randomly, and both use the fitness value to evaluate the system, and both of them conduct random searches according to the fitness value. Both systems are not guaranteed to find the optimal solution. However, PSO does not have genetic operations such as crossover and mutation but decides the search according to its particle's speed and position. Compared with other modern optimization methods, particle swarm optimization (PSO) has many positive characteristics, such as few parameters need to be adjusted, easy to operate and fast convergence speed, It shows a large applicability in many practical problems such as [43, 44, 45, 49]. At the same time, a lot of research and applications also use swarm algorithm similar with PSO [46, 47, 48]. In PSO, the function or model we will optimize is called **fitness function**, in our paper the scoring model is our **fitness function**. In the scoring model, the longitude and latitude of the stops are the input, and the sociality score of the city is the evaluation result of our scoring model. The evaluation result of the fitness function to be optimized in PSO called **fitness value**. In PSO, all particles need to evaluate the fitness values of the current position through fitness function, so all of particles have fitness values.

- **Fitness value:** For example, If we have the fitness function $f(X) = x_1^2 + x_2^2 + x_3^2$, in this fitness function x_1,x_2 and x_3 are three variables. A swarm with N particles, the position of the i-th particle is $X_i = (x_{i,1}, x_{i,2}, x_{i,3}), i \in 1,2,\ldots,N$, and the fitness value is $f(X_i)$

The value space of each particle's position within the optimization problem is called **search domain**.

- **Position and search Domain:** Suppose that in a d-dimensional target value space, there are N particles in the swarm, in which the i-th particle is expressed as a d-dimensional vector, $X_i = (x_{i,1}, x_{i,2}, \ldots, x_{i,d}), i \in 1,2,\ldots,N$,and all X_i are in the search domain

In addition to the X describing the position of the particle, there are also vectors that describe the **velocity** of the particle. The velocity of a particle also determines the direction and distance of its next move.

- **Velocity:** The "flying" speed of the i-th particle is also an d-dimensional vector, which is recorded as $V_i = (v_{i,1}, v_{i,1}, \ldots, v_{i,d}), i \in 1,2,\ldots,N$

In the process of solving the optimal value of PSO, there is an important concept of **iteration number**. In particle swarm optimization with N particles, the concept of one iteration is: N particles record the optimal position of themselves and the group after each fitness equation calculation, and update their velocity vector V according to the optimal position of themselves and the group. The number of iterations represents the number of iterations of the above process. When the number of iterations reaches the set value, the optimal position of the particle swarm is the result.

In the first iteration, the original PSO is initialized as a group of random particles, Then the optimal solution is found by iteration. At each iteration particles update their position by tracking two "extremes", optimal position of particle itself **Pbest** and optimal position of the swarm **Gbest** respectively.

- **Pbest:** It is called individual extreme value. In each iteration, by comparing the calculated fitness value of the particle with the historical optimal fitness value of the particle. Keep the position of the particle with the optimal fitness value and replace the Pbest. Like the example before, the swarm has N d-dimensional particles. The optimal position of the i-th particle, denoted as $Pbest_i = (p_{i,1}, p_{i,2}, \ldots, p_{i,d}), i \in 1{,}2, \ldots, N$. If in the new iteration, the fitness value calculated by the position $X_i = (x_{i,1}, x_{i,2}, \ldots, x_{i,d}), i \in 1{,}2, \ldots, N$ of i-th particle is better than that calculated by the current $Pbest_i$, then the current $pbest_i$ value is modified to X_i

- **Gbest:** It is called global extreme value. Gbest represents the position of the best fitness value obtained from all the experience of all the particles in the particle swarm. Similar to the updating of Pbest, Gbest is also compared and replaced in each iteration. The difference is that Pbest is the best position of each particle's experience, and the number of Pbest is same with the number of particles. For a particle swarm, there is only one Gbest, which represents the best position obtained in the whole particle swarm experience. denoted as $Gbest = (p_{g,1}, p_{g,2}, \ldots, p_{g,d})$

In the first iteration of PSO algorithm, the position will be initialized randomly in the search domain and velocity of particles will be initialized randomly(We usually sets the maximum velocity, velocity not larger than the maximum value), In each subsequent iteration, i-th particle will use position X_i to calculate fitness function to get fitness value. Then pbest and gbest will choose whether to update according to the fitness value. Finally, before the next iteration, the particle needs to know how to move before the next iteration, because if the particle's position does not change, the iteration will be meaningless. So the particle will update the velocity and position according to the velocity formula, so as to prepare for the next iteration.

When the two optimal values Pbest and Gbest are found, the particles are determined according to the following formula to update the velocity and position:

Update velocity:

$$\boldsymbol{v_{i,d} = w * v_{i,d} + c_1 * r_1(p_{i,d} - x_{i,d}) + c_2 * r_2(g_d - x_{i,d})} \tag{3.1}$$

Update position:

$$\boldsymbol{x_{i,d} = x_{i,d} + v_{i,d}} \tag{3.2}$$

where,

- $v_{i,d}$ velocity of i-th particle in d dimension
- $x_{i,d}$ position of i-th particle in d dimension
- w inertia factor
- c_1 determine the relative influence of the cognitive component
- c_2 determine the relative influence of the social component
- $p_{i,d}$ Pbest of i-th particle, values on dimension d
- g_d Gbest of the group, values on dimension d
- r_1, r_2 random numbers, the standard range is [0,1]. Each iteration takes a new random value, so that the algorithm has the ability of random search

The inertia weight w makes the particles keep the inertia of motion, which makes them have the tendency to expand the search space and have the ability to explore new areas. The acceleration constants C1 and C2 represent the weights of the statistical acceleration terms pushing each particle to the Pbest and Gbest positions. Low values allow the particles to wander outside the target area before being pulled back, while high values cause the particles to suddenly rush or cross the target area. Without the latter two parts, $C1 = C2 = 0$, the particles will fly at the current speed until they reach the boundary. Because it can only search a limited region, it is difficult to find a good solution. If there is no first part, i.e. $w = 0$, then the velocity only depends on the current position of the particles and their best historical positions Pbest and Gbest, and the velocity itself has no memory. Suppose a particle is in the best position in the world, it will remain stationary. The other particles fly to the weighted center of their best position Pbest and global best position Gbest. Under this condition, PSO will shrink the statistics to the current global best position, which is more like a local algorithm. After adding the first part, the particles tend to expand the search space, that is, the first part has the ability of global search. This also makes the role of w to adjust the balance of global and local search ability for different search problems. If there is no second part, that is, $C1 = 0$, particles have no cognitive ability, that is, the "social only" model. Under the iteration of particles, it has the ability to reach a new search space. Its convergence speed is faster than that of the Standard Version, but for complex problems, it is easier to fall into local optimal points than that of the standard version. If there is no third part, that is, $C2 = 0$, then there is no social information sharing between particles, that is, the "cognition only" model. Because there is no iteration between individuals, a group of M is equivalent to the operation of m single particles. So the probability of getting the solution is very small. The above is the explanation of PSO and the significance of each part. As shown in the figure 3.3, it shows the change of a particle swarm with 49 particles in the process of finding the optimal value of Ackley function. In this example, Ackley function is the fitness function to be optimized, whose global minimum 0 appears in coordinates [0,0]. This example has 100 iterations. At the beginning, shown in figure 3.3(a) , the particles are evenly distributed in the area of $7*7$ (usually the particles are randomly initialized). After initialization, particles will bring their position X_i into fitness function to calculate fitness value. The particle individual optimal $Pbest_i$ and swarm global optimal $Gbest$ are found. The particles update their velocity V_i and position X_i according to $Pbest_i$ and $Gbest$ with equation 3.1 and 3.2.

With multiple iterations, the particle gradually finds the position close to the local optimal and global optimal shown in figures 3.3(b)3.3(c)3.3(d) The global optimal position and individual optimal position that have been found affect the direction of particles. In the later stage of iteration, the particles gradually converge. It can be seen that most of the particles have converged to the global optimal position, and a few particles are still trapped in their local optimal position shown in figures 3.3(e)3.3(f). However, for the whole PSO, the conclusion is the global optimal solution, some particles do not affect the overall result. So after 100 iterations, PSO finds the Gbest which is close to the optimal position [0,0], and obtains the fitness value of 0 which is the minimum value of Ackley function.

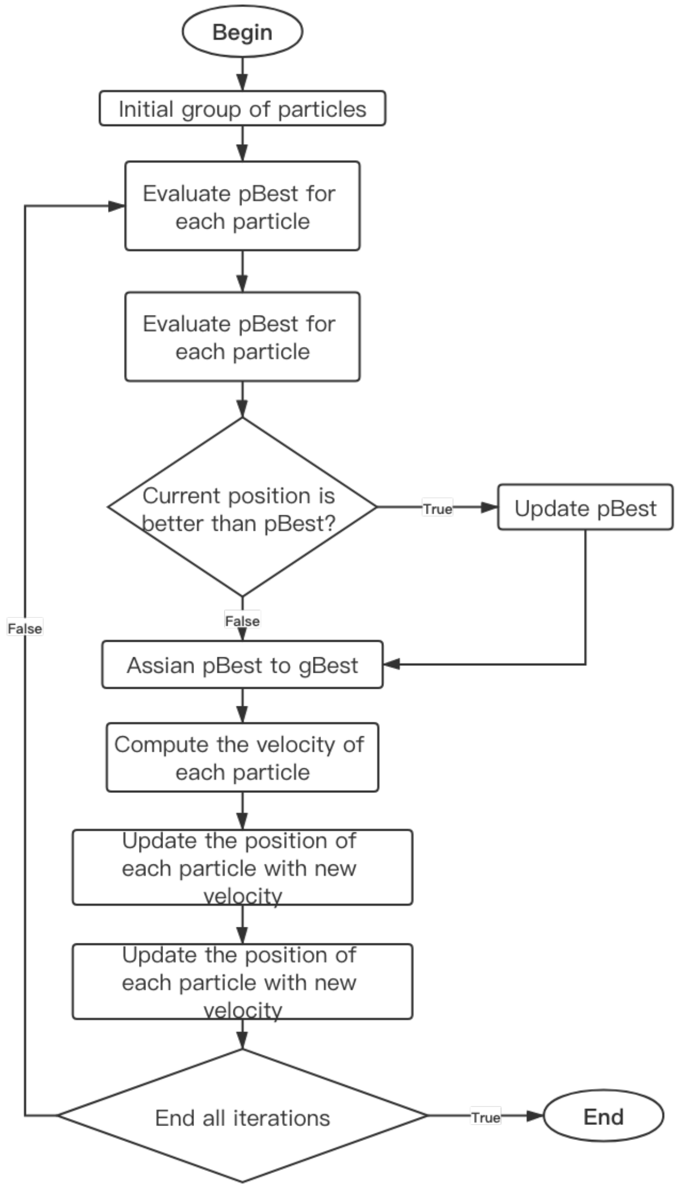
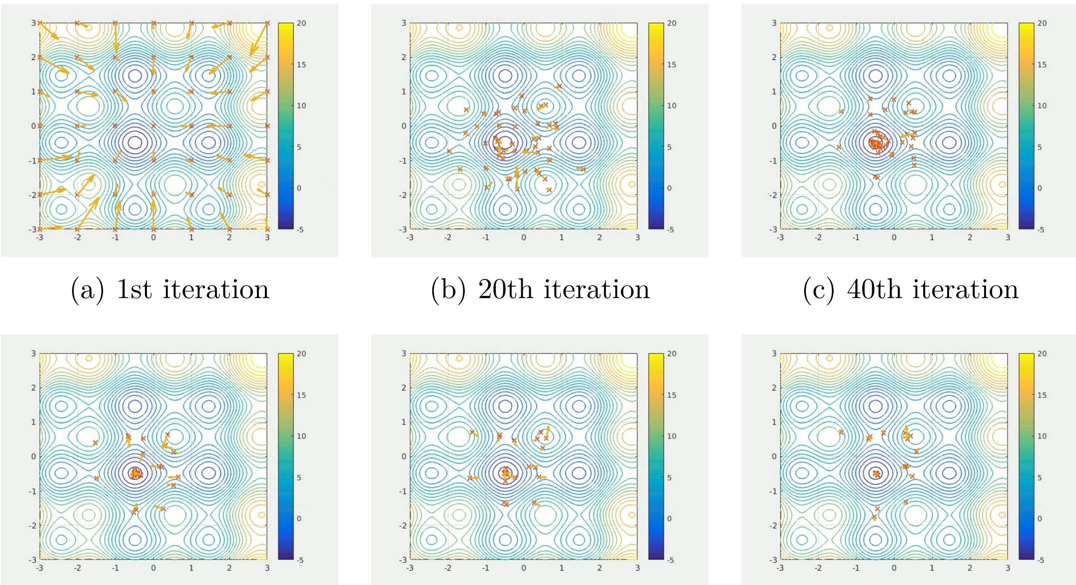


Figure 3.2. PSO algorithm execution flow chart



(a) 1st iteration (b) 20th iteration (c) 40th iteration

(d) 60th iteration (e) 80th iteration (f) 100th iteration

Figure 3.3. A particle swarm with 49 particles shows the position changes of the particles during the 1st to 100th iterations of the optimization of the Ackley function. x indicates the current position of the particle, and the arrow indicates the next iteration position of the particle. The minimum value of Ackley function 0 at position [0,0][20]

Chapter 4

Proposed methodology

Based on the scoring model in the previous chapter as the scoring standard, we can use longitude and latitude to represent the location of new stop, and use the number of new stops and stop positions as parameters as the input of the scoring model to obtain the scientific evaluation of the new line. According to the essence of line planning, a new line can be found on the basis of existing PTN, which can maximize the PTN sociality score of the city. So the variable that can change the score under the fixed number of new stops is the position and sequence of stops in the new line. In this paper, the stops location of the new line is our variable. The scoring model is our fitness function. The model takes the city sociality score of the PTN with new line as fitness value. What we need to do is to optimize the fitness function using PSO optimization algorithm, In order to seek the maximum fitness value. Finally, the line with the largest fitness value is selected as the new line. For the purpose of better line planning with optimization algorithm, we confirm the overall goal. On the basis of using PSO algorithm, the way to get the best line planning is the content to be discussed in this chapter. Facing the line with multiple stops, we need a strategy that can help us to reach the optimal value to select the stop location and order. We use PSO algorithm to support the score optimization scheme. First, we make clear that we want to optimize the position of a public transport line, it is important to transform the stop position with sequence into the parameters that scoring model and PSO algorithm can support. The scoring model supports adding a new line with any number of stops and calculating the city sociality scores of new PTN(with new line added), and it only needs to provide the longitude, latitude and sequence of stations. Latitude and longitude are not only accurate, but also well combined with the map, which is suitable for visualization. And the scoring model also supports longitude and latitude as the location of the stop. As for PSO algorithm, the same as the scoring model, PSO supports the introduction of stop longitude and latitude, and also supports multi-dimensional variables fitness function. So the longitude and latitude coordinates of the stops are selected as the input of our scoring model, that is, fitness function.

If the longitude and latitude of each station are taken as precise coordinates, we will get a two-dimensional coordinate value $[x_i, y_i]$ to represent the stop orientation. Taking the new line of N stations as an example, $L = [x_1, y_1, x_2, y_2, \ldots, x_N, y_N]$ represents a line L composed of N stations, and the order of stations can also be expressed as the order from left to right. Then the starting station of line L is $[x_1, y_1]$ and the terminal station is $[x_N, y_N]$. PSO is used to optimize the scoring model, and L is taken as line planning example. We can divide the optimization strategy into two kinds, global optimization and local optimization.

In general, the difference between them is that the input of each iteration of PSO algorithm is different, although the scoring model is both used as the fitness function. In global optimization, the position of i-th particle X_i(same meaning with X_i in chapter 3) of PSO is the location of all stations in the line L, while the input X_i of local optimization is the location of one stop. It needs to be explained here is that although the local optimization mentioned in this paper is the PSO optimization object with only one stop, but the calculate of fitness value it still uses the scoring model with whole new L as input.

Global optimization means that we introduce new line with N stops $L = [x_1, y_1, x_2, y_2, \ldots, x_N, y_N]$ as a whole parameter into the PSO model, In the first iteration, PSO initializes stops of L with random position(Randomly initialize the position in the search area). The initialized L will be used to calculate the city sociality score in the scoring model. Then using this city sociality score as the fitness value of optimization, through iteration and particle movement, PSO optimization process in Chapter 3 is adopted to update the position of particles in each iteration and bring new particle position into the scoring model for scoring, and keep the highest fitness value particle position as Gbest. Until the end of the iteration, we will get the maximum city sociality score so far and all the stops locations of the L line that got this score. The advantage of this approach is to be able to optimize globally. From the input parameters, we can see that every iteration of optimization is the transformation of the whole line. This method can also search in a large range of maps and change flexibly, which is more suitable for finding the optimal route for the wide range of unconstrained cities with underdeveloped PTN. But the disadvantages are obvious, the lack of constraints will cause a large deviation in the optimization results of each experiment in a limited number of optimization iterations. Moreover, the optimization of multi stations may mean that the dimension of PSO input is very high, which means that if the line y has 20 stations, the object to be optimized by PSO will become an array parameter of 2 * 20 dimensions, and the optimization difficulty can be imagined.

As for local optimization, the line $L = [x_1, y_1, x_2, y_2, \ldots, x_N, y_N]$ is still taken as an example. The location of only one stop is changed by fixing other stations at each time. For example, for line x, fixed $[x_2, y_2, \ldots, x_N, y_N]$ only changes $[x_1, y_1]$, and it can easily add space limit for $[x_1, y_1]$ to limit its moving position. But the calculate of fitness value it still uses the scoring model with whole new L as input. So PSO can search for the best of $[x_1, y_1]$ in a search area. In summary, it is the position of only one stop optimized by PSO at a time, so for the optimization of the i-th stop in new line, only the position of $[x_i, y_i]$ needs to be optimized and changed. Local optimization is more suitable for the line planning in real life. For example, the city PTN is relatively robust. In order to alleviate the traffic pressure between the north and south regions, the initial line can be determined throw south and north first, and then each station can be optimized one by one by using the local optimization strategy through PSO. Finally, the optimization is achieved. The local optimization is in line with the actual station optimization strategy. At the end of this paper, As part of the experiment, the metro D under construction in Rome is optimized, using the local optimization strategy. Through the adjustment of Metro D line station by station, the city sociality score is about 8.0% higher than the original line.

4.1 Greedy strategy

The greedy strategy in this paper is a strategy to search the optimal line, it is the specific implementation strategy based on local optimization. When we do line planning with local optimization method, we need a strategy to select each stop. Considering that the construction of the stop in real life often has a general construction idea in the early stage of the design, mainly to solve the practical problems and design the direction of the line. One of our greedy strategy is based on the fixed first stop and last stop, inserting stops one by one between the first stop and the last stop until the all stops

of line plans is selected. It is called greedy strategy, because it is based on the local optimization mentioned above, and it is to search for the best location of one stop without changing other stops. The location of the stop will find the optimal location and score in the iterations of PSO, and then fix the location of the stop and do not change it, then proceed to the optimal search of the next stop. Every time a new stop location is found, it will be considered that the new stop location is optimal for the current line plans, That is to say, for the whole line, the location of the stop searched is locally optimal. There are two ways to implement the greedy strategy in this paper, taking the construction of a line with N stations $L = (S_1, S_2, \ldots, S_N)$ as an example. details used here are as follows: **The first way** is to fix the location of the starting stop $S1$ and the last stop S_N. After the location of starting station and terminal station are determined artificially, we need to search the optimal location of N-2 stations. First step is to search the location of S_2 which is the next station of S_1. The search of S_2 station will use PSO algorithm. The location of S_2 is searched in the search area(it is the same "search area" within chapter 3 PSO part) Z. In this way, the initialization of all particles in the first iteration of PSO, the particles must be generated randomly in Z. In the process of PSO iterations and particle motion, for those particles moving outside Z, we force the particles to move to the nearest Z boundary. There are many rules for making Z area, which can be flexibly limited according to the angle and distance between stations. Here is what we did in our experiment as an example of greedy strategy, and a method to choose Z.
Example of set a new line in Roma with greedy strategy: The first step is to initialize and fix the first and last stops , In the figure, they are represented by blue and red. Suppose the distance between two stops is 6km and we expect to build 5 stops between the two stations. We will divide 6km equally into 7 stops, $(7-1)/6 = 1km$. If the stops in the line is straight, the distance between each stop should be 1km. We determine the search area Z of the next stop based on this distance of 1km and the angle between the previous station and the last station. We hope that the new station search area is reasonable, so we use 1km as the base to take $1*1.5$ and $1*0.5$ as the two radii r_1 and r_2. And make two quarter circles at the 45-degree area on both sides of the line connecting the previous station and the last station. As shown in the figure, the green area in the picture as the search area Z for the new stop. As shown in the figure 4.1, we searched for the station (orange) in the first Z area and fixed the stop.

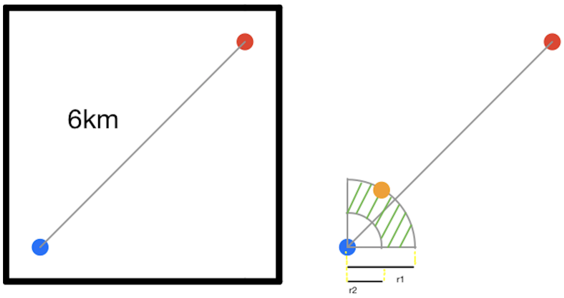


Figure 4.1. search first stop(orange) in the search area Z(green area)

Continue, The rules for zone Z of the next stop are the same as those of the previous stop. Figure 4.2

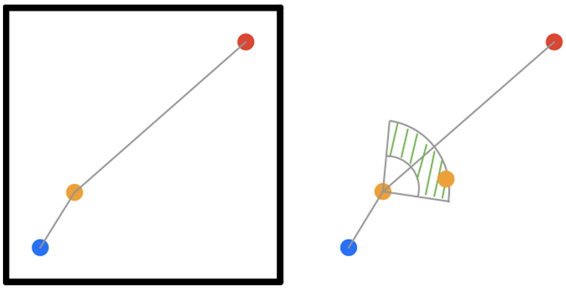


Figure 4.2. search next stop(orange, right one) in the search area Z(green area)

Continue using greedy strategy until all station locations are found, figure 4.3

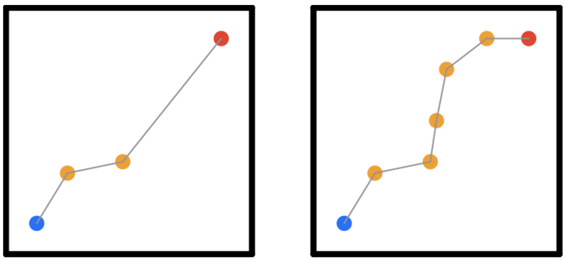


Figure 4.3. search stops

The second way is to make a preliminary planning of the whole line before optimization, so we will get all the stations' location of a line at beginning.

Then based on the line, we do the optimization on each stations one by one from the S_1 to S_N and the fixed station should be from $(S_2, S_3, \ldots, S_N)$ to $(S_1, S_2, \ldots, S_{N-1})$. The difference from the first way in the first step is that the fixed station of the second way is S_1 to S_N at beginning and the first way fix only first stop and last stop. Another difference is that the first method has only two initial stations, while the second method has N initial stations. We have conducted experiments on both of two methods. For detailed experimental content, see Chapter 5

4.2 Improved PSO

Although the particle swarm optimization algorithm in solving the optimization function and multi-dimensional optimal problem, it shows good optimization ability. Through iterative optimization calculation, the approximate optimal solution can be found quickly. But the basic PSO is easy to fall into the local optimum, leading to poor results. And the convergence effect of particles is also a very important aspect of how to improve the algorithm. In view of this situation, we consider how to avoid the PSO into the local optimal strategy. There are two main aspects of the strategy. 1. Transform the PSO itself. Various improved PSO algorithms are studied. 2. Combining PSO algorithm with other intelligent optimization algorithms. Various hybrid optimization algorithms are studied. For example, PSO combined with simulated annealing algorithm can avoid falling into local optimum with the randomness advantage of simulated annealing algorithm[50]. PSO combined with GA algorithm[51]. The above two methods can improve the performance of some aspects of the algorithm. This paper mainly adopts the first way to improve the basic PSO.

Original PSO formula for updating particle velocity:

$$v_{i,d} = w * v_{i,d} + c_1 * r_1[0,1](p_{i,d} - x_{i,d}) + c_2 * r_2[0,1](g_d - x_{i,d}) \tag{4.1}$$

According to the PSO velocity and particle position update formula 4.1, it can be seen that the larger the w is, the faster the particle flies, and the particle will search globally in a larger step size; The smaller the w is, the smaller the particle step size is and tends to local search. w is called the weighting factor. The larger weight factor is beneficial to jump out of the local minimum value and facilitate the global search, while the smaller inertia factor is beneficial to the accurate local search of the current search area and the convergence of the algorithm; However, if w is too large, it will lead to premature convergence and oscillation near the global optimal solution. Therefore, the general method to improve w is to gradually change w from wmax to wmin. It can make the particles have greater activity in the initial stage, and help the particles to search globally in a larger step size. In the later stage of convergence, the search is carried out with smaller step size to increase the accuracy. As shown in equation 4.2, t_{max} can be regarded as the number of iterations.

$$\begin{aligned} w =& w_{max} - \frac{t * (w_{max} - w_{min})}{t_{max}} \\ w_{max} &= 0.9, w_{min} = 0.4 \end{aligned} \tag{4.2}$$

This paper uses the PSO algorithm of weight linear decreasing. Set the maximum and minimum value of w, according to the change of iteration times, keep the large weight factor in the early stage and reduce the inertia factor in the later stage to ensure the function convergence. We call it **w-PSO**

At the same time, this paper also analyzes and compares another improvement of the speed formula. According to the update formula of speed, we can see that the factors that affect the prime are not only the inertia factor w, but also the random value r1,r2 because r1,r2 is a random value in the [0,1] interval. This random value ensures that the particles will always move towards gbest and pbest. According to the improved scheme in [52], in order to avoid the particle falling into the local optimum, the author of this paper changes the random value interval of r1,r2 to [−1,1] 4.3,

$$v_{i,d} = w * v_{i,d} + c_1 * r_1[-1,1](p_{i,d} - x_{i,d}) + c_2 * r_2[-1,1](g_d - x_{i,d}) \tag{4.3}$$

so that the particle has the ability to stay away from the optimal position, which can make the particle more active and explore a larger area in the early stage. In order to verify whether this improvement can be applied to line planning, we change the random value of r1,r2 to the interval of [−1,1], and test whether its better random search ability is applicable to our problem. We call it **lr-PSO**

In the next section, we test PSO and improved PSO with Ackly function.

4.3 PSO optimization example

This section mainly tests the PSO function to test its optimization ability and efficiency. The Ackley function 4.4 is used as the test object .

$$
\begin{aligned}
f(x_0 \cdots x_n) = -20exp(-0.2\sqrt{\frac{1}{n}\sum_{i=1}^{n} x_i^2}) - exp(\frac{1}{n}\sum_{i=1}^{n} cos(2\pi x_i)) + 20 + e \\
-32 \leq x_i \leq 32 \\
\text{minimum at } f(0,\cdots,0) = 0
\end{aligned}
\tag{4.4}
$$

The Ackley function[55] is widely used for testing optimization algorithms. In its two-dimensional form, as shown in the figure 4.4, it is characterized by a nearly flat outer region, and a large hole at the centre. The function poses a risk for optimization algorithms, particularly hillclimbing algorithms, to be trapped in one of its many local minima. Input Domain of Ackley function is usually evaluated on the hypercube $x_i \in [-32, 32]$, for all $i = 1, \ldots, d$, although it may also be restricted to a smaller domain. The global minimum is 0 at $f(0, \ldots ,0)$.

We use ackly function to compare the optimization results for both original PSO algorithm and the improved PSO algorithm. The improved PSO are described last section, one way is to adjust the weight factor adaptively, the other way is to adjust the random value of PSO particle velocity update formula from [0,1] to [−1,1]. For the convenience of observation, we choose two-dimensional variable $[x_0, x_1]$ for equation 4.4.

We use PSO to search the optimal value of Ackley function four times for each PSO algorithm(two improved PSO, and one original PSO), set the number of PSO size to 20, and the number of maximum iterations to 100. For each optimization method, recording average fitness value of four experiments.

As shown in the figure 4.5. In the aspect of optimizing Ackley function, both the original PSO and the adaptive w PSO(w-PSO) have good performance. Although the method of changing the range of random values in the veloity update formula(lr-PSO) does not perform well in the optimization of Ackley equation, its characteristic of not easy convergence is obvious. It can still be desirable in special cases.

The conclusion is that PSO can achieve quite good results in dealing with Ackley function. Both the original PSO and the adaptive w PSO have good performance. Although the method of changing the range of random values in the veloity update formula does not perform well in the optimization of Ackley equation, its characteristic of not easy convergence is obvious. In the experimental stage, this paper also makes some comparison on the optimization ability of these three methods in city line planning problem.

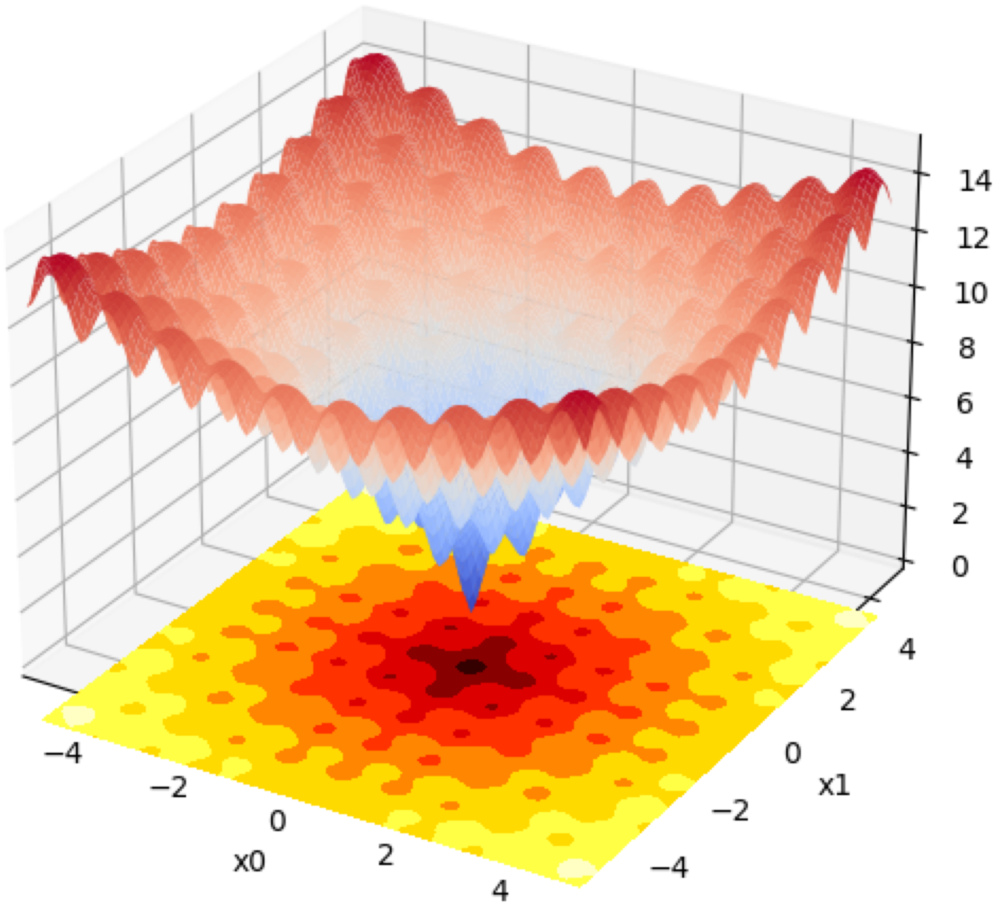


Figure 4.4. The 2-dimensional Ackley functon is as shown in the figure, the global minimum is 0 at $f(0,0)$. There are a lot of local minima.

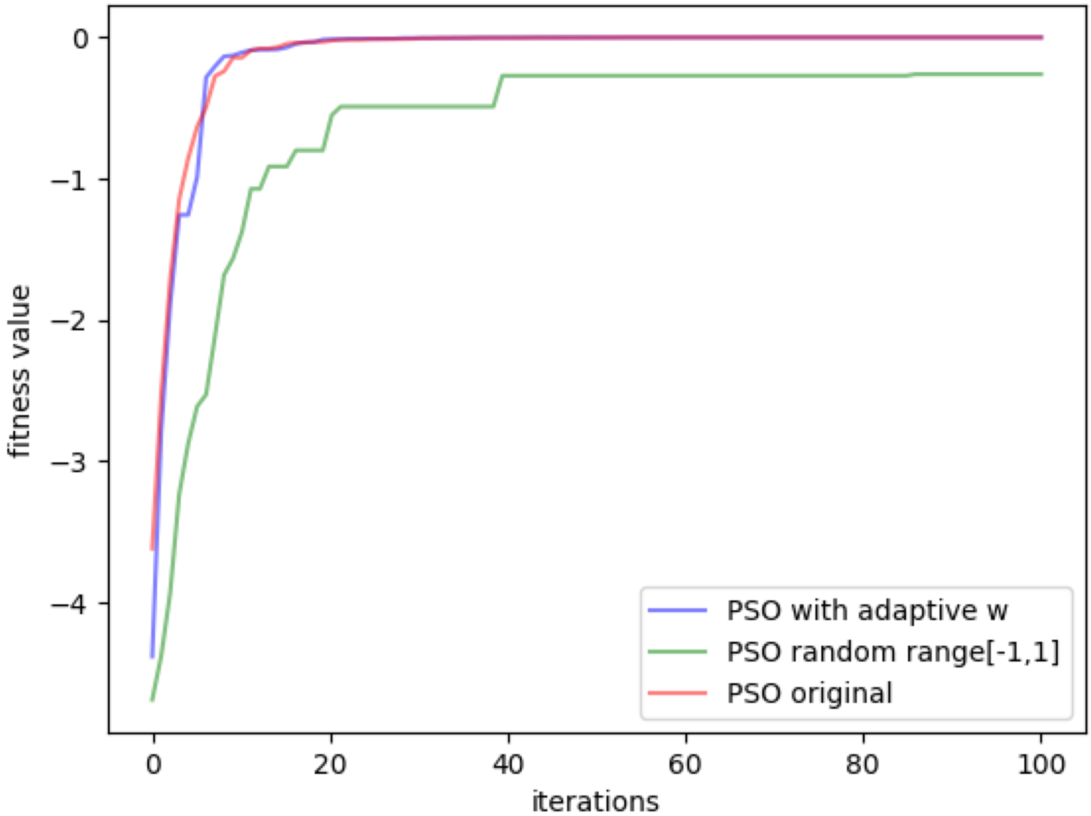


Figure 4.5. The original PSO algorithm and the improved PSO algorithm are compared in optimizing Ackley function

Chapter 5

Experiments and results

In the experiment, we will use improved PSO and greedy strategy to optimize the public transport line. For the two improved-PSO mentioned in Chapter 3, which are respectively for the weight factor and the modified PSO for the range of random values in the velocity formula of PSO. The selection of these two PSO is as follows, For the experiments of 5.1.1, 5.1.2 and 5.1.3, only improved PSO based on dynamic weight factor(w-PSO) is used . In 5.2, we combine the improved PSO based on weight factor with the improved PSO based on modifying the range of random values in the velocity formula of PSO (lr-PSO). Finally, compare the effect of the methods in 5.2 with the same optimized scene in 5.1.3.

5.1 Line optimization

5.1.1 Using greedy strategy to search stops in large scale with no location restriction between stops

Greedy strategy is used on line planning and optimize the transport line. Firstly, greedy strategy and PSO were used to search stations in large scale. The greedy strategy we adopt here is slightly different from the one mentioned above. The one mentioned above is to initialize the start station and last station, but here we only initialize the start station. The idea is to take a fixed station as the starting station, then based on the starting station find the last station in Roma, so that the new last station and the starting station constitute the line with two stations can get the maximum city sociality score. Then fix the position of the last station, add the stations one by one between two stations as the passing stations. During the query process, all station locations will be searched in area Z $(x_{min}, y_{min}, x_{max}, y_{max})$ in the city of Rome, Here Z is a square area bounded by x_{min}, y_{min}, x_{max}, z_{max}. The search area for all stations is the same, all within the area Z, which means In this scenario, two stations can be built at the same location. In the experiment, the position (x_0, y_0) of the starting station S_0 and the number of stations to be added are specified. Taking the starting station S_0 as the starting point, the swarm optimization algorithm is used to find the station S_{last}, So that the line composed of two stations can get the maximum social score. The new line order is S_0 -> S_{last}. On this basis, the stops S_1 to S_n are updated, and the route after N iterations is S_0 -> S_1 -> S_2 -> S_n -> S_{last}. For the optimization of each station, we need to set the parameters of PSO. The main parameters are the maximum particle velocity, the number of particles and the number of iterations etc.

Here the configurations of PSO and search area are as follows:

- $x_{min} = 12.430106$, $x_{max} = 12.614079$, $y_{min} = 41.867929$, $y_{max} = 41.941441$, they are latitude and longitude
- w PSO weight factor dynamic update from 0.9 to 0.4
- v_{max} maximum velocity of particle motion is 0.002
- *size* Size of particle swarm is 20
- *iter_num* iterations of each site update is 200
- *stops_num* number of stops to be built is 9

The experiment is divided into four groups with the same configuration, and the above greedy way and main parameters are used for line optimization query. Figure 5.1 shows the search process of each station in the first group of experiments more directly. It can be seen that the stop optimization shows a kind of irrationality in the case of only limited search space and no limited distance and angle between stops. Although each stop selected represents the stop with the largest city sociality score found in 200 iterations, it is not reasonable as a public transport line on the whole.

Meanwhile, the results of the city sociality score shows in the figure 5.2.

Through the first group of experiment 5.1 , we can see that the stop location without limiting the distance and angle between stops will cause many problems, such as the two stops are too close, the stop spacing is too large, the direction of the line is chaos and so on. Although the idea of generating line is to get a greater city sociality score, the final line has no

(a) find first stop (b) 2th stop (c) 3th stop

(d) 4th stop (e) 5th stop (f) 6th stop

(g) 7th stop (h) 8th stop

Figure 5.1. In the process of line optimization, the optimal position of each stop after 200 iterations optimization is fixed. The figure shows the optimization process of eight stops including last stop. When the 8th stop is found, the optimization process ends.

meaning of practical application, the problems show that we need to further refine the search area of the stops when we use greedy strategy to select stops. And through the comparison of city sociality score optimization of

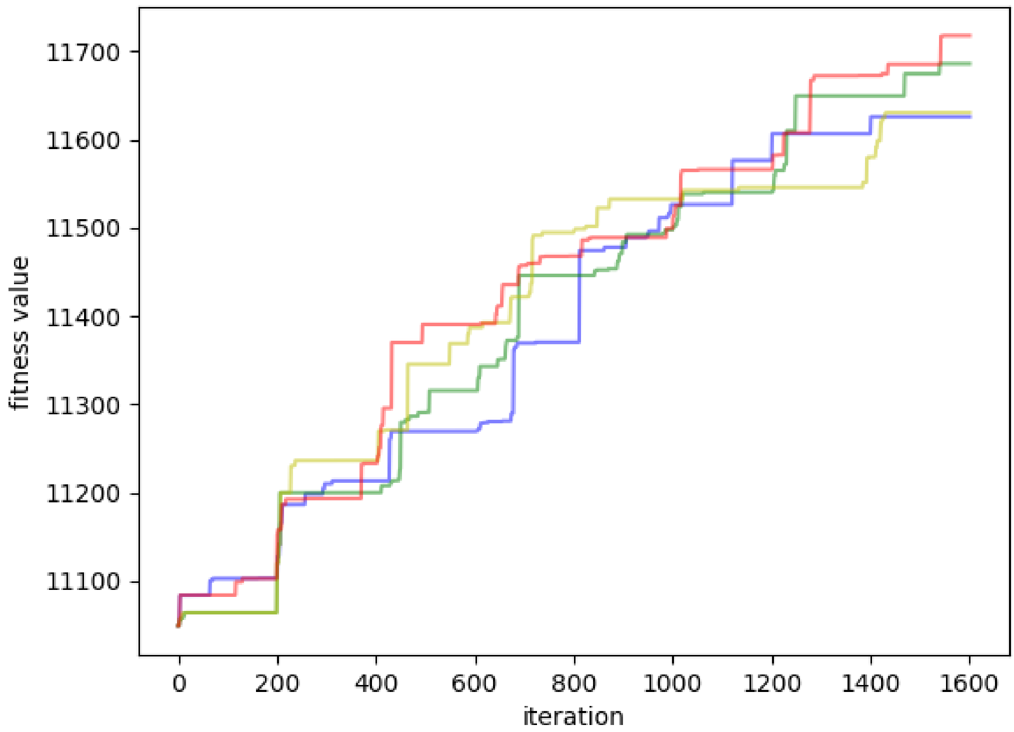


Figure 5.2. In four groups of experiments, the figure shows the fitness value during the optimization iterations. Here the fitness value is the city sociality score

four groups of experiments 5.2, it shows the change of the city sociality score of the line with the number of iterations when the new stop is added, the stop location will be optimized. Because the selection of each station we set is 200 iterations, we can clearly observe that every 200 iterations the city sociality score will have a significant improvement. Finally, after 1600 iterations, all eight stations are selected. We can see that the experimental results of the four groups with the same configuration are quite different. This shows that the optimization of a station with 200 iterations and the number of particles of 20 populations may be insufficient, and this may also be due to the search space is too large, resulting in the particles can not be fully explored.

After experiment, we found that it is difficult to get a valuable route scheme. So we think it is necessary to build new stops based on the existing line to meet some specific conditions. For example, the distance between the new stop and the previous stop is less than the fixed radius (the length of the radius is determined by the distance between the starting and the last stations and the total number of stations. If there are 11 stations within 10km, the interval between each station is required No more than 1.5km). This can not only avoid circuit detours but also reduce the search area.

5.1.2 Using greedy strategy to search stops with location restriction between stops

Because of the problems encountered in the experiment of 5.1.1, we decided to change the way defined the search space. this time we fix the first stop location and the last stop location at beginning, The first station to be searched is between these two stops. We connected the two stops with line L, Use the search space definition method mentioned above, the search area of first new stop is in area Z as shown in the figure 4.1 4.2 4.3, and the distance between first station to be searched and the starting station is less than the linear distance D between the starting station and the last station divided by the N-1 (total number of stations minus 1) multiplied by 1.5 (named **r1**) and larger then D/(N-1)*0.5 (named **r2**). so the new station is chosen from a fan-shaped area, and each station is selected take 200 iterations as optimization times. 4 groups of experiments were carried out. line with 6 stations line will be finally decided.

During the PSO, when the particles move outside the Z area, multiple situations will occur. When the distance between the particles and the previous confirmation stop is greater than r1, the distance from the particles to the previous stop will be restricted to r1. when the distance from the particles to the previous station is less than r2, the distance from the particles to the previous station will be restricted to r2 For the particle moving position outside Z, the particle is selected with Z boundary near to ensure that all particles will always fall in the optional interval Z. The actual experimental results are shown in the figure 5.3, when a greedy selection of new stop location, the particle motion density map of PSO. For the stop search with 200 iterations and 20 particles, a total of $200 * 20$ particle positions are recorded, and their positions are scattered in the fan-shaped area Z. The fan-shaped area in the figure is search area Z. The heat map represents 200 * 2 times of particle movement. It can be found that the density of particles movement is concentrated in several peaks with the highest heat. The high density region is not concentrated in the four corners of the sector, which shows that the algorithm is reasonable when the running position is outside the optional range. The two figures 5.3 5.4 show density maps with different accuracy. The number of particles falling in the region is taken as the density value. The X and Y axes represent the longitude and latitude, and the color represents the number of particles moving in the modified block area. The higher the number, the lighter the color.

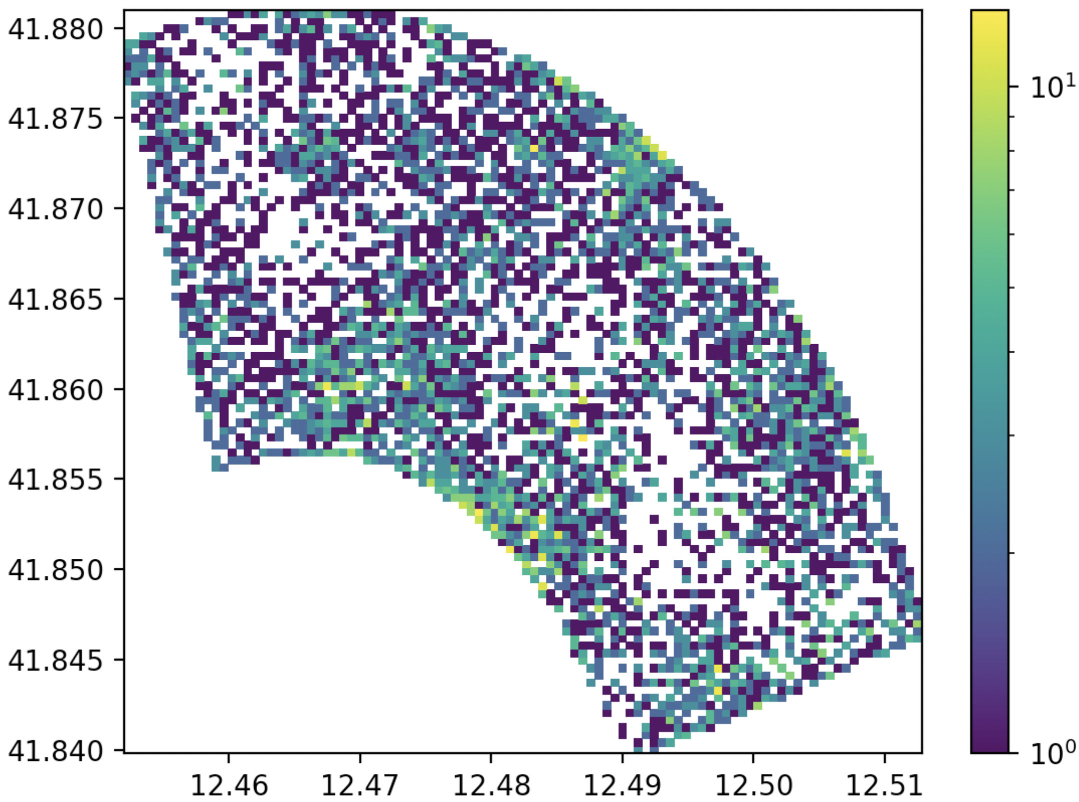


Figure 5.3. The X and Y axes represent the longitude and latitude of a stop, and the color represents the number of particles moving in the modified block area. The higher the number, the lighter the color (Each X-axis and Y-axis has 100 segments, there are 200 times of particle movement, because there are 200 iterations)

Figure 5.6 shows the process of the first group of station addressing in the new method. It can be seen that this time the station addressing is obviously more practical. It can ensure a higher city sociality score and limit the search space of each station, making the new line more reasonable.

The city sociality score optimization of the four groups of experiments with the same parameter configuration is shown in the figure 5.5

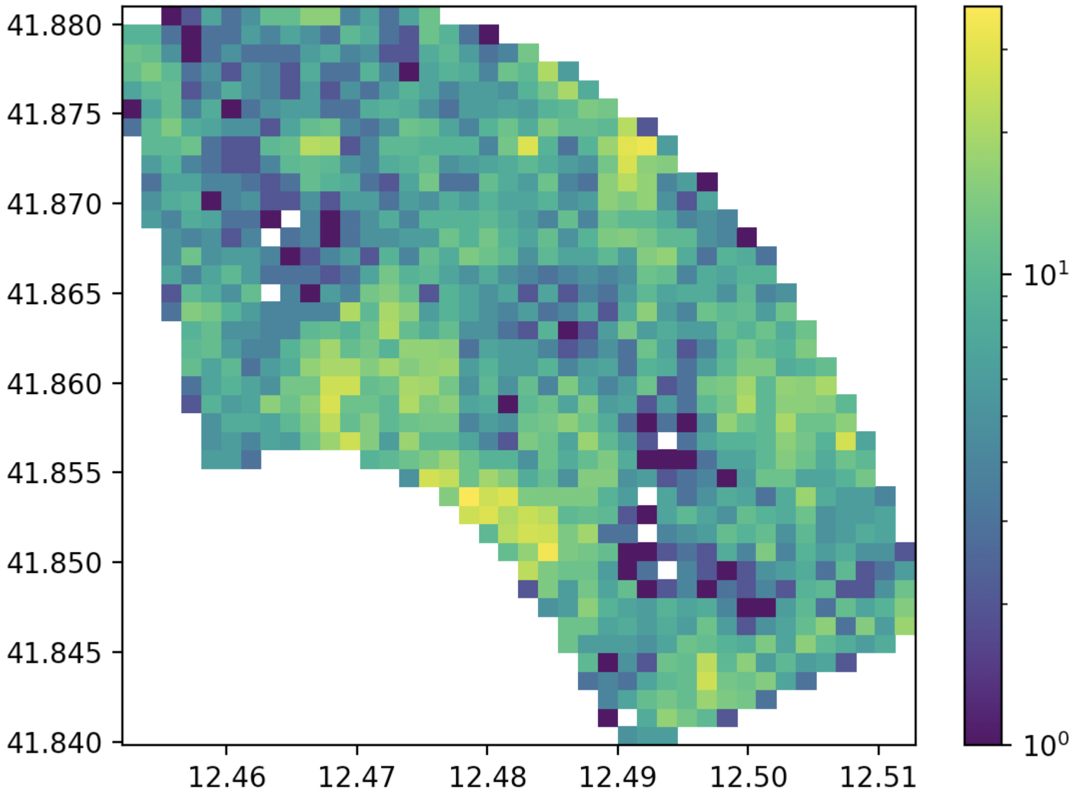


Figure 5.4. The X and Y axes represent the longitude and latitude, and the color represents the number of particles moving in the modified block area. The higher the number, the lighter the color(Each X-axis and Y-axis has 40 segments)

By limiting the location of the new station, we get feasible lines and ensure high city sociality score. This fully solves the shortcomings of the previous experimental design, so that the results can be reference. However, by analyzing the score optimization process of the four groups of data, we found that when the last stop location is finished searching, the results of the four groups are different. And because of the greedy strategy we use for adding new stop between last stop and previous stop, each new stop will make the score get a significant improvement, which makes it difficult to observe the optimization effect of the same number of stations. For example, we can get the city sociality score comparison of four 5-station lines through four groups of experiments with 1600 iterations. And only after all iterations of all stations are completed can we get the city sociality score of the whole

PTN. With such a small number(four groups) of samples, it is difficult to find out whether our final optimization results are superior to the lines with the number of stations. And it is difficult to get a conclusion of the optimization effect. So we try to use greedy strategy for the third kind of experiment. The idea is that we optimize the line of fixed N stops, that is the second greedy strategy mentioned in Chapter 4.

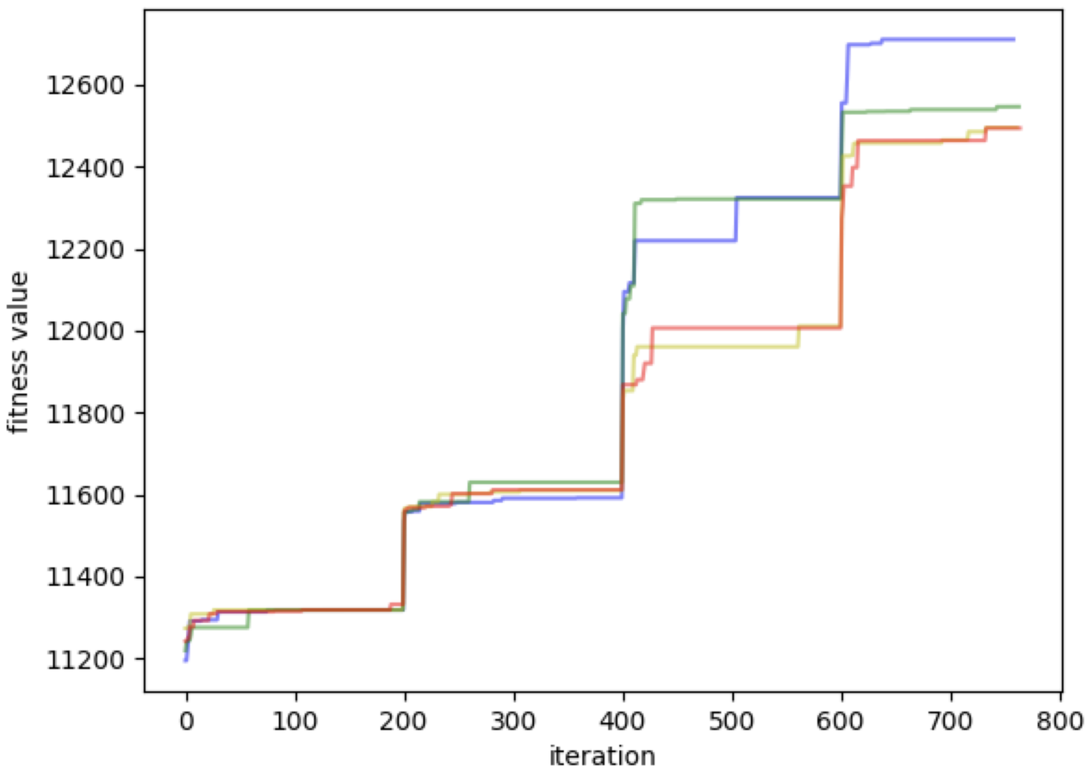
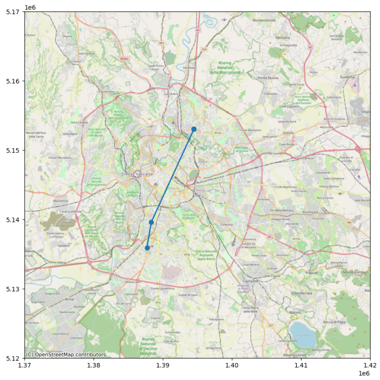
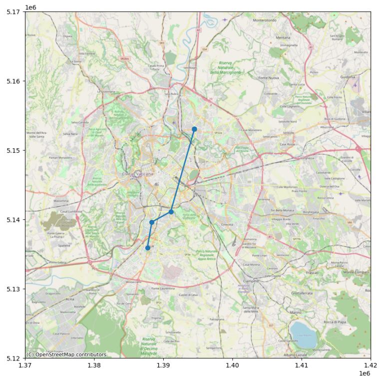


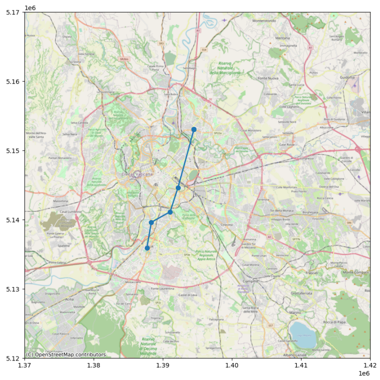
Figure 5.5. Optimization by limiting stops location in fan-shape area



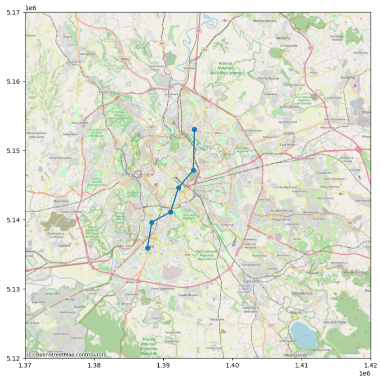
(a) find 1st stop



(b) 2th stop



(c) 3th stop



(d) 4th stop

Figure 5.6. The process of line optimization by limiting stops location in fan-shape area

5.1.3 Optimization on the line with all stops fixed

The second greedy method requires us to not only fix the first stop and last stop, but also give a fixed line for all stops, and then optimize the station location based on the initial line. The advantage of this method is that each optimization result can be easily compared with the city sociality score of the PTN with original line. Every iteration of PSO algorithm can find the influence of stop position change on the city sociality score. And because the number of stops is fixed at the beginning, the whole optimization process will not have the obvious increase of scores caused by adding each new stop in the previous experiment. In order to have a more intuitive and reasonable initialization line, we use the metro line D in the planning of Rome as the initialization metro line. Line D was a proposed line of the Rome Metro system, whose project was shut down in 2012. The project in 2018 was reproposed for a reborn version of the Line. We found the station coordinates of metro D in Google map (table 5.1). Metro line consists of 22 stations, and the main direction of the line runs through the north and south.

Stop name	**Location**
Agricoltura	41.840563548030, 12.473230888930281
Eur Magliana	41.839191506830, 12.462023207004686
Magliana Nuova	41.847371830617, 12.460377626788132
Roma Tre	41.853844931653, 12.466297171703426
Marconi	41.865327493239, 12.467852852897877
Trastevere	41.876065058751, 12.465964577829673
Nievo	41.883190482764, 12.469354889949997
Sonnino	41.887343950958, 12.470856927053982
Venezia	41.895130975886, 12.481435559209263
S.Silvestro	41.902813243343, 12.480899117524887
Spagna	41.906518289413, 12.483055613558003
Fiume	41.911595743555, 12.498129626584431
Salario	41.925933659528, 12.504974623800425
Vescovio	41.933053520764, 12.512248774642222
Prati Fiscati	41.946003516768, 12.514845066219882
Jonio	41.949897560200, 12.525230579297329
Adriatico	41.948237833068, 12.538620166405437
Ojetti	41.947918649976, 12.555099658723092
Talenti	41.950089064552, 12.558017902067167
Cecchinia	41.954493502271, 12.555357150782863
Casal Boccone	41.963560738585, 12.554847531123201
Torraccia	41.974313693871, 12.553860478275505

Table 5.1. Metro D stops location

First of all, we add the original metro D to the city PTN, as shown in the figure 5.7, which shows the change of the city sociality score after adding metro D, the city sociality score of PTN after adding metro D is **12173.27**. It is noteworthy that before adding metro D in Roma, the sociality score of the city's PTN was 10803.24, that is to say, the addition of metro D lines increased the sociality score of the whole city by 13%.

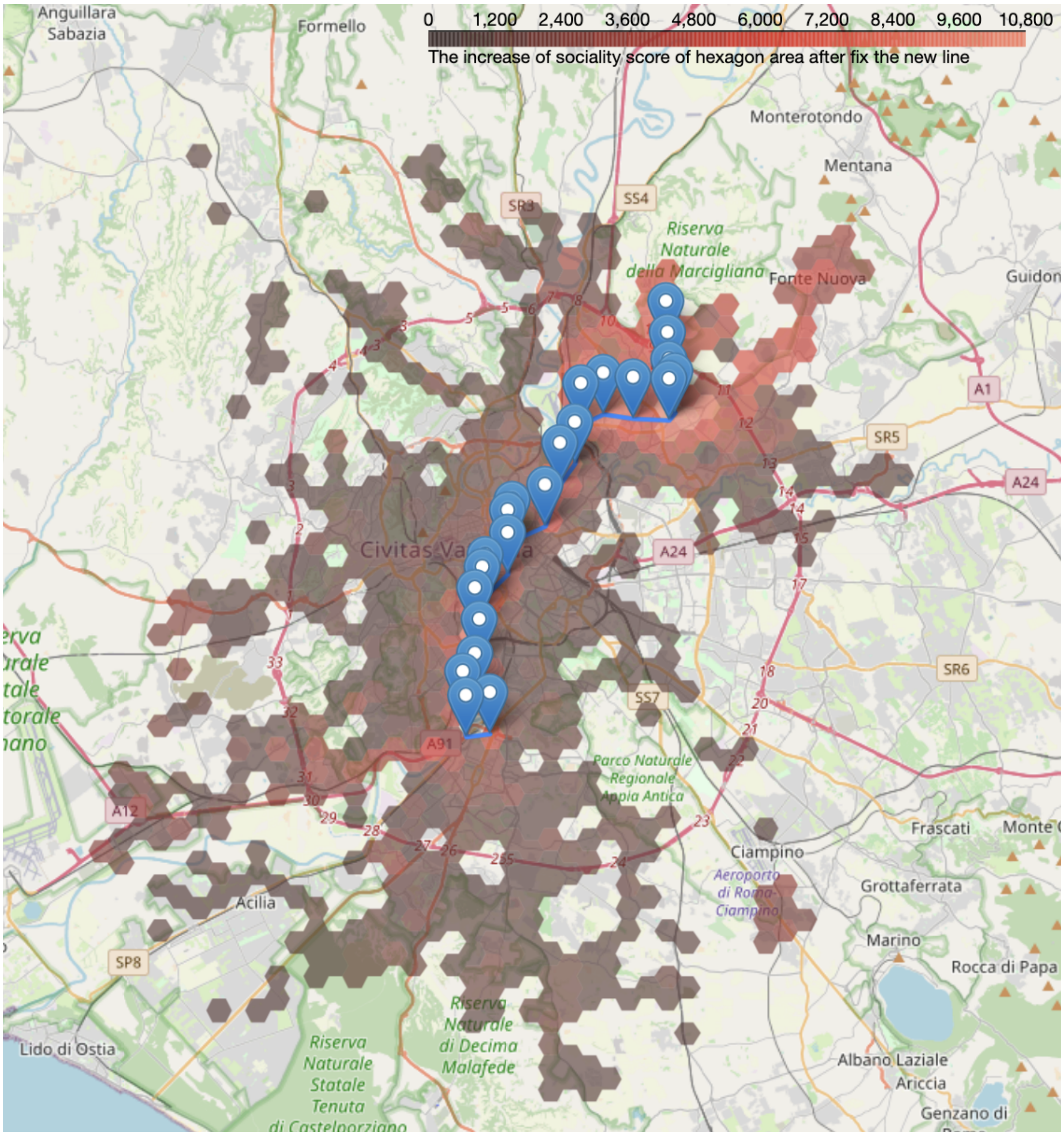


Figure 5.7. Comparison of city sociality score after joining metro D. It shows improvement of city sociality score of hexagon area after built metro D

Based on Metro D, we adjust the stations one by one, and adjust the position of the stations. Each stop is also limited by the search area of the fan-area. In this experiment, we use two groups of experimental 5.8. Take the results of the group with good optimization results from the two groups, the city sociality score optimized from 12173.27 to 12883.57, which increased by 5.8%.

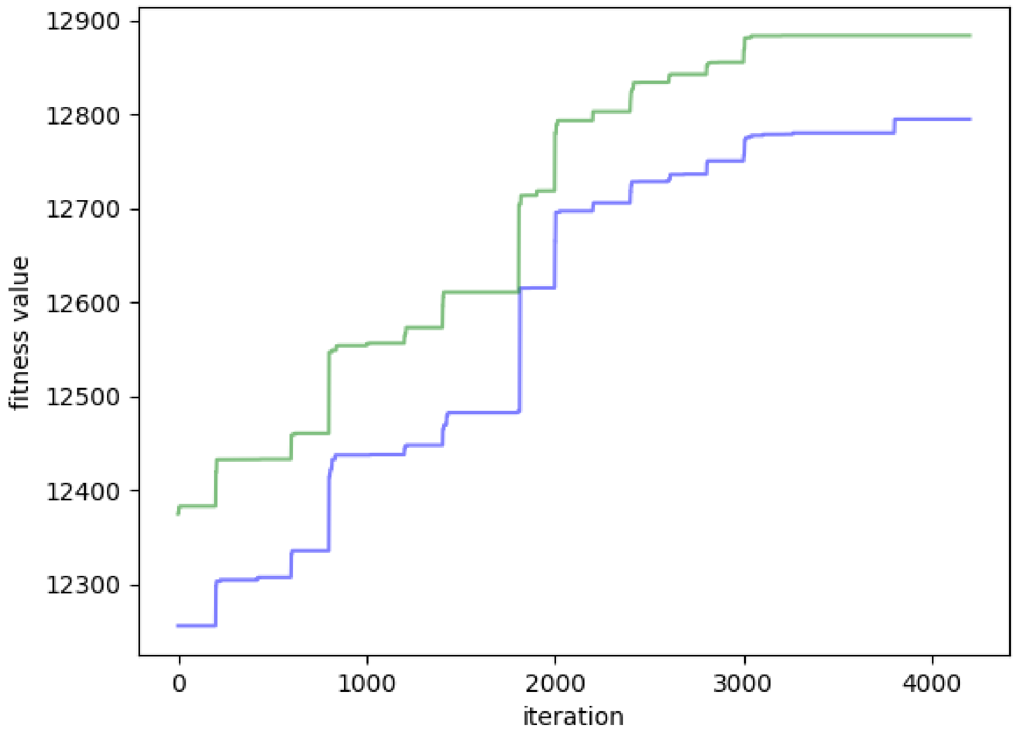


Figure 5.8. The optimization of 21 stations, 200*21 iterations were carried out

It should be noted that the configuration and parameters of the three groups of experiments are as follows

- w PSO weight factor dynamic update from 0.9 to 0.4
- v_{max} maximum velocity of particle motion is 0.002
- *size* Size of particle swarm is 20
- *iter_num* iterations of each site update is 200
- *stops_num* number of stops to be built is 22

5.2 Compare the performance of different improved PSO

In the chapter 4, we mentioned the theory and method of improved PSO. 5.1 experiments are all based on w-PSO the optimization of dynamic weight factor w. In this section, we will use lr-PSO to adjust the random value selection range of PSO partical velocity update formula from [0,1] to [-1,1] to make particles more active, so as to improve the optimization ability of PSO. At the same time, this method does not conflict with the w-PSO, so we will combine the two ways to carry out experiments. We carried out two groups of experiments with the same configuration as in 5.1.3, and the experiment was also optimized for Metro D. The only difference between the experiment and the experiment in 5.1.3 is to modify the random value interval. Compare the results in 5.1.3 as shown in the figure 5.9. Take the results of the group with good optimization results from the two groups, the city sociality score optimized from 12173.27 to 13155.25, which increased by more than 8.0%.

It can be seen that the results of lr-PSO are better than those with w-PSO. Under the same particle swarm optimization configuration, the better results are obtained with the same computing time.

At the same time, as shown in figure 5.10, the comparison of hexagon area sociality score between the new metro D and the original metro D. It can be seen that the new route optimized by our optimization algorithm has a very significant improvement in the urban area compared with the original Metro D.

5.2.1 Parameter hypertuning

In the experimental stage, we analyzed some parameters. For the inertia weight w in PSO, we use dynamic adjustment to update. We also limit the search area of particles. This part, we will show how we choose the maximum velocity of particles v_{max}, as well as the number of population and the number of iterations. The velocity of particles determines the search accuracy of particles, and the v_{max} of particles determines the range of this accuracy. We refer to the comparison between the numerical change of longitude and latitude and the actual distance. In reality, every 0.001 degree of latitude and longitude, the distance difference is about 100 meters. Considering the actual public transport station, we decided to set about 200 meters, which is 0.002 of longitude and latitude, as the maximum particle velocity v_{max}. In general, the population of the partical swarm is better be choosen between 20

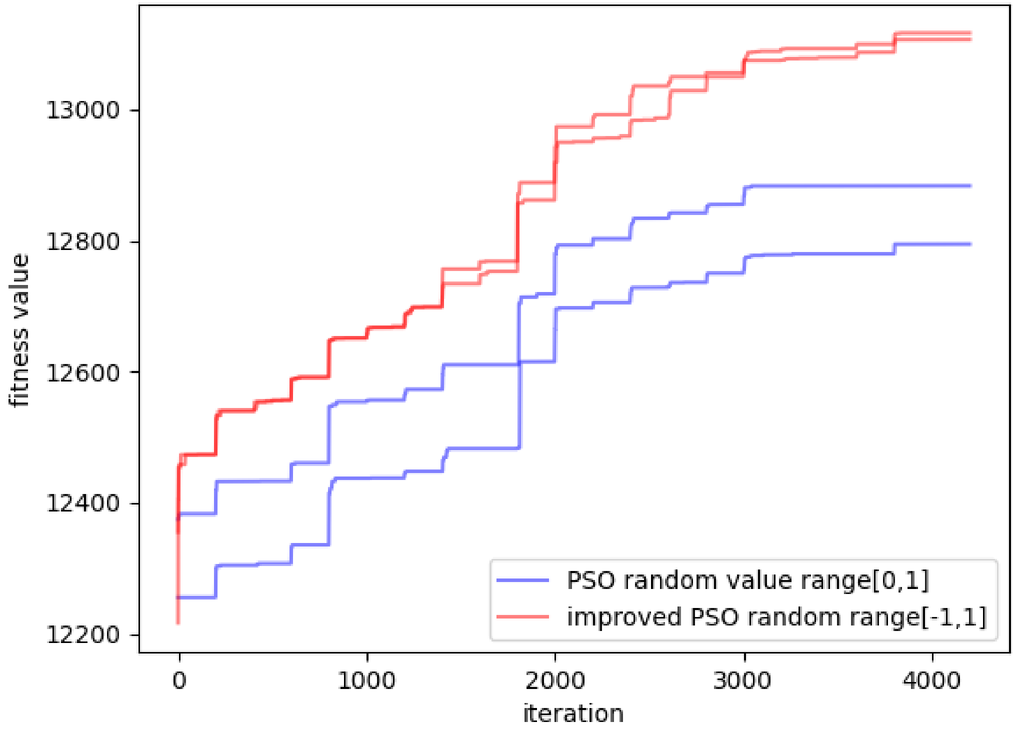


Figure 5.9. Compare the optimization effects of modified random interval and unmodified random interval

and 50 follow the initial suggestion from 1995 [6]. Considering the calculation cost, we decided to set the population of PSO at 20 to reduce the calculation cost. And through experimental observation. We have carried out 42 times of stop location optimization, and each stop has set a maximum number of iterations of 200 times. Among the 42 times of station location optimization, 9 times converge after 100 iterations, and only 4 times converge after 150 iterations. So we finally chose 20 populations and 200 iterations

5.2.2 Complexity and speed

We focus on optimization efficiency, and try to reduce the population P and iteration times I of PSO. For a N stops line. if the time required for PTN score calculation is t, then the time cost could be $N*I*P*t$. Our experiment in 5.2 needs $21*200*20*4s = 336000s$ which almost needs 94 hours. Using greedy

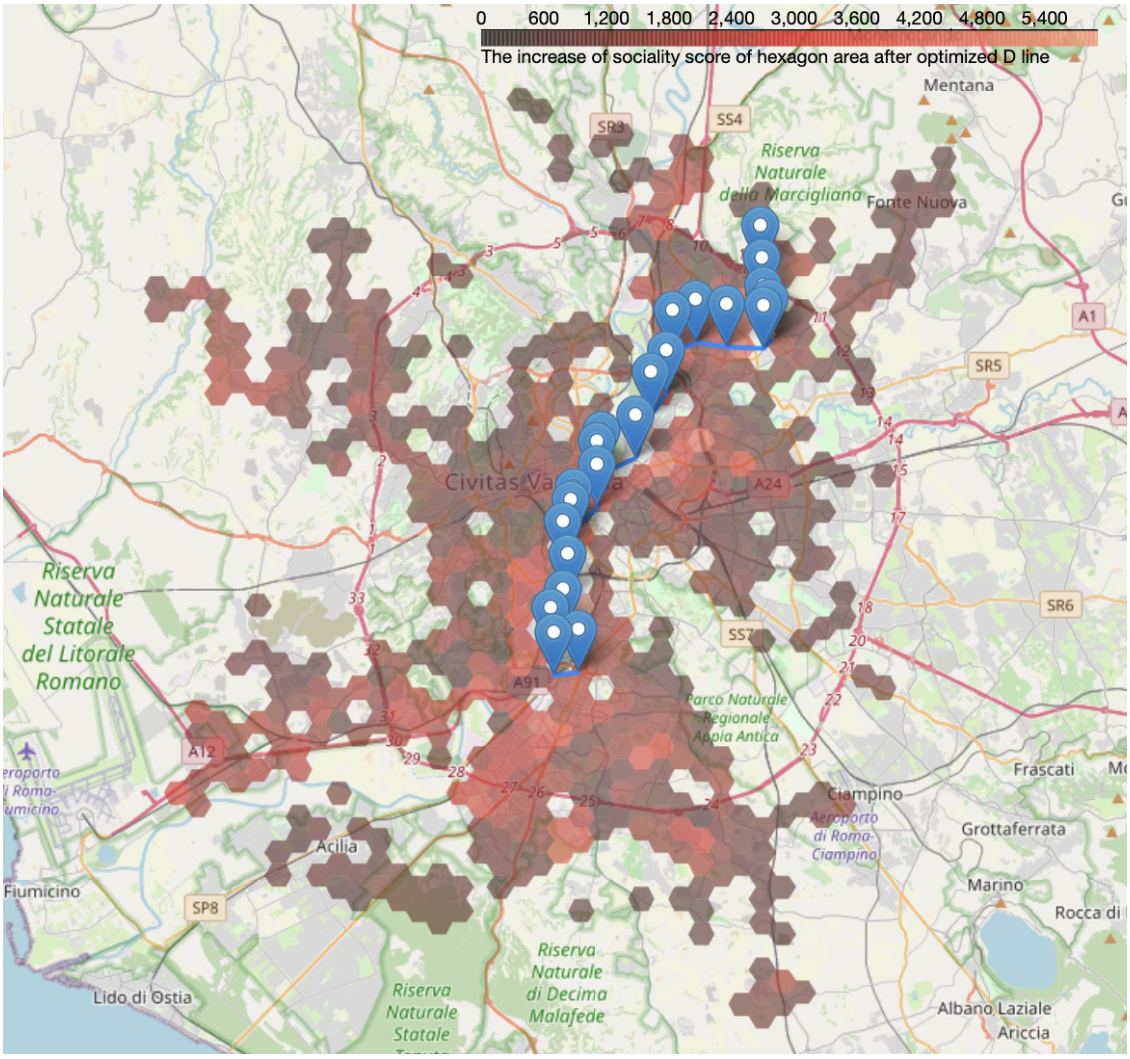


Figure 5.10. Comparison of sociality score after optimized metro D with sociality score of original metro D. It shows improvement of sociality score of hexagon area after optimized metro D

strategy, the calculation time of the whole line will be affected by the total number of stations. We have tried to control the number of iterations and the number of particle population, as described in the previous paragraph, it has been reduced as far as possible within the acceptable range. At the same time, because every particle movement needs to calculate the whole city PTN, reducing the time-consuming of PTN calculation is also an important method to improve the efficiency of our output results. If our average calculation time is 7 seconds like it use to be (See 2.3), the total calculation time will be as high as 163 hours. So the follow-up optimization point of this study can be in the calculation optimization of the fractional model.

improved PSO	sociality score	percentage improvement	time cost
original Line D	12173.27	-	-
w-PSO	12883.57	5.8%	94 hours
lr-PSO	13155.25	8.0%	94 hours

Table 5.2. Comparison between improved PSO and original metro D (take the best score of each experiment)

5.2.3 Robustness

In order to verify the stability of the optimization results of lr-PSO, we conducted five experiments. The highest and lowest scores were 13155.25 and 13072.60, with an average of 13112.85 and a standard deviation of 29.458404. lr-PSO shows good stability, which is what we need. Stable optimization results are more worthy of adoption and application.

Chapter 6

Discussion of results and conclusions

6.1 Discussion

In this work we aim as solving the problem of city line planning, in order to select better public transport lines and stops locations. We use improved PSO and greedy strategy to select the location of the stops. This paper compares the advantages and disadvantages of a variety of greedy strategies to solve the problem of Roma metro line planning. It also proposes and verifies the method of using lr-PSO to replace the original PSO to optimize the line, which greatly improves the sociality score of urban PTN.

After tried a variety of greedy strategies, we choose to use the complete line plan as the basis, and use greedy method and improved PSO to optimize the whole line. At the same time, we compare the optimization ability of a variety of improved PSO algorithms with the same time complexity and code execution efficiency. We focus to compare the results of optimization and the discreteness of result. Finally, we propose a scheme to optimize Rome metro line D by lr-PSO algorithm, and the optimization result is 8.0% on the sociality score of urban PTN.

The lr-PSO method has good robustness with the problem of urban line optimization, and it shows a small difference in many experimental results. Moreover, the stronger activeness of particles in lr-PSO can make the algorithm not easy to fall into local optimum in the case of a small number of iterations and partical population. The algorithm can cooperate with greedy strategy to find a better location for each station in their respective search area.

Looking forward to the future, although our lr-PSO algorithm can optimize the route of a large city like Rome in an acceptable time and give the scheme with the highest sociality score, as mentioned in Chapter 5, it takes 93 hours to optimize the route of a 22 station, which can be normally implemented on a reliable server, But less time is what we need. Therefore, one of the future directions of this study is to optimize the scoring model, so that it can complete the optimization of the whole line in a shorter time. In addition, the score model used in this paper is based on the walking data of openStreetMap data[38] and the bus timetable and route data. However, the setting of the stop is completely dependent on the latitude and longitude, and does not take into account the building and terrain or river factors. Which means in line planning work, the improved PSO and greedy strategy proposed in this paper are used to optimize the line. Then the proposed scheme needs to be combined with the field survey of architectural engineers, and continue to adjust the line position after the survey. Therefore, considering these factors, the second work in the future is to limit the location of the station according to the spatial attributes of the map, such as avoiding the setting of stations on rivers and existing buildings, or choosing the intersection or both sides of the road to set stations and other ways to further optimize the algorithm.

Future work	**Possible solution**
Scoring model optimization	Looking for ways to speed up the calculation of city scores
Limit the location of thestation	Building and terrain data are introduced to limit the location of the station and find a common way to limit the stop location. But in this way, the line planning problem may be transformed into a discrete problem.

Table 6.1. Future work and possible solutions

Bibliography

[1] Ching T Y , Ferreira J J , *Smart cities : concepts, perceptions and lessons for planners.* Springer International Publishing p.145, 2015.

[2] TransitFeeds – Public transit feeds from around the world. Accessed on November 2017. See https://transitfeeds.com/.

[3] Mappa-metropolitana-roma. see https://blog.rome-accommodation.net/atac-guide-to-rome-public-transportation-network

[4] M. Cocca, D. Giordano, M. Mellia, L. Vassio. *Free floating electric car sharing design: Data driven optimisation.* Pervasive and Mobile Computing, Elsevier, 55, pp. 59-75, 2019. DOI: 10.1016/j.pmcj.2019.02.007

[5] M. Cocca, D. Giordano, M. Mellia, L. Vassio. *Data Driven Optimization of Charging Station Placement for EV Free Floating Car Sharing.* In: 2018 21st International Conference on Intelligent Transportation Systems (ITSC), pp. 2490-2495, 2018. DOI: 10.1109/ITSC.2018.8569256

[6] Kennedy, J., Everhart, R., in "Proc. of IEEE international Conference on Neural Networks (ICNN)", *Strumenti Moderni*, 1942–1948 (1995).

[7] E. Torricelli e A. Vasari, in "Proc. of IEEE international Conference on Neural Networks (ICNN)", *Particle Swarm Optimization*, vol. III, n. 2 (feb. 1607), p. 27–31.

[8] Duane J.T., *Learning Curve Approach To Reliability Monitoring*, IEEE Transactions on Aerospace, Vol. 2, pp. 563-566, 1964

[9] Schöbel, Anita. *Optimization in Public Transportation[M]. Springer*, 2007.

[10] M.R. Bussieck, P. Kreuzer, and U.T. Zimmermann. *Optimal lines for railway systems. European Journal of Operational Research*, 96(1):54– 63, 1996.

[11] M.R. Bussieck. *Optimal lines in public transport.* PhD thesis, Technische Universit¨at Braunschweig, 1997.

[12] M.T. Claessens, N.M. van Dijk, and P.J. Zwaneveld. *Cost optimal allocation of rail passenger lines. European Journal on Operational Research*,

110, 1996.
[13] J. Goossens, C.P.M. van Hoesel, and L.G. Kroon. *A branch and cut approach for solving line planning problems.* Transportation Science, 38:379–393, 2004.
[14] J. Goossens, C.P.M. van Hoesel, and L.G. Kroon. *On solving multi-type line planning problems. Technical Report RM/02/009*, University of Maastricht, 2002. METEOR Research Memorandum.
[15] A. Sch¨obel and S. Scholl.*Line planning with minimal transfers.* In proceedings of ATMOS 2005, 2005.
[16] S. Scholl. *Customer-oriented line planning.* PhD thesis, Technische Universit¨at Kaiserslautern, 2005.
[17] R. Bornd¨orfer, M. Gr¨otschel, and M. E. Pfetsch. *Models for line planning in public transport.* Technical Report 04-10, ZIP Berlin, 2004.
[18] R. Bornd¨orfer, M. Gr¨otschel, and M. E. Pfetsch. *A path-based model for line planning in public transport.* Technical Report 05-18, ZIP Berlin, 2004.
[19] R.-S. Schneider. *A new customer-oriented cost model for the line planning problem.* Master's thesis, Technische Universit¨at Kaiserslautern,
[20] *Particle swarm optimization* see https://en.wikipedia.org/wiki/Particle_swarm_optimization.
[21] I. Biazzo, B. Monechi, V. Loreto *General scores for accessibility and inequality measures in urban areas.* 2018.
[22] G. Laporte, A. Marin, J.A. Mesa, and F.A. Ortega. *An integrated methodology for rapid transit network design. In Algorithmic Methods for Railway Optimization*, Lecture Notes in Computer Science. Springer, 2006. presented at ATMOS 2004, to appear.
[23] G. Laporte, J.A. Mesa, and F.A. Ortega.*Maximizing trip coverage in the location of a single rapid transit alignment.* Annals of Operations Research, 136:49–63, 2005.
[24] J. Goossens.*Models and algorithms for railway line planning problems.* PhD thesis, University of Maastricht, 2004.
[25] Geurs KT, Van Wee B. *Accessibility evaluation of land-use and transport strategies: review and research directions.* J. Transp. Geogr. 12, 127–140. 2004
[26] Galton F. *On the construction of isochronic passage-charts.* In Proc. of the Royal Geographical Society and Monthly Record of Geography, vol. 3. JSTOR, pp. 657–658.
[27] Banavar JR, Maritan A, Rinaldo A. *Size and form in efficient transportation networks.* Nature 399, 130–132. 1999

[28] Louf R, Roth C, Barthelemy M. *Scaling in transportation networks.* PLoS ONE 9, e102007. 2014

[29] Masucci AP, Arcaute E, Hatna E, Stanilov K, Batty M. *On the problem of boundaries and scaling for urban street networks.* J. R. Soc. Interface 12, 20150763. 2015

[30] Arcaute E, Molinero C, Hatna E, Murcio R *Cities and regions in Britain through hierarchical percolation.* R. Soc. open sci. 3, 150691. 2016

[31] *Eurostat population grid.* See http://ec.europa.eu/eurostat/web/gisco/geodata/reference-data/population-distribution-demography/geostat.

[32] I. Biazzo, B. Monechi, V. Loreto *General scores for accessibility and inequality measures in urban areas.* 2018. p.5 figure 1

[33] I. Biazzo *metroline_generator* See https://github.com/ocadni/metrolin_generator/blob/master/metro_gen.ipynb

[34] Li R, Dong L, Zhang J, Wang X, Wang WX, Di Z, Stanley HE. *Simple spatial scaling rules behind complex cities.* Nat. Commun. 8. 2017

[35] Schmöcker JD, Quddus MA, Noland RB, Bell MG. *Mode choice of older and disabled people: a case study of shopping trips in London.* J. Transp. Geogr.16, 257–267. 2008

[36] Miller HJ. *Measuring space-time accessibility benefits within transportation networks: basic theory and computational procedures.* Geogr. Anal. 31, 1–26. 1999

[37] Luxen D, Vetter C. *2011 Real-time routing with OpenStreetMap data.* In Proc. of the 19th ACM SIGSPATIAL Int. Conf. on Advances in Geographic Information Systems. GIS '11, pp. 513–516.

[38] *OpenStreetMap.* See https://www.openstreetmap.org/.

[39] I. Biazzo .*Citychrone.* See http://citychrone.org/city/Rome.

[40] *Eurostat population grid.* See http://ec.europa.eu/eurostat/web/gisco/geodata/reference-data/population-distribution-demography/geostat.

[41] *Center for International Earth Science Information Network CIESIN Columbia University C. Gridded Population of the World, Version 4 (GPWv4): Population Count. Palisades, NY: NASA Socioeconomic Data and Applications Center (SEDAC).* 2016

[42] Dibbelt J, Pajor T, Strasser B, Wagner D. *Intriguingly simple and fast transit routing.*In Int. Symp. on Experimental Algorithms, pp. 43–54. 2013

[43] Juang C F . *A hybrid of genetic algorithm and particle swarm optimization for recurrent network design, IEEE Transactions on Systems Man and Cybernetics Part B Cybernetics A Publication of the IEEE Systems*

Man and Cybernetics Society, 2004, 34(2):997-1006.

[44] Suraj P , Wu L , Guru S , et al. A Particle Swarm Optimization (PSO)-based Heuristic for Scheduling Workflow Applications in Cloud Computing Environments. , 2012

[45] Wen X , Huang M , Shi J . *Study on Resources Scheduling Based on ACO Allgorithm and PSO Algorithm in Cloud Computing, International Symposium on Distributed Computing and Applications to Business. IEEE, 2012.*

[46] E. Ampellio, L. Vassio. A hybrid swarm-based algorithm for single-objective optimization problems involving high-cost analyses. *Swarm Intelligence, Springer, 10(2), pp. 1-23, 2016. DOI: 10.1007/s11721-016-0121-6*

[47] E. Ampellio, F. Bertini, A. Ferrero, F. Larocca, L. Vassio. Turbomachinery design by a swarm-based optimization method coupled with a CFD solver. *Advances in Aircraft and Spacecraft Science, Techno-Press, 3(2), pp. 149-170, 2016. DOI: 10.12989/aas.2016.3.2.149*

[48] E. Ampellio, L. Vassio. A hybrid ABC for expensive optimizations: CEC 2016 competition benchmark. *In: 2016 IEEE Congress on Evolutionary Computation, Vancouver, Canada, pp. 1157-1164, 2016. DOI: 10.1109/CEC.2016.7743918*

[49] Wu X, Yang Y W, Han S, Zhao Z L, et al. Multi-objective optimization method for nuclear reactor radiation shielding design based on PSO algorithm. Annals of Nuclear Energy. 160. 10.1016/j.anucene.2021.108404. 2021

[50] Song, X. , C. Yang , and C. Chang *A Hybrid Algorithm of PSO and SA for Solving JSP.* International Conference on Fuzzy Systems and Knowledge Discovery IEEE, 2008.

[51] Shi X H , Lu Y H , Zhou C G , et al. *Hybrid evolutionary algorithms based on PSO and GA.* Congress on Evolutionary Computation. IEEE,2003:2393-2399. 2003

[52] Hou P D, Dong D C, Zhou S Z *Effects of Random Values for Particle Swarm Optimization Algorithm.* 11(2):23. https://doi.org/10.3390/a11020023. 2018

[53] *NP-complete problems* https://people.eecs.berkeley.edu/~vazirani/algorithms/chap8.pdf.

[54] Christensen T V , Fischer P . *Heuristic Algorithms for NP-Complete Problems.* Bachelor of Engineering Thesis, 2007.

[55] J. Goossens.*ACKLEY FUNCTION* http://www.sfu.ca/~ssurjano/ackley.html.