Genetic Sun Shadow Positioning Model for Digital Forensics

by

Yanwen Zhou

A thesis submitted in partial fulfillment
of the requirements for the degree of

Masters of Science

in

Computer Science

University of Ontario Institute of Technology

Supervisor: Dr. Xiaodong Lin

November 2017

Abstract

Extensively growing a set of methodologies to collect, provide, and analyze geo-information in this decade. While due to the privacy concerns or elevated consideration, only less than half of videos and pictures bring geographical tags. Meanwhile, such geographical indications are not always available and reliable because of device dependencies. For example, users forbid location functions on devices or manipulate published geo-tags. In this case, the straightforward images' data for digital forensics will lose value. In this thesis, we propose an approach for shadow positioning model to smooth over this obstacle, such as non-geo-tagged information in videos and photos. Firstly, we briefly summarize existing digital forensics researchers rely on geotagging and figure out the potential limitations and threats for compromising the geo-information from videos and photos; then present a positioning model and algorithms are based on Solar Position Algorithms (SPA) and Genetic Algorithm (GA) to estimate geo-coordinates for non-geo-tagged source files. The experimental results show that our proposed model and algorithm can successfully compute a set of latitude and longitude values, and the average error in $\pm 0.2^{\circ}$ for both latitude and longitude. These combined algorithms derive from the theory of astronomy and evolution, which shapes a novel way to obtain geolocation information from such consequences.

Acknowledgements

I would like to express my appreciation to my supervisor Dr. Xiaodong Lin and co-supervisor Dr. Ying Zhu for their grate helpful. During my two year's graduate study and research in the Master's degree program, Dr. Xiaodong Lin has always been patient and encouraging, and Dr. Ying Zhu has been helpful as well. Furthermore, I would like to thanks all peers in our lab for their kind supports. In the end, I an grateful to my parents for their love.

Yanwen Zhou

Oshawa, Ontario

September 2017

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Background

Digital forensics is a relatively new research area which aims at authenticating digital media by detecting digital evidence. With the digital ecosystem at scale, a large amount of digital evidence is generated. The main digital evidence includes computer evidence, digital audio, digital video, cell phones, digital fax machines, etc. Photos and videos provide a wealth of latent information for digital forensics as they were when and where taken. The obvious scenario that is presented in photos and videos are the most direct information, while they also contain the time, a geographical indication (GI) and other indirect information. In reality, we often see kidnap and threaten videos or photos of terrorists in news. In such case, the more indirect digital evidence can be found in photos or videos, then the more valuable supporting information is gathered for digital forensics. Therefore, the weight of these indirect information cannot be underestimated in digital forensic process.

Among all indirect information, GI is one of the most useful for qualitative and quantitative empirical research in many domains. For instance, in 2012, John McAfee

was arrested after his location in Guatemala was revealed when a post from Vice Magazine, which had reporters following McAfee on his trip, included a photo with metadata attached. In the past, scientific endeavors in fields like social computing and computer vision have generally focused on devices' data , GPS or WLAN data to improve the accuracy of location estimation. The most common way is GPS, which uses cell-ids or signal strength measurements of mobile networks or uses the locations of WLAN hotspots in the mobile's neighborhoods [41].

Likewise, GI may automatically be tagged by current digital devices. For example, the accuracy of geographical coordinates for all photos and videos taken by Apple's iPhone with the internal camera even exceeds GPS [30]. As mobile devices determine their position in combination with cell-tower, which regularly reaches resolutions of +/- $1m$ even indoors [24]. Although such positioning approaches can undertake the high accuracy, the data acquisition of them extremely relies on mobile devices and networks.

1.2 Motivation

In 2015, a research for Yahoo Flickr with a total of 100 million media objects, of which approximately 99.2 million are photos and 0.8 million are videos, indicates that about half of them are geotagged photos and videos [61]. In other words, another half of photos and videos are non-geotagged and cannot provide GI evidence for digital forensics. Similarly, criminals are becoming more aware of digital forensic and investigation capabilities, and some of them even developing 'anti-forensic' methods and tools. These tools are specifically designed to conceal their activities or destroy digital evidence, which generally undermines digital investigators. In 2009, FBI was investigating a security branch for seized computers during military operations in Iraq

and they found most data of military computers has been destroyed [12].

Apparently, both ordinary users and criminals have realized that sharing location information has implications for their privacy. In many cases, position sharing functions on devices are disabled. Take a special circumstance for an example, when committing a crime, GI will definitely be disabled by recorders. However, whether publishing geo-location is only one part of the problem. The more crucial problem is to deliberately forge geo-tags in video and photos. Consequently, the critical GI information for digital forensics are vanished or forged, which would cause a huge barrier to reconnaissance.

The limitation of existing methods to positioning demonstrate in three related aspects: (i) the availability of large-scale easy-to-use location-based data rely on the device functionalities, such as built-in geo-tags function, which subjects to the setting of users; (ii) correlation of findings across diverse independent sources to extract geographical positioning leads to analysis cannot be completed in an instant; (iii) images and videos that make even a small relative percentage of location data, such as the incidence angle of the sun, are ignored but sufficient for mounting deep analysis.

Beyond what is obviously captured in photos and videos, deeper insight, hidden evidence of events are also represented in them. Although there are many kinds of metadata in photos and videos, changes of shadow length and direction causes by changes of the sun's incidence angle in shot period are one of them. From this perspective, according to sun astronomical position at the specific time, the location of where photos and videos are taken can be also estimated in an appropriate way. Since geographical position for latitude and longitude are in limited spaces, a fast global search method can be utilized to fulfill this idea.

1.3 Contributions

Take these limitations into account, we propose a positioning model and algorithms to estimate objects' geographical position (latitude/longitude) by its shadow length and direction. The methodology is based on Solar Position Algorithms (SPA) [50] and Genetic Algorithm (GA) [37].

To achieve our idea inevitably encountered some challenges. The most primary challenge is how to build a relationship between the sun position and shadow on the earth surface. As we mentioned that the incidence angle (elevation and azimuth) of sunlight are varying at different time and spot. According to this clue, we find SPA is the most reliable algorithm that is even used by NASA for positioning the location of the earth and the sun. The angle of incidence includes elevation angle and azimuth angle, can be obtained by SPA at every specific time and spot.

Another challenge to solve this problem is the searching speed in the latitude and longitude search spaces. To deal with this challenge, we settle our thinking on heuristic search. After our literature review, we lock in the method of Genetic Algorithm. GA is a kind of meta-heuristic search approach that mimics natural biological evolution. It is designed for solving a problem more efficiently when classic methods are slow. As GA falls into the category of guided random search, the optimal solution for each round is not always the same. In terms of this concern, we seek an optimization method to reduce the feasible solutions space and improve the accuracy of estimation model. The main contributions that we try to resolve limitations in existing methods by five main aspects:

(i) To get rid of devices' data dependencies, we propose a practical solution to position an object geographically only utilizing the object's shadow length and direction in non-geotagged photos and videos resource.

(ii) We build a shadow vertex coordinate model to describe the relationship between an object's geographical position (latitude/longitude) and its shadow extracted from images, which is based on a high-precision Solar Position Algorithm (SPA) provided by NASA.

(iii) Several cost functions are defined to improve the accuracy of optimization solutions for the sun shadow positing model, including $\ell^p(p = 1,\ 2)$ norm-based metric of shadow coordinate change and ℓ^1 norm-based ratio change of solar elevation angle difference between two adjacent moments.

(iv) A Genetic Shadow Positioning Algorithm (GSPA) is designed to solve the sun shadow positing model and calculate the object's geographical latitude and longitude. To further optimize feasible solutions space, a NSGA-II-based Genetic Shadow Positioning Algorithm (NGSPA) is proposed, since it considers the properties of fast non-dominated sorting, density estimation, crowded-comparison and elitist-preserving strategy.

(v) The feasibility and accuracy of the proposed algorithms - GSPA and NGSPA are proved through extensive simulation. The experimental results show that the average error in $\pm 0.2^\circ$ for both latitude and longitude.

1.4 Thesis Outline

In this thesis, we begin with a brief review of existing research of positioning and fast global search methods in chapter 2. On the basis of reviewed literatures, we proposed our shadow positioning model and algorithms by two stages in chapter 3. In chapter 4, we examine the feasibility of proposed approaches, and compare the feasible solutions space for each cost function as well as multiobjective functions. Finally, besides future work, our objectives and contributions are concluded and restated in the last chapter.

Chapter 2

Literature Review

In this chapter, the work on existing researches for digital positioning is reviewed in the first part. In order to find a proper global search method to fulfill our idea, we review some of main search techniques following section 2.2. Further, considering to find an optimal solution, multi-objective optimization is also reviewed in the final part of this chapter.

2.1 Digital Forensics and Positioning Model

Existing approaches to extract goedata are quite similar, the first step is to extract as much data as possible from the device and its storage media, then through forensic tools to find the relevant data in the memory image [1] [41] [30] [43]. Depending on storage medium and type of geodata, the approach can be different, but it be expected to find a sufficient amount of geodata in an acquired mobile device. Carrier et al. [11] processed an event based model for digital investigations is defined using the techniques from the physical investigation world. The physical examples for this phase include deploying video cameras to record who was in the area at the time of

the crime, then sending and synchronizing the internal clocks on servers with NTP to servers. This idea is poorly suited to finding information that is out-of-the-ordinary, out-of-place, or subtly modified.

There are also some studies trying to get rid of the dependency on the device. In [4], they proposed an enhanced digital investigation process model to involve taking photographs, sketches, and videos of the crime scene. Some other researches try to find marks, fingerprints or other information are hidden in some video [13], [48]. In another research [51], it identifies and extracts of data hidden using steganography, speech recognition technology to analyze voice messages, and generation of summaries for digital video files using keyframe extraction. However, these methods are designed to help examiners collect specific pieces of evidence, but can not carry out positioning for digital forensics.

Currently, only a few researches involve into the sun shadow geotags digital forensics investigations. A method estimates the sun elevation angle based on geometric constraints the scene provides, whose consistency serves as the cue to authenticate image forgery by Cao et al. [10]. Although the research through sun elevation angle to make reasonable judgment with five sampling points, it based on both known geometric and astronomic relationship. In [34], they utilize a two-stage shadow detection process, which is able to estimate the azimuthal direction of the sun in an image and compare it against a calculated theoretical value. This method provides a novel light-based forensic algorithms, while this geometric-based forensic techniques relying on the analysis of vanishing points and the ultimate goal is not exact geographic information.

2.2 Finite Space Search Methods

The purpose of this part of literatures review is to find a quickly global search method for geographical position (latitude/longitude) in finite spaces.

2.2.1 Classes of Search Techniques

The current literatures identify three main types of search methods [25] [27] [32]: (I) Calculus-based Search; (II) Enumerative Search; (III) Guided Random Search. The traditional methods are branched into calculus-based search and enumerative search, these two techniques are simple but at the same time are time consuming [64]. Guided Random Search techniques as a type of intelligent methods are branched into Evolutionary Algorithms and Simulated Annealing. Evolutionary Algorithm is a computational approach which provides artificial evolution theory. It derives four main methods namely Genetic Algorithm (GA), Evolutionary Strategies (ES), Evolutionary Programming (EP), and Genetic Programming (GP). The diagram of search techniques show in Figure 2.1.

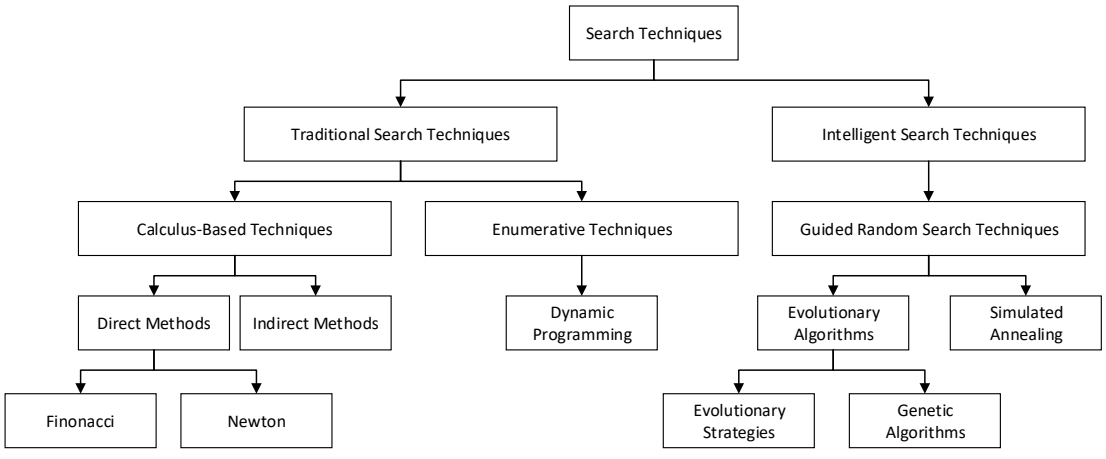

Figure 2.1: Main Types of Searching Techniques

Calculus-based Search

Calculus based methods tend to seek local extreme points, either by searching for points with slopes of zero in all directions (indirect methods), or by moving along a function in a direction with the steepest gradient, i.e. hill-climbing (direct methods). The problem with calculus-based search is that can only be applied to real-valued or continuous-valued functions rather than discrete-valued [33]. It also proved to be more inclined to find local extreme, rather than absolute extreme [59]. This limits these kinds of search techniques to very narrow problem domains .

Enumerative Search

An enumerative search simply looks at all the possible function values one by one - in essence, enumeration. Although they can be useful on very small problem sizes (it is a very human type of search, such as when searching through a short list), its biggest problem is its inefficiency - most search spaces are too large to be searched one item at a time [42].

Guided Random Search

It should be noted that search techniques that use random choice to guide the search are quite different from purely random searches. With randomized algorithms, a randomizer helps to make some decisions in the algorithm, such that transitions are probabilistic rather than deterministic. The result is that performance can differ in an unpredictable fashion between runs. As such, these algorithms fall into two different classes [32]:

- Las Vegas algorithm: always find a solution, but has random execution time
- Monte Carlo algorithm: minimal variation in execution time, but no guarantee

of finding a solution (but may get close)

Strictly random search techniques look at random points in the search space and choose the best elements, while random search founds superior results to those found more quickly within the more restricted search. This tradeoff between exploration and exploitation is central to the design of an effective random search [5]. However, these type of searches will have the same performance as enumerative search spaces in the long run [42].

Meta-heuristics are strategies that *guided* the search process, while heuristics are usually without bound and not guaranteed to find the optimal solution. In other words, meta-heuristics is a master strategy that guides and modifies other heuristics to produce solutions beyond those that are normally generated in a quest for local optimality [45].

2.2.2 Distinction of Evolutionary Algorithms

The common problem with Calculus-based Search and Enumerative Search methods is a general lack of robustness, these methods may perform well in specific circumstances, but it is desirable for a technique that is adaptable and applicable to a wide range of problem environments [64]. It has been identified that GA differ from more traditional techniques in for key ways [25] [27]:

- they work with a coding of the parameter set, not the parameters themselves
- they search from a population of points, not a single point
- they use payoff information from an objective function, not intermediate information, such as derivatives or domain knowledge

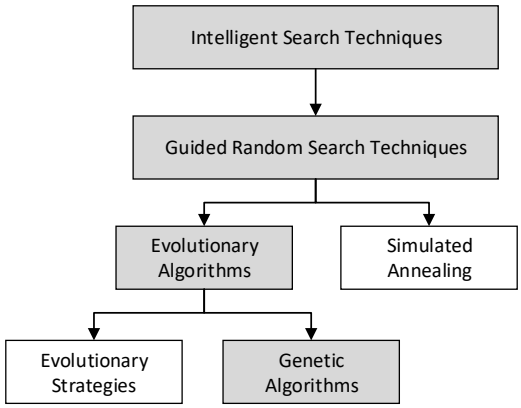

Figure 2.2: Guided Random Search Techniques for Genetic Algorithm

- they use probabilistic and not deterministic transition rules, thus can be considered to be randomized algorithms

The effect of these differences is that intelligent algorithms are an effective approach for general optimization and search. This is especially true if you keep in mind that, for most larger, more complex problems.

2.2.3 Genetic Algorithm (GA)

GA is stochastic global search and optimization methods that mimic the metaphor of natural biological evolution [29]. It's a kind of meta-heuristic search approach that is designed for solving a problem more efficiently when classic methods are too slow, or for finding an approximate solution when classic methods fail to find any exact solution. The essential of GA is the theory of evolution, it is a stochastic iterative process that are not guaranteed to converge. The branch structure of Intelligent search techniques show in Figure 2.2. However, the termination condition may be specified as some fixed, maximal number of generations or as the attainment of an acceptable fitness level for the best individual [45]. O.Kremer [37] introduced the

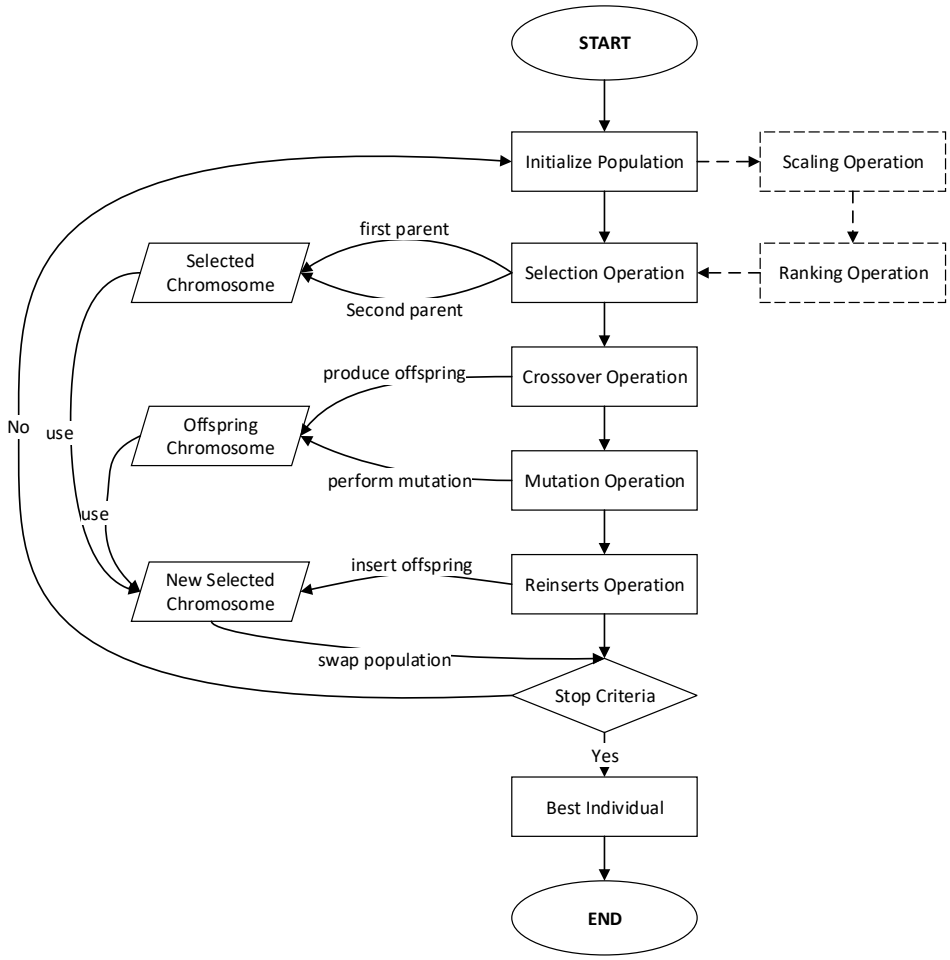

Figure 2.3: Brief Procedures for Genetic Algorithm

basic GA step by step, which are crossover, mutation, and selection. Further,they discussed genotype-phenotype mapping, common termination conditions, and give a short excursus to experimental analysis. For the easy understanding of the GA, we describe the brief procedure of GA in Figure 2.3.

Selection

Some work has been done to conclude differences for various selection strategies, such as proportionate selection, ranking selection, tournament selection, etc. Goldberg and

Deb [26] introduced a novel term of takeover time, which is the number of generations that needed for a single best individual to crossover the whole generation till no recombination is involved. Then Bäck [2] has analyzed the most prominent selection schemes used in Evolutionary Algorithms with respect to their takeover time. While they did not mention truncation selection. In [54] the *selection intensity* is proposed as an import metric to measure the progress in the population, which is derived for proportional selection and truncation selection. Later, authors of [7] made a further analysis derives the selection intensity, selection variance, and the loss of diversity for all selection schemes.

Recombination

Recombination is a procedure loop of the GA to generates new offspring candidates with crossover and mutation until the cover the whole solution space. Crossover operators in GA implements a mechanism that mixes the genetic feature of the parents. Mutation is a genetic operator used to maintain genetic diversity from one generation of a population of genetic algorithm chromosomes to the next generation.

- **Static-state Recombination**

  The most common crossover operator is n-point crossover, which split up a set of features at n position and alternately assembles them to offsprings. This operator can easily be extended to other forms to split up and reassembled alternately. Arithmetic crossover [49], computes the arithmetic mean of all parental features component evenly. Figure 2.4 illustrates a graph for the single crossove. For example, for two parents (2, 3, 4) and (2, 1, 2), the offspring is (2, 2, 3). Dominant crossover [49], successively chooses each features from one of the parental space. Uniform Crossover by Syswerda [60], uses a fix missing ratio like 0.2 to randomly choose bits from either set of the parents features. According

to different crossover methods, Kramer [37] concluded that many GAs simplify crossover step and use crossover operator with uniform distribution.

Mutation is based on random changes, but mutation rate follow by a probability distribution, such as Gaussian mutation. With the mutation rate is arbitrarily scalable, all space of candidates offspring will be reachable. After crossover and mutation, evaluation will subsequently happen. As an optimization solution for evaluation, chromosome can be represented by genotype and phenotype [37]. Although genotype-phenotype mapping is not always required, the significant principle for genotype-phenotype mapping should avoid the bias.

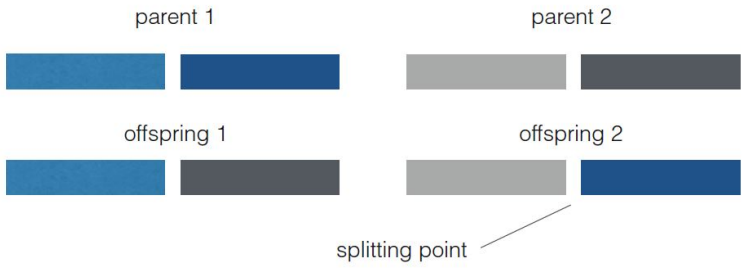

Figure 2.4: crossover splitting point

- **Adapting Recombination**

  The idea of adapting crossover and mutation operators to optimize GA has been employed earlier [22] [53] [62] [65]. Schaffer et al. [53] discussed a crossover mechanism with the distribution of crossover points is adapted based on the performance of the generated offspring, the distribution information is encoded by additional bits into each string. Davis [62] discussed an effective method of adapting operator probabilities, the adaptation mechanism provides for the alteration of probabilities due to the fitness of strings that created by the operators. In both case, these two mechanisms are not related to adapting mutation and crossover rates and the operator probabilities are invariant with the indi-

vidual fitness.

Shortly afterwards, Fogarty [22] has studied the effects of varying the mutation rate over generations, and demonstrated superior performance for a single application for a mutation rate that specifically decreases exponentially with generations, but mutation rates are varied in a predetermined tendency. Whitley's adaptive mutation approach [65], the mutation rate is determined specifically for each possible solution rather than an individual. However, this method only derived from the idea of sustaining the diversity in the population without affecting the convergence properties. In the approach of Srinivas et al. [57], they proposed probabilities of crossover and mutation are determined for each individual instead of an entire string of its fitness, and also concerned with generational replacement. They also demonstrated their experimental results, in same constrains, it shows the high efficacy and the least number of local optimum of their approach compares to other adaptive mechanisms.

Fitness

Fitness computation of individuals or chromosomes is evaluated on a fitness function, which measure the quality of the offspring space that Generic Algorithm has generated. Whitley [66] demonstrated that the best generic mapping of parents and offsprings in case of low fitness values are preferred and vice versa in case of worst problems. To allow convergence towards optimal solution, the best offspring will be selected. The selection process depends on the fitness values in the population. Many selection algorithms are based on randomness. Roulette Wheel selection [66] as a well known fitness proportional selection mechanism, selects parental features randomly with uniform distribution. For this sake, the randomness of the fitness proportional selection allows forgetting of the best solutions. Although genetic selection has been

completed, a success of GA is significantly depends on appropriate parameters. Parameter tuning and control techniques will be repeated during the process until meet the termination condition.

- **Scaling**

  Proportional fitness scaling is an option to set the scaling between quantization points can be used to select either linear or logarithmic decoding to real values from binary strings. Logarithmic scaling is useful when the range of decision variable is unknown at the outset as a wider range of parametric values can be searched with fewer bits [55], thus reducing the memory and computational requirements of the GA.

- **Ranking**

  In rank-based fitness assignment, the population is sorted according to the objective values. The fitness assigned to each individual depends only on its position in the individuals rank and not on the actual objective value. Ranking introduces a uniform scaling across the population and provides a simple and effective way of controlling selective pressure. Selective pressure indicates the probability of the best individual being selected compared to the average probability of selection of all individuals [67]. The reproductive range is limited, so that no individuals generate an excessive number of offspring. However, stagnation in the case where the selective pressure is too small or premature convergence where selection has caused the search to narrow down too quickly [44].

2.3 Multi-objective Optimization with GA

The purpose of optimization is we try to find optimal solution from the feasible space that is generated by GA. Over the past decade, a number of multiobjective

evolutionary algorithms have been suggested [20] [23] [31] [58] [72].

2.3.1 The First Generation Optimization

Multi-objective formulations are realistic models for many complex engineering optimization problems, and optimizing a particular solution may be better for a particular target and may be poor for other goals, so there is a set of compromises called Pareto-optimal set or Nondominated set [20]. At first, the multi-objective optimization problem is often transformed into a single-objective problem by weighting, and then solved by mathematical programming methods [36] [20]. However, only one optimal solution can be obtained with a specific weight value. At the same time, the traditional mathematical programming methods are often less efficient due to the objective function and the constraint function of the multi-objective optimization problem, which are non-linear, non-differential or discontinuous, and they are more sensitive to the order of weights or targets given.

2.3.2 The Second Generation Optimization

In order to solve the drawbacks of mathematical programming methods that the second generation evolutionary multi-objective optimization algorithm, characterized by the elite retention mechanism have been porposed. In 1999, Zitzler and Thiele created the Strength Pareto Evolutionary Algorithm (SPEA) [73], and three years later, they proposed SPEA2 [71]. In 2000, Knowles and Corne proposed the Pareto Archived Evolution Strategy (PAES) [35], and soon they also proposed an improved version of Pareto Envelope-Based Selection Algorithm (PESA) [16] and PESA-II [15]. In 2001, Erichson, Mayer and Horn proposed an improved version NPGA2 of NPGA [erickson2001niched]; Coello and Pulido proposed Micro-Genetic Algorithm (Micro-

GA) [14]. Deb and others through the NSGA to improve, put forward a very classic algorithm: NSGA-II [21].

2.3.3 The Third Generation Optimization

Some scholars have proposed the third generation algorithms in recent years, new dominance mechanisms such as Laumanns and Deb [40] proposed ε-dominance, Brockoff and Zitzler [9] proposed partial dominance, and Alfredo and Coello et al. proposed Pareto adaptive ε-dominance [28]. Although there are many variations of multi-objective GA in cited literatures, these cited GA are well-known and credible algorithms that have been used in many applications and their performances were tested in several comparative studies. Abdullah et al. [36] experimentally compared the performance of NSGA-II [21], SPEA2 [71], PESA-II [15] and NNIA [25].

Overall, NSGA-II is recognized as one of the best evolutionary multi-objective optimization algorithms to date. Meanwhile, in the survey paper by Zitzler et al. [70], it is aimed at introducing the components of multi-objective GA to researchers and practitioners without a background on the multi-objective GA. However, nearly all problems will require some customization of the GA approaches to properly handle the objectives, constraints, encodings and scale.

Chapter 3

Methodology

In this chapter, we introduce the astronomical terminology and Solar Position Algorithm(SPA). Then we propose the shadow vertex coordinate model to describe the relationship between shadow and sunlight. To fulfill rapidly and accurately finite space search, we discuss each operator of Genetic Algorithm (GA) in next section. Due to feasible solutions of GA in a space, not the unique solution, we introduce an optimization method NSGA-II in the last section of this chapter.

3.1 Solar Position and Shadow Model

In this section, after introducing the astronomical terminology, we establish a model to analyze the variation of shadow length and direction, then find the relationship of each impact factor in this solar portion and shadow model. Assuming the length of the object is known, the shadow length can be calculated by the principle of a simple trigonometric function on the earth surface. Meanwhile, according to the basic astronomical knowledge, shadow length and direction are effected by solar elevation angle (e) and azimuth angle (Γ), which is mainly affected by local date, time, longitude

and latitude. At this point, the solar location astronomical algorithm - SPA [50] can be used to analyze the impact of each variable on shadow variation.

3.1.1 Astronomical Terminology and Symbol

For the easy understanding of the astronomical terminology and theory parts, we list some important notions in Table 3.1, and illustrate the geometric graph for the relationship of sunlight and earth in Figure 3.1.

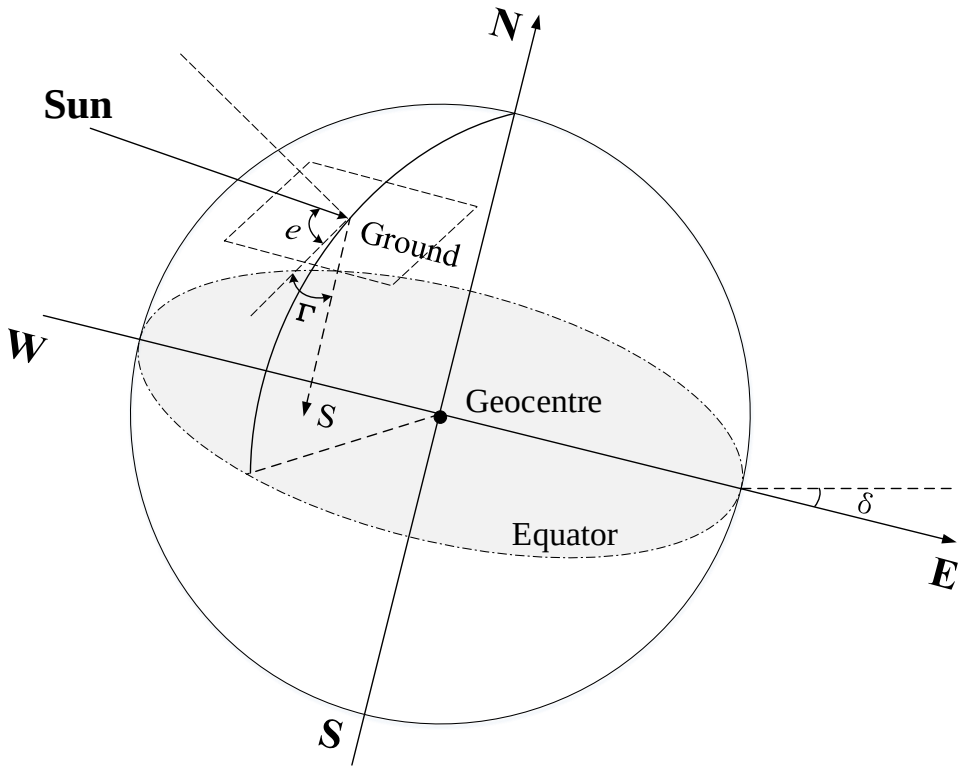

Figure 3.1: Sun Position and Angles Geometric Graph

Solar Elevation Angle: the altitude of the Sun, the angle between the horizon and the centre of the solar's disc.

Solar Azimuth Angle: the position of the sun, the angle between a line due south

Astronomy Terminology	**Symbol**
Observer Solar Declination	δ
Topocentric Solar Declination	δ'
Geocentric Solar Right Ascension	α
Observer Solar Hour Angle	H
Topocentric Solar Hour Angle	H'
Topocentric Solar Zenith Angle	θ
Topocentric Solar Elevation Angle (non-correction)	e_0
Topocentric Elevation Angle (refraction correction)	Δe
Topocentric Elevation Angle (corrected)	e
Topocentric Solar Azimuth Angle (E from N)	Φ
Topocentric Solar Azimuth Angle (W from S)	Γ
Observer Geocentric Longitude	σ
Geocentric Longitude	Θ
Observer Geocentric Latitude	φ
Geocentric Latitude	β
Local Date	d
Local Time	t
Object Horizon Length	l
Object Shadow Length	s
Local Coordinate (West)	x
Local coordinate (South)	y

Table 3.1: Solar Portion Symbol and Description

and the shadow cast by a vertical rod on Earth.

Solar Declination Angle: the incline of the sun, the angle between the equator and a line drawn from the centre of the Earth to the centre of the sun.

Solar Hour Angle: the angle between two planes: one containing the Earth's axis and the zenith (the meridian plane), and the other containing the Earth's axis and the given point (the hour circle passing through the point).

Astronomy of Position Functions Theocratically, the trajectory of the sun and the sunlight position to the earth are roughly determined by the latitude and longitude of observation point and the local time. In practical, location elevation, temperature, and air pressure also impact on its trajectory and sunlight position, especially the

solar elevation angle will be due to atmospheric refraction slightly larger than the theocratical value. The SPA algorithm is provided by NASA, which is the high-precision global positioning system to meet the requirements of the earth at any points on the geography. It also provides correction parameters to ensure that calculating system accuracy, reliability and real-time.

Observer Solar Declination Angle (in degrees)

$$\delta = Arcsin(\sin\beta \cdot \cos\varepsilon + \cos\beta \cdot \sin\varepsilon \cdot \sin\lambda) \tag{3.1}$$

where δ is positive or negative if the sun is north or south of the celestial equator respectively. λ is apparent sun longitude and ε is obliquity of the ecliptic, which is stated and formulated in SPA algorithm.

Solar Right Ascension (in degrees) [50]

$$\alpha = Arctan2(\frac{\sin\lambda \cdot \cos\varepsilon - \tan\beta \cdot \sin\varepsilon}{\cos\lambda}) \tag{3.2}$$

where Arctan2 is an arctangent function that is applied to the numerator and the denominator, instead of the actual division to maintain the correct quadrant of the α, where $\alpha \in [-\pi, \pi]$.

Topocentric Solar Declination Angle (in degrees) [50]

$$\delta' = Arctan2(\frac{(\sin\delta - \rho\sin\varphi \cdot \sin\xi) \cdot \sin\Delta\alpha}{\cos\delta - \rho\cos\varphi \cdot \sin\xi \cdot \cos H}) \tag{3.3}$$

where ρ is the observer's distance to the center of the Earth, ξ is the equatorial horizontal parallax of the sun, and $\Delta\alpha$ is the parallax is the sun right ascension, which is stated and formulated in SPA algorithm [50].

Observer Solar Hour Angle (in degrees) [50]

$$H = \nu + \sigma - \alpha \tag{3.4}$$

where observer's geocentric longitude σ is positive or negative for east or west of Greenwich respectively, and ν is the apparent sidereal time at Greenwich at any given time, which is stated and formulated in SPA. Limit H to the range form 0° to 360° and note that it is measured westward from south in SPA algorithm [50].

Topocentric Solar Hour Angle (in degrees) [50]

$$H' = H - \Delta\alpha \tag{3.5}$$

Topocentric Solar Zenith Angle (in degrees) [50]

$$\begin{cases} e_0 = Arcsin(\sin\varphi \cdot \sin\delta' + \cos\varphi \cdot \cos\delta' \cos H') \\ e = e_0 + \Delta e \\ \theta = 90^\circ - e \end{cases} \tag{3.6}$$

where,

- e_0 is the topocentric elevation angle without atmospheric refraction correction, which calculates the tangent argument in degrees.
- Δe is the atmospheric refraction correction and calculated by the annual average local pressure (in millibar) and temperature (in $^\circ C$), which is stated and formulated in SPA algorithm [50].
- $\Delta e = 0$ when the sun is below the horizon.

Topocentric Solar Azimuth Angle (in degree) [50]

$$\begin{cases} \Gamma = Arctan2(\dfrac{\sin H'}{\cos H' \cdot \sin \varphi - \tan \delta' \cdot \cos \varphi}) \\ \Phi = 180^\circ + \Gamma \end{cases} \tag{3.7}$$

where,

- Γ is the topocentric astronomers azimuth angle, which is limited to the rage from 0° to 360° and noted that it is measured Γ westward from south [50].

- Φ is limited to the range from 0° to 360° and noted that it is measured eastward from north [50].

3.1.2 Shadow Vertex Coordinate Model

In this model, the corrected topocentric elevation angle e is the project angle of sunlight, and the topocentric azimuth angle Γ (westward from south) is the trigonometric angle on the earth surface to calculate shadow vertex coordinate x_i and y_i. Figure 3.2 illustrates the relationship of angles and coordinates. For any point on the geography, set its observer latitude and longitude as φ and σ respectively. The vertex length of the object is l, the shadow length is s, and the shadow vertex coordinate is (x_i, y_i). Based upon, we can establish the following vertex coordinate date model.

Object Shadow Length

$$s_i = \frac{l}{\tan e_i} \tag{3.8}$$

Local Ground Coordinate (West)

$$x_i = s_i \cdot \sin \Gamma_i \tag{3.9}$$

Local Ground Coordinate (South)

$$y_i = s_i \cdot \cos \Gamma_i \tag{3.10}$$

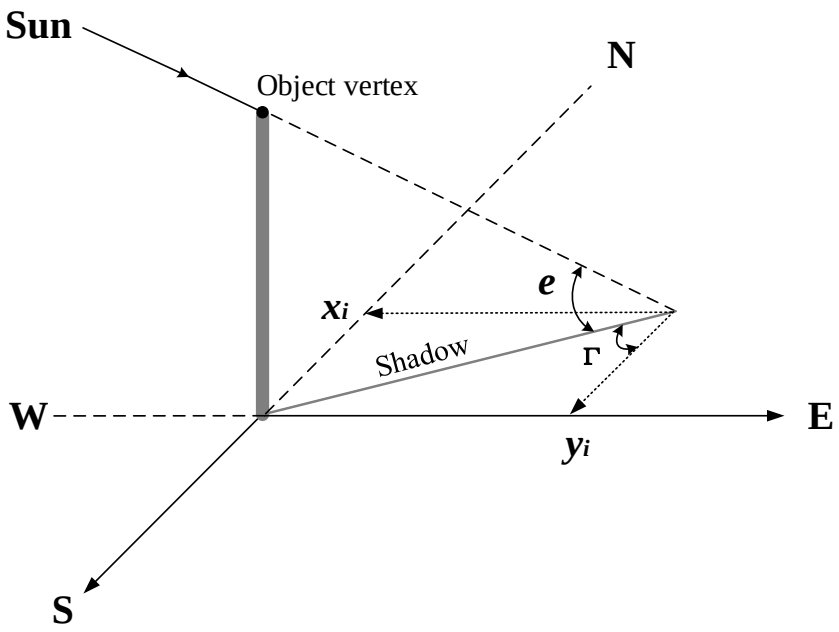

Figure 3.2: Object Vertex Shadow and Coordinates Geometric Graph

3.2 Feasible Cost Functions

In order to solve a group of complex objective functions in GAs to identify the possible geographical location by objects' shadow data, such as shadow vertex coordinates. Several kinds of objective optimization model with the minimum rate of differences can be construct. The cost functions by optimizing differences of solar azimuth angles for shadow length, or optimizing differences between observed shadow length value and estimate value in shadow and position model. Since it is not certain which cost equation can exhibit better granularity, we will estimate them by several metrics in

Chapter 4.

Cost functions can be established through known shadow vertex coordinates (x_i, y_i): Cost Function (I): shadow coordinates change for Manhattan metric (ℓ^1 norm); Cost Function (II): shadow coordinates change for Euclidean metric (ℓ^2 norm). Just as Boyd [8] expounded, the important property is that every norm is a convex function, thus ℓ^1 norm and ℓ^2 norm are feasibly blended to fitness function in Genetic algorithm; Cost Function (III): ℓ^1 metric for ratio of elevation angle with two adjacent measuring moment.

Cost Function (I):

The Manhattan metric was introduced by Hermann Minkowski, and can also be noted as ℓ^1 norm. There is a very popular explanation for Manhattan distance metric, a taxi driver in Manhattan needs to sum up the number of streets or avenues which he has to cross during the drive to get an estimation of the distance, the distance according to the Manhattan metric simply is the sum of the absolute values of the differences in x-coordinate and y-coordinate between departure and arrival points [38]. According to this theory, changes of observer shadow coordinates (x_i, y_i) between two time points can be recorded as departure and arrival. Then by comparing with estimated shadow coordinates x_i', y_i' at the same spot, a cost function shows as following.

$$f_1(x_i, y_i) = \sum_{i=1}^{n} |x_i - x_i'| + |y_i - y_i'| \tag{3.11}$$

Cost Function (II):

In additional to ℓ^1 norm, the Euclidean metric is by far the most commonly used ℓ^2 norm, it gives the ordinary distance between two points as a consequence of the Pythagorean theorem. The theorem of Euclidean metric is ordinary straight-line

distance between two points in Euclidean space, with such distances, Euclidean space becomes a metric space. In terms of this, each (x_i, y_i) as a point to form Euclidean space, and calculate the distance between two adjacent points at two adjacent time spots. Hence the cost function for ℓ^2 norm is,

$$f_2(x_i, y_i) = \sum_{i=1}^{n} \sqrt{(x_i - x_i')^2 + (y_i - y_i')^2} \tag{3.12}$$

where variables definitions and changes are same as in cost function (I).

Cost Function (III):

From $e_0 = Arcsin(\sin\varphi \cdot \sin\delta' + \cos\varphi \cdot \cos\delta' \cos H')$ (Formula 3.7), when object length is unknown, the possible latitude and longitude by SPA algorithms can be utilized to estimate elevation angle e_i'. Hence the estimated ratio of the shadow length at two adjacent moments is,

$$N' = \frac{\tan e_{i+1}'}{\tan e_i'} \tag{3.13}$$

while the observed shadow length is $s_i = l/\tan e_i$ (Formula 3.8), through the formula transformation, the ratio of the elevation angle at two adjacent moments is,

$$N = \frac{\tan e_{i+1}}{\tan e_i} = \frac{l/\tan e_{i+1}}{l/\tan e_i} = \frac{s_{i+1}}{s_i} \tag{3.14}$$

where $s = \sqrt{x_i^2 + y_i^2}$, x_i and y_i are observed shadow local coordinates. Therefore, the cost function by ℓ^1 norm to calculate the changes ratio of solar elevation angles differences between two adjacent moments can be expressed as following equation.

$$f_3(e_i, e_i') = \sum_{i=1}^{n} |N_i - N_i'| \tag{3.15}$$

3.3 Recomposed Genetic Algorithm Procedure

Genetic algorithm is a kind of evolutionary algorithm in the field heuristic algorithm for artificial intelligence, and heuristic algorithms are often used to generate possible solutions to find optimum and resolve global search problems [37]. Global search is not usually applied in complex equation models as it can reduce the convergence pace. However, the possible location information, which is inferred from the objects' shadow vertex data on the ground, requests a global search in the range of latitude and longitude. Therefore, through effective constraints and the smallest possible global value space to achieve fast and accurate positioning search.

In this section, by discussing the constraints for each variables in objective function that is used in GAs, and verifying cost functions which is posted in previous section, then along with appropriate genetic selection, mutation and fitness function, the positioning model by Genetic Algorithm can be established. Algorithm 3.1 shows the pseudocode of the basic Genetic Algorithm, which can serve as the essential model for many extended approaches.

In subsequent sections, we decompose process of basic GA to analyze and compare

Algorithm 3.1 Basic Genetic Algorithm

```
1: procedure INITIALIZE POPULATION
2: repeat
3:     repeat
4:         fitness computation
5:         crossover
6:         mutation
7:         phenotype mapping
8:     until population complete
9:     selection of parental population
10: until termination condition
```

some common used methods for each operator. Discussed each procedure that start from Initialize Populations to termination, and choose the most appropriate one to

apply in our proposed model. Because each operator achieves different functions, the assess metrics for each of them are also different.

3.3.1 Initialize Populations

Definition 3.3.1. (Initialization) *For a set of genes* $C_j \in \mathbf{C}^m$ *where represents the space of all possible individuals, a certain distribution of population* $\mathbf{P} = \{C_1, C_2, ..., C_m\}$ *random cover the all possible space* $\mathbf{C}^m$.

Initial population consisting of arbitrary discrete random chromosomes, and encoding of binary typically. In this paper, to express the feasible solution for possible positions as the chromosomes of genetic algorithm, chromosomes are consisted for observer geocentric latitude (φ) and longitude (σ), which can be presented as $\mathbf{C^m} = \begin{pmatrix} \alpha_{11} & \alpha_{12} & \cdots & \alpha_{1j} & \cdots & \alpha_{1m} \\ \varphi_{21} & \varphi_{22} & \cdots & \varphi_{2j} & \cdots & \varphi_{2m} \end{pmatrix}$, $C_j = (\alpha_{1j},\ \varphi_{2j})$. The lower and upper bounds for this two chromosomes are $(-90°,\ -90°)$ and $(-180°,\ 180°)$ respectively. Population thereby focuses the search on constrained regions in the search space.

In problems where the spread of possible solutions are unknown, a larger search space which can be covered by high precision may reduce the loss of phenotypes, while the computational burden of explore unknown search spaces need to be reduced to a more sustainable level. In GA, several reasons cause premature convergence, such as similar fitness, highly crowded chromosomes, small population size, etc. Therefore, maintain phenotypes in a sufficient level, in order to further maintain the diversity of populations. Consequently, in this perspective, if the size of phenotypes is too small, the optimal individual may not be obtained or be caught in prematurity.

Producing a matrix $\mathbb{R}^{Nind \times (Nvar * Preci)}$ for chromosomes, number of individuals(Nind) uniformly distributed random binary codes of Precision (Preci) $\times$ Number of Variables (Nvar). Number of individuals (Nind) is used to specify the dimensions of the

rows, and (Nvar × Preci) is used to specify the dimensions of the columns in chromosome matrix. For example, date and time t_i, shadow coordinates x_i and y_i are three variables in our positioning model, when set Nind = 80 and Preci = 25, a $\mathbb{R}^{80\times75}$ matrix will be generated.

$$\begin{pmatrix} c_{1,1} & c_{1,2} & c_{1,3} & \dots & c_{1,Nvar*Preci} \\ c_{2,1} & c_{2,2} & c_{2,3} & \dots & c_{2,Nvar*Preci} \\ \dots & \dots & \dots & \dots & \dots \\ c_{Nind,1} & c_{Nind,2} & c_{Nind,3} & \dots & c_{Nind,Nvar*Preci} \end{pmatrix} \begin{matrix} Individual_1 \\ Individual_2 \\ Individual_i \\ Individual_{Nind} \end{matrix}$$

Afterwards, the binary strings in the chromosome matrix are converted to real values according to the lower bound, upper bound and arithmetic scaling, then returns a matrix of real valued phenotypes. The individuates in real-valued matrix passed directly as inputs to other operators of GA and the objective function.

3.3.2 Fitness

Definition 3.3.2. (Fitness) *The fitness function μ assigns and scales to each fitness value $f_i \in \mathbf{P}$ to $\mathbf{C}^m$, the genes of individuals in the population $\mathbf{P}$ carrying this fitness value. Thus, $\mathbf{C}^m$ is called the fitness distribution of a population $\mathbf{P}$.*

Fitness method μ is an algorithm transforms a discrete individuals in $\mathbf{C}^m$ into an fitness distribution. In nature, fitness is usually discussed in terms of genotypes, it means the ability to survive to reproductive, find a mate and produce offspring. In GA, from the state of Blickle [7], proper fitness method depends on a low loss of diversity and high variance are advantageous. In rank-based fitness assignment, the population is sorted according to the objective values. The fitness assigned to each gene depends only on its position in the individuals rank and not on the actual objec-

tive value. Ranking introduces a uniform scaling across the population and provides a simple and effective way of controlling selective pressure. Under the controlled selective pressure, no individuals generate an excessive number of offspring due to the limited reproductive range.

Consider $Nind$ as the number of individuals in the population, Pos is the position of an individual in this population, and SP is the selective pressure. The population is ranked according to a dominance rule, and then each solution is assigned a fitness value based on its rank in the population, not its actual objective function value objectives are assumed to be minimized rank combine to a better solution in the following discussions. The fitness value for an individual is calculated as:

$$Fitness(Pos) = 2 - SP + 2 \times (SP - 1)(Pos - 1)/(Nind - 1) \tag{3.16}$$

for exponential non-linear ranking according to [46]:

$$Fitness(Pos) = \frac{Nind \times X^{Pos-1}}{\sum_{i=1}^{Nind} X^{i-1}} \tag{3.17}$$

where X is computed as the root of the exponential polynomial [67]:

$$0 = (SP - 1) \times X^{Nind-1} + SP \times X^{Nind-2} + ... + SP \times X + SP \tag{3.18}$$

Example 3.3.2. Linear and non-linear fitness assigment

Consider a population with 10 individuals, the current objective values are:

Objective = [5, 4, 3, 2, 1, 10, 9, 8, 7, 6]

Evaluate the fitness by linear ranking and selective pressure SP = 2:

Linear Fitness (Pos) = [1.1111, 1.3333, 1.5556, 1.7778, 2.0000, 0, 0.2222, 0.4444, 0.6667, 0.8889]

Evaluate the fitness by exponential non-linear ranking and selective pressure SP = 2: Non-linear Fitness (Pos) = [0.9568, 1.1504, 1.3833, 1.6633, 2.0000, 0.3807, 0.4577, 0.5504, 0.6618, 0.7957]

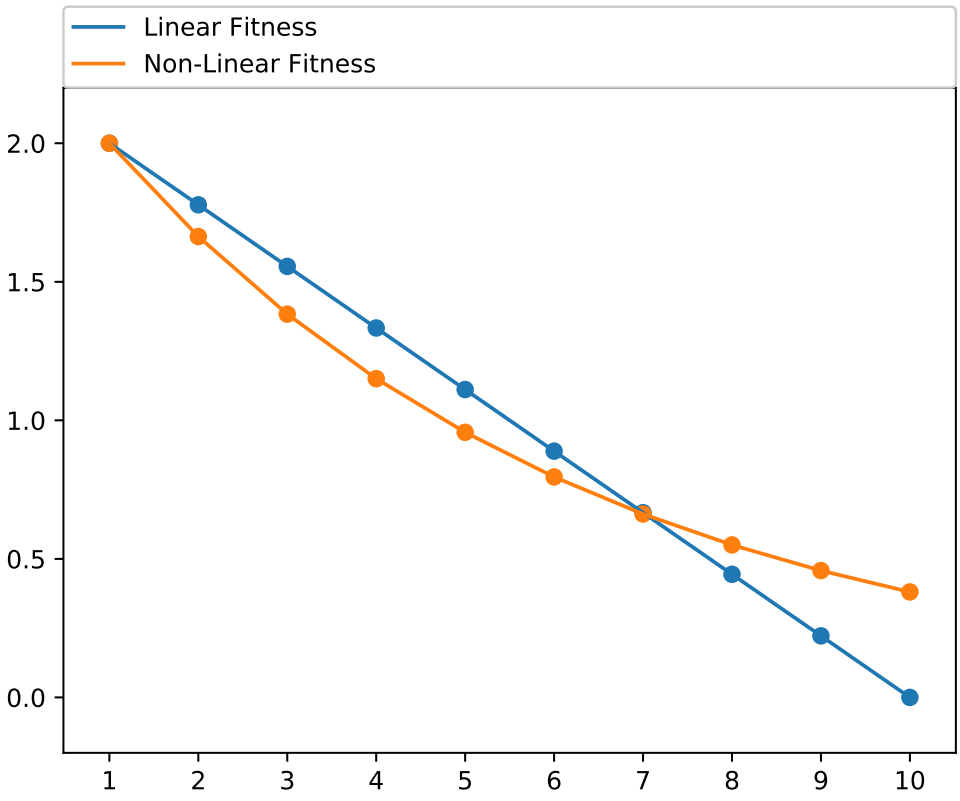

Figure 3.3: Compares Linear and Non-linear Ranking Graphically

Figure 3.3 shows the linear and exponential non-linear ranked fitness assignments for each individual. Both linear and non-linear ranking first sorts values into descending order. The least fit individual is placed in position 1 in the sorted list of objective values and the most fit individual position *Nind*. A fitness value is then assigned to each individual depending on its position, Pos, in the sorted population. Apparently, non-linear ranking can effectively avert zero in the fitness distribution, even though linear ranking shows the smoothness property as same as non-linear ranking.

3.3.3 Selection

Definition 3.3.3. (Selection) *Individuals in a population $C_j \in \mathbf{C}^m$ that carrying fitness value with a certain selection algorithm ω, pick qualified individuals from all possible individuals $\mathbf{C^m} \rightarrowtail \mathbb{C}^n (n \leqslant m)$ in each generation κ.*

The selection operator selects individuals for reproduction on the basis of their relative fitness. It intends to improve the average quality of the population by giving individuals of higher quality a higher probability to be copied into the next generation. For the same selection intensity, truncation selection leads to a much smaller selection variance than ranking or tournament selection [57]. However, truncation selection leads to a much higher loss of diversity for the same selection intensity compared to ranking and tournament selection, and all individuals below a certain fitness threshold don't have a probability to be selected by truncation selection. Although rank-based fitness selection behaviors are similar to tournament selection, ranking selection works in an area where tournament selection doesn't work because of the discrete character of tournament selection. Therefore, in general, ranking selection is the optimal compromise choice to balance low loss of diversity and high variance.

Baker [3] presented three metrics to evaluate selection algorithms, bias, spread and efficiency. Bias is defined as the absolute difference between an individual's actual and expected selection probability [69]. Spread is the range of possible values for the number of offspring of an individual. The selection algorithm should thus achieve zero bias whilst maintaining a minimum spread and not contributing to an increased time complexity of the GA.

Stochastic Universal Sampling (SUS) can be stated to be an optimal sampling algorithm with minimum spread and zero bias [7]. The individuals are mapped to contiguous segments of a line, such that each individual's segment is equal in size

to its fitness. Then using N equally spaced pointers, N is the number of selections required. In another words, only a single spin of the SUS has N markers for the selective individuals and hence all individuals have equal chance at once.

Example 3.3.3. Probability of non-linear fitness assignment

Consider a population with 10 individuals, the current objective values are:

Objectives = [1, 2, 3, 4, 5, 6, 7, 8, 9 ,10]

The sorted fitness and probability by exponential non-linear ranking are: Non-linear Fitness (Pos) = [2.0000, 1.6633, 1.3833, 1.1504, 0.9568, 0.7957, 0.6618, 0.5504, 0.4577, 0.3807]

Accumulate Probability = [0.2000, 0.3663, 0.5047, 0.6197, 0.7154, 0.7950, 0.8611, 0.9162, 0.9619, 1.0000]

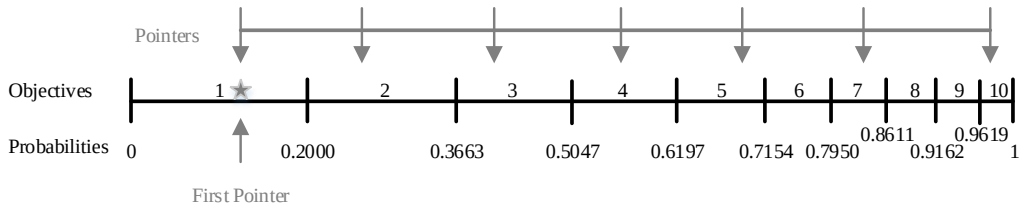

Figure 3.4: Stochastic Universal Sampling Selection Graphically

Consider $NPorinter/Nind$ individuals to be selected, then the distance between the pointers are $1/NPointer$ and the position of the first pointer is given by a randomly generated number in the range $[0, 1/NPointer]$. For 70% individuals to be selected, the distance between the pointers is $1/7 = 0.1429$. Figure 3.4 shows the selection for the above example, the sample of first random number in the range [0, 0.1429]. After selection the mating population consists of the individuals are [1, 2, 3, 4, 5, 7, 10], survived individuals breed offsprings and their chromosomes are inherited. For the easy understanding of SUS selection, Figure 3.4 shows objectives and pointers

graphically.

With the equally space pointers, the outcome of certain runs of SUS selection scheme is as close as possible to the expected distribution. Similarly, even though it is not certify whether has performance advantages in using SUS, it definitely makes the run of a selection method more predictable. To be able to apply SUS, we has to know the expected reproduction rate of offspring of each individuals. The equations for the discrete individuals to calculate reproduction rate for exponential ranking was stated in by Srinivas and Lalit [57], according to that we can obtain an appropriate rate of individuals to be selected to propagate next generation.

Overall, the selection of individuals can be viewed as two separate processes. The first stage is concerned with the transformation of raw fitness values into a real valued expectation of an individual's probability to reproduce and is dealt with in the previous subsection as fitness assignment. The second part is the probabilistic selection of individuals for reproduction based on the fitness of individuals relative to one another.

3.3.4 Recombination

Definition 3.3.4. (Recombination) *Each round of the loop is called a generation and* $\mathbf{P}_\kappa$ *denotes the population at generation* κ*. In each generation, by certain recombination algorithm* γ*, let* $\mathbb{C}^n \rightarrowtail \mathbf{R}^b (b \leqslant m)$.

Recombination includes crossover and mutation or any other operator that changes the genetic material, and probability of crossover (p_c) and probability of mutation (p_m) is acknowledged to critical effect the GA behaviour and performance [18]. From literatures, crossover operator basically simulates mating from two or more individuals and there are a number of ways it is usually implemented in GAs. Not all matings

must reproduce by crossover, if crossover probability $p_c = 1.0$ then all offspring is made by crossover. If it is 0 then whole new generation is made from exact copies of chromosomes from old population, but this does not mean that the new generation is the same. The fulfillment of this condition is the existence of constraints that shrink the whole solution space to a feasible subset.

If only use the crossover operator to produce offspring, one potential problem that may arise is that if all the chromosomes in the initial population have the same value at a particular position then all future offspring will have this same value at this position or called allele. For example, if all the chromosomes have a 0 in position two then all future offspring will have a 0 at position two. This has the effect of tending to inhibit the possibility of converging to a local optimum, rather than the global optimum. To combat this undesirable situation then mutation operator is used. Mutation operator attempts to change the individual genetic representation according to some probabilistic rule. In the initialized binary string representations, mutation will cause a single or multiple alleles to change its state. Such as 0 becomes 1 and vice versa. Typically this occurs infrequently to retain phenotypes in chromosomes, and bit in each chromosome is checked for probability of mutation p_m.

Moderately large values of p_c in [0.5, 1.0) and small values of p_m in (0, 0.05] are commonly employed in GA practice. In an approach of adaptive probabilities for GA, authors proposed an varying p_c and p_m, at achieving this trade-off between exploration and exploitation in a different manner [57]. Adaptively in response to the fitness values of the feasible solutions p_c and p_m are increased when the population tends to get stuck at a local optimum and are decreased when the population is scattered in the solution space.

$$p_c = k_1(f_{max} - f_i^{'})/(f_{max} - \overline{f}), 0.2 < k_1 \leqslant 1.0 \tag{3.19}$$

and

$$p_m = k_2(f_{max} - f_i)/(f_{max} - \overline{f}), 0.1 < k_2 \leqslant 0.5 \tag{3.20}$$

where,

- f_i is the fitness value of an individual of the population.
- $f_i^{'}$ is the larger of the fitness values of the population to be crossed, $f_i^{'} \geqslant \overline{f}$.
- f_{max} is the maximum fitness value of the population.
- $\overline{f}$ is the average fitness value of the population.

In the research of [57], they notice GA is not sensitive to the external parameter k_2, the adaptive p_m gets stuck at local optima fewer times for higher values than for lower values of k_2. Due to such experimental results, in this paper, we designate k_2 = 0.5 as the most appropriate value for p_m. To achieve the most appropriate p_c, we observe that the average fitness of the population increases gradually for the adaptive crossover rate, approximately increase 0.05 for k_1 per generation and the best k_1 is 0.75.

We have observed the property of static $p_c = 0.5$ and $p_m = 0.05$ in all our experiments with static GA, and the adaptive p_c, p_m that has been introduced. On comparing the two plots, we observe that the average fitness of the population increases rapidly while it gradually for the adaptive GA. f_{max} - $\overline{f}$ is one of a possible metric of detecting convergence, when the GA converges to a local optimum with a fitness value then f_{max} - $\overline{f}$ will approach to 0 [62]. Figure 3.5 apparently shows f_{max} - $\overline{f}$ near to 0 after 35th generation, hence the static GA has not located the global optimum on fitness of 1, and has only located a locally optimal solution with a fitness around 0.89.

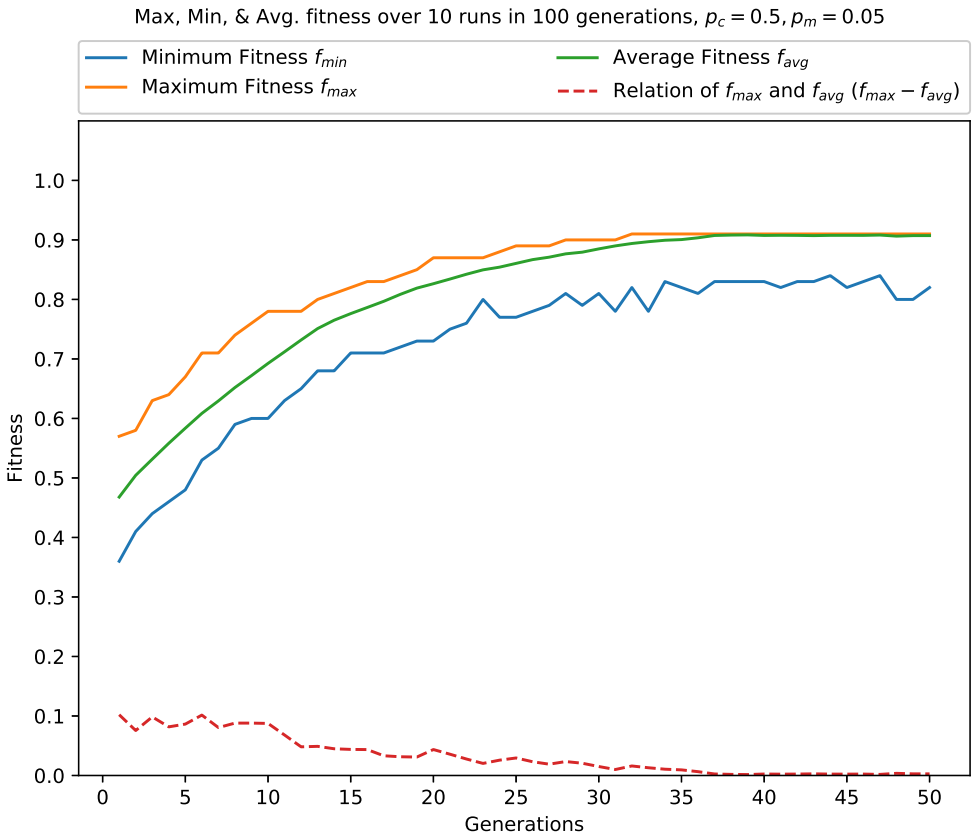

Figure 3.5: Compare Max, Min and Avg Fitness Values for Static GA

Figure 3.6 reveals that, in the first 5 generations, the average fitness for the adaptive GA increases rapidly, and then remains flat until around the 15th generation, and approaches to globally optimal solution has a fitness value of 1. As a drawback of adaptive GA, the lower average fitness value indicates that the population has remained scattered in the search space, thus it has not achieved a better convergence performance. However, the adaptive GA can effectively prevent propagates from getting stuck at the local optimum with a fitness value of around 0.89 that the static GA has succumbed to.

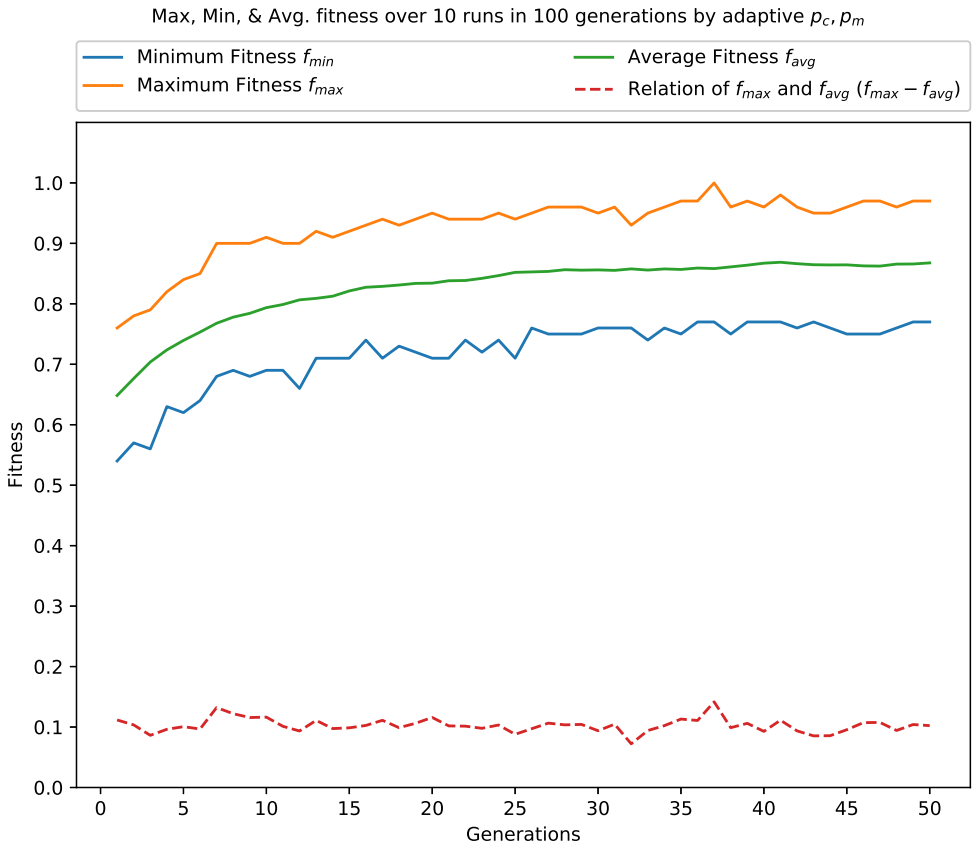

Figure 3.6: Compare Max, Min and Avg Fitness Values for Adaptive GA

After compared static and adaptive approach to determine a better way for probabilities of crossover and mutation, the p_c and p_m. For adaptive p_c and p_m, the probabilities are not predefined, they are determined adaptively for each solution of the population. In this paper, we chosen this particular way of adapting p_c and p_m for recombination in the proposed algorithm.

3.3.5 Reinsertion

Definition 3.3.5. (Reinsertion) *The most fit individuals in population* $\mathbf{P}_\kappa \in \mathbf{R}^b$ *replace the least fit members in* $\mathbf{P}_{\kappa-1}$ *by a certain insert algorithm* χ*, to generate* $\mathbf{P}_{\kappa+1} \in \mathbf{R}^a (a \leqslant b)$ *as the new population in next round.*

To maintain the size of the original population, the new individuals have to be reinserted into the old population. If one or more of the most fit individuals is ultimately allowed to reproduce through successive generations then that will be took for elitist strategy in GA. Similarly, if not all the new individuals are to be used at each generation or if more offspring are generated than the size of the old population then a reinsertion scheme must be used to determine which individuals are to exist in the new population. The important feature of not creating more offspring than the current population size at each generation is that the generational computational time is reduced, the memory requirements are smaller as fewer new individuals need to be stored while offspring are produced.

When fewer individuals are produced by recombination than the size of the original population, then the fractional difference between the new and old population sizes is termed a generation gap (GGAP). In the literatures, no many papers clearly point out how to assign GGAP and most of them use 'steady-state' generation gap. Sarma,J. and DeJong,.K [52] analyzed so-called 'steady-state' incremental generation gap in their research, the analysis emphasizes that the important behavioral differences between 'steady-state' and generational distribution have little if anything to do with the choice of generation gap size. While the result also demonstrated that incremental GGAP value reduces the variance seen on individual GA runs which can be an important issue for final output.

Follow the principle of less offspring size than current population, GGAP less than 1.0 ($GGAP \leqslant 1.0$). This example uses the number of individuals is set to $Nind = 40$ and a generation gap, $GGAP = 0.9$, and fitness-based reinsertion to implement an elitist strategy whereby the most fit individuals always propagate through to successive generations. Thus, in this example 36 ($Nind \times GGAP$) new individuals are produced at each generation.

3.3.6 Termination

Definition 3.3.6. (Termination) *In a predefined number of generations G and testing the quality by objective function $\mathbf{F}$, execute in a loop until some termination criterion Ω is reached.*

The cost function is the only criterion used to distinguish the quality of offspring individuals, and decodes binary chromosome code into vectors of real function values. The quality of an individual is measured by a cost function. According the SPA algorithm to calculate solar elevation angle and azimuth angle, then consequently get the object shadow length. The fitness function will compare all sets of possible latitude and longitude with cost functions to obtain the minimum, which will be the optimal individual. We express SPA algorithm as $g(\sigma, \varphi, t)$, and constrains for termination criterion is

$$\begin{cases} e_i, \Gamma_i = g(\sigma, \varphi, t) \\ s_i = \dfrac{l}{tane_i} \qquad\qquad s = 0, if\ e_i = 90^\circ \\ x_i = s_i \cdot \sin \Gamma_i \\ y_i = s_i \cdot \cos \Gamma_i \\ -180^\circ < \sigma < 180^\circ \\ -90^\circ < \varphi < 90^\circ \end{cases} \tag{3.21}$$

Since the possible latitude and longitude can be consider as a matrix $\mathbf{C} = \begin{pmatrix} \alpha_{11} & \alpha_{12} & \dots & \alpha_{1j} & \dots & \alpha_{1m} \\ \varphi_{21} & \varphi_{22} & \dots & \varphi_{2j} & \dots & \varphi_{2m} \end{pmatrix}$, where $(\sigma_{1j}, \varphi_{2j})$ is a set of possible latitude and longitude. Where cost function can be indicated as $f(x_i, y_i)$, thus objective function $\mathcal{O}$ is,

$$\mathcal{O} = \sum_{j=1}^{m} \sum_{i=1}^{n} g(f(x_i, y_i)) \tag{3.22}$$

3.4 Genetic Shadow Positioning Algorithm (GSPA)

From the view of entire shadow posting schemes, by integrating the SPA algorithm, we can construct a Genetic Shadow Positioning Algorithm in the following ways: First we compute the fitness distribution of the initialized population. Then SUS selection scheme chooses the possible solution to the expected distribution from the global space. Next the expected reproduction rates are calculated using the equations that derived in the proceeding sections. The last step, SPA algorithm as objective function which is used to obtain the new population. The overall scheme is outlined by algorithm GSPA, and procedures are demonstrated in Algorithm 3.2.

Algorithm 3.2 Genetic Shadow Positioning Algorithm (GSPA)

1: **procedure** INITIALIZE POPULATION($\mathbf{P}$)
2: **encode** chromosomes in population $\mathbf{P} \rightarrowtail \mathbf{P^m}$
3: **while** $G \leqslant max(G)$
4: **repeat**
5: fitness computation of μ, $\mathbf{P^m} \rightarrowtail \mathbf{C^m}$
6: selection of ω, $\mathbf{C^m} \rightarrowtail \mathbb{C}^n (m \leqslant n)$
7: recombination of γ, $\mathbb{C}^n \rightarrowtail \mathbf{R}^b (b \leqslant n)$
8: **until** population complete
9: minimization of SPA objective function $\mathcal{O}$ (equation 3.22)
10: reinsert of χ, $\mathbf{P}_\kappa \in \mathbf{R}^b \rightarrowtail \mathbf{P}_{\kappa+1} \in \mathbf{R}^a (a \leqslant b)$
11: **until** termination condition (Ω)

3.5 NSGA-II Multi-objectives Genetic Algorithm

In the single objective function, problem finds the optimal solution that is superior to all other solutions, usually the global minimum or maximum solution. While the optimal solution with respect to a single objective often results in unacceptable results with respect to the other objectives. Multi-objective optimization involves more

than single objective functions to be optimized simultaneously, a reasonable solution to a multi-objective problem is to investigate a set of solutions, each of which satisfies the objectives at an acceptable level without being dominated by any other solution. The multiobjective problem involves two spaces: the decision space and the objective function space. The optimized search process takes place in the objective function space.

A minimization multi-objective decision problem with m objectives is defined as follows: Given an n-dimensional decision variable vector $\mathrm{X} = x_1, ..., x_n$ in the solution space $\mathbf{X}$, find a vector x^* that minimizes a given set of m objective functions $F(x) = (f_1(x), ..., f_i(x), ..., f_m(x))^T, (m \geqslant 2)$. The solution space X is restricted by a series of constraints, such as $h_j(x)$ equal to a constant or in an interval. Generally, a multiobjective problem can be expressed as,

$$\begin{cases} min \quad Y = F(x) = (f_1(x), ..., f_i(x), ..., f_m(x))^T, & m \geqslant 2 \\ s.t. \quad h_j(x) \leqslant 0, & j = 1, 2, ..., q \end{cases} \tag{3.23}$$

where, $\mathrm{X} = x_1, ..., x_n \in \mathbf{X} \subset \mathbb{R}^n$(Euclidean space) is n-dimensional decision variable vector and Y is m-dimensional objective function space. Hence, multiobjective function $F(x)$ defines the mapping from n-dimensional decision variable vector to m-dimensional objective function space, $\mathbb{R}^n \to \mathbb{R}^m$. This multiobjective problems are equivalent to vector optimization problems.

3.5.1 Theorems for Multi-objectives Optimization

Theorem 1. *If $x \in \mathrm{X}$, for $\forall x \in \mathrm{X}$ and $i \in [1, m]$, $\exists f_i(x^*) \leqslant f_i(x)$, thus the vector x^* is the optimal solution of ith objective function, and consists a set of optimal solutions X^*.*

[8] Assume x_1, x_2 are two feasible solutions in X to the ith objective $f_i(x)$, if has $f_i(x_1) \geqslant f_i(x_2)$ for $\forall i \in [1, m]$ and $\exists i \in [1, m]$, then we say that x_1 is better than x_2, or x_1 dominates x_2. Hence, if has an optimal point x^* satisfies, $f_i(x^*) \leqslant f_i(x)$ for every feasible x. In other words, X^* is simultaneously optimal for each of the problems in $F(x)$. When there is an optimal point, we say that the objectives are noncompeting, since *no compromises* have to made made among the objectives; each objective is as small as it could be made, even if the others were ignored [8].

Theorem 2. *If* $x \in \mathrm{X}$*, for* $\forall x \in \mathrm{X}$ *and* $i \in [1, m]$*,* $\nexists f_i(x) \leqslant f_i(x^{po})$*, thus the vector* $\mathrm{x^{po}}$ *is the Pareto optimal solution of ith objective function, and consists a set of Pareto optimal solutions* X^{po}*.*

[8] This can be restated as: a point is Pareto optimal if and only if it is feasible and there is no better or non-dominated feasible points. For the easy understanding of Pareto optimal and non-dominated feasible points, Figure 3.7 shows feasible solutions space, dominated and non-dominated points graphically. In particular, if a feasible point is not Pareto optimal, there is at least one other feasible point that is better or dominated. Similarly, we can clearly limit the search space to Pareto optimal points, which cannot be improved with respect to any objective without worsening at least one other objective. The set of all feasible non-dominated solutions in X^{po} is referred to as the Pareto optimal set, and for a given Pareto optimal set, the corresponding objective function values in the objective space are called the Pareto front. The ultimate goal of a multi-objective optimization algorithm is to identify solutions in the Pareto optimal set. In the presence of multiple Pareto optimal solutions, it is difficult to choose which solution is preferable if there is no extra information about the problem, so the Pareto optimal solution for all problems can be considered equally important. Thus, for the multi-objective optimization problem, two important principles [16] [35] [73]:

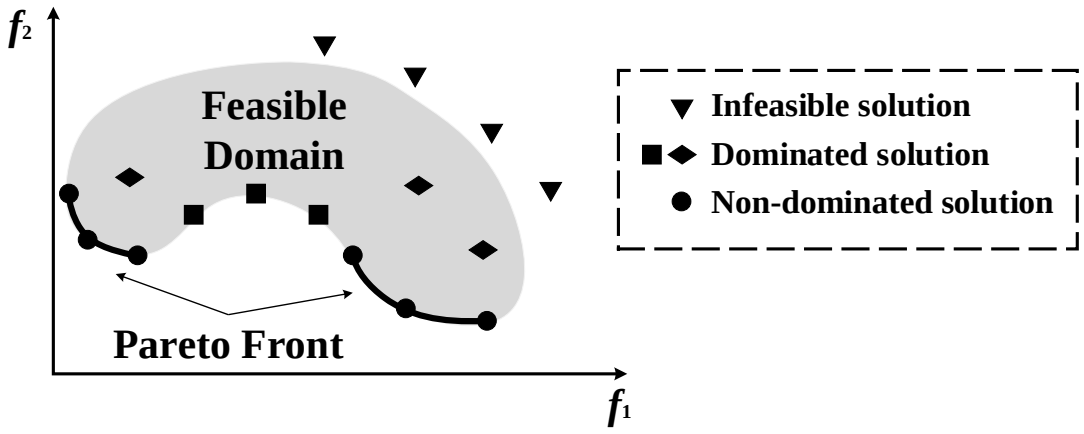

Figure 3.7: Pareto Optimal Front and Non-dominated Points

1. The best Pareto front should be as close as possible to the true Pareto front

2. Solutions in the best Pareto set should be uniformly distributed and diverse over of the Pareto front

The first principle is to be done in any optimization work, and it is not acceptable if the solution does not converge to the searching space. When a set of solutions converges to the exact Pareto optimal solution, the set of solutions is guaranteed to be optimal. Except to the solution that requires the optimization problem to converge to the approximate Pareto optimal space, the resulting solution must be uniformly sparsely distributed over the Pareto optimal space. A fine set of *compromised* solutions between multiple objectives is based on a set of diverse solutions. As mentioned earlier, in the multi-objective algorithm, two types of space - the decision space and the objective function space need to be processed, so the diversity between each solution can be defined in the two spaces.

Some schemes of Non-dominated Sorting Genetic Algorithm-II (NSGA-II) are significant based on multi-objective optimization and Pareto optimal.The main advantage of such extended genetic algorithms, when applied to solve multi-objective optimiza-

tion problems, is the fact that they typically generate sets of solutions, allowing computation of an approximation of the entire Pareto front.

3.5.2 NSGA-II Methodology

The nondominated sorting genetic algorithm (NSGA) proposed in [58] was one of the first nondominated sorting Genetic algorithms, then NSGA-II [21] was proposed to improved the version of NSGA. The main issues of NSGA are: (i) high computational complexity of nondominated sorting; (ii) need for specifying the sharing parameter σ_{share} [58]; (iii) lack of elitism. Since Genetic Algorithms essentially work with a global searching feature, it can be directly extended to maintain a diverse set of solutions. With an emphasised ability for moving toward the true Pareto-optimal region, a GA can be used to find multiple Pareto-optimal solutions in single simulation run. The most widely employed NSGA-II algorithm improved NSGA in three primary aspects: (i) fast nondominated sorting operator; (ii) density estimation and crowded-comparison operator; (iii) elitist-preserving strategy.

(I) Fast Nondominated Sorting Operator

The total complexity of nondominated sorting operator in NSGA and the fast non-dominated sorting operator in NSGA-II, with a population of size N and M number of objectives, have been reduced from $O(MN^3)$ to $O(MN^2)$ [21]. In the naive non-dominated sorting operator of NSGA, each solution should be compared with every other solution in the population to identify if it is dominated or nondominated, which requires $O(MN)$ comparisons for each solution. After this stage, all individuals in the first of nondominated front are found for each objective. In order to find the individuals in the next nondominated front, the solutions of the first front repeats the procedure again. In a worse case, the task of finding the second front also requires.

While fast nondominated sorting approach require less computations. In the first place, for each solution needs to calculate two entities,

- domination count n_p, the number of other solutions which dominate the solution p
- a set of solutions S_p that is dominated the solution p

thus this requires $O(MN^2)$ comparisons. Secondly, find out all $n_{p1} = 0$ and put it in a separate list Q_1 temporarily. For each solution q in S_{p1}, reduce its domination count by one, then repeats above procedure to constitute the second dominated set of solution n_{p2} and S_{p2}. In this stage, put solution $n_{p2}-1 = 0$ into a another new separate set Q_2, which is the second nondomination list. Since all dominated individual q is one of the nondominated individual p in Q_1. At this point, the solution is assigned a nondomination level and will never be visited again. This process continues until all fronts levels are identified. $\mathcal{F}_1$ as the first nondomination level, so every solution in Q_1 is optimal and all individuals in the set are given the same nondominantion ranking, $p_{rank} = 1$, $q_{rank} = 2$, and so forth. For the easy understanding of the fast nondominated sorting operator, we paste its pseudocode in Algorithm 3.3.

(II) Crowded-comparison operator

To maintain sustainable diversity in a population, the parameter σ_{share} is designated by users in NSGA. It denotes the largest value of that distance metric within which any two solutions share each other's fitness. NSAG-II proposed a crowding-distance calculation, points marked in filled circles are solutions of the same nondominated front. They define a density-estimation metric and then build the crowded-comparison operator. Although the crowding distance is calculated in the objective function space, it can also be implemented in the parameter space, if so desired [19].

Algorithm 3.3 Fast Nondominated Sorting Operator [21]

```
procedure INITIALIZE POPULATION(P)
for each p ∈ P
        n_p = 0
        S_p = ∅
    for each q ∈ P
        if (p ≺ q) then
          S_p = S_p ∪ q
        else if q ≺ p then
          n_p = n_p + 1
      if n_p = 0 then
        p_rank = 1
        F_1 = F_1 ∪ {p}
i = 1
while F_1 ≠ ∅
      Q = ∅
      for each p ∈ F_1
        for each q ∈ S_p
          n_q = n_q − 1
          if n_q = 0 then
            q_rank = i + 1
            Q = Q ∪ {q}
      i = i + 1
      F_i = Q
```

However, in this study, we have used the objective-function space.

The crowding distance $i_{distance}$ of individual i is the distance between its two adjacent points $i+1$ and $i-1$ in the objective-function space with the same i_{rank}. This quantity $i_{distance}$ serves as an estimate of the perimeter of the cuboid formed by using the nearest neighbors as the vertices, the crowding distance of the ith solution in its front is the average side length of the cuboid (shown with a dashed box) in Figure 3.8.

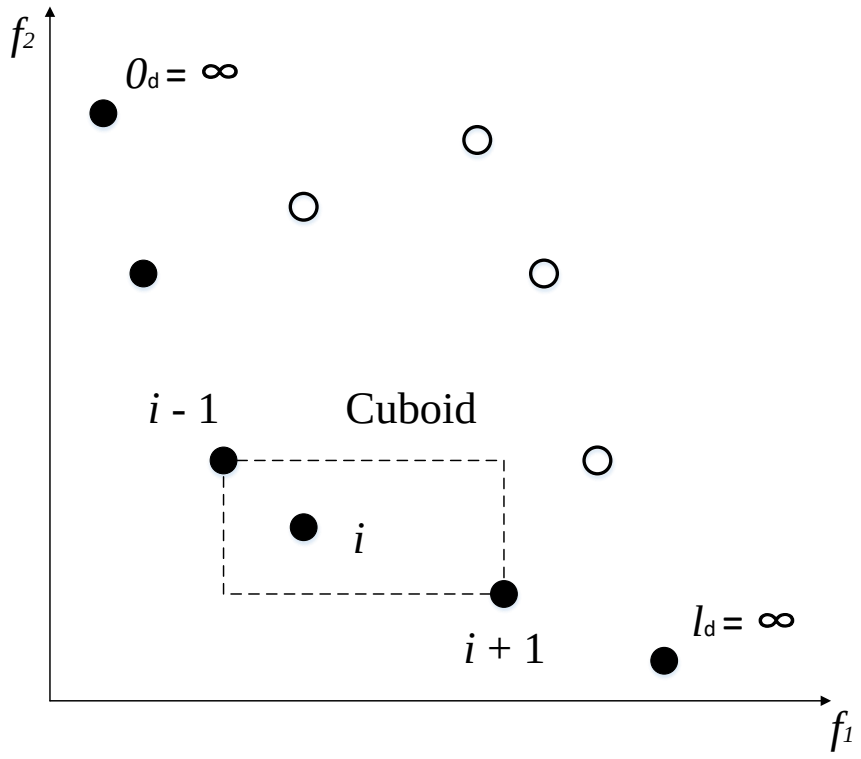

Figure 3.8: Crowding distance calculation in NSGA-II

The calculation of the crowding distance is as follows,

1. initial the $i_{distance} = 0$;
2. sort the population according to each objective function value in ascending order of magnitude;

3. identify f_m^{max} and f_m^{min} are the maximum and minimum values of the m objective function;

4. allow boundary points $O_{distance} = I_{distance} = \infty$, to ensure these two points can be selected;

5. calculate each $i_{distance}$ by

$$i_{distance} = \sum_{p=1}^{n} (|f_p^{i+1} - f_p^{i-1}|)$$

   where f_p^{i+1} and f_p^{i-1} are solution points of $i+1$ and $i-1$ in objective function m respectively;

6. repeat step 2-4 for each objective function, and calculate $i_{distance}$ for each solution

After all population members in the set $\mathcal{F}_i$ are assigned a distance metric, we can compare two solutions for their extent of proximity with other solutions. The individuals with larger crowding distances are preferred so that the calculated results are evenly distributed across the objective space to maintain the diversity of the population. The crowded-comparison operator $\succ_n$ guides the selection process in the Genetic algorithm toward a uniformly spread to Pareto optimal, and assume that every individual in the population has two attributes:

1. nondomination rank $i_{distance} = 0$;

2. crowding distance $i_{distance}$;

the crowded-comparison operator are defined by two properties, the individual i is

considered to the better as long as one condition of these two is met.

$$i \succ_n j \; if \; (i_{rank} < j_{rank} \; or \; (i_{rank} = j_{rank} \; and \; (i_{distance} > j_{distance}))$$

(III) Elitist-preserving Strategy

The elite strategy is to keep the superior individuals in the parent directly into the offspring to prevent the Pareto optimal solution from being lost. Elite strategy select offspring bases on three indicators of parent P_t, offspring Q_t, and combined population R_t, and then form a new parent group P_{t+1}. The NSGA-II procure shows in Figure 3.9.

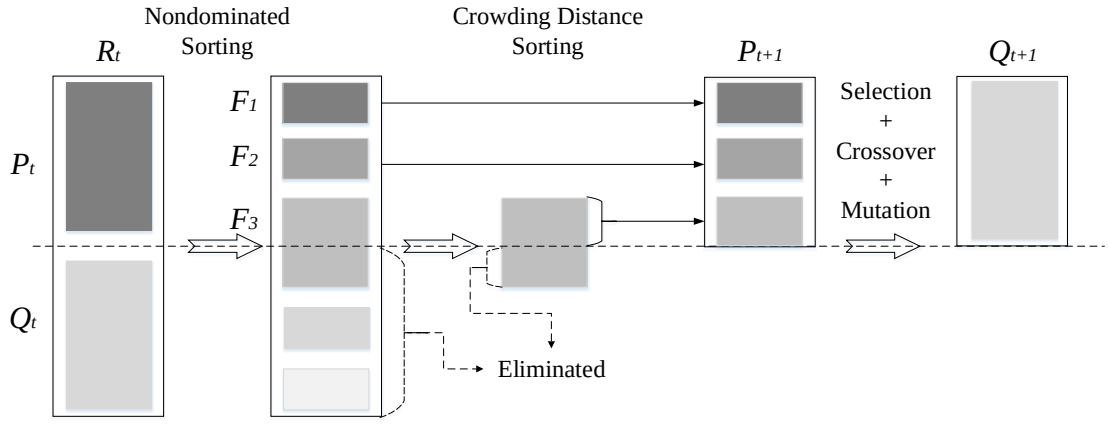

Figure 3.9: Schematic of NSGA-II algorithm [68]

The step-by-step procedure shows that NSGA-II algorithm is simple and straightforward. Firstly, a combined population is formed by $R_t = P_t \cup Q_t$, and population size of R_t is usually $2N$. Then, the population is sorted according to nondomination i_{rank} level $\mathcal{F} = (\mathcal{F}_1, \mathcal{F}_2, ..., \mathcal{F}_l)$. After combination and sorting, solutions belonging to the best nondominated set are of best solutions in R_t, hence the front $\mathcal{F}_i$ will fill into P_{t+1} till the size of $P_{t+1} = N$, and infeasible solutions will be directly eliminated at the same stage. In general, the count of solutions in $\mathcal{F}$ would be larger than the

population size. Repeat steps until the maximum number of generations is reached. If the size of is smaller than N, NSGA-II definitely chooses all members of the set for the new population.

The new parent P_{t+1} will be used for selection, and recombination includes crossover and mutation process in genetic algorithm to generate a new offspring Q_{t+1}. It is important to note that we use SUS selection method during the population reduction phase, but the selection criterion is based on the crowded-comparison operator. Since solutions compete with crowding-distance which depends on the density of solutions among two neighborhoods, no extra parameter is required in NSGA-II.

3.5.3 NSGA-II in GSPA

Implementing NSGA-II to the SPA Genetic algorithm for multiobjective optimization, the steps of this new algorithm are as follows and the entire procedure shows in Figure 3.10:

1. Create an initial parent population $\mathbf{P}$ of size N.

2. Sort the parent population P_t and offspring population Q_t based on nondomination level.

3. For each nondominated solution, assign a fitness (rank) equal to its nondomination level. In this case, $\mathcal{F}_l$ is the best level, $\mathcal{F}_2$ is the second best level, and so on.

4. From the first generation onwards, propagates each new generation by following steps:

   - Combine parents and offspring population as R_t of size $2N$, $R_t = P_t \cup Q_t$.

- Sort the combined population R_t according to the fast nondominated sorting operator [21] to identify all level of nondominated fronts $\mathcal{F} = (\mathcal{F}_1, \mathcal{F}_2, ..., \mathcal{F}_l)$.
- Generate the new parent population P_{t+1} of size N by adding nondominated solutions starting from the first ranked nondominated front $\mathcal{F}_1$ and proceeding with the subsequently ranked nondominated fronts till the size achieve N. This is achieved through a sorting that's made by the crowded comparison operator $\succ_n$, which based on the crowding distance assigned to each solution contained in the ith nondominated front. Likewise, in order to make the total count of the nondominated solutions equal to N, it is required to reject some of the lower ranked solutions from the last ith nondominated front.
- Proceeding with the subsequently ranked nondominated fronts from $\mathcal{F}_1$ to $\mathcal{F}_l$ subsequently, till the size exceeds N.

5. Create a child population Q_{t+1} of size N using SUS selection, crossover and mutation operators, which has been define and described in section 3.3.

6. Repeat step 5 until the maximum number of generations is reached.

From feasible solution results in Figure 4.4 and 4.5, we assume the objective function space by using cost function is $f_2(x_i,\ y_i)$ included the objective function space by $f_1(x_i,\ y_i)$. For a minimizing problem, it's invalid that solutions space of $f_1(x_i,\ y_i)$ is larger than solutions space of $f_2(x_i,\ y_i)$. Thus the multiobjective function is composed by two cost functions.

$$\mathbf{F}(x,\ y) = (f_2(x_i,\ y_i), f_3(e_i,\ e_i'))^T \tag{3.24}$$

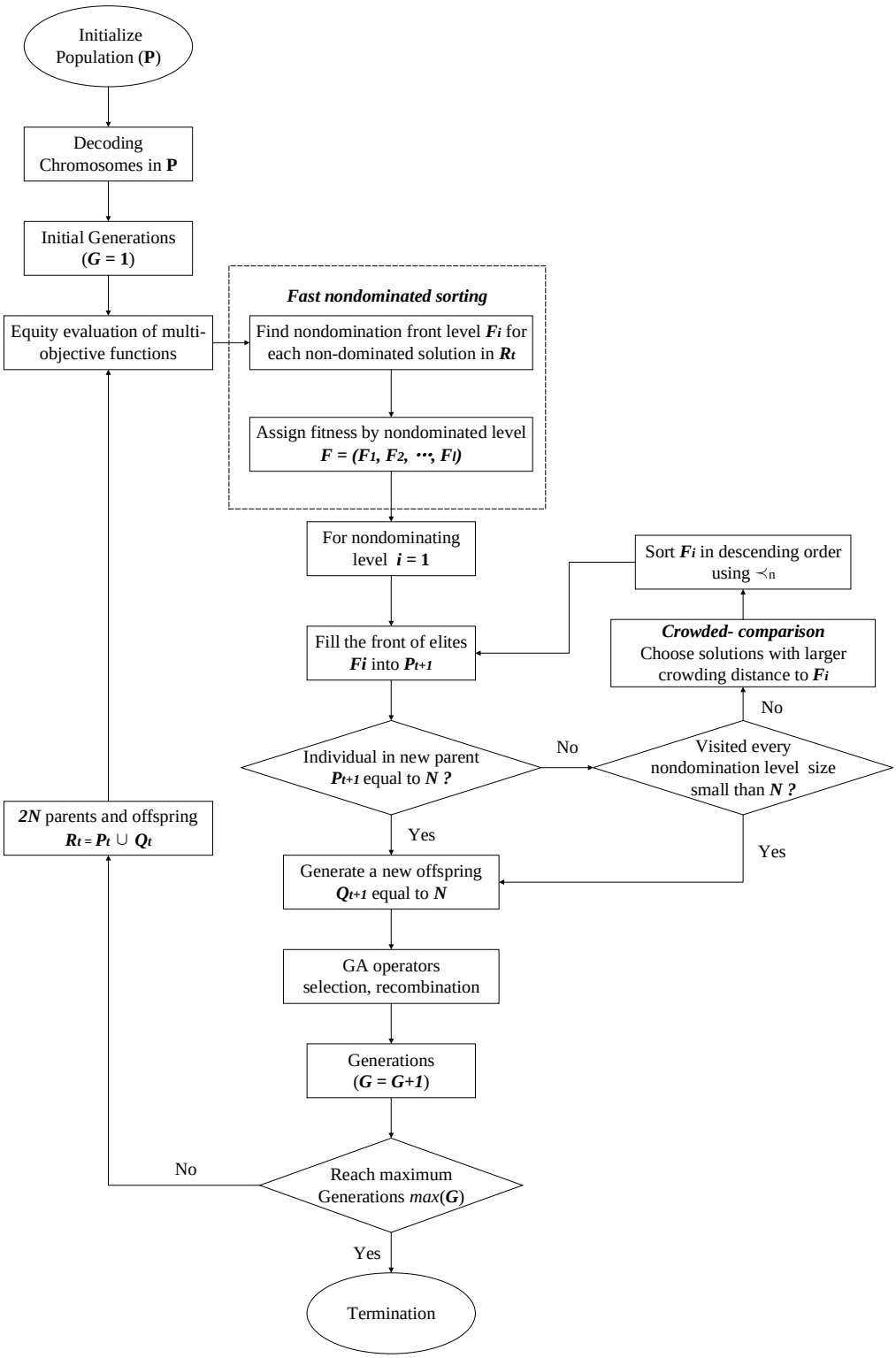

Figure 3.10: NSGA-II Genetic Shadow Positioning Algorithm Diagram

To optimize the minimizing problem, the equality constraints of the form $h_j(x) = 0$, where the functions $h_j(x) = 0$ are linear or affine. For theoretical purposes, equality constraints are redundant; however, it can be beneficial to treat them specially in practice. In the paper, the inequality constraints give the latitude and longitude search space, and the equality constraints for the SPA algorithm $g(\sigma,\ \varphi,\ t)$ concrete the role to calculate objective space. SPA algorithm is composed by intricate trigonometric functions, which is periodic and does not have the convexity. However, Fig 3.2 clearly shows the convexity for $s_i,\ y_i,\ y_i$ in the daytime (when sun has rose). Genetic algorithm go through the whole search space for latitude and longitude by SPA and multiple cost functions $\mathbf{F}(x,\ y)$, the problem is intuitively manifested to the minimizing convex functions over convex sets.

Similarly, with the SPA algorithm $g(\sigma,\ \varphi,\ t)$ and shadow vertex coordinate model (section 3.1.2), the multiobjective function $mathbfF(x,\ y)$ subjects to following constraints,

$$\begin{cases} e_i = g(\alpha, \varphi, t) \\ s_i = \dfrac{l}{tane_i} \qquad\qquad s = 0, if\ e_i = 90^\circ \\ x_i = s_i \cdot \sin\Gamma_i \\ y_i = s_i \cdot \cos\Gamma_i \\ -180^\circ < \sigma < 180^\circ \\ -90^\circ < \varphi < 90^\circ \end{cases} \tag{3.25}$$

By the same token, the corresponding pseudocode for NSGA-II Genetic Shadow Positioning Algorithm (NGSPA) as following,

Algorithm 3.4 NSAG-II Genetic Shadow Positioning Algorithm (NGSPA)

procedure INITIALIZE POPULATION($\mathbf{P}$)
encode chromosomes in population $\mathbf{P} \rightarrowtail \mathbf{P^m}$
while $G \leqslant max(G)$
    repeat
        fitness computation of μ, $\mathbf{P^m} \rightarrowtail \mathbf{C^m}$
        estimation of SPA objective function $\mathbf{F}(x, y)$ (equation 3.24 and 3.25)
            for each individual i in $\mathbf{C^m}$
                assign i_{rank} level $\mathbf{C^m} \rightarrowtail \mathcal{F} = (\mathcal{F}_1, \mathcal{F}_2, ..., \mathcal{F}_l)$
                fast nondominated sorting $\mathcal{F}_i$
            repeat
                calculate $i_{distance}$ for each individual in each $\mathcal{F}_i$
                add individuals to $P_{\kappa+1}$ from best front $\mathcal{F}_1$
                until size$(P_{\kappa+1}) = N$
            offspring $\mathbf{Q}_{\kappa+1}$, size $= N$
            end
        end
        selection of ω, $\mathbf{C^m} \rightarrowtail \mathbb{C}^n (n \leqslant m)$
        recombination of γ, $\mathbb{C}^n \rightarrowtail \mathbf{R}^b (b \leqslant n)$
    until population complete
    reinsert of χ, $\mathbf{P}_\kappa \in \mathbf{R}^b \rightarrowtail \mathbf{P}_{\kappa+1} \in \mathbf{R}^a (a \leqslant b)$
until termination condition (Ω)

Chapter 4

Feasibility Analysis and Experiments

The purpose of the chapter to demonstrate the effectiveness and experimental result of our proposed models GSPA and NGSPA. Estimated latitude and longitude that presents locations in degrees from 180° west through 180° east along the equator, and 90° north through 90° south along the prime meridian. The description formats are decimal degrees with negative numbers for south and west. The observation dataset for our experiments has been completed image process by Gray-Level Co-Occurrence Matrix (GLCM), which is not included in this thesis.

4.1 Experiment Setup

We simulate our experiments in MATLAB 2016.b on 64-bit Windows 10 Pro, which with 2.4GHz CUP and 16GB RAM. The source code of SPA algorithm is distributed by the National Renewable Energy Laboratory in 2014 [39], and programmed in C++. The proposed algorithms partly utilize operators in GA toolkit package *gatbx* [47],

such as SUS selection, nonlinear ranking, etc. The NSGA-II source code is also available in Mathworks [56]. The dataset of observed coordinates $(x_i,\ y_i)$ is used as a benchmark which contains seventeen shadow records for every three minutes' interval in a period of forty-eight minutes, which put in Appendix A1.

4.2 Feasibility Analysis

In this section, we briefly analyzed the relationship between the observed variables through experiments. The major objective is to inspect the convexity of objective functions and constraints. In the thesis, the inequality constraints for our objective functions indicate that the optimal latitude/longitude are searched in restrained spaces, and the equality constraints for the SPA algorithm $g(\sigma,\ \varphi,\ t)$ plays the role to calculate feasible solution spaces. For a minimized optimization problem, the feasible set should be a convex set, and objective functions should with the feature of convexity as well.

The relationship among shadow, elevation/azimuth angle and $g(\sigma,\ \varphi,\ t)$ are discussed shadow vertex coordinates model (section 3.1.2). Elevation/azimuth angle are not going to be direct parameters in cost functions, but shadow's vertex coordinates $(x_i,\ y_i)$ are. As the SPA algorithm is composed of intricate trigonometric functions, which is periodic and does not show the nature of convex. However, for the problem to be solved, we only need to verify the convexity for x-y coordinates and shadow length of the relationship between sunrise and sunset. In terms of this concern, we conducted an experiment to graphical demonstrate the changes of shadow length and its coordinates $(x_i,\ y_i)$ in a daytime. The observed coordinates $(x_i,\ y_i)$ and shadow length s is depicted by figure 4.1.

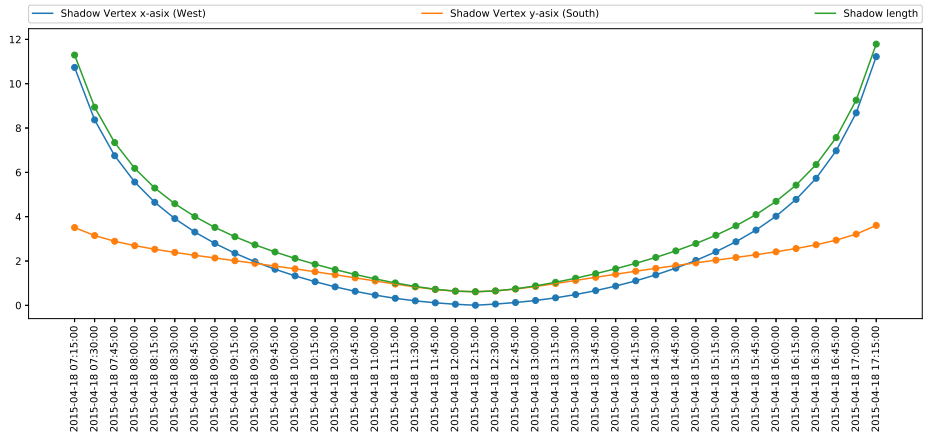

Figure 4.1: Relationship between Time and Shadow Length/Coordinates

4.2.1 Relationship between Time and Shadow Length

The relationship between the time and the shadow length shows in Figure 4.1, we can find that at 12:15 pm when the shadow length is the shortest. Combined with variation of Elevation angle in Figure 4.2, when the sun has the highest elevation angle, the shadow length is minimal. Followed by changes in the shadow, x-y coordinates length is the shortest as well. By observing curves in Figure 4.1, x-coordinate (West) length and shadow length show similar length change rate, but the y-coordinate (South) length changes much flatter than shadow length and x-coordinate length. In a daytime, the length of the shadow shows from long to short, and then by trend of short to long. In fact, these conclusions are also very consistent with common sense. While for the concern of $g(\sigma,\ \varphi,\ t)$ overall convexity, we fit observed data to a quadratic equation (polynomial of degree = 2), then verify a necessary and sufficient condition for it to be convex on that interval is that the second derivative $g^{''}(t) \geqslant 0$ for all x in $\mathbb{R}^n$. From observed data in Appendix A1, the corresponding quadratic regression

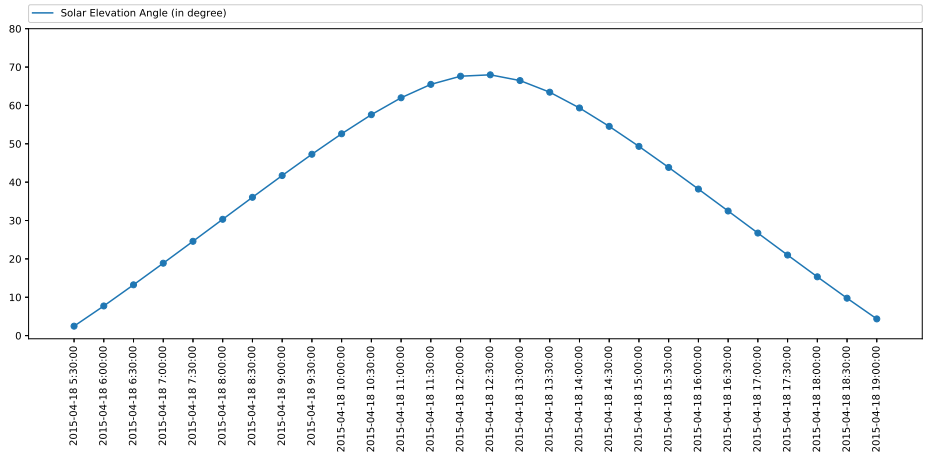

Figure 4.2: Variation Trend of Solar Elevation Angle by Time

equation for time and shadow length at the specific time is

$$g(t) = s = 0.3492t^2 - 8.5478t + 56.0380$$

which represents convexity apparently. Therefore, we conclude that the function set $g(\sigma,\ \varphi,\ t)$ also has the same properties as function $g(t)$.

Overall, we examine the variation trends for all constrains in a daytime period. Results indicate coordinates $(x_i,\ y_i)$ and $g(\sigma,\ \varphi,\ t)$ both meet the convex properties. Meanwhile, as all cost functions that we proposed are based on ℓ^1 and ℓ^2, norms are absolutely satisfying convexity. Therefore, for the verification of the objective equation and the corresponding constraints properties, we verified that the proposed optimization NGSPA is feasible.

4.3 GSPA and NGSPA Simulation

As we proposed the positioning methods in two stages - single cost function and multiple-cost functions. In order to estimate which cost function can estimate the

most proximal position (latitude/longitude) in both GSPA and NGSPA, we evaluate each of them with three metrics: (i) range of feasible solution spaces for objective functions with each cost function, and errors between estimated results and observed values; (ii) accuracy comparison between proposed positioning methods and existing methods; (iii) the time cost of proposed models to estimate the results.

4.3.1 Feasible Solution Spaces Range and Errors

In this subsection, we evaluate results spaces range of objective functions by each cost function and the error between the estimated results and the observed values.

Single Objective Function

From the shadow vertex model, it is very intuitive to find, without knowing the length of the object, all variables are involved in that three cost functions can be obtained in SPA and shadow vertex coordinate model for any possible latitude and longitude. For the same reason, all SPA and shadow vertex coordinate functions combined the same objective function model in the GSPA and NGPSA algorithms that we proposed in this thesis. At first we try to use a simple method to compare the superiority of them. In figure 4.3, we compare previously mentioned three types of cost functions with known position and shadow coordinates length, but without object length, to estimate the robustness and granularity of them. As cost functions are based on ℓ^1 norm and ℓ^2 norm significantly trends to zero over time, but the ratio of elevation angle with two adjacent measuring moment fluctuates apparently. Curves change of tangent slope for ℓ^1 and ℓ^2 norm, ℓ^2 shows slightly gentle than ℓ^1. Although the observation data is in a short period, the overall trends of cost function(I) (ℓ^1 norm) and cost function(II) (ℓ^2 norm) curves are much smoother than the cost function

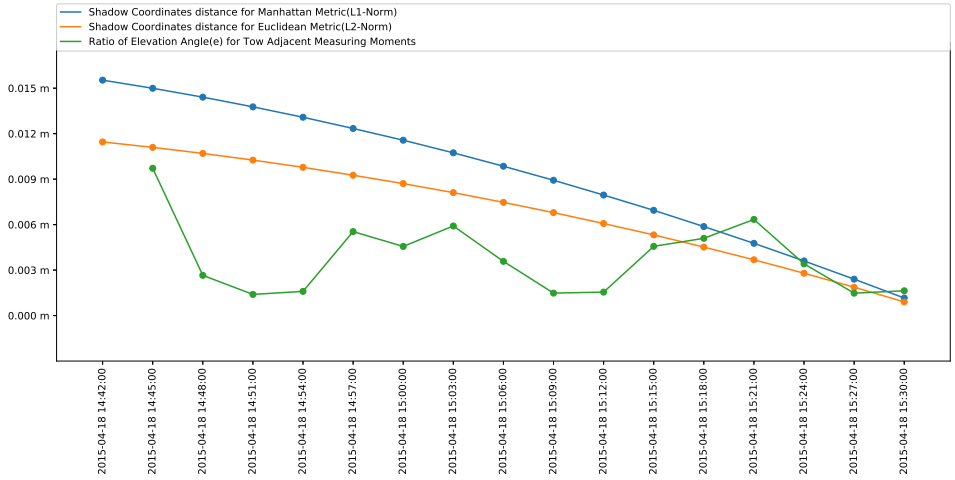

Figure 4.3: Cost Function Comparison without Object Length

(III). However, Cost function (III) fluctuates in the least value space than ℓ^1 and ℓ^2. According to the current analysis, we can not explain which cost function has obvious advantages or disadvantages. The reason for the decline and fluctuation of curves, we infer that it is affected by the local weather conditions on the solar refraction or measuring error.

Single Objective Function Spaces and Errors

As the previous naive method can not be verified the superiority of cost functions, we imply each cost function in GSPA. Due to GA is random search based, the results for each run are not the same. Meanwhile, for a high granularity search, the search space is relatively large. Results return to three decimal places, and the lower and upper bounds for this two search spaces latitude and longitude are $(-90°, 90°)$ and $(-180°, 180°)$ respectively. In experiments, we run GSPA 100 times for each cost functions. Results summary list in table 4.1 and 4.2, corresponding data distributions are plot by box-plot.

Functions	**Maximum**	**Third Quartile**	**Median**	**First Quartile**	**Minimum**	**Errors**
Objective (I)	49.903°	45.991°	42.661°	36.180°	30.003°	+3.118°
Objective (II)	49.926°	44.182°	40.493°	37.227°	30.426°	+0.457°
Objective (III)	49.480°	40.579°	38.913°	34.256°	30.425°	−0.630°

Table 4.1: Single Objective Function Latitude Spaces Summary

Functions	**Maximum**	**Third Quartile**	**Median**	**First Quartile**	**Minimum**	**Errors**
Objective (I)	129.078°	123.171°	117.899°	108.367°	100.799°	+1.666°
Objective (II)	129.011°	120.415°	117.532°	108.840°	100.211°	+1.299°
Objective (III)	129.541°	122.643°	115.664°	106.748°	100.401°	−0.569°

Table 4.2: Single Objective Function Longitude Spaces Summary

The actual latitude and longitude is $(39.543°, 116.233°)$, and we calculate errors by comparing the median of estimations and actual values. From the results, objective function (I) and (II) are both higher than the latitude and longitude of the observation spot, but objective function (I) is closer to the actual vales than the values of objective function (II). Whereas, estimated latitude and longitude by the objective function (III) are lower than the observation values. It should to be mentioned that we select 95% of the data from all runs to take out exceptional values.

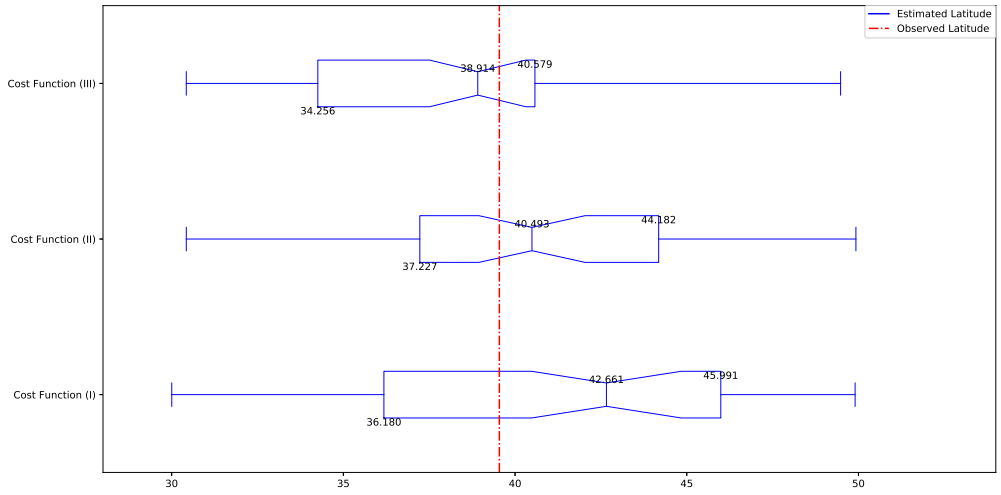

Figure 4.4: Objective Functions Space of GSPA for Latitude

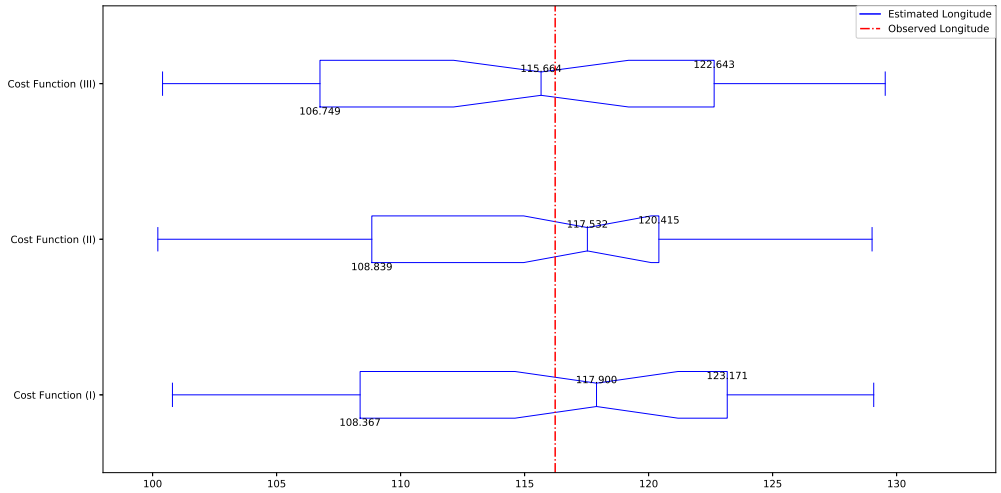

Figure 4.5: Objective Functions Space of GSPA for Longitude

As latitude and longitude are in $(-90°, 90°)$ and $(-180°, 180°)$ from south to north and west to east respectively (figure 3.1). Therefore, the positive errors represent the observed latitude is northward and the observed longitude is eastward, vice versa.

Furthermore, from the spaces summary Table 4.1 and 4.2, as well as boxplot Figure 4.4 and 4.5, we found the feasible solution space of objective function (I) fully covers the space of the objective function (II). In terms of this reason, for the minimization problem, we conclude that objective function (II) is better than objective function (I).

Multiobjective Functions by NSGA-II

As the objective function (I) has the worst feasible solutions space in three, we only use objective function (II) and (III) for multiobjective optimization in proposed algorithms NGSPA. In the same way, we run the model 1000 times and analyze statistical values by boxplot distribution. By comparing the estimated results of multiobjectives with objective function (II) and (III), we found errors diminished for both latitude and longitude. In short, errors have an average range from about ± 1.5 significantly narrow down to around ± 0.2.

Functions	**Maximum**	**Third Quartile**	**Median**	**First Quartile**	**Minimum**	**Errors**
Objective (II)	49.926°	44.182°	40.493°	37.227°	30.426°	+0.457°
Objective (III)	49.480°	40.579°	38.913°	34.256°	30.425°	−0.630°
Multiobjectives	49.879°	42.292°	39.332°	34.766°	32.064°	−0.211°

Table 4.3: Objective Functions Latitude Spaces Comparison

Functions	**Maximum**	**Third Quartile**	**Median**	**First Quartile**	**Minimum**	**Errors**
Objective (II)	129.011°	120.415°	117.532°	108.840°	100.211°	+1.299°
Objective (III)	129.541°	122.643°	115.664°	106.748°	100.401°	−0.569°
Multiobjectives	128.767°	121.400°	116.056°	109.962°	101.450°	−0.177°

Table 4.4: Objective Functions Longitude Spaces Comparison

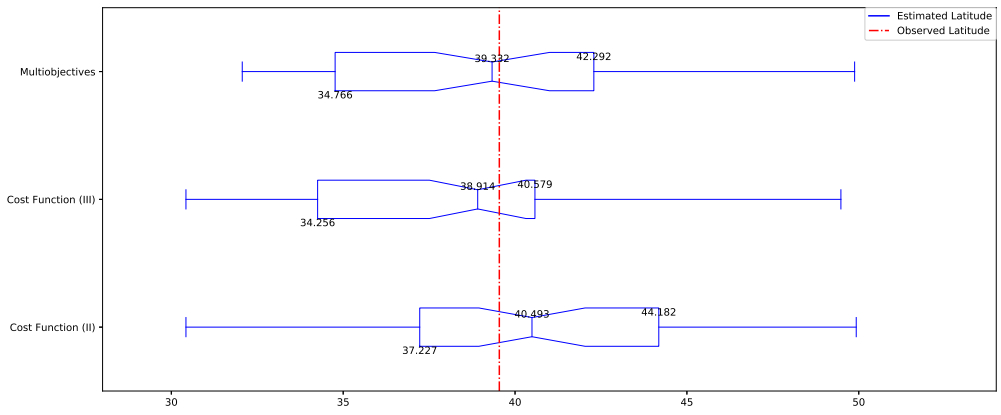

Figure 4.6: Objective Functions Space of NSGA-II in GSPA for Latitude

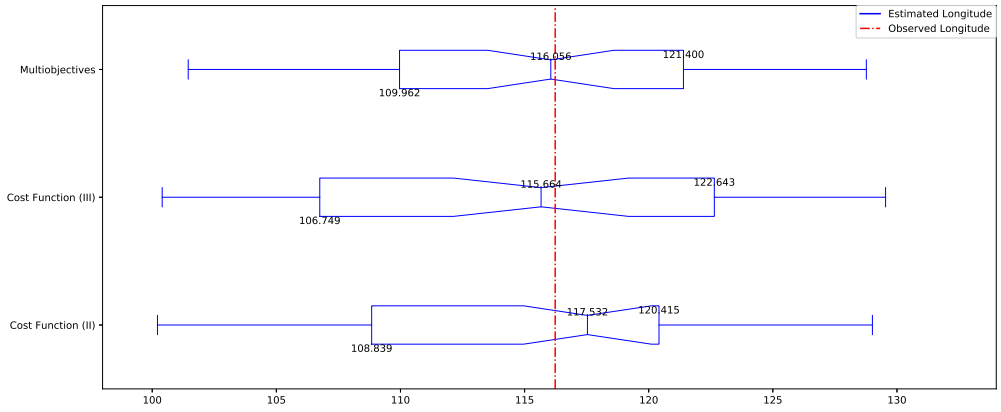

Figure 4.7: Objective Functions Space of NSGA-II in GSPA for Longitude

Similarly, from the Table 4.3 and 4.4, as well as boxplot Figure 4.6 and 4.7, there are no obvious change for the range of feasible solution spaces among two single objective and miltiobjective functions. However, the median of the estimated latitude and longitude are apparently closer to the observed value.

Positioning Method	Accuracy	Usage	Device Data
GPS	$\pm 8m$	Outdoor	Required
Assisted GPS	$5 - 50m$	Indoor and Outdoor	Required
Cell-ID	$100 - 3000m$	Indoor and Outdoor	Required
GSM Cell Tower Triangulation	$\pm 25m$	Outdoor	Required
WLAN Positioning	$20 - 30m$	Indoor and Outdoor	Required
Simple GA Positioning	$\pm 3.0^\circ$	Indoor and Outdoor	Not Required
GSPA	$\pm 1.5^\circ$	Indoor and Outdoor	Not Required
NGSPA	$\pm 0.2^\circ$	Indoor and Outdoor	Not Required

Table 4.5: Accuracy of Positioning Methods

4.3.2 Accuracy

In this subsection, we conduct the proposed positioning method and the traditional positioning methods for comparison, and results show in Table 4.5.

Accuracy Comparison

Traditional methods that rely on mobile devices and network devices, the most precise method is GPS in all positioning models, $\pm 8m$ almost can not be ignored. The method by Cell-ID is the worst one in traditional methods, but it supports indoor positioning as Assisted GPS and WALN positioning.

The errors for proposed methods GSPA and NGSPA are measured by degrees, which can not be directly converted to meters due to the spherical surface of the earth. For these two methods, the error of GSPA is greater than NGSPA. Meanwhile, we also conduct the experiments on simple GA for positioning, while the estimated values are far from the observation values. In experiments, latitude and longitude search spaces are in $(-90^\circ, 90^\circ)$ and $(-180^\circ, 180^\circ)$. However, NGSPA with limited resources, $\pm 0.2^\circ$ is a highly acceptable error range. To a certain extent, NGSPA can perform better than Cell-ID positioning.

Accuracy for Reducing the Search Space

Practical search spaces usually can be reduced with effective information in photos or videos. Therefore, in order to verify whether the accuracy will be affected by the range of the search space, we designed another experiment. Each round reducing the range of search spaces for both east-west and north-south direction which is changed $-5°$ for each, and running the program 100 times to get the feasible solutions space. In statistical phase, we calculate the error between median of estimated latitude/longitude and observation values.

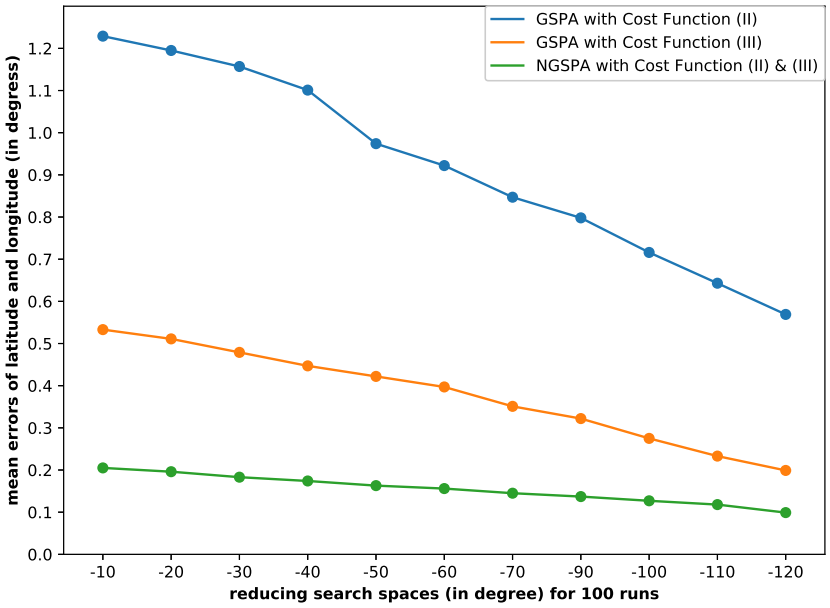

Figure 4.8: Accuracy for Reducing the Search Space

The trend of accuracy for reducing the search space is shown in Figure 4.8. The experimental results indicates that as the search space decreases, the computational accuracy of the model increases. Although NGSPA is slightly smoother than GSPA, the final outcome of NGSPA is still the best. Meanwhile, the error is less than

± 0.1 when latitude and longitude search spaces are in $(-30°, 30°)$ and $(-60°, 60°)$. Therefore, reducing the search space in a practical way to improve the accuracy.

4.3.3 Performance

Performance evaluation of each objective functions with a single cost function and multi-objectives, and the corresponding experimental results show in Figure 4.9. In the results, there is no obvious growth for three different cost functions when runs increase. Similarly, the multi-objectives method NSGA-II shows the same tendency as the single objective. Whereas, the NSGA-II spen more time than single objective on the same runs. In addition, the growth of NSGA-II surges with the runs build up. To facilitate investigation and exploration, there are a few digital forensic tools that can perform a useful 5 minutes analysis, most of them can perform the whole analysis in as few as 20 minutes [43]. However, both of our approaches - GSPA and NGSPA, can complete geo-estimation within five minutes. Although the complexity for calculation of multi-objectives is higher than a single objective function, the total time spent for 100 runs is still guaranteed to be completed within five minutes.

The comparison of simple GA positioning, GSPA and NGSPA are shown in Figure 4.10. Although NGSPA spend the highest, the time reach 5 minutes (300 seconds) when it is close to 500 runs. On the other hand, the improved GA in GSPA does not produce much more time consumption than simple GA. In general, the GA based positioning models have a significant performance advantage over device dependent methods. From the comprehensive accuracy and performance point of view, GPS is the most precise way for all positioning models. However, in addition to relying on devices, GPS only supports outdoor positioning. Likewise, assisted GPS, Cell-ID and WLAN can achieve accurate indoor positioning. If such information can be obtained from photos and videos, these traditional methods are very useful. Otherwise, these

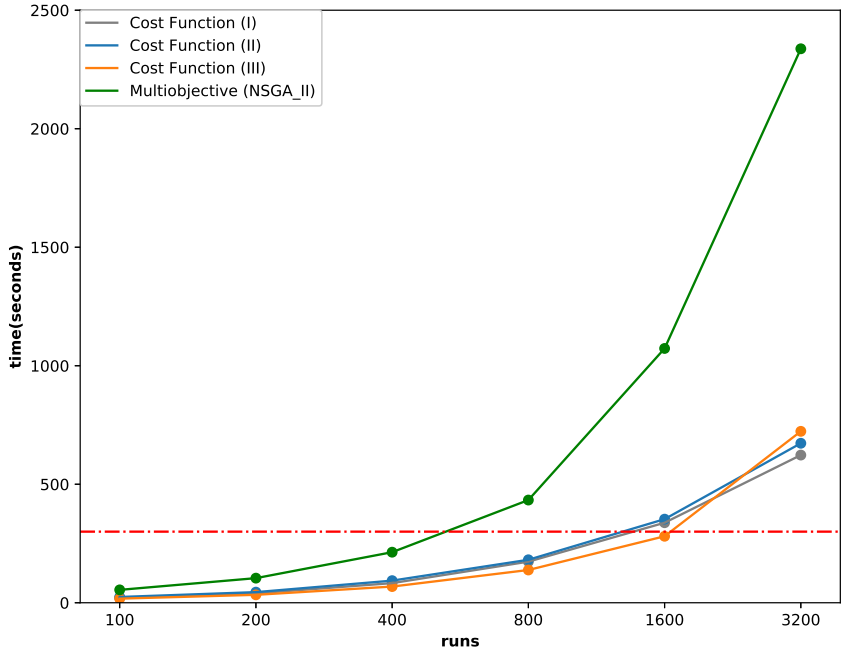

Figure 4.9: GSPA and NGSPA Time Cost Compression

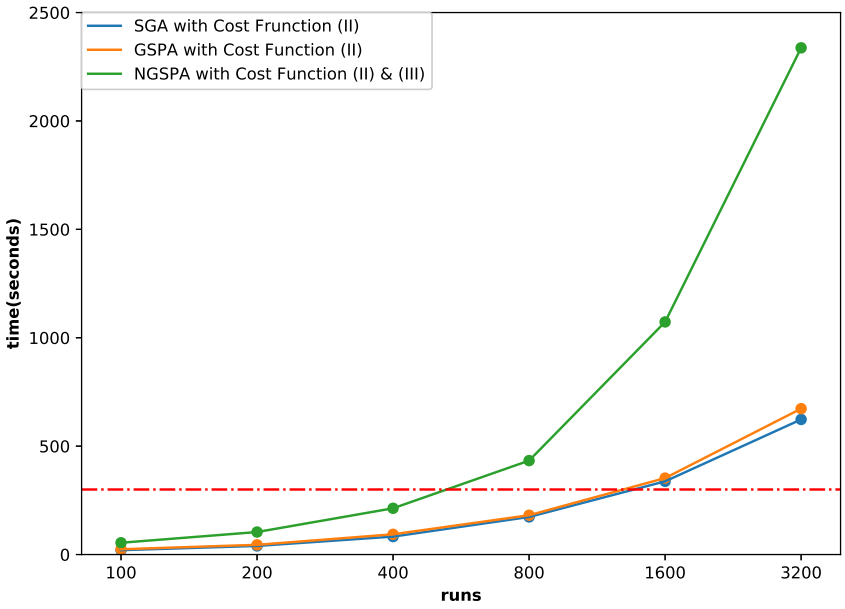

Figure 4.10: Existing Simple GA, GSPA and NGSPA Time Cost Compression

ways will be worthless if they fail to get such devices' data. The forensics will also be in trouble of a lack of information.

Performance for Reducing the Search Space

Likewise, we also verify whether the accuracy will be affected by the range of the search space, and results show in Figure 4.11. But different from accuracy, the time cost has not changed significantly due to the narrowing of the search space. The possible reason is the time costs of GSPA and NGSPA are mainly consumed by SPA algorithms.

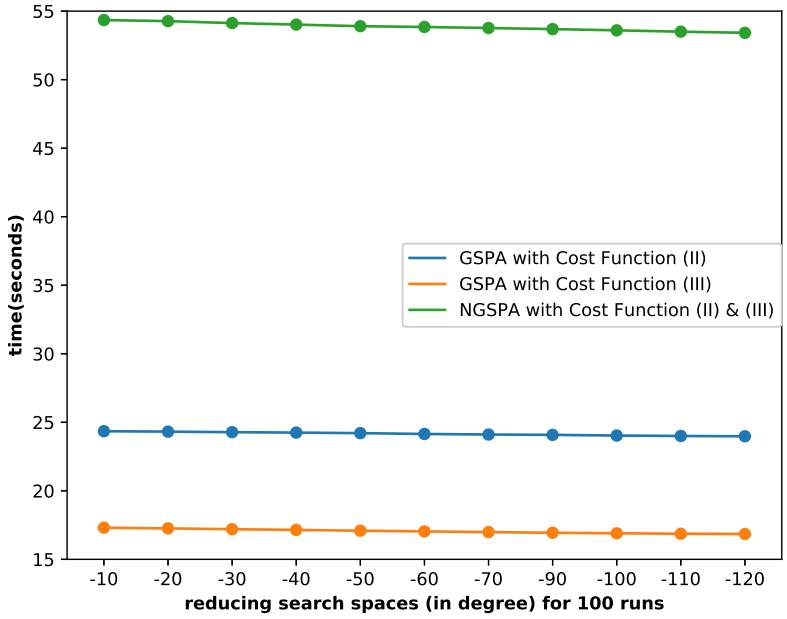

Figure 4.11: Performance for Reducing the Search Space

Overall, the proposed methods get rid of the shackles of traditional methods, it's feasibility only affected by the sun. From the resources perspective, they only acquire data from the picture and video itself, do not need any mobile and network devices' data. In spite of no obvious advantages in accuracy from our experiments, the results

are extremely acceptable. Meanwhile, this is a huge breakthrough on the time cost for positioning models, and play an important supplementary role for existing methods.

4.4 SPA Variable Interaction Analysis

In section 3.1.2, we proposed the shadow vertex coordinate model to calculate shadow length s by SPA. In this model, the main changes of shadows are length and direction. The shadow length changes with elevation angle, and shadow direction changes with azimuth angle. The purpose of this experiment is to understand the relationship between latitude, longitude, elevation angle and azimuth angle.

By SPA algorithm, we generate an experimental datasets. In this artificial datasets, we set the observation point $(39.543°, 116.233°)$ as the center, then extend observer latitude and longitude $30°C$ with the interval of $3°C$. In this case, observer latitude in the range of [24.543, 54.233] from eastward to west, observer longitude in the range of [101.233, 131.233] southward to north, and set a constant value for object length is 3 meters.

4.4.1 Latitude, Longitude and Elevation Angle

Then we established a three-dimension (3D) space and used experimental data to plot the varying surface of these three variable at a specific time and spot. Observer latitude (β), longitude (Θ) and elevation angle (e) show in the figure 4.12 and 4.13 respectively. By observation, the elevation angle gradually get smaller with the growth of longitude, but slightly increased with the growth of latitude.

The Pearson coefficient of elevation angle (e) with latitude and longitude are $cor(s, \beta)$ = 0.425 and $cor(s, \Theta)$ = -0.892 respectively. Apparently, the three variables are non-linear relationship. Although (e, φ) indicates the positive correlation and (e, σ)

indicates the negative correlation, the shadow length changes are more influenced by longitude.

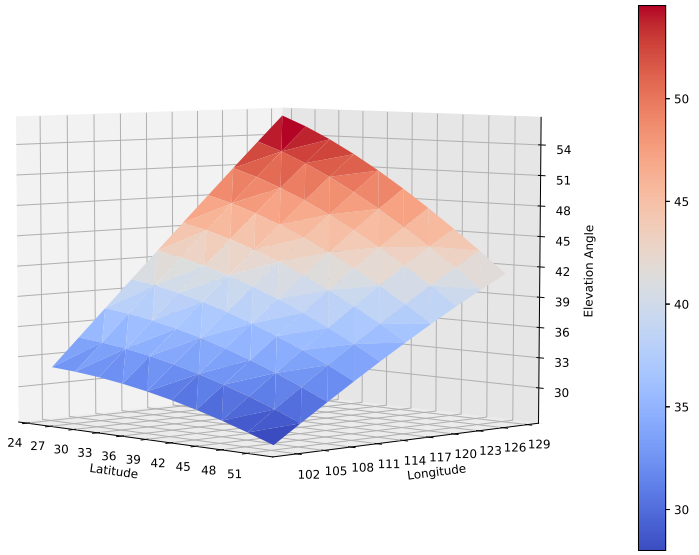

Figure 4.12: (φ, Θ) and (e) relationship 1

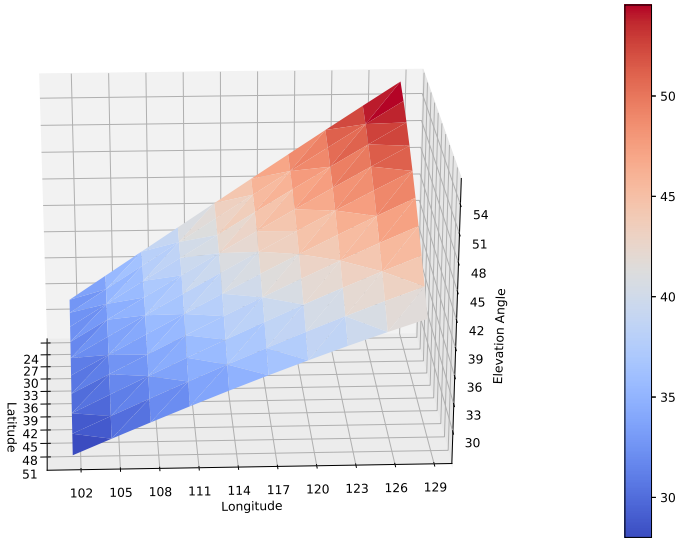

Figure 4.13: (φ, Θ) and (e) relationship 2

4.4.2 Latitude, Longitude and Azimuth Angle

In the same way, we established an three-dimension (3D) space and used experimental data to plot the varying surface of observer latitude, longitude and azimuth angle at a specific date time point. From the graphs figure 4.14 and 4.15, the azimuth angle gradually decrease with the growth of longitude, while the decrease trends is more sharper when latitude getting smaller. Meanwhile, azimuth angle apparently increased with the growth of latitude.

The Pearson coefficient of azimuth angle with latitude and longitude are $cor(\Gamma, \varphi)$ = 0.439 and $cor(\Gamma, \sigma)$ = -0.885 respectively. Similarly, these three variables are non-linear relationship as well. The $cor(\Gamma, \varphi)$ indicates the positive correlation and $cor(\Gamma, \sigma)$ indicates the negative correlation, the Azimuth angle changes are more influenced by longitude.

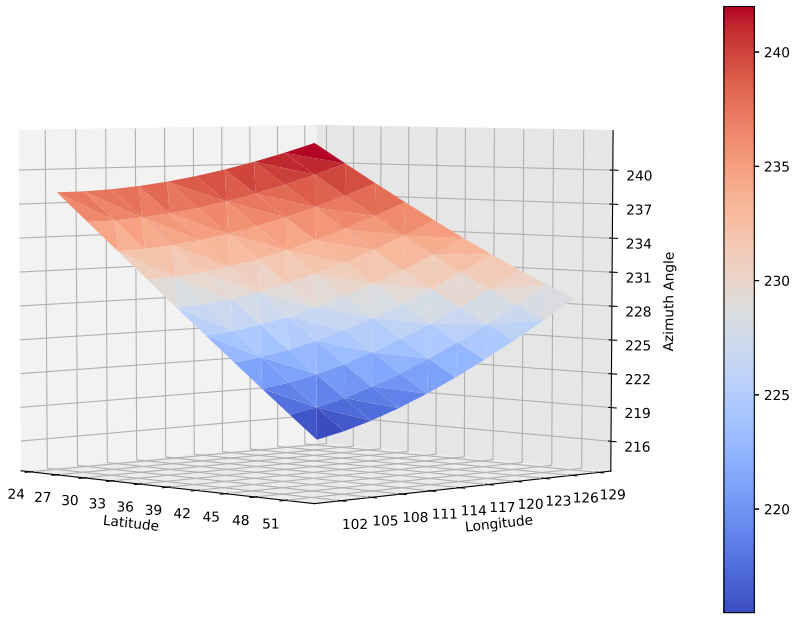

Figure 4.14: (φ, Θ) and (Γ) relationship 1

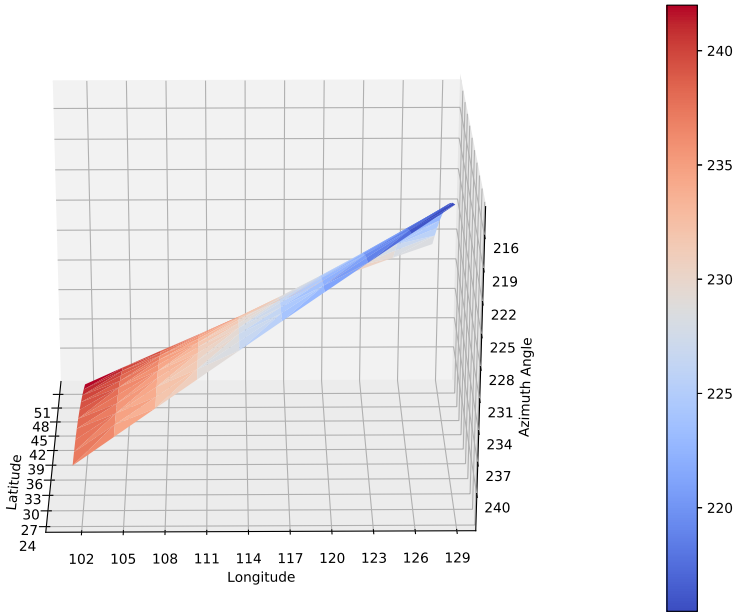

Figure 4.15: (φ, Θ) and (Γ) relationship 2

Chapter 5

Conclusions and Future Work

The purpose of GI positioning stems from the digital forensics for reconnaissance, which requires agile and accurate results for positioning estimation. However, existing positioning methods extremely rely on mobile devices and network devices to obtain data. These methods also need to correlate information across diverse independent sources to extract geographical positioning. Therefore, processes of these methods cause analysis cannot be completed in an instant. Consider these limitations, the most important principle to resolve these problems is without using device and network data. By deeply utilizing the uncommonly used information in pictures and videos, such as object's shadow, the elevation angle and the azimuth angle, we propose a novel model of shadow positioning model.

First of all, we design the GSPA model with SPA and improved GA algorithms to construct a shadow positioning model. By simulation, we verify the feasibility of GSPA, and it also spends low time cost. Meanwhile, the estimated results show substantial accuracy for this method, the prerequisite resources and calculating time are reduced compared with traditional methods.

In addition, since GA is a guided random search, the feasible solution is non-unique.

To improve the optimal solution and optimize feasible solutions space, we add NSGA-II to GSPA model, and named the new model as NGSPA. The simulation results show the accuracy of NGSPA has significantly improved over GSPA, which trades off performance. In spite of this, NSGPA still spends less time for estimation than existing methods. We believe the method will facilitate positioning processes in digital forensics.

For future work, to improve the overall performance of our models, we can use kernel method [6] or functional analysis [63] to address complicated trigonometric equation in SPA to a dual-function. On the other hand, we can optimize the feasible solution spaces by better methods or deep learning methods to improve the accuracy of the model.

Bibliography

[1] Ayers, R. P., Jansen, W., Delaitre, A. M., and Moenner, L. Cell phone forensic tools: an overview and analysis update. *NIST Interagency/Internal Report (NISTIR)-7387* (2007).

[2] Back, T. Selective pressure in evolutionary algorithms: A characterization of selection mechanisms. In *Evolutionary Computation, 1994. IEEE World Congress on Computational Intelligence., Proceedings of the First IEEE Conference on* (1994), IEEE, pp. 57–62.

[3] Baker, J. E. Reducing bias and inefficiency in the selection algorithm. In *Proceedings of the second international conference on genetic algorithms* (1987), pp. 14–21.

[4] Baryamureeba, V., and Tushabe, F. The enhanced digital investigation process model. In *Proceedings of the Fourth Digital Forensic Research Workshop* (2004), pp. 1–9.

[5] Bergstra, J., and Bengio, Y. Random search for hyper-parameter optimization. *Journal of Machine Learning Research 13*, Feb (2012), 281–305.

[6] Bishop, C. M. *Pattern recognition and machine learning*. springer, 2006.

[7] Blickle, T., and Thiele, L. A comparison of selection schemes used in genetic algorithms, 1995.

[8] Boyd, S., and Vandenberghe, L. *Convex optimization*. Cambridge university press, 2004.

[9] Brockhoff, D., and Zitzler, E. Are all objectives necessary? on dimensionality reduction in evolutionary multiobjective optimization. In *Parallel Problem Solving from Nature-PPSN IX*. Springer, 2006, pp. 533–542.

[10] Cao, X., Zhao, H., Wang, C., and Zhang, W. Image composite authentication using a single shadow observation. *Science China Information Sciences 58*, 9 (2015), 1–13.

[11] Carrier, B., and Spafford, E. H. An event-based digital forensic investigation framework. In *Digital forensic research workshop* (2004), pp. 11–13.

[12] Casey, E. *Handbook of digital forensics and investigation*. Academic Press, 2009.

[13] Chou, J., and Ramchandran, K. Robust turbo-based data hiding for image and video sources. In *Multimedia and Expo, 2002. ICME'02. Proceedings. 2002 IEEE International Conference on* (2002), vol. 2, IEEE, pp. 565–568.

[14] Coello, C. A. C., Pulido, G. T., et al. A micro-genetic algorithm for multiobjective optimization. In *EMO* (2001), vol. 1, Springer, pp. 126–140.

[15] Corne, D. W., Jerram, N. R., Knowles, J. D., and Oates, M. J. Pesa-ii: Region-based selection in evolutionary multiobjective optimization. In *Proceedings of the 3rd Annual Conference on Genetic and Evolutionary Computation* (2001), Morgan Kaufmann Publishers Inc., pp. 283–290.

[16] Corne, D. W., Knowles, J. D., and Oates, M. J. The pareto envelope-based selection algorithm for multiobjective optimization. In *International Conference on Parallel Problem Solving from Nature* (2000), Springer, pp. 839–848.

[17] CUMCM-2015. Solar shadow positioning. `http://mcm.blyun.com/`, 2015. [Online; accessed 09-March-2017].

[18] De Jong, K. Genetic algorithms: a 30 year perspective. *Perspectives on Adaptation in Natural and Artificial Systems 11* (2005).

[19] Deb, K. Multi-objective genetic algorithms: Problem difficulties and construction of test problems. *Evolutionary computation 7*, 3 (1999), 205–230.

[20] Deb, K. *Multi-objective optimization using evolutionary algorithms*, vol. 16. John Wiley & Sons, 2001.

[21] Deb, K., Pratap, A., Agarwal, S., and Meyarivan, T. A fast and elitist multiobjective genetic algorithm: Nsga-ii. *IEEE transactions on evolutionary computation 6*, 2 (2002), 182–197.

[22] Fogarty, T. Varying the probability of mutation in the genetic algorithm. In *Proc. 3rd Int'l Conf. on Genetic Algorithms* (1989), Morgan Kaufmann, pp. 104–109.

[23] Fonseca, C. M., Fleming, P. J., et al. Genetic algorithms for multiobjective optimization: Formulationdiscussion and generalization. In *Icga* (1993), vol. 93, pp. 416–423.

[24] Friedland, G., and Sommer, R. Cybercasing the joint: On the privacy implications of geo-tagging. In *HotSec* (2010), pp. 1–6.

[25] Goldberg, D. E. Genetic algorithms in search, optimization, and machine learning, 1989. *Reading: Addison-Wesley* (1989).

[26] Goldberg, D. E., and Deb, K. A comparative analysis of selection schemes used in genetic algorithms. *Foundations of genetic algorithms 1* (1991), 69–93.

[27] Goldberg, D. E., and Holland, J. H. Genetic algorithms and machine learning. *Machine learning 3*, 2 (1988), 95–99.

[28] Hernández-Díaz, A. G., Santana-Quintero, L. V., Coello, C. A. C., and Molina, J. Pareto-adaptive ε-dominance. *Evolutionary computation 15*, 4 (2007), 493–517.

[29] Holland, J. H. *Adaptation in natural and artificial systems: an introductory analysis with applications to biology, control, and artificial intelligence*. MIT press, 1992.

[30] Hoog, A., and Gaffaney, K. iphone forensics. *Via Forensics White paper* (2009).

[31] Horn, J., Nafpliotis, N., and Goldberg, D. E. A niched pareto genetic algorithm for multiobjective optimization. In *Evolutionary Computation, 1994. IEEE World Congress on Computational Intelligence., Proceedings of the First IEEE Conference on* (1994), Ieee, pp. 82–87.

[32] Horowitz, E., and Sahni, S. *Fundamentals of computer algorithms*. Computer Science Press, 1978.

[33] Joines, J. A., and Houck, C. R. On the use of non-stationary penalty functions to solve nonlinear constrained optimization problems with ga's. In

Evolutionary Computation, 1994. IEEE World Congress on Computational Intelligence., Proceedings of the First IEEE Conference on (1994), IEEE, pp. 579–584.

[34] Kakar, P., and Sudha, N. Verifying temporal data in geotagged images via sun azimuth estimation. *IEEE Transactions on Information Forensics and Security 7*, 3 (2012), 1029–1039.

[35] Knowles, J. D., and Corne, D. W. Approximating the nondominated front using the pareto archived evolution strategy. *Evolutionary computation 8*, 2 (2000), 149–172.

[36] Konak, A., Coit, D. W., and Smith, A. E. Multi-objective optimization using genetic algorithms: A tutorial. *Reliability Engineering & System Safety 91*, 9 (2006), 992–1007.

[37] Kramer, O. *Genetic Algorithm Essentials*, vol. 679. Springer, 2017.

[38] Kretz, T., Bönisch, C., and Vortisch, P. Comparison of various methods for the calculation of the distance potential field. *Pedestrian and evacuation dynamics 8* (2008).

[39] Laboratory, N. R. E. Nrel's solar position algorithm (spa). `https://midcdmz.nrel.gov/spa/`, 2014. [Online; accessed 19-June-2017].

[40] Laumanns, M., Thiele, L., Deb, K., and Zitzler, E. Combining convergence and diversity in evolutionary multiobjective optimization. *Evolutionary computation 10*, 3 (2002), 263–282.

[41] Liu, H., Darabi, H., Banerjee, P., and Liu, J. Survey of wireless indoor positioning techniques and systems. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews) 37*, 6 (2007), 1067–1080.

[42] Marino, A. Algorithms for biological graphs: Analysis and enumeration. *Bulletin of EATCS 3*, 114 (2014).

[43] Maus, S., Höfken, H., and Schuba, M. Forensic analysis of geodata in android smartphones. In *International Conference on Cybercrime, Security and Digital Forensics, http://www.schuba.fh-aachen.de/papers/11-cyberforensics.pdf* (2011).

[44] Mühlenbein, H., and Schlierkamp-Voosen, D. Predictive models for the breeder genetic algorithm i. continuous parameter optimization. *Evolutionary computation 1*, 1 (1993), 25–49.

[45] Osman, I. H., and Kelly, J. P. Meta-heuristics: an overview. In *Meta-heuristics*. Springer, 1996, pp. 1–21.

[46] Pohlheim, H. Ein genetischer algorithmus mit mehrfachpopulationen zur numerischen optimierung. *at-Automatisierungstechnik 43*, 3 (1995), 127–135.

[47] Pohlheim, H. The genetic and evolutionary algorithm toolbox (geatbx) for matlab. `http://www.geatbx.com/`, 2007. [Online; accessed 11-July-2017].

[48] Poisel, R., and Tjoa, S. Forensics investigations of multimedia data: A review of the state-of-the-art. In *IT Security Incident Management and IT Forensics (IMF), 2011 Sixth International Conference on* (2011), IEEE, pp. 48–61.

[49] Postavaru, S., Stoean, R., Stoean, C., and Caparros, G. J. Adaptation of deep convolutional neural networks for cancer grading from histopatho-

logical images. In *International Work-Conference on Artificial Neural Networks* (2017), Springer, pp. 38–49.

[50] Reda, I., and Andreas, A. Solar position algorithm for solar radiation applications. *Solar energy 76*, 5 (2004), 577–589.

[51] Roussev, V., and Richard III, G. G. Breaking the performance wall: The case for distributed digital forensics. In *Proceedings of the 2004 digital forensics research workshop* (2004), vol. 94.

[52] Sarma, K. A. D. J. J. Generation gaps revisited. *Foundations of Genetic Algorithms 1993 (FOGA 2) 2* (2014), 19.

[53] Schaffer, J. D., and Morishima, A. An adaptive crossover distribution mechanism for genetic algorithms. In *Genetic Algorithms and their Applications: Proceedings of the Second International Conference on Genetic Algorithms* (1987), Hillsdale, NJ: Lawrence Erlbaum Associates, Inc, pp. 36–40.

[54] Schlierkamp-Voosen, D., and Mühlenbein, H. Predictive models for the breeder genetic algorithm. *Evolutionary Computation 1*, 1 (1993), 25–49.

[55] Schmitendorf, W., and Forrest, S. *Using genetic algorithms for controller design: Simultaneous stabilization and eigenvalue placement in a region.* Department of Computer Science, College of Engineering, University of New Mexico, 1992.

[56] Selvaraj, T. Matlab code for constrained nsga ii - dr.s.baskar, s. tamilselvi and p.r.varshini. `https://www.mathworks.com/matlabcentral/`, 2015. [Online; accessed 11-July-2017].

[57] Srinivas, M., and Patnaik, L. M. Adaptive probabilities of crossover and mutation in genetic algorithms. *IEEE Transactions on Systems, Man, and Cybernetics 24*, 4 (1994), 656–667.

[58] Srinivas, N., and Deb, K. Muiltiobjective optimization using nondominated sorting in genetic algorithms. *Evolutionary computation 2*, 3 (1994), 221–248.

[59] Sumathi, S., Hamsapriya, T., and Surekha, P. *Evolutionary intelligence: an introduction to theory and applications with Matlab*. Springer Science & Business Media, 2008.

[60] Syswerda, G. Simulated crossover in genetic algorithms. *Foundations of genetic algorithms 2* (1993), 239–255.

[61] Thomee, B., Shamma, D. A., Friedland, G., Elizalde, B., Ni, K., Poland, D., Borth, D., and Li, L.-J. Yfcc100m: The new data in multimedia research. *Communications of the ACM 59*, 2 (2016), 64–73.

[62] Tuson, A. L., and Ross, P. Adapting operator probabilities in genetic algorithms. *Master's thesis, Department of Artificial Intelligence, Univeristy of Edinburgh* (1995).

[63] Ulbricht, J. Representing functional data as smooth functions.

[64] Wenger, E. *Artificial intelligence and tutoring systems: computational and cognitive approaches to the communication of knowledge*. Morgan Kaufmann, 2014.

[65] Whitley, D., and Starkweather, T. Genitor ii: A distributed genetic algorithm. *Journal of Experimental and Theoretical Artificial Intelligence 2*, 3 (1990), 189–214.

[66] Whitley, L. D. *Foundations of genetic algorithms 2*, vol. 2. morgan Kaufmann, 1993.

[67] Whitley, L. D., et al. The genitor algorithm and selection pressure: Why rank-based allocation of reproductive trials is best. In *ICGA* (1989), vol. 89, pp. 116–123.

[68] Yandamuri, S., Srinivasan, K., and Murty Bhallamudi, S. Multiobjective optimal waste load allocation models for rivers using nondominated sorting genetic algorithm-ii. *Journal of water resources planning and management 132*, 3 (2006), 133–143.

[69] Zalzala, A. M., and Fleming, P. J. *Genetic algorithms in engineering systems*, vol. 55. Iet, 1997.

[70] Zitzler, E., Deb, K., and Thiele, L. Comparison of multiobjective evolutionary algorithms: Empirical results. *Evolutionary computation 8*, 2 (2000), 173–195.

[71] Zitzler, E., Laumanns, M., and Thiele, L. Spea2: Improving the strength pareto evolutionary algorithm.

[72] Zitzler, E., and Thiele, L. Multiobjective optimization using evolutionary algorithms - a comparative case study. In *international conference on parallel problem solving from nature* (1998), Springer, pp. 292–301.

[73] Zitzler, E., and Thiele, L. Multiobjective evolutionary algorithms: a comparative case study and the strength pareto approach. *IEEE transactions on Evolutionary Computation 3*, 4 (1999), 257–271.

Appendix A

Experimental Data

The appended dataset that we used in experiments is from CUMCM [17]. The data has been image preprocessed by Gray-Level Co-Occurrence Matrix (GLCM) to get the (x_i, y_i) coordinates. Table A.1 lists 17 records between 48 minutes to record the changes of shadow length. Figure A.1 illustrates an example of the observation photo.

Local Time	14:42	14:45	14:48	14:51	14:54	14:57	15:00	15:03
x-coordinate (m)	1.036	1.0699	1.1038	1.1383	1.1732	1.2087	1.2448	1.2815
y-coordinate (m)	0.4972	0.5029	0.5085	0.5142	0.5198	0.5255	0.5311	0.5368
15:06	15:09	15:12	15:15	15:18	15:21	15:24	15:27	15:30
1.3180	1.3568	1.3955	1.4349	1.4751	1.5160	1.5577	1.6003	1.6438
0.5426	0.5483	0.5541	0.5598	0.5657	0.5715	0.5774	0.5833	0.5892

Table A.1: Shadow observation Coordinates and Time

Figure A.1: The Object and Shadow of the Observation