

การปรับปรุงอัตราการหาตัวเลขปริศนาจำนวนเต็มสำหรับการหาตำแหน่งแบบจอน์ในจีพีเอสโดย  
ใช้ขั้นตอนวิธีเชิงพันธุกรรม

นายปณิธาน ศรีนวลดี



จุฬาลงกรณ์มหาวิทยาลัย  
CHULALONGKORN UNIVERSITY

บทคัดย่อและแฟ้มข้อมูลฉบับเต็มของวิทยานิพนธ์ตั้งแต่ปีการศึกษา 2554 ที่ให้บริการในคลังปัญญาจุฬาฯ (CUIR)

เป็นแฟ้มข้อมูลของนิสิตเจ้าของวิทยานิพนธ์ ที่ส่งผ่านทางบัณฑิตวิทยาลัย

วิทยานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรดุษฎีบัณฑิต  
The abstract and full text of theses from the academic year 2011 in Chulalongkorn University Intellectual Repository (CUIR)

สาขาวิชาวิศวกรรมสำรวจ ภาควิชาวิศวกรรมสำรวจ  
are the thesis authors' files submitted through the University Graduate School.

คณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย

ปีการศึกษา 2558

ลิขสิทธิ์ของจุฬาลงกรณ์มหาวิทยาลัย

THE IMPROVEMENT OF FIXED AMBIGUITY RESOLUTION RATE IN GPS  
KINEMATIC POSITIONING MODE USING GENETIC ALGORITHMS

Mr. Panithan Srinuandee



A Dissertation Submitted in Partial Fulfillment of the Requirements  
for the Degree of Doctor of Philosophy Program in Survey Engineering

Department of Survey Engineering

Faculty of Engineering

Chulalongkorn University

Academic Year 2015

Copyright of Chulalongkorn University



ปณิธาน ศรีนวนลดี : การปรับปรุงอัตราการหาตัวเลขปริศนาจำนวนเต็มสำหรับการหาค่าตำแหน่งแบบ  
 จลน์ในจีพีเอสโดยใช้ขั้นตอนวิธีเชิงพันธุกรรม (THE IMPROVEMENT OF FIXED  
 AMBIGUITY RESOLUTION RATE IN GPS KINEMATIC POSITIONING MODE USING  
 GENETIC ALGORITHMS) อ.ที่ปรึกษาวิทยานิพนธ์หลัก: ศ. ดร. เฉลิมชนม์ สติระพจน์, 108 หน้า.

สองตัวแปรสำคัญที่ส่งผลต่อความแม่นยำของการรังวัดหาค่าพิกัดโดยใช้ข้อมูลเฟสของคลื่นส่งในจีพี  
 เอสคือ เทคนิคการหาค่าต่างของข้อมูล และขั้นตอนการหาตัวเลขปริศนาจำนวนเต็ม ซึ่งเทคนิคการหาค่าต่างของ  
 ข้อมูลครั้งที่สอง ได้ถูกนำมาใช้งานอย่างแพร่หลายเพื่อลดค่าการรบกวนชนิดต่างๆในการรังวัดโดยใช้จีพีเอส  
 อย่างไรก็ตามค่าการรบกวนดังกล่าวยังมีหลงเหลืออยู่และส่งผลกระทบต่ออัตราการหาตัวเลขปริศนาจำนวนเต็ม  
 โดยเฉพาะการใช้จีพีเอสในการหาค่าตำแหน่งแบบจลน์ ดังนั้นข้อมูลที่มีการรบกวนชนิดต่างๆ รวมถึงความเทียม  
 บางดวง ต้องถูกค้นหาและคัดแยกออกจากการประมวลผลข้อมูล ซึ่งเป็นขั้นตอนที่ต้องการผู้ประมวลผลข้อมูลที่  
 สามารถแก้ไขและคัดกรองข้อมูลที่มีปัญหาออกได้ โดยปกติขั้นตอนในการคัดกรองข้อมูลที่มีปัญหาจะมีลักษณะ  
 แบบลองผิดลองถูก โดยผู้ประมวลผลข้อมูลจะต้องประมวลผลจนกว่าจะได้ผลลัพธ์ตามที่ต้องการ หลายครั้งที่การ  
 ประมวลผลในลักษณะนี้ใช้เวลามากและต้องการผู้ประมวลผลข้อมูลที่มีประสบการณ์สูง

ในงานวิจัยนี้ หลักการขั้นตอนวิธีเชิงพันธุกรรม ได้ถูกพัฒนาขึ้นเพื่อแก้ไขปัญหาดังที่ได้กล่าวถึงข้างต้น  
 โดยหลักการขั้นตอนวิธีเชิงพันธุกรรมจะถูกนำมาใช้คัดเลือกความเทียมเพื่อปรับปรุงอัตราการหาตัวเลขปริศนาจำนวน  
 เต็มสำหรับการหาค่าตำแหน่งแบบจลน์ในจีพีเอส จากผลการทดลองพบว่าขั้นตอนวิธีเชิงพันธุกรรมสามารถเพิ่ม  
 อัตราการหาตัวเลขปริศนาจำนวนเต็มได้ อย่างไรก็ตามการทดลองข้างต้นไม่ได้ทดสอบกับข้อมูลการรังวัดแบบ  
 จลน์ในภาคปฏิบัติ ดังนั้นจึงได้มีการทดลองเพิ่มเติมรวมถึงมีการปรับปรุงประสิทธิภาพการทำงานของหลักการ  
 ขั้นตอนวิธีเชิงพันธุกรรม โดยนำเทคนิคการเลื่อนคาบเวลาไปประกอบกับหลักการขั้นตอนวิธีเชิงพันธุกรรม เพื่อเพิ่ม  
 ประสิทธิภาพในการปรับปรุงอัตราการหาตัวเลขปริศนาจำนวนเต็ม ด้วยวิธีการเลื่อนรอบคาบเวลาในการ  
 คัดเลือกข้อมูลในขั้นตอนการประมวลผลข้อมูล ซึ่งเทคนิคใหม่ที่น่าสนใจนี้ได้ถูกทดสอบกับข้อมูลการรังวัดแบบ  
 จลน์ในภาคปฏิบัติจากระบบ โมบายแมพพิง ผลลัพธ์ที่ได้แสดงให้เห็นว่าเทคนิคใหม่ที่น่าสนใจนี้สามารถเพิ่มอัตรา  
 การหาตัวเลขปริศนาจำนวนเต็มได้มากกว่าเทคนิคที่ใช้หลักการขั้นตอนวิธีเชิงพันธุกรรมเพียงอย่างเดียวรวมถึงวิธี  
 มาตรฐานในการประมวลผลข้อมูลการหาค่าตำแหน่งแบบจลน์

ผลลัพธ์จากการทดลองแสดงให้เห็นอย่างชัดเจนว่าเทคนิคใหม่ที่น่าสนใจสามารถนำมาใช้ในการ  
 ปรับปรุงอัตราการหาตัวเลขปริศนาจำนวนเต็มสำหรับการหาค่าตำแหน่งแบบจลน์ในจีพีเอสโดยไม่ต้องมีความ  
 ช่วยเหลือจากผู้ประมวลผลข้อมูลได้

ภาควิชา วิศวกรรมสำรวจ

ลายมือชื่อนิพนธ์ .....

สาขาวิชา วิศวกรรมสำรวจ

ลายมือชื่อ อ.ที่ปรึกษาหลัก .....

ปีการศึกษา 2558

# # 5371835021 : MAJOR SURVEY ENGINEERING

KEYWORDS: GPS / PPK / GA / SLIDING WINDOWS

PANITHAN SRINUANDEE: THE IMPROVEMENT OF FIXED AMBIGUITY RESOLUTION RATE IN GPS KINEMATIC POSITIONING MODE USING GENETIC ALGORITHMS. ADVISOR: PROF. CHALERMCHON SATIRAPOD, Ph.D., 108 pp.

The two key factors in achieving high-precision positioning results from GPS carrier phase observations are the data differencing technique and the ambiguity resolution process. A double differencing technique has been widely used to reduce biases in GPS observations. However, un-modeled biases can persist, and these reduce the number of ambiguity-fixed solutions, especially in GPS kinematic positioning mode. Therefore, noisy or unwanted GPS satellites must be identified and removed from the data processing step. This requires users to edit or filter noisy data in the processing step. Data filtering is essentially a trial-and-error process, with users reprocessing the data until satisfactory ambiguity-fixed solutions are obtained. Often, data filtering is a time-consuming process that requires the skills of an experienced user.

In this research, a genetic algorithm (GA) was developed to overcome the aforementioned problem. This GA is used to optimize the selection of GPS satellites in order to improve the number of ambiguity-fixed solutions in GPS kinematic positioning mode. Experimental results demonstrate that the GA can enhance the number of ambiguity-fixed solutions without user-assistance in the data processing step. However, the experiment did not use practical kinematic data. Therefore, further investigations were conducted to improve the ability of the GA. The sliding windows technique was integrated with the GA to enhance the number of ambiguity-fixed solutions by varying the finite-length selection windows in the data processing step. This new proposed methodology was tested with raw GPS kinematic observation data from a practical mobile-mapping system. The results indicate that the proposed method produces considerably more ambiguity-fixed solutions than either the GA alone or the standard GPS kinematic post-processing.

The experimental results clearly demonstrate that the proposed methods can be used to improve the ambiguity-fixed solutions in GPS kinematic positioning mode without the need for user-assistance in the data processing step.

Department: Survey Engineering                      Student's Signature .....

Field of Study: Survey Engineering                      Advisor's Signature .....

Academic Year: 2015

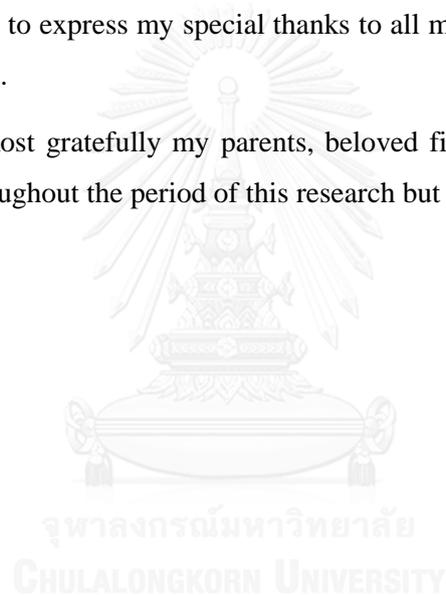
## ACKNOWLEDGEMENTS

I would like to express my deep gratitude to Professor Chalermchon Satirapod, Ph.D. my advisor, who motivate and also give me an opportunity with the scholarship to study in the doctoral program, Department of Survey Engineering, Faculty of Engineering, Chulalongkorn University. I also felt in debt for his patient guidance, enthusiastic encouragement and useful critiques of this research work.

I would like to express my thankful and acknowledge the Faculty of Engineering, Chulalongkorn University for awarding me a scholarship to pursue my Ph.D.

I would like to express my special thanks to all my colleague and supervisor for their help and support.

Finally, I most gratefully my parents, beloved fiancée and all friends for their support, not only throughout the period of this research but always.



## CONTENTS

	Page
THAI ABSTRACT .....	iv
ENGLISH ABSTRACT.....	v
ACKNOWLEDGEMENTS .....	vi
CONTENTS.....	vii
LIST OF TABLES .....	ix
LIST OF FIGURES .....	x
LIST OF ABBREVIATIONS.....	xiii
Chapter 1.....	1
1.1 Background and Rationale .....	1
1.2 Objectives .....	2
1.3 Scope of Thesis.....	3
1.4 Research Methodology .....	4
1.5 Contributions of Thesis .....	6
1.6 Outline of Thesis .....	6
Chapter 2.....	8
2.1 Global Positioning System (GPS).....	8
2.2 Genetic Algorithm (GA).....	32
2.3 Sliding Window .....	44
2.4 Literature Reviews.....	53
Chapter 3.....	56
3.1 Introduction.....	56
3.2 GPS Kinematic Post-Processing using a GA.....	57
3.3 Experimental Data .....	70
3.4 Experimental Results and Analysis .....	71
Chapter 4.....	78
4.1 Introduction.....	78
4.2 Combination of Sliding Windows and GA in GPS Kinematic Positioning Mode.....	79

	Page
4.3 Experimental Data .....	85
4.4 Experimental Results and Analysis .....	85
Chapter 5 .....	94
5.1 Conclusions .....	94
5.2 Recommendations .....	96
REFERENCES .....	98
APPENDIX 1 .....	101
VITA .....	108



## LIST OF TABLES

	Page
<b>Table 2.1</b> The summarize features of each GPS satellite in constellation .....	9
<b>Table 2.2</b> The example of string and its fitness.....	36
<b>Table 2.3</b> The example of schema and its order.....	38
<b>Table 2.4</b> The example of schema and its length .....	39
<b>Table 2.5</b> The example of schema and its properties .....	39
<b>Table 2.6</b> The matching between knot number (k) and solution number (j) based on the computation of figure 2.7.....	50
<b>Table 2.7</b> The matching between knot number (k) and solution number (j) based on the computation of figure 2.9.....	52
<b>Table 3.1</b> The example of bit strings and their meaning.....	62
<b>Table 3.2</b> The configuration of GA within MATLAB.....	63
<b>Table 3.3</b> Q Flag values and their meaning.....	68
<b>Table 3.4</b> Satellites available in each 1-h session and satellites excluded by the GA.....	72
<b>Table 3.5</b> Comparison of solution types given by RTKLib, GA, and RTKLiborGA.....	73
<b>Table 3.6</b> Classification interval of improvement.....	74
<b>Table 3.7</b> Best satellite combinations identified by the GA.....	75
<b>Table 4.1</b> Official coordinates of the benchmark at the base station (WGS84).....	86
<b>Table 4.2</b> Main configuration parameters used in rnx2rtkp of RTKLib .....	88
<b>Table 4.3</b> Sliding window parameters used in the GA-SW method .....	89
<b>Table 4.4</b> Comparison of results given by the standard, GA-based, and GA-SW method.....	90
<b>Table 4.5</b> Comparison between window width and number of fixed-ambiguity solutions given by the GA-SW method .....	91

## LIST OF FIGURES

	Page
<b>Figure 1.1</b> Relationship between the scope of this research in 1.3.1, 1.3.2, and 1.3.3.....	4
<b>Figure 1.2</b> Summary of the methodology used in this research.....	5
<b>Figure 2.1</b> The processes of GPS phase data processing (adapted from (Rizos 1997)).....	24
<b>Figure 2.2</b> The chromosome, gene and DNA in the nucleus of cell (from <a href="http://igbiologyy.blogspot.com/2014/03/chromosomes-dna-genes-and-alleles.html">http://igbiologyy.blogspot.com/2014/03/chromosomes-dna-genes-and-alleles.html</a> , last access: 23/11/2015).....	33
<b>Figure 2.3</b> Operating procedure of standard Genetic Algorithm .....	35
<b>Figure 2.4</b> The crossover process performed at the crossover point as shown in black line .....	35
<b>Figure 2.5</b> The mutation process performed at the black arrow .....	36
<b>Figure 2.6</b> The visualize of schemata as hyperplanes in 3 dimensional cube.....	40
<b>Figure 2.7</b> The diagram of sliding-window in time-fixed point of view. The window is five knot width and the step ratio is 1. The reference time for solution N is shown as “t” (adopted from (Foster, Bevis et al. 2005)).....	46
<b>Figure 2.8</b> The diagram of sliding-window in window-fixed point of view. The window is five knot width and the step ratio is 1. ....	46
<b>Figure 2.9</b> The diagram of sliding-window in window-fixed point of view. The window is nine knot width and the step ratio is 2. (Adopted from (Foster, Bevis et al. 2005)).....	51
<b>Figure 3.1</b> Operating procedure of GPS post-processing using GUI.....	58
<b>Figure 3.2</b> GUIs of the GPS post-processing software: (a) User can select data, time span, solution and so on; (b) User can select processing options such as satellite system, positioning-mode, troposphere model, and so on.....	59
<b>Figure 3.3</b> Operating procedure for command line GPS post-processing tool .....	60
<b>Figure 3.4</b> Interaction between GUI post-processing tool or third-party program and command line post-processing tool through the configuration file and post-processing results .....	61

	Page
<b>Figure 3.5</b> Operating procedure of optimized GA with PPK.....	65
<b>Figure 3.6</b> GA evolution composed of three operators (selection, crossover, and mutation).....	66
<b>Figure 3.7</b> Example of a binary string representing individuals in the population pool .....	67
<b>Figure 3.8</b> Example of a decoded binary string of excluded satellites .....	67
<b>Figure 3.9</b> Example of a modified RTKLib configuration file, with the excluded satellites highlighted .....	67
<b>Figure 3.10</b> Example showing the post-processed results, with Q Flag values highlighted .....	68
<b>Figure 3.11</b> Operating procedure of RTKLiborGA .....	69
<b>Figure 3.12</b> Modified event flag (from 0 to 2) to indicate moving platform (Kinematic) .....	70
<b>Figure 3.13</b> Top: the PKKT base station; bottom: the DPT9 roving station .....	71
<b>Figure 3.14</b> Classification of improvement in results given by the GA-based method.....	74
<b>Figure 3.15</b> Differences between the solutions and the known coordinates for session D (Top: North component; middle: East component; bottom: Height component).....	76
<b>Figure 3.16</b> Differences between the solutions and the known coordinates for session F (Top: North component; middle: East component; bottom: Height component).....	76
<b>Figure 3.17</b> Differences between the solutions and the known coordinates for session Q (Top: North component; middle: East component; bottom: Height component).....	77
<b>Figure 4.1</b> Operating procedures for the combination of sliding windows and GA in GPS kinematic positioning mode .....	79
<b>Figure 4.2</b> Sub-operating procedures of the modified GA-PPK Optimization.....	80
<b>Figure 4.3</b> Difference in sliding window solutions between Median Epoch and Last Epoch .....	84
<b>Figure 4.4</b> Example of time window options in rnx2rtkp.....	84

	Page
<b>Figure 4.5</b> Benchmark of base station located in front of the Survey Building, Department of Land, Pakkret, Nonthaburi province, Thailand .....	86
<b>Figure 4.6</b> Instruments attached to the rooftop of a car. The Trimble R8 used as the rover for this study is shown in the lower left.....	87
<b>Figure 4.7</b> Base Station (red triangle) and trajectory of the rover station (thick yellow line running from lower right to upper left) along the pond in Muangthong Thani, Thailand (Source: Bing Map).....	89
<b>Figure 4.8</b> Results given by the standard method .....	92
<b>Figure 4.9</b> Results given by the GA-based method .....	92
<b>Figure 4.10</b> Results given by the GA-SW method.....	93



## LIST OF ABBREVIATIONS

CODE	Center for Orbit Determination in Europe
DGNSS	Differential GNSS
DGPS	Differential GPS
DNA	DeoxyriboNucleic Acid
EKF	Extended Kalman Filter
GA	Genetic Algorithm
GDOP	Geometric Dilution Of Precision
GLONASS	GLObalnaya NAVigazionnaya Sputnikonaya Sistema
GNSS	Global Navigation Satellite System
GPS	Global Positioning System
GPST	GPS Time
GUI	Graphic User Interface
IGS	International GNSS Service
ILS	Integer Least Squares
LAMBDA	Least-squares AMBIGuity Decorrelation Adjustment
LGO	Leica Geomatic Office
MATLAB	MATrix LABoratory
MMS	Mobile Mapping System
NRT	Near Real Time
NRTK	Network Real Time Kinematic
PPK	Post Processing Kinematic
PWV	Precipitable Water Vapor
RINEX	Receiver Independent Exchange
RTK	Real Time Kinematic
SVN	Satellite Vehicle Number
UTC	Coordinated Universal Time
ZTD	Zenith Total Delay

## **Chapter 1**

### **Introduction**

This research examines the use of genetic algorithm (GA) to select the combination of GPS satellites that generate higher ambiguity-fixed solution for use in static GPS post-processing. Moreover, the developed GA is combined with a sliding windows technique. The resulting system can be used in practical GPS kinematic post-processing.

This chapter is structured as follows. First, the background and rationale for this research are introduced. Second, the research objectives are listed, and the scope of this research is discussed. Then, the research methodology is described in detail, and the contributions of this research are summarized. Finally, the structure of this thesis is summarized.

#### **1.1 Background and Rationale**

There are many data processing techniques related to kinematic positioning, e.g., real-time kinematic (RTK), network real-time kinematic (NRTK), and post-processing kinematic (PPK) (Roggero 2003). Each technique has its own strengths and weaknesses. There are various applications for PPK, as it provides better accuracy and higher stability than the other methods when used with a moving platform. The roving stations that use RTK or NRTK can get very precise and accurate instantaneous positioning in the field, whereas PPK roving stations have to record the raw observation data and perform some post-processing technique at a later point to obtain accurate positioning. However, RTK and NRTK require simplex and duplex data communication, respectively. In contrast, PPK roving stations do not require any data communication, and can provide higher precision and accuracy than RTK and NRTK. The many strengths of PPK, including its achievable precision and accuracy of positioning, have seen it applied in scenarios such as hydrographic surveys, high-precision trajectory determination (Tang, Turley et al. 1994) and building and dam monitoring (Hudnut and Behr 1998, Kijewski-Correa, Kochly et al. 2004). Nevertheless, there are some limitations to PPK, e.g., the high precision and accuracy

of positioning in PPK can only be achieved in good environments, e.g., a place with minimum multipath from nearby constructions, rivers and dense canopies. In practice, it is very difficult for field users based at roving stations to choose the perfect environment for their survey. As a result, some bias may contaminate the recorded raw observation data, and this can reduce the precision and accuracy of PPK. Thus, during post-processing, this bias must be filtered to achieve the desired precision and accuracy (Dach, Hugentobler et al. 2007). The steps of searching and filtering the bias require well-trained and experienced users, and are considered time-consuming procedures.

Given the abovementioned problem, any methodology that can be applied to automatically search for and filter out the bias will help to increase the efficiency of GPS kinematic post-processing. Moreover, well-trained and experienced users will be able to examine and analyze the results in more detail. GA is a suitable tool for this problem. They have many advantages, e.g., they are easy to develop, can be applied to complex problems, and do not require the calculation of a determinant. In comparison with the brute-force approach which checking all candidates that satisfies to the problem (Levitin 2009), GA evolved in the search space and checking only some of candidates in order to get global optima. As a result GA will spend shorter computation time than brute-force approach especially when more GNSS satellites are added in to account. In addition, to enhance the effectiveness of GA, the sliding windows technique was integrated with a GA to apply in dynamic environments.

## **1.2 Objectives**

- 1.2.1 To develop a methodology to use GA in order to increase the number of ambiguity-fixed solutions in GPS kinematic positioning.
- 1.2.2 To develop a methodology to integrate the sliding windows technique with the GA for use in GPS kinematic post-processing in order to increase the number of ambiguity-fixed solutions.
- 1.2.3 To compare the number of ambiguity-fixed solutions given by standard GPS kinematic post-processing with those given by the methods proposed in 1.2.1 and 1.2.3.

### 1.3 Scope of Thesis

- 1.3.1 Apply a GA to GPS kinematic post-processing to increase the number of ambiguity-fixed solutions.

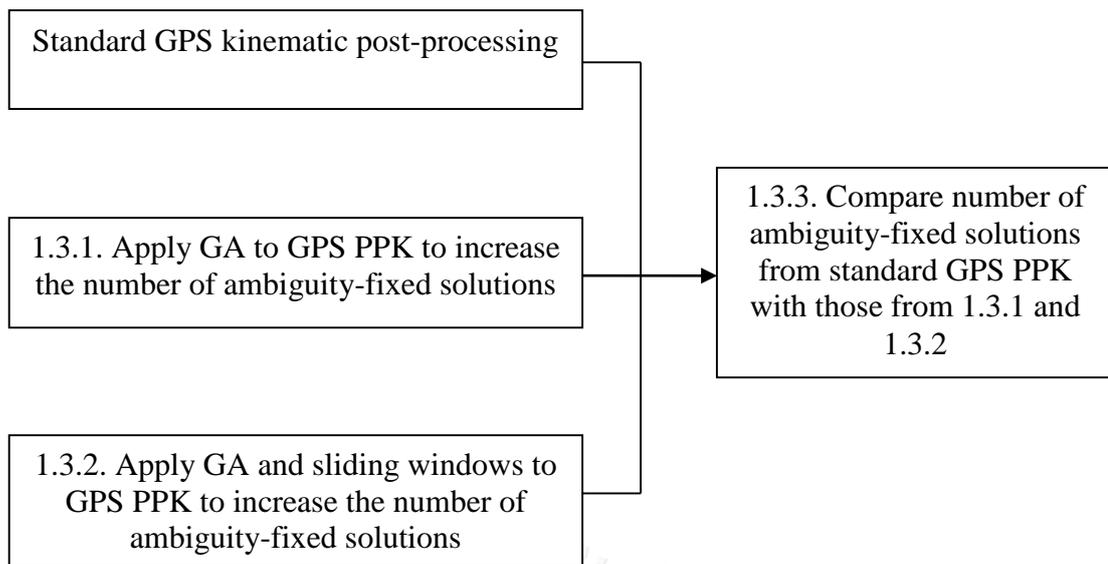
The scope of research in this topic is related to the optimization of the combination of GPS satellites in order to increase the number of GPS ambiguity-fixed solutions given by GPS kinematic post-processing. For this purpose, a standard GA is combined with a GPS post-processing engine. The GA gives an optimized result that is fed forward to the GPS post-processing engine. The results of GPS kinematic post-processing are then fed back to the GA to check the outcome and evolve to get better results. This interaction continues until the desired results are obtained.

- 1.3.2 Combine a GA with sliding windows for GPS kinematic post-processing to increase the number of ambiguity-fixed solutions.

The scope of research in this topic is similar to that of the previous topic. The method in 1.3.1 is enhanced by integrating the sliding windows technique to assist the GA optimization in more dynamic environments. Thus, to test the effectiveness of this method, practical kinematic data are needed.

- 1.3.3 Compare the number of ambiguity-fixed solutions given by standard GPS kinematic post-processing with those given by the methods proposed in 1.3.1 and 1.3.2.

To clarify the purpose of this research, the scope of work described in 1.3.1, 1.3.2, and 1.3.3 is depicted in Figure 1.1.



**Figure 1.1** Relationship between the scope of this research in 1.3.1, 1.3.2, and 1.3.3

- 1.3.4 The data used in this research are based on short baseline only, and the data processing is based on post-processing only.

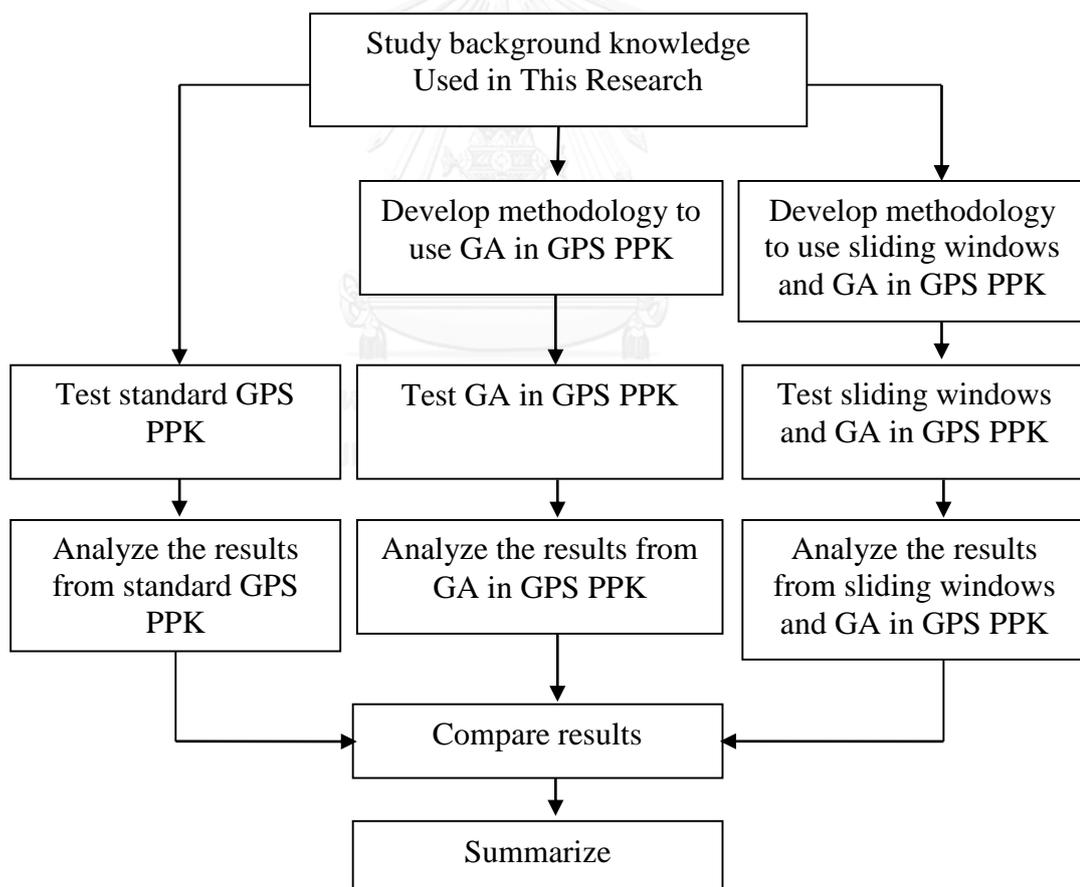
## 1.4 Research Methodology

The methodology used in this research can be enumerated as follows.

- 1.4.1 Study the background knowledge relevant to this research.
  - 1.4.1.1 Study ambiguity-resolution techniques used in GPS data processing.
  - 1.4.1.2 Study techniques related to GPS kinematic positioning.
  - 1.4.1.3 Study the details of GA.
  - 1.4.1.4 Study the details of the sliding windows technique.
- 1.4.2 Test the standard GPS kinematic post-processing.
- 1.4.3 Analyze the results from 1.4.2.
- 1.4.4 Develop the methodology to use a GA in GPS kinematic post-processing in order to increase the number of ambiguity-fixed solutions.
- 1.4.5 Test the proposed methodology from 1.4.4.

- 1.4.6 Analyze the results from 1.4.5.
- 1.4.7 Develop the methodology to integrate sliding windows with the GA for use in GPS kinematic post-processing in order to increase the number of ambiguity-fixed solutions.
- 1.4.8 Test the proposed methodology from 1.4.7.
- 1.4.9 Analyze the results from 1.4.8.
- 1.4.10 Compare the results from 1.4.3, 1.4.6, and 1.4.8.
- 1.4.11 Summarize the results and prepare the thesis.

For ease of understanding and to clarify the overall research methodology, a flowchart of the above steps is shown in Figure 1.2.



**Figure 1.2** Summary of the methodology used in this research

## 1.5 Contributions of Thesis

In this study, the methods proposed in 1.4.4 and 1.4.7 has been investigated. The contributions of this research can be summarized as follows.

- 1.5.1 An increase in the number of ambiguity-fixed solutions in GPS kinematic post-processing using a GA that removes the need for manual data screening.
- 1.5.2 An increase in the number of ambiguity-fixed solutions in GPS kinematic post-processing by integrating the sliding windows and GA without the need for manual data screening.
- 1.5.3 The proposed method can be modified to accept other navigation satellite systems in order to increase the redundancy, availability, and accuracy.
- 1.5.4 In the near future, there will be more than enough GNSS (Global Navigation Satellite System) satellites. The proposed method can be used to select the optimum satellite to achieve the desired result by modifying the objective function.

## 1.6 Outline of Thesis

Chapter 2 discusses the background knowledge related to this research, including GPS, GA, and the sliding windows technique. This chapter also introduces some previous research on the application of GA or sliding windows technique to GPS, giving an insight into how the aforementioned techniques can be integrated.

Chapter 3 describes the methodology of optimizing the satellite combination in kinematic positioning mode with the aid of a GA. The proposed method is applied using raw observation data from one reference station, but the positioning mode flag in the raw observation data is changed to kinematic mode. The results from this experiment are presented and discussed. Finally, the data are analyzed.

Chapter 4, the methodology for integrating the GA with a sliding windows technique is developed. The resulting technique is used to optimize the ambiguity-resolution rate in GPS kinematic mode. The system is evaluated using the data prepared in the previous chapter as well as practical kinematic GPS observation data from the

mobile mapping system (MMS). The results from this experiment are then presented and analyzed.

Chapter 5 is dedicated to some concluding remarks on the proposed methodology discussed in Chapters 3 and 4. This chapter also includes a series of recommendations for future research.



## Chapter 2

### Concepts and Theories Involved

The main objective in this research introduced in previous chapter is to increase the number of ambiguity fixed solution in GPS PPK using either GA alone or the integration between GA and Sliding Windows. Therefore, in this chapter detail about concept and theories involved in this research will be provided together with some literature reviews.

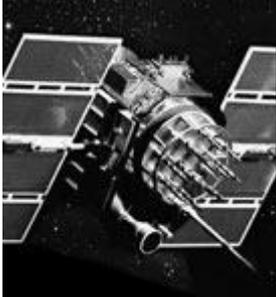
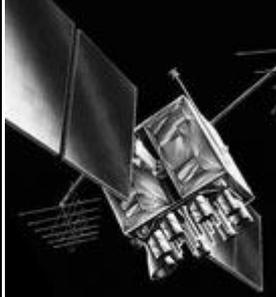
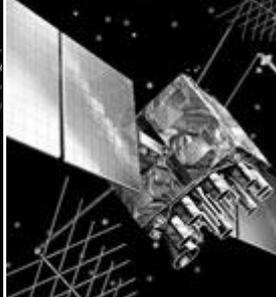
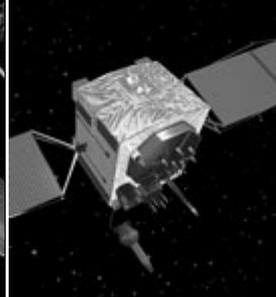
This chapter is structure as follows. Firstly, the background and the current status of GPS will be introduced. Secondly, the mechanism of how GA work will be described. Next, the details of sliding windows will be given. Lastly, some literature reviews regarding to GPS, GA and Sliding Windows will be included.

#### 2.1 Global Positioning System (GPS)

##### 2.1.1 Current Status of GPS

The GPS constellation is a combination of legacy and modernized satellites (Official U.S. Government information about the Global Positioning System (GPS) and related topics [Online], 24 November 2015. Source [www.gps.gov/systems/gps/space](http://www.gps.gov/systems/gps/space)). Recently, there are 31 of block IIA, IIR, IIR-M and IIF satellites in the constellation (U.S. Coast Guard Navigation Center [Online], 24 November 2015. Source [www.navcen.uscg.gov](http://www.navcen.uscg.gov)). The features of each block of GPS satellites are summarized in table 2.1

**Table 2.1** The summarize features of each GPS satellite in constellation  
(Adapted from [www.gps.gov/systems/gps/space/](http://www.gps.gov/systems/gps/space/))

Legacy		Modernized	
			
Block IIA	Block IIR	Block IIR-M	Block IIF
1 Operational	12 Operational	7 Operational	11 Operational
<ul style="list-style-type: none"> <li>▪ C/A code on L1 for civilian user</li> <li>▪ P(Y) code on L1 and L2 for military user</li> <li>▪ 7.5 years design lifespan</li> <li>▪ Launched in 1990-1997</li> </ul>	<ul style="list-style-type: none"> <li>▪ C/A code on L1</li> <li>▪ P(Y) code on L1 and L2</li> <li>▪ On-board clock monitoring</li> <li>▪ 7.5 years design lifespan</li> <li>▪ Launched in 1997-2004</li> </ul>	<ul style="list-style-type: none"> <li>▪ All legacy signals</li> <li>▪ 2<sup>nd</sup> civil signal on L2 (L2C)</li> <li>▪ New military M code signals for enhanced jam resistance</li> <li>▪ Flexible power level for military signals</li> <li>▪ 7.5 years design lifespan</li> <li>▪ Launched in 2005-2009</li> </ul>	<ul style="list-style-type: none"> <li>▪ All Block IIR-M signals</li> <li>▪ 3<sup>rd</sup> civil signal on L5 frequency (L5)</li> <li>▪ Advance atomic clocks</li> <li>▪ Improved accuracy, signal strength, and quality</li> <li>▪ 12 years design lifespan</li> <li>▪ Launched since 2010</li> </ul>

### 2.1.2 Data Observables

To follow the text and equation in this subsection we should explain the notation used as follow. The superscript with lowercase letters p and q generally represent the satellites. Where the subscript with lowercase letters a and b represent the station but the letters i and j represent the frequency  $f$ . Similarly, the carrier frequency L1 and L2 are indicated as 1 and 2 in subscript. Also  $P$ ,  $\varphi$  and  $\Phi$  are represent the pseudorange and carrier phase respectively. Thus,  $M_{1,p}$  is identifies the multipath of L1 code pseudorange measurement.

The basic data observable from GPS used for positioning and timing are pseudorange and carrier phase. Normally, carrier phase is used in multi-frequency receiver for high accuracy surveying at centimeter level (Leick, Rapoport et al. 2015).

Therefore, the fundamental of GPS location determination for pseudorange is based on signal propagation time ( $\Delta T$ ) from phase center of the satellite (emission time) to the phase center of receiver antenna (reception time). This travel time multiply by the speed of light give the apparent range or pseudorange  $P = c\Delta T$  between them. Normally, apparent range or pseudorange is used to distinguish from true range since the true range is based on the geometric range where pseudorange is observable range which may affected from many interference causes them to differ (Hoffmann-Wellenhof, Lichtenegger et al. 1994, Leick 2004).

Let's  $t$  denoted for the GPST or GPS time which the satellite operators will keeps it to remain within one microsecond or better of UTC (Coordinated Universal Time) time. The nominal time is the true times plus a small bias, Thus the time at the receiver can be written as

$$\underline{t}(t) = t + d\underline{t} \quad (\text{Eq. 2.1})$$

Where

$\underline{t}(t)$  is nominal time at receiver

$t$  is true time or GPS time

$d\underline{t}$  is the bias at receiver clock

Also the time at the satellite can be written as

$$\bar{t}(t) = t + d\bar{t} \quad (\text{Eq. 2.2})$$

Where

$\bar{t}(t)$  is nominal time at satellite  
 $d\bar{t}$  is the bias at satellite clock

Let's  $\tau$  be the propagation time of the signal. Thus, if the signal was transmitted from the satellite at satellite nominal time  $\bar{t}(t - \tau)$  and recorded at the receiver at nominal time  $\underline{t}$ . Then the pseudorange can be written as

$$P(t) = c[\underline{t}(t) - \bar{t}(t - \tau)] \quad (\text{Eq. 2.3})$$

Where  $c$  denotes for the speed of light. Replacing nominal time to system time with its bias give

$$P(t) = c \left[ \underline{t} + d\underline{t} - \left( t - \tau + d\bar{t}(t - \tau) \right) \right] = c\underline{t} + cd\underline{t} - cd\bar{t}(t - \tau) \quad (\text{Eq. 2.4})$$

However, as atomic based satellite clock is very stable and can be modeled by a polynomial function where its coefficient being transmitted in the navigation message (Hoffmann-Wellenhof, Lichtenegger et al. 1994). Thus  $cd\bar{t}(t - \tau)$  was replaced by  $cd\bar{t}$  in equation 2.5. Again,  $cd\bar{t}$  is the bias of the satellite clock that left after corrected. Also, the distance in vacuum  $c\tau$  was denoted by  $\rho(t, t - \tau)$  which is the topocentric satellite distance (Leick, Rapoport et al. 2015).

$$P(t) = \rho(t, t - \tau) + cd\underline{t} - cd\bar{t} \quad (\text{Eq. 2.5})$$

Since the signal is travel from the satellite antenna to the receiver antenna where the signal travels through earth atmosphere and also there are others biases that effect to the pseudorange measurement thus more complete equation for pseudorange comparing to equation 2.5 is shows in equation 2.6.

$$\begin{aligned}
P_1(t) = & \rho(t, t - \tau) + cd\bar{t} - cd\bar{t} & \text{(Eq. 2.6)} \\
& -c(d\bar{t}_{sv} - T_{GD} + ISC_{1,P}) + I_{1,P} + T - (h_{1,P} - H_{1,P}) \\
& -(a_{1,P} + A_{1,P}) + M_{1,P} + \varepsilon_{1,P}
\end{aligned}$$

Where

$d\bar{t}_{sv}$  is the satellite clock correction

$T_{GD}$  is time group delays

$ISC_{1,P}$  is intersignal correction

$I_{1,P}$  is ionospheric delays, this bias is caused by dispersive medium to GPS frequency of the ionosphere which depends on the carrier frequency. That why there is the subscript number identifies for the carrier frequency.

$T$  is tropospheric delays, this bias is caused by non-dispersive medium to GPS frequency of the troposphere. Thus this bias does not depend on the carrier frequency. That's why no subscript to identifies the carrier frequency.

$h_{1,P}$  is receiver hardware code delays

$H_{1,P}$  is satellite hardware code delays that is not accounted by  $T_{GD}$  and  $ISC_{1,P}$

$a_{1,P}$  is receiver antenna code center offsets

$A_{1,P}$  is satellite antenna code center offsets

$M_{1,P}$  is multipath. This bias is depends on the direction to the satellite. (Leick 2004)

$\varepsilon_{1,P}$  is random measurement noise. (Hoffmann-Wellenhof, Lichtenegger et al. 1994) stated that traditionally, the precision of pseudorange measurement is about 1% of the chip length. As a result about 0.3m and 3m for P(Y) and C/A code pseudorange respectively. However, it's depending on the technology used, the recent developments shows that the precision of about 0.1% of chip length is achievable.

As mentioned some of biases are transmitted for corrected in the navigation message. These corrections included  $d\bar{t}_{sv}$ ,  $T_{GD}$  and  $ISC_{1,P}$ . Moreover, it assume that the receiver and satellite antenna code center offset are known from the antenna calibration (Bilich, Schmitz et al. 2012)

Thus, the equation in 2.6 can be simplified to more commonly given

$$P_1(t) = \rho(t, t - \tau) + cd\bar{t} - cd\bar{t} + I_{1,P} + T + \delta_{1,P} + \varepsilon_{1,P} \quad (\text{Eq. 2.7})$$

Where

$\delta_{1,P}$  is lumped parameter that combined receiver hardware delay, satellite hardware delay and the multipath. It can be computed from

$$\delta_{1,P} = -h_{1,P} + H_{1,P} + M_{1,P} \quad (\text{Eq. 2.8})$$

Similarly, the pseudorange for second frequency can be written as

$$P_2(t) = \rho(t, t - \tau) + cd\bar{t} - cd\bar{t} + I_{2,P} + T + \delta_{2,P} + \varepsilon_{2,P} \quad (\text{Eq. 2.9})$$

$$\delta_{2,P} = -h_{2,P} + H_{2,P} + M_{2,P} \quad (\text{Eq. 2.10})$$

Please note that the difference between the pseudorange between difference frequency are that the  $T_{GD}$  in equation 2.6 was scaled by the ratio  $\gamma_{12}T_{GD}$  as shows in equation 2.11 before corrected in equation 2.9 for  $P_2(t)$ .

$$\gamma_{ij} = \frac{f_i^2}{f_j^2} \quad (\text{Eq. 2.11})$$

Moreover the instead of  $ISC_{1,P}$  applied in equation 2.6, the correction for  $P_2(t)$  in equation 2.9 the  $ISC_{2,P}$  will be used instead. Similarly, the ionosphere delay and lumped parameter which are depends on the carrier frequency thus  $I_{2,P}$  and  $\delta_{2,P}$  were applied in equation 2.9. Except for the troposphere delay which non-dispersive bias thus it remain unchanged.

Beside from code pseudorange, there is a phase pseudorange computed from phase observable. Conceptually, the phase pseudorange is the sum of the fractional carrier phase at frequency  $f_1$  which arrives at the receiver antenna at nominal time  $\underline{t}$  where this signal was transmitted from satellite  $\tau$  seconds earlier and the unknown integer that represent the number of full waves. Thus the equation in unit of radians for the carrier phase L1 can be express as (Leick 2004, Leick, Rapoport et al. 2015).

$$\begin{aligned} \varphi_1(t) = & \frac{\rho(t, t - \tau)}{\lambda_1} + f_1 d\underline{t} - f_1 (d\bar{t} + d\bar{t}_{sv}) + N_1 \\ & + I_{1,\varphi} + \frac{T}{\lambda_1} - (h_{1,\varphi} - H_{1,\varphi}) - (w_{1,\varphi} - W_{1,\varphi}) - (a_{1,\varphi} - A_{1,\varphi}) + M_{1,\varphi} + \varepsilon_{1,\varphi} \end{aligned} \quad (\text{Eq. 2.12})$$

Where

$\lambda_1$  is the wave length of first frequency

$N_1$  is integer ambiguity, or the number of cycles between the satellite and receiver. This number refers to the first epoch of observations and remains constant during period that tracking is continue without lost of lock. Normally, the lost of lock is cause by cycle slip which introduce an integer jump. Thus the observed carrier phase continues with different integer ambiguity constant

$I_{1,\varphi}$  is the ionospheric delay which advance carrier phase.

$h_{1,\varphi}, H_{1,\varphi}$  are receiver and satellite hardware phase delays

$w_{1,\varphi}, W_{1,\varphi}$  are receiver and satellite phase windup angles. This bias tends to cancel in double difference. It might be allowed to be absorbed by hardware delay

$a_{1,\varphi}, A_{1,\varphi}$  are receiver antenna and satellite antenna phase center offset which can be corrected using the antenna calibration

$M_{1,\varphi}$  is multipath bias.

$\varepsilon_{1,\varphi}$  is carrier phase measurement noise (<0.01 cycles)

To simplify the equation in 2.12, known and corrected biases were hidden. So the equation 2.12 can be written in more standard form of

(Eq. 2.13)

$$\varphi_1(t) = \lambda_1^{-1}\rho(t, t - \tau) + f_1 dt - f_1 d\bar{t} + N_1 - \lambda_1^{-1}I_{1,P} + \lambda_1^{-1}T + \delta_{1,\varphi} + \varepsilon_{1,\varphi}$$

$$\delta_{1,\varphi} = -(h_{1,\varphi} + w_{1,\varphi}) + (H_{1,\varphi} + W_{1,\varphi}) + M_{1,\varphi} \quad (\text{Eq. 2.14})$$

Note that the ionospheric bias on the carrier phase measurement has been changed to parameterize by the pseudorange ionospheric delay. When ionospheric bias on the carrier phase measurement was scaled to the units of meter then the magnitude of the ionospheric bias for both pseudorange and carrier phase measurement are the same except the opposite signs. Therefore the carrier phase advances as the signal propagate to the ionosphere and carrier code delays. In addition the equation in 2.13 and 2.14 can be changed the subscript from 1 to 2 for the second frequency.

The unit of radians of measurable carrier phase in equation 2.13 and 2.14 can be scaled to the unit of length by multiplying with  $\lambda_1 = c/f_1$ . Thus it's becomes

$$\Phi_1(t) = \rho(t, t - \tau) + \lambda_1 N_1 + cdt - cd\bar{t} - I_{1,P} + T + \delta_{1,\Phi} + \varepsilon_{1,\Phi} \quad (\text{Eq. 2.15})$$

$$\delta_{1,\Phi} = -(h_{1,\Phi} + w_{1,\Phi}) + (H_{1,\Phi} + W_{1,\Phi}) + M_{1,\Phi} \quad (\text{Eq. 2.16})$$

Where

The subscript  $\Phi$  implies that the respective quantities are in the unit of length e.g.  $\delta_{1,\Phi} = \lambda_1 \delta_{1,\varphi}$ .

Finally, if both of station subscript and satellite superscript were taken into consideration. Then previous mentioned equation can be written as

$$P_{a,1}^p(t) = \rho_a^p + cdt_a - cd\bar{t}^p + I_{a,1,P}^p + T_a^p + \delta_{a,1,P}^p + \varepsilon_{a,1,P}^p \quad (\text{Eq. 2.17})$$

$$\delta_{a,1,P}^p = -h_{a,1,P}^p + H_{1,P}^p + M_{a,1,P}^p \quad (\text{Eq. 2.18})$$

(Eq. 2.19)

$$\varphi_{a,1}^p(t) = \lambda_1^{-1}\rho_a^p + N_{a,1}^p + f_1 dt_a - f_1 dt^p - \lambda_1^{-1}I_{a,1,P}^p + \lambda_1^{-1}T_a^p + \delta_{a,1,\varphi}^p + \varepsilon_{a,1,\varphi}^p$$

$$\delta_{a,1,\varphi}^p = -(h_{a,1,\varphi} + w_{a,1,\varphi}) + (H_{1,\varphi}^p + W_{1,\varphi}^p) + M_{a,1,\varphi}^p \quad (\text{Eq. 2.20})$$

$$\Phi_{a,1}^p(t) = \rho_a^p + \lambda_1 N_{a,1}^p + cdt_a - cdt^p - I_{a,1,p}^p + T_a^p + \delta_{a,1,\Phi}^p + \varepsilon_{a,1,\Phi}^p \quad (\text{Eq. 2.21})$$

$$\delta_{a,1,\Phi}^p = -(h_{a,1,\Phi} + w_{a,1,\Phi}) + (H_{1,\Phi}^p + W_{1,\Phi}^p) + M_{a,1,\Phi}^p \quad (\text{Eq. 2.22})$$

### 2.1.3 Linear Combination

Currently, linear combination is widely used among the user because of three main motivations. First, the linear combination can eliminate the biases or unwanted terms in the mathematical observation models that are common between the observations. Second, the linear combination is used to alleviate the computation burden of processing GNSS data, e.g. the widelane combination helps to decorrelate the ambiguities and reduce the integer ambiguity search space. Third, the linear combination is used to reduce the bandwidth needed for transmission of the GNSS observation, especially in real-time positioning. (Richert 2005)

#### 2.1.3.1 Linear combination for dual frequency

The general linear combination of two carrier phases  $\varphi_1$  and  $\varphi_2$  is defined by

$$\varphi_{i,j} = i\varphi_1 + j\varphi_2 \quad (\text{Eq. 2.23})$$

$$= \lambda_{i,j}^{-1}\rho + N_{i,j} - f_{i,j}d\underline{t} + f_{i,j}d\bar{t} + I_{i,j,\varphi} + \lambda_{i,j}^{-1}T + \delta_{i,j,\varphi} + \varepsilon_{i,j,\varphi} \quad (\text{Eq. 2.24})$$

Where  $i$  and  $j$  in equation 2.23 are arbitrary numbers. According to the equation 2.24, the wavelength, ambiguity, frequency, ionospheric term for the carrier phase function  $\varphi_{i,j}$  can be defined as

$$\lambda_{i,j} = \frac{c}{f_{i,j}} \quad (\text{Eq. 2.25})$$

$$f_{i,j} = if_1 + jf_2 \quad (\text{Eq. 2.26})$$

$$N_{i,j} = iN_1 + jN_2 \quad (\text{Eq. 2.27})$$

$$I_{i,j,\varphi} = iI_{1,\varphi} + jI_{2,\varphi} = \frac{if_2 + jf_1}{f_2} I_{1,\varphi} \quad (\text{Eq. 2.28})$$

$$I_{i,j,\Phi} = \lambda_{i,j} I_{i,j,\varphi} \quad (\text{Eq. 2.29})$$

The  $\delta_{i,j,\varphi}$  and  $\varepsilon_{i,j,\varphi}$  in equation 2.24 are the hardware delay, multipath and measurement noise respectively. In addition the noise of the linear combination is amplified by the factor  $\sqrt{i^2 + j^2}$  comparing to the noise of undifference or single phase.

- **Linear combination with integer numbers**

The advantage of linear combination with integer number is that the integer nature of ambiguity is preserved. The simplest linear combination of L1 and L2 carrier phase  $\varphi_{L1}$  and  $\varphi_{L2}$  that can be formed regarding to the equation 2.23 are  $i = j = 1$ , yielding the sum

$$\varphi_{L1+L2} = \varphi_{L1} + \varphi_{L2} \quad (\text{Eq. 2.30})$$

And  $i = 1, j = -1$

$$\varphi_{L1-L2} = \varphi_{L1} - \varphi_{L2} \quad (\text{Eq. 2.31})$$

The wave length regarding to the linear combination in equation 2.30 and 2.31 are

$$\begin{aligned} \lambda_{L1+L2} &= 10.7 \text{ cm} \\ \lambda_{L1-L2} &= 86.2 \text{ cm} \end{aligned} \quad (\text{Eq. 2.32})$$

The distance of corresponding to one wavelength is frequently called a lane (Leick 2004). Thus the combination of  $\varphi_{L1+L2}$  is denoted as narrow lane and  $\varphi_{L1-L2}$  is denoted as wide lane.

It also worth to note that, the advantage of using wide lane instead of L1 is that it reduces the number of potential ambiguity combination and greatly reduces the search time. However, disadvantage is that the measurement is significantly noisier than L1. Thus, there are many ambiguity resolution algorithms including LAMBDA (Teunissen, De Jonge et al. 1995) using wide lane to limit the number of ambiguity to be considered (Hoffmann-Wellenhof, Lichtenegger et al. 1994).

- **Linear combination with real numbers**

The disadvantage of linear combination with real numbers is the integer nature of ambiguity is lost. Using real numbers in linear combination is just slightly more complicated. Consider

$$i = 1, \quad j = -\frac{f_{L2}}{f_{L1}} \quad (\text{Eq. 2.33})$$

Which give the linear combination

$$\varphi_{L1,L2} = \varphi_{L1} - \frac{f_{L2}}{f_{L1}} \varphi_{L2} \quad (\text{Eq. 2.34})$$

The linear combination in equation 2.34 is frequently denoted as geometric residual which usually used to reduce the ionospheric effects.

Another useful linear combination is

$$i = 1, \quad j = -\frac{f_{L1}}{f_{L2}} \quad (\text{Eq. 2.35})$$

The linear combination in equation 2.35 is frequently denoted as ionospheric residual which used for cycle slip detection.

Please note that there is also the linear combination for triple frequency, due to the new carrier frequency of L5 for GPS satellite block IIF and newer, which can be finds detail from (Leick, Rapoport et al. 2015).

## 2.1.4 Data Differencing

### 2.1.4.1 Single Difference (SD)

There are three type of single difference among the observation if satellites were observed by two receivers simultaneously, which are across receivers difference, across satellites difference and across time difference. The details for each of them are described below. (Leick, Rapoport et al. 2015)

- **Across Receiver**

It can be formed the general notation to represent the across receivers differences as shows in equation 2.36.

$$(\blacksquare)_{ab}^p = (\blacksquare)_a^p - (\blacksquare)_b^p \quad (\text{Eq. 2.36})$$

$$P_{ab,1}^p = \rho_{ab}^p + cdt_{ab} + I_{ab,1,P}^p + T_{ab}^p - h_{ab,1,P} + M_{ab,1,P}^p + \varepsilon_{ab,1,P}^p \quad (\text{Eq. 2.37})$$

$$\varphi_{ab,1}^p = \lambda_1^{-1} \rho_{ab}^p + N_{ab,1}^p + f_1 dt_{ab} - \lambda_1^{-1} I_{ab,1,P}^p + \lambda_1^{-1} T_{ab}^p - h_{ab,1,\varphi} + M_{ab,1,\varphi}^p \quad (\text{Eq. 2.38})$$

$$+ \varepsilon_{ab,1,\varphi}^p \quad (\text{Eq. 2.39})$$

$$\Phi_{ab,1}^p = \rho_{ab}^p + \lambda_1 N_{ab,1}^p + cdt_{ab} - I_{ab,1,P}^p + T_{ab}^p - h_{ab,1,\Phi} + M_{ab,1,\Phi}^p + \varepsilon_{ab,1,\Phi}^p$$

$$\sigma_{(\blacksquare)_{ab}^p} = \sqrt{2} \sigma_{(\blacksquare)} \quad (\text{Eq. 2.40})$$

The advantage of single difference across receiver is the satellite clock bias  $dt^p$  and the satellite hardware delay  $H_{1,\varphi}^p$  are cancel. Please note that the windup angles are omitted for simplicity in this expression and also in subsequence expressions. It is also worth to note that in relative positioning over short distances, the ionospheric bias and

tropospheric bias are almost the same at each station due to high spatial correlation which are largely cancel in the differencing. (Leick, Rapoport et al. 2015)

- **Across Satellite**

$$(\blacksquare)_a^{pq} = (\blacksquare)_a^p - (\blacksquare)_a^q \quad (\text{Eq. 2.41})$$

$$P_{a,1}^{pq} = \rho_a^{pq} + cdt^{pq} + I_{a,1,P}^{pq} + T_a^{pq} + H_{1,P}^{pq} + M_{a,1,P}^{pq} + \varepsilon_{a,1,P}^{pq} \quad (\text{Eq. 2.42})$$

$$(\text{Eq. 2.43})$$

$$\varphi_{a,1}^{pq} = \lambda_1^{-1} \rho_a^{pq} + N_{a,1}^{pq} + f_1 dt^{pq} - \lambda_1^{-1} I_{a,1,P}^{pq} + \lambda_1^{-1} T_a^{pq} + H_{1,\varphi}^{pq} + M_{a,1,\varphi}^{pq} + \varepsilon_{a,1,\varphi}^{pq}$$

$$\Phi_{a,1}^{pq} = \rho_a^{pq} + \lambda_1 N_{a,1}^{pq} + cdt^{pq} - I_{a,1,P}^{pq} + T_a^{pq} + H_{1,\Phi}^{pq} + M_{a,1,\Phi}^{pq} + \varepsilon_{a,1,\Phi}^{pq} \quad (\text{Eq. 2.44})$$

The advantages of single difference across satellite are the cancelation of receiver clock error  $dt_a$  and receiver hardware delay (Leick, Rapoport et al. 2015).

- **Across Time**

$$\Delta(\blacksquare)_a^p = (\blacksquare)_a^p(t_2) - (\blacksquare)_a^p(t_1) \quad (\text{Eq. 2.45})$$

$$\Delta P_{a,1}^p = \Delta \rho_a^p + c\Delta dt_a - c\Delta dt^p + \Delta I_{a,1,P}^p + \Delta T_a^p + \Delta M_{a,1,P}^p + \Delta \varepsilon_{a,1,P}^p \quad (\text{Eq. 2.46})$$

$$(\text{Eq. 2.47})$$

$$\Delta \varphi_{a,1}^p = \lambda_1^{-1} \Delta \rho_a^p + f_1 \Delta dt_a - f_1 \Delta dt^p - \lambda_1^{-1} \Delta I_{a,1,P}^p + \lambda_1^{-1} \Delta T_a^p + \Delta M_{a,1,\varphi}^p + \Delta \varepsilon_{a,1,\varphi}^p$$

$$\Delta \Phi_{a,1}^p = \Delta \rho_a^p + c\Delta dt_a - c\Delta dt^p - \Delta I_{a,1,P}^p + \Delta T_a^p + \Delta M_{a,1,\Phi}^p + \Delta \varepsilon_{a,1,\Phi}^p \quad (\text{Eq. 2.48})$$

According to the expression above, the clock errors might not cancel over long time interval. However, it is assumed that the hardware delays are sufficiently stable to cancel. If there is no cycle slip occur between epochs therefore the ambiguity also

cancel. As a result the above expressions reflect the change of topocentric satellite distance over time and referred to as delta ranges (Leick, Rapoport et al. 2015).

#### 2.1.4.2 Double Difference (DD)

The double difference can be formed when two receivers observes two satellites simultaneously. It can be either difference of two across-receiver differences or two across-satellite differences. The double difference can be formed as shows in equation 2.49 – 2.52

$$(\blacksquare)_{ab}^{pq} = (\blacksquare)_{ab}^p - (\blacksquare)_{ab}^q = (\blacksquare)_a^{pq} - (\blacksquare)_b^{pq} \quad (\text{Eq. 2.49})$$

$$P_{ab,1}^{pq} = \rho_{ab}^{pq} + I_{ab,1,P}^{pq} + T_{ab}^{pq} + M_{ab,1,P}^{pq} + \varepsilon_{ab,1,P}^{pq} \quad (\text{Eq. 2.50})$$

$$\varphi_{ab,1}^{pq} = \lambda_1^{-1} \rho_{ab}^{pq} + N_{ab,1}^{pq} - \lambda_1^{-1} I_{ab,1,P}^{pq} + \lambda_1^{-1} T_{ab}^{pq} + M_{ab,1,\varphi}^{pq} + \varepsilon_{ab,1,\varphi}^{pq} \quad (\text{Eq. 2.51})$$

$$\Phi_{ab,1}^{pq} = \rho_{ab}^{pq} + \lambda_1 N_{ab,1}^{pq} - I_{ab,1,P}^{pq} + T_{ab}^{pq} + M_{ab,1,\Phi}^{pq} + \varepsilon_{ab,1,\Phi}^{pq} \quad (\text{Eq. 2.52})$$

$$\sigma_{(\blacksquare)_{ab}^{pq}} = 2\sigma_{(\blacksquare)} \quad (\text{Eq. 2.53})$$

The double difference is very famous and used among users because of its features of the receiver clock error, satellite clock error, and receiver hardware delay and satellite hardware delay cancelation. Moreover, since double difference implies across-receiver difference thus the effect from the ionosphere and troposphere are largely cancel in short distance. The integer ambiguity term in the double difference  $N_{ab,1}^{pq}$  play an important role in relative positioning. The estimating of ambiguity together with other parameters as a real number is called float solution. If the ambiguity can be successfully constrained to integer then it will be called fixed solution (Leick, Rapoport et al. 2015).

### 2.1.4.3 Triple Difference (TD)

The triple difference is the difference between two double differences over time.

$$\Delta(\blacksquare)_{ab}^{pq} = (\blacksquare)_{ab}^{pq}(t_2) - (\blacksquare)_{ab}^{pq}(t_1) \quad (\text{Eq. 2.54})$$

$$\Delta P_{ab,1}^{pq} = \Delta \rho_{ab}^{pq} + \Delta I_{ab,1,P}^{pq} + \Delta T_{ab}^{pq} + \Delta M_{ab,1,P}^{pq} + \Delta \varepsilon_{ab,1,P}^{pq} \quad (\text{Eq. 2.55})$$

$$\quad (\text{Eq. 2.56})$$

$$\Delta \varphi_{ab,1}^{pq} = \lambda_1^{-1} \Delta \rho_{ab}^{pq} - \lambda_1^{-1} \Delta I_{ab,1,P}^{pq} + \lambda_1^{-1} \Delta T_{ab}^{pq} + \Delta M_{ab,1,\varphi}^{pq} + \Delta \varepsilon_{ab,1,\varphi}^{pq}$$

$$\Delta \Phi_{ab,1}^{pq} = \Delta \rho_{ab}^{pq} - \Delta I_{ab,1,P}^{pq} + \Delta T_{ab}^{pq} + \Delta M_{ab,1,\Phi}^{pq} + \Delta \varepsilon_{ab,1,\Phi}^{pq} \quad (\text{Eq. 2.57})$$

$$\sigma(\Delta \blacksquare_{ab}^{pq}) = \sqrt{8} \sigma_{(\blacksquare)} \quad (\text{Eq. 2.58})$$

In triple difference the ambiguity cancels. As a result, the triple difference is probably the easiest observable to process. Often, the triple difference solution serves as an initial position for subsequent double-difference solution. In addition, the triple difference also used to map individual cycle slips, which can be detected and removed later on. (Leick, Rapoport et al. 2015).

### 2.1.5 Relative Positioning

Relative positioning or differential positioning required at least two GPS receivers, one as base station setup at known point and the others one as rover station setup at location to be determined its coordinate. The rover coordinate will be computed with respect to the base station's coordinate. The data from both stations will be collected simultaneously. After applied the data differencing with the data from both stations the major biases (e.g. tropospheric delay, ionospheric delay, receiver clock bias and satellite clock bias) can be largely reduced. Based on this reason the relative positioning is widely used for applications required high accuracy (cm level). However, the effectiveness of relative positioning is depend on the distance between base station and rover station. The longer of base line be likely to produce larger residual errors

because of uncorrelated biases between both stations. Based on carrier phase-based differential GPS, it can be further categorized as static positioning and kinematic positioning (Satirapod 2002).

#### **2.1.5.1 Static Positioning**

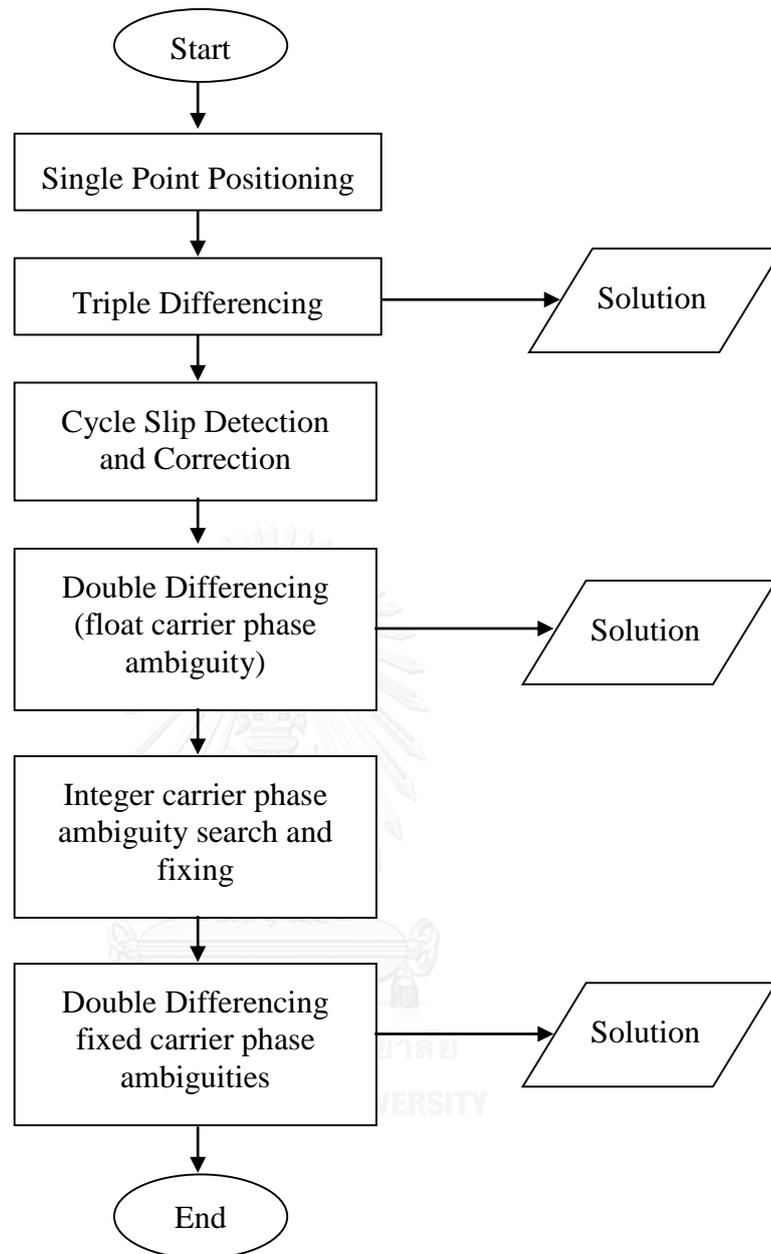
Static positioning is required that the receivers at both base station and rover are stationary. Both base station and rover must collect the data simultaneously or very near simultaneously. The duration to collect the data is depends on many factors e.g. the number of satellites, satellite geometry, distance between two receiver, receiver type and accuracy requirement. But normally, both receivers must collect the data from same four satellites (at least) simultaneously with the duration in between 1-2 hours. In addition, to obtained good result from the static positioning the baseline length should not exceed 20km (Satirapod 2002).

#### **2.1.5.2 Kinematic Positioning**

Kinematic positioning implied that either (or both) the receiver at base station or rover station is (are) in motion. But in most cases the receiver at base station is stationary while the receiver at the rover will be move. Previously, to obtain the position in kinematic positioning is difficult due to the difficulties that the ambiguity must be resolved before the survey and the ambiguity must be reinitialized during the survey when the cycle slip occurs (Rizos 1997). However, with the recent receiver technology including the advancement of ambiguity resolution algorithm the user can fixed the ambiguity “On-The-Fly” or in real-time. (Satirapod 2002)

#### **2.1.5.3 Processing of differencing data**

The process to obtain precise solution in relative positioning is related to the processing of differential data shows in figure 2.1. According to the figure 2.1 the processing strategies are applied in sequence starting from the triple difference through the double difference aiming to obtained ambiguity-fixed solution.



**Figure 2.1** The processes of GPS phase data processing (adapted from (Rizos 1997))

### 2.1.6 The Processing of Relative Positioning in RTKLib

It is worth to include the details about the mathematic description of positioning mode used in RTKLib (Takasu 2013). In RTKLib the EKF (Extended Kalman Filter) is used in several positioning solutions e.g. DGPS, DGNSS, Static, Kinematic and Moving baseline. The following details will be focusing on the strategies used to obtain the solution for static and kinematic positioning mode only. Further details of another positioning mode can be obtained from Appendix E of RTKLib user manual (Takasu 2013). Actually the EKF is related to the prediction (time update) and correction (measurement update) of the state vector including its covariance.

**Firstly**, EKF formulation, the state vector of unknown model parameters  $x$  and its covariance matrix  $P$  which can be estimated from the measurement vector  $y$  at epoch  $t_k$  by

$$\hat{x}_k(+) = \hat{x}_k(-) + K_k(y_k - h(\hat{x}_k(-))) \quad (\text{Eq. 2.59})$$

$$P_k(+) = (I - K_k H(\hat{x}_k(-))) P_k(-) \quad (\text{Eq. 2.60})$$

$$K_k = P_k(-) H(\hat{x}_k(-)) (H(\hat{x}_k(-)) P_k(-) H(\hat{x}_k(-))^T + R_k)^{-1} \quad (\text{Eq. 2.61})$$

Where

$\hat{x}_k$  is the estimated state vector at the epoch  $t_k$

$P_k$  is the covariance matrix of  $\hat{x}_k$  at the epoch  $t_k$

$h(x)$  is measurement model vector

$H(x)$  is matrix of partial derivatives

$R_k$  is the covariance matrix of measurement error

Please noted that the (-) and (+) are indicated before and after measurement update (correction) of EKF.

Assuming the system-model is linear, then the time update (prediction) of the state vector and its covariance matrix for EKF can be written as

$$\hat{x}_{k+1}(-) = F_k^{k+1} \hat{x}_k(+) \quad (\text{Eq. 2.62})$$

$$P_{k+1}(-) = F_k^{k+1} P_k(+)(F_k^{k+1})^T + Q_k^{k+1} \quad (\text{Eq. 2.63})$$

Where

$F_k^{k+1}$  is transition matrix of system noise from epoch  $t_k$  to  $t_{k+1}$

$Q_k^{k+1}$  is covariance matrix of the system noise from epoch  $t_k$  to  $t_{k+1}$

**Secondly**, the double difference measurement model for short baseline are formed. According to the equation 2.52 and 2.50, For short baseline (<10 km) (Takasu 2013) where the spatial correlation between two point are highly correlated. Thus the ionospheric delay and tropospheric delay are largely reduced. These equation can be re-written as

$$\Phi_{ab,i}^{pq} = \rho_{ab}^{pq} + \lambda_i(N_{ab,i}^p - B_{ab,i}^q) + \delta_{ab,i,\Phi}^{pq} + \varepsilon_{ab,1,\Phi}^{pq} \quad (\text{Eq. 2.64})$$

$$P_{ab,i}^{pq} = \rho_{ab}^{pq} + \varepsilon_{ab,1,P}^{pq} \quad (\text{Eq. 2.65})$$

If based station is setup at location  $a$  with known coordinate,  $r_a$ . Thus the geometric range of  $\rho_b^p$  and  $\rho_b^q$  can be computed. Please note that the lump parameter term in equation 2.64 also can be neglected in short-baseline.

For the single difference across receiver, it's recommended in RTKLib that it should be made with the data at same epoch. The epoch time difference between receivers is sometimes referred to "Age of Differential". As the difference of time is increased, the accuracy of the solution is decreased.

For the single difference across satellite, RTKLib will select the reference satellite with maximum elevation angle on the epoch-by-epoch basis. Please also noted that there is no across satellite system difference (e.g. GPS vs. Glonass or GPS vs. Galileo) because the receiver usually has different group delay for the signal from difference satellite system.

It's assuming that the triple-frequency GPS receivers are used for both reference station at point  $a$  and rover station at point  $b$ . Therefore the unknown parameters state vector  $x$  which they are going to be estimated can be defined as

$$x = (r_b^T, v_r^T, N_1^T, N_2^T, N_5^T)^T \quad (\text{Eq. 2.66})$$

Where

$N_i$  is single difference carrier-phase biases

$$N_i = (N_{ab,i}^1, N_{ab,i}^2, \dots, N_{ab,i}^m)^T \quad (\text{Eq. 2.67})$$

The measurement vector  $y$  is also defined with double difference of phase range and pseudorange measurement as followed

$$y = (\Phi_{1,}^T, \Phi_{2,}^T, \Phi_{5,}^T, P_{1,}^T, P_{2,}^T, P_{5,}^T)^T \quad (\text{Eq. 2.68})$$

Where

$$\Phi_i = (\Phi_{ab,i}^{12}, \Phi_{ab,i}^{13}, \Phi_{ab,i}^{14}, \dots, \Phi_{ab,i}^{1m})^T \quad (\text{Eq. 2.69})$$

$$P_i = (P_{ab,i}^{12}, P_{ab,i}^{13}, P_{ab,i}^{14}, \dots, P_{ab,i}^{1m})^T \quad (\text{Eq. 2.70})$$

**Thirdly**, The measurement update (correction) of EKF for short baseline by using equation in 2.64 and 2.65, the measurement model vector  $h(x)$ , the matrix of partial derivatives  $H(x)$  and the covariance matrix of measurement errors  $R$  can be written as

$$h(x) = (h_{\Phi,1}^T, h_{\Phi,2}^T, h_{\Phi,5}^T, h_{P,1}^T, h_{P,2}^T, h_{P,5}^T)^T \quad (\text{Eq. 2.71})$$

$$H(x) = \left. \frac{\partial h(x)}{\partial x} \right|_{x=\hat{x}} = \begin{bmatrix} -DE & 0 & \lambda_1 D & 0 & 0 \\ -DE & 0 & 0 & \lambda_2 D & 0 \\ -DE & 0 & 0 & 0 & \lambda_5 D \\ -DE & 0 & 0 & 0 & 0 \\ -DE & 0 & 0 & 0 & 0 \\ -DE & 0 & 0 & 0 & 0 \end{bmatrix} \quad (\text{Eq. 2.72})$$

$$R = \begin{bmatrix} DR_{\Phi,1}D^T & & & & & & \\ & DR_{\Phi,2}D^T & & & & & \\ & & DR_{\Phi,5}D^T & & & & \\ & & & DR_{P,1}D^T & & & \\ & & & & DR_{P,2}D^T & & \\ & & & & & DR_{P,5}D^T & \\ & & & & & & DR_{P,5}D^T \end{bmatrix} \quad (\text{Eq. 2.73})$$

$$h_{\Phi,i} = \begin{bmatrix} \rho_{ab}^{12} + \lambda_i(N_{ab}^1 - N_{ab}^2) \\ \rho_{ab}^{13} + \lambda_i(N_{ab}^1 - N_{ab}^3) \\ \vdots \\ \rho_{ab}^{1m} + \lambda_i(N_{ab}^1 - N_{ab}^m) \end{bmatrix}, h_{P,i} = \begin{bmatrix} \rho_{ab}^{12} \\ \rho_{ab}^{13} \\ \vdots \\ \rho_{ab}^{1m} \end{bmatrix} \quad (\text{Eq. 2.74})$$

$$D = \begin{bmatrix} 1 & -1 & 0 & \dots & 0 \\ 1 & 0 & -1 & \dots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & 0 & 0 & \dots & -1 \end{bmatrix} \quad (\text{Eq. 2.75})$$

$$E = (e_b^1, e_b^2, \dots, e_b^m)^T \quad (\text{Eq. 2.76})$$

$$R_{\Phi,i} = \text{diag}(2\sigma_{\Phi,i}^{12}, 2\sigma_{\Phi,i}^{22}, \dots, 2\sigma_{\Phi,i}^{m2}) \quad (\text{Eq. 2.77})$$

$$R_{P,i} = \text{diag}(2\sigma_{P,i}^{12}, 2\sigma_{P,i}^{22}, \dots, 2\sigma_{P,i}^{m2}) \quad (\text{Eq. 2.78})$$

Where

$D$  is single differencing matrix

$\sigma_{\Phi,i}^S$  is standard deviation of  $L_i$  phase-range measurement error (m)

$\sigma_{P,i}^S$  is standard deviation of  $L_i$  pseudorange measurement error (m)

With these equations, the EKF can be estimate the rover position, velocity and float solution of single difference carrier-phase biases at the epoch  $t_k$

**Next**, Time update (prediction) of EKF for the kinematic positioning mode with receiver dynamics (in RTKLib's setting, Positioning Mode = kinematic and REC Dynamics = ON), the time update (prediction) in EKF can be expressed as

(Eq. 2.79)

$$F_k^{k+1} = \begin{bmatrix} I_{3 \times 3} & I_{3 \times 3} \tau_b & \\ & I_{3 \times 3} & \\ & & I_{(3m-3) \times (3m-3)} \end{bmatrix}, Q_K^{k+1} = \begin{bmatrix} 0_{3 \times 3} & & \\ & Q_v & \\ & & 0_{(3m-3) \times (3m-3)} \end{bmatrix}$$

$$Q_v = E_b^T \text{diag}(\sigma_{ve}^2 \tau_b, \sigma_{vn}^2 \tau_b, \sigma_{vu}^2 \tau_b) E_b \quad (\text{Eq. 2.80})$$

$$\tau_b = t_{k+1} - t_k \quad (\text{Eq. 2.81})$$

Where

$\tau_b$  is GPS receiver sampling interval (s)

$\sigma_{ve}, \sigma_{vn}, \sigma_{vu}$  are the standard deviations of east, north and up component of the rover velocity system noise

For the pure kinematic mode without receiver dynamics (in RTKLib's setting, Positioning Mode = Kinematic and REC Dynamics = OFF), thus the equation 2.79 can be replaced by

(Eq. 2.82)

$$F_k^{k+1} = \begin{bmatrix} I_{3 \times 3} & & \\ & I_{3 \times 3} & \\ & & I_{(3m-3) \times (3m-3)} \end{bmatrix}, Q_K^{k+1} = \begin{bmatrix} \infty_{3 \times 3} & & \\ & 0_{3 \times 3} & \\ & & 0_{(3m-3) \times (3m-3)} \end{bmatrix}$$

By adding the large process noise  $\infty_{3 \times 3}$  ( $10^4$ ) to the variance in RTKLib, the receiver position state are reset to initial guess value at every epoch

For static mode (In RTKLib's setting, Positioning Mode = Static) the equations 2.79 is replaced by

(Eq. 2.83)

$$F_k^{k+1} = \begin{bmatrix} I_{3 \times 3} & & \\ & I_{3 \times 3} & \\ & & I_{(3m-3) \times (3m-3)} \end{bmatrix}, Q_K^{k+1} = \begin{bmatrix} 0_{3 \times 3} & & \\ & 0_{3 \times 3} & \\ & & 0_{(3m-3) \times (3m-3)} \end{bmatrix}$$

For the instantaneous ambiguity resolution mode (In RTKLib's setting, Integer Ambiguity Resolution = Instantaneous) the time update (prediction) of the single difference carrier-phase biases  $N_i$  are handle differently. In this mode, the values of the carrier phase bias state are not succeeded to the next epoch by the EKF time update by adding the process noise ( $10^4$ ) to the variance.

**Finally**, Integer ambiguity resolution after the state obtained in the EKF measurement update (correct). Then the float carrier-phase ambiguities can be resolved into the integer values (In RTKLib's setting, Integer Ambiguity Resolution = Continuous, Instantaneous or Fix and Hold).

First of all, the estimated state of single difference will be transformed to the double difference forms by

$$\hat{x}'_k = G\hat{x}_k(+) = (\hat{r}_b^T, \hat{v}_b^T, \hat{N}^T)^T \quad (\text{Eq. 2.84})$$

$$P'_k = GP_k(+)G^T = \begin{bmatrix} Q_R & Q_{NR} \\ Q_{RN} & Q_N \end{bmatrix} \quad (\text{Eq. 2.85})$$

$$G = \begin{bmatrix} I_{6 \times 6} & & & \\ & D & & \\ & & D & \\ & & & D \end{bmatrix} \quad (\text{Eq. 2.86})$$

Where

$G$  is the SD to DD transformation matrix

This SD to DD transformation is to transform the SD carrier-phase biases to be used as the initial value for the integer ambiguity  $\hat{N}$  and there covariance  $Q_N$ . The equation expressed in 2.1.6.3.29 is the formula to solving an ILS (Integer Least Square) which in this case to obtained the appropriate integer vector  $\check{N}$ .

$$\tilde{N} = \underset{N \in \mathbb{Z}}{\operatorname{argmin}}((N - \hat{N})^T Q_N^{-1}(N - \hat{N})) \quad (\text{Eq. 2.87})$$

Recently, a very well-known tools to solve the ILS problem is called LAMBDA (Teunissen, De Jonge et al. 1995). Once the integer ambiguities are obtained, it will be validated by the Ratio-Test which expressed in equation 2.88.

$$R = \frac{(\tilde{N}_2 - \hat{N})^T Q_N^{-1}(\tilde{N}_2 - \hat{N})}{(\tilde{N} - \hat{N})^T Q_N^{-1}(\tilde{N} - \hat{N})} > R_{thres} \quad (\text{Eq. 2.88})$$

Where

$\tilde{N}$  is the best estimation of integer ambiguity

$\tilde{N}_2$  is the second best estimation of integer ambiguity

$R_{thres}$  is the validation Threshold

The Ratio-Test expressed in equation 2.88 is used to check the reliability of the solution (Takasu 2013)

If the validation is success, then the “FIXED” solution of the rover position and velocity can be obtained from  $\check{r}_b$  and  $\check{v}_b$  as expressed in the equation 2.89

$$\begin{pmatrix} \check{r}_b \\ \check{v}_b \end{pmatrix} = \begin{pmatrix} \hat{r}_b \\ \hat{v}_b \end{pmatrix} - Q_{RN} Q_N (\hat{N} - \tilde{N}) \quad (\text{Eq. 2.89})$$

However, if the validation is fail, then the output for “FLOAT” solution are  $\hat{r}_b$  and  $\hat{v}_b$  respectively.

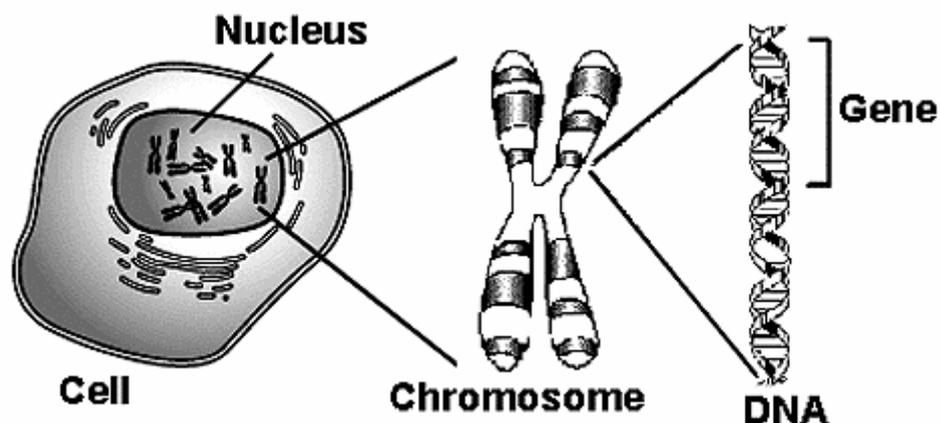
## 2.2 Genetic Algorithm (GA)

### 2.2.1 Background

Genetic Algorithm (GA) is a member of the global optimum search algorithms that mimics the natural evolution known as Darwinian Evolution. The notation of natural evolution is the process by which organism change over time in both behavioral and physical traits to help them better adopt to the environment where they can survive and produced more offspring. It is first introduced by Charl Robert Darwin in his book “On the Origin of Species” in 1859. Thus, The GA is based on an iterative procedure, an “evolutionary” process where the virtual population will be created and each of them will be evaluated for its fitness which is the mechanism to measure how well it can fit to the problem or naturally how well it can adapt to the environment. The population that give high fitness score are allow to get more chance to survive and produce more offspring, based on genetically reproduction, in next generation. As a result, the good traits from the parent will be passing through its offspring, therefore this process of evolutionary will keep continue until it convergent to the global optimum. Moreover, it tries to extract the best solution out of a search space without any apriori knowledge or mathematical model related to the problem. Because of its advantages (simple, supports parallel computation and immune to noise), it has been extensively used in various fields of study and applications (Holland 1992).

Before further details about the GA will be discussed. It is worth to introduce about the terminology used throughout this section and also this thesis. Mitchell (Mitchell 1998) gave very understandable introductory as followed. All living organism compose of cell(s) such as skin cells, muscle cells and nerve cells. These cell are needed in order to keep organism body functioning. Each cell is commanded and controlled by a nucleus which consist of one or more chromosomes (or string of DNA, Deoxyribonucleic acid) that contain genetic information. As depict in figure 2.2, a chromosome contains many genes and each of them have specific locus (or location) on the chromosome where physical traits such as tongue rolling, eyes color and dimples are mostly influenced by the combination between genes and environment. The different setting of trait such as black, blue and brown in eye color is called alleles.

Naturally, during the sexual reproduction, the recombination or crossover occurs. The crossover is the mechanism that genes between parents are exchanged to form new set(s) of chromosome of the offspring. Where mutation, caused by the copying errors, is the mechanism that change of single nucleotides (or elementary bits of DNA) from the parent to the offspring. In addition, the success of an organism usually measured from its success in living to reproduce more offspring.



**Figure 2.2** The chromosome, gene and DNA in the nucleus of cell (from <http://igbiology.blogspot.com/2014/03/chromosomes-dna-genes-and-alleles.html>, last access: 23/11/2015)

In GA, the chromosome is referred to the candidate of possible solution to the problem. It's normally encoding with bit string. Normally, each bit in the bit string is referred to genes which encode the particular parameter of the particular candidate solution. However, it's also possible that genes may be encoded by short block of adjacent bits. In case of a Crossover, it is acts as natural crossover where chromosome from two parents are exchanged. Mutation is consist of randomly changing the alleles (or flipping bit in normal encoding chromosome) at a randomly select locus or location in the chromosome. Finally, the fitness of particular candidate solution is defined by the probability that particular chromosome are survived to reproduce or as the function of number of the offspring.

### 2.2.2 The basic of GA

As mentions earlier the candidate of possible solution referred to the chromosomes which they can also be considered as a virtual organism. Therefore, as show in figure 2.3, the first step of implementation in any GA is to generate an initial population which in most cases it is generated randomly. Normally, the population is made up of binary string where the length of the string ( $l$ ) is according to encoding of the problem.

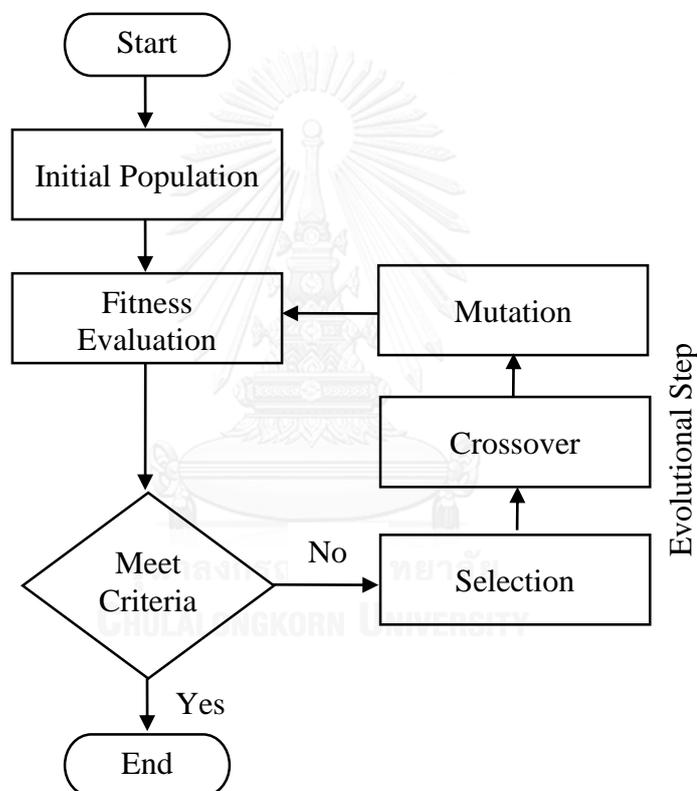
After the initial population was created then for each evolutionary step also known as generation, each string (chromosome) in the population is then evaluated and assigned a fitness score based on its ability of solving a problem which shown as the “fitness evaluation process”. To prevent misunderstanding with fitness evaluation process where it compost of two functions, the evaluation function or objective function and the fitness function respectively. The objective function is used to evaluate or measure the performance of each chromosome regarding to the problem, where the fitness function is used to transform the measured performance into the fitness score which will effect to the opportunity to make a reproduction of an offspring (Whitley 1994).

Subsequently, a criterion will be evaluated. If the criterion is not met, then a new set of population for the next generation will be created. Typically, the criterion can be either a number of generations or closeness to the solution.

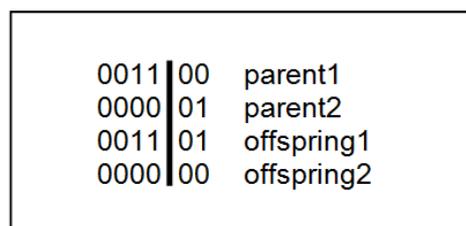
To establish a new set of population, the chromosome will be selected from the entire populations based on a “genetical selection” method. Currently, there are many genetical selection methods being used but the fitness-proportionate selection (or roulette wheel) method is normally used. The fitness-proportionate selection method will select an individual chromosome based on its fitness score. Thus, a high-score fitness chromosome tends to have better chance to reproduce a new offspring, while a low-score fitness chromosome is more likely to disappear.

However, the implementation of the fitness-proportionate selection method alone is not sufficient since the GA may not lead to a converged solution. Therefore, genetically-based operators, known as a *crossover* and *mutation*, are needed to create new point in the search space. If the probability between two selected parents is lower than a crossover rate denoted  $p_c$ , the crossover will be performed by swapping their

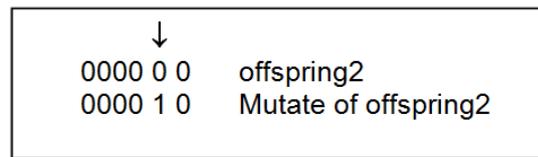
genes at randomly selected point along the length of the chromosome. Figure 2.4 illustrates the crossover process. The mutation is the method to add change to each gene in chromosome of the offspring with very small probability  $p_m$ , (called *mutation rate*), the mutation process is illustrated in Figure 2.5. The change is required since it yields diversity to the population and sometime it produces an outstanding advantage among the group. On the other hand, the mutation process aims to prevent the diversity of the solution to the local optima. Figure 2.3 shows the operating procedure of the standard GA (Mitchell 1998, Buckland and Collins 2002).



**Figure 2.3** Operating procedure of standard Genetic Algorithm



**Figure 2.4** The crossover process performed at the crossover point as shown in black line



**Figure 2.5** The mutation process performed at the black arrow

### 2.2.3 The schema theorem

According to previous section where the way how does GA work was described. One may have a question that how simple genetic operator can find a new point in search space and later on reach to the global optimum point. To answer the question, (Holland 1992) gives an example of string and their fitness shows in table 2.2

**Table 2.2** The example of string and its fitness

String	Fitness
00111	49
10100	400
01000	64
11001	625

According to the string and fitness shown in table 2.2, if the string and fitness was considered one by one, then there are not much useful information can be obtained. But, if one start glancing up and down among string, fitness, similarities and its pattern, then one will start to recognize some patterns of similarities that give high fitness score. Moreover, the longer of glancing among the string, the more information can be obtained from the pattern of similarities and fitness score. Thus important similarities of highly fit string can help guide a search in search space.

From the above idea, Holland (Holland 1992) introduces the concept of schema (plural schemata) or similarity template to describing a subset of strings with similarities at certain string position. Normally, in the binary string there are only two alphabet  $\{0, 1\}$  where the concept of schema will introduce one more alphabet “\*” or don’t care symbol so the schema will compose of ternary alphabet  $\{0, 1, *\}$ . To be

easier to understand, the string that contain \* are referred to as “schema”. Moreover, it can be think of schema as matching device, the string said to match with particular schemata if and only if at every location in the string a 1 match with 1 in schemata, 0 is match with 0 in schemata and either 0 or 1 match with the \* in schemata such as \*100 is matched with {1100, 0100}. From above mentioned one may argued about number of schema in the search space because with the string of length  $l$ , there are  $3^l$  schemata where if the specific string was considered then the search space will contained only  $2^l$ .

To answer the question of “why the complexity should be added into the search space”. Please be reminded that the schema is just a notion used to describe the similarities of the string in the population; it does not do the real job of GA. Actually, the complexity were added into the search space because it has not much information if each of the strings in the population was considered individually as mention earlier thus it is much better and useful if the relation and similarity among them were considered.

Therefore, it is good to know how much of schema in the search space for population in one generation. For any string of length  $l$  such as a string 10001 which is a string of length 5 characters will compose of the schema up to  $2^l$  since each location in the string can be either the present bit value or \*. Thus in population of size  $n$  there will be number of schema in between  $2^l$  to  $n \cdot 2^l$  up to the population diversity. In addition, not all schemata are being used to locate the global optimum only the important of schema that may grow and decay from generation to generation. Let’s consider what happened to the schema when the genetic operator of selection, crossover and mutation do their action. For the selection which select most appropriate string based on its fitness or in other words the more fitness the more string got probability of selection. As a result, the schema contains in highly fit string will got more chance to survive to produce more offspring.

Now, if string of length 5 and its representative of schemata is considered

1	0	1	1	0	String
1	*	*	*	0	Schema #1
*	*	1	1	*	Schema #2

if crossover site was randomly generate at locus 2 then

1	0		1	1	0	String
1	*		*	*	0	Schema #1
*	*		1	1	*	Schema #2

From above example one will see that both schema #1 and #2 match with the string where the crossover site took place at locus 2 which preserve the schema #2 but destroy schema #1. In the case for mutation where it effect to the schema less than the crossover since  $p_m \ll p_c$ . As the number of don't care symbol in schema #1 and #2 is equal but different in location. There are two schema properties to distinguish type of schemata which are schema order and defining length.

The order of schema H, denoted by  $o(H)$ , is number of binary alphabet in the string of schema for example

**Table 2.3** The example of schema and its order

Schema	Symbolic writing	Schema Order
1***0	$o(1 \text{ *** } 0)$	2
**1**	$o(** 1 **)$	1

The defining length of schema H, denoted by  $\delta(H)$ , is the distance between the first and last binary alphabet in the string of schema for example

**Table 2.4** The example of schema and its length

Schema	Symbolic writing	Length of Schema
1***0	$\delta(1 \text{ *** } 0)$	4
**11*	$\delta(\text{** } 11 \text{ *})$	1

Let's go back to schema #1 and #2 when apply mentioned schema properties then it will get

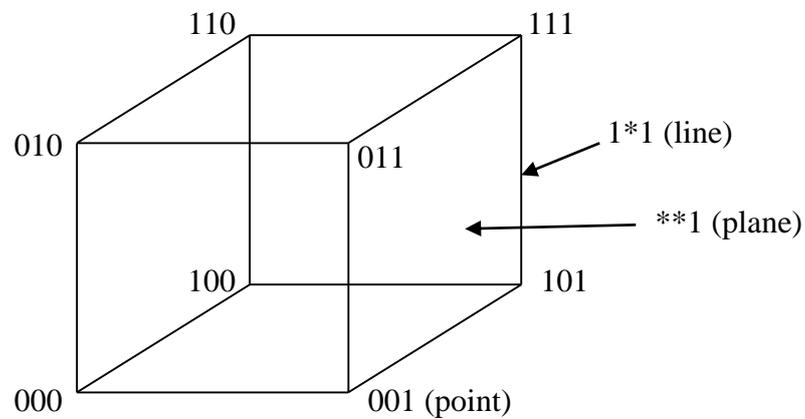
**Table 2.5** The example of schema and its properties

Schema Name	Schema	Schema Order	Length of Schema
Schema #1	1***0	2	2
Schema #2	**11*	2	1

Therefore the chance that schema #2 destroyed by crossover are less than schema #1 because the length of schema #2 is shorter than #1. Also, (Holland 1992) made a conclusion that highly fit and short length schemata (called "building blocks") are exponential increasing from generation to generation.

Finally, when capture the picture of the mechanism of the GA under its evaluation using schema that there are a lots more schemata processed by GA comparing to the number of population size. This bring the notion *Implicit Parallelism* which referred to the number of schemata processed by GA when a population was evaluated is greater than the number of population size. (Holland 1992)

(Holland 1992) also demonstrated the behavior of the GA on how it can give a robust search by using a sampling of hyperplane of a search space. The short of schemata of length  $L = 3$  was used for easier to draw a search space. According to the figure 2.6, all the point or corner in the space are represent by string of schemata order 3, the line are represent by the string of schemata order 2, the plane by string of schemata order 1 and the whole of the space by string of schema order 0 or the schema \*\*\*.



**Figure 2.6** The visualize of schemata as hyperplanes in 3 dimensional cube

#### 2.2.4 The mathematical explanation of GA

As mentioned in previous subsection, schema was used as a tool to capture the picture how GA is evaluated and evolved from generation to generation. This section dedicated to give the same picture but now with mathematical language. Holland (1975) developed the fundamental theorem of genetic algorithm called The Schema Theorem, to describe the change of sampling rate for a single schema from generation  $t$  to generation  $t+1$  (Whitley 1994).

Suppose in generation “ $t$ ”, population  $A(t)$  contained  $h$  strings (Chromosomes) of schema  $H$  which can be written as

$$h = m(H, t) \quad (\text{Eq. 2.89})$$

During the selection operation, where string  $A_i$  will be selected based on the probabilities

$$p_i = \frac{f_i}{\sum_{j=1}^{n-1} f_j} \quad (\text{Eq. 2.89})$$

Where

$p_i$  is the probabilities that string  $A_i$  will be selected

$f_i$  is the fitness of particular string  $A_i$   
 $f_j$  is the fitness of string  $A_j$  in population  $A(t)$   
 $n$  is number of population  $A(t)$

After selection, it's expect to have number of string that represent schema  $H$  in generation  $t + 1$  as followed

$$m(H, t + 1) = m(H, t) \cdot \left[ n \cdot \frac{f(H)}{\sum_{j=1}^{n-1} f_j} \right] \quad (\text{Eq. 2.91})$$

Where

$m(H, t + 1)$  is number of string represent schema  $H$  in generation  $t + 1$   
 $f(H)$  is average fitness of schema  $H$

From equation 2.91, it can be simplify and written as

$$m(H, t + 1) = m(H, t) \frac{f(H)}{\bar{f}} \quad (\text{Eq. 2.92})$$

Where

$\bar{f}$  is the average fitness of all string in the population at time  $t$

Please also noted that the equation 2.92 is reproductive schema growth equation which considered only selection operation not included crossover and mutation.

If  $f(H)$  greater than average fitness was rewrite as  $c\bar{f}$  where  $c$  is a constant then above equation can be rewrite as

$$m(H, t + 1) = m(H, t) \frac{(\bar{f} + c\bar{f})}{\bar{f}} = m(H, t)(1 + c) \quad (\text{Eq. 2.93})$$

From the equation above, one can see that the given equation looks like compound interest equation which can be describe as if the fitness of particular schema in current generation  $t$  is greater than population average fitness then  $c$  will be positive

and this schema will get increasing offspring in the next generation while if the fitness of schema lower and the average fitness then  $c$  will be negative and this schema will get decreasing offspring in the next generation.

Now, let's see the effect of the crossover which it may keep or destroy the schema based on a pattern of particular schema. As already mentioned in the previous section that the crossover is the genetic operator that will randomly select site for crossover within the selected string with the probabilities of crossover,  $p_c$ . Thus, the probabilities that schema will be destroyed ( $p_d$ ) can be calculated using equation 2.94

$$p_d = p_c \cdot \frac{\delta(H)}{(l-1)} \quad (\text{Eq. 2.94})$$

Where

$l$  is length of string

Therefore, from the equation 2.94 the probabilities that the schema will be kept ( $p_k$ ) can be calculated as

$$p_k = 1 - p_d \quad (\text{Eq. 2.95})$$

Thus, if combine the effect of selection and crossover by assuming the selection and crossover are independent then it will get

$$m(H, t+1) \geq m(H, t) \cdot \frac{f(H)}{\bar{f}} \left[ 1 - p_c \cdot \frac{\delta(H)}{l-1} \right] \quad (\text{Eq. 2.96})$$

From the above equation, an assumption can be made as the schema that has above average fitness and short length are going to be exponentially increasing.

Next, the mutation which is the last operator will be included to complete the schema theorem. According to mentioned in previous subsection, there is a probability called mutation rate ( $p_m$ ) that each gene in the string will be changed their value based on this probability. The mutation will effect only the position of the gene that not

contained \* or don't care symbol. Thus the number of gene in schema that might be modified by mutation is  $o(H)$ , as a result if each gene have survival rate of  $(1-p_m)$  then the schema is said to be surviving with the probability  $(1 - p_m)^{o(H)}$ . However, since the value of  $p_m$  usually has a very small value ( $p_m \ll 1$ ), thus the schema surviving probability can be written as  $1 - o(H) \cdot p_m$

Finally, the number of copies of particular schema in the next generation can be estimated under the rule of genetic evolution (selection, crossover and mutation) by the following expression which Holland (Holland 1992) called this equation: the schema theorem, or the fundamental of genetic algorithm.

$$m(H, t + 1) \geq m(H, t) \cdot \frac{f(H)}{\bar{f}} \left[ 1 - p_c \cdot \frac{\delta(H)}{l - 1} - o(H)p_m \right] \quad (\text{Eq. 2.97})$$

The above equation is used to predict the frequency of the particular schema. It was not completely capture details about GA e.g. the estimation from the population composition, speed of the solution convergence or the distribution of the fitness over time. There are several researcher such as (Vose and Liepins, 1991), (Nix and Vose, 1991) and (Whitley, 1993a). That construct the exact mathematical models of simple GA (Mitchell 1998).

### 2.3 Sliding Window

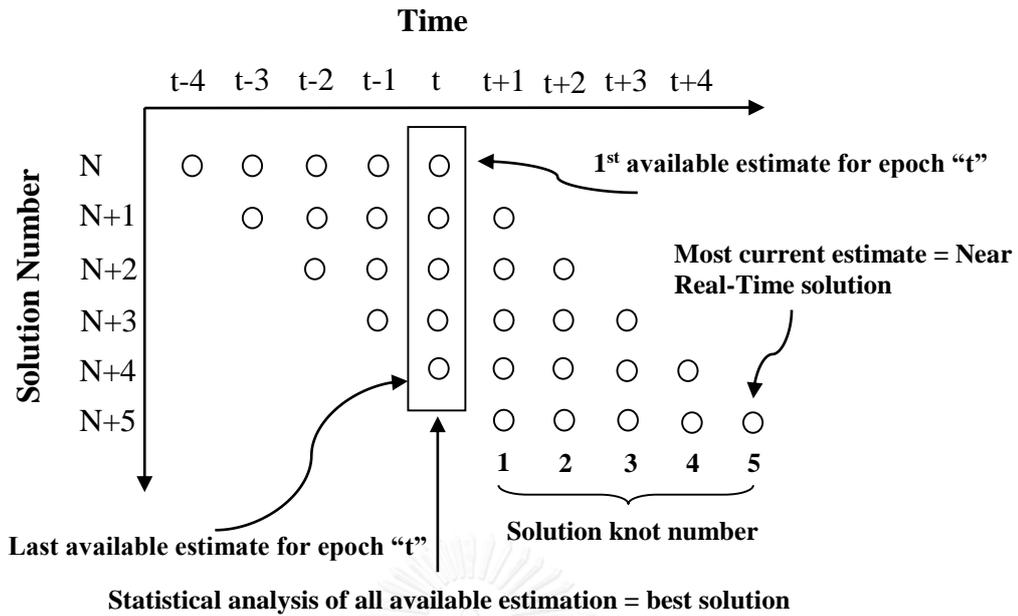
The sliding window or moving time window is the technique designed to reduce the computation work load of GPS data processing. Normally this technique is widely used in GPS meteorology (Foster, Bevis et al. 2005) and digital data communication. It can reduce work load for data processing by using finite time windows to filter the data for processing. This window will be slide along the time span to produce multiple solutions overlapping in time as shown in figure 2.7. Thus, beside from the single 24 hours batch post-processing which consume lots of computation time and resources for data processing, the several estimated solution in near real-time as well as the post-processing of statistical analysis will be produced instead.

According to figure 2.7, it reveals that the sliding window technique can produce several overlapped solutions which can be inspected by several statistical analyses. For example there are two kinds of solutions that can be extracted from the result shows in figure 1, the post-processing and near real-time solution respectively. The post-processing solution can be computed from the median of all available estimation from several solutions in the same epoch in time, in another hand the near real-time solution can be extracted from the latest epoch of time windows. In addition, this latest result is very useful in NRT monitoring such as Atmospheric monitoring.

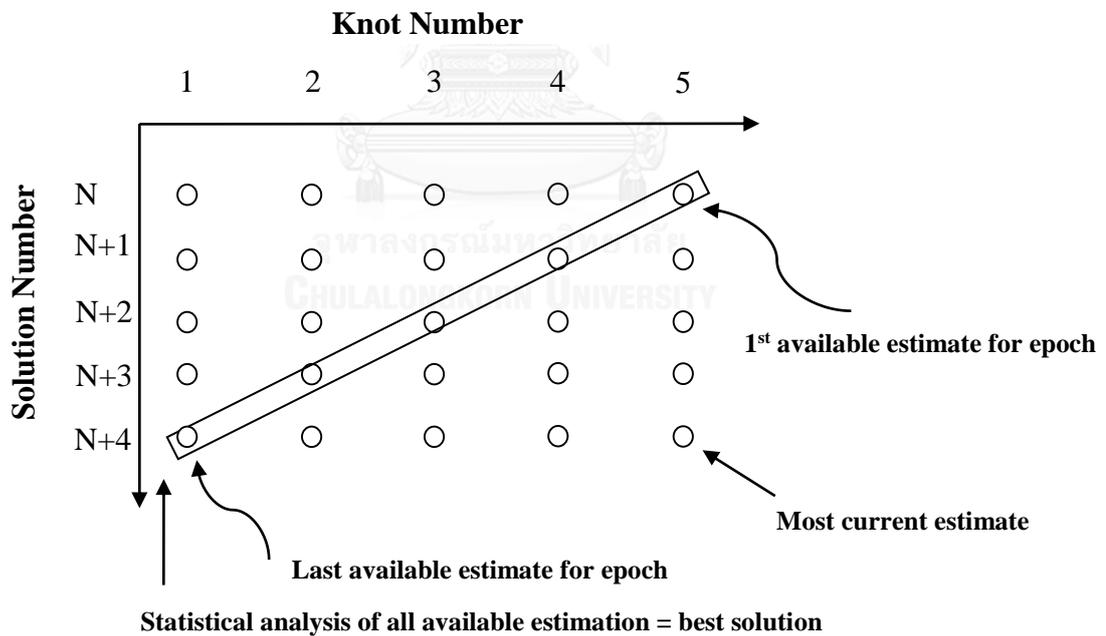
Because of its simplicity, light weight computation workload and statistical analysis capability, there are several GPS related applications using this technique, e.g. (i) (Zhang and Chu 2011) used sliding window to monitor the water vapor and create high accuracy of real-time tropospheric parameters from 6 IGS ground-based GPS stations. Moreover, the best time window width is also recommended to being 6-8 hours. (ii) (Yingni and Haifeng 2012) used Sliding window together with fading Kalman filter to improve the precision of long-baseline GPS kinematic positioning. (iii) (YoonSong and Kim 2012) create mobile device user position filtering using sliding window.

For deeper details, (Foster, Bevis et al. 2005) gave very precise and intuitive introduction about this technique. The sliding window can be described in two points of view: time-fixed and window-fixed respectively. Where the former give more natural and intuitive understanding as shown in figure 2.7 is the time-fixed sliding window

with 5 knots window width. The term “knot” is estimations within solution. Thus in solution “N” the time window start at time “t-4” and producing 5 knots of estimation which are at epoch “t-4, t-3...t-1 and t” respectively. Actually, in near real-time application the end of time window will be define to be the most current epoch in this case at time “t”. As time goes by the time window will be step forward producing another solution where the older epoch will be consecutively increase the number of estimation until the time window is pass by, as shows in figure 1 the first available estimation at epoch “t” is in solution “N” and the last available estimation at epoch “t” is in solution “N+4” respectively. These multiple estimation at the same epoch allowed statistical analysis to produce more meaningful result used in post-processing. In the latter case is window-fixed point of view, as shows in figure 2.8, sliding window process produce successive sequence of solution. Indicated as solution “N”, it composed of 5 knots estimation where knot 5 is normally the most current epoch of solution “N” and it is the first available estimation for that epoch. As time goes by, the solution N+1, N+2,... and N+4 will be produce respectively. Moreover, the estimation for the same epoch or knot 5 in solution “N” will be in the knot number “4, 3 and 2” and the last estimation of epoch will be in knot number 1 respectively. This window-fixed point of view may be look uncomfortable but it’s very useful for programming and solutions handling. However, the transformation between these two points of view is possible and is described below.



**Figure 2.7** The diagram of sliding-window in time-fixed point of view. The window is five knot width and the step ratio is 1. The reference time for solution N is shown as “t” (adopted from (Foster, Bevis et al. 2005))



**Figure 2.8** The diagram of sliding-window in window-fixed point of view. The window is five knot width and the step ratio is 1.

There are three parameters needed to initiate the sliding window which are position of the window, windows width and window step (or sliding step) respectively. The detail of initiation for each parameter is described below.

Firstly, for the position of the window, as mentioned earlier in near real-time application the position of the window usually defined the end of the time window to the most current epoch as  $t$ . As shows in figure 2.7, the window width is 5 knot thus in solution “N” the last knot or the end of time window was set to epoch at time “t”. In the case of window-fixed point of view shows in figure 2.8, the window width also 5 knot thus the knot number 5 of the solution “N” also usually set to the most current epoch.

Secondly, the window width ( $W$ ), is the parameter used to define how much of the data needed for processing usually the size of the window should be set to the number that not effect to the quality of the result. In piecewise-linear solution it can be specified the window width using equation 2.98

$$(n - 1)\delta = W \quad (\text{Eq. 2.98})$$

Where

$n$  is number of knots available in window

$\delta$  is knot interval (or knot step), the interval that the solution estimation appear repeatedly in the sliding window

Thirdly, the window step ( $\Delta$ ), is the time step which the window progress. This window step is usually related to knot interval because in practical this window step should be bigger than the knot interval thus to describe the ratio between them the “Step ratio” ( $S$ ) will be used and can be specified using equation 2.99

$$S = \Delta/\delta \quad (\text{Eq. 2.99})$$

It is crucial to define the reference epoch ( $\tau_0$ ) since it can be establish the time frame where the epochs can be identified by their “lead” or “lag” time relate to  $\tau_0$ . The epoch at earlier time comparing to  $\tau_0$  is called “lead” and those at later time called “lag”. As shows in figure 2.7 and 2.8, the appearance of the epoch was controlled by knot interval thus it can form the relation between lead, lag and knot interval comparing to reference epoch using equation 2.100.

$$\tau_i = \tau_0 - i\delta \quad (\text{Eq. 2.100})$$

Where

$\tau_i$  is lead time epoch if “i” is positive and lag time epoch if negative

$i$  is integer value represent the sequential epoch number comparing to the reference epoch

Similarly with the window time where it can be set the window reference time ( $T_0$ ) so each solution e.g. solution N, N+1, and so on it can be establish their sequential solution number related to  $T_0$ . The intuitive notation is written in equation below

$$T_j = T_0 - j\Delta \quad (\text{Eq. 2.101})$$

Where

$T_j$  is lead time solution if “j” is positive and lag time solution if negative

$j$  is integer value represent the sequential solution number comparing to the window reference time

From equation 2.100 and 2.101 by setting  $\tau_0 = T_0$ , the new equation describing the relation between epoch, knot and windows as shown below

$$i = jS + n - k \quad (\text{Eq. 2.102})$$

Where

$k$  is knot number within solution window

It also possible to calculate the number of estimate (N) available for that epoch or it can be called the “Order” from the following equation.

$$N(\tau_i) = \text{floor} \left( \frac{i}{S} \right) + 1, \quad \text{for } i = 0 \rightarrow n - 1 \quad (\text{Eq. 2.103})$$

Where “floor” is indicate the operation that resultant is rounded to next smaller integer

Lastly, the maximum number of estimates or maximum order ( $N_{max}$ ) for any epoch can be determined by the following equation

$$\text{Case 1: } \text{mod}\left(\frac{T - \tau}{\Delta}\right) = 0; \quad N_{max} = \frac{n + 1}{S} \quad (\text{Eq. 2.104})$$

$$\text{Case 2: } \text{mod}\left(\frac{T - \tau}{\Delta}\right) \neq 0; \quad N_{max} = \frac{n - 1}{S}$$

Where “mod” indicate the modulo 2 operation which is the remainder after division by 2

To be an example, the figure 2.7 which is five knot width and step ratio is 1 will be used. So if there is the question “what are the knot number and solution number of all estimation at epoch “t” start from solution “N”? the following is the way how to calculate

From equation 2.102

$$i = jS + n - k$$

Then

$$0 = j(1) + 5 - k, \quad k = \{1, 2, \dots, 5\}$$

$$j = -5 + k, \quad k = \{1, 2, \dots, 5\}$$

Because all knots of k can produces integer solution can matching with the solution number of j as summarized in table 2.6

**Table 2.6** The matching between knot number (k) and solution number (j) based on the computation of figure 2.7

Knot number (k)	Solution number (j)
1	-4
2	-3
3	-2
4	-1
5	0

It can be seen from the table 2.6 that all solution number are indicated lag solution because of minus sign. Thus it can be answer the question that all the estimation at spoch “t” start from solution “N” composted of 5 knots which are knot number 5 in solution N, knot number 4 in solution N+1, knot number 3 in solution N+2, knot number 2 in solution N+3 and knot number 1 in solution N+4 respectively.

Moreover it can be calculate the number of available estimate for that epoch as followed

From equation 2.103

$$N(\tau_i) = \text{floor}\left(\frac{i}{S}\right) + 1, \quad \text{for } i = 0 \rightarrow n - 1$$

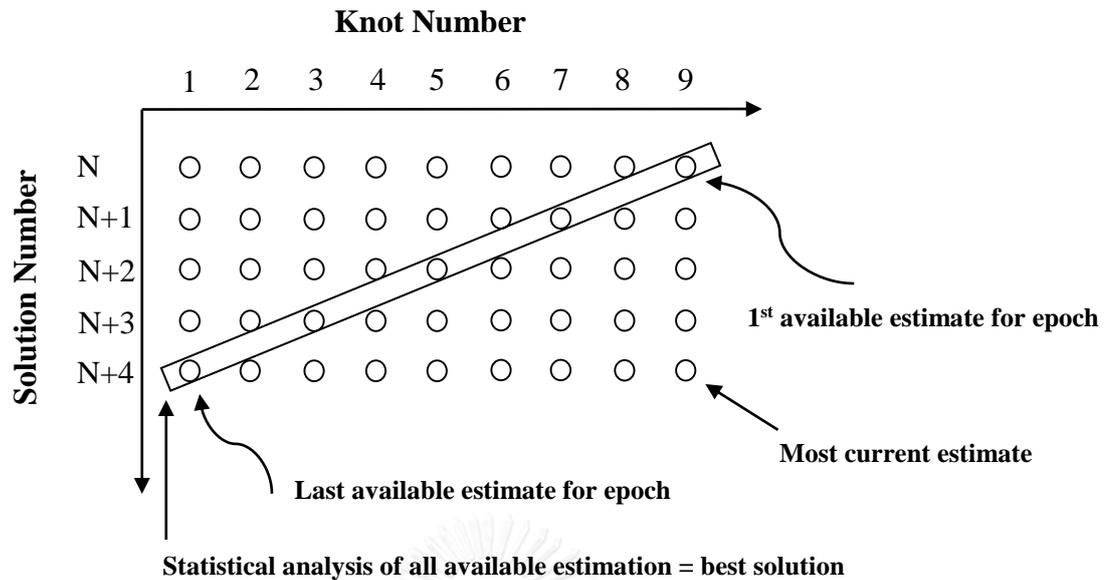
then

$$N(\tau_0) = \text{floor}\left(\frac{0}{1}\right) + 1, \quad \text{for } i = 0$$

Thus

$$N(\tau_0) = 1$$

This mean that at solution N where i=0 the order is equal to 1 thus there is only one estimation available which is knot 5, according to the table 2.6.



**Figure 2.9** The diagram of sliding-window in window-fixed point of view. The window is nine knot width and the step ratio is 2. (Adopted from (Foster, Bevis et al. 2005))

Another example based on figure 2.9 which is nine knot width and step ratio is 2. If the problem is locating all estimation at knot 5 in the solution (N+2), then following is the solution.

From 2.102

$$i = jS + n - k$$

Then

$$(9 - 5) = j(2) + 9 - k, \quad k = \{1, 2, \dots, 9\}$$

$$j = \frac{(-5 + k)}{2}, \quad k = \{1, 2, \dots, 9\}$$

Because only some value of  $k$  that produce integer solution which all matching were summarized in table 2.7

**Table 2.7** The matching between knot number (k) and solution number (j) based on the computation of figure 2.9

Knot number (k)	Solution number (j)
1	-2
3	-1
5	0
7	1
9	2

It can be seen from the table 2.7 that there are 3 lead solution which are knot 5 in solution N+2, knot 7 in solution N+1 and knot 9 in solution N respectively. Thus the two lag solutions which are knot number 3 in solution N+3 and knot number 1 in solution N+4 will be estimated in next two solutions.

Moreover it can be calculate the number of available estimate for that epoch as followed From 2.103

$$N(\tau_i) = \text{floor} \left( \frac{i}{S} \right) + 1, \quad \text{for } i = 0 \rightarrow n - 1$$

Then

$$N(\tau_4) = \text{floor} \left( \frac{4}{2} \right) + 1, \quad \text{for } i = 4$$

Thus

$$N(\tau_0) = 3$$

Regarding to the computation above, the order is 3 which related to the result show in table 2.7.

## 2.4 Literature Reviews

The GA has been introduced in the field of GPS data processing by many researchers. (Liu, Xiong et al. 2010) proposed the methodology to use GA for ambiguity resolution. The linearized double difference observation equation is used as the objective function in GA. Moreover the modified version of crossover operator that more fitted to research problem was also introduced. Together with, the technique to adaptively control the number of population which is called “adaptive multi-population controlled factor”. The result from the research indicated that the proposed technique can help to resolve for ambiguity resolution in GPS very well.

Another good example is the technique to use GA for GDOP classification. Normally, GDOP is a tool to measure the suitability of GPS distribution. Therefore, in this case GDOP is used to select the suitable satellite for precise positioning calculation. In this research, (Mosavi and Divband 2010) classified GDOP into three groups that are small, medium and large respectively. Only the GDOP that fall into the small group will be used for position determination. These result in lighter weight of computation load for GPS receiver position computation. Moreover, the comparison between the results from new proposed methods and the neural network which proposed in previous paper is made. The result of comparison indicated that GA can be compute more accurate and faster than his neural network proposed in his previous paper.

The sliding windows play important role in GPS meteorological data processing. Since the conventional GPS meteorological data processing required long observation data to obtained appropriate results which influenced computation time and resources. Moreover, there are requirement so called “real-time” or “near real-time” solutions where the result should be obtained as soon as possible after the latest data was retrieved. Zhang and Chu (2011) used sliding windows to obtained efficient real-time tropospheric parameters estimation for “Ground-based GPS Real-time Water Vapor Retrieval of North China” system. It was mentioned that the improvement of the resolution accuracy of the ambiguity to fulfill the demands of PWV computation accuracy required the sequence of long time observation which cost the computation resources and times. Therefore, the sliding windows was used to select the latest observation data added to the real-time data solution to increase the efficiency of real-

time property of data analysis and also the ambiguity resolution can be guaranteed with the least occupation of computation resources by the fixed width of time windows. In the experiments, the window width of time windows was varies between 2 to 22 hours and sliding step was fixed to 2 hours. The proposed method was used the data from six IGS station in china and compute for the ZTD using Bernese 5.0 GPS software. Results from the computation were compared with the CODE daily ZTD using standard deviation. The experimental result reveal that the increase of time windows also tends to increase the resolution accuracy. However, at the window width of 6 hours the resolution accuracy appear good enough. Regarding to the computation time, the result is under expected that the increase of time window lead to a longer computation time. According to the experiment the window width of 16 hours required twice computation time than that of window width of 6 hours. Thus it was concluded that sliding windows can gain high-accuracy of troposphere parameters estimation while meet the real-time demands the operation. Moreover, the best window width was concluded to be 6-8 hours.

Another example of sliding windows that was applied in GPS is found in the work of Yoonsong and Kim (2012). In this research the positioning data obtained from navigation system in mobile device are focused. Because of erroneous mainly from the environment that influence to the positioning data and lead to fault positioning. Therefore, the calculation of velocity and acceleration from the positioning data was introduced. Together with some statistical analysis for filtered out fault positioning data such as the moving average of speed, the moving standard deviation at a time and sensitivity level of filtering are also used. Moreover, since the computation resources in mobile platform are very limited thus the sliding windows was used to limited and filtered data that suit for mobile computation resources. The data used for experimental are based on the data collected over the area of Seoul, Korea more than 20 days using iPhone 3GS. In the experiment the window width was varies in the range of 5, 10, 25, 50 and 100 to inspect the effect to the calculation of moving average and moving standard deviation. The result from the experiment indicated that large windows size tend to produce under-filtering of erroneous data, where to small windows size tend to produce over-filtering of erroneous data. In the experiment indicate the windows size of 5 to 10 are suitable where it seems to produce correct positioning data. There are also

another experiment using difference equipment to check the positioning data accuracy. The data collected inside the building are fluctuated and meaningless where the data collected from outside building are accurate enough for precise localization. In the conclusion several weakness over the sliding windows technique are found and discussed. However, one notable limitation of the sliding windows is that it cannot work at the starting phase of data collection since there are not enough data to fill the window.



## Chapter 3

### Enhancement of the Ambiguity-Fixed Solution Rate in GPS Kinematic Positioning Using a Genetic Algorithm

#### 3.1 Introduction

The relative kinematic positioning mode is widely used in many applications, as it provides high-precision positioning, especially for moving platforms. There are various relative kinematic positioning techniques, such as RTK, NRTK, and PPK. Each of these techniques has its own advantages and disadvantages. For example, RTK and NRTK can obtain instantaneous precise positioning results in the field, whereas PPK has to first download and process the data. However, PPK mode can provide more reliable positioning results, and it does not require a communication link between the base and roving receivers. In practice, the PPK technique is still widely used in many applications (e.g., vessel tracking, aircraft tracking, monitoring of high-rise buildings and dams).

Generally, in good observation environments where the data observed by both the base station and the roving stations are highly correlated, the data differencing technique mentioned in the previous chapter can cancel most biases to compute an ambiguity-fixed solution. As a result, high-quality PPK positioning can be obtained. However, less favorable environments in which large numbers of outliers remain after data differencing make ambiguity-fixed solutions difficult to compute. This may result in unreliable positioning. In this case, users must edit or filter the noisy data during the processing step. Data filtering can be considered a trial and error process in which users re-process the data repeatedly until satisfactory ambiguity-fixed solutions are obtained. Often, this is a time-consuming process that requires the skills of an experienced user.

This chapter introduces a new method in which a GA is employed to overcome the abovementioned problem. The proposed technique will be called “GA-based method”, which seeks to optimize the combination of GPS satellites to yield the highest number of successful ambiguity-fixed solutions in PPK mode. Moreover, it is demonstrated using sample data from a dedicated network of GPS receivers in

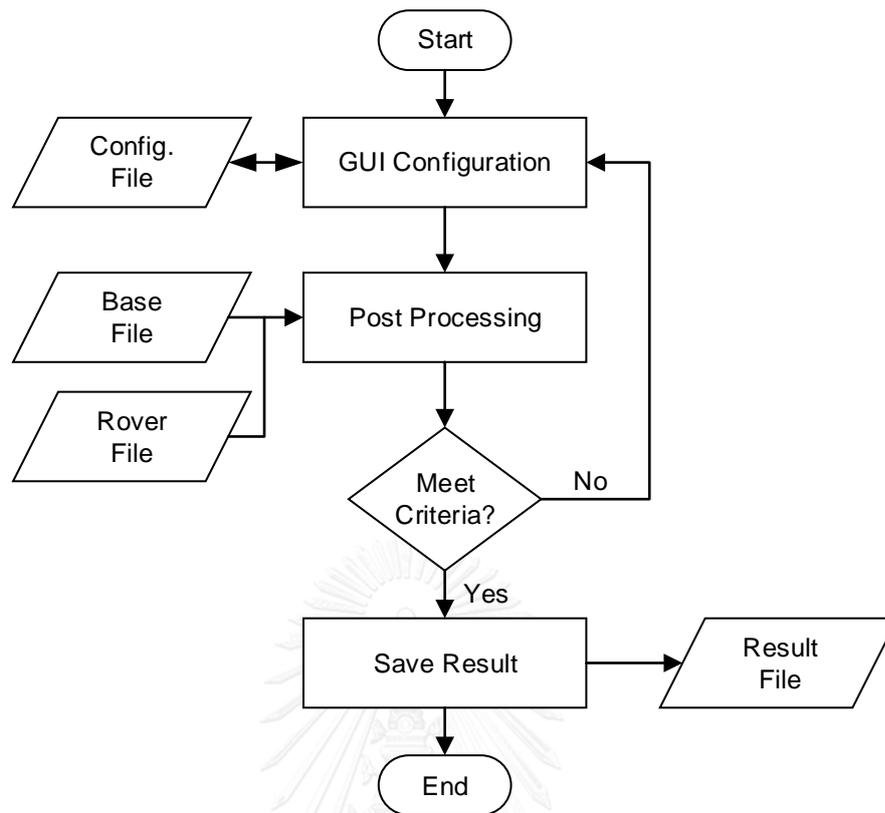
Thailand. The results indicate that the use of GA can produce more ambiguity-fixed solutions than the standard data processing technique. In addition, the use of a GA does not require expert users to conduct the manual data screening step.

### **3.2 GPS Kinematic Post-Processing using a GA**

This section the GA-based method will be explained. Starting from the background to GPS post-processing tools, which will explain the method of interaction between the user and the post-processing step together with their operating procedure. The modified GA used in this method is then explained, before the operating procedure of the integrated GPS PPK tool and GA is discussed. Finally, the real interaction between the GA and RTKLib (the selected GPS PPK tool) is explained.

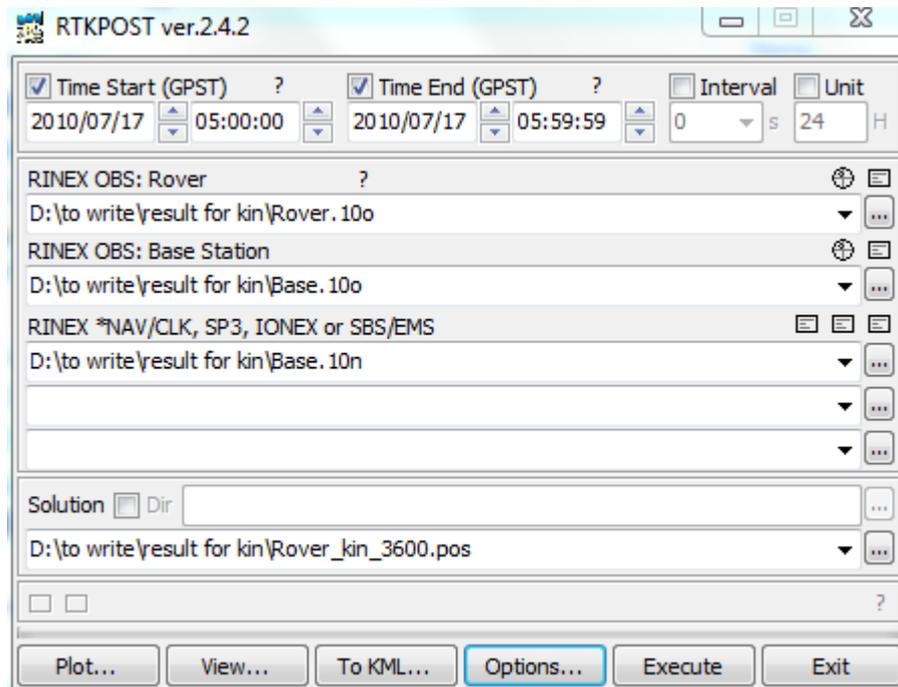
#### **3.2.1 Background to GPS Post-Processing Tools**

In GPS post-processing, users are typically familiar with the operating procedure shown in Figure 3.1. Users can access all functionalities through a graphical user interface (GUI). Functions include satellite system selection (e.g., GPS, GLONASS), elevation mask, and excluded satellites. Examples of specific GUIs are shown in Figure 3.2.

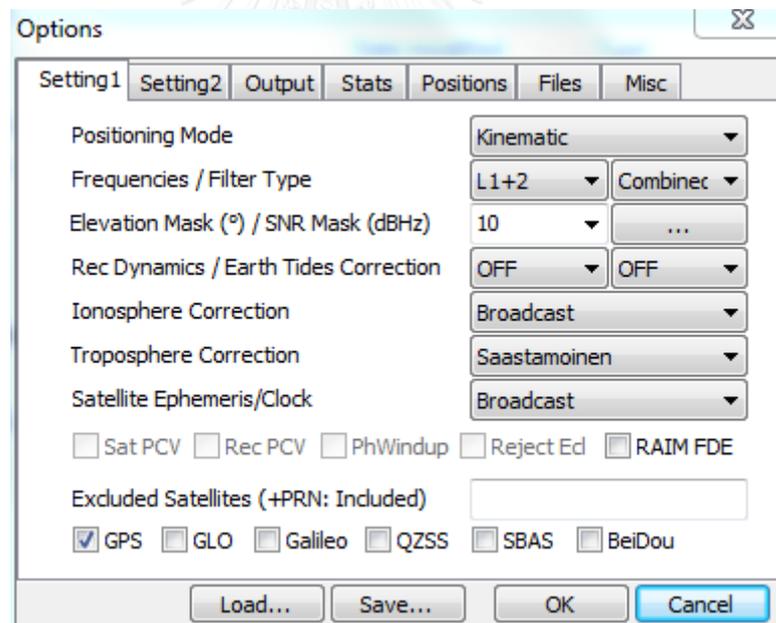


**Figure 3.1** Operating procedure of GPS post-processing using GUI

From the operating procedure shown in Figure 3.1, it can be seen that, besides the ability to select the configuration through the GUI, users can also store configurations in the memory for further processing. After the user has configured all functions needed for processing, these settings, together with raw observation data from the base station (the Base File) and the rover station (Rover File), are subjected to post-processing. The results of post-processing are then evaluated based on selected criteria, which are normally related to accuracy. If the results do not meet the selected criteria, the user can modify the configuration and repeat the post-processing. Once the results satisfy the criteria, they will be saved.



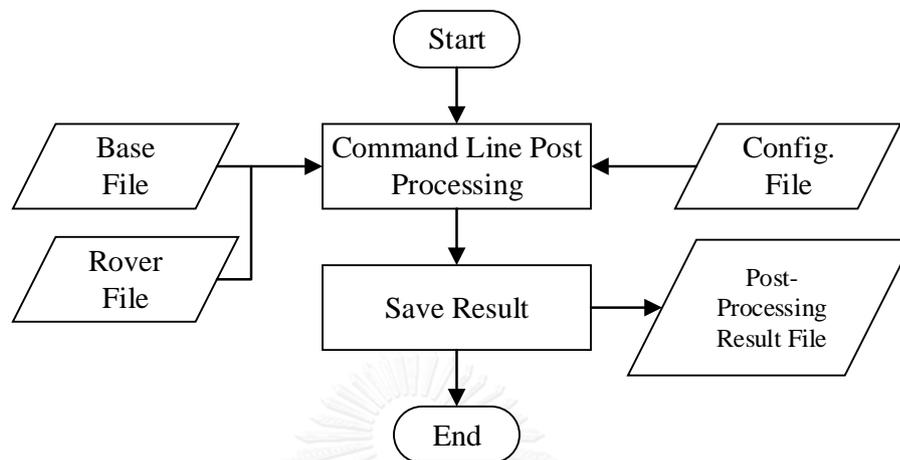
(a)



(b)

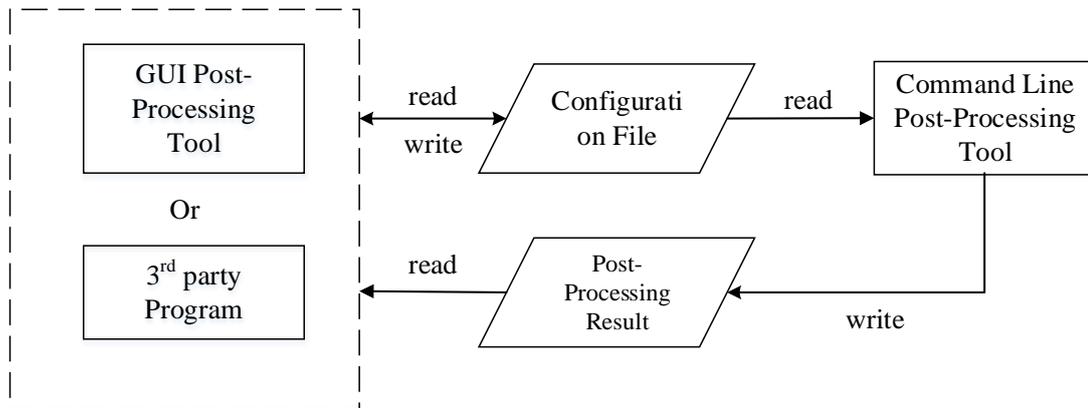
**Figure 3.2** GUIs of the GPS post-processing software: (a) User can select data, time span, solution and so on; (b) User can select processing options such as satellite system, positioning-mode, troposphere model, and so on.

As mentioned, this operating procedure is familiar to most users. However, it is also possible for users to perform the post-processing step without a GUI, as illustrated in Figure 3.3.



**Figure 3.3** Operating procedure for command line GPS post-processing tool

In this manner, users can perform the necessary post-processing from the command line using either a stored configuration from the GUI post-processing tool or a prepared configuration file. The difference between this operating procedure and that shown in Figure 3.1 is that there is no GUI that can be used to modify the configuration to process the data. Instead, in this technique, users have to modify their configuration file using either the GUI post-processing tool shown in Figure 3.1 or some third-party software such as a text editor. The relationship between these tools and the configuration file is illustrated in Figure 3.4. This figure reveals that third-party software can not only be used to modify the configuration file, but also to activate and retrieve the post-processed results for further analysis. This is the concept behind the integration of the GA and GPS PPK, which will be discussed in the next section. Normally, this technique is suitable for batch processing, whereby users run a command line post-processing tool through third-party software or a specially coded script. However, both techniques give identical results after post-processing.



**Figure 3.4** Interaction between GUI post-processing tool or third-party program and command line post-processing tool through the configuration file and post-processing results

### 3.2.2 Modification of the GA used in the Proposed Technique

According to the discussion of GA in the previous chapter, the key to modifying the standard GA for any problem lies in translating the scenario into the chromosomes and judging the quality of solutions. Thus, the two main modifications are to the population type and the objective function, as explained below.

- **Population Type**

As mentioned in the previous chapter, there are several types of population that GAs can understand, e.g., binary, real numbers. To obtain reliable and accurate positioning, the GA must determine the best combination of satellites. Thus, all satellites must be encoded into an appropriate format. In this study, satellites are encoded using the binary format. Therefore, each satellite will be represented by either '0' (off) or '1' (on). The satellite vehicle number (SVN) will be represented by a location bit in the bit string. Examples of encoding and its meaning are presented in Table 3.1.

**Table 3.1** The example of bit strings and their meaning

<i>Bit String</i>	<i>Meaning</i>
$[1,1,1,\dots,1]_{1 \times 32}$	<i>All satellites are used</i>
$[0,1,1,\dots,1]_{1 \times 32}$	<i>Only SVN = 1 is excluded</i>
$[0,0,0,\dots,0]_{1 \times 32}$	<i>All satellites are excluded</i>

- **Objective Function**

In any GA, it is crucial to define an appropriate objective function. The results obtained from the data processing engine will be used to evaluate the fitness of each individual member in the population. The objective function can be defined as

$$Fitness = \sum_{i=1}^n w_i X_i \quad (\text{Eq. 3.1})$$

In Equation 3.1,  $n$  is the number of fitness parameters,  $X_i$  are the fitness parameters, and  $w_i$  are the weights of the fitness parameters.

The number of ambiguity-fixed solutions, ambiguity-float solutions, and autonomous solutions are selected as fitness parameters. These numbers are extracted from the results obtained from the standard data processing step. In addition, the success of the solution can be judged by the number of ambiguity-fixed solutions, with more representing a better overall solution, followed by the number of ambiguity-float and number of autonomous solutions (i.e., more autonomous solutions indicates a lower success rate of the solutions). Thus, in this case, the simple penalty approach (Yeniay 2005) was used to reduce the fitness values of ambiguity-float and autonomous solutions by setting their weight values to 10 and 100, respectively. The weight for the ambiguity-fixed solutions was set to 1. Appendix 1 shows the MATLAB source code of the objective function.

A summary of the GA configuration is given in Table 3.2.

**Table 3.2** The configuration of GA within MATLAB

<i>GA Setting</i>	<i>Value</i>
<i>population type</i>	<i>Bit string</i>
<i>population size</i>	20
<i>objective function</i>	@myObjFunc
<i>number of generation</i>	30
<i>stall generation limit</i>	15
<i>initial population data</i>	[1,1,1,...,1] <sub>1x32</sub>
<i>scaling function</i>	Rank
<i>selection function</i>	roulette
<i>elite count</i>	2
<i>crossover fraction</i>	0.8
<i>mutation function</i>	constraint dependent default
<i>crossover function</i>	two point

### 3.2.3 Integration between GA and GPS Kinematic Post-Processing

The previous subsection revealed that the success of the integration depends mostly on the flexibility of the data processing engine (e.g., in terms of configuration for interaction with an external program). For the GUI post-processing, this integration might be in the form of plug-in tools or software modules that can be called internally. Thus, if the GUI post-processing tool does not provide these functionalities, integration will be impossible. However, integration is much easier with a command line post-processing tool such as that illustrated in Figure 3.4. The third-party software can interact with the command line post-processing tool via the configuration and post-processed results. Hence, integration between the GA and GPS processing in PPK mode was developed based on this idea.

It is relatively straightforward to apply the GA with GPS processing in PPK mode. The operating procedure is shown in Figure 3.5, and the details of each operation are as follows.

*First*, the GA randomly creates a population pool in which each individual contains information about one satellite combination.

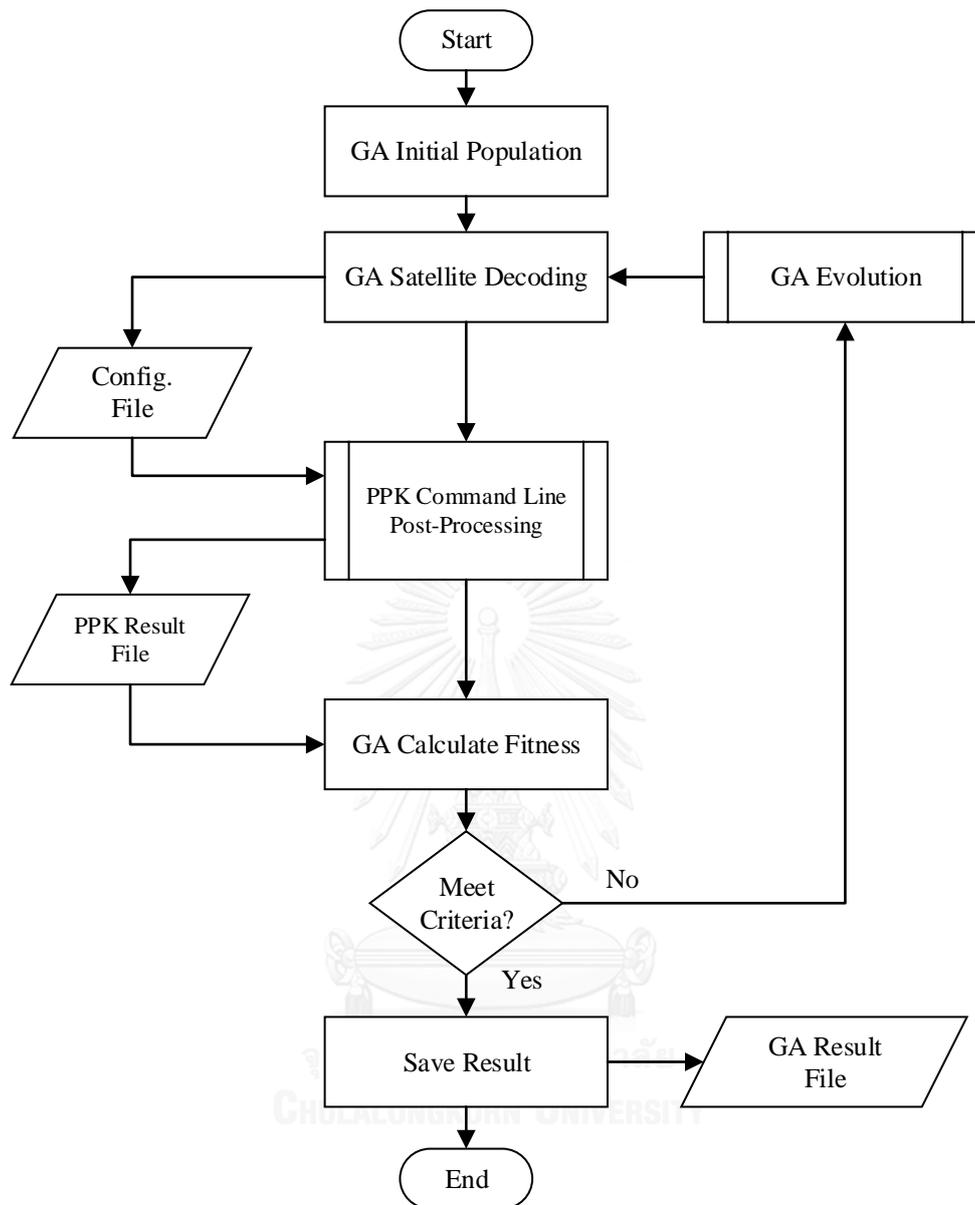
*Second*, each population will be decoded into a format understood by the data processing engine and passed through the data processing engine in the form of a configuration file.

*Third*, the data processing engine will be activated to carry on standard processing steps for the data of each individual in the population (as shown in Figure 3.3). After being processed, the data are fed back to the GA for evaluation.

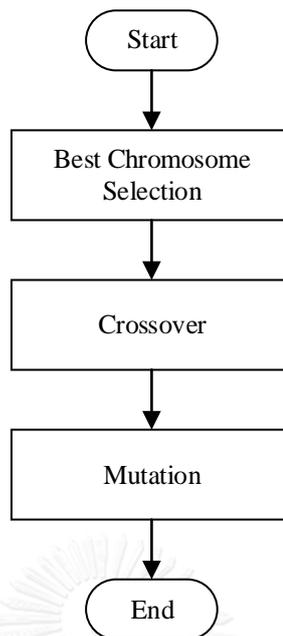
*Fourth*, the GA evaluates the level of success of each member of the population according to the objective function.

Next, the GA evolves as shown in Figure 3.6 using the GA operators. The GA selects two of the best members in the population pool to be parents for the next generation of chromosomes. The parents are then subjected to the crossover and mutation operators in the hope of creating alternative offspring that are stronger (i.e., better satisfy the objective) than their parents.

*Finally*, this workflow continues until the selected criterion is met.



**Figure 3.5** Operating procedure of optimized GA with PPK



**Figure 3.6** GA evolution composed of three operators (selection, crossover, and mutation)

To evaluate the performance of the GA-based method, the open-source RTKLib GNSS software package (version 2.4.1) was selected. The RTKLib Toolset can be easily modified to work with external programs (Takasu 2011).

### 3.2.4 Interaction between GA and RTKLib

This subsection explains the interaction process between GA and RTKLib. It begins with the population pool generated by the GA. Each individual in the population pool is a binary string representing a satellite combination, as illustrated in Figure 3.7. Note that the initial population includes a member that uses all satellites available for data post-processing (see Table 3.2). This ensures that the GA does not obtain a worse result than the standard RTKLib, because standard post-processing normally takes all available satellites into account.



the GA-based method and standard processing), and only the text string representing the excluded satellites will be modified. For more information about the RTKLib configuration file, see RTKLib user manual (Takasu 2011).

Next, the command line post-processing module in RTKLib is activated. After post-processing, the results are fed back to the GA and interpreted to count how many epochs produced ambiguity-fixed, ambiguity-float, and autonomous solutions, respectively. An example of the post-processed results is shown in Figure 3.10, where the grey box indicates each solution type in terms of a Q Flag value. The meaning of each Q Flag value used in this study is given in Table 3.3.

**Table 3.3** Q Flag values and their meaning

<i>Q Flag Value</i>	<i>Meaning</i>
<i>1</i>	<i>Ambiguity-Fixed</i>
<i>2</i>	<i>Float</i>
<i>5</i>	<i>Autonomous</i>

⌘ (x/y/z-ecef=WGS84,Q=1:fix,2:float,3:sbas,4:dgps,5:single,6:ppp,ns=# of satellites)									
⌘ GPST	x-ecef (m)	y-ecef (m)	z-ecef (m)	Q	ns	sdx (m)	sdy (m)	sdz	
2010/11/24 09:00:00.000	-1136983.6350	6091176.0395	1506867.0999	1	5	0.0255	0.0349	0.00	
2010/11/24 09:00:05.000	-1136983.6270	6091176.0196	1506867.0936	1	5	0.0289	0.0332	0.00	
2010/11/24 09:00:10.000	-1136983.6274	6091176.0265	1506867.0978	1	5	0.0000	0.0499	0.00	
2010/11/24 09:00:15.000	-1136983.6378	6091176.0018	1506867.0825	1	5	0.0358	0.0210	0.00	
2010/11/24 09:00:20.000	-1136983.6443	6091175.9998	1506867.0815	1	5	0.0000	0.0607	0.12	
2010/11/24 09:00:25.000	-1136983.6386	6091176.0179	1506867.0908	1	5	0.0000	0.0000	0.00	
2010/11/24 09:00:30.000	-1136983.6424	6091176.0045	1506867.0847	1	5	0.0000	0.0334	0.00	
2010/11/24 09:00:35.000	-1136983.6445	6091176.0153	1506867.0890	1	5	0.0000	0.0000	0.00	
2010/11/24 09:00:40.000	-1136983.6446	6091176.0101	1506867.0862	1	5	0.0000	0.0461	0.00	
2010/11/24 09:00:45.000	-1136983.6454	6091176.0298	1506867.0913	1	5	0.0000	0.0219	0.00	
2010/11/24 09:00:50.000	-1136983.6486	6091175.9964	1506867.0827	1	5	0.0000	0.0091	0.00	
2010/11/24 09:00:55.000	-1136983.6401	6091176.0071	1506867.0830	1	5	0.0000	0.0000	0.00	
2010/11/24 09:01:00.000	-1136983.6356	6091176.0288	1506867.0926	1	5	0.0420	0.0765	0.00	

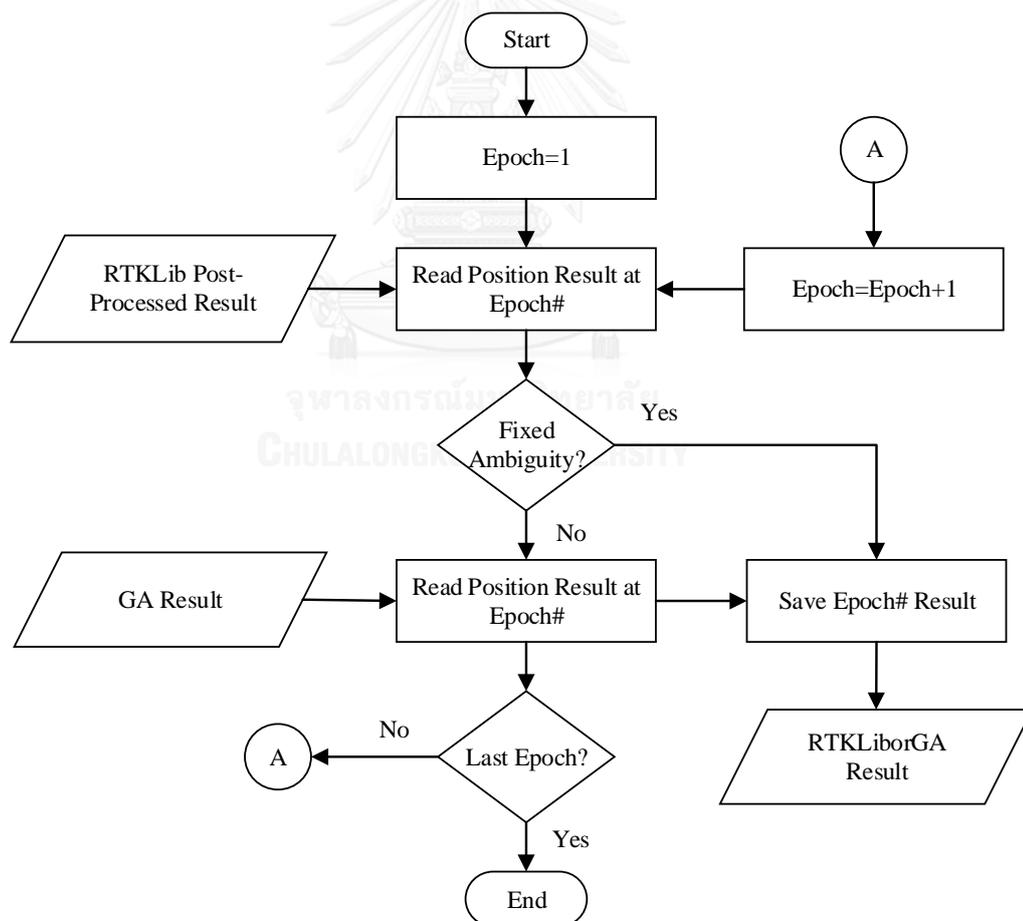
**Figure 3.10** Example showing the post-processed results, with Q Flag values highlighted

Finally, the number of epochs for each solution type is used to evaluate the strength of each individual according to the objective function. The GA evolution steps then continue until the termination criterion is satisfied.

### 3.2.5 RTKLiborGA

There is one further procedure to be explained, named RTKLiborGA. This procedure is designed to obtain fixed-ambiguity solutions from both standard RTKLib and GA to increase the number of ambiguity-fixed solutions, because the standard RTKLib solution is the initial population in the GA. From this perspective, RTKLiborGA can be considered as a representation of the results given by the GA-based method.

The idea behind RTKLiborGA is to take solutions from RTKLib only if the RTKLib solution is ambiguity-fixed; otherwise, RTKLiborGA takes the GA solution, regardless of type, as shown in Figure 3.11. Therefore, RTKLiborGA will give the highest number of ambiguity-fixed solutions, followed by GA and standard RTKLib.



**Figure 3.11** Operating procedure of RTKLiborGA

### 3.3 Experimental Data

The GPS data used in this experiment were collected using two dual-frequency geodetic receivers with a baseline length (distance between the two receivers) of approximately 17.6 km. The data were collected in static mode on November 20, 2010, using a common observation interval of 5-s and RINEX version 2.11 (Gurtner and L., 2007). Both stations had very good observation environments, as they are each set on top of a building (see Figure 3.13). For kinematic analysis, the PKKT station was selected as the reference station, and the DPT9 station was used as a roving station. To validate the performance of the GA-based method, the data obtained from DPT9 were treated as a kinematic dataset.

To convert the observation mode of the roving station from static to kinematic, the event flag denoting a moving antenna was changed from 0 to 2 during the first epoch of observation, as shown in Figure 3.12.

10	7	17	4	41	19.0000000	2	8G03G06G16G20G32R04R05R18
22310945.773	5	117244811.83515	91359621.57155	22310951.53145			
21609534.758	5	113558902.11715	88487453.17656	21609539.05946			
20171086.500	7	105999768.23117	82597240.05658	20171090.87548			
22297043.992	6	117171724.23116	91302696.44757	22297049.54347			
21326317.836	6	112070551.35916	87327728.73957	21326324.13347			
22229581.086	6	119038287.79316	92585416.38917	22229585.867	7		
19618454.328	8	104871884.42518	81567046.03219	19618462.438	9		
21495984.313	2	114747054.11112					

**Figure 3.12** Modified event flag (from 0 to 2) to indicate moving platform (Kinematic)



(top)



(bottom)

**Figure 3.13** Top: the PKKT base station; bottom: the DPT9 roving station

### 3.4 Experimental Results and Analysis

#### 3.3.1 Data Processing

First, a 24-h dataset from the DPT9 station was processed in static mode using the Leica Geomatic Office (LGO) software (version 5.0). The resulting coordinates were found to be accurate to within a few millimeters, and thus suitable for use as reference or “known coordinates” for later comparison with the processed kinematic data.

Kinematic data processing was carried out using the RTKLib software. The 24-h dataset was segmented into 24 1-h sessions. For subsequent data processing and analyses, these refer to the consecutive 1-h sessions using the letters A, B, C, D, and so forth. Under this scheme, each 1-h session was processed using both the GA-based method and the standard procedure (without the GA). Table 3.4 lists the satellites available for each session and the satellites excluded by the GA. The maximum number of fixed-ambiguity solutions will be obtained if some of the available satellites were excluded by the GA. There are only five sessions in which the GA can improve the number of ambiguity-fixed solutions in GPS PPK, namely sessions D, F, Q, R, and X.

The results obtained from the GA-based method were compared with those given by the standard procedure to determine the improvement in the number of

ambiguity-fixed solutions. Table 3.5 lists the percentage of ambiguity-fixed, float, and autonomous solutions for RTKLib, GA, and RTKLiborGA were listed respectively.

**Table 3.4** Satellites available in each 1-h session and satellites excluded by the GA

Session	Satellite Available	Num Sat Available	GA Excluded Sat	Num Sat Used In GA
A	3, 7, 8, 11, 13, 17, 19, 20, 23, 24, 28	11	-	11
B	7, 8, 11, 13, 17, 19, 20, 23, 24, 28	10	-	10
C	4, 7, 8, 11, 13, 17, 20, 24, 28, 32	10	-	10
D	2, 4, 7, 8, 10, 11, 17, 20, 28, 32	10	2, 10	8
E	2, 4, 7, 8, 10, 17, 20, 27, 28	9	-	9
F	2, 4, 5, 8, 9, 10, 12, 17, 20, 27, 28	11	17, 27	9
G	2, 4, 5, 9, 12, 13, 17, 27, 28	9	-	9
H	2, 4, 5, 10, 12, 17, 25, 26, 28	9	-	9
I	2, 4, 5, 10, 12, 15, 17, 25, 26, 29	10	-	10
J	2, 4, 5, 10, 12, 15, 18, 25, 26, 27, 29	11	-	11
K	2, 5, 9, 12, 15, 18, 21, 25, 26, 27, 29	11	-	11
L	2, 5, 9, 12, 15, 18, 21, 25, 26, 27, 29	11	-	11
M	5, 9, 14, 15, 18, 21, 22, 26, 27, 29	10	-	10
N	9, 10, 12, 14, 15, 21, 22, 25, 27, 29	10	-	10
O	9, 12, 14, 15, 18, 21, 22, 25, 27, 29, 30, 31	12	-	12
P	9, 12, 14, 18, 21, 22, 25, 30, 31	9	-	9
Q	12, 14, 16, 18, 21, 22, 24, 25, 30, 31	10	21	9
R	14, 16, 18, 22, 24, 25, 30, 31, 32	9	14, 30	7
S	3, 6, 14, 16, 18, 20, 22, 24, 25, 30, 31, 32	12	-	12
T	3, 6, 14, 16, 19, 20, 22, 23, 24, 30, 31, 32	12	-	12
U	3, 6, 13, 14, 16, 19, 20, 23, 30, 31, 32	11	-	11
V	3, 6, 13, 16, 19, 20, 23, 24, 31, 32	10	-	10
W	3, 6, 7, 11, 13, 16, 19, 20, 23, 24, 31, 32	12	-	12
X	3, 6, 7, 8, 11, 13, 17, 19, 20, 23, 24, 28	12	8, 17	10

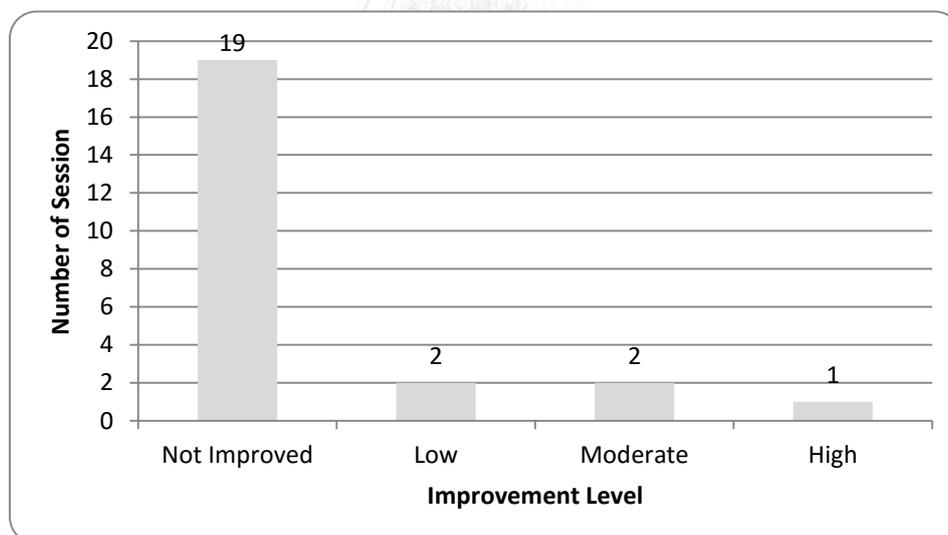
**Table 3.5** Comparison of solution types given by RTKLib, GA, and RTKLibOrGA

Session	RTKLib			GA			RTKLibOrGA			% Fixed Improved
	Fixed	Float	Auto	Fixed	Float	Auto	Fixed	Float	Auto	
A	57.40	37.20	5.40	57.40	37.20	5.40	57.40	37.20	5.40	0.00
B	12.80	87.20	0.00	12.80	87.20	0.00	12.80	87.20	0.00	0.00
C	0.00	100.00	0.00	0.00	100.00	0.00	0.00	100.00	0.00	0.00
D	0.00	100.00	0.00	47.22	52.78	0.00	47.22	52.78	0.00	47.22
E	0.00	100.00	0.00	0.00	100.00	0.00	0.00	100.00	0.00	0.00
F	46.80	53.20	0.00	56.90	43.10	0.00	64.86	35.14	0.00	18.06
G	0.00	100.00	0.00	0.00	100.00	0.00	0.00	100.00	0.00	0.00
H	0.00	100.00	0.00	0.00	100.00	0.00	0.00	100.00	0.00	0.00
I	0.00	100.00	0.00	0.00	100.00	0.00	0.00	100.00	0.00	0.00
J	0.00	100.00	0.00	0.00	100.00	0.00	0.00	100.00	0.00	0.00
K	0.00	100.00	0.00	0.00	100.00	0.00	0.00	100.00	0.00	0.00
L	0.00	100.00	0.00	0.00	100.00	0.00	0.00	100.00	0.00	0.00
M	0.00	100.00	0.00	0.00	100.00	0.00	0.00	100.00	0.00	0.00
N	3.80	93.30	2.90	3.80	93.30	2.90	3.80	93.30	2.90	0.00
O	0.00	36.20	62.80	0.00	36.20	62.80	0.00	36.20	62.80	0.00
P	0.00	100.00	0.00	0.00	100.00	0.00	0.00	100.00	0.00	0.00
Q	7.10	68.20	24.70	9.70	65.60	24.70	12.22	63.08	24.70	5.12
R	34.00	66.00	0.00	41.80	58.20	0.00	53.06	46.94	0.00	19.06
S	6.00	93.50	0.60	6.00	93.50	0.60	6.00	93.50	0.60	0.00
T	0.00	12.40	87.60	0.00	12.40	87.60	0.00	12.40	87.60	0.00
U	0.00	27.90	72.10	0.00	27.90	72.10	0.00	27.90	72.10	0.00
V	0.00	85.00	15.00	0.00	85.00	15.00	0.00	85.00	15.00	0.00
W	0.00	95.00	5.00	0.00	95.00	5.00	0.00	95.00	5.00	0.00
X	84.30	15.70	0.00	85.70	14.30	0.00	86.81	13.19	0.00	2.51

In addition, the percentage improvement indicated in Table 3.5 can be further classified. Four levels of improvement were thus defined according to the classification shown in Table 3.6. Using this classification, the frequency of each improvement level was counted, and the results are plotted in Figure 3.1.

**Table 3.6** Classification interval of improvement

<i>Level</i>	<i>Percentage of improvement</i>
<i>No improvement</i>	<i>0%</i>
<i>Low improvement</i>	<i>0.01–10.00%</i>
<i>Moderate improvement</i>	<i>10.01–20.00%</i>
<i>High improvement</i>	<i>&gt;20.01%</i>



**Figure 3.14** Classification of improvement in results given by the GA-based method

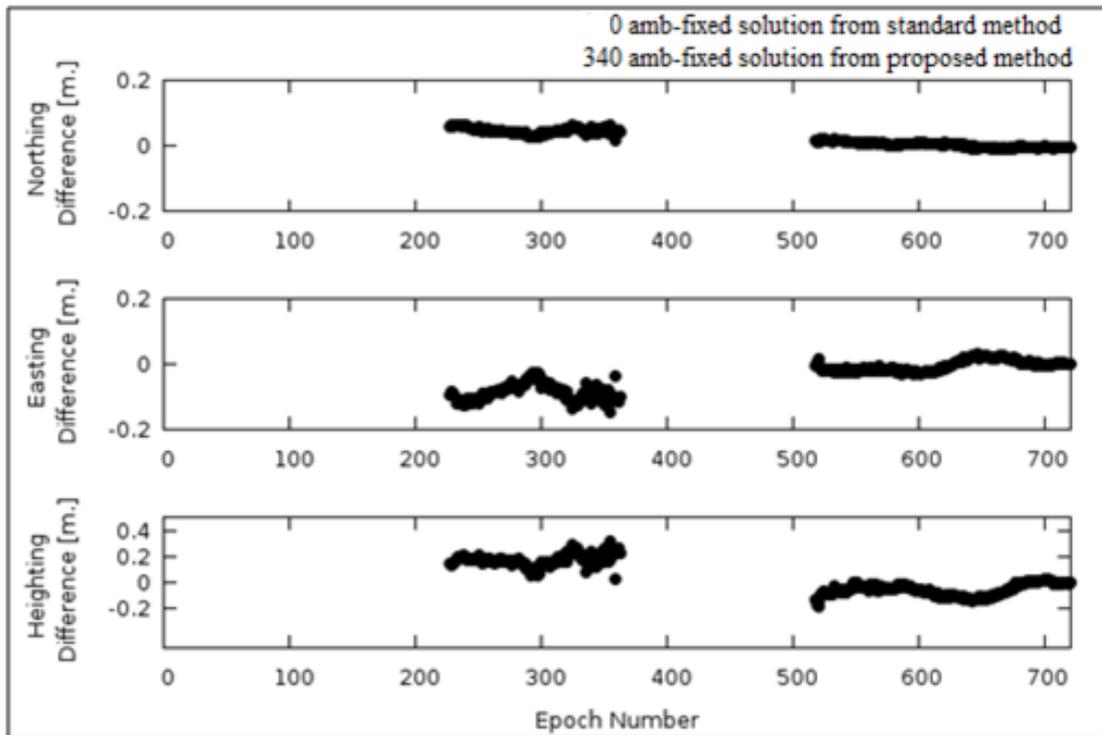
### 3.3.2 Analysis of Results

On the basis of the classified results shown in Figure 3.14, sessions D, F, and Q were selected as representative “High”, “Moderate”, and “Low” improvements, respectively. The best satellite combination for each of sessions D, F, and Q was identified by the GA (see Table 3.7). The respective discrepancies in the three coordinate components (N, E, H) (solutions compared to known coordinates) for each of these sessions are plotted in Figures 3.15, 3.16, and 3.17. In each subplot, the  $x$ -axis represents the solution epoch and the  $y$ -axis represents the magnitude of the error in meters. The grey line represents the results obtained from the standard PPK, whereas the black line represents the results from the GA-based method.

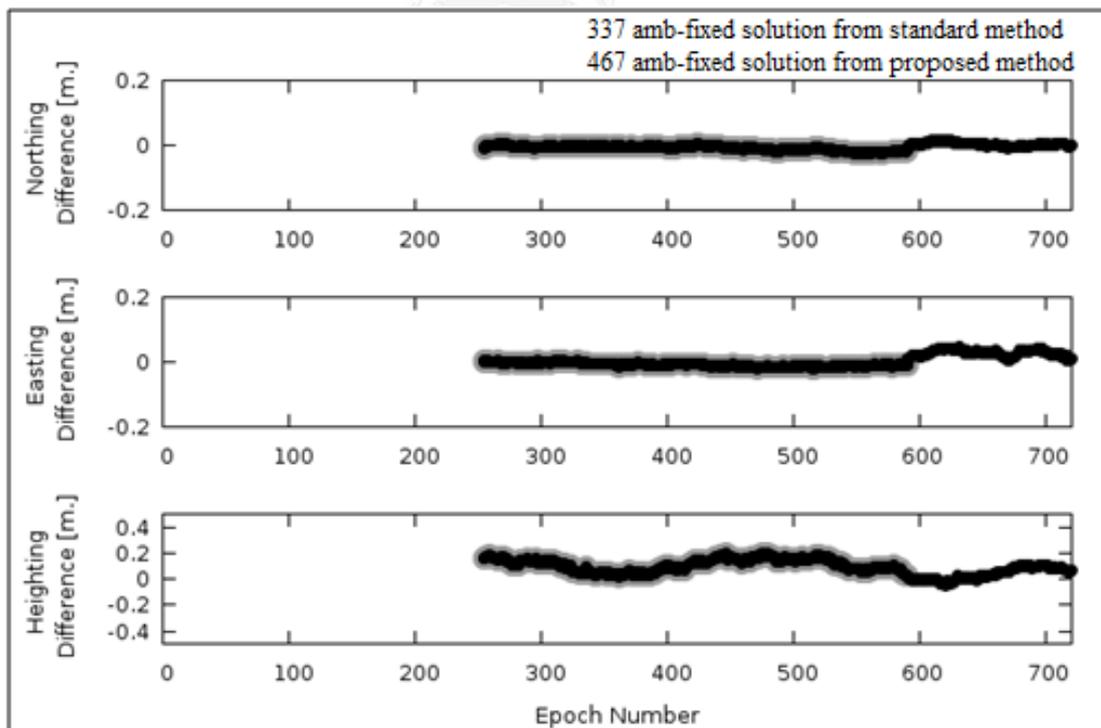
From Figure 3.14, it can be seen that the proposed procedure improved the number of ambiguity-fixed solutions in five of the 24 sessions. With reference to Table 3.7 and Figures 3.15–3.17, it is evident that the use of all available satellites may not always lead to the optimal solution. In some cases, excluding specific satellites from the data processing can yield better results. Thus, it can be concluded that the GA-based method improves the success rate of ambiguity resolution in the kinematic positioning mode. However, the positioning accuracy is not intention in this research thus the figures 3.15-3.17 are to illustrate the difference between the positioning obtained from the ambiguity-fixed of the GA-based method and the standard PPK method for each category of improvement in results given by the GA-based method. In addition, the objective function of the GA is only designed to improve the success rate of ambiguity resolution. It is very difficult to set the objective function to improve the positioning accuracy, because the true coordinates for each solution in the real world are unknown.

**Table 3.7** Best satellite combinations identified by the GA

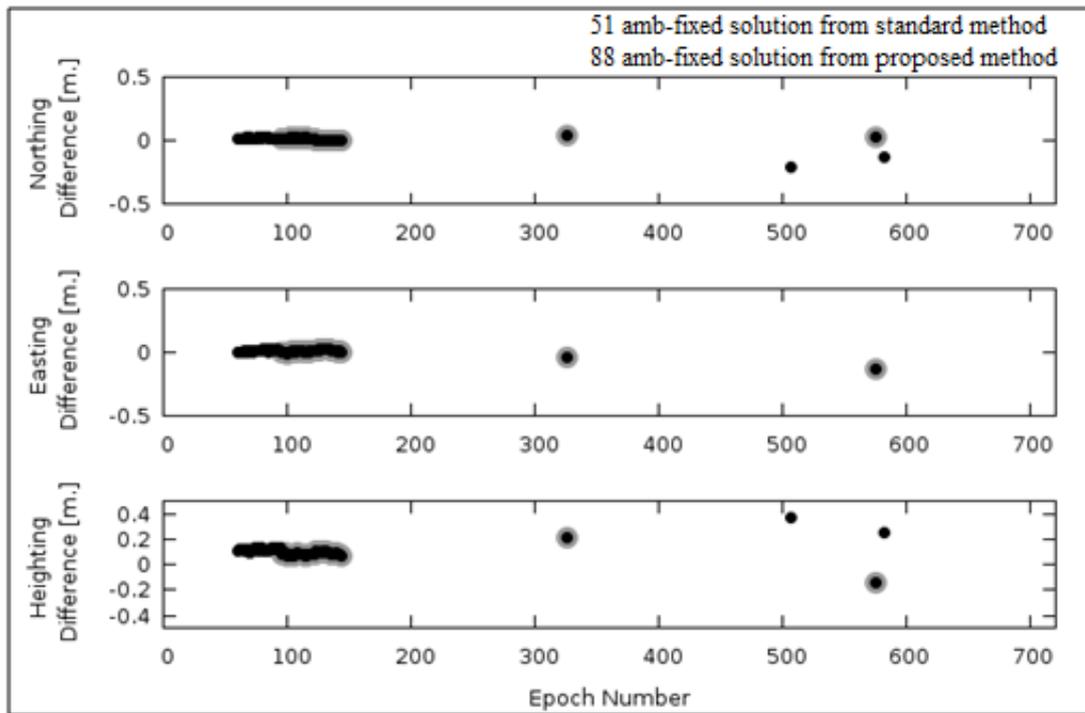
<i>Session</i>	<i>Available Satellites</i>	<i>Best satellite combination</i>
<i>D</i>	2, 4, 7, 8, 10, 11, 17, 20, 28	4, 7, 8, 11, 17, 20, 28
<i>F</i>	2, 4, 5, 8, 9, 10, 12, 17, 20, 27, 28	2, 4, 5, 8, 9, 10, 12, 20, 28
<i>Q</i>	12, 14, 16, 18, 21, 22, 24, 25, 30, 31	12, 14, 16, 18, 22, 24, 25, 30, 31



**Figure 3.15** Differences between the solutions and the known coordinates for session D (Top: North component; middle: East component; bottom: Height component)



**Figure 3.16** Differences between the solutions and the known coordinates for session F (Top: North component; middle: East component; bottom: Height component)



**Figure 3.17** Differences between the solutions and the known coordinates for session Q (Top: North component; middle: East component; bottom: Height component)

## Chapter 4

### **Enhancement of the Ambiguity-Fixed Rate in GPS Kinematic Positioning Using a Genetic Algorithm and Sliding Windows**

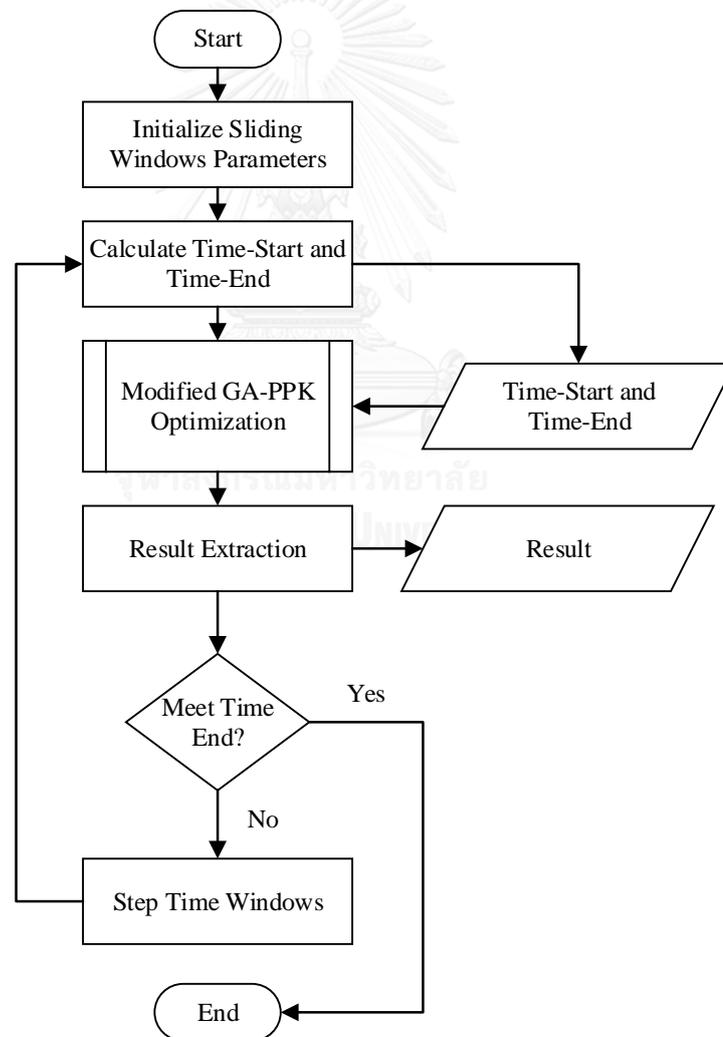
#### **4.1 Introduction**

In the previous chapter, an alternative to the trial and error process of identifying and removing noisy data was presented. The use of a GA to optimize the selection of GPS satellites successfully improved the number of ambiguity-fixed solutions compared with the standard computation in GPS PPK mode. However, because the data used in the demonstration were obtained from static observations, in which both the base station and roving station were stationary, the flag that indicated the positioning mode in the raw observation data had to be manually changed to signify kinematic positioning (for the roving station) in the pre-computation process. Consequently, the physical phenomena and satellite geometry of the roving station were considered to be correlated from epoch to epoch. In addition, the optimization of the satellite combination was performed on an individual 1-h session basis, whereas in practical kinematic positioning the result is based on an epoch by epoch basis. Therefore, for practical GPS observations in PPK mode where the roving station (and/or base station) is moving and the physical phenomena are considered to be varying, the results given by the method developed thus far may not be sufficient or optimized.

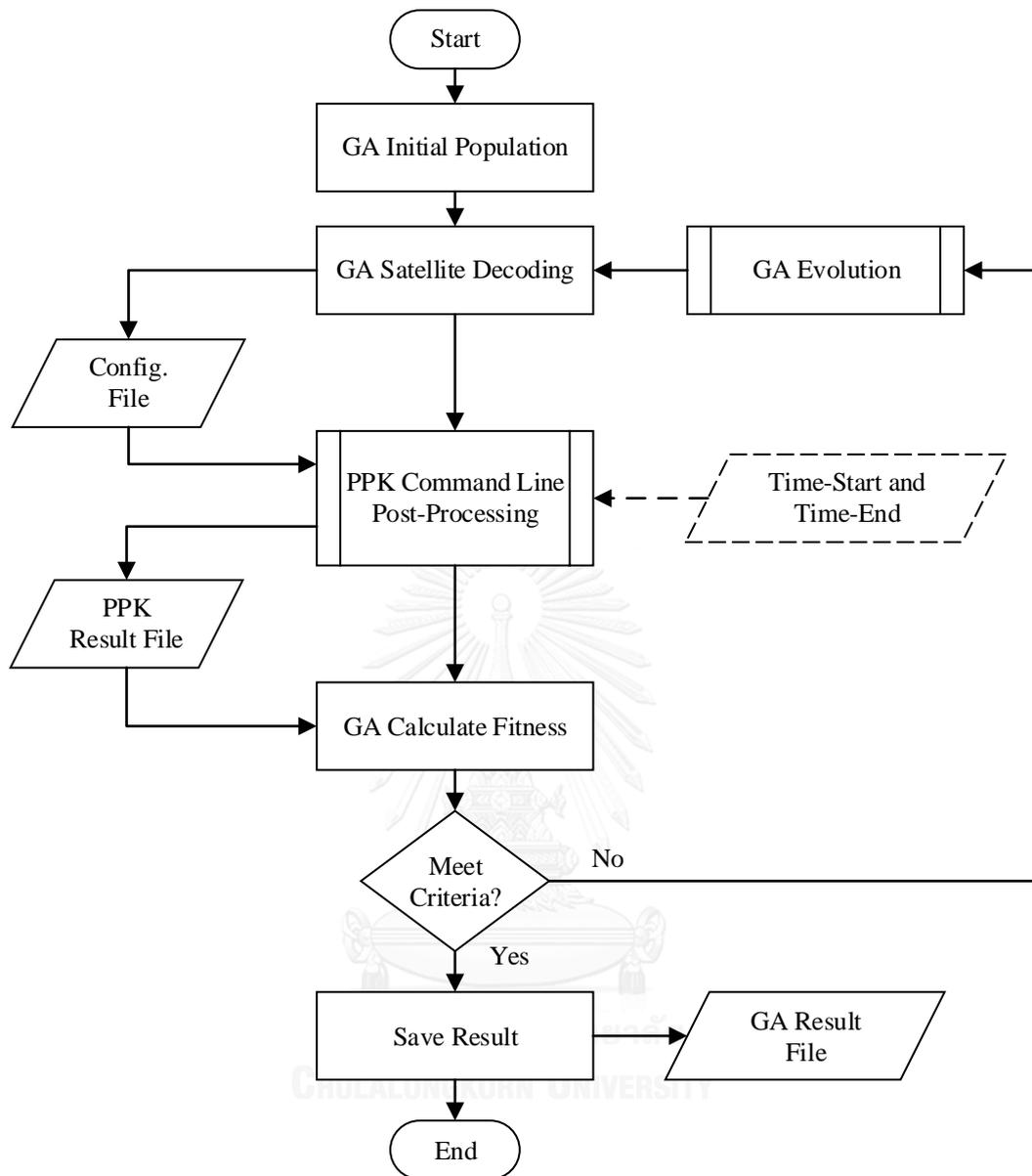
To overcome such problems, this chapter introduces a sliding window technique to the GA. The purpose of this sliding window integration is to increase the capability of the previous method to evolve the optimal satellite combination from a time-span basis (1-h sessions) into an epoch by epoch basis. The proposed method in this chapter and throughout this thesis will be called “GA-SW method”. The GA-SW method is demonstrated using practical mobile mapping observation data from Thailand. The results indicate that it produces a higher number of ambiguity-fixed solutions than the GA-based and standard methods.

## 4.2 Combination of Sliding Windows and GA in GPS Kinematic Positioning Mode

The previous chapter introduced the methodology of using a GA in GPS PPK mode, where the combination of satellites was optimized by the GA and used for an entire 1-h data processing session. However, this 1-h time-span basis might not be sufficient for practical GPS kinematic positioning in a dynamic environment, where epoch by epoch solutions are needed. Thus, in this chapter, the previous technique is enhanced by integrating a sliding window technique to create epoch-based solutions. The operating procedures of the GA-SW method are shown in Figures 4.1 and 4.2 the details of each process are discussed below.



**Figure 4.1** Operating procedures for the combination of sliding windows and GA in GPS kinematic positioning mode



**Figure 4.2** Sub-operating procedures of the modified GA-PPK Optimization

### 4.2.1 Initialize Sliding Windows Parameters

As shown in Figure 4.1, the enhanced operating procedure first initializes the sliding windows parameters of the “start epoch,” “window width,” and “sliding step.”

These parameters, which are necessary for the sliding windows technique, are described in detail in Chapter 2. They can be briefly described as follows.

- The “start epoch” is used to define the first epoch for which data must be processed.
- The “window width” is used to define number of epochs that must be processed.
- The “sliding step” is used to define the time interval needed to move the window forward by a defined interval.

In addition, the “start epoch” and “window width” should be initialized so as to yield sufficient data for the GPS post-processing engine to compute a solution. Note that there is no “end epoch” because the sliding window is designed to process all available data for a user-defined period.

### 4.2.2 Calculate Start Time and End Time

This process calculates the start time and end time from the predefined sliding window parameters. These values are used to filter data for the PPK engine, which will be described later in this chapter.

The calculation of the start time and end time depends on the desired solution. In this study the final epoch of the solution will be focusing on, as shown in Figure 4.3. The start and end times can be calculated as:

$$time^{start} = epoch^{start} - (window_{width} * sliding_{step}) / (24 * 3600) \quad (\text{Eq. 4.1})$$

$$time^{end} = epoch^{start} \quad (\text{Eq. 4.2})$$

Where

$time^{start}$  is the start time in the format YYYY/MM/DD HH:MM:SS

$time^{end}$  is the end time in the format YYYY/MM/DD HH:MM:SS

$epoch^{start}$  is the start of an epoch in the format YYYY/MM/DD HH:MM:SS

$window_{width}$  is the window width or a number of epochs

$sliding_{step}$  is the sliding step or the time interval between epochs (in seconds).

From equations 4.1 and 4.2, if the start epoch is “2010/07/17 06:00:00” the window width is 3600 epochs, and the sliding step is 1-s, then  $time^{start}$  will be “2010/07/17 05:00:01” and  $time^{end}$  will be “2010/07/17 06:00:00”

After the start and end times have been calculated, they are stored as internal variables for use by the Modified GA-PPK Optimization, which is described in the next section.

### 4.2.3 Modified GA-PPK Optimization

This section describes the main process of optimizing the satellite combination. This is a modified version of the optimization procedure described in the previous chapter, and is customized to accept the start and end time variables, as shown in Figure 4.2. Most of the sub-processes in Modified GA-PPK Optimization are equivalent to those of the GA Optimization described in Chapter 3. However, the interaction with the sliding windows (Figure 4.1) and the Modified GA-PPK Optimization process (Figure 4.2) have not previously been mentioned. Thus, they are described below.

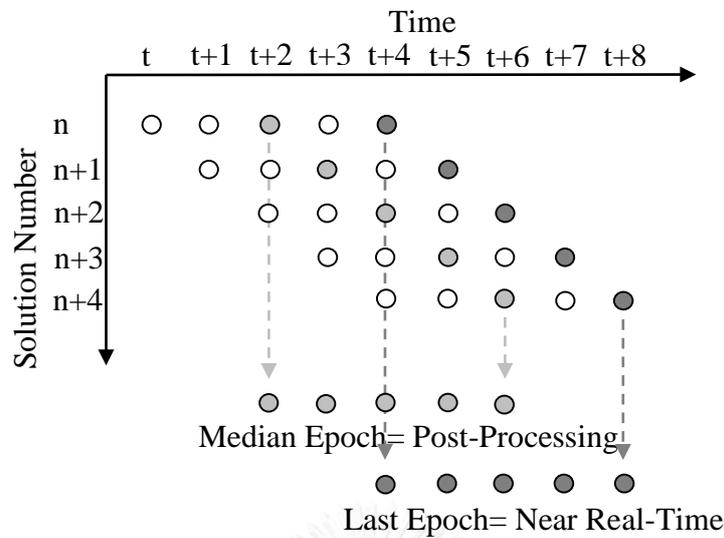
1. In Figure 4.1, once  $time^{start}$  and  $time^{end}$  have been calculated from the “Calculate Time-Start and Time-End” process, the “Modified GA-PPK Optimization” process is triggered and the  $time^{start}$  and  $time^{end}$  variables are stored as internal variables.
2. The overall Modified GA-PPK Optimization process is shown in Figure 4.2. This process commences optimization within the given time window.
3. The optimization process starts with the “GA Initial Population” process, where the GA randomly creates a population pool of individuals containing information about satellite combinations.

4. Each population is decoded into a format understood by the data processing engine and passed to the data processing engine in the form of a configuration file.
5. The data processing engine is activated to perform the standard processing steps for each individual in the population. After processing, the results are fed back to the GA for evaluation.
6. The GA evaluates the level of success of each member of the population according to the objective function.
7. The GA evolution process, illustrated in Figure 3.6, applies the three GA operators to the population. The GA first selects two of the best members in the population pool as parents for the next generation. The parents are then subjected to the crossover and mutation operators in the hope that this will create alternative offspring that are stronger (i.e., better satisfy the objective) than their parents.
8. This workflow continues until the termination criterion is met.
9. Once the criterion has been met, the “GA Result File” is saved and the “Result Extraction” process shown in Figure 4.1 is called.

As already mentioned, most of the processes shown in Figure 4.2 are the same as those used in the GA optimization described in the previous chapter. The “PPK Command Line Post-Processing” has been modified to accept the  $time^{start}$  and  $time^{end}$  variables. This process was easy to modify, since the same processing engine namely “RTKLib” was utilized, whose command line post-processing engine (rnx2rtkp) accepts  $time^{start}$  and  $time^{end}$  by default (see Figure 4.4). For more information about rnx2rtkp options, see RTKLib user manual (Takasu, 2011).

#### 4.2.4 Result Extraction

This process extracts the results from GA-PPK optimization and saves the extracted results to the global solution result file, which contains the results for each GA-PPK optimization. The result extraction process is dependent on the desired solution, e.g., median or final epoch. In this study, the final epoch was used. Thus, the epoch at  $time^{end}$  is used to represent the results of the GA-PPK optimization.



**Figure 4.3** Difference in sliding window solutions between Median Epoch and Last Epoch

```

rnx2rtkp -ts str_ts -te str_te -o ofil -k confil BaseFile RoverFile NavFile
Example
Rnx2rtkp -ts 2002/10/26 14:00:00 -te 2002/10/26 15:00:00 -o output.pos -k
config.txt basefile.02o roverfile.02o navfile.02n
where
-ts ds ts      start day/time (ds=y/m/d ts=h:m:s) [obs start time]
-te de te      end day/time (ds=y/m/d ts=h:m:s) [obs start time]
-o output      output file
-k file        input options from configuration file

```

**Figure 4.4** Example of time window options in rnx2rtkp

#### 4.2.5 Moving the Time Windows

This is the final process in the GA-SW method. If  $time^{end}$  has not been reached,  $epoch^{start}$  is incremented by  $sliding_{step}$ , and further optimization is performed; otherwise, the process is terminated.

In this chapter, the default GA optimization for PPK is the same as for the GA-based method. Thus, all default configurations for both the GA solver and the objective function are the same. The only modification to the previous method is to adjust the MATLAB code for the GA to accept the  $time^{start}$  and  $time^{end}$  of the sliding window and pass these parameters to RTKLib for processing.

### 4.3 Experimental Data

The data used in this experiment were extracted from a real kinematic environment on July 17, 2010. The base station was set up using dual-frequency GNSS (Novatel ProPak V3) and a GNSS antenna (Novatel GPS-702-GG) attached to the benchmark in front of the Survey Building, Department of Land, Pakkret, Nonthaburi Province, Thailand (see Figure 4.5). The official coordinates for this benchmark are given in Table 4.1. The roving station was set on the rooftop of a car using a multi-frequency receiver (Trimble R8, yellow label in the lower left of Figure 4.6). The trajectory of the car was collected along the road inside “Muangthong Thani” which can be considered a suburban area. All data for both receivers were recorded at 1 Hz. The duration of the full dataset is about 2-h 54-min 15-s. (These data have been used in a previous study (Bureerak 2010). Various other data were recorded during the experiment; hence, there are other instruments attached to the rooftop of the car, e.g., Ladybug Camera, Novatel SPAN-CPT, and Novatel GPS-703-GG; see Figure 4.6.)

## 4.4 Experimental Results and Analysis

### 4.4.1 Data Processing

The main focus of this study is kinematic positioning for data processing. The selected processing software was RTKLib (version 2.4.2), which is an open-source project under the BSD 2-clause license and an additional two exclusive clauses (Takasu 2013). The processing was divided into three different strategies, a standard method (without the aid of sliding windows and the GA), the GA-based method (using only the GA), and the GA-SW method (using both sliding windows and the GA). All data processing strategies used the same core data processing module of rnx2rtkp, the command-line user interface application program in the RTKLib project. Thus, the configuration of rnx2rtkp was fixed and shared among all processing strategies (see Table 4.2).

This study evaluated the performance of the GA-SW method using data from a portion of the total trajectory (see Figure 4.7). The dataset was composed of 227 epochs, from 06:00:00 to 06:03:46 GPST. The environment along the trajectory was relatively clear, with only a few obstructions. The distance to the base station varied from 0.8–

2.5 km. Limitations in the sliding window technique where the data for processing must be the same length as the window width.

In this experiment, the window width was specified to have a 1-h time span (Table 4.3). As a consequence, the raw observation data needed for processing must cover the duration from (start time – window width) to (end time). In this case, the data ranged from 05:00:00 to 06:03:46 GPST. These data were used in all computations, but only the results of selected trajectories are extracted and analyzed. Note that the window width used in this study has not been optimized, and different window widths may give different results.

**Table 4.1** Official coordinates of the benchmark at the base station (WGS84)

Latitude	13° 54' 43.02384'' N
Longitude	100° 32' 22.36937'' E
Height	-27.622 m



**Figure 4.5** Benchmark of base station located in front of the Survey Building, Department of Land, Pakkret, Nonthaburi province, Thailand



**Figure 4.6** Instruments attached to the rooftop of a car. The Trimble R8 used as the rover for this study is shown in the lower left.

The following describes the details of each processing strategy. First, the standard method processed a fully prepared dataset using the given configuration by `rn2rtkp` without any aid of GA or sliding windows. Thus, approximately 3827 solutions were available. However, due to the selected trajectory, only solutions from 06:00:00 to 06:03:46 GPST were extracted, and these will represent the results from the standard method.

Second, the GA-based method processed the prepared data to determine the satellite combination that produced the highest number of ambiguity-fixed solutions. The use of the GA in PPK positioning mode is illustrated in Figure 3.5 (previous chapter). As for the first strategy, only the results that matched the selected trajectory were extracted.

Finally, the GA-SW method used the sliding window parameters given in Table 4.3 to match the selected trajectory. The data were then processed and optimized using the GA. The window width was 1-h and the start epoch was at 06:00:00 GPST. The results of the sliding window were extracted from the last epoch. In this experiment, the GA-related configuration and objective function were unchanged from those in a previous chapter.

**Table 4.2** Main configuration parameters used in rnx2rtkp of RTKLib

Parameters	Value
Satellite System	GPS only
Positioning Mode	Kinematic
Frequencies	L1+2
Filter Type	Combined
Elevation Mask	10
Receiver Dynamic	Off
Ionosphere Correction	Broadcast
Troposphere Correction	Saastamoinen
Satellite Ephemeris	Broadcast
Integer Ambiguity Resolution	Continuous
Validation Threshold to Fix Ambiguity	12

The results from all strategies were compared in terms of the number of ambiguity-fixed solutions. To compute the success rate from the experiment, the following expression was used:

$$\%success = \frac{n \times 100}{N + 1} \quad (\text{Eq. 4.3})$$

Where

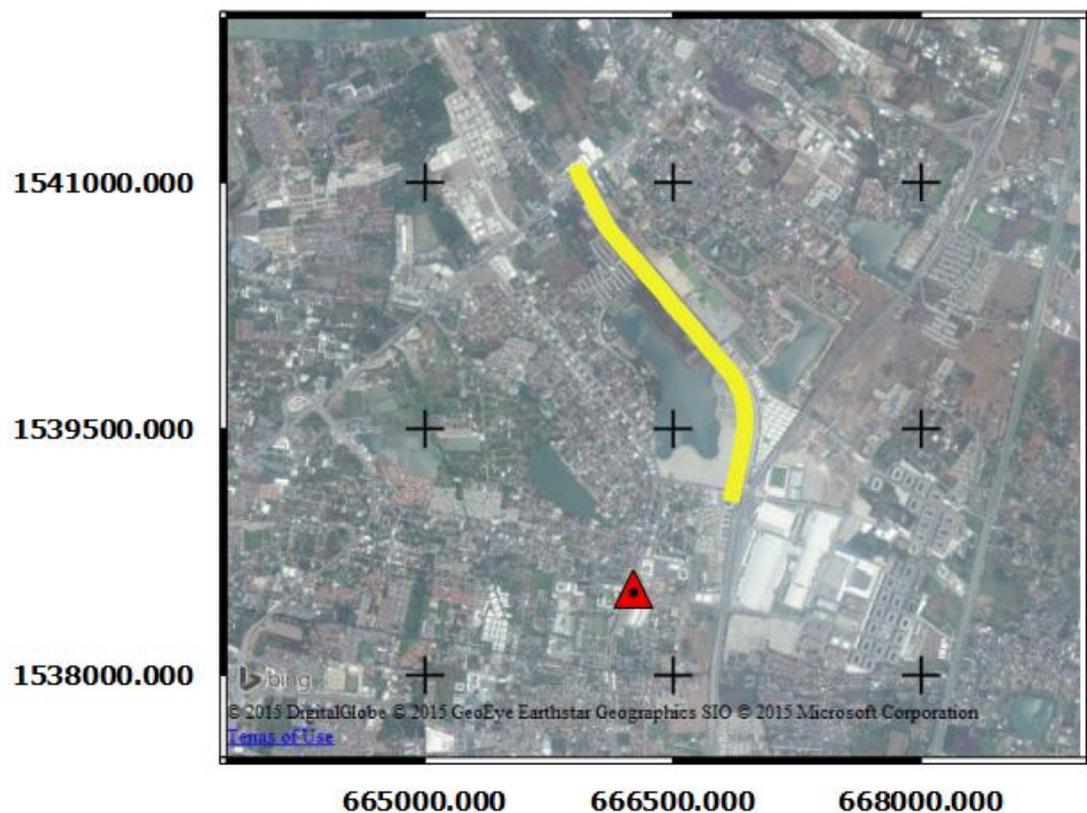
$\%success$  is success rate of fixed-ambiguity.

$n$  is the total number of ambiguity-fixed solutions found.

$N$  is the total number of solutions

**Table 4.3** Sliding window parameters used in the GA-SW method

Start Epoch	06:00:00 GPST
Window Width	1-h
Sliding Step	1-s

**Figure 4.7** Base Station (red triangle) and trajectory of the rover station (thick yellow line running from lower right to upper left) along the pond in Muangthong Thani, Thailand (Source: Bing Map)

#### 4.4.2 Results and Discussion

Figures 4.8, 4.9, and 4.10 show the processed trajectories given by the standard, GA-based, and GA-SW method, respectively. The dark color in the figures represents the ambiguity-fixed solutions, and the light-grey color represents other solutions, e.g., ambiguity-float and autonomous. According to Figure 4.8, some parts near the beginning of the trajectory indicate ambiguity-fixed solution epoch given by the

standard ambiguity computation, but these are very limited. In Figure 4.9, there is an improvement in the number of ambiguity-fixed solution epochs, especially near the beginning of the trajectory. However, in both Figure 4.8 and Figure 4.9, the number of ambiguity-fixed solution epochs in the middle and at the end of the trajectory is very limited. However, the results in Figure 4.10, given by the GA-SW method, show that most epochs along the trajectory are ambiguity-fixed solutions.

Moreover, after data processing, the number of ambiguity-fixed solutions given by each method was counted. The results are given in Table 4.4. From this table, it is clear that the GA-based method gives an improvement in the number of ambiguity-fixed solutions, from 43 to 85 epochs, compared with the standard method. What is remarkable, however, is the number of ambiguity-fixed solutions found by the GA-SW method. This gives an increase from 43 to 173 epochs, which represents an increase from 18.94% to 76.21%, over the standard method, and is twice as good as the GA.

**Table 4.4** Comparison of results given by the standard, GA-based, and GA-SW method

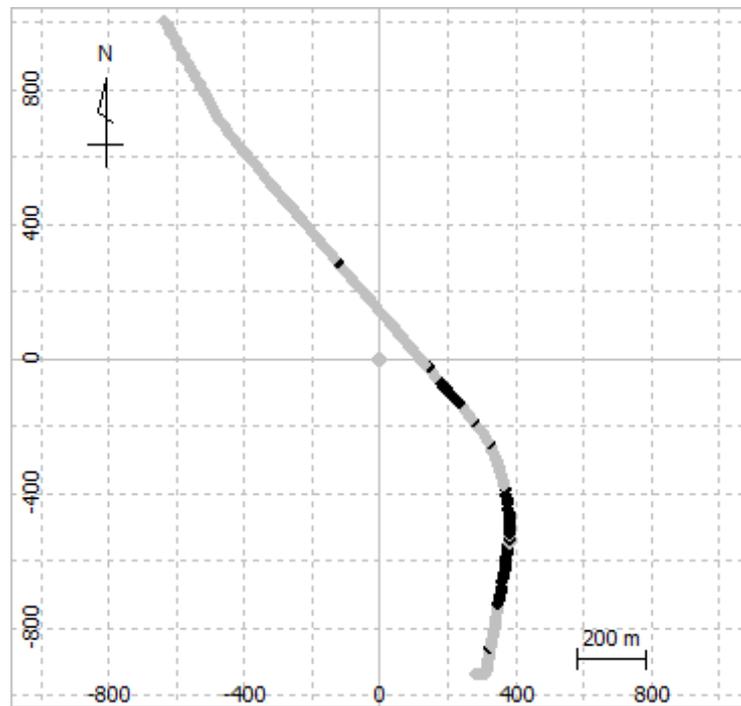
Method	Number of Ambiguity-Fixed Solutions	Rate (%)
Standard	43	18.94
GA-based	85	37.44
GA-SW	173	76.21

In addition, the GA-SW method was tested with window widths from 1-s up to 1-h. The results are presented in Table 4.5.

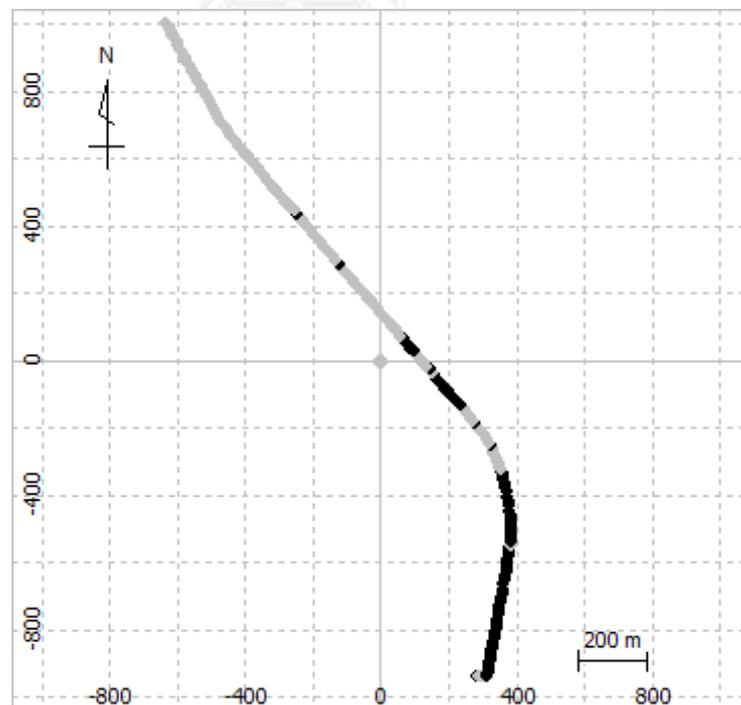
**Table 4.5** Comparison between window width and number of fixed-ambiguity solutions given by the GA-SW method

Window Width (s)	Number of Ambiguity-Fixed Solutions
1	129
5	184
10	193
20	194
30	196
40	198
50	196
3600	173

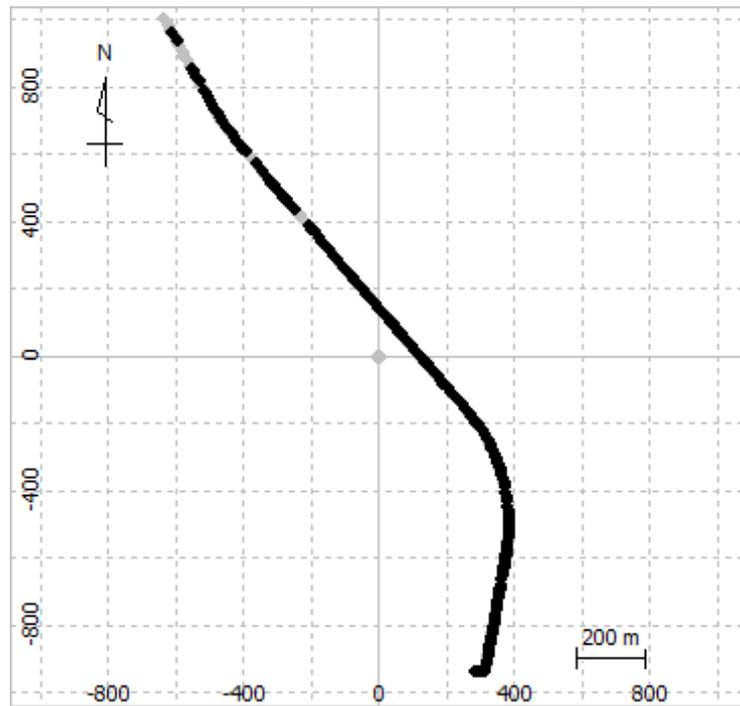
From the results in Table 4.5, it can be seen that a window width of 1 s produced the fewest ambiguity-fixed solutions. Bigger window widths resulted in more ambiguity-fixed solutions. However, at a width of 1 h, only 173 epochs were correctly determined, whereas it was expected that at least 196 epochs would produce ambiguity-fixed solutions. Thus, further investigation is needed to determine the optimal window width.



**Figure 4.8** Results given by the standard method



**Figure 4.9** Results given by the GA-based method



**Figure 4.10** Results given by the GA-SW method



## Chapter 5

### Conclusions and Recommendations

#### 5.1 Conclusions

As PPK provides better accuracy and stability for moving platforms than other positioning methods, it has a number of applications. However, PPK suffers from certain data processing problems, especially in practical cases where raw observation data are contaminated or biased. Normally, most of this contamination can be canceled out using double differencing, and successful ambiguity-fixed solutions can be computed easily. However, when the base station and the roving station are located in dissimilar environments, some bias will remain, resulting in the unsuccessful computation of ambiguity-fixed solutions. As a result, users are required to filter the noisy data using a process of trial and error. Therefore, this study investigated the use of a GA-based method and a GA-SW method to increase the number of ambiguity-fixed solutions.

##### 5.1.1 Enhancement of ambiguity-fixed solutions in GPS kinematic positioning using a GA

The standard GA was modified and developed to improve the success rate of ambiguity-fixed solutions. The resulting method returned the optimal satellite combination using a standard bit-string population with a simple penalty approach for the objective function. Based on the GA optimization, therefore it was called “GA-based” method. In GA-based method, RTKLib was chosen as a GPS PPK tool. Experimental data from 24 1-h sessions were collected from two dual-frequency GPS reference stations with known baseline. These data were used to demonstrate the performance of the GA technique. However, as the data were obtained from reference station and a flag in the raw observation data was changed to indicate kinematic data. All 24 sessions were then processed by both the standard PPK method and the GA-based method, and the number of ambiguity-fixed solutions in each session was compared. There were five sessions in which the number of ambiguity-fixed solutions was enhanced by the GA-based method. These five sessions were then classified as

being a high, medium, or low improvement level. Finally, one session from each group was selected, and the northing, easting, and height given by the GA-based method and the standard PPK method were compared with the known coordinates. However, it was not intention in this research to check the accuracy of the result, instead the comparison of positioning is to illustrate the level of difference between the positioning obtained from the ambiguity-fixed of GA-based method and standard PPK method.

The experimental results allow the following conclusions to be made.

- a) The use of all available satellites may not always lead to the optimal solution.
- b) The number of ambiguity-fixed solutions given by the GA-based method is greater than or equal to that given by standard PPK method.
- c) The GA-based method can improve the success rate of ambiguity-fixed solutions in GPS PPK compared with standard PPK.

In addition, the results also indicated the limitations of the GA-based method.

- d) The data used in this demonstration were from static observation.
- e) The GA-based method has only been tested with GPS.
- f) The objective function was only designed to optimize the number of ambiguity-fixed solutions.
- g) The obtained solutions were based on a time-span rather than an epoch.

### **5.1.2 Enhancement of ambiguity-fixed solutions in GPS kinematic positioning using a GA and sliding windows**

The time-span solution described in 5.1.1 might not be appropriate for practical kinematic positioning, where epoch-based solutions are needed. Therefore, the method in 5.1.1 was modified so that the satellite combination was optimized by the GA on an epoch by epoch basis. The sliding window technique was combined with the GA to optimize the satellite combination in each epoch and it was called “GA-SW method”. The GA-SW method was initially assigned a window width of 1-h and a sliding step of 1-s. It was then applied with practical kinematic data for 227 epochs as the base line varied from 0.8–2.5 km. Moreover, the GA-based method described in 5.1.1 and the standard PPK method were also tested with the same dataset. The results from all methods were compared, and the GA-SW method was found to give twice as many

ambiguity-fixed solutions as the GA-based and four-times the number of the standard PPK method. Experiments with window widths from 1–3600-s were also conducted. A window width of 40-s was found to produce the highest number of ambiguity-fixed solutions.

The experimental results indicated the following.

- a) The GA-SW method can be used with practical kinematic data.
- b) The GA-SW method produced twice as many ambiguity-fixed solutions as the GA and four times as many as the standard PPK method.
- c) The window width used in this study was not optimized, but experiments suggest that a width of 40-s gives the highest number of ambiguity-fixed solutions.

## 5.2 Recommendations

Considering the experimental results given by the methods developed in this research, the following recommendations for future work are now listed.

### 1) Modify the objective function used in the GA

The current objective function is based on the simple penalty approach with custom weight parameters. Thus, there are a number of possibilities for modifying the objective function:

- Optimize the weight parameters used in the objective function
- Change the objective function from a simple penalty approach to some other function such as a neural network

### 2) Reduce the optimization time of the GA

Both of the proposed methods require a long processing time. This is because the post-processing engine has to re-compute the same satellite combination several times. Thus, lookup tables could be used to store the computed results, thus avoiding time-consuming PPK re-computations.

**3) Analysis of excluded satellites**

It would be interesting to analyze why specific satellites were excluded from the computation. Are they really contaminated by some bias?

**4) Include the coordinate precision in the optimization**

To increase the precision of the results after optimization, the precision of the coordinates could be included in the optimization. However, this would require the objective function of the GA to be modified into a multi-objective function.

**5) Include other GNSS satellites in the optimization**

To increase the availability, accuracy, and reliability of positioning, other GNSS satellites besides those in the GPS could be included in the optimization.

**6) Adapt to different GPS post-processing tools**

As both GA-SW and GA-based method can optimize the satellite combination very effectively, it could be adapted to optimize the satellite combination in GPS post-processing tools such as Bernese or other scientific software.

## REFERENCES

- Bilich, A., M. Schmitz, B. Görres, P. Zeimetz, G. Mader and G. Wübbena (2012). Three-method absolute antenna calibration comparison. IGS Workshop 2012.
- Buckland, M. and M. Collins (2002). AI techniques for game programming, Premier press.
- Bureerak, A. (2010). A Study Of Vehicle Trajectory Determination Using GNSS. Master, Chulalongkorn University.
- Dach, R., U. Hugentobler, P. Fridez and M. Meindl (2007). "Bernese GPS software version 5.0." Astronomical Institute, University of Bern **640**: 114.
- Foster, J., M. Bevis and S. Businger (2005). "GPS Meteorology: Sliding-Window Analysis\*." Journal of atmospheric and oceanic technology **22**(6): 687-695.
- Hoffmann-Wellenhof, B., H. Lichtenegger and J. Collins (1994). "GPS: theory and practice." 3rd ed Springer-Verlag, New York.
- Holland, J. H. (1992). Adaptation in natural and artificial systems: an introductory analysis with applications to biology, control, and artificial intelligence, MIT press.
- Hudnut, K. W. and J. A. Behr (1998). "Continuous GPS monitoring of structural deformation at Pacoima Dam, California." Seismological Research Letters **69**(4): 299-308.
- Kijewski-Correa, T., M. Kochly and J. Stowell (2004). "On the emerging role of GPS in structural health monitoring." Proc., CTBUH 2004: 144-151.
- Leick, A. (2004). "GPS SATELLITE SURVEYING Third Edition." John Wiley & Sons.
- Leick, A., L. Rapoport and D. Tatarnikov (2015). GPS satellite surveying, John Wiley & Sons.
- Levitin, A. V. (2009). Introduction to Design & Analysis of Algorithms: For Anna University, 2/e, Pearson Education India.
- Liu, Z., W. Xiong, Z. Kang, H. Zhang and M. Qu (2010). GPS ambiguity resolution of single epoch data using genetic algorithms. Natural Computation (ICNC), 2010 Sixth International Conference on, IEEE.
- Mitchell, M. (1998). An introduction to genetic algorithms, MIT press.
- Mosavi, M. R. and M. Divband (2010). Calculation of Geometric Dilution of Precision Using Adaptive Filtering Technique Based on Evolutionary Algorithms: 4842-4845.
- Richert, T. (2005). The impact of future global navigation satellite systems on precise carrier phase positioning, Citeseer.
- Rizos, C. (1997). Principles and Practice of GPS Surveying. Monograph 17, School of Geomatic Engineering, The University of New South Wales, ISBN 0-85839--071-X, approx. 560pp.
- Roggero, M. (2003). "RTK positioning in cadastral GIS updating." INTERNATIONAL ARCHIVES OF PHOTOGRAMMETRY AND REMOTE SENSING XXXIV-6/W11: 78-82.
- Satirapod, C. (2002). Improving the GPS data processing algorithm for precise static relative positioning, The University of New South Wales Sydney.
- Takasu, T. (2011). "RTKLIB ver. 2.4. 1 Manual [http://www.rtklib.com/prog/manual\\_2.4.1.pdf](http://www.rtklib.com/prog/manual_2.4.1.pdf)." Last checked **11**: 2015.

- Takasu, T. (2013). Rtklib ver. 2.4. 2 Manual. <http://www.rtklib.com/prog/manual2.4.2.pdf>, Tech. Rep., April.
- Tang, W., D. Turley, G. Howell and L. Wilkinson (1994). A kinematic carrier phase tracking system for high precision trajectory determination. ION GPS-94.
- Teunissen, P., P. De Jonge and C. Tiberius (1995). The lambda method for fast GPS surveying. International Symposium "GPS Technology Applications" Bucharest, Romania, Citeseer.
- Whitley, D. (1994). "A genetic algorithm tutorial." Statistics and computing **4**(2): 65-85.
- Yeniay, O. (2005). "Penalty function methods for constrained optimization with genetic algorithms." Mathematical and Computational Applications **10**(1): 45-56.
- Yingni, D. and Z. Haifeng (2012). Research of experimentation data kinematics positioning of GPS. Remote Sensing, Environment and Transportation Engineering (RSETE), 2012 2nd International Conference on, IEEE.
- YoonSong, H. and H.-g. Kim (2012). "A real-time filtering method of positioning data with moving window mechanism." Computer Engineering and Intelligent Systems **3**(7): 20-31.
- Zhang, J. and Y. Chu (2011). The use of sliding time window in the real-time solution of ground-based GPS data. Electronics and Optoelectronics (ICEOE), 2011 International Conference on, IEEE.

**APPENDIX**



จุฬาลงกรณ์มหาวิทยาลัย  
**CHULALONGKORN UNIVERSITY**



```
[status,result] = dos(strcat('rmx2rtkp -o "c:\experiment2\session.pos" -k
"c:\experiment2\conf2run.con" "c:\experiment2\02 u299',txt,'.02o" "c:\experiment2\07
1299',txt,'.02o" "c:\experiment2\02 u2997.02n"));
```

*%currently, we got the result from the processing. Next we have to evaluate*  

```
pos = importfile(strcat('c:\experiment2\session.pos'));
```

```
if (pos == 0)
    display(79920);
    ret = 79920;
```

```
else
```

```
    fitness = calFitness(pos, 15);
    display(fitness);
    ret = fitness;
```

```
end
```

```
function output = importfile(fileToRead1)
%IMPORTFILE(FILETOREAD1)
% Imports data from the specified file
% FILETOREAD1: file to read
```

*% Auto-generated by MATLAB on 23-Mar-2011 14:19:31*

```
DELIMITER = ',';
HEADERLINES = 24;
```

*% Import the file*  

```
newData1 = importdata(fileToRead1, DELIMITER, HEADERLINES);
```

```
try
    output = newData1('data');
```

```
catch ME
    output = 0;
```

```
end
```

```
function fitness_val = calFitness(pos_values, sec)
```

```
fixed_cnt = 0;
float_cnt = 0;
std_cnt = 0;
```

```
num_fixed=0;
num_float=0;
num_std=0;
```

```

if (length(pos_values) == (3600/sec))

for i = 1:length(pos_values)
    if pos_values(i,4) == 1
        fixed_cnt = fixed_cnt +1;
    elseif pos_values(i,4) == 2
        float_cnt = float_cnt +1;
    else
        std_cnt = std_cnt+1;
    end
end

%return the number of fixed ambiguity
num_fixed = fixed_cnt;
num_float = float_cnt*100;
num_auto = std_cnt*1000;

num_sat = 0;

sdx = 0;
sdy = 0;
sdz = 0;

for i = 1:length(pos_values) %this will use only the last epoch
    if pos_values(i,4) == 1
        num_sat = 100 - (pos_values(i,5)*100/12);

        %add in s12
        sdx = sdx+(pos_values(i,6)*100);
        sdy = sdy+(pos_values(i,7)*100);
        sdz = sdz+(pos_values(i,8)*100);

    end

end

else
    num_fixed = 720;
    num_float = 72000;
    num_auto = 720000;
    num_sat = 100;

    sdx = 3600;
    sdy = 3600;
    sdz = 3600;

end

```

```

disp(['num_fixed: ', num2str(num_fixed), ', num_float:', num2str(num_float), ',
num_auto:', num2str(num_auto)]);
disp(['sdx: ', num2str(sdx), ', sdy:', num2str(sdy), ', sdz:', num2str(sdz)]);

fitness_val = num_fixed + num_float + num_auto;

function result = update_config_file(foldername, filename, PRN)

fid = fopen(strcat(foldername, filename), 'r+');

txt = fscanf(fid, '%c', inf);

fclose(fid);

txt2save = strrep(txt, 'pos1-exclsats = # (prn ...)', strcat('pos1-exclsats
=', PRN));

fid = fopen(strcat(foldername, 'conf2run.con'), 'w');
fprintf(fid, '%c', txt2save);
fclose(fid);

result = 0;

```

## 2. The source code used in the enhancement of the ambiguity-fixed rate in GPS kinematic positioning using a genetic algorithm and sliding windows

```

function result = RunSlidingWindow(
    %basefile, roverfile, navfile, parent_folder, output_folder,
    %start_datetime, end_datetime, interval_in_sec, window_width, number_of_iter)
    SlidingWindows('C:\Experiment Sliding Window\Base.10o',...
    'C:\Experiment Sliding Window\Rover.10o',...
    'C:\Experiment Sliding Window\Rover.10n',...
    'C:\Experiment Sliding Window\',...
    'C:\Experiment Sliding Window\KinEx_LastEpoch_15_065800to070000\',...
    '2010/07/17 06:58:00',...
    '2010/07/17 07:00:00',...
    1,...
    3600,...
    30)
return
function result = SlidingWindows(basefile, roverfile, navfile, parent_folder,
output_folder, start_datetime, end_datetime, interval_in_sec, window_width,
number_of_iter)

```

```

%this is to check that the file are exist or not
if(exist(basefile,'file')~=2)
    display('No Base file was found!!!');
    return;
end

if(exist(roverfile,'file')~=2)
    display('No Rover file was found!!!');
    return;
end

if(exist(navfile,'file')~=2)
    display('No Navigation file was found!!!');
    return;
end

%this is to check that the output_folder is exist or not
if(exist(output_folder,'dir')~=7)
    %this is the case that the directory is not exist so we will create
    %it
    display('Creating Output Directory...');
    [ret,msg] = mkdir(output_folder);
    if(msg ~= "")
        display('Error! Can not create the specific directory');

        return
    end
end

%#####This is for extract the last Epoch #####
% we will use the data back to 1 hrs to process
ts = datenum(start_datetime)-floor(window_width-1)*interval_in_sec/(24*3600);

    %this is to loop through the processing among the session
    i=1;
    while(ts < datenum(end_datetime)-floor(window_width-
1)*interval_in_sec/(24*3600))
        te = ts+(window_width-1)*interval_in_sec/(24*3600);
        %should display the timewindow that is processing...

display('#####');

display('#####');

```

```

display('#####');
display(strcat('Time Start : ', datestr(ts, 'yyyy/mm/dd HH:MM:SS'),' Time End : ',datestr(te,'yyyy/mm/dd HH:MM:SS')));
ga_result = RunGAWithInitial(ts,te,basefile,roverfile,navfile,parent_folder,
output_folder>window_width,
interval_in_sec,15,4,[1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1]);
pause(2);
saveas(gcf,strcat(output_folder,num2str(i)), 'fig');

ts= ts+interval_in_sec/(24*3600);
i=i+1;
end

```

```

function [x,fval,exitflag,output,population,score] =
RunGAWithInitial(Start_DateTime,End_DateTime,BaseFile,RoverFile,NavFile,Parent_Folder, Output_Folder,Window_Width,
Interval_In_Sec,Generations_Data,StallGenLimit_Data,InitialPopulation_Data)
% This is an auto generated M-file from Optimization Tool.

```

```

% Start with the default options
options = gaoptimset;
% Modify options setting
options = gaoptimset(options,'PopulationType', 'bitstring');
options = gaoptimset(options,'Generations', Generations_Data);
options = gaoptimset(options,'StallGenLimit', StallGenLimit_Data);
options = gaoptimset(options,'InitialPopulation', InitialPopulation_Data);
options = gaoptimset(options,'FitnessLimit', Window_Width);
options = gaoptimset(options,'FitnessScalingFcn', @fitscalingprop);
options = gaoptimset(options,'SelectionFcn', @selectionroulette);
options = gaoptimset(options,'CrossoverFcn', @crossovertwopoint);
options = gaoptimset(options,'CrossoverFraction', 0.9);
options = gaoptimset(options,'Display', 'off');
options = gaoptimset(options,'PlotFcns', { @gaplotbestf @gaplotbestindiv });
fitness_function =
{ @MyObjFunc,Start_DateTime,End_DateTime,BaseFile,RoverFile,NavFile,Parent_Folder, Output_Folder,Window_Width, Interval_In_Sec };
[x,fval,exitflag,output,population,score] = ...
ga(fitness_function,32,options);

```

```

function ret =
MyObjFunc(x,Start_DateTime,End_DateTime,BaseFile,RoverFile,NavFile,Parent_Folder, Output_Folder,Window_Width, Interval_In_Sec)
%this is to convert the bit string to the list of sat that want to exclude
%from the processing
sats = getsat(x);

```

```

display(sats);
% now we have to exit the configuration file to include the list of exclude
% sats
UpdateConfigFile(Parent_Folder,'setting.con',sats);

% Time Windows Procedure
str_ts = datestr(Start_DateTime,'yyyy/mm/dd HH:MM:SS');
str_te = datestr(End_DateTime,'yyyy/mm/dd HH:MM:SS');
ofil = strcat(Parent_Folder,'session.pos');
confil = strcat(Parent_Folder,'conf2run.con');

[status,result] = dos(['rnx2rtkp -ts ',str_ts,' -te ',str_te,' -o "',ofil,'" -k "',confil,'"
"',RoverFile,'" "',BaseFile,'" "',NavFile,'"']);

% currently, we got the result from the processing. Next we have to evaluate
pos = importfile(ofil);
[x,y] = size(pos);

if (x==1)
    [x,y] = size(pos.('data'));
    if (x <= Window_Width/2)
        % if (x ~= Window_Width)
        % display(3600);
        % ret = 3600;
        % 3600 + 1000 + 32
        display(num2str(Window_Width^5));
        ret = Window_Width^5;
    else
        fitness = CalFitnessSliding(pos, Window_Width, Interval_In_Sec,
Start_DateTime, Output_Folder);

        display(fitness);
        ret = fitness;
    end
else
    display(num2str(Window_Width^5));
    ret = Window_Width^5;
end
end

```

## VITA

Full Name: Panithan Srinuandee

Gender: Male

Date of Birth: March 26, 1979

Nationality: Thai

Religion: Buddhism

Education:

2011-Present: Ph.D. candidate, Survey Engineering, Faculty of Engineering, Chulalongkorn University

2003-2005: Master of Engineer in Remote Sensing and Geographic Information System, Asian Institute of Technology

1997-2000: Bachelor of Engineer in Electrical Engineering, Kasetsart University

Academic Contribution:

Srinuandee, P., Honda, K., Chemin, Y., Ines, A.V.M.. SWAP crop model parameter identification using SPOT Vegetation in Suphan Buri, Thailand. In Proceedings of the 25th Asian Conference of Remote Sensing, ChiangMai, Thailand, 22-26 November 2004.

Srinuandee, P., Satirapod, C., Ogaja, C., Lee, H. K. Optimization of satellite combination in kinematic positioning mode with the aid of genetic algorithm. *Artificial Satellites*, 47.2(2012): 35-46.

Srinuandee, P., and Satirapod, C., Use of genetic algorithm and sliding windows for optimising ambiguity fixing rate in GPS kinematic positioning mode. *Survey Review*, 47.340 (2015): 1-6.