

**ชื่อเรื่องภาษาไทย**

นาย ชื่อ นามสกุล รหัสประจำตัวนิสิต ภาควิชา ........................

นาย ชื่อ นามสกุล รหัสประจำตัวนิสิต ภาควิชา ........................

นาย ชื่อ นามสกุล รหัสประจำตัวนิสิต ภาควิชา ........................

(โครงร่าง) ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาวิชาโครงงานทางวิศวกรรม (ข้ามสาขาวิชา)

ตามหลักสูตรวิศวกรรมศาสตรบัณฑิต

คณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย

ปีการศึกษา 2560

****

**ชื่อเรื่องภาษาอังกฤษ**

Mr. Name Surname Student ID Department …………………

Mr. Name Surname Student ID Department …………………

Mr. Name Surname Student ID Department …………………

A Senior Project Submitted in Partial Fulfillment of the Requirements

For the Degree of Bachelor of Engineering Program

Faculty of Engineering Chulalongkorn University

Academic Year 2017

**หัวข้อโครงงาน** ………………………………………………………………………………………………………

**โดย**  นาย/นางสาว ……………………….. …………...............

นาย/นางสาว ……………………….. ……………………….

**อาจารย์ที่ปรึกษา** ผู้ช่วยศาสตราจารย์ ดร.……………… ………………………..

# บทคัดย่อ

…………………………………………………………………………………………………………………………………………………………..…………………………………………………………………………………………………………………………………………………………..…………………………………………………………………………………………………………………………………………………………..…………………………………………………………………………………………………………………………………………………………..…………………………………………………………………………………………………………………………………………………………..……………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………….

คำสำคัญ: ……………; …………………; ………………… (ใส่ key word ไม่เกิน 5 คำ)

**Title** ………………………………………………………………………………………………………

**Students** ……………………….. …………...............

……………………….. ……………………….

**Advisor**  Assistant Professor ……………… ……………………….., Ph.D.

# Abstract

…………………………………………………………………………………………………………………………………………………………..…………………………………………………………………………………………………………………………………………………………..…………………………………………………………………………………………………………………………………………………………..…………………………………………………………………………………………………………………………………………………………..…………………………………………………………………………………………………………………………………………………………..……………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………….

Key words: ……………; …………………; ………………… (ใส่ key word ไม่เกิน 5 คำ)

# กิตติกรรมประกาศ

…………………………………………………………………………………………………………………………………………………………..…………………………………………………………………………………………………………………………………………………………..…………………………………………………………………………………………………………………………………………………………..…………………………………………………………………………………………………………………………………………………………..…………………………………………………………………………………………………………………………………………………………..……………………………………………………………………………………………………………………………………………………..……..…………………………………………………………………………………………………………………………………………………………..

สารบัญ **(เป็นเพียงตัวอย่าง ให้นิสิตปรับเนื้อหาตามสมควร)**

หน้า

[บทคัดย่อ ก](#_Toc496013410)

[Abstract ข](#_Toc496013411)

[กิตติกรรมประกาศ ค](#_Toc496013412)

[สารบัญ ง](#_Toc496013413)

[สารบัญตาราง ช](#_Toc496013414)

[สารบัญรูป ฉ](#_Toc496013415)

[บทที่ 1 บทนำ 1](#_Toc496013416)

[1.1 ที่มาและความสำคัญ 1](#_Toc496013417)

[1.2 วัตถุประสงค์ 1](#_Toc496013418)

[1.3 ขอบเขตการวิจัย 1](#_Toc496013419)

[1.4 ประโยชน์ที่คาดว่าจะได้รับ 1](#_Toc496013420)

[บทที่ 2 วรรณกรรมที่เกี่ยวข้อง 2](#_Toc496013422)

[2.1 การจัดหน้ากระดาษ 2](#_Toc496013423)

[2.2 คำแนะนำการเขียน 2](#_Toc496013424)

[2.2.1 ขนาดตัวอักษร 2](#_Toc496013425)

[2.2.2 รูปภาพ 2](#_Toc496013426)

[2.2.3 ตาราง 3](#_Toc496013427)

[2.2.4 สมการ 3](#_Toc496013428)

[บทที่ 3 ระเบียบวิธีวิจัย 5](#_Toc496013429)

[3.1 .......................... 5](#_Toc496013430)

[3.2 .......................... 5](#_Toc496013431)

[3.2.1 .......................... 5](#_Toc496013432)

[บทที่ 4 ผลการดำเนินงานวิจัย 6](#_Toc496013433)

[4.1 ……………………………… 6](#_Toc496013434)

[บทที่ 5 สรุปผลการวิจัย อภิปรายผล และข้อเสนอแนะ 7](#_Toc496013435)

[เอกสารอ้างอิง 8](#_Toc496013436)

# สารบัญตาราง

|  |  |
| --- | --- |
| ตารางที่ 1.1 ……………………………………………… | 3 |
| ตารางที่ 2.1 ……………………………………………… | 9 |

# **สารบัญรูป**

|  |  |
| --- | --- |
| รูปที่ 2.1 ……………………………………………… | 4 |
| รูปที่ 2.2 ……………………………………………… | 5 |
| รูปที่ 2.3 ……………………………………………… | 7 |

# บทที่ 1 บทนำ

## ที่มาและความสำคัญ

Bangkok is full of businesses and companies, and even more companies today due to the booming of startups. Therefore, the demands of the messenger services are high. Moreover, Bangkok is also full of bike taxis (Win-motorcycle, motorcycle depots) which is good for us. However, the existing messenger services have some downsides. For example, the traditional messenger services generate all requests separately (1 job = 1 request). However, in the system we are trying to develop, the requests will be combined to minimize total travel distances (1 job can be many requests from many users. We will separate requests into groups to generate jobs for deliverymen to take, and when all jobs are generated, the deliverymen can choose the nearest jobs to do. This is the advantage of our system because the bikes (deliverymen) can do many requests in one job with fewer distances than existing systems, assuming that the bikes have to return to their depots. For more understanding, please see the pictures below.

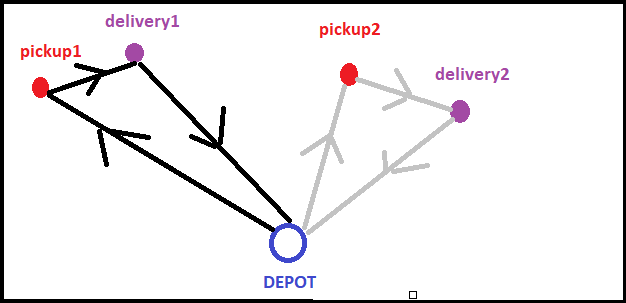


Fig. 1.1 – The existing system. 2 job 2 requests; Red nodes are pickup points; purple nodes are delivery points.

Assume that bikes have to travel to pickup points (red nodes), then to delivery points (blue nodes), then to their depots, which is somewhere near the pickup points (the lowest distances case is that the depots are at pickup points).

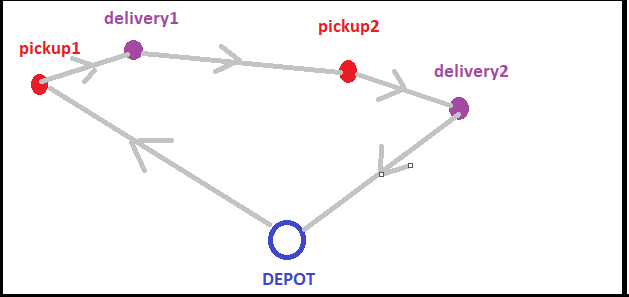


Fig. 1.2 – The new system. 1 job, 2 requests.

The requests are combined into 1 job, the total travel distances are decreased.

## วัตถุประสงค์

## The objective of this project is to find and analyze an algorithm for the messenger service problem. The messenger service is a service which users have requests to send documents, and service providers (deliverymen) pick the documents from users at the pickup points and deliver them to the delivery points. This problem can be classified as a pickup and delivery problem. Given, the requests from users (the pickup points, the delivery points, time windows, etc.), we want to generate jobs (the tours of vehicles) for the service providers (deliverymen) with the lowest costs (travel distances) and still satisfy the constraints. Our objective is to find a way to generate the best jobs.

## ขอบเขตการวิจัย

The scope of this project includes studying research papers, researching and developing our own algorithm for the problem, simulating and running test instances. Also, developing preliminary business model for messenger service including cost and revenue management.

About the problem, we are trying to find the best ways to generate jobs for service providers (deliverymen); each job consists of request(s) from user(s) who want to send documents, given coordinates of places ,traveling distances between places, time windows that users appoint to receive the documents, load capacities of vehicles and load demands of objects to be delivered. We want to generate the jobs with the lowest total travel distances . We also assume that the amount of bikes in Bangkok are high enough so that we do not have to think about constraints of the number of vehicles, and there will always be a bike (deliveryman) who accepts any jobs we created.

After researching, we have found that the messenger business can be categorized into 2 types. First, the business that own the asset. Second, the business that do not. Our group are focusing on the second one as we intend to provide the complete route for any driver (motorcycle) to participate in the business.

## ประโยชน์ที่คาดว่าจะได้รับ

- To develop the algorithms for solving the practical problem.

- To develop the business model for the problem.

- To learn more about algorithms.

- To work as a team.

# บทที่ 2 วรรณกรรรมที่เกี่ยวข้อง

~~นิสิตสามารถเริ่มต้นการเขียนบทความโดยการแทนที่เนื้อหาในเอกสารต้นแบบฉบับนี้ โดยก่อนนิสิตจะเขียนปริญญานิพนธ์ขอให้นิสิตศึกษา “คู่มือการจัดทำปริญญานิพนธ์” ที่ได้จัดทำขึ้นอย่างละเอียด นอกเนื้อจากรายละเอียดใน “คู่มือการจัดทำปริญญานิพนธ์” แล้วรายละเอียดเพิ่มเติมในเรื่องรูปแบบการจัดทำปริญญานิพนธ์ มีดังต่อไปนี้~~

## ~~2.1 การจัดหน้ากระดาษ~~

~~ขนาดของบทความจะอยู่ในพื้นที่ของกระดาษ A4 โดยเว้นระยะด้านบนและด้านซ้ายเป็นระยะ 3.81 ซม. และ 2.54 ซม. สำหรับด้านบนและด้านขวา การลำดับหัวข้อในส่วนของเนื้อเรื่อง ให้ใส่เลขกำกับ และหากมีการแบ่งหัวข้อย่อย ให้ใช้เลขระบบทศนิยมกำกับหัวข้อย่อย เช่น 2.1, 2.1.1 เป็นต้น~~

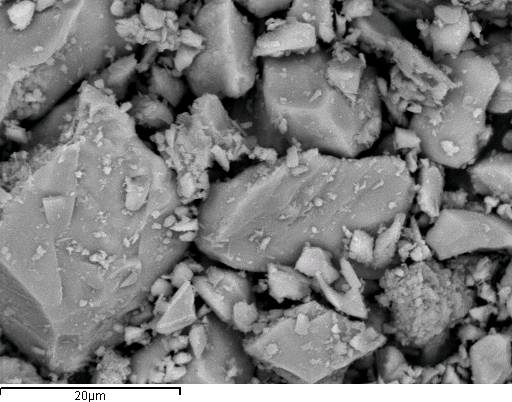
## ~~2.2 คำแนะนำการเขียน~~

### ~~2.2.1 ขนาดตัวอักษร~~

~~ตัวอักษรที่ใช้คือ “TH Sarabun New” รายละเอียดตัวอักษรแสดงในตารางที่ 2.1 เนื้อเรื่องในแต่ละบรรทัดให้จัดเรียงชิดซ้ายและขวาอย่างสวยงามโดยตั้งค่า Alignment แบบ Thai Distributed (“Justify”)~~

### ~~2.2.2 รูปภาพ~~

~~รูปภาพจะต้องวางไว้ตำแหน่งหน้ากระดาษ โดยรูปภาพทุกรูปจะต้องมีหมายเลขแสดงลำดับและคำบรรยายได้ภาพ ตัวอย่างการจัดวางรูปดังแสดงในรูปที่ 2.1 คำบรรยายใต้ภาพ ห้ามใช้คำว่า “แสดง” เช่น ห้ามเขียนว่า” รูปที่ 1 แสดงความสัมพันธ์...” ที่ถูกต้องควรเป็น “รูปที่ 1 ความสัมพันธ์ระหว่าง...”~~

~~~~

**~~รูปที่ 2.1~~** ~~ภาพถ่ายขยายกำลังสูงของปูนซีเมนต์ปอร์ตแลนด์~~

## ~~2.2.3 ตาราง~~

~~ตารางทุกตารางจะต้องมีหมายเลขและคำบรรยายกำกับเหนือตารางดังแสดงใน ตารางที่ 2.1 ในคำบรรยายเหนือตารางห้ามใช้คำว่า “แสดง” เช่นเดียวกับกรณีรูปภาพ ตารางควรจะอยู่หน้าเดียวทั้งตาราง ไม่ควรคร่อมระหว่างหน้า~~

**~~ตารางที่ 2.1~~** ~~สรุปรายละเอียดรูปแบบตัวอักษร~~

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ~~หัวข้อ~~ | ~~ลักษณะ~~ | ~~ตัวอักษร~~ | ~~รูปแบบ~~ | ~~ขนาด (จุด)~~ |
| ~~ชื่อบท (1,2,..)~~ | ~~Heading1~~ | ~~TH Sarabun New~~ | ~~หนา~~ | ~~18~~ |
| ~~หัวเรื่อง 2 (1.1,1.2,..)~~ | ~~Heading2~~ | ~~TH Sarabun New~~ | ~~หนา~~ | ~~16~~ |
| ~~หัวเรื่อง 3 (1.1.1, …)~~ | ~~Heading3~~ | ~~TH Sarabun New~~ | ~~หนา~~ | ~~16~~ |
| ~~เนื้อหา~~ | ~~Normal~~ | ~~TH Sarabun New~~ | ~~ธรรมดา~~ | ~~16~~ |
| ~~ตัวแปรในสมการ \*\*~~ | ~~N/A~~ | ~~Times New Roman~~ | ~~เอียง~~ | ~~12~~ |
| ~~เอกสารอ้างอิง~~ | ~~Reference\_new~~ | ~~TH Sarabun New~~ | ~~ธรรมดา~~ | ~~16~~ |

~~\*\* จัดทำโดยการใช้ MathType/ Ms Equation Object.~~

### ~~2.2.4 สมการ~~

~~สมการที่ใช้ในปริญญานิพนธ์ควรจะเป็นการสร้างจากโปรแกรม MathType หรือเป็นวัตถุของ Microsoft Equation มีขนาด 12 จุด และเป็นตัวอักษร“Times New Roman” ขนาด 12 จุด สมการทุกสมการจะต้องมีหมายเลขกำกับอยู่ภายในวงเล็บ และเรียงลำดับที่ถูกต้อง ตำแหน่งของหมายเลขสมการ (ใช้ตัวอักษร TH Sarabun New ธรรมดาขนาด 16 จุด) จะต้องอยู่ชิดขอบด้านขวาของหน้า ตำแหน่งของสมการให้จัดตามความสวยงาม ดังแสดงในสมการที่ (2.1)~~

~~ (2.1)~~

~~โดยการอธิบายตัวแปรที่ระบุในสมการ ให้ใช้ตัวอักษร Times New Roman ตัวอักษรเอียง ขนาด 12 จุด ตัวอย่างเช่น~~ *~~c~~* ~~หมายถึงจำนวนช่างไฟฟ้า,~~ *~~d~~* ~~คือ จำนวนช่างฝ้า,~~ *~~e~~* ~~คือค่าคงที่เท่ากับ 2 และ~~ *~~f~~* ~~คือผลลัพธ์ที่ได้~~

# บทที่ 3 ระเบียบวิธีวิจัย

## 3.1 Algorithm Part

**3.1.1 – Using Genetic Algorithm (GA) to solve the single depot Pickup and Delivery problems with time windows (PDPTW)**

We chose Genetic Algorithm (GA) to solve this kind of problems. The reason we chose GA is that we want to solve problems in a limit amount of time, and in many papers (read the project proposal for more information) GA can give good solutions with not so much computational times.

We found an interesting research paper that we can use[1]. In the paper he used an algorithm called Grouping Genetic Algorithm to solve the PDPTW. The algorithm can be described below.

## Our group had reached the preliminary business model. Next, we aim to elaborate the realistic model.

## GGA for PDPTW (adapted from [1])

## The solutions of the problems are encoded into individuals (chromosomes) represented by an array of genes. A gene represents information of a vehicle’s route, as can bee seen below.

## Populations = An array of chromosomes

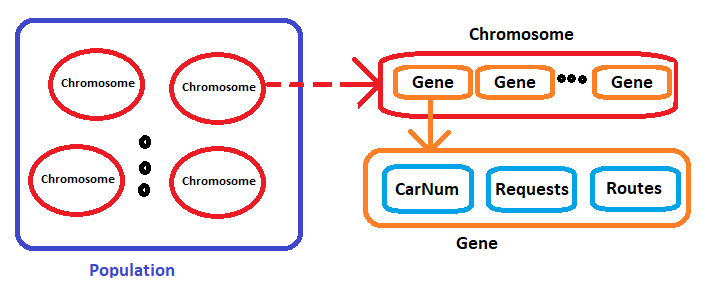
## Chromosome = [Gene1, Gene2, Gene3, ... ,Genev] ; v= the number of vehicle used (the number of routes)

## Gene = [num, {Requests} , [Route] ]

## {Requests} = A set of indices of requests.

## request = a pair of an index pickup node and the corresponding index delivery node

## [Route] = An array of a route (permutation of the indices of nodes)



**Procedures:**

First, we create populations of feasible chromosomes of size n. Each chromosome has feasible routes for all nodes. After that, we apply GA operations to the populations as can be seen from the pseudocode below.

Generation = 1

While not(termination criteria is met):

evaluate\_fitness(populations)

sort\_by\_fitness(populations)

remove 2 chromosomes with the worst fitness

select the best 2 chromosomes, name them elite1, elite2. These 2 chromosomes will go to the next generation unchanged

randomly select a pair of chromosomes, name them parent1 and parent2, remove them from the poplutions.

child1,child2 <= crossOver(parent1,parent2,prob =1.0)

child1,child2<=mutate(child1,prob=0.5),mutate(child2,prob=0.5)

put child1,child2 back into the populations

Generation <= Generation + 1

**GA operations**

Crossover:



(Image from [1])

1.First, we have the chromosomes parent1 and parent2. We randomly specify a crossing section on parent2.

2. Remove the duplicate vehicles on parent1 that also have on part from parent2. Insert the section into parent1, now parent1 became child1.

3.Remove the duplicate requests that already have on child1 and also have on the part from parent2 before the insertion.

4.Insert the remaining requests (see descriptions below) to ensure that all of the requests are served.

5. Repeat the same processes on child2.

**Mutate**:

Randomly select one gene (vehicle) in the chromosome. Remove all requests and their corresponding route. Insert the requests again to the chromosome.

**Inserting requests**:

First, we shuffle the requests to insert so that requests are processed in a random order. Then we insert every request into the chromosome.

To insert a request (p,d) into a tour [a,b,c,d,e,...]; (a,b,c,.. are nodes), we check if the tour is empty, if it is, the request can be inserted without any further calculation.

If the tour is not empty, we insert p into the tour and check whether the new route violate constraints (time windows, load capacities) or not. If the new route is invalid, we just move on to the new position to insert p, else do the similar processes to d. The d has to be inserted after p. If the final tour is invalid, we discard it. Else, we calculate the new distance and subtract them by the old tour distance and call it a “cost”. We do this to every genes (vehicles) in the chromosome and choose the one with minimum cost. If we cannot find a point to insert the request that produce a feasible solution at all, we have to add a new vehicle (gene) into the chromosome to handle the request.

**Test Results**

We adapted and implemented the algorithm and tested it on some of the Li and Lim’s test instances to how it performs. We compared our results to the best known solutions (world records). The results are shown below.

**Parameters for GA**

Generations : 3000, break if qualities of the best solutions in the populations are the same for 500 generations

Crossover rate = 1.0

Mutation rate = 0.5

Elitism (select the best 2 chromosomes and copy them unchanged to the next generation)



**(\*) Results that equal to the the best known solutions**

(\*\*) Results that have less distances than best known solutions. Best known solutions try to minimize number of vehicles first, then try to minimize distances, that’s why my solution has less distances.

The results above are just one time results. For better approximation of algorithm efficiency the results should be recalculated many times and use the averages of them later. As can be seen above, some of the results are equal to the best known solutions. By the way, the qualities of solutions maybe increase if we do more generations of GA’s process, but it will take more time, of course.

Now, we had an algorithm to solve the single-depot problems which worked pretty well. We decided to move on to the multi-depot problems which are more realistic.

**3.1.2 – Finding the best way to assign requests to depots to solve the multi-depot pickup and delivery problems**

The Li and Lim’s test instances are single-depot instances. In order to make them multi-depot, we inserted another 4 depots to each of the instances. We used Li and Lim’s 100 instances which have approximately 100 nodes, all of them have the depot at the coordinate (40,50). We insert the other 4 depots at coordinates (25,25),(25,75),(75,25), and (75,75) respectively. For more understanding, see the pictures below.

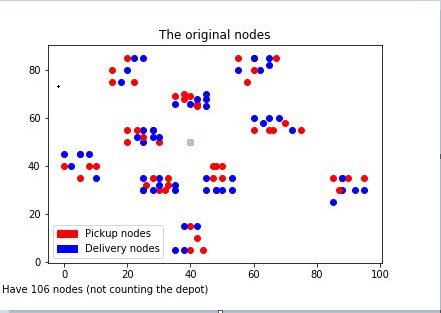


Fig. Original instances (the depot is at the gray dot)

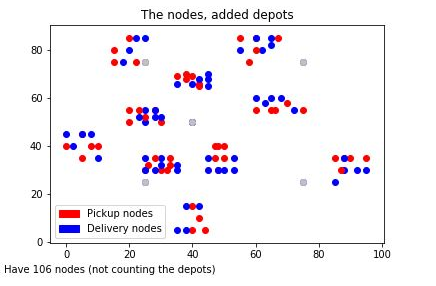


Fig. Added depots instance (the depots are at the gray dots)

Now, we want to find the best we to assign requests (pickup,delivery) to the depots so that after they are assigned we can treat them as 5 single-depot PDPTWs and we can use the GA above to solve them.

We came up with 4 different ways for assigning requests to the depots.

Given the requests consist of the request (pi,di) where pi is the pickup node, di is the delivery node of the request.

3.1.2.1 – Simple assign

This is the simpliest way of assigning depots. Calculate the distances from all depots to pi and from di to all depots, sum them up. Assign the request to the depot that give the least distance.

3.1.2.2 – Nearest Routable

First, calcualte the simple assigned depots of every requests as described in 3.1.2.1 and store them in a table. Then, sort the requests by the latest time to be vistsed of the pickup node. This table is the ‘default’ depots for the requests.

After that, loop through the sorted requests (pi,di),set the minimum distance to the distance of pi to the depot +the depot todi ,and check for every other requests (pj,dj) if they can route to each other or not. If yes, calculate the distance (pi ->pj) + (di ->dj) and check if it is less than the minimum, if yes update the depot.

3.1.2.3 - Nearest Routable 3-vote

Do things similar to 3.1.2.2, but instead of finding just one nearest candidate depot, this time find the nearest 3 depots from 3 requests and let them vote. If they cannot vote, use the depot from the request which give the nearest distance.

3.1.2.4 - Nearest Routable vote with condition

Check first if the nearest depot from pickup node and delivery node of the request are the same. If they are, assign the request to that depot. Else, mark the request as “problematic”. After that do things similar to 3.1.2.3 to the problematic requests.

**3.1.3 – Testing the algorithm on the real locations**

After we tested the algorithm on the Li and Lim’s instances, we moved on to the more realistic testings. We created our own 12 test instances from real places in Bangkok using GoogleMap. We call them “Goo” instances with number 1-12.

About the Goo instances, each instances has 80 places to visit, so there is 40 requests per instances. All distances between places were calculated using GoogleMap API. All vehicles are the same and have load capacities of 90. Load demands of requests are randomly generated from {10,20,30,40}. Time windows are randomly generated, and latest time that can be visited of the pickup nodes are at least 1 hour earlier to latest time of the delivery nodes to ensure that the instances can be solved. Service times of places are randomly generated from 1 second to 15 minutes.

About the depots, we use 5 real motorcycle depots in Bangkok which are BTS ChatuChak, Sanam-Pao , CentralRama9 MRT, Soi-SongPrha, and BTS-Siam depots. The distances from depot to places and from places to depots are also calculated using GoogleMap API.

## 3.2 Buisiness Model

### 3.2.1 ..........................

..................................................................................................................................................................................................................................................................................................................................................................................................................................

# บทที่ 4 ผลการดำเนินงานวิจัย

..................................................................................................................................................................................................................................................................................................................................................................................................................................

## 4.1 ………………………………

..................................................................................................................................................................................................................................................................................................................................................................................................................................

# บทที่ 5 สรุปผลการวิจัย อภิปรายผล และข้อเสนอแนะ

..................................................................................................................................................................................................................................................................................................................................................................................................................................

# เอกสารอ้างอิง

(นิสิตต้องอ้างอิงให้ถูกต้องตามหลักการ สามารถศึกษาวิธีการเขียนอ้างอิงได้จากเอกสาร “คู่มือการจัดทำปริญญานิพนธ์”)