ใบรับรองวิทยานิพนธ์

บัณฑิตวิทยาลัย มหาวิทยาลัยเกษตรศาสตร์

วิศวกรรมศาสตรดุษฎีบัณฑิต (วิศวกรรมอุตสาหการ)

ปริญญา

วิศวกรรมอุตสาหการ | วิศวกรรมอุตสาหการ

สาขา | **ภาควิชา**

เรื่อง ขั้นตอนวิธีการสำหรับการหาผลเฉลยเชิงทนทานของปัญหาการจัดเส้นทางเดินรถขนส่งแบบมีกรอบเวลาและเวลาเดินทางไม่แน่นอน

Algorithm for Determining Robust Solutions of Vehicle Routing Problem with Time Windows and Uncertain Travel Times

นามผู้วิจัย นางธรินี มณีศรี

ได้พิจารณาเห็นชอบโดย

อาจารย์ที่ปรึกษาวิทยานิพนธ์หลัก ....

(รองศาสตราจารย์อนันต์ มุ่งวัฒนา, Ph.D.)

หัวหน้าภาควิชา ....

(รองศาสตราจารย์อนันต์ มุ่งวัฒนา, Ph.D.)

บัณฑิตวิทยาลัย มหาวิทยาลัยเกษตรศาสตร์รับรองแล้ว

....

(รองศาสตราจารย์กัญจนา ธีระกุล, D.Agr.)

คณบดีบัณฑิตวิทยาลัย

วันที่ .... **เดือน** .... **พ.ศ.** ....

วิทยานิพนธ์

เรื่อง

ขั้นตอนวิธีการสำหรับการหาผลเฉลยเชิงทนทานของปัญหาการจัดเส้นทางเดินรถขนส่งแบบมีกรอบเวลาและเวลาเดินทางไม่แน่นอน

Algorithm for Determining Robust Solutions of Vehicle Routing Problem with Time Windows and Uncertain Travel Times

โดย

นางธรินี มณีศรี

เสนอ

บัณฑิตวิทยาลัย มหาวิทยาลัยเกษตรศาสตร์
เพื่อความสมบูรณ์แห่งปริญญาวิศวกรรมศาสตรดุษฎีบัณฑิต
พ.ศ. 2552

ธรินี มณีศรี 2552: ขั้นตอนวิธีการสำหรับการหาผลเฉลยเชิงทนทานของปัญหาการจัดเส้นทางเดินรถขนส่งแบบมีกรอบเวลาและเวลาเดินทางไม่แน่นอน ปริญญาวิศวกรรมศาสตรดุษฎีบัณฑิต (วิศวกรรมอุตสาหการ) สาขาวิศวกรรมอุตสาหการ ภาควิชาวิศวกรรมอุตสาหการ อาจารย์ที่ปรึกษาวิทยานิพนธ์หลัก: รองศาสตราจารย์อนันต์ มุ่งวัฒนา, Ph.D. 172 หน้า

งานวิจัยนี้ได้พัฒนาขั้นตอนวิธีการในการหาผลเฉลยเชิงทนทานด้วยวิธีฮิวริสติกส์แบบผสมผสาน โดยมีวัตถุประสงค์หลัก เพื่อหาผลเฉลยเชิงทนทานสำหรับปัญหาการจัดเส้นทางเดินรถขนส่งแบบมีกรอบเวลาและเวลาเดินทางไม่แน่นอน ซึ่งกำหนดอยู่ในรูปของช่วงเวลา ความซับซ้อนของปัญหาการจัดเส้นทางเดินรถขนส่งแบบมีกรอบเวลานี้ เกิดจากความไม่แน่นอนของเวลาเดินทางที่ขึ้นอยู่กับสภาพการจราจร เช่น การเกิดอุบัติเหตุ สภาพการจราจรติดขัด หรือสภาวะอากาศ ฉะนั้นส่วนสำคัญ คือ การสร้างรูปแบบของความไม่แน่นอนและขั้นตอนวิธีการในการแก้ปัญหา งานวิจัยนี้ได้นำเสนอวิธีการเข้าสู่ผลเฉลยแบบการสร้างสถานการณ์ และขั้นตอนวิธีการแบบผสมผสานในการค้นหาผลเฉลยเชิงทนทาน

ขั้นตอนวิธีการถูกสร้างด้วยวิธีฮิวริสติกส์การแทรกไปข้างหน้าแบบดัดแปลง ในการสร้างผลเฉลยเริ่มต้น จากนั้นทำการปรับปรุงผลเฉลยด้วยวิธีการค้นหาเฉพาะที่แบบสลับสับเปลี่ยนปมด้วยค่าแลมด้า และการค้นหาแบบทาบู การทดลองใช้ปัญหาเทียบเคียงของ Solomon ในการหาผลเฉลยเชิงทนทาน และเลือกใช้เกณฑ์การตัดสินใจเชิงทนทาน ได้แก่ สภาพทนทานสัมบูรณ์ ความแตกต่างทนทาน และสภาพทนทานแบบสัมพัทธ์ มีการวัดสมรรถนะของวิธีการเข้าสู่ผลเฉลยแบบการสร้างสถานการณ์ โดยเปรียบเทียบกับวิธีการแบบกรณีแย่ที่สุด และแบบค่าคาดหวัง โดยกำหนดปัญหาการจัดเส้นทางเดินรถขนส่งแบบมีกรอบเวลา และมีเปอร์เซ็นต์ความไม่แน่นอน, β = 0.2, 0.4, 0.6 และ 0.8 ขนาดของสถานการณ์, $|S|$ = 10, 20, 40 และ 80 ตามลำดับ สุดท้ายเป็นการวัดสมรรถนะของขั้นตอนวิธีการ โดยเปรียบเทียบคุณภาพของผลเฉลย และเวลาในการประมวลผล ระหว่างขั้นตอนวิธีการที่พัฒนาขึ้นกับขั้นตอนวิธีการเชิงพันธุกรรม โดยการทดลองใช้ปัญหาเทียบเคียงที่มีจำนวนลูกค้าขนาด 25, 50 และ 100 เมือง

ผลที่ได้จากการทดลองแสดงให้เห็นว่า ขั้นตอนวิธีการแบบผสมผสานสามารถหาผลเฉลยที่มีความทนทานต่อสภาพการจราจร หรือปัจจัยอื่นๆ ที่ไม่แน่นอนได้ โดยเกณฑ์การตัดสินใจที่เหมาะสมได้แก่ ความแตกต่างทนทาน และสภาพทนทานแบบสัมพัทธ์ ผลการวัดสมรรถนะแสดงให้เห็นว่า วิธีการเข้าสู่ผลเฉลยแบบการสร้างสถานการณ์มีสมรรถนะที่ดีกว่าวิธีการอื่นๆ เมื่อค่าความไม่แน่นอนสูงขึ้นมากกว่า 60 % ขึ้นไป ส่วนขั้นตอนวิธีการที่พัฒนาขึ้นให้คุณภาพผลเฉลยที่ดีกว่าวิธีการเชิงพันธุกรรมสำหรับปัญหาเทียบเคียงทุกประเภท แต่สมรรถนะด้านการใช้เวลาในการประมวลผลด้วยคอมพิวเตอร์ด้อยกว่า เมื่อจำนวนลูกค้าเพิ่มขึ้นมากกว่า 50 เมืองขึ้นไป

____________________ ____________________ ___ / ___ / ____

ลายมือชื่อนิสิต ลายมือชื่อประธานกรรมการ

Tharinee Manisri 2009: Algorithm for Determining Robust Solutions of Vehicle Routing Problem with Time Windows and Uncertain Travel Times. Doctor of Engineering (Industrial Engineering), Major Field: Industrial Engineering, Department of Industrial Engineering. Thesis Advisor: Associate Professor Anan Mungwattana, Ph.D. 172 pages.

This research develops an algorithm to determine robust solutions using a hybrid-heuristic technique. The main objective was finding the robust solutions for a vehicle routing problem with time windows (VRPTW) and uncertain travel times. The travel times are explained by interval with finite set of data. This VRPTW model becomes complex when the travel times are uncertain due to traffic conditions such as accidents, traffic jams, or weather conditions. Accordingly, the critical elements of this research include the model development which accounts for uncertainty and the algorithm development for solving such problems. This research presents a scenario-based approach and a hybrid algorithm to find the robust solutions.

The algorithm is constructed based on a modified push-forward insertion heuristic (MPFIH) was used to find the initial solution which was improved by using a lambda- interchange local search descent method (λ-LSD) and tabu search (TS). The experiment used Solomon's benchmarking problems to find the robust solutions. Robust criteria were the absolute robustness, the robust deviation and the relative robustness. The performance of the scenario-based approach is comparing with the worst-case performance approach and the expected value approach. The experiments set the VRPTW with percentages of uncertainty, β , equal to 0.2, 0.4, 0.6 and 0.8. The number of scenarios, $|S|$ were 10, 20, 40 and 80, respectively. Finally, the algorithm performance was measured in the quality of solution and computer run time by comparing with the genetic algorithm. The experiment used the Solomon's VRPTW benchmarking problems with 25, 50 and 100 nodes of customers.

The experiments clearly show that the hybrid algorithm can find the robust solutions that are not affected substantially by the traffic or other uncertain conditions. The appropriate robust criteria for this problem are the robust deviation and the relative robustness. The performance measure results show the scenario-based approach has higher performance than the benchmarking approaches when the percentage of uncertainty increases over 60 %. The developed algorithm has higher performance than the genetic algorithm in the quality of solutions but lower performance in computer run time when the number of customers increases over 50 nodes.

_________________________ _________________________ ____ / ____ / ______

Student's signature Thesis Advisor's signature

กิตติกรรมประกาศ

ผู้วิจัยขอกราบขอบพระคุณ รศ.ดร.อนันต์ มุ่งวัฒนา อาจารย์ที่ปรึกษาวิทยานิพนธ์หลัก ที่ให้คำปรึกษาในการเรียน การค้นคว้าวิจัย ตลอดจนการตรวจแก้ไขวิทยานิพนธ์จนกระทั่งเสร็จสมบูรณ์

ขอกราบขอบพระคุณอาจารย์ภาควิชาวิศวกรรมอุตสาหการทุกท่าน ที่ได้อบรมสั่งสอนและมอบความรู้อันเป็นประโยชน์อย่างยิ่งในการนำไปใช้ประโยชน์ต่อไป ขอขอบคุณ Professor Dr. Gerrit K. Janssens ที่ให้คำปรึกษาในการค้นคว้าวิจัย ขอขอบคุณ ผศ.นุชรินทร์ ทิพยวรรณากร และ อ.วฐา มินเสน ที่ให้คำปรึกษาและแนะนำการใช้โปรแกรม MATLAB เป็นอย่างดี

ด้วยความดีหรือประโยชน์อันใดเนื่องจากวิทยานิพนธ์เล่มนี้ ขอมอบแด่คุณพ่อ คุณแม่ ที่ได้อบรมและให้กำลังใจผู้วิจัยมาตลอดในทุกเรื่อง

ธรินี มณีศรี

มีนาคม 2552

สารบัญ

	หน้า
สารบัญ	(1)
สารบัญตาราง	(2)
สารบัญภาพ	(4)
คำอธิบายสัญลักษณ์และคำย่อ	(8)
คำนำ	1
วัตถุประสงค์	3
การตรวจเอกสาร	4
อุปกรณ์และวิธีการ	25
อุปกรณ์	25
วิธีการ	25
ผลและวิจารณ์	71
ผล	71
วิจารณ์	110
สรุปและข้อเสนอแนะ	115
สรุป	115
ข้อเสนอแนะ	116
เอกสารและสิ่งอ้างอิง	118
ภาคผนวก	123
ภาคผนวก ก ผลเฉลยเชิงทนทาน	124
ภาคผนวก ข ผลการเปรียบเทียบขั้นตอนวิธีการ	149
ประวัติการศึกษา และการทำงาน	172

สารบัญตาราง

ตารางที่		หน้า
1	รูปแบบปัญหา VRP	4
2	การหาผลเฉลยสภาพทนทานสัมบูรณ์ (Absolute Robustness Solution)	31
3	การหาผลเฉลยความแตกต่างทนทาน (Robust Deviation Solution)	32
4	การหาผลเฉลยสภาพทนทานสัมพัทธ์ (Relative Robustness Solution)	33
5	ลำดับการทำงานของตัวดำเนินการเคลื่อนย้าย	50
6	ผลการสลับสับเปลี่ยนในรอบที่ 1	55
7	ผลการสลับสับเปลี่ยนในรอบที่ 2	57
8	ผลการสลับสับเปลี่ยนในรอบที่ 3	58
9	ประเภทของปัญหา VRPTW โดย Solomon	61
10	ตัวอย่างปัญหา VRPTW จำนวน 25 เมือง (ความสามารถบรรทุกสินค้า = 100)	62
11	การออกแบบการทดลองในการหาผลเฉลยเชิงทนทาน	63
12	การกำหนดรหัสเครื่องคอมพิวเตอร์	66
13	แผนการประมวลผลปัญหาเทียบเคียงของ Solomon	67
14	ตัวอย่างการแสดงผลเฉลยเชิงทนทานสำหรับปัญหา R101, $\|S\|$=10	
	และ β=0.2	74
15	ผลการเปรียบเทียบวิธีการเข้าสู่ผลเฉลย	88
16	ผลการเปรียบเทียบคุณภาพของผลเฉลย (จำนวนรถขนส่งต่ำสุด)	87
17	ผลการเปรียบเทียบคุณภาพของผลเฉลย (เวลาเดินทางโดยรวมต่ำสุด)	88
18	ตัวอย่างปัญหา RVRPTW	105
19	ผลเฉลยเวลาเดินทางต่ำสุดสำหรับตัวอย่างปัญหา, $\|S\|$=4	106
20	การหาผลเฉลยสภาพทนทานสัมบูรณ์, Z_A คำนวณจากสมการที่ (2)	106
21	การหาผลเฉลยสภาพทนทานสัมบูรณ์, Z_D คำนวณจากสมการที่ (3)	107
22	การหาผลเฉลยสภาพทนทานสัมบูรณ์, Z_R คำนวณจากสมการที่ (4)	107
23	ผลการคำนวณผลเฉลยเชิงทนทานสำหรับปัญหาตัวอย่าง	108
24	เปรียบเทียบข้อมูลนำเข้าและออกของวิธีการเข้าสู่ผลเฉลย	111
25	เปรียบเทียบข้อดีและข้อเสียของวิธีการเข้าสู่ผลเฉลย	112

สารบัญตาราง (ต่อ)

ตารางผนวกที่		**หน้า**
ก1	ผลเฉลยเชิงทนทานสำหรับปัญหา R101	125
ก2	ผลเฉลยเชิงทนทานสำหรับปัญหา R201	129
ก3	ผลเฉลยเชิงทนทานสำหรับปัญหา C101	133
ก4	ผลเฉลยเชิงทนทานสำหรับปัญหา C201	137
ก5	ผลเฉลยเชิงทนทานสำหรับปัญหา RC101	141
ก6	ผลเฉลยเชิงทนทานสำหรับปัญหา RC201	145
ข1	ผลการเปรียบเทียบขั้นตอนวิธีการสำหรับปัญหา R1 และ R2	150
ข2	ผลการเปรียบเทียบขั้นตอนวิธีการสำหรับปัญหา C1 และ C2	156
ข3	ผลการเปรียบเทียบขั้นตอนวิธีการสำหรับปัญหา RC1 และ RC2	161
ข4	ผลการเปรียบเทียบขั้นตอนวิธีการสำหรับปัญหาขนาด 100 เมือง	166

สารบัญภาพ

ภาพที่ | หน้า

ภาพที่		หน้า
1	ปัญหาการจัดเส้นทางเดินรถขนส่ง (Vehicle Routing Problem)	5
2	a เส้นทางเดินรถขนส่งแบบเดิม และ b เส้นทางเดินรถขนส่งแบบเซฟวิ่ง	8
3	กรอบการทำงานของการตัดสินใจเชิงทนทาน (The Robust Decision Making Framework)	21
4	ความซับซ้อนของปัญหา VRP	26
5	ความซับซ้อนของปัญหา VRP ที่มีความไม่แน่นอน	27
6	กรอบการทำงานวิจัย (Research Framework)	28
7	ตัวอย่างช่วงเวลาที่ใช้ในการเดินทางขนส่งจากคลังสินค้า 1 ไปยังลูกค้า 3 จุด	29
8	ตัวอย่างการกำหนดสถานการณ์ที่ 1 กรณีการจราจรติดขัด	30
9	ตัวอย่างการกำหนดสถานการณ์ที่ 2 กรณีเกิดอุบัติเหตุ ณ เส้นทางระหว่างลูกค้า 2 และ 3	30
10	ตัวดำเนินการเคลื่อนย้าย (0, 1)	45
11	ตัวดำเนินการเคลื่อนย้าย (1, 0)	45
12	ตัวดำเนินการเคลื่อนย้าย (1, 1)	46
13	ตัวดำเนินการเคลื่อนย้าย (0, 2)	47
14	ตัวดำเนินการเคลื่อนย้าย (2, 0)	47
15	ตัวดำเนินการเคลื่อนย้าย (1, 2)	48
16	ตัวดำเนินการเคลื่อนย้าย (2, 1)	49
17	ตัวดำเนินการเคลื่อนย้าย (2, 2)	49
18	เปรียบเทียบวิธี FB และวิธี GB	51
19	การสลับสับเปลี่ยนภายในเส้นทาง	52
20	โครงสร้างหน่วยความจำทาบู (เริ่มต้น)	54
21	หน่วยความทรงจำทาบู (รอบที่ 1)	56
22	หน่วยความทรงจำทาบู (รอบที่ 2)	56
23	หน่วยความทรงจำทาบู (รอบที่ 3)	59
24	อัตราการเพิ่มขึ้นของเวลาประมวลผลด้วยคอมพิวเตอร์	71

สารบัญภาพ (ต่อ)

ภาพที่		หน้า
25	เส้นทางการเดินรถขนส่งสำหรับปัญหา R101, $\|S\|=10$, $\beta=0.2$ ด้วยเกณฑ์ตัดสินใจ Z_A	72
26	เส้นทางการเดินรถขนส่งสำหรับปัญหา R101, $\|S\|=10$, $\beta=0.2$ ด้วยเกณฑ์ตัดสินใจ Z_D	73
27	เส้นทางการเดินรถขนส่งสำหรับปัญหา R101, $\|S\|=10$, $\beta=0.2$ ด้วยเกณฑ์ตัดสินใจ Z_R	73
28	กราฟเปรียบเทียบเกณฑ์ตัดสินใจเชิงทนทานสำหรับจำนวนรถขนส่งของปัญหา R101	75
29	กราฟเปรียบเทียบเกณฑ์ตัดสินใจเชิงทนทานสำหรับเวลาเดินทางรวมของปัญหา R101	75
30	กราฟเปรียบเทียบเกณฑ์ตัดสินใจเชิงทนทานสำหรับจำนวนรถขนส่งของปัญหา R201	76
31	กราฟเปรียบเทียบเกณฑ์ตัดสินใจเชิงทนทานสำหรับเวลาเดินทางรวมของปัญหา R201	76
32	กราฟเปรียบเทียบเกณฑ์ตัดสินใจเชิงทนทานสำหรับจำนวนรถขนส่งของปัญหา C101	77
33	กราฟเปรียบเทียบเกณฑ์ตัดสินใจเชิงทนทานสำหรับเวลาเดินทางรวมของปัญหา C101	77
34	กราฟเปรียบเทียบเกณฑ์ตัดสินใจเชิงทนทานสำหรับจำนวนรถขนส่งของปัญหา C201	78
35	กราฟเปรียบเทียบเกณฑ์ตัดสินใจเชิงทนทานสำหรับเวลาเดินทางรวมของปัญหา C201	78
36	กราฟเปรียบเทียบเกณฑ์ตัดสินใจเชิงทนทานสำหรับจำนวนรถขนส่งของปัญหา RC101	79
37	กราฟเปรียบเทียบเกณฑ์ตัดสินใจเชิงทนทานสำหรับเวลาเดินทางรวมของปัญหา RC101	79

สารบัญภาพ (ต่อ)

ภาพที่ หน้า

38 กราฟเปรียบเทียบเกณฑ์ตัดสินใจเชิงทนทานสำหรับจำนวนรถขนส่งของปัญหา RC201 80
39 กราฟเปรียบเทียบเกณฑ์ตัดสินใจเชิงทนทานสำหรับเวลาเดินทางรวมของปัญหา RC201 80
40 การวิเคราะห์ความไวเมื่อเปลี่ยนแปลงค่า $|S|$ 82
41 การวิเคราะห์ความไวเมื่อเปลี่ยนแปลงค่า β 83
42 กราฟเปรียบเทียบวิธีการเข้าสู่ผลเฉลยเมื่อ $|S|=10$ 85
43 กราฟเปรียบเทียบวิธีการเข้าสู่ผลเฉลยเมื่อ $|S|=20$ 85
44 กราฟเปรียบเทียบวิธีการเข้าสู่ผลเฉลยเมื่อ $|S|=40$ 86
45 กราฟเปรียบเทียบวิธีการเข้าสู่ผลเฉลยเมื่อ $|S|=80$ 86
46 กราฟเปรียบเทียบคุณภาพของผลเฉลยโดยเฉลี่ย 89
47 กราฟเปรียบเทียบสมรรถนะระหว่าง HY และ GA สำหรับปัญหา R1 (จำนวนรถขนส่งต่ำสุด, ความสามารถในการบรรทุกสินค้า = 200 หน่วย) 90
48 กราฟเปรียบเทียบสมรรถนะระหว่าง HY และ GA สำหรับปัญหา R1 (เวลาเดินทางต่ำสุด) 90
49 กราฟเปรียบเทียบสมรรถนะระหว่าง HY และ GA สำหรับปัญหา R2 (จำนวนรถขนส่งต่ำสุด, ความสามารถในการบรรทุกสินค้า = 1000 หน่วย) 91
50 กราฟเปรียบเทียบสมรรถนะระหว่าง HY และ GA สำหรับปัญหา R2 (เวลาเดินทางต่ำสุด) 91
51 กราฟเปรียบเทียบสมรรถนะระหว่าง HY และ GA สำหรับปัญหา C1 (จำนวนรถขนส่งต่ำสุด, ความสามารถในการบรรทุกสินค้า = 200 หน่วย) 92
52 กราฟเปรียบเทียบสมรรถนะระหว่าง HY และ GA สำหรับปัญหา C1 (เวลาเดินทางต่ำสุด) 92
53 กราฟเปรียบเทียบสมรรถนะระหว่าง HY และ GA สำหรับปัญหา C2 (จำนวนรถขนส่งต่ำสุด, ความสามารถในการบรรทุกสินค้า = 700 หน่วย) 93
54 กราฟเปรียบเทียบสมรรถนะระหว่าง HY และ GA สำหรับปัญหา C2 (เวลาเดินทางต่ำสุด) 93

สารบัญภาพ (ต่อ)

ภาพที่ **หน้า**

55 กราฟเปรียบเทียบสมรรถนะระหว่าง HY และ GA สำหรับปัญหา RC1 (จำนวนรถขนส่งต่ำสุด, ความสามารถในการบรรทุกสินค้า = 200 หน่วย) 94
56 กราฟเปรียบเทียบสมรรถนะระหว่าง HY และ GA สำหรับปัญหา RC1 (เวลาเดินทางต่ำสุด) 94
57 กราฟเปรียบเทียบสมรรถนะระหว่าง HY และ GA สำหรับปัญหา RC2 (จำนวนรถขนส่งต่ำสุด, ความสามารถในการบรรทุกสินค้า = 1000 หน่วย) 95
58 กราฟเปรียบเทียบสมรรถนะระหว่าง HY และ GA สำหรับปัญหา RC2 (เวลาเดินทางต่ำสุด) 95
59 กราฟเปรียบเทียบสมรรถนะระหว่าง HY และ GA สำหรับปัญหาขนาด 100 เมือง (จำนวนรถขนส่งต่ำสุด) 96
60 กราฟเปรียบเทียบสมรรถนะระหว่าง HY และ GA สำหรับปัญหาขนาด 100 เมือง (เวลาเดินทางต่ำสุด) 97
61 กราฟเปรียบเทียบเวลาในการประมวลผลระหว่างวิธี HY และ GA (Com 1) 99
62 กราฟเปรียบเทียบเวลาในการประมวลผลระหว่างวิธี HY และ GA (Com 2) 99
63 กราฟเปรียบเทียบเวลาในการประมวลผลระหว่างวิธี HY และ GA (Com 3) 100
64 กราฟเปรียบเทียบเวลาในการประมวลผลระหว่างวิธี HY และ GA (Com 4) 100
65 กราฟเปรียบเทียบเวลาในการประมวลผลระหว่างวิธี HY และ GA (Com 5) 101
66 กราฟเปรียบเทียบเวลาในการประมวลผลระหว่างวิธี HY และ GA (Com 6) 101
67 กราฟเปรียบเทียบเวลาในการประมวลผลสำหรับขั้นตอนวิธีการ HY 103
68 ผลเฉลยสภาพทนทานสัมบูรณ์สำหรับปัญหาตัวอย่าง 108
69 ผลเฉลยความแตกต่างทนทานสำหรับปัญหาตัวอย่าง 109
70 ผลเฉลยสภาพทนทานแบบสัมพัทธ์สำหรับปัญหาตัวอย่าง 109

คำอธิบายสัญลักษณ์และคำย่อ

CCP = Chance Constrained Program

CVRPSD = Capacitate Vehicle Routing Problem with Stochastic Demands

GA = Genetic Algorithm

HY = Hybrid Algorithm

LP = Linear Programming

LSD = Local Search Descent

MPFIH = Modified Push Forwards Insertion Heuristic

NV = Number of Vehicle

PFIH = Push Forwards Insertion Heuristic

SA = Simulated Annealing Algorithm

SP = Stochastic Programming

SVRP = Stochastic Vehicle Routing Problem

TS = Tabu Search

TSP = Traveling Salesman Problem

TSP = Traveling Salesman Problem

TT = Total Time

VRPSC = Vehicle Routing Problem with Stochastic Customer

VRPSD = Vehicle Routing Problem with Stochastic Demands

VRPSST = Vehicle Routing Problem with Stochastic Service Time

VRPSTT = Vehicle Routing Problem with Stochastic Travel Time

VRPTW = Vehicle Routing Problem with Time Windows

ขั้นตอนวิธีการสำหรับการหาผลเฉลยเชิงทนทานของปัญหาการจัดเส้นทางเดินรถขนส่งแบบมีกรอบเวลาและเวลาเดินทางไม่แน่นอน

Algorithm for Determining Robust Solutions of Vehicle Routing Problem with Time Windows and Uncertain Travel Times

คำนำ

ปัญหาการจัดเส้นทางเดินรถขนส่ง (Vehicle Routing Problem) เป็นส่วนหนึ่งของการจัดการด้านการขนส่งและโลจิสติกส์ ในการจัดเส้นทางเดินรถขนส่งจากคลังสินค้า ไปยังลูกค้าที่กระจายอยู่ตามจุดต่างๆ ซึ่งเป็นปัญหาการตัดสินใจที่มีความซับซ้อนในระดับเอ็นพีฮาร์ด (NP-Hard Problem) การหาผลเฉลยที่เหมาะสมด้วยวิธีแม่นตรง (Exact Method) จึงกระทำได้ยาก โดยเฉพาะเมื่อปัญหามีขนาดใหญ่ และมีเงื่อนไขเพิ่มมากขึ้น ปัจจุบันนักวิจัยมีความพยายามในการค้นหาวิธีการในการแก้ปัญหาดังกล่าว ซึ่งแต่ละวิธีที่ค้นพบอาจมีความเหมาะสม และใช้งานได้ดีเฉพาะบางปัญหา และอาจมีประสิทธิภาพต่ำลงเมื่อทดลองใช้กับปัญหาอื่นที่มีเงื่อนไขที่แตกต่างกัน

งานวิจัยนี้ ได้นำเสนอขั้นตอนวิธีการสำหรับปัญหา VRP แบบมีกรอบเวลา (Time Windows) หรือ VRPTW และมีระยะเวลาในการขนส่งที่ไม่แน่นอนจากสภาพการจราจรในแต่ละวัน เช่น การเกิดอุบัติเหตุ การจราจรคับคั่งในช่วงเวลาเร่งรีบ และสภาวะอากาศที่เลวร้าย สิ่งเหล่านี้ล้วนมีผลกระทบต่อการวางแผนเส้นทางเดินรถในแต่ละวัน ความไม่แน่นอนกลายเป็นส่วนสำคัญในการแก้ปัญหา การหาผลเฉลยที่เหมาะสมภายใต้สภาวะการณ์ที่ไม่แน่นอนนี้ เป็นสิ่งที่สนองตอบต่อความต้องการของผู้ตัดสินใจ (Decision Makers) มากที่สุด

หลักการทั่วไปที่นิยมใช้สำหรับการหาผลเฉลยภายใต้ความไม่แน่นอนนี้ อาจแบ่งได้เป็น 2 ประเภทด้วยกัน คือ วิธีการเข้าสู่ค่าเหมาะที่สุดเชิงกำหนด (Deterministic Optimization Approach) และวิธีการเข้าสู่ค่าเหมาะที่สุดเชิงสโทแคสติก (Stochastic Optimization Approach) วิธีการเข้าสู่ค่าเหมาะที่สุดเชิงกำหนดนั้น มีจุดอ่อนค่อนข้างมากเมื่อปัญหามีความไม่แน่นอน เนื่องจากวิธีการนี้เป็นการหาผลเฉลยจากการประมาณค่าความไม่แน่นอนให้เป็นค่าคงที่ หรือเลือกสถานการณ์ที่คาดว่าอาจเกิดขึ้นบ่อยมากที่สุด เป็นตัวแทนของปัญหา ผลเฉลยที่ได้จึงตอบสนองต่อสถานการณ์ที่ถูกเลือกมาเพียงหนึ่งเดียวเท่านั้น เมื่อความไม่แน่นอนมีปริมาณสูงขึ้น ประสิทธิภาพของวิธีการนี้จะด้อยลงทันที

ในช่วงเวลากว่า 50 ปีที่ผ่านมา การค้นคว้าวิจัยที่เกี่ยวข้องกับวิธีการเข้าสู่ค่าเหมาะที่สุดเชิงสโทแคสติกได้รับความนิยมอย่างกว้างขวาง มีการพัฒนาและปรับปรุงจนกลายเป็นวิธีการที่มีประสิทธิภาพมากที่สุดในแง่ของการแก้ปัญหาภายใต้ความไม่แน่นอน สามารถแก้ไขจุดด้อยที่เกิดจากวิธีเชิงกำหนดได้ โดยผลเฉลยที่ได้มีความทนทานต่อความไม่แน่นอนมากกว่าสถานการณ์เดียว ถึงแม้วิธีการสโทแคสติก จะพิสูจน์ให้เห็นถึงความสามารถในการต้านทานกับความไม่แน่นอนได้นั้น หากพิจารณาในแง่ของการนำไปประยุกต์ใช้งานจริง พบว่ามีความยุ่งยากและไม่สนองตอบต่อความต้องการของผู้ตัดสินใจ เนื่องจากวิธีการนี้ต้องอาศัยข้อมูลความไม่แน่นอนในอดีต ที่มีการวิเคราะห์ด้วยวิธีทางสถิติและแสดงในรูปของค่าความน่าจะเป็น หรือการแจกแจง อันเป็นตัวแทนของความไม่แน่นอน ซึ่งการเก็บรวบรวมข้อมูลนั้น นอกจากต้องอาศัยระยะเวลานานในการรวบรวมข้อมูลจำนวนมากแล้ว ขั้นตอนการประมาณค่าพารามิเตอร์จะต้องเกิดขึ้นอย่างละเอียดถี่ถ้วน และการวิเคราะห์ต้องเป็นไปอย่างรอบคอบ มิฉะนั้นผลเฉลยที่ได้อาจเบี่ยงเบนไปจากค่าจริง จุดด้อยที่สำคัญของวิธีการสโทแคสติก คือ ผลเฉลยที่ได้เป็นผลเฉลยที่เกิดจากค่าคาดหวังเท่านั้น (Expected Outcome) อันหมายถึงผู้ตัดสินใจอาจมีโอกาสเสี่ยง เมื่อสถานการณ์จริงที่เกิดขึ้นอาจตกอยู่ในช่วงสูงสุด หรือต่ำสุด ซึ่งในแง่ของธุรกิจการเกิดเหตุการณ์เช่นนี้เพียงครั้งเดียว อาจส่งผลกระทบร้ายแรงในแง่ของความน่าเชื่อถือ หรือการเสียโอกาส

ด้วยเหตุนี้ งานวิจัยนี้จึงมุ่งเน้นผลงานที่สนองตอบต่อความต้องการในการใช้งานจริงของผู้ตัดสินใจ โดยนำเสนอวิธีการเข้าสู่ค่าเหมาะที่สุดเชิงทนทาน (Robust Optimization Approach) ในแง่ของการศึกษาวิจัยของวิธีแบบสภาพทนทานนี้ อาจไม่ใช่เรื่องใหม่ในแวดวงการศึกษา เนื่องจากวิธีการดังกล่าวถูกใช้ในการออกแบบผลิตภัณฑ์ให้เกิดคุณภาพสูงสุด โดยผลิตภัณฑ์นี้ต้องมีสภาพทนทานต่อสภาวะการใช้งานที่แตกต่างกันไป ผู้ผลิตจะต้องออกแบบให้ผลิตภัณฑ์มีความทนทานสูงสุด เพื่อป้องกันการแตกหักเสียหาย Bertsimas (1992) เป็นนักวิจัยคนแรกที่นำเอาแนวคิดของสภาพทนทานนี้มาใช้กับปัญหาการตัดสินใจ เพื่อหาผลเฉลยที่เหมาะสมภายใต้ความไม่แน่นอน และหลังจากปี 2000 เป็นต้นมา วิธีการแบบสภาพทนทานจึงได้รับความนิยมมากขึ้นตามลำดับ การพัฒนาขั้นตอนวิธีการแนวใหม่โดยอาศัยวิธีการแบบสภาพทนทาน (Robustness Approach) จึงเป็นวัตถุประสงค์สำคัญของงานวิจัย เพื่อค้นหาค่าเหมาะที่สุดในการจัดเส้นทางเดินรถขนส่งให้มีสภาพทนทานต่อความไม่แน่นอนของเวลาเดินทางมากที่สุด

วัตถุประสงค์

วัตถุประสงค์หลักของงานวิจัยนี้ คือ การพัฒนาขั้นตอนวิธีการแนวใหม่ที่ผสมผสานระหว่างวิธีฮิวริสติกส์ (Heuristic) และวิธีเมต้าฮิวริสติกส์ (Meta-Heuristic) ในการหาผลเฉลยเชิงทนทาน สำหรับปัญหาการจัดเส้นทางเดินรถขนส่งแบบมีกรอบเวลา และเวลาเดินทางไม่แน่นอน โดยมีวัตถุประสงค์ย่อยดังต่อไปนี้

1. หาเวลาเดินทางรวมที่สั้นที่สุดและมีความทนทานต่อความไม่แน่นอนมากที่สุด
2. หาจำนวนรถขนส่งที่น้อยที่สุด

ขอบข่ายงานวิจัย ประกอบด้วย เป็นปัญหาแบบคลังสินค้าเดี่ยว (Single Depot) ความต้องการของลูกค้ามีความหลากหลาย (Multiple Demands) ใช้รถขนส่งประเภทเดียวกัน (Same Type of Vehicle) รถขนส่งมีขีดจำกัดในการบรรทุกสินค้า (Capacitate Vehicle) มีการกำหนดกรอบเวลาในการขนส่ง (Time Windows) และกำหนดเวลาเดินทางเป็นแบบช่วงเวลา (Finite Interval Travel Times)

สมมุติฐานของงานวิจัย คือ การจัดเส้นทางเดินรถขนส่งด้วยผลเฉลยเชิงทนทาน อาจให้ผลเฉลยที่ครอบคลุมความไม่แน่นอนทั้งหมด เมื่อสามารถสร้างข้อมูลที่เป็นตัวแทนของทุกๆ สถานการณ์ที่คาดว่าจะเกิดขึ้นในปัจจุบันและอนาคต

การตรวจเอกสาร

จากการศึกษาทฤษฎีและงานวิจัยที่เกี่ยวข้องพบว่า งานวิจัยที่เกี่ยวข้องกับปัญหา VRP ได้รับความนิยมอย่างแพร่หลายทั้งในและต่างประเทศ Dantzig and Ramser (1959) ได้นำเสนอวิธีการแก้ปัญหา VRP ซึ่งนับเป็นจุดเริ่มต้นของการพัฒนาขั้นตอนวิธีการต่างๆ ในการแก้ปัญหาดังกล่าว มาจนกระทั่งถึงปัจจุบันนี้ ซึ่งมีทั้งการพัฒนาที่รูปแบบของปัญหา และเทคนิควิธีการ งานวิจัยของ Toth and Vigo (2001) ได้แบ่งปัญหา VRP ไว้ดังต่อไปนี้

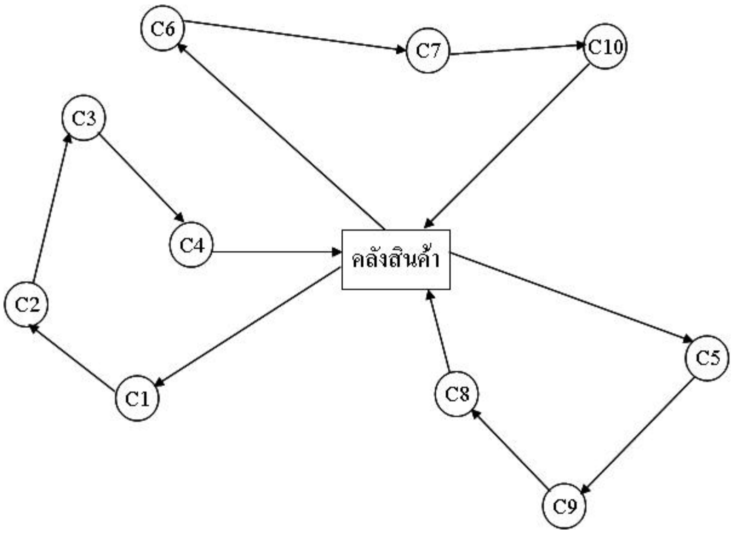
ตารางที่ 1 รูปแบบปัญหา VRP

รูปแบบปัญหา	ชื่อย่อ	ความหมาย
VRP with Capacity Restriction	CVRP	VRP แบบมีข้อจำกัดด้านความสามารถในการบรรทุกสินค้า
VRP with Time Windows	VRPTW	VRP แบบมีกรอบเวลา
VRP with Backhaul	VRPB	VRP แบบมีการขนส่งกลับมายังคลังเดิม
Pickup and Delivery	VRPPD	VRP แบบมีการรับและส่งสินค้า
VRPPD with Time Windows	VRPPDTW	VRPPD แบบมีกรอบเวลา
VRP with Multiple Depot	MDVRP	VRP แบบมีหลายคลังสินค้า
Periodic VRP	PVRP	VRP แบบมีช่วงเวลา
Periodic VRPTW	PVRPTW	VRP แบบมีช่วงเวลาและกรอบเวลา
Stochastic VRP	SVRP	VRP แบบมีความไม่แน่นอน

จากตารางที่ 1 ปัญหาสำหรับงานวิจัยนี้ จัดอยู่ในประเภท VRPTW ที่มีเวลาเดินทางในการขนส่งไม่แน่นอน ฉะนั้นการศึกษางานวิจัยที่เกี่ยวข้องจึงมุ่งเน้นขั้นตอนวิธีการในการแก้ปัญหา VRPTW และการสร้างรูปแบบปัญหาของความไม่แน่นอน ในบทนี้ได้กล่าวถึงงานวิจัยที่เกี่ยวข้องอันประกอบด้วย ปัญหาพื้นฐาน VRP และ VRPTW ปัญหาการจัดรูปแบบของความไม่แน่นอนและวิธีการแบบสภาพทนทาน

สำหรับพื้นฐานของ VRPTW คือ VRP หรือในกรณีที่มีรถขนส่งเพียงคันเดียวนั้น ปัญหา VRP สามารถเทียบได้กับปัญหาการจัดเส้นทางของพนักงานขาย (Traveling salesman problem) ซึ่ง

เป็นปัญหาที่รู้จักกันอย่างแพร่หลายในแวดวงวิชาการ และปัญหา TSP ได้รับการพิสูจน์ทางทฤษฎีแล้วว่าเป็นปัญหาเอ็นพี-สมบูรณ์ (NP-complete) หรือเป็นปัญหาที่มีความซับซ้อนในการหาผลเฉลย (Savelsbergh, 1985) สำหรับปัญหา VRP ประกอบด้วยรถขนส่งจำนวนมากขึ้น ทำให้เกิดเส้นทางเดินรถหลายเส้นทาง หรืออาจกล่าวอีกนัยหนึ่งว่า VRP ประกอบด้วย TSP ที่มีความเป็นอิสระต่อกันจำนวนหลายเส้นทาง เพราะฉะนั้น VRP จึงถูกจัดเป็นปัญหาแบบเอ็นพี-สมบูรณ์ โดยมิต้องมีการพิสูจน์ทางทฤษฎี ตัวอย่างปัญหาจริงที่สามารถจัดเป็น VRP ได้แก่ การจัดเส้นทางเดินรถรับส่งนักเรียน การส่งไปรษณีย์ การส่งหนังสือพิมพ์ การขนส่งน้ำมัน หรือการขนส่งสินค้าทั่วไป โดยมีเงื่อนไขสำคัญ คือ รถขนส่งทุกคันต้องออกจากคลังสินค้าเพื่อบริการลูกค้าให้ครบทุกจุด และกลับมายังคลังสินค้าเดิมทุกครั้ง โดยจะต้องส่งสินค้าให้ครบตามจำนวนความต้องการของลูกค้าภายใต้ขีดจำกัดความสามารถในการบรรทุกสินค้า โดยทั่วไปลักษณะของปัญหา VRP จะแสดงในรูปแบบของทิศทางการเดินรถจากคลังสินค้าไปยังลูกค้า ณ จุดต่างๆ ดังภาพที่ 1



ภาพที่ 1 ปัญหาการจัดเส้นทางเดินรถขนส่ง (Vehicle Routing Problem)

โครงสร้างของรูปแบบปัญหาแทนด้วย $G = (V, A)$ โดยที่ $V = \{\upsilon_1, \upsilon_2, \upsilon_3, ..., \upsilon_n\}$ แทนเซตของปม (Nodes) υ_1 แทนคลังสินค้า และ υ อื่นๆ แทนตำแหน่งของลูกค้าแต่ละราย และ A แทนเส้นเชื่อมระหว่างคลังสินค้าไปยังลูกค้าแต่ที่จุด และระหว่างลูกค้า υ_i ไปยังลูกค้า υ_j โดยที่ $i \neq j$

วิธีการที่นิยมใช้ในการแก้ปัญหา VRP สามารถแบ่งได้ดังนี้

1. วิธีแม่นตรง (Exact Method)

เป็นวิธีการคำนวณทุกทางเลือกของผลเฉลย แล้วจึงเลือกผลเฉลยที่ให้ค่าที่ดีที่สุด เช่น วิธีการขยายและจำกัดเขต (Branch and Bound) (Fisher, 1993) ซึ่งเหมาะสมสำหรับปัญหา VRP ที่มีจำนวนลูกค้าไม่เกิน 100 ปม วิธีกำหนดการเชิงพลวัต (Dynamic Programming) และขั้นตอนวิธีการการตัดระนาบ (Cutting Plane Algorithm) เป็นต้น Agarwal *et al.* (1989) ได้นำเสนอวิธีการแบ่งส่วนของเซตผลเฉลย (Set Partitioning) บนพื้นฐานของวิธีแม่นตรงเพื่อแก้ปัญหา VRPส่วนการประยุกต์ใช้ขั้นตอนวิธีการอื่นๆ ที่เกี่ยวข้องได้มีการอธิบายในรายละเอียดโดย Christofides *et al.* (1985); Laporte (1992) ข้อด้อยของวิธีแม่นตรง คือ เมื่อปัญหามีขนาดใหญ่เพิ่มมากขึ้น วิธีการอาจไม่สามารถรับประกันการหาค่าเหมาะที่สุด (Optimization) ได้ เมื่อจำนวนลูกค้าเพิ่มมากขึ้น (มากกว่า 100 ปม) รวมทั้งใช้เวลานานในการประมวลผลด้วยคอมพิวเตอร์ซึ่งเป็นการสิ้นเปลืองทรัพยากรโดยใช่เหตุ

2. วิธีฮิวริสติกส์ (Heuristic Method)

เป็นวิธีการหาผลเฉลยที่ดีพอเพียง ภายในเวลาจำกัด หรือ “Good enough and fast enough solution” ซึ่งเป็นวิธีที่เหมาะสมสำหรับปัญหาเอ็นพี-สมบูรณ์ วิธีฮิวริสติกส์ถูกสร้างขึ้นเพื่อใช้ในการหาผลเฉลยของแต่ละปัญหาเท่านั้น ดังนั้นวิธีฮิวริสติกส์ที่สามารถหาผลเฉลยที่ดีสำหรับปัญหาหนึ่งจึงไม่สามารถนำไปใช้หาผลเฉลยของอีกปัญหาหนึ่งได้ นอกจากนี้ ในบางปัญหาที่มีความซับซ้อนมาก การสร้างตัวแปรและเงื่อนไขในตัดสินใจให้อยู่ในรูปของแบบจำลองทางคณิตศาสตร์ (Mathematical Model) อาจกระทำได้ยาก จึงไม่สามารถใช้วิธีแม่นตรง หรือใช้เทคนิคบางอย่างจากวิธีกำหนดการเชิงเส้นตรง (Linear Programming) ได้

วิธีฮิวริสติกส์จึงเป็นวิธีที่เหมาะสมในการหาผลเฉลยของปัญหาการตัดสินใจที่มีลักษณะต่างๆ ดังนี้ ปัญหาการตัดสินใจที่มีโครงสร้างไม่สมบูรณ์ หรือไม่สามารถเขียนขั้นตอนวิธีในการหาผลเฉลยที่ชัดเจนได้ เช่น ปัญหาไม่สามารถเขียนในรูปแบบจำลองทางคณิตศาสตร์ที่เหมาะสมได้ แบบจำลองที่สร้างขึ้นไม่ครอบคลุมผลเฉลยที่สอดคล้องกับเงื่อนไขทั้งหมด หรือปัญหาที่ไม่มีขั้นตอนวิธีการใดๆ สามารถหาค่าที่เหมาะที่สุดได้ ในหลายกรณี ที่นักวิจัยพยายามที่จะแก้ไขปัญหาที่มีโครงสร้างไม่สมบูรณ์นี้ โดยไม่พิจารณาเงื่อนไขบางประการ หรือตั้งสมมุติฐานของปัญหาให้ง่ายขึ้น เพื่อให้ปัญหามีโครงสร้างที่สมบูรณ์ และสามารถนำวิธีการที่มีอยู่มาประยุกต์ใช้เพื่อหาค่าที่เหมาะที่สุด ตัวอย่างของวิธีการนี้พัฒนาขึ้นโดย Fisher and Jaikumar (1981); Christofides *et al.*

(1981) ได้อธิบายการใช้วิธีการผ่อนปรนแบบลากรานจ์เจียน (Lagrangean Relaxation) สำหรับปัญหา VRP และในกรณีที่ปัญหาการตัดสินใจที่มีตัวแปรและเงื่อนไขของปัญหาเป็นจำนวนมาก ถึงแม้บางปัญหาสามารถเขียนแบบจำลองทางคณิตศาสตร์ได้ แต่ขั้นตอนวิธีการที่มีอยู่ในปัจจุบันไม่สามารถหาผลเฉลยของปัญหาได้ในทางปฏิบัติ ซึ่งมักพบโดยทั่วไปในปัญหาของภาคธุรกิจ และอุตสาหกรรมขนาดใหญ่ ในทางปฏิบัตินักวางแผนอาจไม่ต้องการผลเฉลยที่ดีที่สุด แต่ต้องการผลเฉลยที่ดีเพียงพอที่สามารถหาได้ภายในเวลาที่กำหนด และด้วยวิธีการที่ไม่ยุ่งยากซับซ้อนมากนัก หรือกล่าวอีกนัยหนึ่งว่า นักวางแผนต้องการค่าฟังก์ชันวัตถุประสงค์ที่ดีที่ไม่ขัดแย้งต่อเงื่อนไข ทั้งนี้โดยทั่วไปนักวางแผนมักมีผลเฉลย หรือทางเลือกอยู่ในใจอยู่แล้ว แต่ต้องการเพียงเครื่องมือหรือขั้นตอนวิธีการที่จะสนับสนุนทางเลือกนั้นๆ หรือใช้เป็นเครื่องมือช่วยในการตัดสินใจนั่นเอง (ณกร , 2548)

ในกรณีปัญหา VRP จัดเป็นปัญหาการหาค่าเหมาะที่สุดแบบไม่ต่อเนื่อง (Discrete Optimization) ซึ่งสามารถสร้างแบบจำลองทางคณิตศาสตร์ได้ แต่ด้วยรูปแบบที่ยุ่งยากซับซ้อน การประยุกต์ใช้วิธีแม่นตรง หรือใช้เทคนิค LP อาจกระทำได้ยาก หรือไม่สามารถกระทำได้เลย เมื่อจำนวนตัวแปรตัดสินใจเพิ่มจำนวนมากขึ้นอย่างรวดเร็ว จนบางครั้งอาจไม่สามารถหาขอบเขตต่ำสุดและสูงสุดได้ วิธีฮิวริสติกส์จึงน่าจะเป็นวิธีการที่เหมาะสมสำหรับปัญหา VRP ในขณะนี้ โดยวิธีฮิวริสติกส์ที่นิยมนำมาใช้แก้ปัญหา VRP ประกอบด้วย

2.1 วิธีการสร้างผลเฉลย (Constructive Method) เป็นวิธีการที่ไม่สามารถรับประกันการหาค่าเหมาะที่สุดได้ วิธีการนี้อาศัยข้อมูลของปัญหาในการสร้างขั้นตอนวิธีการหาผลเฉลย ตัวอย่างของวิธีการนี้ที่ได้รับความนิยม และได้รับการยอมรับว่าเป็นวิธีที่มีประสิทธิภาพ ได้แก่ วิธีการเซฟวิ่งหรือการแทรก (Savings or Insertion Procedure) ซึ่งพัฒนาขึ้นโดย Clarke and Wright (1964) สำหรับแก้ปัญหา VRP โดยเฉพาะ วิธีการนี้เป็นการสร้างผลเฉลยทีละขั้นตอน โดยเริ่มจากผลเฉลยเริ่มต้นซึ่งอาจยังมีความเป็นไปไม่ได้ และสร้างผลเฉลยในลำดับถัดมาที่ทำให้ฟังก์ชั่นของการเซฟวิ่งมีค่าเพิ่มมากขึ้น หรือเลือกแทรกลูกค้าเข้ามาในเส้นทางเดิมที่มีอยู่ โดยที่ความต้องการรวมต้องไม่เกินความสามารถในการบรรทุกสินค้าของรถขนส่ง การแทรกจะเกิดขึ้นเรื่อยๆ จนกระทั่งได้เส้นทางที่เหมาะสมที่สุด โดยขั้นตอนมีดังนี้

ขั้นตอนที่ 1 คำนวณค่า $S_{ij} = C_{1j} + C_{1i} - C_{ij}$ สำหรับทุกคู่ของลูกค้า i และ j โดยที่ $i, j = 2,3,\ldots,n$ และ S_{ij} คือ ต้นทุนที่สามารถประหยัดได้จากผลของการเชื่อมโยงปม (i, j) ทำให้เกิดเส้นทาง $(1,i,j,1)$ แทนการเสียต้นทุนในการเดินทาง 2 เส้นทาง คือ $(1,i,1)$ และ $(1,j,1)$

ภาพที่ 2 a เส้นทางการเดินรถแบบเดิม และ b เส้นทางการเดินรถแบบเซฟวิ่ง

ขั้นตอนที่ 2 เรียงลำดับค่า S_{ij} จากมากไปหาน้อย

ขั้นตอนที่ 3 เริ่มต้นสลับสับเปลี่ยนจากค่า S_{ij} ที่มีค่ามากไปหาน้อย หรือจากบนลงล่าง

เวอร์ชั่นขนาน (Parallel Version)

ขั้นตอนที่ 4 ถ้าการเชื่อมโยงปมมีความเป็นไปได้และอยู่ภายใต้เงื่อนไขที่กำหนด ให้ยอมรับการเชื่อมโยงปมนั้นๆ หากมิใช่ให้ทำการปฏิเสธ

ขั้นตอนที่ 5 ทำการเชื่อมโยงปมต่อไป (ทำซ้ำขั้นตอนที่ 4) จนกระทั่งไม่สามารถเชื่อมโยงได้อีก หรือไม่เป็นไม่ตามเงื่อนไข

เวอร์ชั่นตามลำดับ (Sequential Version)

ขั้นตอนที่ 4 ค้นหาการเชื่อมโยงปมแรกที่มีความเป็นไปได้ ซึ่งสามารถใช้ได้ตั้งแต่ปม 1 และสิ้นสุดที่ปม 2

ขั้นตอนที่ 5 ถ้าเส้นทางไม่สามารถขยายต่อได้ เป็นอันสิ้นสุดการจัดเส้นทาง จากนั้นจึงเลือกเชื่อมโยงเส้นทางที่เป็นไปได้เส้นใหม่ตามลำดับ

ขั้นตอนที่ 6 ทำซ้ำขั้นตอน 4 และ 5 จนกระทั่งไม่สามารถเชื่อมโยงเส้นทางได้อีก

พฤติกรรมในกรณีที่แย่ที่สุดสำหรับวิธีการนี้ คือ ข้อจำกัดโดยฟังก์ชั่นเส้นตรง $\log_2(n)$ การคำนวณค่า S_{ij} ในขั้นตอนที่ 2 ต้องการการคำนวณทั้งสิ้น cn^2 โดยที่ c คือ ค่าคงที่ ในขั้นตอนที่ 3 ต้องการ $cn^2 \log(n)$ ขั้นตอนที่ 4 คือ n^2 ดังนั้นขั้นตอนวิธีการของ Clark and Wright (1964) ต้องการการคำนวณทั้งสิ้น $n^2 \log_2(n)$

2.2 ขั้นตอนวิธีการ 2-เฟส (Two-Phase Algorithm) เป็นวิธีการที่แบ่งปัญหา VRP ออกเป็น 2 ส่วน คือ การแบ่งกลุ่มลูกค้าเข้าสู่เส้นทางที่เป็นไปได้ภายใต้เงื่อนไข และการจัดลำดับเส้นทางเพื่อหาผลเฉลยที่ดีที่สุด ขั้นตอนวิธีการนี้ถูกแบ่งเป็น 2 วิธี คือ

2.2.1 วิธีการแบ่งส่วนก่อนแล้วจึงจัดเส้นทาง (Cluster-First Rout-Second Procedures) โดยเริ่มต้นจากการแบ่งกลุ่มลูกค้าโดยใช้ความต้องการสินค้าของลูกค้าเป็นตัวกำหนดโซนในการกระจายสินค้า จากนั้นจึงจัดเส้นทางที่เหมาะสมให้กับแต่ละส่วนที่ถูกแบ่งไว้แล้ว (Gillet and Miller, 1974)

2.2.2 วิธีการจัดเส้นทางก่อนแล้วจึงแบ่งส่วน (Rout-First Cluster-Second Procedures) เป็นการกำหนดเส้นทางในการเดินรถขนส่งก่อน แล้วจึงจัดลูกค้าเข้าสู่เส้นทางภายใต้เงื่อนไขของปัญหา ขั้นตอนวิธีการนี้ได้นำไปประยุกต์ใช้สำหรับการแก้ปัญหา VRP ในกรณีที่มีรถขนส่งหลายขนาด (Golden *et al.*, 1983)

2.3 วิธีการปรับปรุงหรือสลับสับเปลี่ยน (Improvement or Exchange Procedure) เป็นเทคนิคการสลับสับเปลี่ยนปมบนเส้นทางแบบฮิวริสติกส์ ซึ่งให้ผลเฉลยที่เข้าใกล้ค่าที่เหมาะสมที่สุด ส่วนการปรับปรุงโดยวิธีอื่น ได้นำเสนอไว้โดย Potvin and Rousseau (1995) เรียกว่า “การสลับสับเปลี่ยนแบบออร์-ออฟท์ (Or-Opt Exchange) ซึ่งเป็นเทคนิคการสลับสับเปลี่ยนปมขนาด 1, 2 หรือ 3 ปม โดยการแทรกหรือตัดทิ้งไปจากเส้นทางเดิม หรือในเส้นทางอื่นๆ ที่เลือกพิจารณา วิธีการ 2 ออฟท์ เป็นการสลับสับเปลี่ยน 2 ปม ที่เกิดจาก 2 เส้นทางที่แตกต่างกัน

จากงานวิจัยจำนวนมากแสดงให้เห็นว่าวิธีฮิวริสติกส์ยังคงเป็นวิธีการที่สามารถใช้ในการแก้ปัญหา VRP ได้อย่างมีประสิทธิภาพ ถึงแม้ปัจจุบันปัญหา VRP จะถูกดัดแปลงเพิ่มเติมไปจากรูปแบบปัญหาดั้งเดิม ความซับซ้อนจึงมีมากขึ้น ในขณะเดียวกันวิธีการฮิวริสติกส์ยังคงได้รับการพัฒนาปรับปรุงให้เหมาะสมตามไปด้วย ดังแสดงในงานวิจัยของ Paletta (2002) ได้นำเสนอวิธีการทางฮิวริสติกส์แบบใหม่ที่ง่าย และมีประสิทธภาพสำหรับการแก้ปัญหา VRP แบบมีรถขนส่งเพียงคันเดียว หรือปัญหา TSP แบบมีช่วงเวลา นอกจากนี้ Thangiah *et al.* (1996) ได้นำเสนอวิธีการฮิวริสติกส์แบบ 2-เฟส สำหรับแก้ปัญหา VRP แบบมีการขนส่งสินค้ากลับมายังคลังเดิม และมีกรอบเวลา (VRP with Backhauls and Time Windows) โดยเฟสแรก เริ่มจากการสร้างผลเฉลยเริ่มต้น (Initial Solution) ตามขั้นตอนวิธีฮิวริสติกส์แบบมีการแทรกไปข้างหน้า (Push-Forward Insertion Heuristic) (Solomon, 1987) และเฟสที่ 2 ทำการปรับปรุงผลเฉลยโดยการสลับสับเปลี่ยนคู่อันดับ λ และการสลับสับเปลี่ยนแบบ 2-ออฟท์ เพื่อให้ได้ผลเฉลยที่ดีขึ้น

วิธีฮิวริสติกส์เป็นวิธีที่พัฒนาขึ้นเพื่อหาผลเฉลยสำหรับปัญหาการตัดสินใจในแต่ละปัญหา นั่นคือ วิธีฮิวริสติกส์ที่ใช้ในการหาผลเฉลยของปัญหาหนึ่งอาจไม่สามารถหาผลเฉลยในปัญหาอื่นๆ ได้ หรือแม้กระทั่งนำไปใช้ในการหาผลเฉลยของปัญหาเดิมที่มีฟังก์ชั่นวัตถุประสงค์และเงื่อนไขของปัญหาเปลี่ยนแปลงไป ดังนั้น การพัฒนาวิธีฮิวริสติกส์ให้มีความยืดหยุ่นมากขึ้น และสามารถดัดแปลงเพื่อใช้ในการหาผลเฉลยของปัญหาใดๆ ได้อย่างรวดเร็วและมีประสิทธิภาพ ซึ่งในหัวข้อถัดไปได้กล่าวถึงขั้นตอนวิธีการเมต้าฮิวริสติกส์ ซึ่งเป็นวิธีที่พัฒนามาจากวิธีฮิวริสติกส์

3. วิธีเมต้าฮิวริสติกส์ (Meta-Heuristic Method)

เป็นวิธีที่ได้จากการพัฒนาและดัดแปลงวิธีฮิวริสติกส์ให้มีความยืดหยุ่นในการหาผลเฉลยของปัญหาการตัดสินใจใดๆ ที่มีความซับซ้อนและมีตัวแปรตัดสินใจจำนวนมาก ได้อย่างรวดเร็วและมีประสิทธิภาพ ถึงแม้ผลเฉลยที่ได้อาจไม่ใช่ผลเฉลยที่ให้ค่าเหมาะที่สุด หรือไม่สามารถรับประกันผลเฉลยที่ดีในทุกครั้งที่ทำการประมวลผลได้ แต่ผลเฉลยที่ได้เป็นที่ยอมรับ และค้นหาได้ภายในระยะเวลาอันเหมาะสม จึงเป็นที่นิยมอย่างกว้างขวางในงานวิจัยทุกแขนง ซึ่งในหัวข้อนี้จะขอกล่าวเฉพาะวิธีที่ได้รับความนิยมในการแก้ปัญหา VRP อันประกอบด้วย

3.1 ขั้นตอนวิธีการจำลองการอบอ่อน (Simulated Annealing Algorithm) ในการหาผลเฉลยของปัญหาการตัดสินใจ เริ่มต้นครั้งแรกเมื่อประมาณ ค.ศ. 1980 ลักษณะเด่นของวิธีการนี้ คือ มีขั้นตอนวิธีการหาผลเฉลยที่ง่าย และมีประสิทธิภาพ ซึ่งสามารถหาผลเฉลยที่ดีได้ภายในเวลาอันรวดเร็ว แนวคิดพื้นฐานของวิธี SA ได้ถูกตีพิมพ์ครั้งแรกในวารสารวิชาการโดย Metropolis *et al.* (1953) ซึ่งเป็นการใช้ขั้นตอนวิธีการจำลองการควบคุมการเย็นตัวลงของวัสดุในอ่างความร้อน (Heat Bath) ซึ่งเรียกขั้นตอนนี้ว่าการอบอ่อน (Annealing) วัสดุจะถูกให้ความร้อนจนกระทั่งถึงจุดหลอมเหลว ต่อจากนั้นเมื่อเวลาผ่านไปอุณหภูมิของวัสดุจะค่อยๆ ลดลง และทำให้วัสดุมีความแข็งเมื่อเย็นตัวลง โครงสร้างคุณสมบัติของวัสดุที่เย็นตัวลงนี้จะขึ้นอยู่กับอัตราการทำให้วัสดุนั้นเย็นตัวลง (Rate of Cooling)

ขั้นตอนวิธีการของ Metropolis นี้ได้จำลองการเปลี่ยนแปลงพลังงานของระบบที่แปรผันโดยกระบวนการเย็นตัวจนกระทั่งการเปลี่ยนแปลงนั้นลู่เข้าสู่สภาวะคงที่ ประมาณ 30 ปีต่อมา ได้มีการนำขั้นตอนวิธีการดังกล่าวมาใช้ในการแก้ปัญหาการตัดสินใจที่ต้องการหาค่าเหมาะต่ำสุดหรือสูงสุด ซึ่งเปรียบเสมือนว่า ผลเฉลยที่ได้จากวิธี SA จะค่อยๆ ลู่เข้าสู่ผลเฉลยที่ดีที่สุดเหมือนในช่วงเวลาการเย็นตัวลงของวัสดุ

SA มีลักษณะคล้ายกับวิธีการหาผลเฉลยแบบเฉพาะที่ (Local Search) จุดอ่อนที่สำคัญของ SA คือ การได้ค่าเหมาะที่สุดเฉพาะที่ (Local Optima) การแก้ปัญหานี้ในขั้นตอนวิธีการ โดยใช้การควบคุมอุณหภูมิที่มีลักษณะเป็นเทอร์โมไดนามิค ซึ่งกฎของเทอร์โมไดนามิคกล่าวว่า ณ ที่อุณหภูมิ υ ความน่าจะเป็นในการยอมรับการเปลี่ยนแปลงของพลังงาน (Energy Change or ΔE) มีค่าเท่ากับ $\mathrm{P}_{accept} = \exp\left(\frac{-\Delta \mathrm{E}}{\mathrm{k_B}\upsilon}\right)$ โดยที่ k_B คือ ค่าคงที่โบล์ทซแมน (Boltzmann) ขั้นตอนวิธีการ SA มีดังต่อไปนี้

ขั้นตอนที่ 1 สร้างเส้นทางเริ่มต้น เลือกอุณหภูมิเริ่มต้น $\upsilon = 0$ และรอบการทำซ้ำ r

ขั้นตอนที่ 2 กรณีไม่เป็นไปตามเงื่อนไขการหยุด ให้ทำตามขั้นตอนต่อไปนี้

ขั้นตอนที่ 2.1 ทำซ้ำ r รอบต่อไป

ขั้นตอนที่ 2.1.1 ปรับปรุงตัวเลขสุ่มของเส้นทางปัจจุบันให้เป็น (T') และกำหนด $\Delta = c(T') - c(T)$

ขั้นตอนที่ 2.1.2 เปรียบเทียบเลขสุ่ม $X, 0 \le X \le 1$

ขั้นตอนที่ 2.1.3 ถ้า $\Delta < 0$ หรือ $X < \exp\left(\frac{-\Delta}{\upsilon}\right)$

ขั้นตอนที่ 2.2 ปรับปรุงค่า υ และ r

ขั้นตอนที่ 3 ผลเฉลยที่ได้จากการแก้ปัญหา

3.2 ขั้นตอนวิธีการค้นหาแบบทาบู (Tabu Search) วิธี TS เป็นวิธีที่ค่อนข้างได้รับความนิยมอย่างมาก เพราะมีโครงสร้างของขั้นตอนวิธีการที่ไม่ซับซ้อนมากนัก และจากผลการวิจัยจำนวนมากสรุปว่า วิธี TS มีประสิทธิภาพมากกว่า SA ข้อแตกต่างที่เด่นชัดระหว่าง 2 วิธีนี้ คือ วิธี TS แก้ปัญหาการยอมรับค่าเหมาะสุดเฉพาะที่ โดยป้องกันไม่ให้มีการเกิดขึ้น (Pro-Active) แต่วิธี SA เป็นแก้ไขปัญหาเมื่อเกิดขึ้นแล้ว (Post-Active)

ความหมายของคำว่าทาบู (Tabu หรือ Taboo) ในขั้นตอนวิธีการทางคอมพิวเตอร์นั้น หมายถึง การห้ามหรือป้องกันขั้นตอนวิธีการเข้าไปยังผลเฉลยข้างเคียงที่ไม่ต้องการ โดยแนวคิดสำคัญ คือ การเพิ่มความฉลาด ซึ่งใช้ความทรงจำของคอมพิวเตอร์มาเกี่ยวข้อง คอมพิวเตอร์จะเรียนรู้จากรอบการวนซ้ำที่ผ่านมา (Search History) ในการแนะนำ หรือบอกทิศทางของผลเฉลยที่ดีหรือดีที่สุดในรอบการกระทำซ้ำถัดไปข้างหน้า

Glover (1989) เป็นผู้ริเริ่มขั้นตอนวิธีการ TS ซึ่งเป็นเทคนิคการประมาณค่าสำหรับการคำนวณเชิงคณิตศาสตร์ที่ดี เมื่อเทียบกับเทคนิคอื่นๆ ด้วยความยืดหยุ่น และมีประสิทธิภาพเหนือกว่าวิธีการคลาสสิกอื่นๆ TS เป็นการผสมผสานระหว่างการวนซ้ำเชิงกำหนด (Deterministic) และแบบความน่าจะเป็นเพื่อการยอมรับ ลักษณะการค้นหาเป็นแบบทางตรง คือ เมื่อเจอค่าต่ำสุดเฉพาะที่แล้วจึงเลื่อนไปตามผลเฉลยข้างเคียง หากพบว่าผลเฉลยนั้นให้ค่าฟังก์ชั่นวัตถุประสงค์ที่ดีขึ้น จะทำการปรับปรุงให้เป็นผลเฉลยปัจจุบัน หากพบว่าผลเฉลยแย่ลงกว่าเดิม ผลเฉลยนั้นจะถูกป้องกันด้วยวิธีการทาบู เพื่อไม่ให้มีการเลือกซ้ำได้อีก ซึ่งขั้นตอนวิธีการนี้จะทำการปรับปรุงผลเฉลยให้มีการเคลื่อนไหวอย่างต่อเนื่อง TS อาจยังไม่เป็นที่ยอมรับสำหรับวิธีการวนซ้ำแบบเพิ่มค่าทีละน้อย แต่อย่างไรก็ตามการออกแบบ TS เพื่อไม่ให้เลือกผลเฉลยที่ได้รับการปฏิเสธแล้วซ้ำอีกอาจเป็นประโยชน์ต่อการค้นหาเมื่อเทียบกับวิธีอื่นๆ โดยขั้นตอนทั่วไปของ TS เป็นดังต่อไปนี้

ขั้นตอนที่ 1 สร้างผลเฉลยเริ่มต้น x และกำหนดให้ $x^* = x, k = 0$

ขั้นตอนที่ 2 กำหนดค่า $k = k + 1$ และสร้างเซตย่อย V^* ของผลเฉลยใน $N(x, k)$ เป็นผลเฉลยข้างเคียงสำหรับการวนซ้ำครั้งที่ k

ขั้นตอนที่ 3 เลือกค่า $y \in V^*$ ที่เป็นไปตามฟังก์ชันวัตถุประสงค์หรือเงื่อนไข ปรับปรุงผลเฉลย และกำหนดให้ $x = y$

ขั้นตอนที่ 4 ถ้า $f(x) < f(x^*)$ ให้กำหนดค่า $x^* = x$

ขั้นตอนที่ 5 หยุด เมื่อเป็นไปตามเงื่อนไขของการหยุด ไม่เช่นนั้นให้กลับไปที่ขั้นตอนที่ 2

3.3 ขั้นตอนวิธีการของ Willard (Willard’s Algorithm) Willard เป็นหนึ่งในนักวิจัยชุดแรกๆ ที่พยายามประยุกต์ใช้วิธีการ TS ในการแก้ปัญหา VRP วิธีการนี้เริ่มจากการแปลงให้เป็นเส้นทางขนาดใหญ่ด้วยการทำซ้ำของแต่ละคลังสินค้า การหาผลเฉลยข้างเคียงกระทำโดยหาผลเฉลยที่เป็นไปได้ทั้งหมดด้วยวิธีการสลับสับเปลี่ยนแบบ 2-ออฟท์ หรือ 3-ออฟท์ (Gendreau *et al.*, 1997)

3.4 ขั้นตอนวิธีการของ Osman (Osman’s Algorithm) Osman ได้นำวิธีการค้นหาผลเฉลยข้างเคียงโดยใช้วิธีสลับสับเปลี่ยนค่า λ ร่วมกับการสลับสับเปลี่ยนแบบ 2-ออฟท์ เพื่อจัดปมลูกค้าเข้าสู่เส้นทางที่แตกต่างกัน ซึ่งการสลับสับเปลี่ยนระหว่างเส้นทาง 2 เส้น เพื่อให้ได้ผลเฉลยที่ดีขึ้นจะใช้ขั้นตอนวิธีการที่พัฒนาขึ้นเรียกว่า “การยอมรับได้ที่ดีที่สุด (Best-Admissible)” โดยผลเฉลยข้างเคียงทั้งหมดจะถูกสำรวจ และเลือกจากค่าผลเฉลยที่เป็นไปได้และดีที่สุด ส่วนขั้นตอนวิธีการอื่นที่พัฒนาต่อมาเรียกว่า “การยอมรับได้ที่ดีที่สุดในครั้งแรก (First-Best-Admissible)” โดยหลักการคือ การปรับปรุงค่าการยอมรับได้ในครั้งแรกจะถูกเลือก เมื่อมีค่าหนึ่งเกิดขึ้นก่อนหน้าแล้ว

3.5 วิธีการจัดเส้นทางแบบทาบู (Tabu Route) วิธีการจัดเส้นทางแบบทาบู พัฒนาขึ้นสำหรับปัญหาการจัดเส้นทางโดยเฉพาะ ซึ่งมีลักษณะที่แตกต่างจาก TS ดังนี้ (Gendreau *et al.*, 1997)

3.5.1 โครงสร้างของผลเฉลยข้างเคียง นิยามจากผลเฉลยทั้งหมดที่สามารถหาได้จากผลเฉลยในปัจจุบันโดยการย้ายปมลูกค้าจากเส้นทางหนึ่งไปแทรกอีกเส้นทางหนึ่งที่ใกล้เคียงโดยใช้หลักการ GENI

3.5.2 ขั้นตอนการค้นหาผลเฉลยอาจมีความเป็นไปไม่ได้ต่อความสามารถของรถขนส่ง หรือเงื่อนไขระยะทางที่มากที่สุด

3.5.3 วิธีการจัดเส้นทางแบบทาบูไม่ได้ใช้รายการทาบู (Tabu List) ที่บันทึกไว้ตามลำดับ แต่มีวิธีการเลือกแบบสุ่ม

3.5.4 วิธีการจัดเส้นทางแบบทาบูใช้กลยุทธ์ในการสลับสับเปลี่ยน โดยพิจารณาจากค่าปรับ (Penalizing) ของการโยกย้ายแต่ละปม

3.6 ขั้นตอนวิธีการของ Taillard (Taillard’s Algorithm) Taillard ได้นำคุณลักษณะของ TABUROUTE ที่มีวิธีการเลือกแบบสุ่ม และใช้กลยุทธ์การสลับสับเปลี่ยน วิธีการนี้หาผลเฉลยข้างเคียงโดยใช้การสลับสับเปลี่ยนค่า λ คล้ายกับขั้นตอนวิธีการของ Osman และแทนวิธีการแทรกด้วยวิธี GENI (Gendreau *et al.*, 1997)

3.7 ขั้นตอนวิธีการเชิงพันธุกรรม (Genetic Algorithm) กลวิธีนี้อาศัยหลักการและทฤษฎีจากการทำให้เกิดพันธุกรรมประชากร ในการสร้างขั้นตอนวิธีการที่เลียนแบบพฤติกรรมดังกล่าวโดยวิธีการจับคู่และการเกิดใหม่ของยีนส์เพื่อให้ได้ผลเฉลยที่เหมาะสมที่สุด ขั้นตอนทั่วไปของขั้นตอนวิธีการเชิงพันธุกรรมมีดังต่อไปนี้ (Thangiah *et al.*, 1995); (Aarts and Lenstra, 1997); (Muhlenbein, 1997)

ขั้นตอนที่ 1 (เริ่มต้น) สร้างกลุ่มประชากรเริ่มต้นจำนวน n ผลเฉลย

ขั้นตอนที่ 2 (ปรับปรุง) ใช้วิธีการค้นหาเฉพาะที่ โดยวิธีแทนประชากรจำนวน n ผลเฉลยด้วยค่าเหมาะสมเฉพาะที่ n

ขั้นตอนที่ 3 (การรวมกัน) ขยายขนาดจำนวนประชากรโดยเพิ่ม m ผลเฉลยของการเกิดประชากรกลุ่มใหม่ ซึ่งขนาดของจำนวนประชากรปัจจุบันเท่ากับ $n+m$

ขั้นตอนที่ 4 (ปรับปรุง) ใช้วิธีการค้นหาเฉพาะที่แทน *m* ผลเฉลยของการเกิดประชากรกลุ่มใหม่ ด้วยค่าเหมาะสมเฉพาะที่ *m*

ขั้นตอนที่ 5 (เลือก) ลดจำนวนประชากรลงสู่ขนาดเดิม โดยเลือก *n* ผลเฉลยจากประชากรปัจจุบัน

ขั้นตอนที่ 6 (การเกิดใหม่) ทำซ้ำขั้นตอนที่ 3 ถึง 5 จนกระทั่งบรรลุเงื่อนไขของการหยุด

3.8 ขั้นตอนวิธีการ GIDEON จัดอยู่ในกลุ่มของ GA สำหรับปัญหา VRPTW ที่มีเงื่อนไขของความสามารถในการบรรทุกสินค้า โดยใช้วิธีการในการแบ่งกลุ่มก่อนแล้วจึงจัดเส้นทาง ในกรณีนี้ GA ถูกประยุกต์ใช้ในเฟสของการแบ่งกลุ่ม เรียกว่า “การแบ่งกลุ่มแบบ GA” เป็นการแบ่งส่วนของปมออกเป็นส่วนๆ หรือแบ่งโดยมีศูนย์กลางอยู่ที่คลังสินค้า คล้ายกับ “ขั้นตอนวิธีการแบบกวาด (Sweep Algorithm)” (Thangiah, 1993); (Gendreau *et al.*, 1997)

3.9 ขั้นตอนวิธีการ GENEROUS จัดอยู่ในกลุ่มของ GA เช่นเดียวกับ GIDEON ใช้สำหรับปัญหา VRPTW ขั้นตอนวิธีการนี้หลีกเลี่ยงความยุ่งยากที่เกิดจากขั้นตอนการกำหนดรหัส (Encoding) ของผลเฉลยเพื่อเข้าสู่การแบ่งโครโมโซม โดยใช้วิธีการผสมยีนส์แบบข้ามฟาก (Crossover) และแบบผ่าเหล่า (Mutation) ของผลเฉลย โดยผลเฉลยใหม่จะถูกสร้างขึ้นจากประชากรรุ่นบรรพบุรุษ (Parent Population) จำนวน 2 ตัว เป็นการเชื่อมโยงระหว่างปมแรกบนเส้นทางของผลเฉลยรุ่นบรรพบุรุษตัวที่ 1 กับปมสุดท้ายของผลเฉลยรุ่นบรรพบุรุษตัวที่ 2 คล้ายกับขั้นตอนการวิธีการสลับสับเปลี่ยนแบบ 2-ออฟท์ เส้นทางใหม่จะถูกแทนที่ของเดิมในกลุ่มประชากร อันดับต่อมา คือ วิธีการผสมยีนส์แบบผ่าเหล่า ซึ่งสามารถสร้างได้โดยการสลับเป็นตรงกันข้ามของผลเฉลยรุ่นบรรพบุรุษ (Gendreau *et al.*, 1997)

3.10 ขั้นตอนวิธีการโครงข่ายนิวรอน (Neural Nets Algorithm) การใช้โครงข่ายนิวรอนเทียม ในการหาค่าเหมาะที่สุดสำหรับปัญหาเชิงการจัดนี้ เริ่มได้รับความสนใจอย่างแพร่หลายในระยะหลัง (Aarts and Lenstra, 1997); (Peterson, 1997) โครงข่ายนิวรอน ประกอบขึ้นจากโครงข่ายของปมซึ่งเชื่อมต่อกันด้วยค่าน้ำหนัก แต่ละปมเป็นตัวแทนในแต่ละหน่วยของการจัดคู่ ซึ่งสามารถสร้างขึ้นจากการคำนวณแบบง่ายๆ เกิดจากผลรวมของค่าน้ำหนัก และบวกด้วยค่าคงที่ที่เรียกว่า “Threshold” หรือ “Bias” สามารถประยุกต์ใช้กับฟังก์ชั่นไม่เชิงเส้น (Non-Linear Function) ได้ ผลการคำนวณจะทำให้เกิดหน่วยของเอาท์พุต ซึ่งจะใช้เป็นอินพุตสำหรับปมที่เชื่อมโยงต่อไปจนกระทั่งงานทั้งหมดของโครงข่ายได้รับการจัดรูปร่างจนสำเร็จ ด้วยความสัมพันธ์ของการทำงาน

ระหว่างอินพุตและเอาท์พุตนี้ ทำให้กระบวนการนี้ถูกเรียกว่า "กระบวนการการจัดการตัวเอง (Self-Organization Processes)"

3.11 ขั้นตอนวิธีการฝูงมด (Ant Colony Algorithm) เป็นการจำลองพฤติกรรมในการหาอาหารของมด นำเสนอไว้โดย Dorigo *et al.* (1996) ซึ่งเป็นเมต้าฮิวริสติกส์สำหรับปัญหาการหาค่าเหมาะที่สุดเชิงการจัดแบบยาก ที่เริ่มต้นมาจากปัญหา TSP โดยขั้นตอนวิธีการนี้ เป็นการสังเกตพฤติกรรมของฝูงมดในการค้นหาอาหาร ซึ่งมดจะมีการติดต่อสื่อสารส่งผ่านข้อมูลเกี่ยวกับแหล่งอาหารด้วยฟีโรโมน (Pheromone) โดยมดจะทำสัญลักษณ์ตามเส้นทางที่เดินผ่านด้วยฟีโรโมนนี้ ซึ่งจะมีจำนวนมากหรือน้อยขึ้นอยู่กับระยะทางและคุณภาพของแหล่งอาหารที่ค้นพบ โดยมดตัวอื่นๆ จะตามรอยฟีโรโมนมายังแหล่งอาหารในที่สุด จากพฤติกรรมของมดดังกล่าว ถูกนำมาสร้างเป็นขั้นตอนวิธีการในการแก้ปัญหาการหาเชิงการจัดเพื่อหาค่าเหมาะที่สุด ในการจำลองพฤติกรรมนี้ มดเทียมจะถูกสร้างขึ้นให้คล้ายคลึงกับสภาวะแวดล้อมในการค้นหาพื้นที่ของผลเฉลย วัตถุประสงค์คือ ค้นหาให้เจอแหล่งอาหารที่มีคุณภาพ และดัดแปลงความจำไว้ที่ฟีโรโมน โดยมดเทียมนี้สร้างขึ้นจากฟังก์ชั่นฮิวริสติกส์เฉพาะที่ เพื่อเป็นแนวทางในการค้นหาเซตผลเฉลยที่เป็นไปได้ต่อไป

จากการตรวจเอกสารที่เกี่ยวข้องกับขั้นตอนวิธีเมต้าฮิวริสติกส์พบว่า ยังมีวิธีอื่นอีกมายมายนอกเหนือจากที่ได้นำเสนอในหัวข้อที่ 3 นี้ แสดงให้เห็นว่าความนิยมใช้ขั้นตอนวิธีนี้ในการแก้ปัญหายังคงได้รับความสนใจอย่างกว้างขวาง มีการพัฒนาปรับปรุงให้มีประสิทธิภาพในการหาค่าเหมาะที่สุดอย่างต่อเนื่อง เช่น Taillard *et al.* (1997) ได้นำเสนอขั้นตอนวิธีการ TS สำหรับปัญหา VRPTW โดยทำการดัดแปลงปัญหา VRP แบบดั้งเดิมด้วยการเพิ่มเงื่อนไขของกรอบเวลา มีการเพิ่มค่าปรับสำหรับการส่งสินค้าไม่ทันกำหนด และใช้วิธีการสลับสับเปลี่ยนลำดับของปมลูกค้าระหว่างเส้นทางจำนวน 2 เส้น และสุดท้ายใช้ขั้นตอนการคัดเลือก (Selection Procedure) ในการค้นหาค่าเหมาะที่สุด

วิธี SA และ TS ได้ถูกพัฒนาขึ้นในช่วงแรกๆ เพื่อนำมาใช้ในการหาผลเฉลยของปัญหาการตัดสินใจที่มีตัวแปรและเงื่อนไขจำนวนมาก ทั้งวิธี SA และ TS มีจุดกำเนิดและแนวคิดแบบวิธีการหาผลเฉลยเฉพาะที่ ลักษณะเด่น 2 ประการ คือ การหลีกเลี่ยงปัญหาการตกอยู่ในผลเฉลยที่ไม่ขัดแย้งต่อเงื่อนไข แต่มีค่าฟังก์ชั่นวัตถุประสงค์ที่ไม่ดี (Poor Feasible Solution) หรือกล่าวอีกนัยหนึ่ง คือ ผลเฉลยที่ได้อยู่ห่างจากค่าเหมาะที่สุด ซึ่งเรียกปัญหานี้ว่าค่าเหมาะที่สุดเฉพาะที่ ซึ่งวิธี SA จะแก้ปัญหาโดยอนุญาตให้ขั้นตอนวิธีการออกจากจุดเหมาะที่สุดเฉพาะที่ ด้วยความน่าจะเป็น ส่วนวิธี TS ใช้หน่วยความจำทาบู เพื่อป้องกันไม่ให้ขั้นตอนวิธีการเลือกผลเฉลยที่ไม่ดีในรอบการ

กระทำซ้ำที่ผ่านมา ลักษณะเด่นประการที่ 2 คือ การลู่เข้าหาค่าเหมาะที่สุด (Convergence to Optimal Solution) สำหรับวิธี SA เมื่อจำนวนรอบการกระทำซ้ำในการหาผลเฉลยเพิ่มมากขึ้น ผลเฉลยที่ได้ในปัจจุบันจะค่อยๆลู่เข้าหาค่าเหมาะที่สุด โดยอาศัยพื้นฐานทฤษฎีความน่าจะเป็น ส่วนวิธี TS การลู่เข้าหาค่าเหมาะที่สุดกระทำได้โดยการกำหนดค่าคงที่ (Fix Value) ของตัวแปรที่เคยให้ผลเฉลยที่ดีไว้ในหน่วยความจำทาบู เมื่อขั้นตอนวิธีการเจอคำตอบที่ดีกว่า ตัวแปรที่เก็บค่าผลเฉลยที่ดีน้อยกว่าจะถูกลบออกจากหน่วยความจำ ทำให้ผลเฉลยที่ได้ ณ รอบการกระทำซ้ำปัจจุบันลู่เข้าหาค่าเหมาะที่สุด

วิธีเมต้าฮิวริสติกส์ที่ถูกพัฒนาขึ้นหลังจากสองวิธีแรก คือ ขั้นตอนวิธีการเชิงพันธุกรรม ซึ่งเป็นวิธีที่ได้รับความนิยมอย่างกว้างขวางในการหาผลเฉลยสำหรับปัญหาการตัดสินใจในหลายๆแขนงสาขาวิชา จากขั้นตอนที่ได้นำเสนอในเบื้องต้น สังเกตได้ว่าขั้นตอนวิธีการเชิงพันธุกรรมมีโครงสร้างและส่วนประกอบที่ยุ่งยากซับซ้อนกว่าในขณะที่แนวความคิดไม่แตกต่างเมื่อเปรียบเทียบกับวิธี SA และ TS แต่สุดท้ายแล้ว เมื่อแต่ละวิธีได้ถูกนำไปพัฒนาอย่างเต็มประสิทธิภาพ ผลเฉลยที่ได้จึงไม่แตกต่างกัน (ณกร, 2548)

สำหรับปัญหา VRPTW โดยทั่วไปมีรูปแบบเหมือน VRP แต่เพิ่มเงื่อนไขเกี่ยวกับระยะเวลาที่ลูกค้ากำหนดให้มีการส่งสินค้าภายในกรอบเวลา (Time Windows) ปัญหาลักษณะนี้ทั่วไปจัดเป็นปัญหาเชิงการจัด (Combinatorial Problem) โดยการจัดลูกค้าเข้าสู่เส้นทางการเดินรถ เพื่อให้เกิดต้นทุนต่ำสุด หรือเวลาเดินทางสั้นที่สุด ความซับซ้อนแบ่งออกเป็น 2 ส่วนที่สำคัญ ได้แก่ ความซับซ้อนในการสร้างแบบจำลองการตัดสินใจ (Decision Model) และความซับซ้อนของวิธีการหาผลเฉลย (Solution Method) สำหรับความซับซ้อนของการสร้างแบบจำลองการตัดสินใจนั้น คือการที่ปัญหาไม่สามารถเขียนให้อยู่ในรูปแบบจำลองเชิงคณิตศาสตร์อย่างง่าย สั้นและกะทัดรัดเพื่อให้ครอบคลุมฟังก์ชั่นวัตถุประสงค์ และเงื่อนไขทั้งหมดของปัญหาได้ หรือแม้ในบางครั้งอาจสามารถเขียนให้อยู่ในรูปแบบเชิงคณิตศาสตร์ได้ แต่ไม่สามารถใช้วิธีทางคณิตศาสตร์มาแก้ปัญหาได้ เนื่อง จากปัญหานี้เป็นการตัดสินใจแบบไม่ต่อเนื่อง (Discrete Decision Problem) ถึงแม้จะสร้างขั้นตอนวิธีทางคอมพิวเตอร์เพื่อใช้ในการหาผลเฉลย อาจต้องประสบปัญหาการใช้เวลาในการค้นหาผลเฉลย เมื่อลูกค้าและเงื่อนไขของปัญหามีเป็นจำนวนมาก ปัญหา VRPTW จัดเป็นปัญหาเอ็นพีฮาร์ด ที่ใช้เวลาในการหาผลเฉลยเพิ่มขึ้นแบบเลขชี้กำลัง (Exponential) หรือแบบแฟกทอเรียล (Factorial) ในกรณีที่สร้างแบบจำลองโดยใช้ตัวแปรตัดสินใจฐานสอง (Binary Variable) จำนวนตัวแปรเท่ากับ $N^2 \times K$ โดยที่ N คือ จำนวนลูกค้าและคลังสินค้า และ K คือ จำนวนเส้นทางหรือจำนวนรถขนส่ง เห็นได้ว่าผลเฉลยที่เป็นไปได้มีจำนวนมาก ซึ่งไม่สามารถหาผลเฉลยได้ด้วยวิธี

แม่นตรง งานวิจัยส่วนมากจึงนิยมใช้ขั้นตอนวิธีการฮิวริสติกส์ และเมต้าฮิวริสติกส์ ในการแทรกลูกค้าเข้าไปในเส้นทางการเดินรถ และจัดลำดับใหม่ให้ได้ผลเฉลยที่ดีขึ้น

Dondo *et al.* (2006) เป็นการแก้ปัญหา VRPTW เริ่มจากการหาผลเฉลยเริ่มต้น และใช้อนุกรมเชิงคณิตศาสตร์ในการหาผลเฉลยที่เหมาะสม โดยการสลับสับเปลี่ยนลูกค้าบนเส้นทางและจัดลำดับใหม่ การปรับปรุงผลเฉลยนี้อยู่บนพื้นฐานของโดเมนเวลาต่อเนื่อง (Continuous Time-Domain) ซึ่งจัดเป็นลักษณะของปัญหาการมอบหมายงาน และการจัดลำดับด้วยเซตของตัวแปรแบบไบนารี่

Solomon (1986) ได้ออกแบบและวิเคราะห์โครงสร้างของเส้นทางเดินรถที่มีลูกค้าจำนวนมาก ด้วยวิธีฮิวริสติกส์สำหรับปัญหา VRP และ Solomon (1987) ได้เสนอขั้นตอนวิธีการ PFIH สำหรับการหาผลเฉลยเริ่มต้นของปัญหา VRPTW ซึ่งเป็นวิธีที่ได้รับความนิยมและค่อนข้างมีประสิทธิภาพ ซึ่งในเวลาต่อมาวิธีดังกล่าวก็ได้รับการพัฒนา และนำไปใช้ร่วมกับวิธีการฮิวริสติกส์อื่นๆ ซึ่งมักเรียกกันว่าขั้นตอนวิธีการแบบผสมผสาน (Hybrid Algorithm) ซึ่งเป็นการรวมเอาวิธีการฮิวริสติกส์หลายๆ วิธีมาประยุกต์ใช้ร่วมกันในการหาผลเฉลยที่เหมาะสมที่สุด ซึ่งขั้นตอนวิธีการที่มักถูกนำมาผสมผสานกันนี้ ประกอบด้วย GA, SA และ TS วิธีการเริ่มจากการหาผลเฉลยเริ่มต้นด้วยวิธี PFIH และปรับปรุงผลเฉลยโดยวิธีการสืบค้นเฉพาะที่ (Local Search Descent Method)

Thangiah (1999) ได้นำขั้นตอนวิธีการดังกล่าวสำหรับการแก้ปัญหาที่มีลูกค้าตั้งแต่ 100 - 417 ปม ซึ่งนับเป็นงานวิจัยที่เป็นต้นแบบของการพัฒนาขั้นตอนวิธีการแบบผสมผสาน ทั้งวิธีการแบบฮิวริสติกส์และเมต้าฮิวริสติกส์ ในการหาผลเฉลยของปัญหาเชิงการจัดที่มีขนาดใหญ่ โดยนับตั้งแต่ปี ค.ศ. 2000 เป็นต้นมา นักวิจัยหันมามุ่งเน้นพัฒนาขั้นตอนวิธีการเมต้าฮิวริสติกส์ให้มีประสิทธิภาพมากยิ่งขึ้น ทำให้วิธีการแก้ปัญหาด้วยวิธีแม่นตรง ได้รับความนิยมน้อยลงตามลำดับอันเนื่องมาจากความซับซ้อนของปัญหาที่มีตัวแปรจำนวนมาก ทำให้ประสิทธิภาพของวิธีแม่นตรงลดลง จนไม่อาจหาผลเฉลยที่เหมาะสมที่สุด หรือผลเฉลยเฉพาะที่ได้ Beatrice *et al.* (2006) ได้เสนองานวิจัยเกี่ยวกับปัญหา VRPTW แบบมีหลายวัตถุประสงค์ (Multi-Objective Problem) โดยการหาจำนวนรถขนส่งน้อยที่สุด และผลรวมต้นทุนในการเดินทางขนส่ง หรือระยะทางสั้นที่สุด โดยใช้ GA และเทคนิคการเรียงลำดับแบบพาเรโต้ (Pareto Ranking) ซึ่งสามารถใช้แก้ปัญหาดังกล่าวได้ค่อนข้างมีประสิทธิภาพ วิธีการดังกล่าวสามารถประยุกต์ใช้ได้กับปัญหา VRPTW ที่มีจำนวนลูกค้ามากกว่า 100 ปม

การหาค่าเหมาะที่สุดสำหรับปัญหา VRP ที่ประกอบด้วยค่าพารามิเตอร์ที่คงที่ เรียกว่า ปัญหาเชิงกำหนด ความซับซ้อนของปัญหาจะเพิ่มเป็นเท่าทวีคูณ เมื่อเกิดค่าพารามิเตอร์ที่ไม่คงที่ หรือเรียกว่า ปัญหาเชิงสโทแคสติก สำหรับปัญหาการตัดสินใจภายใต้ความไม่แน่นอนนี้เริ่มขึ้นใน ปี 1955 โดย Dantzig (1955) และมีการพัฒนาต่อโดย Charnes and Cooper (1959) ซึ่งทั้ง 3 นักวิจัย ถือได้ว่าเป็นผู้ริเริ่มสร้างรากฐานของกำหนดการเชิงสโทแคสติก และการหาค่าเหมาะที่สุดภายใต้ เงื่อนไขความน่าจะเป็น (Optimization Under Probabilistic Constraints) ในปัจจุบันหาก เปรียบเทียบกันระหว่างกำหนดการเชิงกำหนดและกำหนดการเชิงสโทแคสติกแล้ว ปัญหาเชิงสโท แคสติกค่อนข้างได้รับการยอมรับในแง่ของความเสมือนจริงมากกว่า เนื่องจากหากพิจารณาระบบ ใดๆ อย่างถี่ถ้วน จะพบว่าทุกระบบย่อมเกิดความไม่แน่นอน (Uncertainty) อันเนื่องมาจากสาเหตุ ปัจจัยทั้งที่วัดได้ และวัดไม่ได้ ดังนั้นในระยะหลังงานวิจัยที่เกี่ยวข้องกับกำหนดการเชิงสโทแคสติก จึงเกิดขึ้นอย่างมากมาย งานวิจัยทางด้านสโทแคสติกที่ได้รับการเผยแพร่ในปัจจุบันมีจำนวนมากถึง 4,791 ฉบับ (Committee on Stochastic Programming, 2008) ซึ่งในจำนวนนั้นรวมถึงงานวิจัย ทางด้าน VRP ที่มีความไม่แน่นอนรวมอยู่ด้วย จะเห็นได้ว่าวิธีการแบบสโทแคสติกสามารถพัฒนา ได้อย่างมีประสิทธิภาพ เมื่อความไม่แน่นอนของข้อมูลสามารถแสดงในรูปแบบของการแจกแจง หรือความน่าจะเป็นที่ถูกต้องชัดเจน และเป็นตัวแทนของระบบจริงได้

ปัญหา VRP ที่มีความไม่แน่นอนของพารามิเตอร์ต่างๆ หรือเรียกว่า SVRP ซึ่งปรับปรุงมา จากปัญหา VRP เชิงกำหนด (Deterministic VRP) (Titiporn, 2001) เพื่อทำให้ปัญหามีความเสมือน จริงมากขึ้น เนื่องจากปัจจุบันเงื่อนไขต่างๆ มีความไม่แน่นอน เช่น ความต้องการของลูกค้า ระยะทางที่ใช้ในการเดินทาง ระยะเวลาในการให้บริการ เป็นต้น ปัญหา SVRP สามารถแสดงได้ใน แง่มุมที่แตกต่างกัน โดยส่วนใหญ่มักแบ่งเป็นดังนี้

1. VRP แบบมีลูกค้าไม่แน่นอน (VRP with Stochastic Customer) และ VRP แบบมีความต้องการ ไม่แน่นอน (VRP with Stochastic Demands)

ซึ่งเป็นปัญหาที่มีลูกค้าและความต้องการไม่แน่นอน มีงานวิจัยเกิดขึ้นในส่วนนี้อย่าง กว้างขวาง โดยมีเงื่อนไข คือ

1.1 ความต้องการของลูกค้าเป็นตัวแปรเชิงสุ่มซึ่งทราบการแจกแจง
1.2 เส้นทางต้องถูกออกแบบก่อนที่จะทราบความต้องการที่แท้จริง
1.3 วัตถุประสงค์ของปัญหาคือ ต้องการหาระยะทางขนส่งคาดหวังต่ำสุด

2. VRP แบบมีเวลาเดินทางไม่แน่นอน (VRP with Stochastic Travel Times) และ VRP แบบมีระยะเวลาในการให้บริการไม่แน่นอน (VRP with Stochastic Service Time)

ในกรณีนี้เกี่ยวข้องกับเวลาในการขนส่งและเวลาในการให้บริการที่ไม่แน่นอน วิธีการสร้างรูปแบบของปัญหา SVRP จัดอยู่ในประเภทของกำหนดการเชิงสโทแคสติก (Stochastic Programming) ซึ่งอาจแบ่งได้เป็นกำหนดการเชิงโอกาส (Chance Constrained Program; CCP) และกำหนดการเชิงสโทแคสติกแบบรีคอร์ส (Stochastic Program with Recourse; SPR) ส่วนหลักการอื่นๆ ในงานวิจัยปัจจุบัน ประกอบด้วย กระบวนการตัดสินใจแบบมาร์คอฟ (Markov Decision Process) และกำหนดการเชิงพลวัต (Dynamic Programming) เป็นต้น เทคนิคการแก้ปัญหา SVRP จะแตกต่างกันไปตามวิธีการสร้างรูปแบบของปัญหา โดยทั่วไปมีลักษณะคล้ายคลึงกับเทคนิคการแก้ปัญหา VRP ได้แก่ วิธีการแม่นตรง วิธีการฮิวริสติกส์ และเมต้าฮิวริสติกส์ โดยวิธีการแม่นตรงที่ประยุกต์ใช้กับปัญหา SVRP ประกอบด้วย วิธีการขยายและจำกัดเขต วิธีการขยายและตัดส่วน (Branch and Cut) วิธีการจำนวนเต็มแบบแอลเชฟ (Integer L-Shape Method) และกำหนดการเชิงพลวัตทั่วไป (Generalized Dynamic Programming) ส่วนวิธีการฮิวริสติกส์และเมต้าฮิวริสติกส์ส่วนมากที่นักวิจัยนิยมนำมาใช้แก้ปัญหา SVRP ได้แก่ ขั้นตอนวิธีการเซฟวิ่ง และ TS เป็นต้น

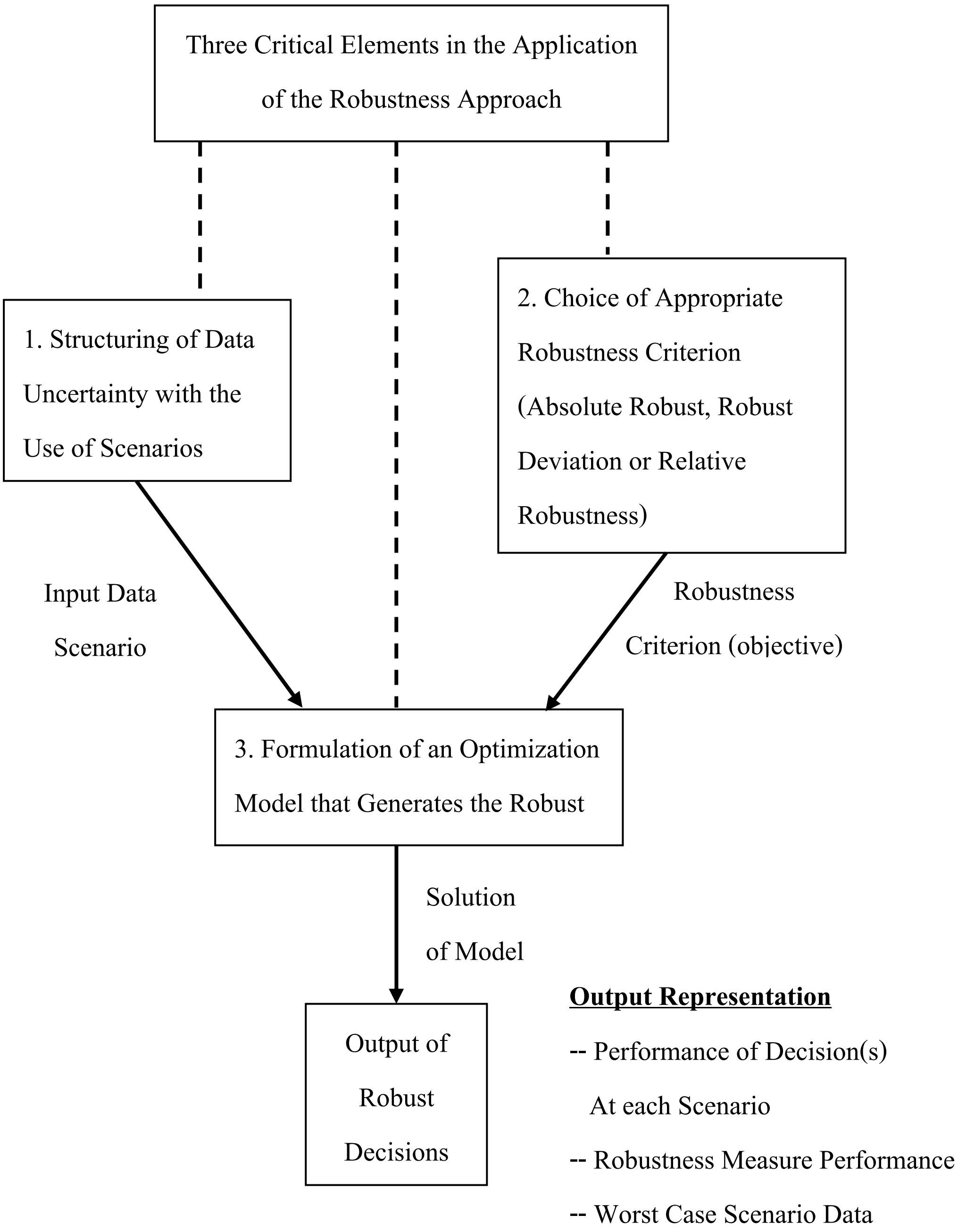
ปัญหา SVRP เป็นการผสมผสานคุณลักษณะของสโทแคสติก (Stochastic) และกำหนดการเชิงจำนวนเต็ม (Integer Programming) ซึ่งจัดเป็นปัญหาที่มีการคำนวณยุ่งยากและซับซ้อน (Gendreu *et al.*, 1996) การแก้ปัญหา SVRP ส่วนใหญ่จึงใช้วิธีการแก้ปัญหาด้วยวิธีการแบบสองขั้นตอน (Two-Stage) (Zhihong *et al.*, 2006) ในงานวิจัยที่พิจารณาปัญหา SVRP กรณีที่มีความต้องการไม่แน่นอนและทราบค่าการแจกแจงเป็นแบบปัวซอง หรือเรียกว่า “CVRPSD” (Tillman, 1969) ใช้วิธีการแก้ปัญหาที่ปรับปรุงมาจากขั้นตอนวิธีการเซฟวิ่ง นอกจากนี้ในปีต่อๆ มา งานวิจัยนี้ยังได้รับการพัฒนาขึ้นไปอีกหลายขั้น ซึ่งวิธีการที่นิยมใช้กัน คือ กำหนดการเชิงโอกาส และกำหนดการเชิงสโทแคสติกแบบสองขั้นตอน (Two-Stage Stochastic Programming) (Gendreau *et al.*, 1996)

ในงานวิจัยนี้สนใจปัญหา VRP แบบมีเวลาเดินทางไม่แน่นอน หรือ VRPSTT โดย Kao (1978) เป็นนักวิจัยคนแรกที่นำเสนอวิธีการแก้ปัญหา TSP แบบมีเวลาเดินทางไม่แน่นอน ด้วยวิธีการฮิวริสติกส์บนพื้นฐานของกำหนดการเชิงพลวัต ประมาณ 10 ปีต่อมา Carraway *et al.* (1989) ใช้วิธีกำหนดการเชิงพลวัตทั่วไปในการแก้ปัญหา TSPSTT และ Laporte *et al.* (1992) ได้เสนองานวิจัยสำหรับปัญหา VRP ที่มีระยะเวลาให้บริการและเวลาเดินทางเป็นแบบสโทแคสติก โดยคณะผู้วิจัยชุดนี้เสนอวิธีการแก้ปัญหา VRPSST โดยใช้วิธีการเข้าสู่ผลเฉลยแบบกำหนดการเชิง

โอกาส และขั้นตอนวิธีการขยายและตัดส่วนในการแก้ปัญหา ส่วนงานวิจัยของ Jula *et al.* (2005) ได้มีการพัฒนาวิธีการประมาณเวลาในการเข้าสู่ผลเฉลยสำหรับปัญหา VRPTW และใช้กำหนดการเชิงพลวัตในการหาค่าเหมาะที่สุด

สังเกตได้ว่า ข้อเสียที่ชัดเจนของกำหนดการเชิงสโทแคสติก คือ ความยุ่งยากซับซ้อน ทั้งโครงสร้าง และขั้นตอนวิธีการในการแก้ปัญหา ในขณะที่ผลเฉลยขึ้นอยู่กับข้อมูลที่มีความไม่แน่นอน ที่สามารถประมาณด้วยตัวแปรเชิงสุ่มซึ่งทราบการแจกแจง (Distribution) หรือความน่าจะเป็นในการเกิด (Probability) ส่วนผลเฉลยหรือค่าเหมาะที่สุดที่ได้กลับเป็นเพียงแค่ค่าคาดหวังเท่านั้น ผลเฉลยที่ได้จะยังคงมีประสิทธิภาพก็ต่อเมื่อข้อมูลที่เกิดขึ้นจริงเป็นข้อมูลที่เกิดขึ้นบ่อยครั้ง (Most Likely) นอกเหนือจากนี้ผลเฉลยที่ได้จะมีประสิทธิภาพด้อยลง และส่วนมากผลเฉลยที่ได้จากกำหนดการเชิงสโทแคสติกมักใช้เป็นเพียงการวางแผนก่อนหน้า (Pre-Planning) เท่านั้น เมื่อสถานการณ์เกิดขึ้นจริง ผลเฉลยที่ได้จากการวางแผนก่อนหน้าอาจต้องมีการปรับปรุงเพื่อให้ได้ผลเฉลยที่เหมาะสมยิ่งขึ้น

ในแง่มุมของงานวิจัยนี้ มุ่งเน้นการหาผลเฉลยที่มีสภาพทนทาน (Robustness) ต่อความไม่แน่นอน และด้วยรูปแบบปัญหา VRPTW เป็นลักษณะของปัญหาแบบไม่ต่อเนื่อง การตรวจเอกสารในลำดับถัดไปจึงมุ่งเน้นการศึกษาขั้นตอนวิธีการเข้าสู่ค่าเหมาะที่สุดเชิงทนทานแบบไม่ต่อเนื่อง (Robust Discrete Optimization) ปัจจุบันนักวิจัยเริ่มหันมาสนใจในการศึกษาปัญหาและพัฒนาเครื่องมือในการแก้ปัญหาเพื่อหาผลเฉลยที่มีสภาพทนทานต่อความไม่แน่นอน หรือการหาค่าเหมาะที่สุดในสถานการณ์ที่แย่ที่สุด Kouvelis and Yu (1977) ได้อธิบายกรอบการทำงานของการตัดสินใจเชิงทนทาน (The Robust Decision Making Framework) และการประยุกต์ใช้การเข้าสู่ผลเฉลยเชิงสภาพทนทานอันประกอบด้วย 3 ส่วนหลักๆ คือ การสร้างข้อมูลที่ไม่แน่นอนด้วยสถานการณ์ต่างๆ (Structuring of Data Uncertainty with the Use of Scenarios) การเลือกเกณฑ์การตัดสินผลเฉลยเชิงสภาพทนทานที่เหมาะสม (Choice of Appropriate Robustness Criterion) และการสร้างรูปแบบการหาค่าเหมาะที่สุดเพื่อการตัดสินใจเชิงทนทาน (Formulation of an Optimization Model that Generates the Robust Decisions) แสดงดังภาพที่ 3



ภาพที่ 3 กรอบการทำงานของการตัดสินใจเชิงทนทาน (The Robust Decision Making Framework)

ที่มา: Kouvelis and Yu (1977)

จากภาพที่ 3 แสดงให้เห็นว่าการสร้างรูปแบบการหาค่าเหมาะที่สุดเชิงทนทาน ไม่จำเป็นต้องทราบค่าความไม่แน่นอน หรือรูปแบบการแจกแจงของข้อมูลที่มีความไม่แน่นอน ซึ่งลดความยุ่งยากในขั้นตอนการเก็บรวบรวมข้อมูลและการวิเคราะห์ข้อมูลเชิงสถิติ อีกทั้งเป็นการง่ายต่อการประยุกต์ใช้งานจริง โดยผู้ตัดสินใจส่วนมากมักสามารถคาดเดาเหตุการณ์ที่เกิดขึ้นจริง ทั้งในปัจจุบัน และอนาคตได้ด้วยประสบการณ์ หรือการเรียนรู้จากข้อมูลที่เกิดขึ้นแล้ว และถ่ายทอดออกมาในรูปแบบของสถานการณ์ต่างๆ ได้ (Scenarios)

ช่วงกลางปี ค.ศ. 1990 ทีมนักวิจัยนำโดย Ben-Tal and Nemirovski (1998, 1999, 2000) ร่วมด้วย El-Ghaoui and Lebret (1997) พยายามคิดค้นการสร้างค่าความไม่แน่นอนของข้อมูลให้อยู่ในรูปเซตทรงรี (Ellipsoidal Set) และขจัดปัญหาในการเลือกผลเฉลยด้วยค่าคาดหวัง โดยใช้ค่าพารามิเตอร์ของความไม่แน่นอนที่เลวร้ายที่สุด (Worst-Case Value of Parameters) ซึ่งผลเฉลยที่ได้สอดคล้องกับการเข้าสู่ผลเฉลยเชิงสภาพทนทาน แต่เมื่อพิจารณาเทียบกับค่าเหมาะที่สุดแล้ว ผลเฉลยที่ได้ค่อนข้างห่างจากค่าเหมาะที่สุดมากเกินไป (Overly Conservative Solution) หากผู้ประกอบการนำผลเฉลยไปใช้จริงอาจเกิดต้นทุนสูงเกินกว่าเหตุ เพราะการเกิดสถานการณ์ที่เลวร้ายที่สุด (Worst-Case Scenario) เกิดขึ้นไม่บ่อยครั้งนัก หรืออาจไม่เกิดขึ้นเลยก็เป็นได้ เพราะฉะนั้นวิธีการแก้ปัญหาการเกิดผลเฉลยที่ห่างจากค่าเหมาะที่สุดมากเกินไป คือ การสร้างสถานการณ์ขึ้นทีละหลายๆ สถานการณ์ โดยแต่ละสถานการณ์มีค่าพารามิเตอร์ของความไม่แน่นอนที่แตกต่างกันไป ทั้งนี้ผู้ที่สามารถสร้างสถานการณ์ได้ดี และเหมาะสมที่สุด คือ ผู้ตัดสินใจที่มีประสบการณ์ในงานนั้นๆ และทราบเหตุการณ์ต่างๆ ที่เกิดขึ้นในระบบงานจริงเป็นอย่างดี วิธีการนี้เรียกว่า วิธีการสร้างสถานการณ์ (Scenario-Based Approach) ซึ่งเหมาะสำหรับองค์กรขนาดใหญ่ (Kouvelis and Yu, 1977)

วิธีการสร้างสถานการณ์ เป็นส่วนหนึ่งในการสร้างข้อมูลที่มีความไม่แน่นอน สำหรับการตัดสินใจเชิงทนทาน ที่นำเสนอไว้โดย Kouvelis and Yu (1977) วิธีการนี้แสดงค่าความไม่แน่นอนได้ภายใต้สถานการณ์ต่างๆ ที่เปลี่ยนไป เช่น ความไม่แน่นอนของดอกเบี้ยภายใต้สภาวะเศรษฐกิจ ความไม่แน่นอนของกำลังการผลิตภายใต้สภาวะการเจริญเติบโตของเทคโนโลยี หรือความไม่แน่นอนของผลกำไรภายใต้สภาวะการบริหารงานธุรกิจ เป็นต้น เห็นได้ว่าการสร้างข้อมูลที่มีความไม่แน่นอนลักษณะนี้ ไม่จำเป็นต้องทราบค่าความน่าจะเป็นในการเกิดข้อมูลแต่ละตัว เว้นแต่ว่าผู้ตัดสินใจต้องการทราบความน่าจะเป็นในการเกิดสถานการณ์แต่ละสถานการณ์ วิธีการสร้างสถานการณ์ เริ่มต้นจากผู้ตัดสินใจต้องแยกแยะปัจจัยหลักที่ส่งผลกระทบต่อความไม่แน่นอนของข้อมูล อธิบายความสัมพันธ์ของปัจจัยเมื่อป้อนข้อมูลเข้าสู่รูปแบบการตัดสินใจ ซึ่งส่วนมากมักเกิด

จากความรู้และประสบการณ์ของผู้ตัดสินใจที่มีต่อระบบงานนั้นๆ ตัวอย่างเช่น การสร้างสถานการณ์เวลากระบวนการผลิต (Processing Times) สำหรับงานทุกงานที่ผลิตโดยเครื่องจักร ปัจจัยที่ส่งผลกระทบต่อความไม่แน่นอนของเวลากระบวนการผลิต ได้แก่ สภาพเครื่องมือเครื่องจักร เวลาในการรับงาน ประสบการณ์ของพนักงานเครื่องจักร หรือเทคโนโลยีในการผลิต เป็นต้น ซึ่งปัจจัยเหล่านี้สามารถนำมาสร้างเป็นสถานการณ์ต่างๆ เพื่อหาเวลากระบวนการผลิต เมื่อสถานการณ์นั้นๆ เกิดขึ้นได้

Herrmann (1999) ประยุกต์ใช้วิธีการสร้างสถานการณ์ในการสร้างเวลากระบวนการผลิตที่ไม่แน่นอน สำหรับปัญหาการจัดตารางเครื่องจักรแบบขนาน (Parallel Machine Scheduling) และใช้วิธี GA แบบ 2 พื้นที่ (Two-Space Genetic Algorithm) ในการหาผลเฉลยที่มีความทนทานต่อความไม่แน่นอน โดยพื้นที่ที่ 1 ใช้สำหรับสร้างกลุ่มประชากรของผลเฉลย และพื้นที่ที่ 2 ใช้สำหรับสร้างกลุ่มประชากรของสถานการณ์ ซึ่งเป็นขั้นตอนวิธีการที่ค่อนข้างมีประสิทธิภาพในการหาผลเฉลยเชิงสภาพทนทาน

ต่อมา Montemanni *et al.* (2005) ได้นำเสนอขั้นตอนวิธีการแก้ปัญหา TSP ที่มีเวลาเดินทางเป็นแบบช่วง (Interval Time) โดยใช้เกณฑ์การตัดสินเชิงความแตกต่างของความทนทาน (Robust Deviation Criteria) และใช้ขั้นตอนวิธีการแม่นตรงในการแก้ปัญหา ซึ่งในงานวิจัยนี้เหมาะสมสำหรับปัญหาขนาดเล็ก หากปัญหามีขนาดใหญ่ขึ้นอาจต้องมีการดัดแปลงแก้ไขขั้นตอนวิธีการให้เหมาะสม มิฉะนั้นผลเฉลยที่ได้อาจมีคุณภาพต่ำ หรือไม่สามารถหาผลเฉลยได้เลย

Ordonez *et al.* (2006) นำเสนอวิธีเข้าสู่ค่าเหมาะที่สุดเชิงทนทานสำหรับปัญหา CVRP ที่มีความต้องการสินค้าไม่แน่นอน งานวิจัยนี้แสดงความไม่แน่นอนของความต้องการสินค้าด้วยเซตแบบมีขอบเขต (Bounded Set) ประกอบด้วย ช่วงคอนเวกซ์ (Convex Hull) สี่เหลี่ยม (Box) และทรงรี (Ellipsoidal) โดยใช้วิธีแม่นตรงในการหาผลเฉลย จากการประมาณค่าความไม่แน่นอนด้วยเซตแบบมีขอบเขตนี้ ค่อนข้างจำกัดการประยุกต์ใช้สำหรับนักวิชาการ ไม่ใช่ผู้ตัดสินใจทั่วไป เนื่องจากเป็นสูตรทางคณิตศาสตร์ที่ค่อนข้างเข้าใจยาก นอกจากนี้การแก้ปัญหาด้วยวิธีแม่นตรงเหมาะสำหรับปัญหาที่มีขนาดเล็ก จึงไม่เหมาะสำหรับการนำมาประยุกต์ใช้กับปัญหาที่มีขนาดใหญ่

จุดเด่นของวิธีการเข้าสู่ผลเฉลยเชิงทนทาน เมื่อเทียบกับวิธีการเชิงกำหนด และวิธีการเชิงสโทแคสติก นอกจากการคำนวณที่ไม่ยุ่งยากซับซ้อนแล้ว ยังสามารถให้ผลเฉลยที่ครอบคลุมและทนทานต่อค่าความไม่แน่นอนได้มากกว่า การหาผลเฉลยแต่ละครั้งยังคงมีประสิทธิภาพในการ

วางแผนระยะยาว (Long-Term Planning) ไม่ใช่เป็นเพียงการวางแผนก่อนหน้าเท่านั้น ผลเฉลยจะเปลี่ยนแปลงไปก็ต่อเมื่อผู้ตัดสินใจมีการปรับปรุง หรือเพิ่มเติมสถานการณ์เข้าสู่เซตของความไม่แน่นอน หากผู้ตัดสินมั่นใจว่า ได้สร้างสถานการณ์ทุกอย่างได้ครอบคลุมระบบงานจริงแล้ว การประกันคุณภาพของผลเฉลยจะครอบคลุมสถานการณ์นั้นๆ เช่นกัน

งานวิจัยนี้ มุ่งเน้นการแก้ปัญหาความไม่แน่นอนของปัญหา VRPTW โดยอาศัยหลักการที่สามารถนำไปปฏิบัติได้จริง และลดความยุ่งยากซับซ้อนในขั้นตอนการสร้างข้อมูลที่มีความไม่แน่นอนด้วยวิธีการสร้างสถานการณ์ ที่ไม่จำเป็นต้องประมาณด้วยรูปแบบการแจกแจง หรือการหาค่าความน่าจะเป็นของข้อมูล และหาผลเฉลยของปัญหาด้วยขั้นตอนวิธีการแบบผสมผสาน ซึ่งเป็นการรวมเอาวิธีฮิวริสติกส์และเมต้าฮิวริสติกส์มากกว่า 1 ประเภท มาใช้ร่วมกันเพื่อหาผลเฉลยที่มีคุณภาพ ขั้นตอนวิธีการแบบผสมผสานนี้ ได้พยายามนำเอาจุดเด่นของแต่ละวิธีมาใช้ร่วมกัน และใช้วิธีการหนึ่งกำจัดจุดด้อยของอีกวิธีการหนึ่ง ซึ่งรายละเอียดได้นำเสนอไว้ในหัวข้อวิธีการ

อุปกรณ์และวิธีการ

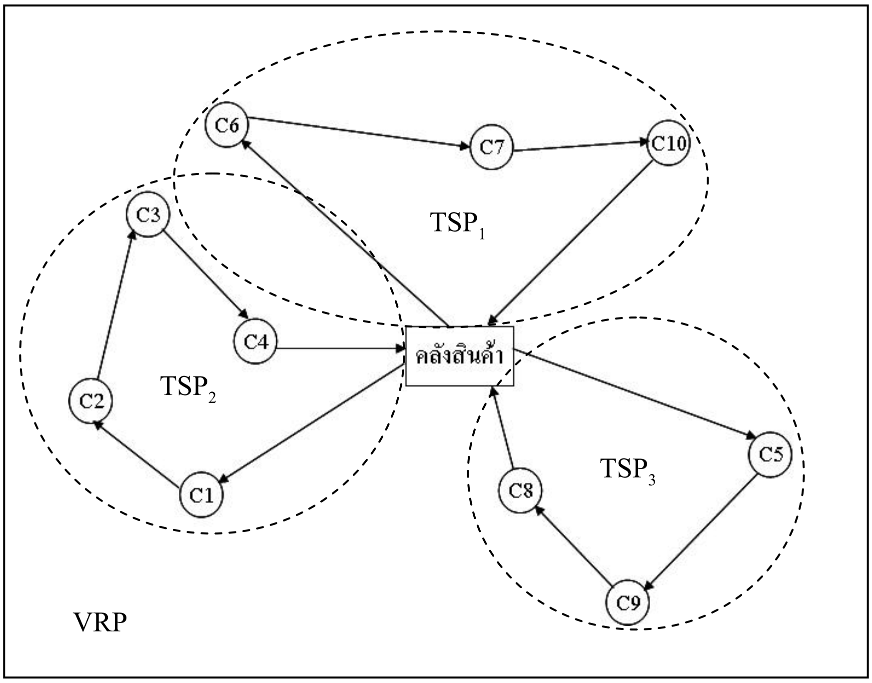
อุปกรณ์

1. คอมพิวเตอร์ส่วนตัว (Personal Computer) จำนวน 5 เครื่อง ประกอบด้วย
   1.1 Pentium4 3.40 GHz. หน่วยความจำ 512 Mb. ความจุฮาร์ดดิสค์ 80 Gb.
   1.2 Dual Core E2180 2.00 GHz. หน่วยความจำ 1.00 Gb. ความจุฮาร์ดดิสค์ 80 Gb.
   1.3 Dual Core D805 2.66 GHz. หน่วยความจำ 2.00 Gb. ความจุฮาร์ดดิสค์ 80 Gb.
   1.4 Core2 Quad Q6600 2.40 GHz. หน่วยความจำ 3.25 Gb. ความจุฮาร์ดดิสค์ 250 Gb.
   1.5 Dual Core D925 3.00 GHz. หน่วยความจำ 1.00 Gb. ความจุฮาร์ดดิสค์ 80 Gb.
2. เครื่องคอมพิวเตอร์โน๊ตบุ๊ค (Notebook Computer) Core 2 T7300 2.00 GHz. หน่วยความจำ 2.00 Gb. ความจุฮาร์ดดิสค์ 160 Gb.
3. โปรแกรม MATLAB
4. โปรแกรม MS-EXCEL

วิธีการ

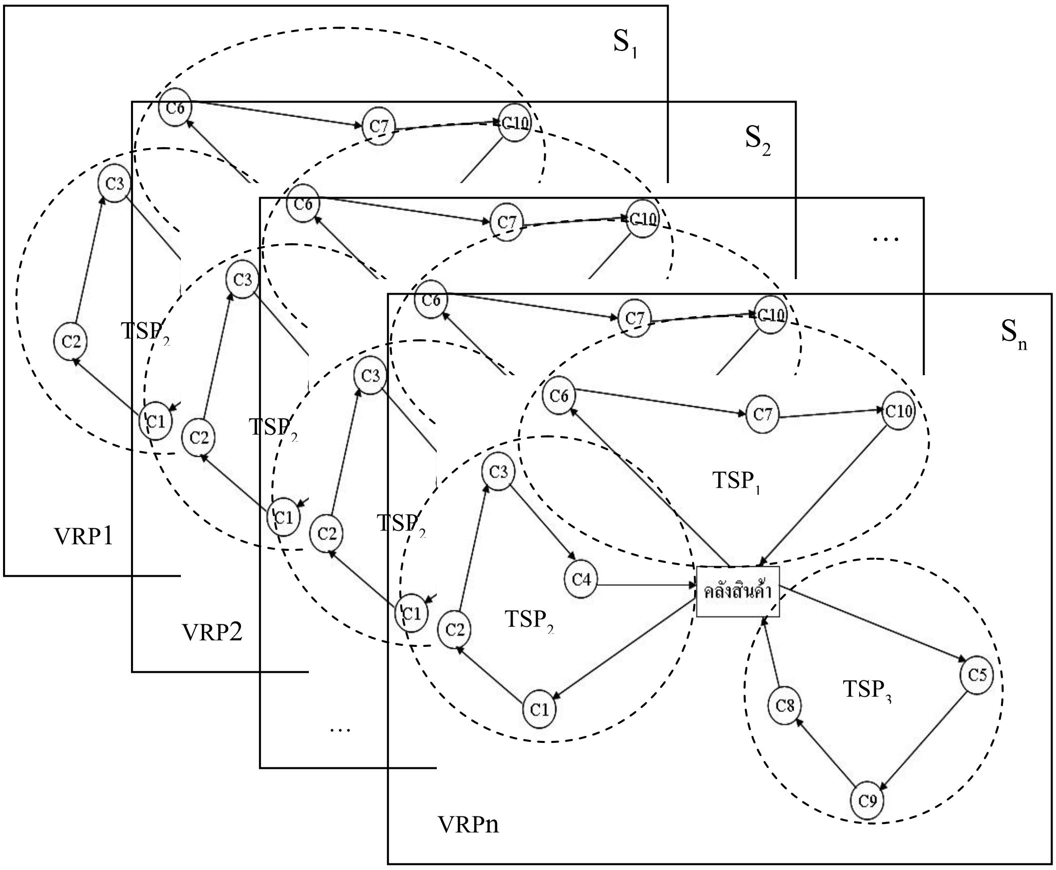
จากการตรวจเอกสารเป็นการยืนยันโดยไม่ต้องพิสูจน์ว่า ปัญหา VRPTW เป็นปัญหาแบบเอ็นพี-สมบูรณ์ ซึ่งหมายความว่า ไม่มีใครในโลกนี้สามารถคิดค้นขั้นตอนวิธีการที่มีประสิทธิภาพในการหาค่าเหมาะที่สุดได้อย่างแน่นอน จากทฤษฎีความยากและความซับซ้อนในการหาค่าเหมาะที่สุดนั้น คือ การประมาณค่าความสามารถในการประมวลผลด้วยคอมพิวเตอร์ในกรณีที่เลวร้ายที่สุด (Worst Case) เป็นตัวบ่งชี้ โดยพิจารณาจากอัตราการเพิ่มขึ้นของเวลาที่ใช้ในการหาผลเฉลยเมื่อ n มีขนาดเพิ่มขึ้น ในกรณีปัญหา TSP หรือ VRP กำหนดให้ n คือ จำนวนลูกค้าที่ต้องให้บริการในทางเทคนิคจะกำหนดอัตราการเพิ่มขึ้นของเวลาในการหาผลเฉลยเป็นแบบลำดับ (Order) n ตัวอย่างเช่น $O(\log n) < O(n) < O(n \log n) < O(n^2) < O(n^3) < O(2^n) < O(n!)$ และสำหรับปัญหา TSP ได้มีการพิสูจน์ทางทฤษฎีแล้วว่าเป็นปัญหาเอ็นพี-สมบูรณ์ และในกรณีเลวร้ายที่สุด อัตราการเพิ่มขึ้นของเวลาในการหาผลเฉลยด้วยคอมพิวเตอร์เป็น $O(n!)$ (Savelsbergh, 1985) หมายความว่า หากปัญหามีจำนวน $n = 10$ และใช้เวลาในการประมวลผลเพียง 1 วินาที ในการหาค่าเหมาะที่สุด ในกรณีนี้เมื่อจำนวน n เพิ่มขึ้นเป็น $n = 20$ เวลาที่ใช้ คือ $\frac{20!}{10!}$ เท่ากับ

ประมาณ 20000 ปี ดังนั้นจึงมีความจำเป็นอย่างยิ่งในการพัฒนาขั้นตอนวิธีการที่มีประสิทธิภาพในการหาผลเฉลยแค่ดีพอเพียง หรือค่าเหมาะที่สุดใกล้เคียง (Near Optimum Value) ซึ่งเรียกว่าวิธีฮิวริสติกส์ ถึงแม้วิธีนี้จะไม่สามารถรับประกันผลเฉลยว่าจะเป็นค่าเหมาะที่สุดได้ หรือไม่สามารถรับประกันได้ว่า ทุกๆ ครั้งที่คอมพิวเตอร์ประมวลผลจะให้ผลเฉลยที่เท่ากันทุกครั้ง โดยแนวคิดหลักๆ ของวิธีฮิวริสติกส์ คือ การเข้าใจปัญหาการตัดสินใจนั้นๆ อย่างลึกซึ้ง และพยายามค้นหาเส้นทางลัดในลู่เข้าสู่ค่าเหมาะที่สุด หรือค่าเหมาะที่สุดใกล้เคียงในเวลาอันรวดเร็ว ส่วนปัญหา VRP นั้นยังไม่พบผู้พิสูจน์ทางทฤษฎี อาจเป็นเพราะปัญหา VRP เป็นปัญหาที่ขยายมาจากปัญหา TSP หรือ ปัญหา VRP สามารถเทียบได้กับปัญหา TSP ในกรณีที่มีพนักงานขายหลายคน หรือรถขนส่งหลายคันนั่นเอง



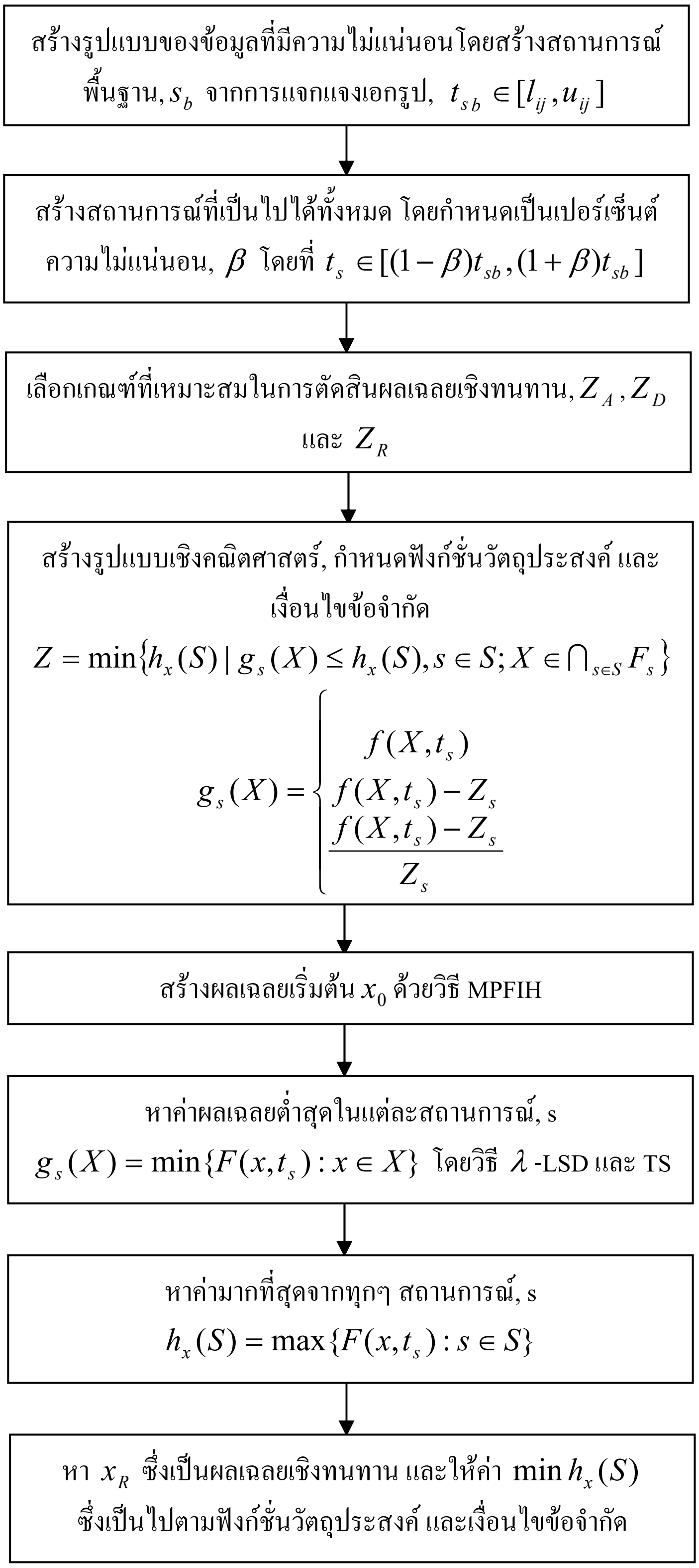
ภาพที่ 4 ความซับซ้อนของปัญหา VRP

จากภาพที่ 4 แสดงให้เห็นอย่างชัดเจนว่าปัญหา TSP เป็นเพียงเซตย่อยของปัญหา VRP เพราะฉะนั้นเมื่อกำหนดให้ V แทนจำนวนรถขนส่งในปัญหา VRP อัตราการเพิ่มขึ้นของเวลาในการหาผลเฉลยด้วยคอมพิวเตอร์จึงเป็น $O(Vn!)$ สำหรับงานวิจัยนี้ เพิ่มความซับซ้อนของปัญหาให้เป็นปัญหา VRPTW ที่มีความไม่แน่นอน ความยากและความซับซ้อนของปัญหายิ่งเพิ่มเป็นเท่าทวีคูณ เมื่อการแก้ปัญหาจะต้องทำการประมวลผลปัญหา VRP เป็นจำนวนรอบเท่ากับสถานการณ์, S ที่กำหนด



ภาพที่ 5 ความซับซ้อนของปัญหา VRP ที่มีความไม่แน่นอน

จากภาพที่ 5 อัตราการเพิ่มขึ้นของเวลาในการหาผลเฉลยด้วยคอมพิวเตอร์เท่ากับ $O(Vn!|S|)$ เมื่อ $|S|$ คือ ขนาดของสถานการณ์ที่กำหนด ดังนั้นในงานวิจัยนี้จึงมีความพยายามในการพัฒนาขั้นตอนวิธีการที่สามารถหาค่าเหมาะที่สุดใกล้เคียง ภายในเวลาที่สามารถยอมรับได้ โดยใช้วิธีฮิวริสติกส์ และเมต้าฮิวริสติกส์ ซึ่งในงานวิจัยนี้ ได้กำหนดกรอบการทำงานไว้ดังภาพที่ 6



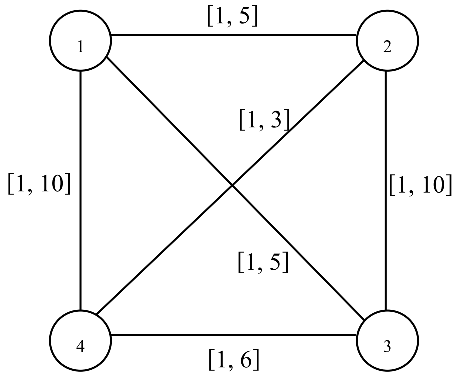
ภาพที่ 6 กรอบการทำงานวิจัย (Research Framework)

วิธีการหาผลเฉลยในการจัดเส้นทางเดินรถขนส่งให้เกิดเวลาเดินทางรวมสั้นที่สุดและมีความทนทานต่อความไม่แน่นอนมากที่สุด โดยใช้ขั้นตอนวิธีการผสมผสานระหว่างวิธีฮิวริสติกส์และวิธีเมต้าฮิวริสติกส์ มีรายละเอียดดังต่อไปนี้

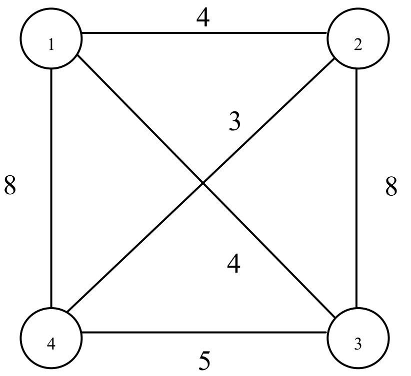
1. สร้างรูปแบบของข้อมูลที่มีความไม่แน่นอน (Structuring of Data Uncertainty)

การสร้างรูปแบบของเวลาเดินทางที่ไม่แน่นอนอาศัยวิธีการสร้างสถานการณ์ โดยพิจารณาปัจจัยหลักที่ส่งผลกระทบต่อระยะเวลาการขนส่งสินค้าจากคลังสินค้าไปยังลูกค้าที่อยู่ ณ ตำแหน่งต่างๆ ตัวอย่างปัจจัยที่มีผลกระทบ ได้แก่ สภาพการจราจร การเกิดอุบัติเหตุ หรือสภาวะอากาศ เป็นต้น การกำหนดสถานการณ์ที่ดีไม่ควรกำหนดเป็นสถานการณ์ที่ดีที่สุด หรือเลวร้ายที่สุด เนื่องจากอาจส่งผลกระทบต่อผลเฉลยที่ไม่เที่ยงตรง ผู้ตัดสินใจที่ดีควรสร้างสถานการณ์ตามความเป็นจริง โดยพิจารณาทีละปัจจัย หรือพิจารณาความสัมพันธ์ระหว่างปัจจัย อาจกระทำในลักษณะของการออกแบบการทดลอง (Design of Experiments) เพื่อหาปัจจัยที่ส่งผลกระทบต่อระยะเวลาในการขนส่ง และนำค่าตอบสนองที่ได้ (Response Values) มาใช้เป็นพารามิเตอร์ในการสร้างสถานการณ์ต่อไป ซึ่งในงานวิจัยไม่ขอกล่าวในรายละเอียด

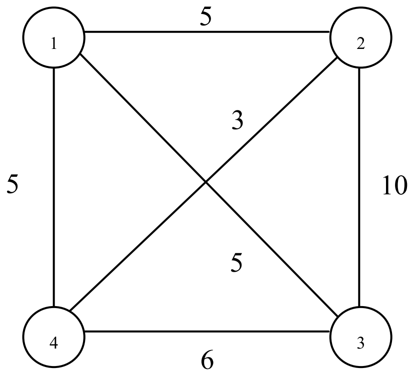
ตัวอย่างในการกำหนดสถานการณ์เพื่อนำไปสู่การหาผลเฉลยที่ครอบคลุมทุกสถานการณ์แสดงดังภาพที่ 7-9



ภาพที่ 7 ตัวอย่างช่วงเวลาที่ใช้ในการเดินทางขนส่งจากคลังสินค้า 1 ไปยังลูกค้า 3 จุด



ภาพที่ 8 ตัวอย่างการกำหนดสถานการณ์ที่ 1 กรณีการจราจรติดขัด



ภาพที่ 9 ตัวอย่างการกำหนดสถานการณ์ที่ 2 กรณีเกิดอุบัติเหตุ ณ เส้นทางระหว่างลูกค้า 2 และ 3

ในงานวิจัยนี้มิได้มุ่งเน้นในการสร้างสถานการณ์ที่เกิดขึ้นจริง เนื่องจากข้อมูลดังกล่าว ผู้ตัดสินใจในการใช้งานเท่านั้นที่จะทราบข้อมูลโดยแท้จริง งานวิจัยนี้จึงได้เพียงเสนอแนวทางในการสร้างสถานการณ์เพื่อเป็นตัวแทนของข้อมูลที่มีความไม่แน่นอนจากปัจจัยต่างๆ สำหรับในงานวิจัยนี้ใช้วิธีการสร้างสถานการณ์ของเวลาเดินทางจากคลังสินค้าไปยังลูกค้า และระหว่างลูกค้า i ไป j ด้วยนิยามดังต่อไปนี้

นิยามที่ 1: สถานการณ์ทั้งหมด S คือ เวลาเดินทางที่เกิดขึ้นระหว่างเส้นทางจากคลังสินค้าไปยังลูกค้า และระหว่างลูกค้า i ไป j โดยกำหนดเป็นช่วงเวลาที่ใช้ในการเดินทาง นิยามด้วย $[l_{ij}, u_{ij}]$ ซึ่ง l_{ij} คือ เวลาเดินทางต่ำสุด และ u_{ij} คือ เวลาเดินทางสูงสุด ในกรณีสถานการณ์พื้นฐาน (Base Case Scenario; s_b) โดย s_b คือ ค่าคาดหวังของ $[l_{ij}, u_{ij}]$ หรือเวลาเดินทางสำหรับกรณีสถานการณ์พื้นฐาน, $t_{s_b} = \dfrac{l_{ij} + u_{ij}}{2}$ สำหรับสถานการณ์อื่นๆ, t_s ถูกเลือกอย่างสุ่มจากการแจงแจงเอกรูป (Uniform Distribution), $u(l_{ij}, u_{ij})$

จากนิยามที่ 1 ทำให้สามารถสร้างเมทริกซ์ของค่าเวลาเดินทาง (Travel Times Cost Matrix) ได้ และความไม่แน่นอนเกิดจากการสร้างสถานการณ์ s ซึ่งส่งผลให้ค่าในเมทริกซ์เปลี่ยนไปตามสถานการณ์ที่เกิดขึ้น ขนาดของความไม่แน่นอนหรือเปอร์เซ็นต์ความไม่แน่นอนกำหนดด้วยค่าพารามิเตอร์, β ที่มีค่าอยู่ในช่วง $0 \leq \beta \leq 1$ ในกรณีที่มีการกำหนดเปอร์เซ็นต์ความไม่แน่นอน t_s จะถูกเลือกอย่างสุ่มจากการแจกแจงเอกรูป $u((1-\beta)t_{s_b},(1+\beta)t_{s_b})$ จากขั้นตอนนี้ทำให้สามารถสร้างเวลาเดินทางที่ไม่แน่นอนจากคลังสินค้าไปลูกค้า และจากลูกค้า i ไป j

2. เลือกเกณฑ์ที่เหมาะสมในการตัดสินผลเฉลยเชิงทนทาน (Choice of Appropriate Robustness Criterion)

ในงานวิจัยนี้ นำเสนอเกณฑ์การตัดสินผลเฉลยเชิงทนทาน 3 วิธี อันประกอบด้วย

2.1 สภาพทนทานสัมบูรณ์ (Absolute Robustness) เป็นตัววัดสมรรถนะ (Performance measure) ของผลเฉลย โดยการประเมินผลจากผลเฉลยที่ได้จากสถานการณ์ทั้งหมด และเลือกผลเฉลยที่มีค่ามากต่ำที่สุดเป็นตัวชี้บ่งความทนทาน (The Best Worst Case Performance) ตัวอย่างเช่น ค่าเหมาะที่สุดที่ได้จากการสร้างสถานการณ์ 4 สถานการณ์ ได้แก่ Z_{s_1} = 100, Z_{s_2} = 103, Z_{s_3} = 98 และ Z_{s_4} =99 และได้ผลเฉลยทั้งสิ้น 4 ทางเลือกจากสถานการณ์ทั้งหมด ดังต่อไปนี้

ตารางที่ 2 การหาผลเฉลยสภาพทนทานสัมบูรณ์ (Absolute Robustness Solution)

ผลเฉลย	สถานการณ์				$h_x(S)$ สูงสุด
	s_1	s_2	s_3	s_4	
x_1	100	105	110	108	110
x_2	103	103	119	121	121
x_3	110	112	98	110	112
x_4	115	105	104	99	115
$g_s(X)$ ต่ำสุด	100	103	98	99	110

จากตารางที่ 2 ค่า $h_x(S)$ ที่ต่ำที่สุด คือ 110 แสดงให้เห็นว่า x_1 คือ ผลเฉลยสภาพทนทานสัมบูรณ์ หมายความว่า เมื่อจัดเส้นทางเดินรถขนส่งแบบ x_1 ในกรณีที่เกิดสถานการณ์ที่แย่ที่สุด เวลาเดินทางโดยรวมมีค่าไม่เกิน 110 ในขณะที่ทางเลือกอื่นใช้เวลาเดินทางโดยรวมมากกว่า 110 เพราะฉะนั้น x_1 คือ ค่าเหมาะสูงสุดต่ำสุด (Minimax Optimization)

2.2 ความแตกต่างทนทาน (Robust Deviation) การวัดสมรรถนะของผลเฉลย ประเมินจากความแตกต่างระหว่างค่าเหมาะที่สุดเชิงทนทาน และค่าเหมาะที่สุดในแต่ละสถานการณ์ จากนั้นเลือกผลต่างที่มีค่ามากต่ำที่สุดเป็นตัวบ่งชี้ความทนทาน อ้างถึงตัวอย่างในหัวข้อ 2.1 การหาผลเฉลยความแตกต่างทนทาน แสดงดังตารางที่ 3

ตารางที่ 3 การหาผลเฉลยความแตกต่างทนทาน (Robust Deviation Solution)

ผลเฉลย	สถานการณ์				$h_x(S)$ สูงสุด
	s_1	s_2	s_3	s_4	
x_1	0	2	12	9	12
x_2	3	0	21	22	22
x_3	10	9	0	11	11
x_4	15	2	6	0	15
$g_s(X)$ ต่ำสุด	0	0	0	0	11

จากตารางที่ 3 ค่า $h_x(S)$ ที่ต่ำที่สุด คือ 11 แสดงให้เห็นว่า x_3 คือ ผลเฉลยความแตกต่างทนทาน หมายความว่า เมื่อจัดเส้นทางเดินรถขนส่งแบบ x_3 ในกรณีที่เกิดสถานการณ์ที่แย่ที่สุด ค่าความแตกต่างระหว่างผลเฉลยเชิงทนทาน กับค่าเหมาะที่สุดในกรณีเกิดสถานการณ์ที่เลวร้ายที่สุด มีค่าไม่เกิน 11 ในขณะที่ทางเลือกอื่นให้ค่าความแตกต่างมากกว่า 11 เพราะฉะนั้น x_3 คือ ค่าเหมาะสูงสุดต่ำสุด

2.3 สภาพทนทานแบบสัมพัทธ์ (Relative Robustness) เป็นตัววัดสมรรถนะสภาพทนทานแบบสัมพัทธ์ที่ใกล้เคียงกับความแตกต่างทนทานในหัวข้อที่ 2.2 ต่างกันที่ผลเฉลยสภาพทนทานแบบสัมพัทธ์วัดจากเปอร์เซ็นต์ความแตกต่างระหว่างผลเฉลยเชิงทนทานที่ได้กับค่าเหมาะที่สุดที่เกิดขึ้นในแต่ละสถานการณ์ และเลือกเปอร์เซ็นต์ความแตกต่างที่มีค่ามากต่ำสุด แสดงดังตารางที่ 4

จากตารางที่ 4 ค่า $h_x(S)$ ที่ต่ำที่สุด คือ 0.11 แสดงให้เห็นว่า x_3 คือ ผลเฉลยสภาพทนทานแบบสัมพัทธ์ หมายความว่า เมื่อจัดเส้นทางเดินรถขนส่งแบบ x_3 ในกรณีที่เกิดสถานการณ์ที่แย่ที่สุด เปอร์เซ็นต์ความแตกต่างระหว่างผลเฉลยเชิงทนทาน กับค่าเหมาะที่สุดในกรณีเกิด

สถานการณ์ที่เลวร้ายที่สุด มีค่าไม่เกิน 0.11 ในขณะที่ทางเลือกอื่นให้ค่าเปอร์เซ็นต์ความแตกต่างมากกว่า 0.11 เพราะฉะนั้น x_3 คือ ค่าเหมาะสูงสุดต่ำสุด

ตารางที่ 4 การหาผลเฉลยสภาพทนทานแบบสัมพัทธ์ (Relative Robustness Solution)

ผลเฉลย	สถานการณ์				$h_x(S)$ สูงสุด
	s_1	s_2	s_3	s_4	
x_1	0.00	0.02	0.12	0.09	0.12
x_2	0.03	0.00	0.21	0.22	0.22
x_3	0.10	0.09	0.00	0.11	0.11
x_4	0.15	0.02	0.06	0.00	0.15
$g_s(X)$ ต่ำสุด	0.00	0.00	0.00	0.00	0.11

เกณฑ์ตัดสินใจทั้ง 3 วิธีนี้ ถูกนำมาสร้างเป็นฟังก์ชั่นวัตถุประสงค์สำหรับการหาผลเฉลยที่มีสภาพทนทานต่อความไม่แน่นอน โดยกำหนดให้ S คือ สถานการณ์ทั้งหมดที่ผ่านการวางแผนและออกแบบมาแล้วในหัวข้อที่ 1, t คือ เวลาเดินทาง และ t_s คือ เวลาเดินทางที่เกิดขึ้น ณ สถานการณ์ย่อย s, กำหนดให้ F_s คือ เซตของผลเฉลยที่เป็นไปได้ทั้งหมดเมื่อสถานการณ์ s เกิดขึ้น โดยที่ X เป็นตัวแปรตัดสินใจ, $X \in F_s$ และประเมินคุณภาพของผลเฉลย X ด้วยฟังก์ชั่น $f(X,t_s)$, X_s^* คือ ผลเฉลยเหมาะที่สุดหรือเส้นทางการเดินรถที่ทำให้ค่าเวลาเดินทางโดยรวมต่ำสุดในแต่ละสถานการณ์ โดยมีค่าเวลาเดินทาง คือ t_s และกำหนดให้ Z_s คือ ค่าเวลาเดินทางต่ำสุดที่ได้จากผลเฉลย X_s^* ดังสมการที่ 1

$$Z_s = f(X_s^*, t_s) = \min_{X \in F_s} f(X, t_s) \qquad (1)$$

จากสมการที่ 1 สามารถแก้ปัญหาได้ด้วยกำหนดการเชิงกำหนด (Deterministic Programming) ทั่วไป หรืออาจเรียกสมการที่ 1 ฟังก์ชั่นวัตถุประสงค์ของส่วนโปรแกรมนับความทนทาน (Robust Counter Part) ซึ่งจำนวนรอบของการคำนวณค่าเหมาะที่สุดเท่ากับจำนวนของสถานการณ์ที่ผู้ตัดสินใจกำหนดขึ้น จากวัตถุประสงค์ย่อยนำไปสู่วัตถุประสงค์หลักของงานวิจัย ซึ่งกำหนดไว้ตามเกณฑ์ตัดสินใจ 3 แบบ คือ สภาพทนทานสัมบูรณ์ ความแตกต่างทนทาน และสภาพทนทานแบบสัมพัทธ์ นำมาสร้างเป็นวัตถุประสงค์หลักดังสมการที่ 2 – 3 โดยมีเป้าหมายที่ค่าสูงสุด

สภาพทนทานสัมบูรณ์ : โดยที่ Z_A คือ ค่าสัมบูรณ์สูงสุด, X_A คือ ตัวแปรตัดสินใจของสภาพทนทานสัมบูรณ์, t_s คือ เวลาเดินทางในสถานการณ์ s, F_s คือ เซตของผลเฉลยที่เป็นไปได้ทั้งหมดเมื่อสถานการณ์ s เกิดขึ้น โดยที่ $\bigcap_{s\in S} F_s \neq \phi$

$$Z_A = \max_{s\in S} f(X_A, t_s) = \min_{X\in\bigcap_{s\in S} F_s} \max_{s\in S} f(X, t_s) \tag{2}$$

ความแตกต่างทนทาน: โดยที่ Z_D คือ ค่าความแตกต่างสูงสุด, X_D คือ ตัวแปรตัดสินใจของความแตกต่างทนทาน

$$Z_D = \max_{s\in S}(f(X_D, t_s) - f(X_s^*, t_s)) = \min_{X\in\bigcap_{s\in S} F_s} \max_{s\in S}(f(X, t_s) - f(X_s^*, t_s)) \tag{3}$$

สภาพทนทานแบบสัมพัทธ์: โดยที่ Z_R คือ ค่าสัมพัทธ์สูงสุด, X_R คือ ตัวแปรตัดสินใจของความแตกต่างทนทาน

$$Z_R = \max_{s\in S} \frac{f(X_R, t_s) - f(X_s^*, t_s)}{f(X_s^*, t_s)} = \min_{X\in\bigcap_{s\in S} F_s} \max_{s\in S} \frac{f(X, t_s) - f(X_s^*, t_s)}{f(X_s^*, t_s)} \tag{4}$$

หากจัดรูปแบบสมการ 2-4 ให้อยู่ในรูปทั่วไป แสดงได้ดังสมการที่ 5-6

$$Z = \min\{h_x(S) \mid g_s(X) \le h_x(S), s \in S; X \in \bigcap_{s\in S} F_s\} \tag{5}$$

$$\text{โดยที่ } g_s(X) = \begin{cases} f(X, t_s) \\ f(X, t_s) - Z_s \\ \dfrac{f(X, t_s) - Z_s}{Z_s} \end{cases} \tag{6}$$

$h_x(S)$ คือ ค่ามากที่สุดที่ได้จากสถานการณ์ทั้งหมด ด้วยเกณฑ์การตัดสินใจผลเฉลยเชิงทนทาน $Z_A = f(X, t_s)$, $Z_D = f(X, t_s) - Z_s$ และ $Z_R = \dfrac{f(X, t_s) - Z_s}{Z_s}$ ดังนั้นค่าเหมาะที่สุดเชิงทนทาน $Z =$ ค่า $h_x(S)$ ต่ำสุดดังสมการที่ (5)

ซึ่งปัญหาในลักษณะนี้เรียกว่า ปัญหาการหาค่าเหมาะสูงสุดต่ำสุด (Minimax Optimization Problem) หมายถึง การหาค่า $h_x(S)$ ที่มีค่าสูงสุดจากสถานการณ์ที่กำหนดทั้งหมด จากนั้นเลือกผลเฉลย x_R จากผลเฉลยที่เป็นไปได้ทั้งหมดที่ให้ค่า $h_x(S)$ ต่ำที่สุด เป็นผลเฉลยเชิงทนทาน และให้ค่าเหมาะที่สุดใกล้เคียงในกรณีเกิดสถานการณ์เลวร้ายที่สุด โดยตัวอย่างของการหาค่าเหมาะสูงสุด

ต่ำสุด แสงดังตารางที่ 2-4 ซึ่งกำหนดสถานการณ์จำนวน 4 เหตุการณ์ และคำนวณค่าเหมาะสูงสุดต่ำสุดโดยใช้เกณฑ์ตัดสินใจทั้ง 3 เกณฑ์ ได้แก่ $Z_A = 110$, Z_D=11 และ Z_R =0.11 โดยมีผลเฉลยเชิงทนทานเป็น x_1, x_3 และ x_3 ตามลำดับ

3. สร้างรูปแบบเชิงคณิตศาสตร์ (Mathematical Model Formulation)

ปัญหา VRPTW ที่มีความไม่แน่นอน เป็นปัญหาที่มีความซับซ้อนที่สามารถสร้างรูปแบบเชิงคณิตศาสตร์ได้ โดยมีฟังก์ชั่นวัตถุประสงค์ตามที่ได้กล่าวไว้ในหัวข้อที่ 2 และฟังก์ชั่นเงื่อนไขเป็นไปตามเงื่อนไขของปัญหา VRPTW แบบกำหนดการเชิงกำหนดทั่วไป เนื่องจากค่าพารามิเตอร์ของความไม่แน่นอนในงานวิจัยนี้ มีผลกระทบต่อฟังก์ชั่นวัตถุประสงค์เท่านั้น ส่วนฟังก์ชั่นเงื่อนไขยังคงเดิม อย่างไรก็ตามการแก้ปัญหาด้วยวิธีแม่นตรง, กำหนดการไม่เชิงเส้น (Non-Linear Programming) หรือกำหนดการเชิงจำนวนเต็ม (Integer Programming) อาจกระทำได้ยาก เนื่องจากสมการมีเป็นจำนวนมาก และมีความซับซ้อน การแก้ปัญหาทางตรงอาจใช้เวลานาน หรืออาจแก้ปัญหาไม่ได้เลย ซึ่งในหัวข้อนี้ได้แสดงรูปแบบปัญหาเชิงคณิตศาสตร์ เพื่อเป็นแนวทางนำไปสู่การพัฒนาขั้นตอนวิธีการแบบผสมผสานในหัวข้อถัดไป

3.1 การสร้างรูปแบบเชิงคณิตศาสตร์สำหรับ VRPTW ทั่วไป เป็นการสร้างรูปแบบกำหนดการเชิงกำหนด หรือในกรณีที่มีสถานการณ์เกิดขึ้นเพียงสถานการณ์เดียว (Single Scenario) โดยมีฟังก์ชั่นวัตถุประสงค์ 2 ประการ คือ การหาเวลาเดินทางต่ำสุด และจำนวนรถขนส่งน้อยที่สุด

3.1.1 การใช้สัญกรณ์ (Notation) สำหรับปัญหา VRPTW

K : จำนวนรถขนส่งทั้งหมด $k = 1,2,...K$

N : จำนวนลูกค้าและคลังสินค้า

C_i : ลูกค้าคนที่ i โดยที่ $i = 2,3...,N$

C_1 : คลังสินค้า

d_i : ความต้องการสินค้าของลูกค้า i

q_k : ความสามารถในการบรรทุกสินค้าของรถขนส่ง k

t_{ij} : เวลาเดินทางระหว่างลูกค้า i ไปยังลูกค้า j
โดยที่ $i, j = 1,...,N$, $i \neq j$ และ $i, j = 1$ คือ คลังสินค้า

e_i : เวลาที่อนุญาตให้รถขนส่งมาถึงลูกค้า i ได้เร็วที่สุด (Earliest Arrival Time)

l_i : เวลาที่อนุญาตให้รถขนส่งมาถึงลูกค้า i ได้ช้าที่สุด (Latest Arrival Time)

A_i : เวลาที่รถขนส่งมาถึงลูกค้า i (Arrival Time)

b_i : เวลาในการให้บริการลูกค้า i (Service Time)

w_{ij} : เวลารอคอยจากลูกค้า i ไป j (Waiting Time)

โดยที่ $w_{ij} = \max[e_j - (A_i + t_{ij}),0]$, $i, j = 2,...,N$ และ $i \neq j$

M_k : เวลามากสุดที่อนุญาตให้รถขนส่งเดินทางให้บริการลูกค้าและกลับมายังคลังสินค้า (Maximum Route Time) โดยที่ $k = 1,2,...K$

3.1.2 ตัวแปรตัดสินใจ (Decision Variable)

$$X_{ijk} = \begin{cases} 1 & \text{ถ้ารถขนส่ง } k \text{ ถูกจัดให้ส่งสินค้าจากลูกค้า } i \text{ ไป } j \\ 0 & \text{กรณีอื่นๆ} \end{cases}$$

3.1.3 ฟังก์ชั่นวัตถุประสงค์ (Objective Function)

เวลาเดินทางต่ำสุด

$$Z_1 = \min \sum_{i=1}^{N} \sum_{j=1}^{N} \sum_{k-1}^{K-1} t_{ij} X_{ijk} \tag{7}$$

จำนวนรถขนส่งน้อยที่สุด

$$Z_2 = \min K \tag{8}$$

3.1.4 ฟังก์ชั่นเงื่อนไข (Constraints Function)

$$\sum_{k=1}^{K-1} \sum_{j=2}^{N} X_{ijk} = K \qquad \text{สำหรับ } i = 1 \tag{9}$$

$$\sum_{k=1}^{K-1} \sum_{\substack{j=2 \\ j \neq i}}^{N} X_{ijk} = 1 \qquad \text{สำหรับ } i = 2,3,...,N \tag{10}$$

$$\sum_{\substack{i=1 \\ i\neq h}}^{N} X_{ihk} - \sum_{\substack{j=1 \\ j\neq h}}^{N} X_{ijk} = 0 \qquad \text{โดยที่ } \forall h \in [1,N]; k \in [1,K-1] \tag{11}$$

$$u_i - u_j + Nx_{ij} \leq N-1 \qquad \text{สำหรับ } i = 2,3,...,N; j = 2,3,...,N, i \neq j \tag{12}$$

$$\sum_{i=1}^{N} \sum_{\substack{j=1 \\ j\neq i}}^{N} d_i X_{ijk} \leq q_k \qquad \text{สำหรับ } \forall k \in [1,K-1] \tag{13}$$

$$\sum_{i=1}^{N} \sum_{\substack{j=1 \\ j\neq i}}^{N} t_{ij} X_{ijk} \leq M_k \qquad \text{สำหรับ } \forall k \in [1,K-1] \tag{14}$$

$$t_0 = 0 \tag{15}$$

$$A_i + X_{ijk}(t_{ij} + b_i + w_{ij}) \leq Ai \qquad \text{สำหรับ } i, j \in [1,N]; i \neq j, k \in [1,K-1] \tag{16}$$

$$e_i \leq t_i \leq l_i \qquad \text{สำหรับ } k \in [1,K-1] \tag{17}$$

เงื่อนไขในสมการที่ 9 ประกันว่าจำนวนเส้นทางที่ใช้ในการขนส่งทั้งหมดมีจำนวนเท่ากับจำนวนรถขนส่ง เงื่อนไขในสมการที่ 10-11 มีจำนวนรถขนส่ง 1 คันเท่านั้น ที่ไปรับสินค้าจากลูกค้าและเดินทางกลับยังคลังสินค้าเดิม เมื่อให้บริการลูกค้าในเส้นทางจนครบ เงื่อนไขในสมการที่ 12 ป้องกันการเกิดเส้นทางย่อย (Sub-Tour) ขึ้นในผลเฉลยเงื่อนไขในสมการที่ 13 รถขนส่งต้องรับน้ำหนักไม่เกินความสามารถในการบรรทุก เงื่อนไขในสมการที่ 14 เวลาทั้งหมดที่รถขนส่งแต่ละคันใช้ในการเดินทางต้องไม่เกินเวลามากสุดที่อนุญาตให้รถขนส่งเดินทางให้บริการลูกค้าและกลับมายังคลังสินค้าเริ่มต้น และเงื่อนไขในสมการที่ 15-17 การขนส่งสินค้าต้องไม่ขัดแย้งกับเงื่อนไขของกรอบเวลา

3.2 การสร้างรูปแบบเชิงคณิตศาสตร์สำหรับ VRPTW และเวลาเดินทางไม่แน่นอน เป็นการสร้างรูปแบบสำหรับปัญหา VRPTW ที่มีเวลาเดินทางไม่แน่นอน ใช้วิธีการเข้าสู่ผลเฉลยแบบการสร้างสถานการณ์ และใช้เกณฑ์ตัดสินประสิทธิภาพเชิงทนทาน ซึ่งอาจเรียกปัญหานี้ว่า “ปัญหาการจัดเส้นทางเดินรถขนส่งแบบมีกรอบเวลาเชิงทนทาน (Robust Vehicle Routing Problem with Time Windows)

3.2.1 การใช้สัญกรณ์ (Notation) สำหรับปัญหา RVRPTW

S : จำนวนสถานการณ์ทั้งหมด $s = 1,2,...S$

t_s : เวลาเดินทางระหว่างลูกค้า i ไปยังลูกค้า j ณ สถานการณ์ s

Z_s : ค่าเหมาะที่สุดที่ได้จากสมการที่ 7 และ 8

3.2.2 ตัวแปรตัดสินใจ (Decision Variable)

$$X_{ijk} = \begin{cases} 1 & \text{ถ้ารถขนส่ง } k \text{ ถูกจัดให้ส่งสินค้าจากลูกค้า } i \text{ ไป } j \\ 0 & \text{กรณีอื่นๆ} \end{cases}$$

3.2.3 ฟังก์ชั่นวัตถุประสงค์ (Objective Function)

$Z = \min\{h_x(S) \mid g_s(X) \le h_x(S), s \in S; X \in \bigcap_{s \in S} F_s\}$ เป็นไปตามสมการที่ 5

3.2.4 ฟังก์ชั่นเงื่อนไข (Constraints Function)

$$g_s(X) = \begin{cases} f(X,t_s) \\ f(X,t_s) - Z_s \\ \dfrac{f(X,t_s) - Z_s}{Z_s} \end{cases}$$

$g_s(X)$ เป็นไปตามสมการที่ 6 และเพิ่มฟังก์ชั่นเงื่อนไขสำหรับปัญหา VRPTW กรณีมีสถานการณ์เดียว (สมการที่ 9-17) สังเกตได้ว่าการสร้างรูปแบบเชิงคณิตศาสตร์สำหรับปัญหา RVRPTW ค่อนข้างมีความซับซ้อน โดยต้องทำการแก้ปัญหา VRPTW เป็นจำนวน S เพื่อนำมาหาฟังก์ชั่นวัตถุประสงค์เชิงทนทานในสมการที่ 5 การคำนวณทางตรง หรือวิธีแม่นตรงจึงไม่สามารถแก้ปัญหาได้ การแก้ปัญหาที่เหมาะสม คือ การพัฒนาขั้นตอนวิธีการทางคอมพิวเตอร์ โดยประยุกต์ใช้วิธีการฮิวริสติกส์ และเม้ต้าฮิวริสติกส์จึงน่าจะเป็นวิธีการที่เหมาะสมที่สุด ซึ่งในงานวิจัยได้นำเสนอขั้นตอนวิธีการในหัวข้อถัดไป

4. สร้างขั้นตอนวิธีการในการแก้ปัญหา (Algorithm Construction)

ในการพัฒนาขั้นตอนวิธีการในการแก้ปัญหา RVRPTW นี้ ได้ดัดแปลงวิธีการแทรกไปข้างหน้า หรือ “Modified Push-Forward Insertion Heuristic” ซึ่งเดิมเป็นวิธีฮิวริสติกส์ที่นำเสนอไว้โดย Solomon (1987) และ Thangiah (1993) ในการสร้างผลเฉลยเริ่มต้น และทำการปรับปรุงผลเฉลยด้วยวิธีการค้นหาเฉพาะที่แบบการสลับสับเปลี่ยนค่าแลมด้า หรือ “ λ -Interchange Local Search Descent (λ -LSD)” ซึ่งนำเสนอไว้โดย Osman (1993) ขั้นตอนวิธีการถูกป้องกันการเกิดปัญหาการตกหลุมค่าเหมาะเฉพาะที่ด้วยขั้นตอนวิธีการ TS ซึ่งเริ่มต้นนำเสนอไว้โดย Glover (1989) จากขั้นตอนวิธีการดังกล่าวเป็นดัดแปลงและผสมผสานระหว่างวิธีฮิวริสติกส์และเม้ต้าฮิวริสติกส์ที่ได้รับความนิยมในการแก้ปัญหา VRPTW เพื่อให้เกิดประสิทธิภาพในการค้นหาค่าเหมาะที่สุดหรืออย่างน้อยให้ได้ค่าเหมาะที่สุดใกล้เคียง ซึ่งรายละเอียดของขั้นตอนเป็นดังต่อไปนี้

4.1 การใช้สัญกร (Notation) สำหรับขั้นตอนวิธีการทางคอมพิวเตอร์

K : จำนวนรถขนส่งทั้งหมด $k = 1,2,...K$

K_{LB} : จำนวนรถขนส่งขอบเขตล่าง โดยที่ $K_{LB} = \dfrac{\sum_{i=2}^{N} d_i}{q_k}$

N : จำนวนลูกค้าและคลังสินค้า

C_i : ลูกค้าคนที่ i โดยที่ $i = 2,3...,N$

C_1 : คลังสินค้า

d_i : ความต้องการสินค้าของลูกค้า i

D_k : ความต้องการสินค้ารวมสำหรับรถขนส่ง k

q_k : ความสามารถในการบรรทุกสินค้าของรถขนส่ง k

t_{ij} : เวลาเดินทางระหว่างลูกค้า i ไปยังลูกค้า j
โดยที่ $i, j = 1,...,N$, $i \neq j$ และ $i, j = 1$ คือ คลังสินค้า

e_i : เวลาที่อนุญาตให้รถขนส่งมาถึงลูกค้า i ได้เร็วที่สุด (Earliest Arrival Time)

l_i : เวลาที่อนุญาตให้รถขนส่งมาถึงลูกค้า i ได้ช้าที่สุด (Latest Arrival Time)

A_i : เวลาที่รถขนส่งมาถึงลูกค้า i (Arrival Time)

b_i : เวลาในการให้บริการลูกค้า i (Service Time)

w_{ij} : เวลารอคอยจากลูกค้า i ไป j (Waiting Time)
โดยที่ $w_{ij} = \max[e_j - (A_i + t_{ij}),0]$, $i, j = 2,...,N$ และ $i \neq j$

M_k : เวลามากสุดที่อนุญาตให้รถขนส่งเดินทางให้บริการลูกค้าและกลับมายังคลังสินค้า (Maximum Route Time) โดยที่ $k = 1,2,...K$

R_k : เส้นทางการเดินรถขนส่ง k โดยที่ $k = 1,2,...K$

W_k : เวลารอคอยรวมสำหรับรถขนส่ง k โดยที่ $k = 1,2,...K$ (Total Waiting Time)

B_k : เวลาให้บริการรวมสำหรับรถขนส่ง k โดยที่ $k = 1,2,...K$ (Total Service Time)

O_k : เวลาล่วงเวลารวมสำหรับรถขนส่ง k โดยที่ $k = 1,2,...K$ (Total Overtime)

L_k : เวลาล่าช้ารวมสำหรับรถขนส่ง k โดยที่ $k = 1,2,...K$ (Total Tardiness)

T_k : เวลาเดินทางรวมสำหรับรถขนส่ง k โดยที่ $k = 1,2,...K$ (Total Travel Times)

Tot_k : เวลาเดินทางรวมทั้งหมด (Overall of Total Travel Times) หรือ $Tot_k = T_k + W_k + B_k$ สำหรับรถขนส่ง k โดยที่ $k = 1,2,...K$

$F(X)$: ฟังก์ชั่นวัตถุประสงค์สำหรับเส้นทาง R_k

α : ค่าน้ำหนักโทษสำหรับกรณีเกิดเวลารอคอย

γ : ค่าน้ำหนักโทษสำหรับกรณีเกิดเวลาล่าช้า

η : ค่าน้ำหนักโทษสำหรับกรณีเกิดเวลาล่วงเวลา

4.1.1 ฟังก์ชั่นวัตถุประสงค์ (Objective Function)

$$F(X) = T_k + (\alpha \times W_k) + (\gamma \times L_k) + (\eta \times O_k) \tag{18}$$

สมการที่ 18 คือ ฟังชั่นวัตถุประสงค์ ที่แสดงเวลาเดินทางรวมที่สั้นที่สุด โดยขั้นตอนวิธีการยอมให้เกิดเวลารอคอย เวลาล่าช้า และเวลาล่วงเวลา ทั้งนี้เพื่อเป็นการผ่อนปรนให้ขั้นตอนวิธีการสามารถค้นหาผลเฉลยที่ดีภายในเวลาที่ยอมรับได้ ตัวอย่างเช่น กำหนดค่าน้ำหนักโทษ α = 0.01, γ = 0.5 และ η = 0.05 หมายความว่า ขั้นตอนวิธีการยอมให้เกิดเวลารอคอยได้มากที่สุด จากค่าน้ำหนักโทษที่น้อยที่สุด และยอมให้เกิดเวลาล่าช้าน้อยที่สุด โดยการกำหนดค่าน้ำหนักโทษสูงที่สุดถึง 50 % ของเวลาล่าช้าที่เกิดขึ้น ซึ่งผลเฉลยที่ได้จะมีการเปลี่ยนแปลงตามค่าน้ำหนักโทษที่กำหนดทั้งนี้ขึ้นอยู่กับวัตถุประสงค์ของผู้ตัดสินใจว่าต้องการให้ผลเฉลยมีความเข้มงวด หรือผ่อนปรนมากน้อยเพียงใด

4.2 ขั้นตอนวิธีการสร้างผลเฉลยเริ่มต้น (Initial Solution) การสร้างผลเฉลยเริ่มต้น โดยใช้วิธี MPFIH ซึ่งเป็นวิธีการแบ่งกลุ่มลูกค้าและจัดเส้นทางไปพร้อมๆกัน (Simultaneous Route and Cluster Method) เช่นเดียวกับวิธีการเซฟวิ่ง แต่เปลี่ยนจากการเรียงลำดับค่าเซฟวิ่ง เป็นการ

เรียงลำดับค่า e_i จากน้อยไปมาก และเรียงลำดับค่า l_i จากมากไปน้อย จากนั้น เริ่มต้นจากการสร้างเส้นทาง R_k โดยประมาณจำนวนรถขนส่งเริ่มต้นเท่ากับ K_{LB} สร้างเซตของกลุ่มลูกค้าเริ่มต้นจากกลุ่มลูกค้าที่มีค่า e_i น้อยที่สุด จำนวน k ลำดับแรก และสร้างเซตของกลุ่มลูกค้าสิ้นสุดจากกลุ่มลูกค้าที่มีค่า l_i มากที่สุดจำนวน k ลำดับแรกเช่นกัน เพราะฉะนั้นเมทริกซ์ของเส้นทางเริ่มต้นจึงมีขนาดเท่ากับ $k \times 2$

$$R_k = \begin{bmatrix} C_{l1} & C_{m1} \\ C_{l2} & C_{m2} \\ \vdots & \vdots \\ C_{lk} & C_{mk} \end{bmatrix} \tag{19}$$

ในขั้นตอนเริ่มแรกนี้กำหนดให้ $k = K_{LB}$ โดยที่ C_{lk} คือเซตของกลุ่มลูกค้าเริ่มต้นที่มีค่า e_i น้อยที่สุด และ C_{mk} คือ เซตของกลุ่มลูกค้าสิ้นสุดจากกลุ่มลูกค้าที่มีค่า l_i มากที่สุด ขั้นตอนต่อไป คือ การหาเซตของลูกค้าแทรก C_{ik} ระหว่างเซตลูกค้าเริ่มต้นและลูกค้าสิ้นสุด โดยพิจารณาจากลูกค้าที่มีค่า e_i น้อยในลำดับถัดมาอีกจำนวน k ปม โดยในการแทรกแต่ละรอบมีเงื่อนไขดังต่อไปนี้

เงื่อนไขที่ 1: ค่าความต้องการสินค้ารวมในแต่ละเส้นทาง หรือในแต่ละแถวของเมทริกซ์ไม่เกินความสามารถในการบรรทุกสินค้าของรถขนส่งk , $D_k = \sum_{i=l}^{m} d_i \le q_k$

เงื่อนไขที่2 : เวลาเดินทางโดยรวมต้องไม่เกินกว่าเวลามากสุดที่อนุญาตให้รถขนส่งเดินทางให้บริการลูกค้าและกลับมายังคลังสินค้า, $Tot_k \le M_k$

และเงื่อนไขที่ 3 : ต้องไม่เกิดเวลาล่าช้าในแต่ละเส้นทาง หรือในแต่ละแถวของเมทริกซ์, $L_k = 0$

$$D_k = \begin{bmatrix} d_{l1} & d_{i1} & d_{m1} \\ d_{l2} & d_{i2} & d_{m2} \\ \vdots & \vdots & \vdots \\ d_{lk} & d_{ik} & d_{mk} \end{bmatrix} \le \begin{bmatrix} q_1 \\ q_2 \\ \vdots \\ q_k \end{bmatrix}, \ Tot_k = \begin{bmatrix} Tot_1 \\ Tot_2 \\ \vdots \\ Tot_k \end{bmatrix} \le \begin{bmatrix} q_1 \\ q_2 \\ \vdots \\ q_k \end{bmatrix} \text{ และ } L_k = \begin{bmatrix} L_1 \\ L_2 \\ \vdots \\ L_k \end{bmatrix} \le \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 0 \end{bmatrix} \tag{20}$$

หากเงื่อนไขในการแทรกเป็นไปตามสมการ 19 ขั้นตอนวิธีการนี้จะยอมให้แทรกเซตของลูกค้า C_{ik} เข้าไปยังเมทริกซ์ในสมการที่ 18 และตัดลูกค้าที่ได้รับการแทรกเรียบร้อยแล้วออกจากกลุ่มลูกค้าที่เหลือ เพื่อป้องกันไม่ให้เกิดการเลือกลูกค้าซ้ำเดิมอีก การแทรกจะกระทำเช่นเดียวกันไปเรื่อยๆ จนกว่าจะมีการผิดเงื่อนไขแม้เพียงข้อใดข้อหนึ่ง หรือเกิดขึ้นเฉพาะเส้นทางใดเส้นทางหนึ่ง กระบวนการแทรกที่ละเซตของกลุ่มลูกค้าจะหยุดลงทันที จากนั้นขั้นตอนวิธีการจะพิจารณาแทรกลูกค้าทีละปม โดยเลือกแทรกลูกค้า C_i ที่ยังคงเหลืออยู่ และจัดให้แทรกลงในเส้นทางหรือแถวของเมทริกซ์ที่ยังไม่ผิดเงื่อนไข จนกว่าไม่สามารถแทรกได้ ขั้นตอนวิธีการจึงยอมให้สร้างเส้นทางใหม่ขึ้นเป็น $k+i$, $i = 1,2,...,n$ โดยที่ n คือ จำนวนเส้นทางทั้งหมดที่เพิ่มขึ้นจาก K_{LB} ขั้นตอนวิธีการจะเสร็จสิ้นก็ต่อเมื่อลูกค้าทั้งหมดถูกจัดให้อยู่ในเส้นทางทั้งหมด โดยขั้นตอนวิธีการทางคอมพิวเตอร์มีรายละเอียดดังนี้

ขั้นตอนที่ 1 เรียงลำดับค่า e_i จากน้อยไปมาก และเรียงลำดับค่า l_i จากมากไปน้อย

ขั้นตอนที่ 2 สร้างเมทริกซ์ R_k เริ่มต้น, $k = K_{LB}$

ขั้นตอนที่ 3 สร้างเซต C_{lk} ที่มีค่า e_i น้อยที่สุด และเซต C_{mk} ที่มีค่า l_i มากที่สุดจำนวน k ลำดับแรก

ขั้นตอนที่ 4 ตัดค่า e_i ของเซต C_{lk} และ C_{mk} ออก เพื่อป้องกันการเลือกซ้ำปมเดิม

ขั้นตอนที่ 5 เลือกเซต C_{ik} ที่มีค่า e_i น้อยที่สุดใน k ลำดับถัดมา

ขั้นตอนที่ 6 ตรวจสอบเงื่อนไข $D_k = \sum_{i=l}^{m} d_i \le q_k$, $Tot_k \le M_k$ และ $L_k = 0$ ในแต่ละเส้นทาง หรือแต่ละแถวของเมทริกซ์ R_k ถ้าเป็นไปตามเงื่อนไขทุกแถวให้ไปขั้นตอนที่ 7 ถ้าไม่เป็นไปตามเงื่อนไขให้ไปขั้นตอนที่ 9

ขั้นตอนที่ 7 แทรกเซต C_{ik} ที่เป็นไปตามเงื่อนไขระหว่างเซต C_{lk} และ C_{mk} และทำตามขั้นตอนที่ 4-6

ขั้นตอนที่ 8 เมื่อเซต C_{ik} ทั้งหมดถูกจัดเข้าสู่เส้นทาง R_k จนครบเป็นอันจบขั้นตอนวิธีการ MPFIH

ขั้นตอนที่ 9 เลือก C_i ที่เหลือที่มีค่า e_i น้อยในลำดับถัดมา

ขั้นตอนที่ 10 ตรวจสอบเงื่อนไข $D_k = \sum_{i=l}^{m} d_i \le q_k$, $Tot_k \le M_k$ และ $L_k = 0$ ในแต่ละเส้นทาง หรือแต่ละแถวของเมทริกซ์ R_k ที่เหลือ ถ้าเป็นไปตามเงื่อนไขให้ไปขั้นตอนที่ 11 ถ้าไม่เป็นไปตามเงื่อนไขให้ไปขั้นตอนที่ 14

ขั้นตอนที่ 11 แทรก C_i ในเส้นทางที่เป็นไปตามเงื่อนไข

ขั้นตอนที่ 12 ตัดค่า e_i ของ C_i ออก เพื่อป้องกันการเลือกซ้ำปมเดิม ทำตามขั้นตอนที่ 9-12

ขั้นตอนที่ 13 เมื่อ C_i ทั้งหมดถูกจัดเข้าสู่เส้นทาง R_k จนครบเป็นอันจบขั้นตอนวิธีการ MPFIH

ขั้นตอนที่ 14 เพิ่มเส้นทางใหม่เป็น R_{k+i}, $i = 1,2,...,n$ และทำตามขั้นตอนที่ 9-13

4.3 ขั้นตอนวิธีการปรับปรุงผลเฉลย (Improving Solution)

4.3.1 ขั้นตอนวิธีการ λ-LSD เป็นขั้นตอนวิธีการค้นหาค่าเหมาะเฉพาะที่ (Local Search) มีความหมายเดียวกับวิธีการค้นหาข้างเคียง (Neighborhood Search) วิธีการปรับปรุงวนซ้ำ (Iterative Improvement) เป็นการหาค่าฟังก์ชั่นวัตถุประสงค์ที่ดีที่สุด จากผลเฉลยที่เป็นไปได้ทั้งหมด โดยค่าฟังก์ชั่นวัตถุประสงค์ แสดงดังสมการที่ 18

วิธีการนี้อาจทำให้เสียเวลามาก เมื่อปัญหามีขนาดใหญ่ขึ้น แต่ผลเฉลยที่ได้อาจเป็นค่าเหมาะที่สุด หรือค่าเหมาะที่สุดใกล้เคียง จึงเหมาะกับปัญหาที่ต้องการวางแผนระยะยาว โดยไม่ต้องมีการคำนวณบ่อยครั้งนัก เช่นเดียวกับปัญหา RVRPTW ในงานวิจัยนี้ เมื่อได้ผลเฉลยเริ่มต้นจากวิธี MPFIH ในหัวข้อที่ 4.2 แล้ว ขั้นตอนต่อไป คือ การปรับปรุงผลเฉลยที่ได้จากการค้นหาในแต่ละรอบการวนซ้ำให้มีคุณภาพ หรือมีค่าฟังก์ชั่นวัตถุประสงค์ดีขึ้นเรื่อยๆ หรือได้จำนวนเส้นทางที่ลดลง ซึ่งเป็นการประหยัดต้นทุนในการจัดเตรียมรถขนส่ง การค้นหาผลเฉลยจะกระทำต่อไปเรื่อยๆ จนกว่าคุณภาพของผลเฉลยจะไม่เปลี่ยนแปลง หรือไม่สามารถปรับปรุงให้ดีขึ้นอีกได้ หรือครบจำนวนรอบการวนซ้ำที่กำหนด ซึ่งผลเฉลยที่ได้ต้องมีการปรับปรุงค่าฟังก์ชั่นวัตถุประสงค์หรืออย่างน้อยต้องเท่ากับค่าเดิมที่ได้จากผลเฉลยเริ่มต้น

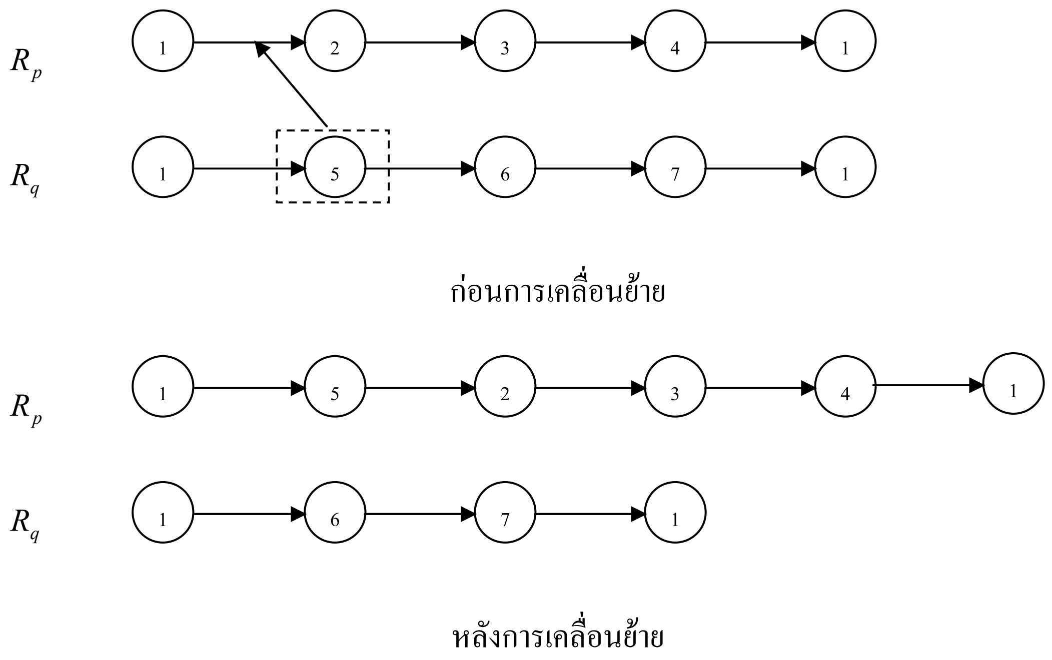
ขั้นตอนวิธีการเริ่มจากการผลเฉลยเริ่มต้น $X = \{R_1,...R_p,...R_q,...,R_k\}$ โดยที่ R แทนเส้นทางในแต่ละแถวของผลเฉลย X หรืออาจแสดงในรูปของเมทริกซ์ดังสมการที่ 21

$$X = \begin{bmatrix} C_{11} & C_{12} & \cdots & C_{1n} \\ \vdots & \vdots & \vdots & \vdots \\ C_{p1} & C_{p2} & \cdots & C_{pn} \\ \vdots & \vdots & \vdots & \vdots \\ C_{q1} & C_{q2} & \cdots & C_{qn} \\ \vdots & \vdots & \vdots & \vdots \\ C_{k1} & C_{k2} & \cdots & C_{kn} \end{bmatrix} = \begin{bmatrix} R_1 \\ \vdots \\ R_p \\ \vdots \\ R_q \\ \vdots \\ R_k \end{bmatrix} \quad (21)$$

โดยที่ R_p คือ เซตของกลุ่มลูกค้าที่ให้บริการโดยรถขนส่ง p และ R_q คือ เซตของกลุ่มลูกค้าที่ให้บริการโดยรถขนส่ง q ตามลำดับ การหาผลเฉลยที่เป็นไปได้ สร้างจากการสลับสับเปลี่ยนปมลูกค้าระหว่างคู่ของแถว R_p และ R_q $(R_p,\ R_q)$ ซึ่งแทนที่เซตย่อย $X_1 \subseteq R_p$ ที่มีขนาด $|X_1| \leq \lambda$ ด้วยเซตย่อย $X_2 \subseteq R_q$ ที่มีขนาด $|X_2| \leq \lambda$ เพื่อให้เกิดเซตของเส้นทางใหม่จำนวน 2 เส้นทาง คือ $R_p{}' = (R_p - X_1) \cup X_2, R_q{}' = (R_q - X_2) \cup X_1$ และผลเฉลยข้างเคียงใหม่ คือ $X = \{R_1,...R_p{}',...R_q{}',...,R_k\}$ ซึ่งผลเฉลยข้างเคียง $N_\lambda(X)$ ทั้งหมด คือ เซตของผลเฉลย X' ที่เป็นไปได้ และถูกสร้างขึ้นโดยวิธี LSD ซึ่งมีการกำหนดค่าจำนวนเต็ม λ

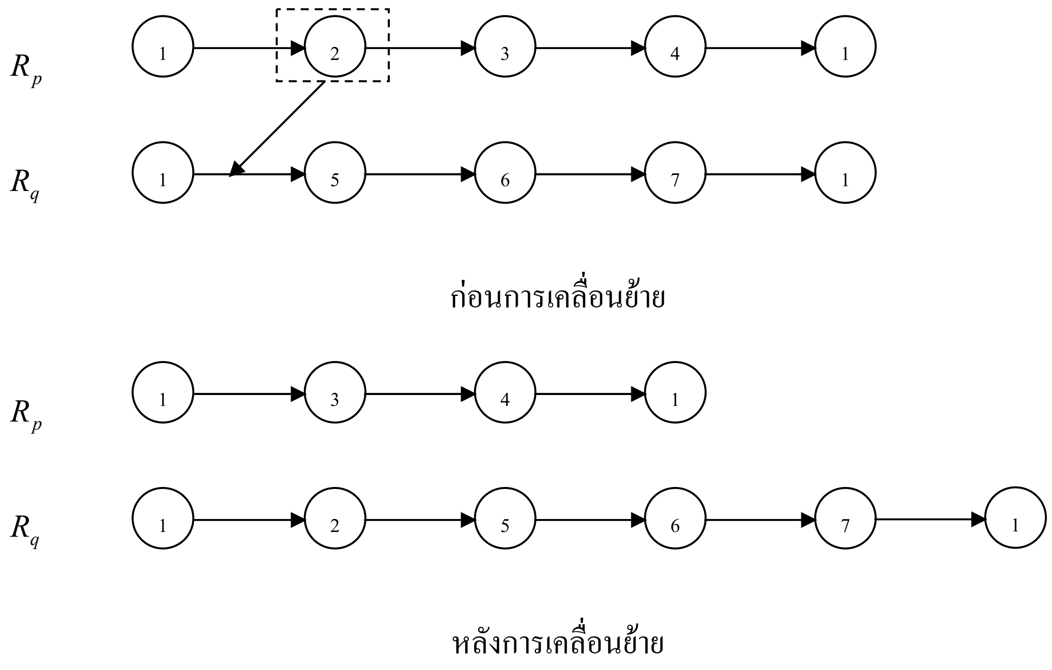
ลำดับการค้นหาผลเฉลยถูกเลือกจากการจัดคู่ที่เป็นไปได้ทั้งหมด (All Possible Combination) ของคู่ลำดับ (R_p, R_q) ซึ่งจำนวนที่เป็นไปได้ในการจับคู่ทั้งหมดเท่ากับ $\frac{K \times (K-1)}{2}$ ทางเลือก โดยที่ K คือ จำนวนเส้นทางทั้งหมด ตัวอย่างเช่น กรณีมี 3 เส้นทาง ประกอบด้วย R_1, R_2 และ R_3 เพราะฉะนั้นทางเลือกที่เป็นไปได้ทั้งหมด คือ $\frac{3 \times (3-1)}{2} = 3$ ได้แก่คู่ลำดับ (R_1, R_2), (R_1, R_3) และ (R_2, R_3) นอกจากนี้ในแต่ละคู่ลำดับ (R_p, R_q) ต้องกำหนดลำดับในการค้นหาผลเฉลย (Search Order) สำหรับการย้ายลูกค้าจากเส้นทางหนึ่งไปยังอีกเส้นทางหนึ่ง หรือสลับสับเปลี่ยนตำแหน่งกันในแต่ละคู่ลำดับ โดยเลือกใช้โครงสร้างผลเฉลยข้างเคียงที่มีค่า $\lambda = 1$ และ $\lambda = 2$ สาเหตุที่ไม่เลือกค่าที่มากไปกว่านี้เนื่องจากอาจทำให้ขั้นตอนวิธีการเสียเวลามาก เมื่อผลเฉลยที่เป็นไปได้มีจำนวนเพิ่มมากขึ้น การดำเนินการของวิธี λ-LSD ประกอบด้วย 2 กระบวนการหลักๆ ที่ดำเนินการระหว่าง 2 เส้นทาง คือ การย้ายลูกค้าบางปมออกจากเส้นทางหนึ่ง ไปยังอีกเส้นทางหนึ่ง (Shift Operation) และ การสลับสับเปลี่ยนลูกค้าบางปมระหว่างเส้นทาง 2 เส้นทาง (Exchange Operation) โดยการดำเนินการกำหนดด้วยคู่ลำดับ (0, 1) , (1, 0), (1, 1), (0, 2), (2, 0), (1, 2), (2, 1) และ (2, 2) หรือเรียกคู่ลำดับเหล่านี้ว่า “ตัวดำเนินการเคลื่อนย้าย (Move Operator)” โดยจำนวนของตัวดำเนินการขึ้นอยู่กับการกำหนดค่า λ ซึ่งในกรณีนี้กำหนดค่า $\lambda = 1$ และ $\lambda = 2$ จึงเกิดตัวดำเนินการทั้งสิ้น 8 ตัวด้วยกัน ซึ่งแต่ละตัวมีการดำเนินการที่แตกต่างกันดังต่อไปนี้

ตัวดำเนินการเคลื่อนย้าย (0, 1) สำหรับเส้นทาง (R_p, R_q) หมายถึง การย้ายลูกค้าจำนวน 1 ปม ออกจากเส้นทาง R_q ไปแทรกยังเส้นทาง R_p แสดงดังภาพที่ 10



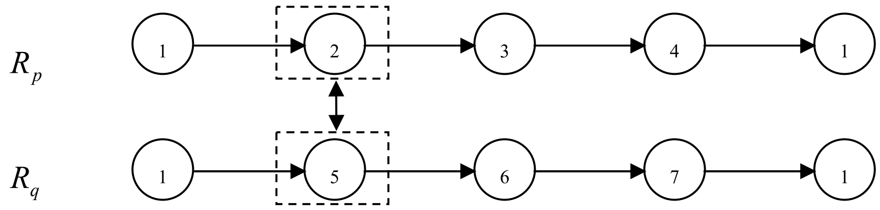
ภาพที่ 10 ตัวดำเนินการเคลื่อนย้าย (0, 1)

ตัวดำเนินการเคลื่อนย้าย (1, 0) หมายถึง การย้ายลูกค้าจำนวน 1 ปม ออกจากเส้นทาง R_p ไปแทรกยังเส้นทาง R_q

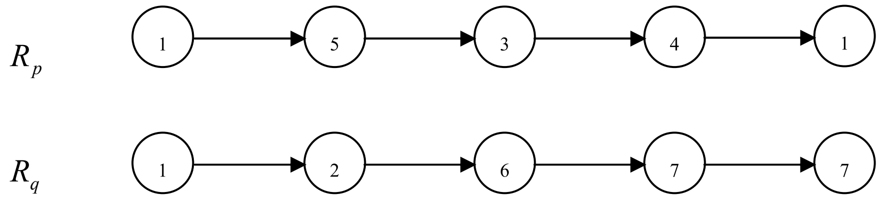


ภาพที่ 11 ตัวดำเนินการเคลื่อนย้าย (1, 0)

ตัวดำเนินการเคลื่อนย้าย (1, 1) หมายถึง การสลับสับเปลี่ยนลูกค้าจำนวน 1 ปม ระหว่างเส้นทาง R_p และเส้นทาง R_q



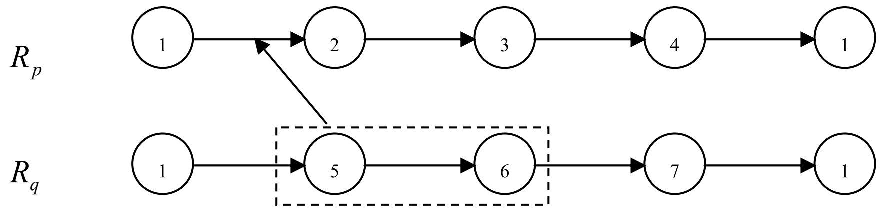
ก่อนการเคลื่อนย้าย



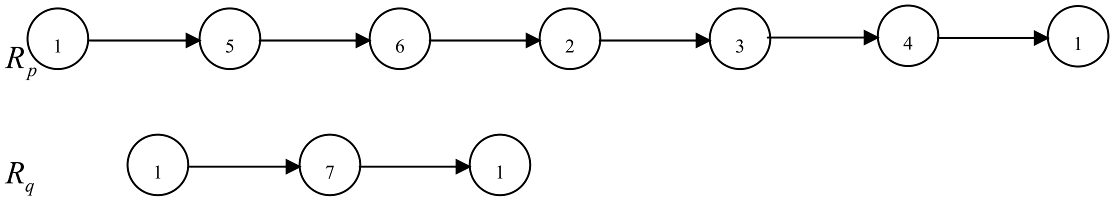
หลังการเคลื่อนย้าย

ภาพที่ 12 ตัวดำเนินการเคลื่อนย้าย (1, 1)

ตัวดำเนินการเคลื่อนย้าย (0, 2) หมายถึง การย้ายลูกค้าจำนวน 2 ปม ออกจากเส้นทาง R_q ไปแทรกยังเส้นทาง R_p



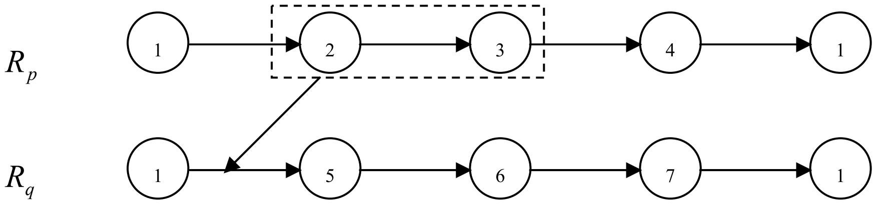
ก่อนการเคลื่อนย้าย



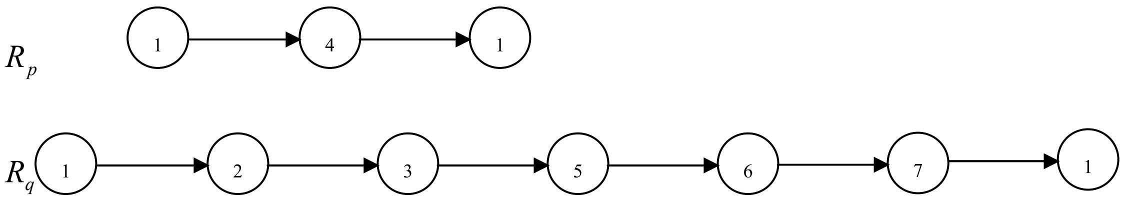
หลังการเคลื่อนย้าย

ภาพที่ 13 ตัวดำเนินการเคลื่อนย้าย (0, 2)

ตัวดำเนินการเคลื่อนย้าย (2, 0) หมายถึง การย้ายลูกค้าจำนวน 2 ปม ออกจากเส้นทาง R_p ไปแทรกยังเส้นทาง R_q



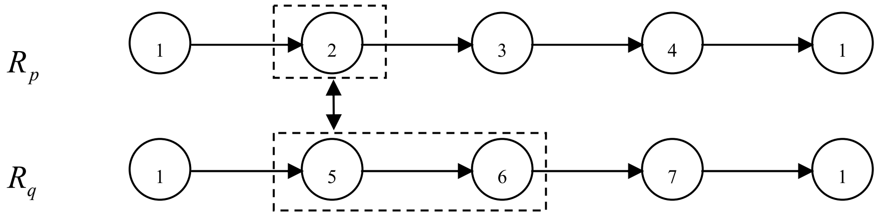
ก่อนการเคลื่อนย้าย



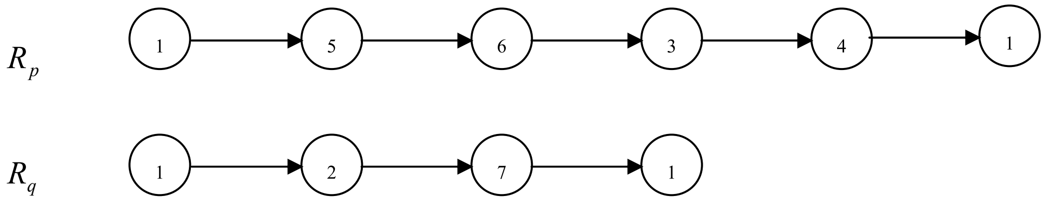
หลังการเคลื่อนย้าย

ภาพที่ 14 ตัวดำเนินการเคลื่อนย้าย (2, 0)

ตัวดำเนินการเคลื่อนย้าย (1, 2) หมายถึง การสลับสับเปลี่ยนลูกค้าจำนวน 2 ปม จาก เส้นทาง R_q และลูกค้าจำนวน 1 ปมจากเส้นทาง R_p



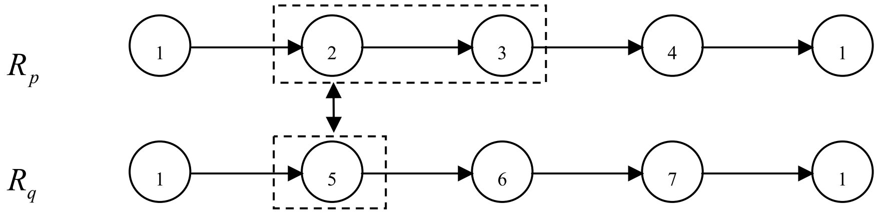
ก่อนการเคลื่อนย้าย



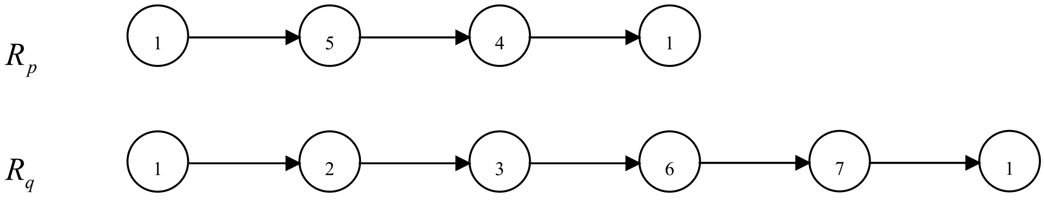
หลังการเคลื่อนย้าย

ภาพที่ 15 ตัวดำเนินการเคลื่อนย้าย (1, 2)

ตัวดำเนินการเคลื่อนย้าย (2, 1) หมายถึง การสลับสับเปลี่ยนลูกค้าจำนวน 1 ปม จาก เส้นทาง R_q และลูกค้าจำนวน 2 ปมจากเส้นทาง R_p



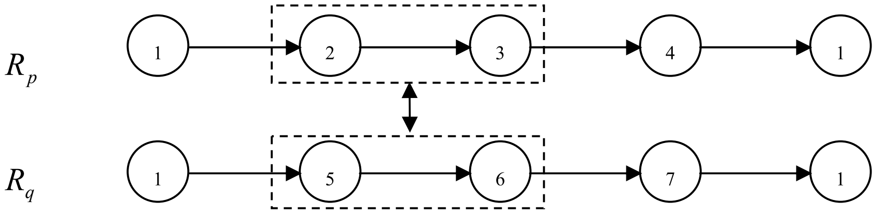
ก่อนการเคลื่อนย้าย



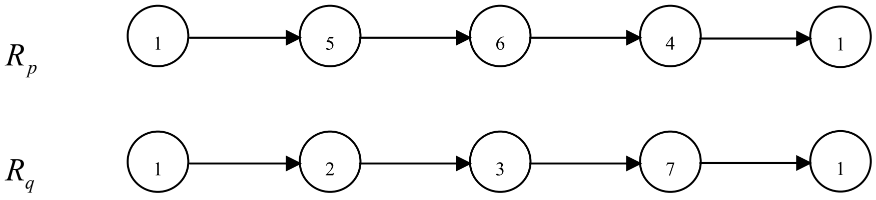
หลังการเคลื่อนย้าย

ภาพที่ 16 ตัวดำเนินการเคลื่อนย้าย (2, 1)

ตัวดำเนินการเคลื่อนย้าย (2, 2) หมายถึง การสลับสับเปลี่ยนลูกค้าจำนวน 2 ปม จากเส้นทาง R_q และลูกค้าจำนวน 2 ปมจากเส้นทาง R_p



ก่อนการเคลื่อนย้าย



หลังการเคลื่อนย้าย

ภาพที่ 17 ตัวดำเนินการเคลื่อนย้าย (2, 2)

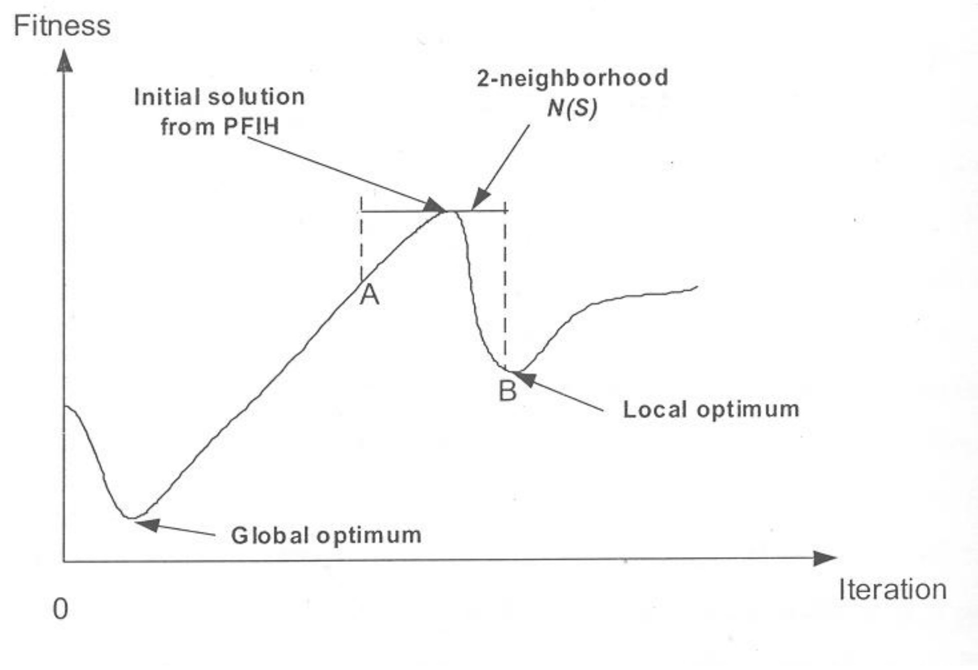
จากลักษณะการโยกย้ายและสลับสับเปลี่ยนของตัวดำเนินการดังภาพที่ 10-17 ในแต่ละครั้งของการดำเนินการ จะได้เส้นทางใหม่เกิดขึ้น 2 เส้น โดยทั้งสองเส้นทางจะต้องเป็นไปตามเงื่อนไข $D_k = \sum_{i=l}^{m} d_i \leq q_k$ สังเกตได้ว่าการใช้ตัวดำเนินการเคลื่อนย้าย (0, 1), (1, 0) และ (1, 1) น่าจะมี

ประสิทธิภาพเพียงพอแล้วสำหรับการเคลื่อนย้ายผลเฉลยจากหนึ่งไปยังอีกตัวหนึ่งภายในผลเฉลยข้างเคียงที่กำหนด อย่างไรก็ตามการใช้ตัวดำเนินการเคลื่อนย้าย ที่มีทั้งการโยกย้ายและสลับสับเปลี่ยนจำนวนปมลูกค้าระหว่างเส้นทาง 2 เส้น เช่น (0, 2), (2, 0), (1, 2), (2, 1) และ (2, 2) จะช่วยเพิ่มโอกาสในการหาผลเฉลยที่ดีขึ้นโดยไม่ขัดแย้งกับเงื่อนไข และไม่ทำให้ขั้นตอนวิธีการใช้เวลาในการค้นหาผลเฉลยมากเกินไป ในงานวิจัยนี้จึงนำหลักการการผสมผสานระหว่างตัวดำเนินการเคลื่อนย้ายแบบต่างๆ มาดำเนินการตามลำดับที่กำหนด โดยจัดลำดับการทำงานดังตารางที่ 5

ตารางที่ 5 ลำดับการทำงานของตัวดำเนินการเคลื่อนย้าย

ลำดับ	1	2	3	4	5	6	7	8
ตัวดำเนินการเคลื่อนย้าย	(0, 1)	(1, 0)	(1, 1)	(0, 2)	(2, 0)	(2, 1)	(1, 2)	(2, 2)

ตัวดำเนินการเหล่านี้จะถูกใช้ในการโยกย้ายและสลับสับเปลี่ยนตำแหน่งของลูกค้าระหว่างเส้นทาง (R_p, R_q) เพื่อสร้างกลุ่มเส้นทางใหม่ หรือผลเฉลยใหม่ ขั้นตอนต่อไป คือ การเลือกผลเฉลยที่มีการเคลื่อนย้ายโดยตัวดำเนินการทั้ง 8 ตัว วิธีการเลือกผลเฉลยที่ได้จากตัวดำเนินการที่ดีที่สุดจากกลุ่มของผลเฉลยที่มีอยู่ (Candidate Move) นั้นมีหลายวิธี อย่างไรก็ตามในงานวิจัยนี้เลือกใช้กลยุทธ์ในการเลือกตัวดำเนินการตัวแรกที่ให้ค่าฟังก์ชั่นวัตถุประสงค์ที่ดีกว่าผลเฉลยปัจจุบัน (ในกรณีนี้หมายถึงค่าเดินทางรวมที่ต่ำลง) เรียกกลยุทธ์นี้ว่า “ดีที่สุดเป็นตัวแรก (First Best)” ซึ่งกลยุทธ์นี้ ถึงแม้ไม่ได้ให้ผลเฉลยที่ดีที่สุดจากตัวดำเนินการทั้งหมด แต่ให้ผลเฉลยที่ยอมรับได้ภายในระยะเวลาอันรวดเร็ว หากใช้กลยุทธ์ที่ต้องดำเนินการทั้งหมดและเลือกผลเฉลยที่ดีที่สุด หรือเรียกกลยุทธ์นี้ว่า “ดีที่สุดจากทั้งหมด (Global Best) กลยุทธ์นี้ถึงแม้จะให้คำตอบที่ดีที่สุดแต่ใช้เวลาในการค้นหาค่อนข้างยาวนาน ซึ่งทำให้เสียเวลาโดยไม่จำเป็น อย่างไรก็ตามการเลือกกลยุทธ์แบบ FB อาจไม่ได้ให้คำตอบที่แย่กว่ากลยุทธ์แบบ GB เสมอไป แสดงดังภาพที่ 18

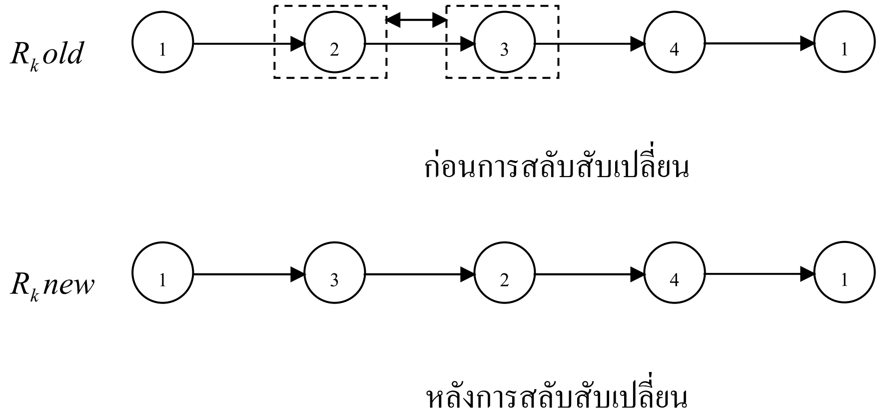


ภาพที่ 18 เปรียบเทียบวิธี FB และวิธี GB
ที่มา: ณกร (2548)

จากภาพสังเกตได้ว่าในกรณีที่เลือกใช้กลยุทธ์แบบ GB ทิศทางการหาผลเฉลย (Search Direction) จะค่อยๆ เคลื่อยย้ายจากจุด A มาที่จุด B ซึ่งผลเฉลยที่ได้ให้ค่าเหมาะสุดเฉพาะที่ ขั้นตอนวิธีการไม่สามารถหาผลเฉลยที่ดีกว่านี้ได้อีกภายในผลเฉลยข้างเคียงที่กำหนด $N(S)$ และเมื่อทำหาค้นหาจนครบรอบการวนซ้ำที่กำหนด ขั้นตอนวิธีการจะประกาศว่าจุด B คือ ผลเฉลยที่ให้ค่าฟังก์ชั่นวัตถุประสงค์ที่ดีที่สุด ซึ่งจากภาพที่ 18 พบว่าจุด B ไม่ใช่ผลเฉลยที่ให้ค่าเหมาะที่สุด เป็นเพียงค่าเหมาะสุดเฉพาะที่เท่านั้น ในอีกกรณีหนึ่ง หากเลือกใช้กลยุทธ์แบบ FB ขั้นตอนวิธีการอาจจะยอมรับผลเฉลยที่จุด A เป็นจุดแรกที่ให้ค่าฟังก์ชั่นวัตถุประสงค์ที่ดีกว่าผลเฉลยปัจจุบัน (Current Solution) จากนั้นขั้นตอนวิธีการจะค้นหาผลเฉลยที่ดีขึ้นต่อไป และเคลื่อนเข้าสู่ค่าเหมาะที่สุดแบบวงกว้าง (Global Optimum) ได้

หลังจากได้ผลเฉลยที่ดีที่สุดจากตัวดำเนินการเคลื่อนย้ายระหว่าง 2 เส้นทาง (R_p, R_q) แล้ว ขั้นตอนวิธีการจะทำการสลับสับเปลี่ยนตำแหน่งลูกค้าทีละ 1 คู่ หรือ 2 ปมใดๆ ภายในเส้นทางเดียวกันตั้งแต่ $R_1, R_2, ..., R_k$ เพื่อเพิ่มโอกาสในการหาผลเฉลยที่ดีกว่า โดยขั้นตอนนี้ไม่จำเป็นต้องคำนึงถึงเงื่อนไข $D_k = \sum_{i=l}^{m} d_i \leq q_k$ เนื่องจากผลเฉลยที่ได้จากขั้นตอนการเคลื่อนย้ายตัวดำเนินการ

ได้จำกัดเงื่อนไขนี้ไว้ในแต่ละเส้นทางแล้ว ขั้นตอนนี้จึงเป็นการทดลองสลับสับเปลี่ยนตำแหน่งลูกค้าภายในเส้นทางเดียวกันเท่านั้น แสดงดังภาพที่ 19



ภาพที่ 19 การสลับสับเปลี่ยนภายในเส้นทาง

จากภาพที่ 19 จำนวนผลเฉลยที่เป็นไปได้ทั้งหมดในพื้นที่การค้นหาผลเฉลย คือ $n!$ โดยที่ n คือ จำนวนลูกค้าทั้งหมด ที่ต้องการสลับสับเปลี่ยนภายในเส้นทาง R_k ตัวอย่างเช่น กรณีมีลูกค้าจำนวน 3 ปม ที่ต้องการสลับสับเปลี่ยน เพราะฉะนั้นทางเลือกที่เป็นไปได้ทั้งหมด คือ $3! = 3 \times 2 \times 1 = 6$ อย่างไรก็ตาม ในการเพิ่มประสิทธิภาพในการค้นหาผลเฉลยของขั้นตอนวิธีการต้องใช้โครงสร้างผลเฉลยข้างเคียงที่ดีด้วย อันได้แก่ โครงสร้างต้องง่ายต่อการทำงานในเชิงการเขียนโปรแกรม การประมวลผล และโครงสร้างจะต้องครอบคลุมผลเฉลยที่เป็นไปได้ทั้งหมด และนำไปสู่ค่าเหมาะที่สุด หรือค่าเหมาะที่สุดใกล้เคียง

ในกระบวนการทำงานจริงของขั้นตอนวิธีการ อาจไม่สามารถสร้างพื้นที่ในการค้นหาผลเฉลยที่เป็นไปได้ทั้งหมดได้ เนื่องจากจำนวนทางเลือกที่เพิ่มขึ้นแบบแฟคทอเรียล ซึ่งอาจต้องใช้เวลายาวนานในการหาผลเฉลยเมื่อจำนวนลูกค้าเพิ่มมากขึ้น แนวคิดในการลดพื้นที่ผลเฉลยจึงเป็นส่วนหนึ่งที่สำคัญ โดยกำหนดเป็นโครงสร้างในการสลับสับเปลี่ยนตำแหน่งของลูกค้าภายในเส้นทาง ในงานวิจัยนี้ใช้วิธีการสลับสับเปลี่ยนแบบทีละคู่ (Pair-Wise Exchange) นั่น คือ การสลับสับเปลี่ยนตำแหน่งระหว่างลูกค้า 2 ปมใดๆ ภายในเส้นทางเดียวกัน ซึ่งจำนวนทางเลือกของผลเฉลยเท่ากับ $\frac{n(n-1)}{2}$ จากปัญหาเดิมที่มีลูกค้าจำนวน 3 ปม ภายในเส้นทาง ทางเลือกที่เป็นไปได้ทั้งหมดเท่ากับ $\frac{3(3-1)}{2} = 3$ ซึ่งในกรณีสามารถลดพื้นที่ในการค้นหาผลเฉลยได้ถึง 50% จากนั้น

ขั้นตอนวิธีการจะทำการสลับเปลี่ยนตำแหน่งลูกค้าทีละคู่ไปเรื่อยๆ จนกว่าจะได้ค่าฟังก์ชั่นวัตถุประสงค์ที่ดีขึ้น หากไม่สามารถปรับปรุงผลเฉลยได้ ขั้นตอนวิธีการจะยังคงผลเฉลยปัจจุบันไว้ วิธีการ λ-LSD แสดงรายละเอียดดังต่อไปนี้

ขั้นตอนที่ 1 กำหนดให้ผลเฉลย X สำหรับ VRPTW ที่ได้จาก MPFIH เป็นผลเฉลยปัจจุบัน

ขั้นตอนที่ 2 เลือกผลเฉลย $X' \in N_{\lambda}(X)$ ตามกลยุทธ์การเคลื่อนย้ายตัวดำเนินการแบบ FB

ขั้นตอนที่ 3 ถ้า $F(X') < F(X)$, ให้ยอมรับผลเฉลย X' และดำเนินการต่อในขั้นตอนที่ 2 ถ้าไม่ใช่ ให้ดำเนินการต่อในขั้นตอนที่ 4

ขั้นตอนที่ 4 ถ้าผลเฉลยข้างเคียงภายในพื้นที่ $N_{\lambda}(X)$ ถูกค้นหาจนครบทุกความเป็นไปได้ และไม่มีผลเฉลยใดให้ค่า $F(X')$ ที่ต่ำกว่าผลเฉลยปัจจุบัน ให้ดำเนินการตามขั้นตอนที่ 5 ถ้าไม่ใช่ ให้ดำเนินการต่อในขั้นตอนที่ 2

ขั้นตอนที่ 5 สิ้นสุดวิธีการ λ-LSD

4.3.2 ขั้นตอนวิธีการ TS จากปัญหาที่มักพบในขั้นตอนการค้นหาผลเฉลยข้างเคียง คือ การยอมรับค่าเหมาะที่สุดเฉพาะที่ ในงานวิจัยนี้เลือกใช้ขั้นตอนวิธีการ TS ในการป้องกันการเกิดค่าเหมาะที่สุดเฉพาะที่ หรือค่าต่ำสุดเฉพาะที่ เนื่องจากวิธีการ TS เป็นเทคนิคการประมาณค่าสำหรับการคำนวณเชิงคณิตศาสตร์ที่ดี เมื่อเทียบกับเทคนิคอื่นๆ ด้วยความยืดหยุ่น และมีประสิทธิภาพเหนือกว่าวิธีการคลาสสิกอื่นๆ ดังที่ได้นำเสนอไว้ในการตรวจเอกสาร จากงานวิจัยของ Glover (1989) ใช้วิธีการ TS ในการป้องกันเพื่อไม่ให้เกิดการเลือกซ้ำผลเฉลยเดิมที่ให้ค่าฟังก์ชั่นวัตถุประสงค์ที่แย่ แต่สำหรับในงานวิจัยนี้ นำมาประยุกต์ใช้ในวิธีการที่ตรงกันข้ามเพื่อเป้าหมายที่แตกต่างกัน ซึ่ง Golver (1989) ต้องการลดเวลาในการค้นหาผลเฉลย โดยการทาบูผลเฉลยที่ไม่ดีไว้ ซึ่งขั้นตอนวิธีการจะไม่เลือกผลเฉลยนั้นเลยจนกว่าจะครบรอบการทาบูที่กำหนดไว้ ในทางตรงข้าม งานวิจัยนี้ต้องการใช้วิธีการ TS ในการเปิดโอกาสให้ผลเฉลยอื่นๆ ได้รับการคัดเลือก ถึงแม้จะมีค่าฟังก์ชั่นวัตถุประสงค์ที่แย่ลงก็ตาม ทั้งนี้เพื่อป้องกันการเกิดปัญหาค่าต่ำสุดเฉพาะที่ดังที่ได้กล่าวไว้ในเบื้องต้น โดยใช้งานร่วมกับการค้นหาผลเฉลยแบบ λ-LSD คือ เมื่อเจอค่าต่ำสุดเฉพาะที่แล้วจึงเลื่อนไปตามผลเฉลยข้างเคียงอื่นๆ ใน $N_{\lambda}(X)$ หากพบว่าผลเฉลยนั้นให้ค่าฟังก์ชั่นวัตถุประสงค์ที่ดีขึ้น จะทำการปรับปรุงให้เป็นผลเฉลยปัจจุบัน และผลเฉลยนั้นจะถูกป้องกันด้วยวิธีการทาบู เพื่อไม่ให้มีการเลือกซ้ำได้อีกจนกว่าจะครบรอบการทาบูที่กำหนด หรือตามเงื่อนไขในปล่อยค่าทาบู หรือครบรอบการวนซ้ำที่กำหนด ซึ่งเป็นไปตามเงื่อนไขการหยุดที่กำหนดไว้

ส่วนสำคัญที่สุดของขั้นตอนวิธีการ TS คือ การออกแบบหน่วยความทรงจำ (History Record; H) เพื่อใช้ในการบันทึกค่าที่ถูกทาบูไว้ในรายการทาบู ซึ่งในงานวิจัยนี้ได้นำเสนอโครงสร้างหน่วยความทรงจำทาบูแบบระยะยาว (Long-Term Memory) ประกอบด้วย ส่วนของการบันทึกตัวแปรที่ถูกทาบูในรอบการวนซ้ำที่ผ่านมา (Recently-Based Memory) และส่วนของการบันทึกความถี่ที่ตัวแปรนั้นๆ ถูกเลือกจากรอบการวนซ้ำทั้งหมด (Frequency-Based Memory) ตัวอย่างการทำงานของหน่วยความทรงจำทาบู ในการสลับสับเปลี่ยน (Swap) ลูกค้าจำนวน 7 ปม ในเส้นทางเดียวกัน โดยกำหนดจำนวนรอบการทาบูเท่ากับ 3 เป็นดังนี้

เริ่มต้น X_0 = [4, 7, 12, 26, 5, 23, 15] โดยที่ $F(X_0) = 391$ และสร้างหน่วยความทรงจำในรูปของเมทริกซ์ว่างขนาด 7x7

Recently

	1(4)	2(7)	3(12)	4(26)	5(5)	6(23)	7(15)
1(4)	-						
2(7)		-					
3(12)			-				
4(26)				-			
5(5)					-		
6(23)						-	
7(15)							-

Frequency

ภาพที่ 20 โครงสร้างหน่วยความทรงจำทาบู (เริ่มต้น)

รอบที่ 1 ดำเนินการสลับเปลี่ยนดังตารางที่ 6 ค่าฟิตเนสฟังก์ชั่น (Fitness Function) ที่ดีที่สุดคือ 3 ซึ่งเป็นการสลับสับเปลี่ยนระหว่างลูกค้าปมที่ 4 และ 15 โดยที่ X_1 = [15, 7, 12, 26, 5, 23, 4] และ $F(X_1)$=388 ค่าฟิตเนสฟังก์ชั่น f^*=391-388=3 หน่วยความทรงจำทาบูจะเริ่มบันทึกตัวแปรที่ถูกเลือก โดยในส่วนสามเหลี่ยมบน (Recently) สำหรับบันทึกจำนวนรอบการทาบูตัวแปรที่ถูกเลือก และสามเหลี่ยมล่าง (Frequency) สำหรับการบันทึกความถี่ที่ตัวแปรนั้นๆ ถูกเลือก แสดงดังภาพที่ 20

ตารางที่ 6 ผลการสลับสับเปลี่ยนในรอบที่ 1

ปมที่ทำการสลับสับเปลี่ยน	ค่าฟิตเนสฟังก์ชั่น $f^* = F(X) - F(X')$
(4,15)	3
(4,5)	-13
(7,15)	-13
(7,26)	-21
(12,5)	-31
(26,23)	-45
(23,15)	-61
(5,15)	-66
(7,23)	-69
(5,23)	-92
(7,5)	-119
(4,12)	-120
(12,26)	-122
(4,7)	-130
(7,12)	-131
(12,15)	-140
(26,15)	-149
(26,5)	-158
(4,23)	-172
(12,23)	-207
(4,26)	-224

	Recently						
	1(4)	2(7)	3(12)	4(26)	5(5)	6(23)	7(15)
1(4)	-						3
2(7)		-					
3(12)			-				
4(26)				-			
5(5)					-		
6(23)						-	
7(15)	1						-

Frequency

ภาพที่ 21 หน่วยความทรงจำทาบู (รอบที่ 1)

รอบที่ 2 ดำเนินการสลับเปลี่ยนดังตารางที่ 7 ค่าฟิตเนสฟังก์ชั่นที่ดีที่สุด คือ 3 ซึ่งเป็นการสลับสับเปลี่ยนระหว่างลูกค้าปมที่ 26 และ 23 โดยที่ X_2 = [15, 7, 12, 23, 5, 26, 4] และ $F(X_2)$ = 385 ค่าฟิตเนสฟังก์ชั่น $f*$ = 388-385 = 3 หน่วยความทรงจำทาบูจะบันทึกตัวแปรที่ถูกเลือกตัวใหม่ และลดจำนวนรอบการทาบูของตัวแปรที่ถูกเลือกในรอบก่อนหน้าลง 1 รอบ ส่วนการบันทึกความถี่ยังคงดำเนินการเช่นเดิม โดยไม่มีการลดลงสำหรับตัวแปรที่ถูกเลือกก่อนหน้า และจะต้องเพิ่มขึ้นเมื่อในรอบการวนซ้ำอื่นๆ ย้อนกลับมาเลือกตัวแปรนั้นๆ อีกครั้งหนึ่ง

	Recently						
	1(4)	2(7)	3(12)	4(26)	5(5)	6(23)	7(15)
1(4)	-						2
2(7)		-					
3(12)			-				
4(26)				-		3	
5(5)					-		
6(23)				1		-	
7(15)	1						-

Frequency

ภาพที่ 22 หน่วยความทรงจำทาบู (รอบที่ 2)

ตารางที่ 7 ผลการสลับสับเปลี่ยนในรอบที่ 2

ปมที่ทำการสลับสับเปลี่ยน	ค่าฟิตเนสฟังก์ชั่น $f^* = F(X) - F(X')$
(4,7)	-45
(4,12)	-3
(4,26)	-228
(4,5)	-5
(4,23)	-89
(4,15)	รอบการทาบู = 3-1=2
(7,12)	-99
(7,26)	-28
(7,5)	-138
(7,23)	-22
(7,15)	-150
(12,26)	-123
(12,5)	-31
(12,23)	-246
(12,15)	-270
(26,5)	-159
(26,23)	3
(26,15)	-150
(5,23)	-109
(5,15)	-79
(23,15)	-145

รอบที่ 3 ดำเนินการสลับเปลี่ยนดังตารางที่ 8 ค่าฟิตเนสฟังก์ชั่น (Fitness Function) ที่ดีที่สุด คือ -3 ซึ่งเป็นการสลับสับเปลี่ยนระหว่างลูกค้าปมที่ 4 และ 12 โดยที่ X_3 = [15, 7, 4, 23, 5, 26, 12] และ $F(X_3)$ =388 ค่าฟิตเนสฟังก์ชั่น f^*=385-388= -3 สังเกตได้ว่า ขั้นตอนวิธีการจะยอมเลือกค่าฟิตเนสฟังก์ชั่นที่แย่ลงในแต่ละรอบการวนซ้ำ เพื่อเปิดโอกาสในการเลือกผลเฉลยข้างเคียงอื่นๆ

ที่อาจนำไปสู่ผลเฉลยที่ดีกว่าปัจจุบัน และการบันทึกความทรงจำทาบูจะดำเนินการเช่นเดิมไปเรื่อยๆ จนกว่าจะเป็นไปตามเงื่อนไขของการหยุด

ตารางที่ 8 ผลการสลับสับเปลี่ยนในรอบที่ 3

ปมที่ทำการสลับสับเปลี่ยน	ค่าฟิตเนสฟังก์ชั่น $f^* = F(X) - F(X')$
(4,7)	-61
(4,12)	-3
(4,26)	-83
(4,5)	-23
(4,23)	-237
(4,15)	รอบการทาบู = 2-1=1
(7,12)	-74
(7,26)	-25
(7,5)	-138
(7,23)	-28
(7,15)	-150
(12,26)	-252
(12,5)	-9
(12,23)	-120
(12,15)	-187
(26,5)	-132
(26,23)	รอบการทาบู = 3-1=2
(26,15)	-141
(5,23)	-138
(5,15)	-73
(23,15)	-156

	Recently						
	1(4)	2(7)	3(12)	4(26)	5(5)	6(23)	7(15)
1(4)	-						2
2(7)		-					
3(12)			-				
4(26)				-		3	
5(5)					-		
6(23)				1		-	
7(15)	1						-

Frequency

ภาพที่ 23 หน่วยความทรงจำทาบู (รอบที่ 3)

จากตัวอย่างแสดงรอบการวนซ้ำเพียง 3 รอบ ในการทำงานจริงอาจกำหนดรอบการวนซ้ำให้มีจำนวนมากขึ้น เพื่อหาผลเฉลยที่ดีที่สุด และเงื่อนไขในการหยุดขั้นตอนวิธีการ มี 2 กรณีด้วยกัน คือ เมื่อครบรอบการวนซ้ำที่กำหนด และเมื่อค่า $f*$ ของการสลับสับเปลี่ยนลูกค้าคู่ใดคู่หนึ่ง มีค่าที่ดีกว่าการสลับสับเปลี่ยนในคู่อื่นๆ เป็นจำนวนรอบที่กำหนด ถึงแม้ว่าระหว่างรอบนั้นตัวแปรดังกล่าวจะถูกทาบูไว้ ก็สามารถเลือกให้เป็นผลเฉลยที่ดีที่สุดได้ โดยขั้นตอนวิธีการ TS มีรายละเอียดดังต่อไปนี้

ขั้นตอนที่ 1 กำหนดจำนวนรอบการวนซ้ำ, r, จำนวนรอบการทาบู, r_{ts} และจำนวนรอบในการปล่อยผลเฉลยที่ดีออกจากการทาบู, r_{rel}

ขั้นตอนที่ 2 สร้างผลเฉลยเริ่มต้น X ที่ได้จากวิธีการ λ-LSD และสร้างโครงสร้างหน่วยความจำ H เพื่อการบันทึกค่าทาบู

ขั้นตอนที่ 3 กำหนดค่า $r = r+1$ และสร้างเซตย่อย $V*$ ของผลเฉลยใน $N(X,r)$ เป็นผลเฉลยข้างเคียงสำหรับการวนซ้ำครั้งที่ r

ขั้นตอนที่ 4 เลือกผลเฉลย $X' \in N(X,r)$ จากพื้นที่ค้นหาผลเฉลยที่กำหนดเลือกค่า $Y \in V*$ ที่เป็นไปตามฟังก์ชันวัตถุประสงค์หรือเงื่อนไข ปรับปรุงผลเฉลย และกำหนดให้ $X = Y$

ขั้นตอนที่ 5 ถ้า $F(X') < F(X)$, ให้ยอมรับผลเฉลย X'

ขั้นตอนที่ 6 สิ้นสุดวิธีการ TS เมื่อเป็นไปตามเงื่อนไขของการหยุด ไม่เช่นนั้นให้กลับไปดำเนินตามขั้นตอนที่3

สรุปขั้นตอนวิธีการในการแก้ปัญหา RVRPTW ดังต่อไปนี้

ขั้นตอนที่ 1 สร้างสถานการณ์พื้นฐาน s_b โดยสุ่มค่า t_{s_b} จากการแจกแจงแบบเอกรูป $U[l_{ij}, u_{ij}]$ และกำหนดพารามิเตอร์ที่จำเป็นต้องใช้สำหรับ MPFIH, λ-LSD และ TS

ขั้นตอนที่ 2 สร้างสถานการณ์ที่เป็นไปได้ทั้งหมด โดยกำหนดเปอร์เซ็นต์ของความไม่แน่นอน β โดยที่ $t_s \in [(1-\beta)t_{s_b}, (1+\beta)t_{s_b}]$

ขั้นตอนที่ 3 ในแต่ละสถานการณ์ s สร้างผลเฉลยเริ่มต้นด้วยวิธี MPFIH

ขั้นตอนที่ 4 ค้นหาผลเฉลยต่ำสุด $Z_s = \min\{F(x,t_s) : x \in X\}$ โดยใช้วิธี λ-LSD และ TS

ขั้นตอนที่ 5 ค้นหาผลเฉลยสูงสุด จากสถานการณ์ s ที่เป็นไปได้ทั้งหมด

$$g_s(x) = \max\{F(x,t^s) : s \in S\} \text{ โดยที่ } g_s(x) = \begin{cases} f(x,t_s) \\ f(x,t_s) - Z_s \\ \dfrac{f(x,t_s) - Z_s}{Z_s} \end{cases} \text{ และ } x \in X$$

ขั้นตอนที่ 6 หาผลเฉลย x ซึ่งเป็นผลเฉลยเชิงทนทาน (Robust Solution)

$$Z = \min\left\{h_x(S) \mid g_s(X) \le h_x(S), s \in S; X \in \bigcap_{s \in S} F_s\right\}$$

5. ออกแบบการทดลอง (Design of Experiments)

5.1 การเลือกใช้ข้อมูลในการทดลอง งานวิจัยนี้ เลือกใช้ข้อมูลจากปัญหาเทียบเคียง (Benchmarking Problem) สำหรับปัญหา VRPTW ที่สร้างขึ้นโดย Marius Solomon (Solomon, 2005) จำนวน 56 ตัวอย่าง แบ่งเป็น 6 ประเภท ดังตารางที่ 9 ซึ่งในแต่ละตัวอย่างประกอบด้วย คลังสินค้าจำนวน 1 เมือง และลูกค้าจำนวน 100 เมือง สำหรับปัญหา VRPTW ของ Solomon นี้เป็นที่นิยมสำหรับนักวิจัยที่พัฒนาขั้นตอนวิธีการและต้องการเปรียบเทียบผล Solomon ได้แบ่งปัญหานี้ออกเป็น 6 ประเภท โดยการออกแบบตามปัจจัยต่างๆ ที่ส่งผลต่อพฤติกรรมของขั้นตอนวิธีการจัดเส้นทาง และการจัดตารางการเดินรถขนส่ง ซึ่งปัจจัยต่างๆ ประกอบด้วย ตำแหน่งและการกระจายของข้อมูล ข้อจำกัดความสามารถของรถขนส่งในการให้บริการลูกค้า เปอร์เซ็นต์ข้อจำกัดทางด้านเวลาที่กำหนดโดยลูกค้า ความเข้มงวดและตำแหน่งของกรอบเวลา

ตารางที่ 9 ประเภทของปัญหา VRPTW โดย Solomon

ประเภท	จำนวน
R1	12
C1	9
RC1	8
R2	11
C2	8
RC2	8

ปัญหาในกลุ่ม R1 และ R2 ถูกกำหนดตำแหน่งและการกระจายอย่างสุ่ม กลุ่ม C1 และ C2 มีการแบ่งกลุ่มลูกค้าแล้ว และกลุ่ม RC1 และ RC2 เป็นการผสมผสานระหว่างแบบสุ่มและแบบจัดกลุ่มแล้ว ปัญหาในกลุ่ม R1, C1 และ RC1 เป็นลักษณะการจัดตารางเดินรถแบบสั้น (Short Scheduling Horizon) โดยมีจำนวนลูกค้าไม่มากภายในเส้นทางหนึ่งๆ (ประมาณ 5 ถึง 10 เมือง) ส่วนในกลุ่ม R2, C2 และ RC2 เป็นลักษณะการจัดตารางเดินรถแบบยาว (Long Scheduling Horizon) โดยมาลูกค้าจำนวนมากอยู่ในเส้นทางหนึ่งๆ (มากกว่า 30 เมือง) ที่ให้บริการโดยรถขนส่งเพียงคันเดียว การกำหนดตำแหน่งที่ตั้งของลูกค้าจะเหมือนกันสำหรับปัญหาเดียวกันในแต่ละกลุ่ม R, C และ RC ต่างกันที่การกำหนดช่วงกว้างของกรอบเวลา ซึ่งบางกลุ่มจะมีช่วงของกรอบเวลาแคบมาก โดยกำหนดเป็นเปอร์เซ็นต์ของกรอบเวลา คือ 25%, 50%, 75% และ 100%

ปัญหามีขนาดลูกค้าจำนวน 100 เมือง โดยกำหนดให้เวลาเดินทางในแต่ละเมืองเท่ากับระยะทางแบบยุคลิด (Euclidean Distance) $t_{ij} = \sqrt{x^2 + y^2}$ ในกรณีที่ต้องการพิจารณาปัญหาที่มีขนาดเล็กลง เช่น 25 หรือ 50 เมือง สามารถเลือกได้จากลูกค้าในลำดับที่ 25 และ 50 เมืองแรกตามลำดับ โดยทุกปัญหาถูกจัดให้อยู่ในรูปแบบเดียวกัน คือ มีจำนวนแถวเท่ากับจำนวนคลังสินค้าและจำนวนลูกค้า และ 7 สดมภ์ แสดงตัวอย่างการจัดรูปแบบปัญหา VRPTW ในตารางที่ 10 ปัญหาที่ใช้ในการทดลองสำหรับการหาผลเฉลยเชิงทนทาน เลือกใช้ปัญหาจำนวน 6 ตัวอย่าง ประกอบด้วย R101, C101, RC101, R201, C201 และ RC201 โดยพิจารณาที่ขนาดลูกค้า 25 เมือง และปัญหาที่ใช้ในการเปรียบเทียบค่าเหมาะที่สุด แบบใช้สถานการณ์เดียว ($s = 1$) เลือกใช้ปัญหาจำนวน 168 ตัวอย่าง แบ่งเป็น R101-R112, C101-C109, RC101-RC108, R201-R211, C201-C208 และ RC201-RC208 โดยในแต่ละตัวอย่างแบ่งเป็นจำนวนลูกค้าขนาด 25, 50 และ 100 เมือง เช่น R101.25, R101.50 และ R101.100 เป็นต้น

ตารางที่ 10 ตัวอย่างปัญหา VRPTW จำนวน 25 เมือง (ความสามารถบรรทุกสินค้า =100 หน่วย)

ลำดับที่	ตำแหน่งที่ตั้ง		ความต้องการสินค้า	กรอบเวลา		เวลาในการให้บริการ
	X	Y		เวลามาถึงเร็วสุด	เวลามาถึงช้าสุด	
1	70	70	0	0	24	0
2	33	78	20	2	10	1
3	59	52	20	1	12	1
4	10	137	30	1	5	1
5	4	28	10	3	7	1
6	25	26	20	10	17	1
7	86	37	10	12	18	1
8	1	109	10	15	22	1
9	6	135	40	18	22	1
10	32	79	20	1	5	1
11	24	26	20	2	3	1
12	86	36	20	15	18	1
13	95	35	10	19	22	1
14	63	50	10	10	24	1
15	100	106	10	2	6	1
16	99	112	20	5	9	1
17	36	135	10	6	10	1
18	57	59	10	10	12	1
19	8	124	10	11	18	1
20	85	106	20	12	19	1
21	103	69	30	3	5	1
22	109	131	20	8	10	1
23	43	140	10	9	18	1
24	115	134	30	22	24	1
25	98	70	10	5	10	1
26	112	67	10	6	18	1

5.2 การเลือกค่าพารามิเตอร์ S และ β กำหนดให้ S มีขนาด 10, 20, 40 และ 80 โดยที่ β มีค่าเท่ากับ 0.2, 0.4, 0.6 และ 0.8 ในการศึกษาพฤติกรรมของขั้นตอนวิธีการในการหาผลเฉลยเชิงทนทาน

โดยการออกแบบการทดลองสามารถสรุปได้ดังนี้

ตารางที่ 11 การออกแบบการทดลองในการหาผลเฉลยเชิงทนทาน

ปัญหา	$\|S\|$	β	ปัญหา	$\|S\|$	β	ปัญหา	$\|S\|$	β
R101	10	0.2	C101	10	0.2	RC101	10	0.2
	10	0.4		10	0.4		10	0.4
	10	0.6		10	0.6		10	0.6
	10	0.8		10	0.8		10	0.8
	20	0.2		20	0.2		20	0.2
	20	0.4		20	0.4		20	0.4
	20	0.6		20	0.6		20	0.6
	20	0.8		20	0.8		20	0.8
	40	0.2		40	0.2		40	0.2
	40	0.4		40	0.4		40	0.4
	40	0.6		40	0.6		40	0.6
	40	0.8		40	0.8		40	0.8
	80	0.2		80	0.2		80	0.2
	80	0.4		80	0.4		80	0.4
	80	0.6		80	0.6		80	0.6
	80	0.8		80	0.8		80	0.8

ตารางที่ 11 (ต่อ)

ปัญหา	$\|S\|$	β	ปัญหา	$\|S\|$	β	ปัญหา	$\|S\|$	β
R201	10	0.2	C201	10	0.2	RC201	10	0.2
	10	0.4		10	0.4		10	0.4
	10	0.6		10	0.6		10	0.6
	10	0.8		10	0.8		10	0.8
	20	0.2		20	0.2		20	0.2
	20	0.4		20	0.4		20	0.4
	20	0.6		20	0.6		20	0.6
	20	0.8		20	0.8		20	0.8
	40	0.2		40	0.2		40	0.2
	40	0.4		40	0.4		40	0.4
	40	0.6		40	0.6		40	0.6
	40	0.8		40	0.8		40	0.8
	80	0.2		80	0.2		80	0.2
	80	0.4		80	0.4		80	0.4
	80	0.6		80	0.6		80	0.6
	80	0.8		80	0.8		80	0.8

6. วัดสมรรถนะของวิธีการเข้าสู่ผลเฉลย (Approach Performance Measure)

6.1 ตัววัดสมรรถนะและวิธีการเข้าสู่ผลเฉลยเปรียบเทียบ (Performance Measure and Benchmarking Approaches) การวัดสมรรถนะของวิธีการเข้าสู่ผลเฉลยโดยเปรียบเทียบระหว่างวิธีการเข้าสู่ผลเฉลยแบบการสร้างสถานการณ์ (Scenario-Based Approach) ที่ได้นำเสนอไว้ในงานวิจัยนี้ กับวิธีการเข้าสู่ผลเฉลยแบบกรณีแย่ที่สุด (Worst-Case Approach) หรือการใช้สถานการณ์ที่แย่ที่สุดเป็นตัวแทนของความไม่แน่นอน และวิธีการเข้าสู่ผลเฉลยแบบคาดหวัง (Expected-Based Approach) โดยผลเฉลยที่ได้ถูกเลือกมาจากค่าพารามิเตอร์ของความไม่แน่นอนที่มีโอกาสเกิดขึ้นบ่อยมากที่สุด หรือเป็นหลักการหนึ่งในการแก้ปัญหาเชิงสโทแคสติก

โดยกำหนดตัวชี้วัดสมรรถนะด้วยค่าสัดส่วนสัมพัทธ์ (Relative Ratio) ซึ่งเป็นค่าที่แสดงเวลาเดินทางเชิงทนทานเทียบกับกับเวลาเดินทางเหมาะที่สุดในแต่ละสถานการณ์แบบสัมพัทธ์

$R = \frac{Z_r - Z_s}{Z_s}$ โดยที่ Z_r คือ ค่าเหมาะที่สุดเชิงทนทาน สำหรับการทดลองในหัวข้อนี้เลือกใช้เกณฑ์ตัดสินสภาพทนทานแบบสัมพัทธ์ หรือ Z_R เป็นค่าเหมาะที่สุดเชิงทนทาน และ Z_s คือ ค่าเหมาะที่สุดตามวัตถุประสงค์ของปัญหา VRPTW ในแต่ละสถานการณ์ โดยค่า R นี้แสดงให้เห็นว่า เมื่อใช้ผลเฉลยเชิงทนทานในการจัดเส้นทางเดินรถขนส่ง และสถานการณ์ต่างๆ เกิดขึ้นจริง ผลเฉลยเชิงทนทานให้ค่าเวลาเดินทางที่แตกต่างไปจากค่าเดินทางเหมาะที่สุดสำหรับสถานการณ์นั้นๆ มากน้อยเพียงใด โดยเมื่อ $R = 0$ แสดงว่าผลเฉลยเชิงทนทานมีสมรรถนะสูง และเมื่อ R มีค่าเพิ่มขึ้น แสดงให้เห็นถึงสมรรถนะที่ด้อยลง

6.2 วิธีการเปรียบเทียบ (Comparing Method) วิธีการเปรียบเทียบโดยเลือกตัวอย่างปัญหา VRPTW จำนวน 1 ปัญหา ที่มีขนาดลูกค้า 25 เมือง 1 คลังสินค้า ความต้องการสินค้าโดยรวมเท่ากับ 430 หน่วย และความสามารถในการบรรทุกสินค้าเท่ากับ 100 หน่วย รายละเอียดของปัญหาแสดงดังตารางที่ 10 และใช้วิธีการสร้างแบบจำลองสถานการณ์ (Simulation Modeling) โดยทดลองเปลี่ยนค่าพารามิเตอร์ β และ $|S|$ เพื่อศึกษาสมรรถนะของวิธีการเข้าสู่ผลเฉลยแบบต่างๆ ประกอบด้วย แบบการสร้างสถานการณ์ แบบใช้กรณีแย่ที่สุด และแบบค่าคาดหวัง โดยกำหนดตัวชี้วัด คือ ค่าสัดส่วนสัมพัทธ์ R

การกำหนดค่าพารามิเตอร์ β ซึ่งเป็นตัวแสดงเปอร์เซ็นต์ความไม่แน่นอนของเวลาเดินทาง เช่น เมื่อการจราจรสะดวกค่า β จะต่ำ และมีค่าสูงขึ้นเมื่อการจราจรมีความหนาแน่นอันเกิดจากอุบัติเหตุในเส้นทางใดเส้นทางหนึ่ง หรือสภาวะอากาศที่ไม่ดี เป็นต้น ในการทดลองนี้กำหนดเปอร์เซ็นต์ β เท่ากับ 0.2, 0.4, 0.6 และ 1.0 โดยเวลาเดินทางต่ำสุดเท่ากับ l_{ij} ซึ่งเลือกอย่างสุ่มจากการแจกแจงเอกรูป U [1, 2] เวลาล่าช้า (Delay Time) d_{ij} สำหรับแต่ละเส้นทางถูกเลือกอย่างสุ่มจากการแจกแจงเอกรูป U [1, (1+β)*2] และเวลาเดินทางมากสุดคำนวณได้จาก $u_{ij} = l_{ij} + d_{ij}$ กำหนดจำนวนสถานการณ์ |S|= 10, 20, 40 และ 80 ในการทดลองเปรียบเทียบวิธีการจะใช้ค่าสัดส่วนสัมพัทธ์เฉลี่ย $\overline{R}$ ตัวอย่างเช่น เมื่อ $|S|$=10 ดังนั้น $\overline{R} = \frac{1}{10}\sum_{s=1}^{10}\frac{Z_r - Z_s}{Z_s}$ การทดลองต้องดำเนินการทั้งสิ้น 16 ครั้ง ตามความเป็นได้เชิงการจัดทั้งหมด (All Combinatorial) ของพารามิเตอร์ β และ $|S|$ ที่กำหนด

7. วัดสมรรถนะของขั้นตอนวิธีการ (Algorithm Performance Measure)

7.1 ตัววัดสมรรถนะและขั้นตอนวิธีการเปรียบเทียบ (Performance Measure and Benchmarking Algorithms) สำหรับการวัดสมรรถนะของขั้นตอนวิธีการแบบผสมผสานนี้ ใช้วิธีการเปรียบเทียบกับวิธี GA ซึ่งเป็นที่นิยมในการนำมาประยุกต์ใช้แก้ปัญหา VRPTW รวมทั้งในงานวิจัยที่เกี่ยวกับการหาค่าเหมาะสุดเชิงทนทานแบบไม่ต่อเนื่อง Herrmann (1999) ได้นำเสนอวิธี GA แบบ 2 พื้นที่ (Two-Space Genetic Algorithm) ในการหาผลเฉลยที่มีความทนทานต่อความไม่แน่นอน โดยพื้นที่ที่ 1 ใช้สำหรับสร้างกลุ่มประชากรของผลเฉลย และพื้นที่ที่ 2 ใช้สำหรับสร้างกลุ่มประชากรของสถานการณ์ ซึ่งเป็นขั้นตอนวิธีการที่ค่อนข้างมีประสิทธิภาพในการหาผลเฉลยเชิงสภาพทนทานสำหรับปัญหาการจัดตารางเครื่องจักรแบบขนาน และตัววัดสมรรถนะประกอบด้วย คุณภาพของผลเฉลย (Quality of Solutions) และเวลาในการประมวลผลด้วยคอมพิวเตอร์ (Computer Run Time)

7.2 วิธีการเปรียบเทียบ (Comparing Method) ในการเปรียบเทียบคุณภาพของผลเฉลย และเวลาในการประมวลผล ทำการทดลองโดยนำปัญหาเทียบเคียง (Benchmarking Problems) ของ Solomon จำนวนทั้งสิ้น 56 ปัญหา โดยแต่ละปัญหาแบ่งเป็น 25, 50 และ 100 เมือง ประมวลผลด้วยคอมพิวเตอร์จำนวน 6 เครื่อง ที่มีความสามารถแตกต่างกัน ทั้งนี้เพื่อความสะดวก รวดเร็ว ในการหาผลเฉลย ค่าเหมาะที่สุด (เวลาเดินทางต่ำสุด และจำนวนรถขนส่งต่ำสุด) และเวลาในการประมวลผล โดยมีเงื่อนไขว่า ในการประมวลผลปัญหาเดียวกัน ต้องใช้เครื่องคอมพิวเตอร์ตัวเดียวกันในการเปรียบเทียบระหว่างวิธีแบบผสมผสาน และวิธี GA

ตารางที่ 12 การกำหนดรหัสเครื่องคอมพิวเตอร์

รหัส	CPU	หน่วยความจำ	ความจุฮาร์ดดิสค์
Com 1	Core 2 Duo T7300 2.00 GHz.	2.00 Gb.	SATA 160 Gb.
Com 2	Pentium 4 3.40 GHz.	512 Mb.	SATA 80 Gb.
Com 3	Dual Core E2180 2.00 GHz.	1.00 Gb.	SATA 80 Gb.
Com 4	Dual Core D805 2.66 GHz.	2.00 Gb.	SATA 80 Gb.
Com 5	Core 2 Quad Q6600 2.40 GHz.	3.25 Gb.	SATA 250 Gb.
Com 6	Dual Core D925 3.00 GHz	1.00 Gb.	SATA 80 Gb.

ตารางที่ 13 แผนการประมวลผลปัญหาเทียบเคียงของ Solomon

ปัญหา						คอมพิวเตอร์ที่ใช้ประมวลผล
R1	C1	RC1	R2	C2	RC2	
R101.25	C101.25	RC101.25	R201.25	C201.25	RC201.25	
R101.50	C101.50	RC101.50	R201.50	C201.50	RC201.50	
R101.100	C101.100	RC101.100	R201.100	C201.100	RC201.100	
R102.25	C102.25	RC102.25	R202.25			
R102.50	C102.50	RC102.50	R202.50			Com 1
R102.100	C102.100	RC102.100	R202.100			
R103.25	C103.25		R203.25			
R103.50	C103.50		R203.50			
R103.100	C103.100		R203.100			
R104.25	C104.25	RC103.25	R204.25	C203.25	RC203.25	
R104.50	C104.50	RC103.50	R204.50	C203.50	RC203.50	
R104.100	C104.100	RC103.100	R204.100	C203.100	RC203.100	
R105.25			R205.25			Com 2
R105.50			R205.50			
R105.100			R205.100			
R106.25	C105.25	RC104.25	R206.25	C204.25	RC204.25	
R106.50	C105.50	RC104.50	R206.50	C204.50	RC204.50	
R106.100	C105.100	RC104.100	R206.100	C204.100	RC204.100	
R107.25			R207.25			Com 3
R107.50			R207.50			
R107.100			R207.100			

ตารางที่ 13 (ต่อ)

ปัญหา						คอมพิวเตอร์ที่ใช้ประมวลผล
R1	C1	RC1	R2	C2	RC2	
R108.25	C106.25	RC105.25	R208.25	C205.25	RC205.25	
R108.50	C106.50	RC105.50	R208.50	C205.50	RC205.50	
R108.100	C106.100	RC105.100	R208.100	C205.100	RC205.100	
R109.25	C107.25	RC106.25	R209.25			
R109.50	C107.50	RC106.50	R209.50			Com 4
R109.100	C107.100	RC106.100	R209.100			
R110.25						
R110.50						
R110.100						
R111.25	C108.25	RC107.25	R210.25	C202.25	RC202.25	
R111.50	C108.50	RC107.50	R210.50	C202.50	RC202.50	
R111.100	C108.100	RC107.100	R210.100	C202.100	RC202.100	
				C206.25	RC206.25	
				C206.50	RC206.50	Com 5
				C206.100	RC206.100	
				C207.25	RC207.25	
				C207.50	RC207.50	
				C207.100	RC207.100	
R112.25	C109.25	RC108.25	R211.25	C208.25	RC208.25	
R112.50	C109.50	RC108.50	R211.50	C208.50	RC208.50	Com 6
R112.100	C109.100	RC108.100	R211.100	C208.100	RC208.100	

ค่าเหมาะที่สุดที่ได้จากการประมวลผลเปรียบเทียบระหว่างขั้นตอนวิธีการแบบผสมผสานและวิธี GA โดยมีค่าเหมาะสุดยังสามารถเปรียบเทียบกับงานวิจัยอื่นๆ ที่ปรากฏชื่ออยู่ในปัญหาเทียบเคียงของ Solomon และเนื่องจากปัญหาเทียบเคียงเป็นปัญหา VRPTW (กำหนดการเชิงกำหนด) ไม่ใช่ปัญหา RVRPTW (กำหนดการเชิงทนทาน) เพื่อให้ทุกปัญหาสามารถเปรียบเทียบผลเฉลยกันได้ จึงทำให้ปัญหา RVRPTW กลายเป็นปัญหากำหนดการเชิงกำหนดทั่วไป โดยใช้การ

พิจารณาเพียงสถานการณ์เดียว คือ สถานการณ์พื้นฐาน s_b ซึ่งมีเวลาเดินทางในแต่ละเมืองเท่ากับระยะทางแบบยุคลิด $t_{ij} = \sqrt{x^2 + y^2}$ จากนั้นจึงทำการประมวลผลด้วยขั้นตอนวิธีการแบบผสมผสานเปรียบเทียบกับวิธี GA ที่สรุปขั้นตอนวิธีการ และค่าพารามิเตอร์ที่จำเป็นไว้ดังต่อไปนี้

1. ขั้นตอนวิธีการแบบผสมผสาน

ขั้นตอนที่ 1 สร้างเมทริกซ์ t_{s_b} โดยที่ $t_{ij} = \sqrt{x^2 + y^2}$ กำหนดพารามิเตอร์ค่าปรับโทษในฟังก์ชั่นวัตถุประสงค์ คือ $\alpha = 0.01$ $\gamma = 0.1$ และ $\eta = 0.05$ กำหนดพารามิเตอร์สำหรับ λ-LSD และ TS คือ จำนวนรอบการวนซ้ำ = 100 จำนวนรอบการทาบู = 5 และจำนวนรอบการยอมรับค่าฟิตเนสฟังก์ชั่นที่ไม่มีการปรับปรุง = 10

ขั้นตอนที่ 2 สร้างผลเฉลยเริ่มต้นด้วยวิธี MPFIH

ขั้นตอนที่ 3 ค้นหาผลเฉลยต่ำสุด $Z_{s_b} = \min\{F(x,t_{s_b}) : x \in X\}$ โดยใช้วิธี λ-LSD และ TS

ขั้นตอนที่ 4 สิ้นสุดขั้นตอนวิธีการ เมื่อเป็นไปตามเงื่อนไขการหยุด

2. ขั้นตอนวิธีการ GA

ขั้นตอนที่ 1 สร้างเมทริกซ์ t_{s_b} โดยที่ $t_{ij} = \sqrt{x^2 + y^2}$ กำหนดพารามิเตอร์ค่าปรับโทษในฟังก์ชั่นวัตถุประสงค์ คือ $\alpha = 0.01$ $\gamma = 0.1$ และ $\eta = 0.05$ กำหนดพารามิเตอร์สำหรับ GA คือ จำนวนรอบการวนซ้ำ = 100 จำนวนประชากรเริ่มต้น = 100 อัตราการผสมยีนส์ข้ามฟาก (Crossover Rate; P_c) = 0.7 และอัตราการผสมยีนส์ผ่าเหล่า (Mutation Rate; P_m) = 0.3

ขั้นตอนที่ 2 สร้างกลุ่มประชากรเริ่มต้น $F(x,t_{s_b}); x \in X$

ขั้นตอนที่ 3 ประเมินค่าฟังก์ชั่นวัตถุประสงค์ $F(x,t_{s_b})$ ของกลุ่มประชากรเริ่มต้น

ขั้นตอนที่ 4 เลือกโครโมโซมบรรพบุรุษ (Parent) โดยใช้วิธีเลือกวงล้อรูเลท (Roulette Wheel Selection) ที่คำนวณค่าความน่าจะเป็นในการเลือกโครโมโซมจากสัดส่วนฟังก์ชั่นวัตถุประสงค์

ขั้นตอนที่ 5 สร้างตัวเลขสุ่มเพื่อเลือกวิธีการผสมยีนส์ในแต่ละรอบการวนซ้ำ

ขั้นตอนที่ 5.1 ถ้าตัวเลขสุ่มน้อยกว่า P_c=0.7 เลือกวิธีผสมยีนส์ข้ามฟาก

ขั้นตอนที่ 5.2 ถ้าตัวเลขสุ่มน้อยกว่า P_m=0.3 เลือกวิธีผสมยีนส์ผ่าเหล่า

ขั้นตอนที่ 6 ประเมินค่าฟังก์ชั่นวัตถุประสงค์ $F(x,t_{s_b})$ ของประชากรรุ่นลูก (Offsprings)

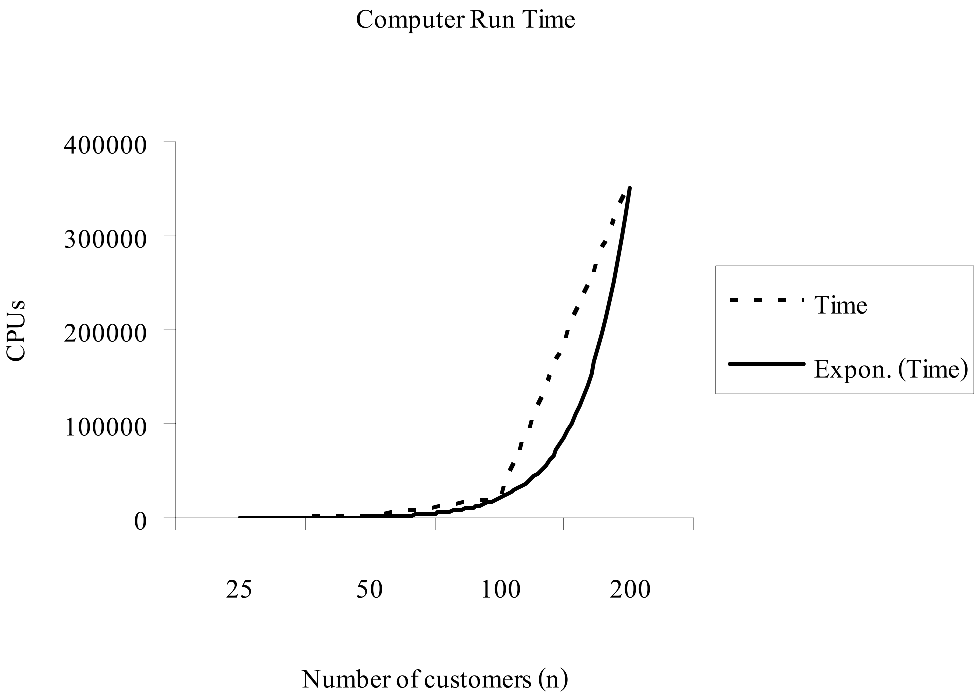
ขั้นตอนที่ 7 เพิ่มประชากรรุ่นลูกในกลุ่มประชากรเริ่มต้น และคำนวณค่าความน่าจะเป็นในการเลือกโครโมโซม

ขั้นตอนที่ 8 ทำซ้ำในขั้นตอนที่ 3-7 และสิ้นสุดขั้นตอนวิธีการเมื่อเป็นไปตามเงื่อนไขการหยุด

ผลและวิจารณ์

ผล

จากผลการพัฒนาขั้นตอนวิธีการแบบผสมผสาน สามารถลดความซับซ้อนของปัญหาจากอัตราการเพิ่มขึ้นของเวลาประมวลผลด้วยคอมพิวเตอร์ในการหาค่าเหมาะที่สุดใกล้เคียงจากการเพิ่มขึ้นแบบแฟคทรอเรียล $O(n!)$ เป็นแบบเลขชี้กำลัง $O(\exp n)$ โดยสามารถคำนวณปัญหา VRPTW จำนวน $n = 200$ ภายในระยะเวลาประมาณ 4 วัน ในขณะที่กรณีแบบแฟคทรอเรียลจะใช้เวลาเป็นนิรันดร์ (Infinity)



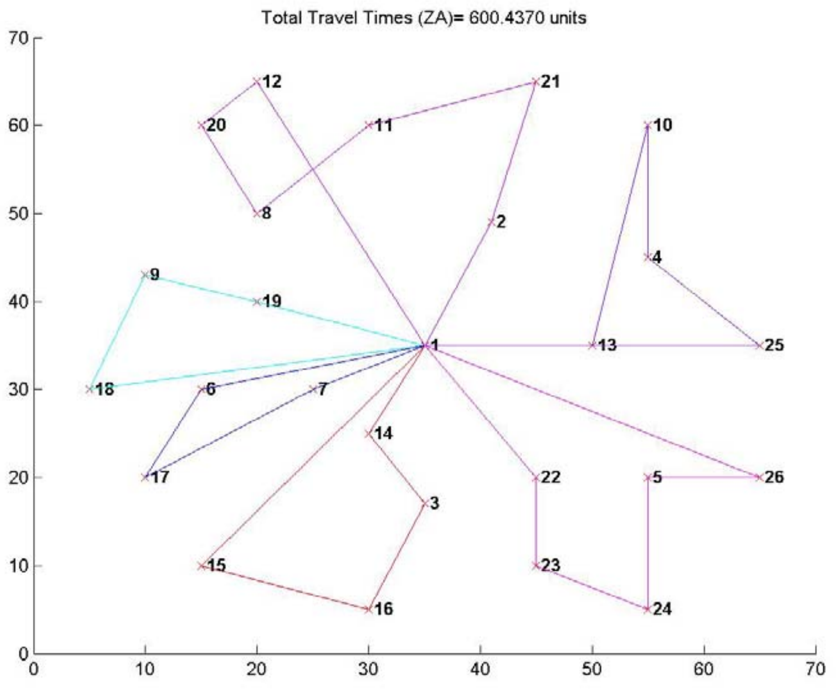
ภาพที่ 24 อัตราการเพิ่มขึ้นของเวลาประมวลผลด้วยคอมพิวเตอร์

จากภาพที่ 24 เป็นผลที่ได้จากการทดลองแสดงให้เห็นว่าขั้นตอนวิธีการแบบผสมผสานสามารถลดอัตราการเพิ่มขึ้นของเวลาประมวลผลด้วยคอมพิวเตอร์เป็นอย่างมาก โดยแนวโน้มของกราฟมีลักษณะเพิ่มขึ้นแบบเลขชี้กำลัง อย่างไรก็ตามผลที่ได้ยังคงต้องรอการพิสูจน์ทางทฤษฎีต่อไป

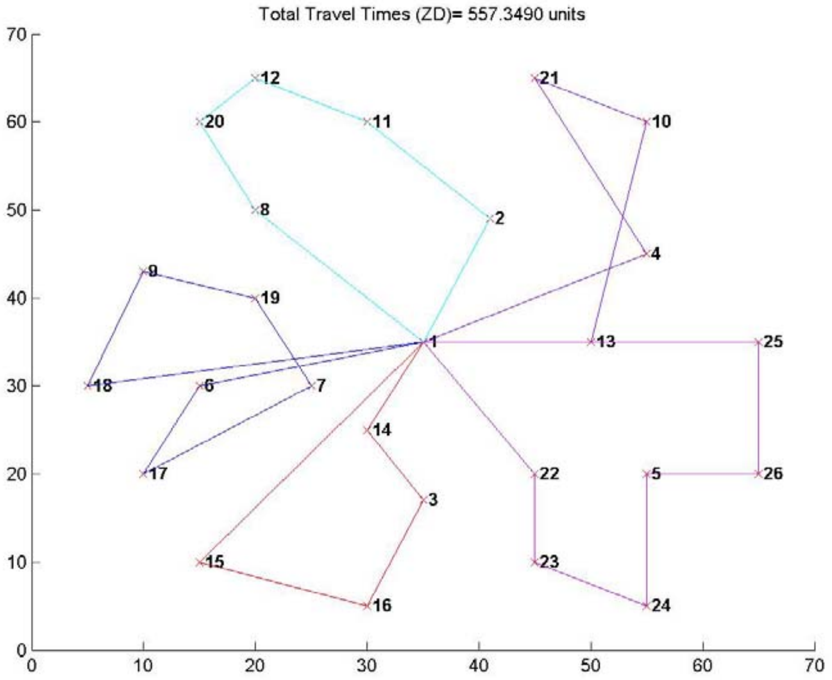
1. ผลการหาผลเฉลยเชิงทนทาน

ผลเฉลยเชิงทนทานสามารถแสดงในรูปของเส้นทางการเดินรถขนส่งที่กระจายไปตามตำแหน่งที่ตั้งของลูกค้าตามเมืองต่างๆ แสดงตัวอย่างของการหาผลเฉลยสำหรับปัญหา R101 จำนวนลูกค้า 25 เมือง $|S| = 10$ และ $\beta = 0.2$ ดังภาพที่ 21-23 และตารางที่ 15

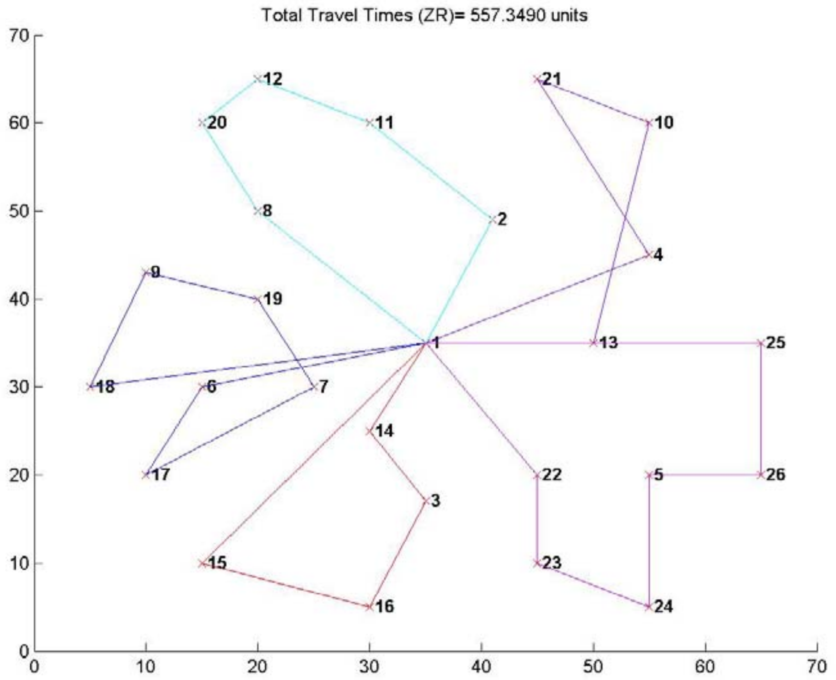
ในงานวิจัยนี้ได้แสดงการหาผลเฉลยเชิงสภาพทนทานสำหรับปัญหา R101, C101, RC101, R201, C201 และ RC201 ที่มีจำนวนลูกค้า 25 เมือง กำหนดให้ค่า $|S|$ มีขนาด 10, 20, 40 และ 80 โดยที่ β มีค่าเท่ากับ 0.2, 0.4, 0.6 และ 0.8 ผลเฉลยเชิงทนทานสำหรับปัญหา R101, C101, RC101, R201, C201 และ RC201 แสดงดังภาคผนวก ก



ภาพที่ 25 เส้นทางการเดินรถขนส่งสำหรับปัญหา R101, $|S|$=10, β=0.2 ด้วยเกณฑ์ตัดสินใจ Z_A



ภาพที่ 26 เส้นทางการเดินรถขนส่งสำหรับปัญหา R101, $|S|$=10, β=0.2 ด้วยเกณฑ์ตัดสินใจ Z_D

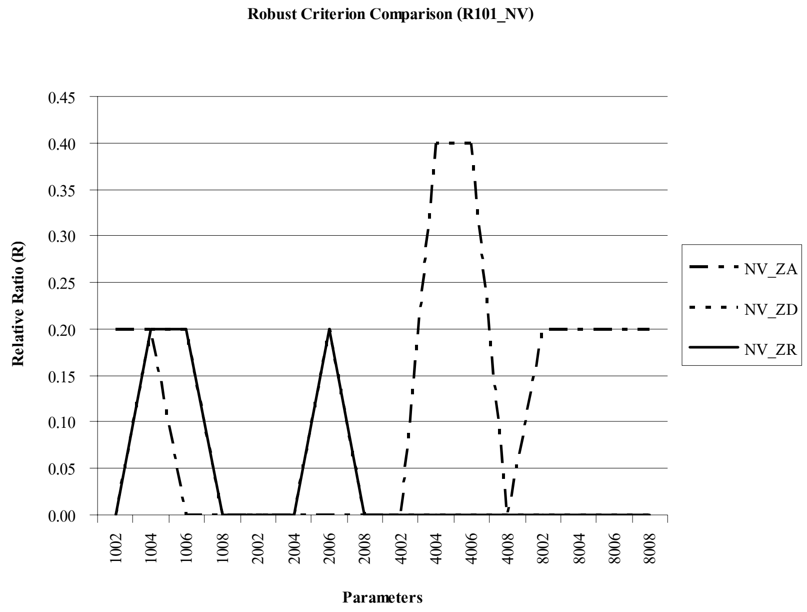


ภาพที่ 27 เส้นทางการเดินรถขนส่งสำหรับปัญหา R101, $|S|$=10, β=0.2 ด้วยเกณฑ์ตัดสินใจ Z_R

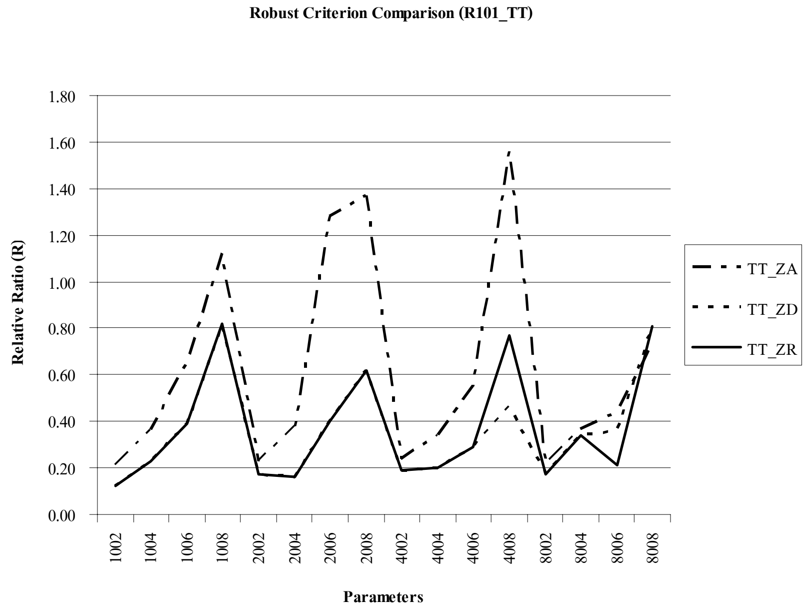
ตารางที่ 14 ตัวอย่างการแสดงผลเฉลยเชิงทนทานสำหรับปัญหา R101, $|S|$=10 และ β=0.2

เกณฑ์ตัดสินใจ	เส้นทางเดินรถขนส่ง	จำนวนรถขนส่ง (คัน)	เวลาเดินทางโดยรวม (หน่วย)
Z_A	1 --> 15 --> 16 -->3 --> 14 --> 1 1 --> 19 --> 9 --> 18 --> 1 1 --> 6 --> 17 --> 7--> 1 1 --> 13 --> 10 -->4 --> 25 --> 1 1 --> 12 --> 20 -->8 --> 11 --> 21 -->2 --> 1 1 --> 22 --> 23 -->24 --> 5 --> 26--> 1	6	600.44
Z_D	1 --> 15 --> 16 --> 3 --> 14 --> 1 1 --> 8 --> 20 --> 12 --> 11 --> 2 --> 1 1 --> 6 --> 17 -->7 --> 19 --> 9 -->18 --> 1 1 --> 13 --> 10 --> 21 --> 4 --> 1 1 --> 22 --> 23 --> 24 --> 5 --> 26 --> 25--> 1	5	557.35
Z_R	1 --> 15 --> 16 --> 3 --> 14 --> 1 1 --> 8 --> 20 --> 12 --> 11 --> 2 --> 1 1 --> 6 --> 17 -->7 --> 19 --> 9 -->18 --> 1 1 --> 13 --> 10 --> 21 --> 4 --> 1 1 --> 22 --> 23 --> 24 --> 5 --> 26 --> 25--> 1	5	557.35

ผลเฉลยเชิงทนทานที่ได้ เมื่อมีการเปลี่ยนแปลงค่าพารามิเตอร์ $|S|$ และ β ให้ผลเฉลยที่แตกต่างกัน เมื่อทดลองเปรียบเทียบกับผลเฉลยพื้นฐาน ที่มีพารามิเตอร์ $|S|$=1 และ $\beta = 0$ หรืออาจเทียบได้กับผลเฉลยเชิงกำหนด ด้วยค่าสัดส่วนสัมพัทธ์ $R = \dfrac{Z_r - Z_{s_b}}{Z_{s_b}}$ โดยที่ Z_r คือ ผลเฉลยเชิงทนทาน (Z_A, Z_D และ Z_R) และ Z_{s_b} คือ ผลเฉลยพื้นฐาน หรือผลเฉลยเชิงกำหนด การทดลองแบ่งตามประเภทของปัญหาเทียบเคียง ซึ่งแสดงผลดังภาพต่อไปนี้

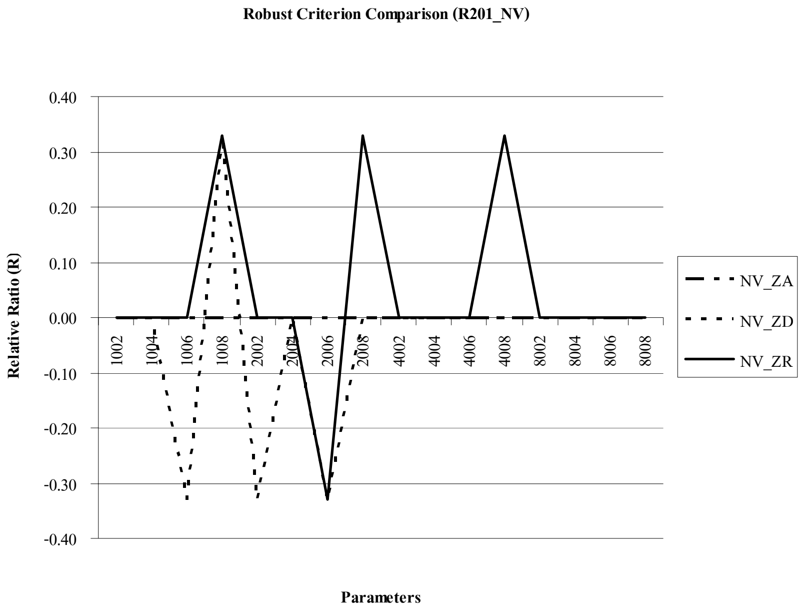


ภาพที่ 28 กราฟเปรียบเทียบเกณฑ์ตัดสินใจเชิงทนทานสำหรับจำนวนรถขนส่งของปัญหา R101

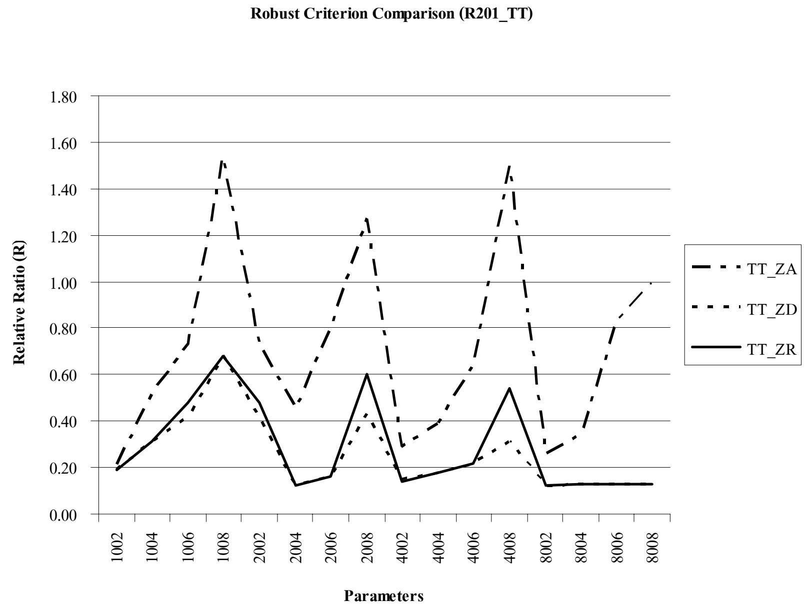


ภาพที่ 29 กราฟเปรียบเทียบเกณฑ์ตัดสินใจเชิงทนทานสำหรับเวลาเดินทางของปัญหา R101

จากกราฟแสดงสมรรถนะของเกณฑ์การตัดสินใจเชิงทนทานทั้ง 3 ประเภท เมื่อเทียบกับกรณีสถานการณ์พื้นฐาน โดยแกน y แสดงสัดส่วนสัมพัทธ์ R และแกน x แสดงการเปลี่ยนแปลงของค่าพารามิเตอร์ $|S|$ และ β ซึ่งตัวเลข 2 ตัวแรก แทนจำนวนสถานการณ์ และ 2 ตัวหลัง แทนเปอร์เซ็นต์ความไม่แน่นอน ตัวอย่างเช่น 1002 หมายถึง $|S| = 10$ และ $\beta = 0.2$

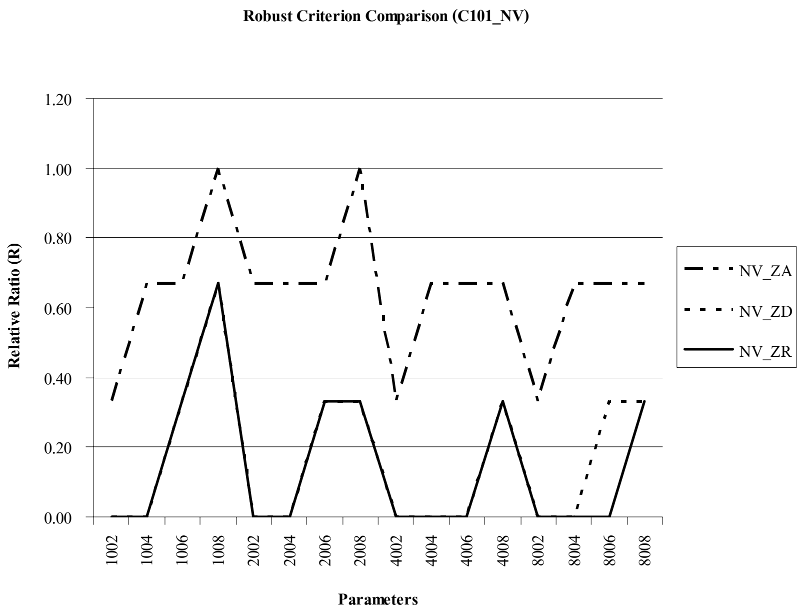


ภาพที่ 30 กราฟเปรียบเทียบเกณฑ์ตัดสินใจเชิงทนทานสำหรับจำนวนรถขนส่งของปัญหา R201

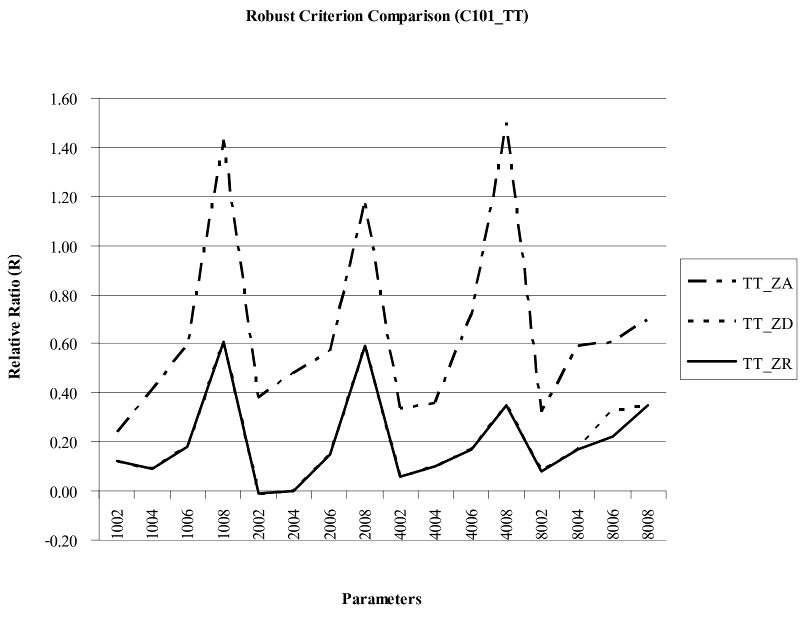


ภาพที่ 31 กราฟเปรียบเทียบเกณฑ์ตัดสินใจเชิงทนทานสำหรับเวลาเดินทางของปัญหา R201

จากกราฟสำหรับปัญหา R101 และ R201 แสดงแนวโน้มเหมือนกัน คือ เมื่อค่า β เพิ่มมากขึ้น สมรรถนะของเกณฑ์ตัดสินใจทั้ง 3 จะลดลง สังเกตได้จากค่า R ที่เพิ่มมากขึ้นตามค่า β โดยที่สมรรถนะของเกณฑ์ตัดสินใจ Z_D และ Z_R ดีกว่า Z_A ในทุกกรณี

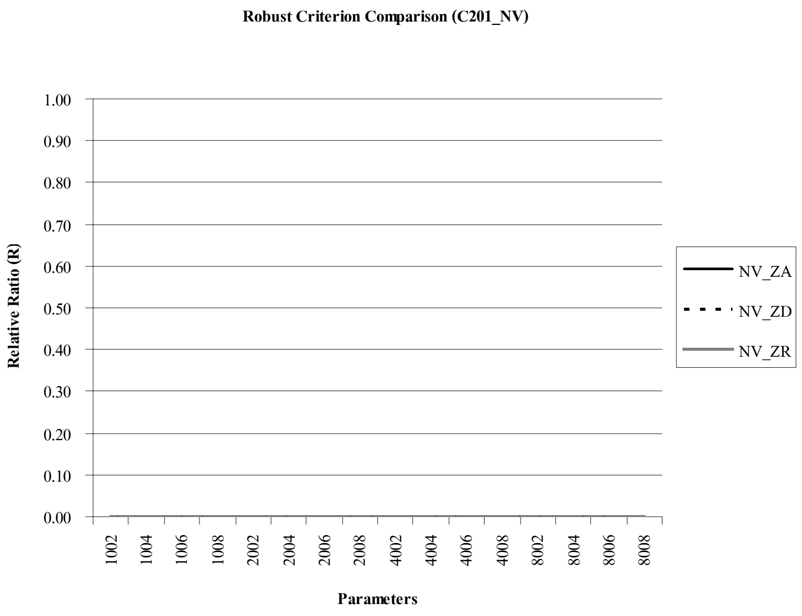


ภาพที่ 32 กราฟเปรียบเทียบเกณฑ์ตัดสินใจเชิงทนทานสำหรับจำนวนรถขนส่งของปัญหา C101

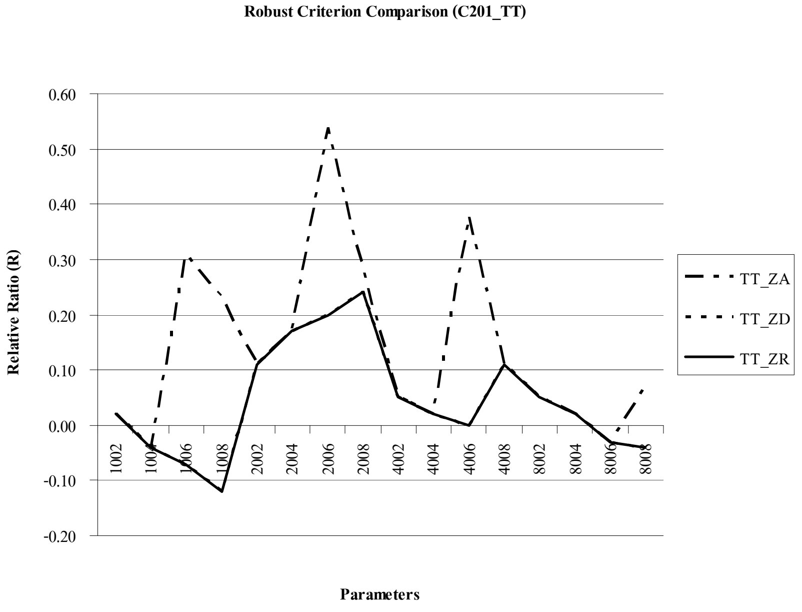


ภาพที่ 33 กราฟเปรียบเทียบเกณฑ์ตัดสินใจเชิงทนทานสำหรับเวลาเดินทางของปัญหา C101

สำหรับปัญหา C101 แนวโน้มของกราฟไปในทิศทางเดียวกับปัญห R101 และ R201 ต่างกันที่ค่า R ของทั้ง 2 ภาพ มีค่ามากกว่าหรือเท่ากับศูนย์ หมายความว่า ผลเฉลยเชิงทนทานที่ได้ทั้งหมดจากเกณฑ์การตัดสินทั้ง 3 ประเภท ให้ค่ามากกว่าผลเฉลยเชิงกำหนดในทุกกรณี

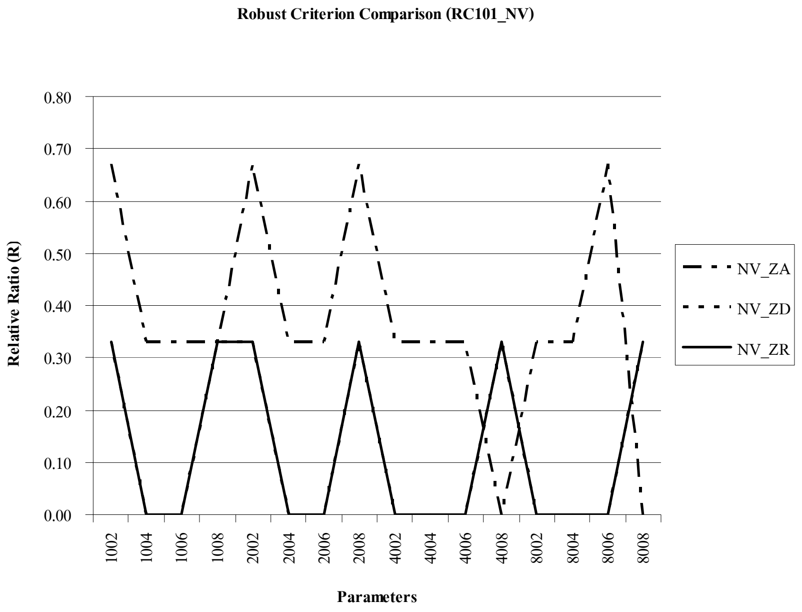


ภาพที่ 34 กราฟเปรียบเทียบเกณฑ์ตัดสินใจเชิงทนทานสำหรับจำนวนรถขนส่งของปัญหา C201

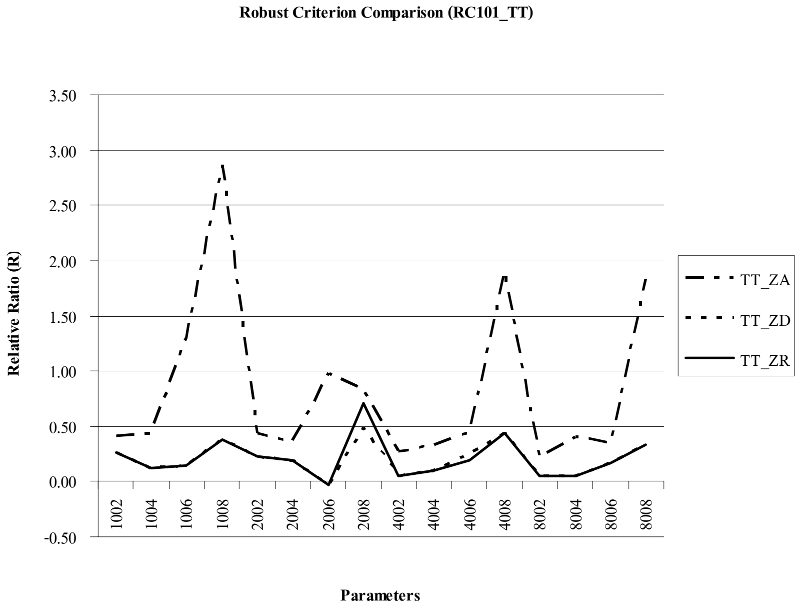


ภาพที่ 35 กราฟเปรียบเทียบเกณฑ์ตัดสินใจเชิงทนทานสำหรับเวลาเดินทางของปัญหา C201

สำหรับปัญหา C201 ภาพที่ 34 แสดงให้เห็นว่าผลเฉลยเชิงทนทานมีเค่าเท่ากับผลเฉลยเชิงกำหนดทุกประการ เป็นการพิสูจน์จากการทดลองได้ว่า สำหรับบางปัญหาการหาผลเฉลยเชิงทนทานอาจให้ค่าเท่ากับผลเฉลยเชิงกำหนด แสดงว่าการเลือกใช้เกณฑ์การตัดสินใจเชิงทนทานไม่จำเป็นต้องได้ค่ามากกว่าเสมอไป อาจให้ค่าที่น้อยกว่าหรือเท่ากับก็ได้

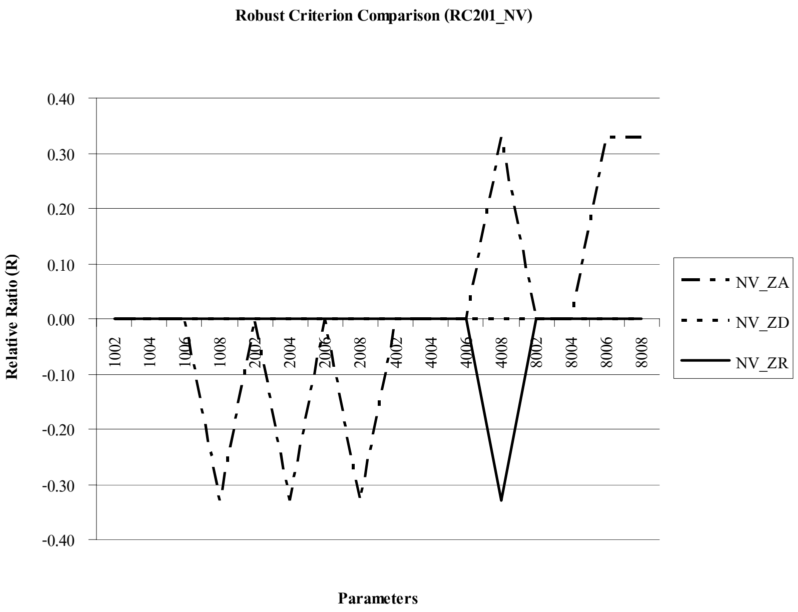


ภาพที่ 36 กราฟเปรียบเทียบเกณฑ์ตัดสินใจเชิงทนทานสำหรับจำนวนรถขนส่งของปัญหา RC101

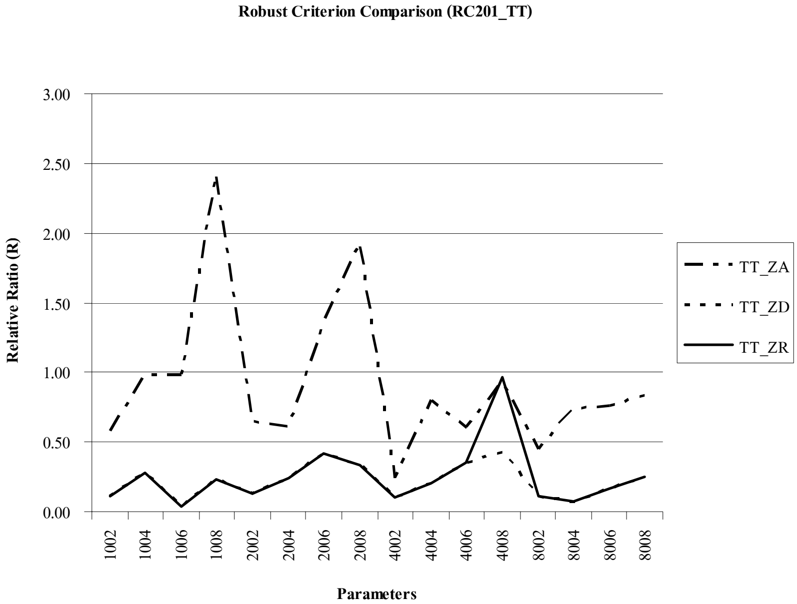


ภาพที่ 37 กราฟเปรียบเทียบเกณฑ์ตัดสินใจเชิงทนทานสำหรับเวลาเดินทางของปัญหา RC101

สำหรับปัญหา RC101 แสดงผลการเปรียบเทียบที่เหมือนกับปัญหาอื่นๆ คือ เกณฑ์ตัดสินใจแบบ Z_D และ Z_R ยังคงมีสมรรถนะที่ดีที่สุด คือ ให้ค่าผลเฉลยเชิงทนทานที่ไม่แตกต่างจากค่าผลเฉลยเชิงกำหนดมากเกินไป



ภาพที่ 38 กราฟเปรียบเทียบเกณฑ์ตัดสินใจเชิงทนทานสำหรับจำนวนรถขนส่งของปัญหา RC201



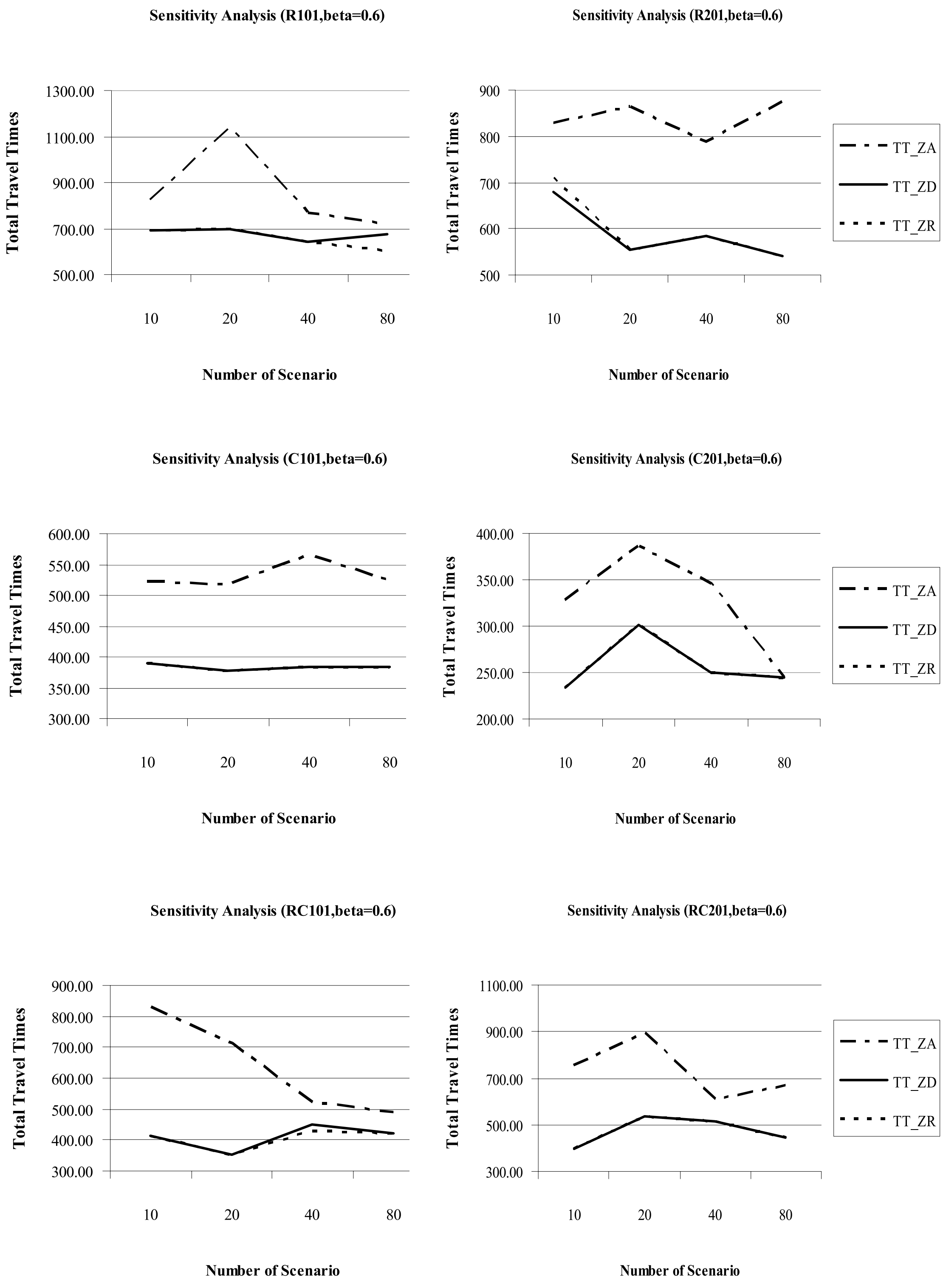
ภาพที่ 39 กราฟเปรียบเทียบเกณฑ์ตัดสินใจเชิงทนทานสำหรับเวลาเดินทางของปัญหา RC201

ปัญหา RC201 แสดงผลการเปรียบเทียบเช่นเดียวกับปัญหาอื่นๆ และมีข้อสังเกตว่าในการเลือกเกณฑ์การตัดสินใจเชิงทนทาน ควรเลือกเกณฑ์ที่ให้ค่าผลเฉลยไม่มากเกินไป เมื่อเทียบกับผลเฉลยเชิงกำหนด เนื่องจากในบางครั้งการยอมรับค่าที่เลวร้ายมากเกินไป อาจทำให้ผู้ตัดสินใจเสียโอกาสโดยใช่เหตุ

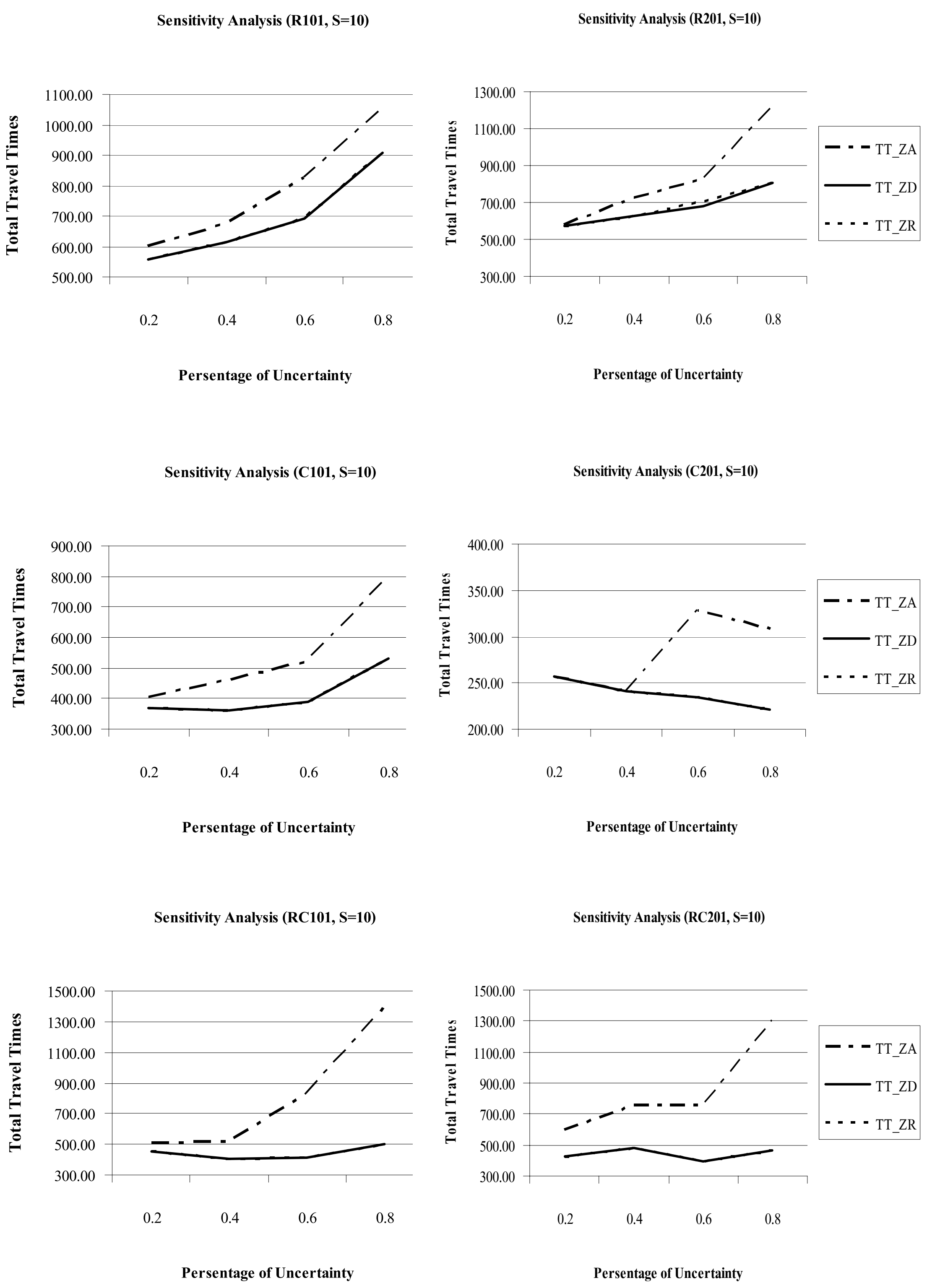
การเปรียบเทียบเกณฑ์การตัดสินใจ ทำให้สามารถวิเคราะห์สมรรถนะของผลเฉลยเชิงทนทานที่เปลี่ยนแปลงตามค่าพารามิเตอร์ $|S|$ และ β ไปพร้อมๆ กัน ทั้งนี้เพื่อเป็นการวิเคราะห์ความไว (Sensitivity Analysis) ของผลเฉลยเชิงทนทานที่ได้ กับค่าพารามิเตอร์แต่ละตัว จึงได้ทำการทดลองเปลี่ยนแปลงค่า $|S|$ = 10, 20, 40 และ 80 ในขณะที่คงที่ค่า β = 0.6 และทดลองเปลี่ยนแปลงค่า β = 0.2, 0.4, 0.6 และ 0.8 โดยคงที่ค่า $|S|$ = 10 ผลการวิเคราะห์ความไวสำหรับปัญหาทั้ง 6 ประเภท แสดงดังภาพที่ 40 และ 41

จากภาพที่ 40 แสดงให้เห็นอย่างชัดเจนว่าการเปลี่ยนแปลงค่า $|S|$ ไม่มีผลกระทบต่อค่าผลเฉลยของปัญหาทั้ง 6 ประเภท สังเกตได้จากลักษณะของกราฟที่ไม่มีแนวโน้มไปในทิศทางใดทิศทางหนึ่ง หมายความว่า จำนวนของสถานการณ์มากน้อย มิได้มีผลกระทบต่อคุณภาพผลเฉลยเชิงทนทาน เพราะฉะนั้นการกำหนดจำนวนสถานการณ์ที่เหมาะสม จึงมาจากการวิเคราะห์และความเชี่ยวชาญของผู้ตัดสินใจเอง ว่าต้องการให้ผลเฉลยเชิงทนทานครอบคลุมสถานการณ์ใดๆ บ้าง การกำหนดสถานการณ์จำนวนที่มากขึ้น เป็นการลดความเสี่ยงต่อการเกิดเหตุการณ์ที่ไม่คาดคิด ดังนั้นผู้ตัดสินใจควรมีความรอบคอบในขั้นตอนการกำหนดสถานการณ์ เพราะผลเฉลยเชิงทนทานที่ได้จากขั้นตอนวิธีการ จะเป็นผลเฉลยเชิงทนทานเหมาะที่สุดสำหรับการกำหนดสถานการณ์นั้นๆ ไม่ว่าจะเป็นจำนวน 3, 10, 50 หรือ 100 หากเป็นตัวแทนของความไม่แน่นอนที่ดีแล้ว ไม่ว่าจะมีขนาดเท่าใด ผลเฉลยที่ได้ก็สามารถรับประกันผลเฉลยเชิงทนทานเหมาะที่สุดสำหรับชุดสถานการณ์นั้นๆ

สำหรับภาพที่ 41 เป็นการวิเคราะห์ความไวเมื่อเปลี่ยนแปลงค่า β จาก 0.2 ถึง 0.8 ซึ่งแนวโน้มของกราฟในปัญหาทุกประเภท มีแนวโน้มไปในทิศทางเดียวกัน คือ เมื่อค่า β เพิ่มขึ้น ค่าของผลเฉลยมีค่าเพิ่มขึ้น ซึ่งหมายความว่า เมื่อเปอร์เซ็นต์ความไม่แน่นอนมีค่าเพิ่มมากขึ้น ส่งผลให้ค่าผลเฉลยมีค่ามากขึ้นตามไปด้วย ในแง่ของธุรกิจ เมื่อธุรกิจมีความผันผวน หรือความไม่แน่นอนมาก การวางแผนจึงจำเป็นต้องใช้ค่าผลเฉลยที่สูงขึ้น เพื่อป้องกันความเสี่ยงที่อาจเกิดขึ้นจากความไม่แน่นอน



ภาพที่ 40 การวิเคราะห์ความไวเมื่อเปลี่ยนแปลงค่า $|S|$



ภาพที่ 41 การวิเคราะห์ความไวเมื่อเปลี่ยนแปลงค่า β

2. ผลการวัดสมรรถนะของวิธีการเข้าสู่ผลเฉลย

ผลการวัดสมรรถนะของวิธีการเข้าสู่ผลเฉลย โดยเปรียบเทียบระหว่างวิธีการสร้างสถานการณ์ กับวิธีการเข้าสู่ผลเฉลยแบบกรณีแย่ที่สุด และวิธีการเข้าสู่ผลเฉลยแบบคาดหวัง โดยใช้ตัวอย่างปัญหาจากตัวอย่างปัญหา RVRPTW ในตารางที่ 10 ผลการเปรียบเทียบแสดงดังตารางที่ 15

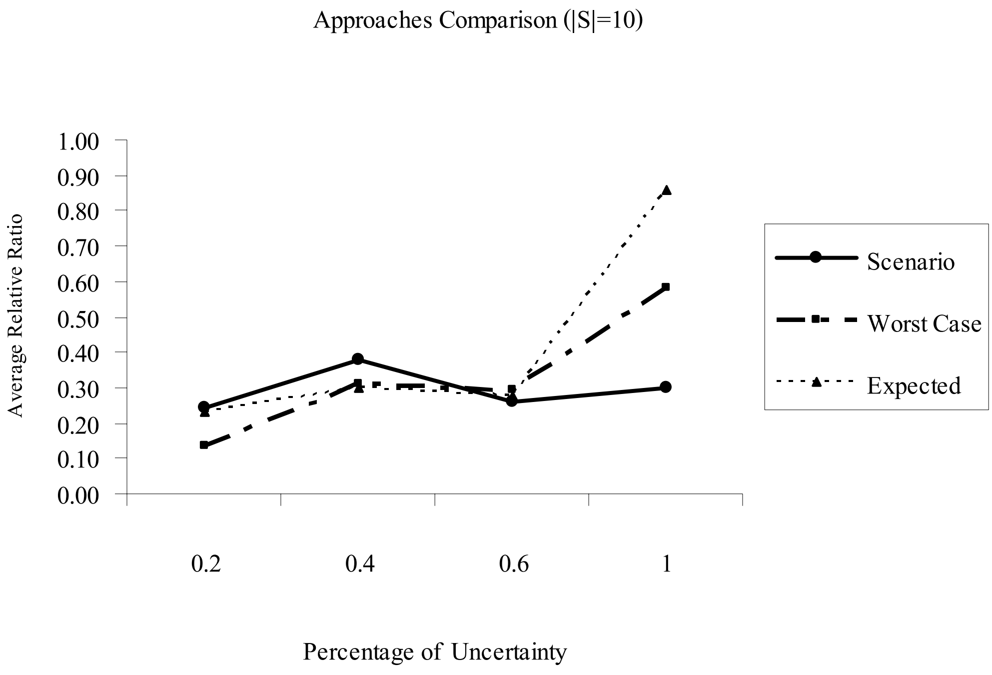
ตารางที่ 15 ผลการเปรียบเทียบวิธีการเข้าสู่ผลเฉลย

$\|S\|$	β	สมรรถนะของวิธีการเข้าสู่ผลเฉลย		
		$\overline{R}_S$	$\overline{R}_W$	$\overline{R}_E$
10	0.2	0.24	0.13	0.23
10	0.4	0.38	0.31	0.30
10	0.6	0.26	0.29	0.28
10	1.0	0.30	0.58	0.86
20	0.2	0.24	0.15	0.21
20	0.4	0.36	0.29	0.28
20	0.6	0.30	0.30	0.32
20	1.0	0.34	0.57	0.77
40	0.2	0.28	0.16	0.26
40	0.4	0.37	0.29	0.26
40	0.6	0.39	0.33	0.32
40	1.0	0.39	0.59	0.66
80	0.2	0.27	0.15	0.27
80	0.4	0.23	0.29	0.28
80	0.6	0.41	0.34	0.32
80	1.0	0.37	0.60	0.65

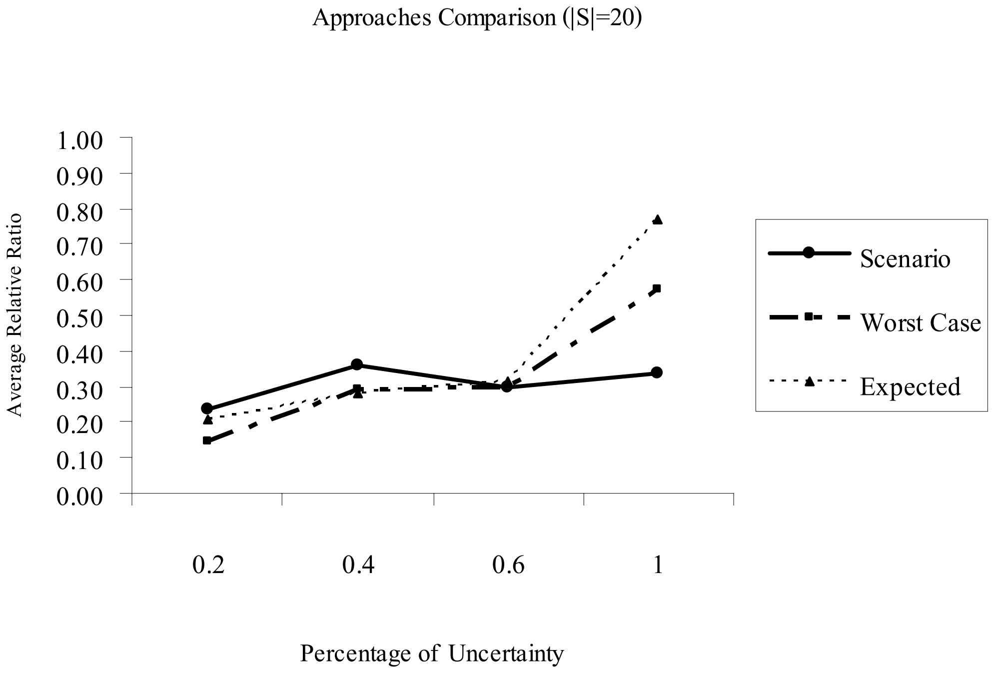
หมายเหตุ : $\overline{R}_S$ = สัดส่วนสัมพัทธ์เฉลี่ยสำหรับวิธีการสร้างสถานการณ์

$\overline{R}_W$ = สัดส่วนสัมพัทธ์เฉลี่ยสำหรับวิธีการแบบกรณีแย่ที่สุด

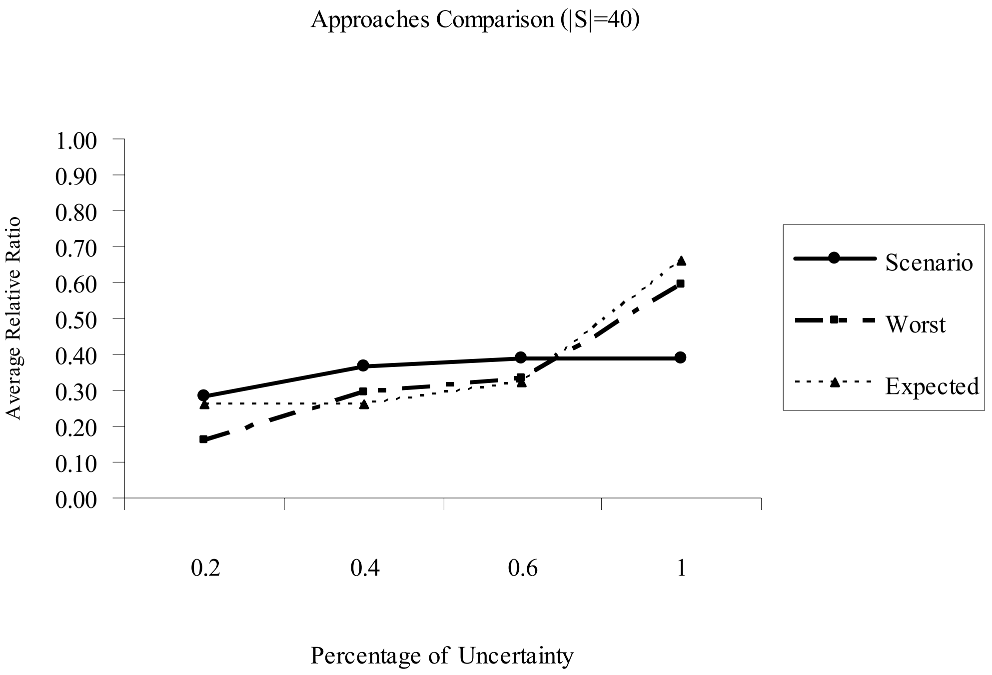
$\overline{R}_E$ = สัดส่วนสัมพัทธ์เฉลี่ยสำหรับวิธีการแบบค่าคาดหวัง



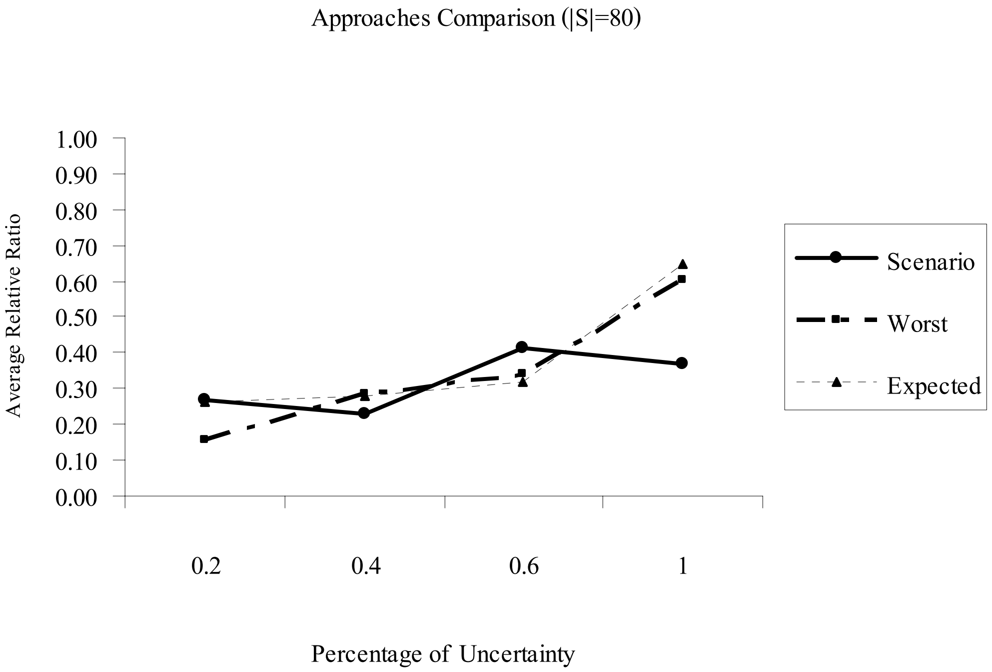
ภาพที่ 42 กราฟเปรียบเทียบวิธีการเข้าสู่ผลเฉลยเมื่อ $|S|=10$



ภาพที่ 43 กราฟเปรียบเทียบวิธีการเข้าสู่ผลเฉลยเมื่อ $|S|=20$



ภาพที่ 44 กราฟเปรียบเทียบวิธีการเข้าสู่ผลเฉลยเมื่อ $|S|$=40



ภาพที่ 45 กราฟเปรียบเทียบวิธีการเข้าสู่ผลเฉลยเมื่อ $|S|$=80

จากภาพที่ 42-45 แสดงการเปรียบเทียบวิธีการเข้าสู่ผลเฉลย โดยที่แกน x คือ เปอร์เซ็นต์ความไม่แน่นอน β และแกน y คือ ค่าเฉลี่ยสัดส่วนสัมพัทธ์ R ที่ได้จากการเปรียบเทียบผลเฉลยเชิงทนทานกับค่าเหมาะที่สุดที่เกิดขึ้นในแต่ละสถานการณ์ วิธีการเข้าสู่ผลเฉลยที่ดี คือ ต้องให้ผลเฉลยที่มีความทนทานและไม่แตกต่างจากค่าเหมาะที่สุดในแต่ละสถานการณ์มากเกินไป จากภาพแสดงให้เห็นว่าผลที่ได้จากกราฟทั้ง 4 มีแนวโน้มไปในทิศทางเดียวกัน คือ วิธีการเข้าสู่ผลเฉลยแบบการสร้างสถานการณ์ มีสมรรถนะคงที่ไม่ว่าปัญหานั้นๆ จะมีค่า β ที่เพิ่มมากขึ้น ในขณะที่อีกทั้งสองวิธีมีลักษณะกราฟที่เพิ่มขึ้นอย่างรวดเร็ว เมื่อค่า β มากกว่า 0.6 หรือ 60 % ซึ่งแสดงถึงสมรรถนะที่ด้อยลง

3. ผลการวัดสมรรถนะของขั้นตอนวิธีการ

3.1 คุณภาพของผลเฉลย ผลการเปรียบเทียบขั้นตอนวิธีการแบบผสมผสานหรือ HY กับขั้นตอนวิธีการ GA โดยใช้ปัญหาเทียบเคียงของ Solomon จำนวน 56 ตัวอย่าง โดยแต่ละตัวอย่างแบ่งเป็นจำนวนลูกค้า 25, 50 และ 100 เมือง และวัดคุณภาพของผลเฉลยด้วยค่าสัดส่วนสัมพัทธ์ $R_{HY} = \frac{Z_{HY} - Z_{OP}}{Z_{OP}}$ โดยที่ Z_{HY} คือ ผลเฉลยของวิธี HY และ Z_{OP} คือ ผลเฉลยเหมาะที่สุดจากปัญหาเทียบเคียง เช่นเดียวกันในกรณีของวิธี GA สามารถหาสัดส่วนสัมพัทธ์ได้จาก $R_{GA} = \frac{Z_{GA} - Z_{OP}}{Z_{OP}}$ โดยที่ Z_{GA} คือ ผลเฉลยของวิธี GA แสดงผลการเปรียบเทียบขั้นตอนวิธีการในภาคผนวก ข และสามารถสรุปผลเป็นกราฟแสดงความสัมพันธ์ระหว่างปัญหา VRPTW (Problems) และสัดส่วนสัมพัทธ์ของวัตถุประสงค์ (Min Number of Vehicles Ratio and Min Travel Times Ratio) ดังต่อไปนี้

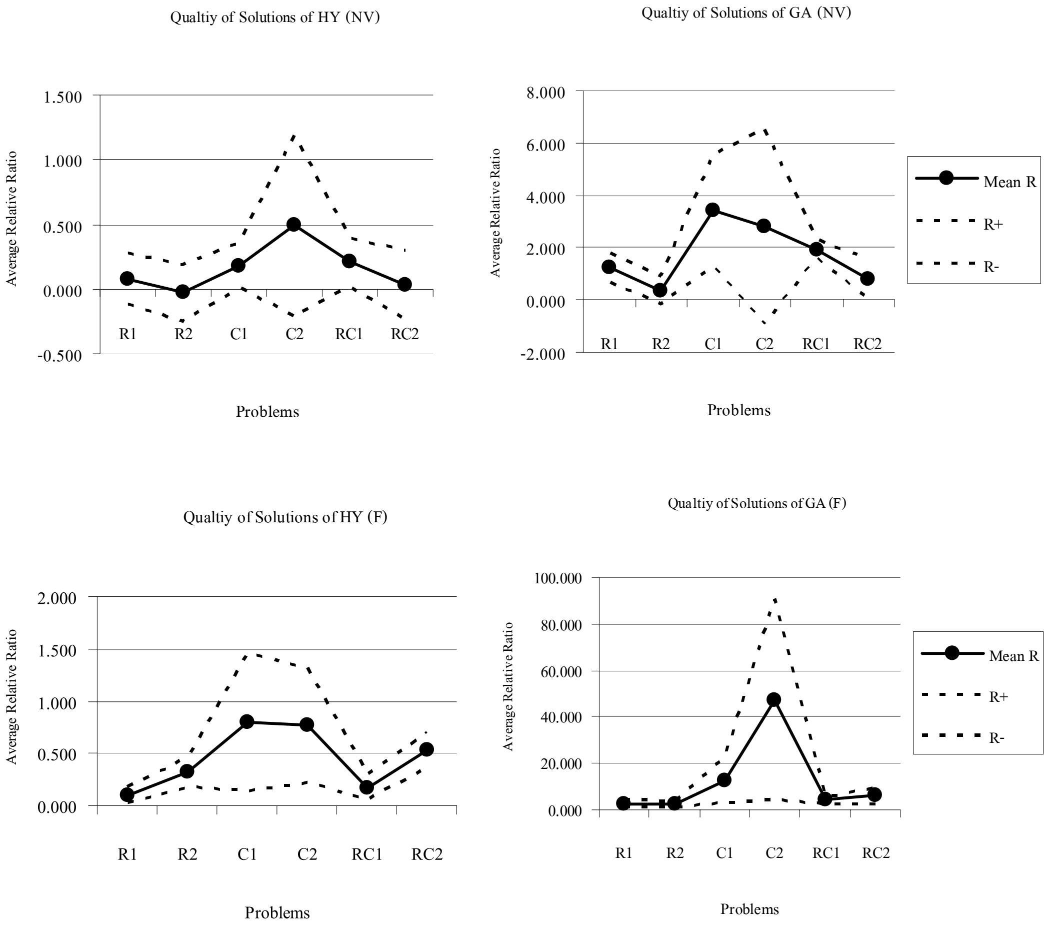
ตารางที่ 16 ผลการเปรียบเทียบคุณภาพของผลเฉลย (จำนวนรถขนส่งต่ำสุด)

ปัญหา	HY		GA	
	$\overline{R}_{HY}$	σ_{HY}	$\overline{R}_{GA}$	σ_{GA}
R1	0.080	0.198	1.240	0.575
R2	-0.027	0.221	0.373	0.508
C1	0.181	0.177	3.436	2.122
C2	0.493	0.695	2.797	3.763
RC1	0.206	0.192	1.922	0.357
RC2	0.032	0.273	0.788	0.759

ตารางที่ 17 ผลการเปรียบเทียบคุณภาพของผลเฉลย (เวลาเดินทางโดยรวมต่ำสุด)

ปัญหา	HY		GA	
	$\overline{R}_{HY}$	σ_{HY}	$\overline{R}_{GA}$	$\overline{R}_{HY}$
R1	0.104	0.075	2.679	1.452
R2	0.325	0.143	2.637	1.230
C1	0.801	0.659	12.834	9.965
C2	0.764	0.545	47.289	43.080
RC1	0.172	0.116	4.326	1.499
RC2	0.532	0.172	6.078	3.643

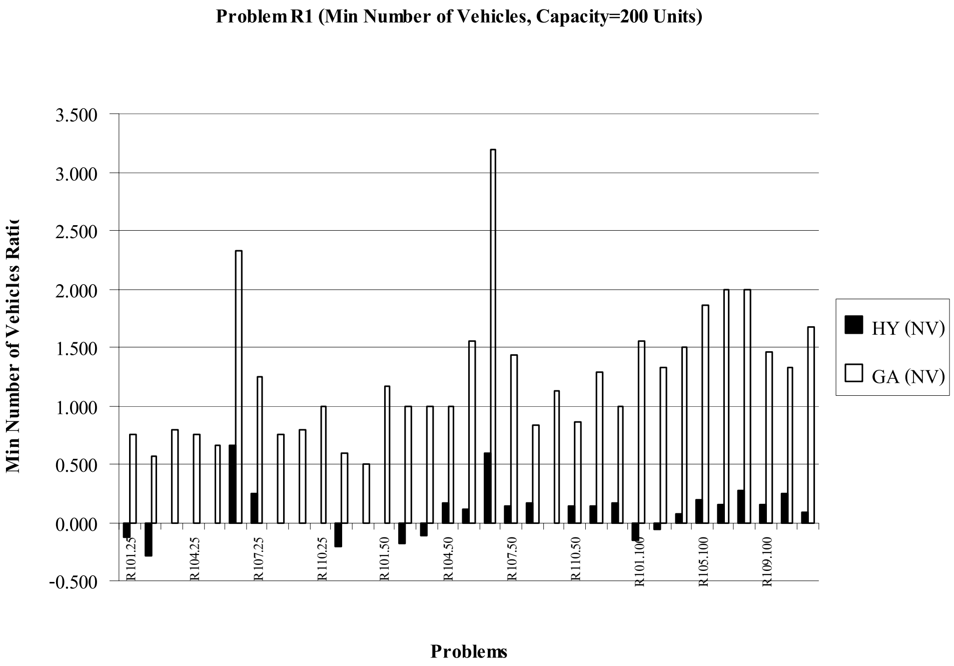
การเปรียบเทียบคุณภาพของผลเฉลยแสดงดังภาพที่ 46 โดยที่แกน x คือ ปัญหาเทียบเคียงแต่ละประเภท และแกน y คือ สัดส่วนสัมพัทธ์เฉลี่ยระหว่างค่าที่ได้จากขั้นตอนวิธีการกับค่าที่ดีที่สุดจากปัญหาเทียบเคียงของ Solomon โดยที่ HY (NV) หมายถึง การหาจำนวนรถขนส่งต่ำสุด ด้วยวิธีแบบผสมผสาน HY (F) หมายถึง การหาเวลาเดินทางต่ำสุด ด้วยวิธีแบบผสมผสาน GA (NV) หมายถึง การหาจำนวนรถขนส่งต่ำสุด ด้วยวิธีเชิงพันธุกรรม และ GA (F) หมายถึง การหาเวลาเดินทางต่ำสุด ด้วยวิธีเชิงพันธุกรรม โดยผลการเปรียบเทียบคุณภาพของผลเฉลยระหว่างขั้นตอนวิธี HY และ GA จากตารางที่ 16 และ 17 แสดงดังภาพต่อไปนี้



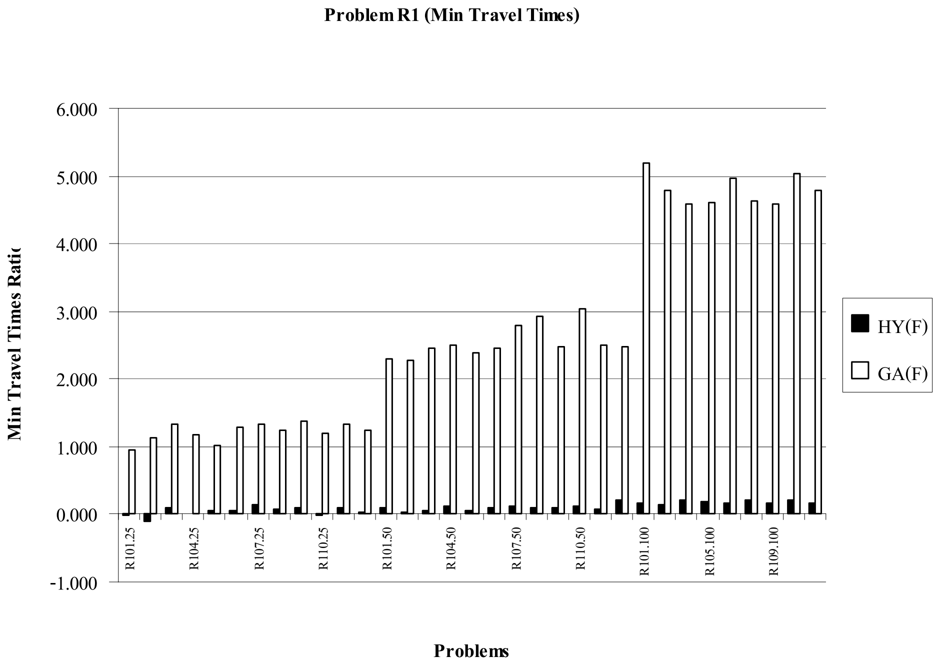
ภาพที่ 46 กราฟเปรียบเทียบคุณภาพของผลเฉลยโดยเฉลี่ย

จากภาพแสดงให้เห็นว่าสมรรถนะโดยเฉลี่ยของวิธี HY ดีกว่า GA สำหรับปัญหาเทียบเคียงทุกประเภท ซึ่งสังเกตจากค่าสัดส่วนสัมพัทธ์เฉลี่ยมีค่าไม่เกิน 1 ในขณะที่วิธี GA มีค่ามากกว่า 1 เกือบทุกกรณีของปัญหาเทียบเคียง โดยเฉพาะการหาเวลาเดินทางรวมต่ำสุดที่วิธี GA มีสมรรถนะที่ด้อยกว่าวิธี HY อย่างมาก แสดงจากค่าสัดส่วนสัมพัทธ์เฉลี่ยสูงสุดถึง 47

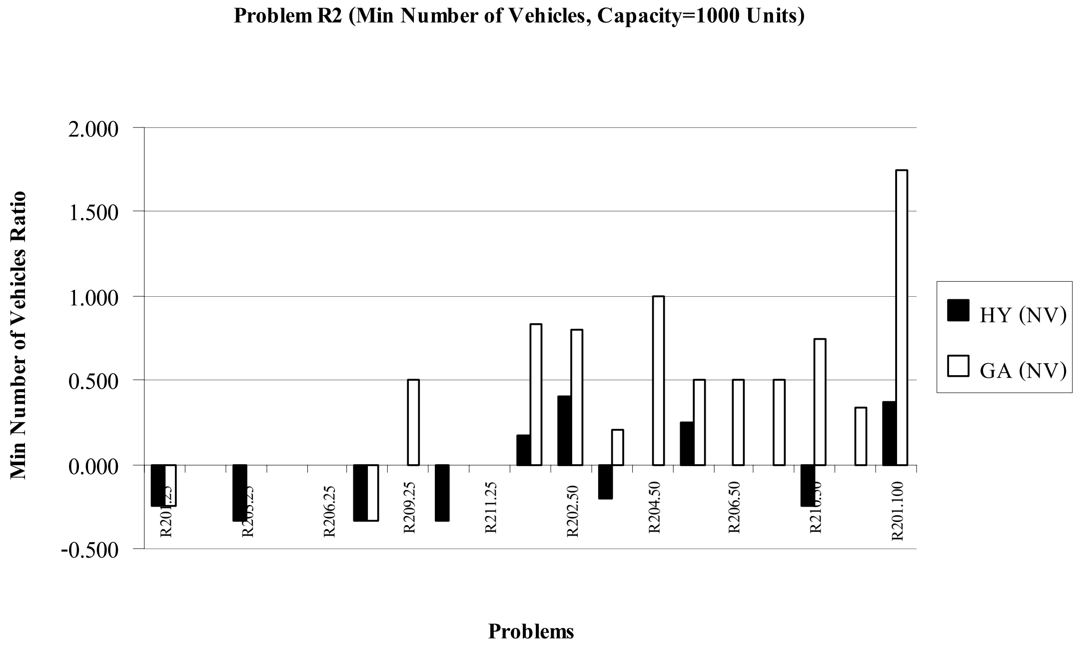
นอกจากนี้ กราฟยังแสดงให้เห็นว่าสมรรถนะของวิธี HY เหมาะสมสำหรับปัญหาเทียบเคียงทุกประเภท ยกเว้นปัญหาประเภท C1 และ C2 ที่มีค่าสัดส่วนสัมพัทธ์เฉลี่ยมากกว่าปัญหาประเภทอื่นๆ ซึ่งปัญหาประเภทนี้เป็นปัญหาที่มีการจัดแบ่งกลุ่มของลูกค้าไว้แล้ว อย่างไรก็ตาม ถึงแม้ว่าปัญหาประเภท C จะมีค่าสัดส่วนสัมพัทธ์เฉลี่ยมากกว่าปัญหาประเภทอื่นๆ แต่ค่ายังคงอยู่ระหว่าง 0-1 ซึ่งในทางปฏิบัติจัดว่าเป็นค่าที่ยอมรับได้ว่ามีสมรรถนะที่ดี



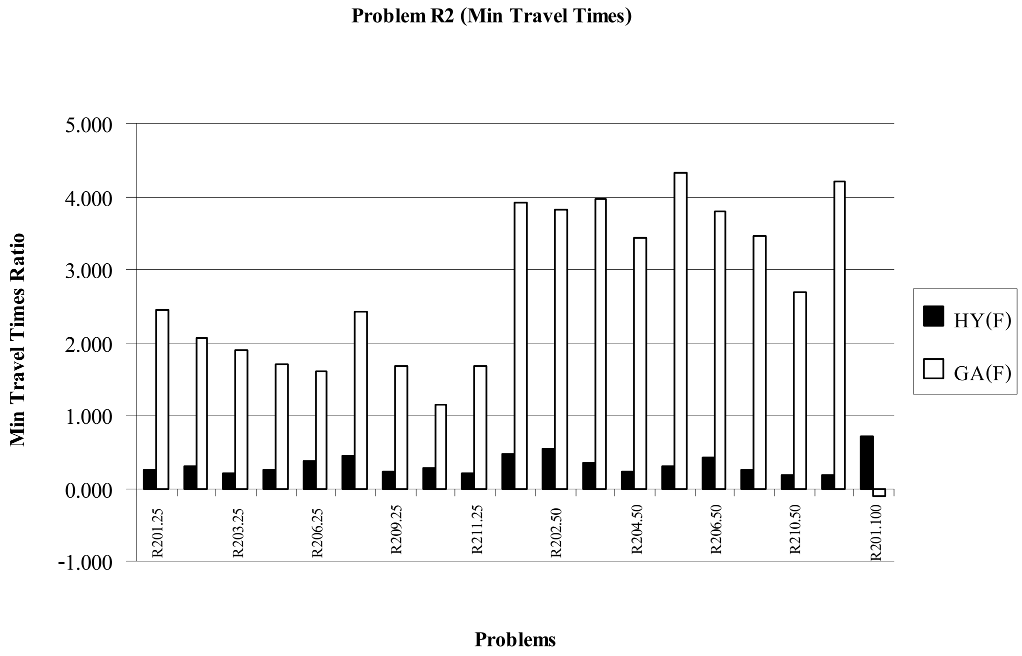
ภาพที่ 47 กราฟเปรียบเทียบสมรรถนะระหว่าง HY และ GA สำหรับปัญหา R1 (จำนวนรถขนส่งต่ำสุด, ความสามารถในการบรรทุกสินค้า = 200 หน่วย)



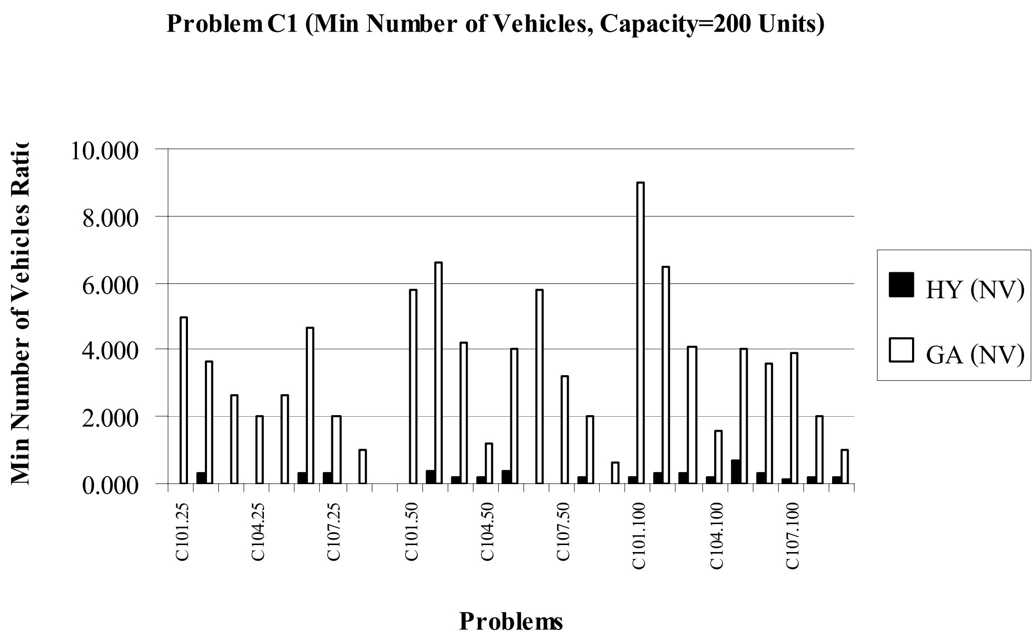
ภาพที่ 48 กราฟเปรียบเทียบสมรรถนะระหว่าง HY และ GA สำหรับปัญหา R1 (เวลาเดินทางต่ำสุด)



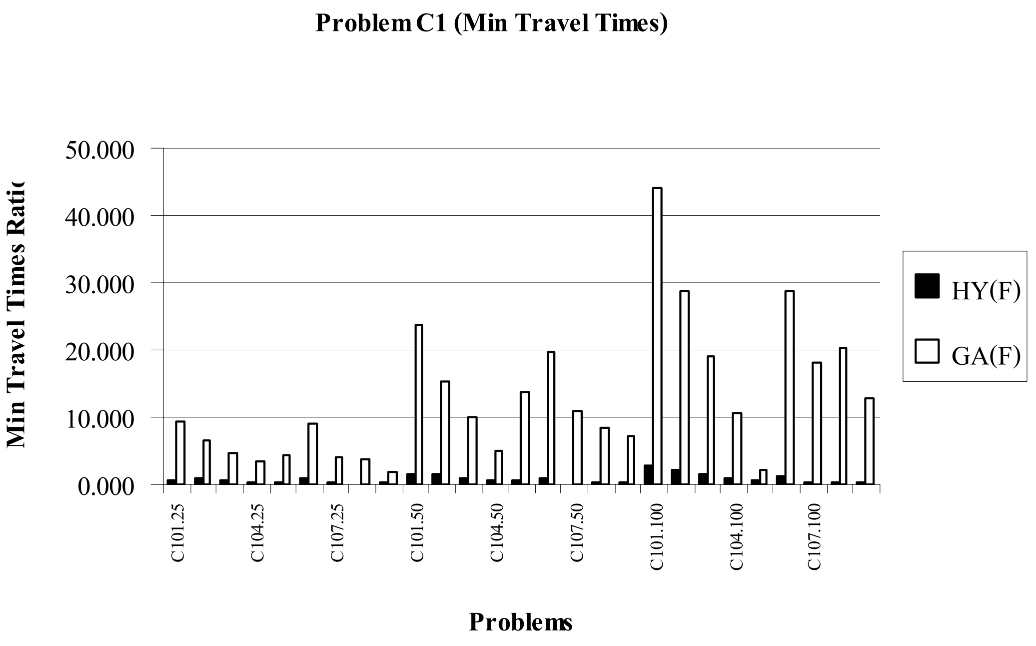
ภาพที่ 49 กราฟเปรียบเทียบสมรรถนะระหว่าง HY และ GA สำหรับปัญหา R2 (จำนวนรถขนส่งต่ำสุด, ความสามารถในการบรรทุกสินค้า = 1000 หน่วย)



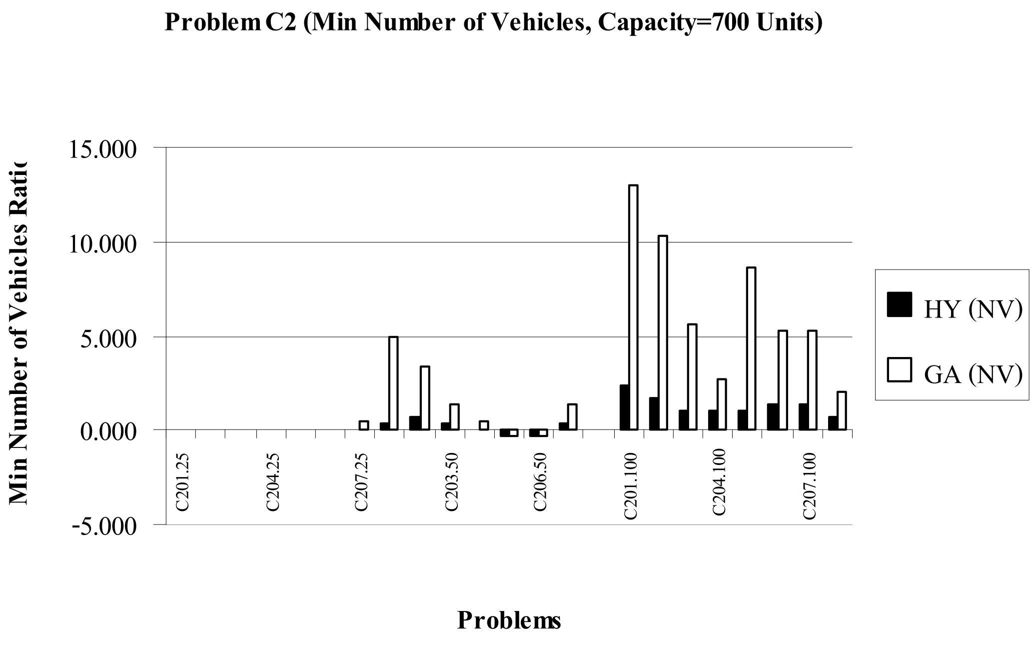
ภาพที่ 50 กราฟเปรียบเทียบสมรรถนะระหว่าง HY และ GA สำหรับปัญหา R2 (เวลาเดินทางต่ำสุด)



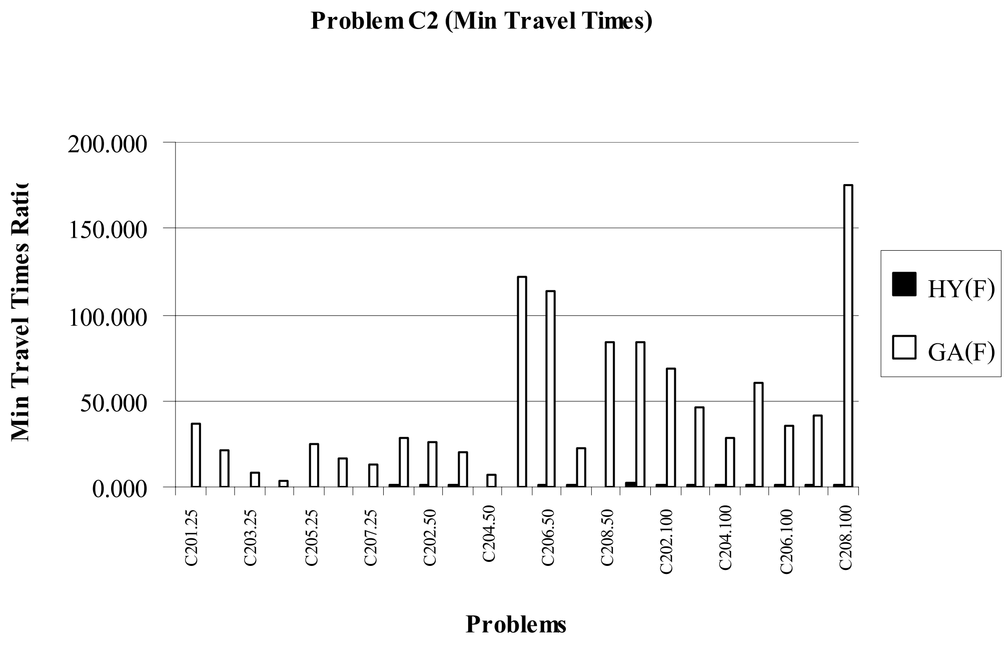
ภาพที่ 51 กราฟเปรียบเทียบสมรรถนะระหว่าง HY และ GA สำหรับปัญหา C1 (จำนวนรถขนส่งต่ำสุด, ความสามารถในการบรรทุกสินค้า = 200 หน่วย)



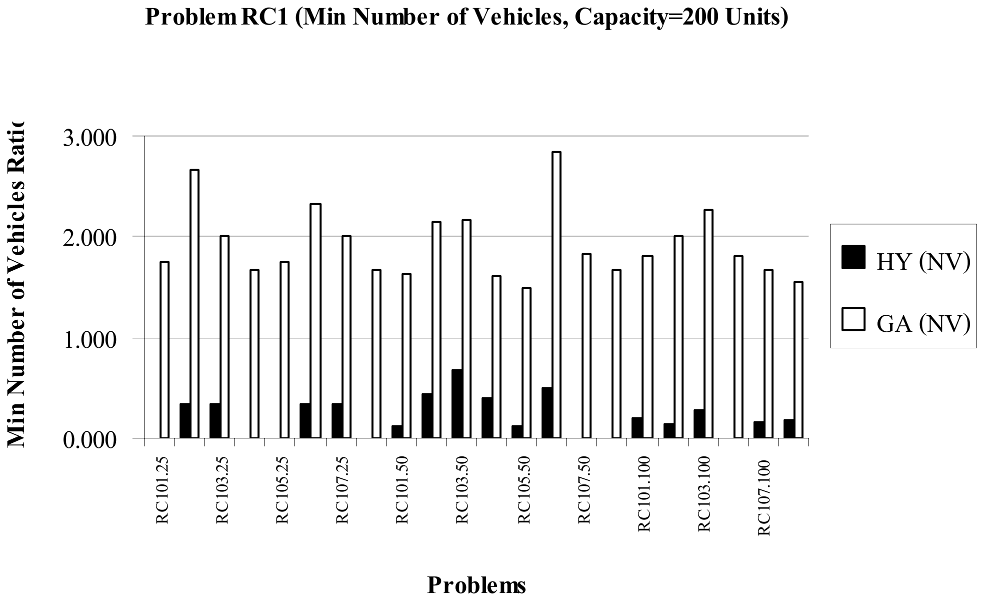
ภาพที่ 52 กราฟเปรียบเทียบสมรรถนะระหว่าง HY และ GA สำหรับปัญหา C1 (เวลาเดินทางต่ำสุด)



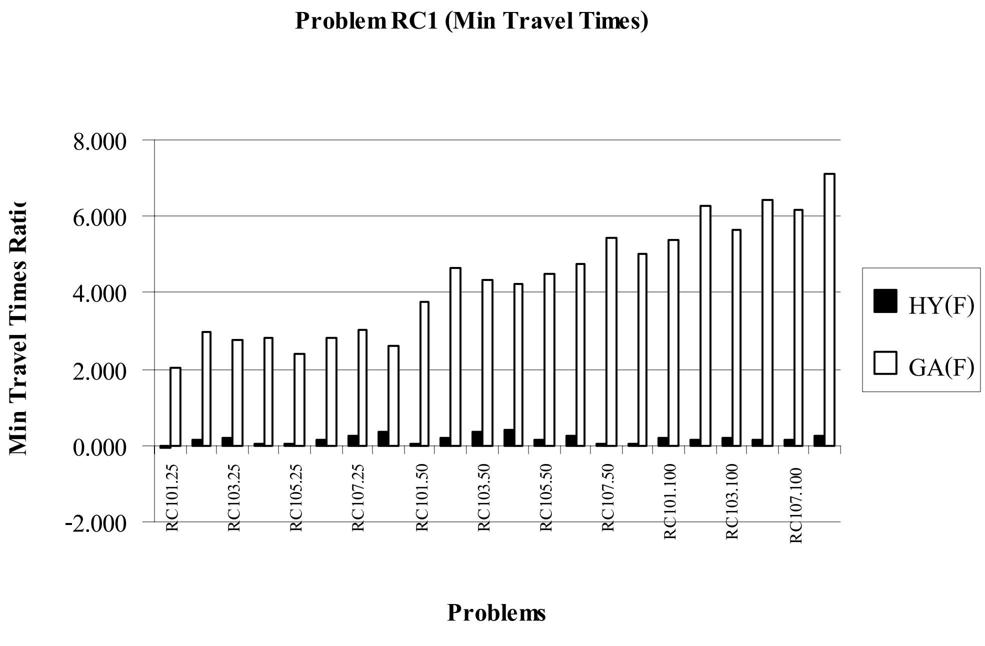
ภาพที่ 53 กราฟเปรียบเทียบสมรรถนะระหว่าง HY และ GA สำหรับปัญหา C2 (จำนวนรถขนส่งต่ำสุด, ความสามารถในการบรรทุกสินค้า = 700 หน่วย)



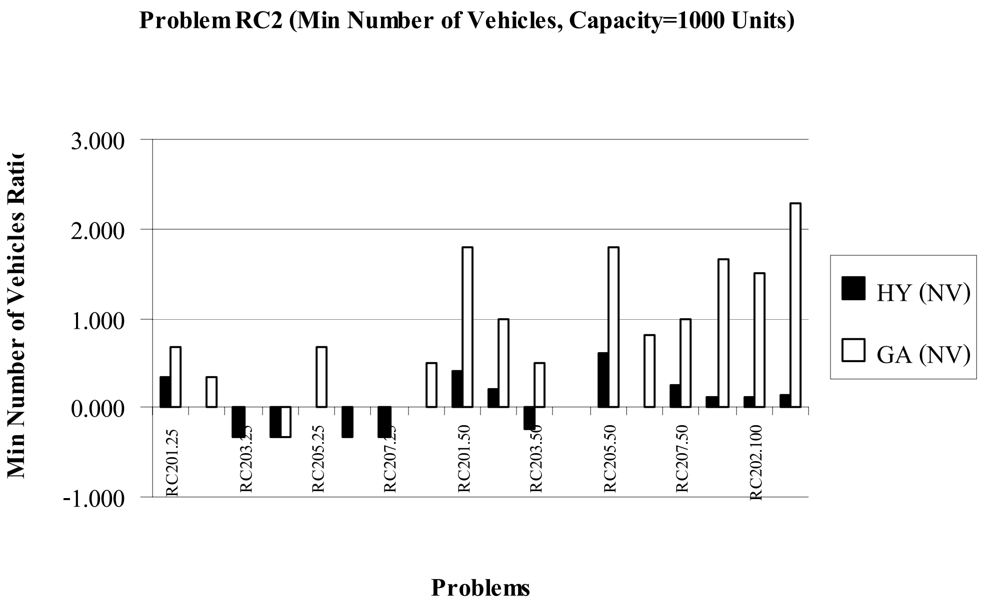
ภาพที่ 54 กราฟเปรียบเทียบสมรรถนะระหว่าง HY และ GA สำหรับปัญหา C2 (เวลาเดินทางต่ำสุด)



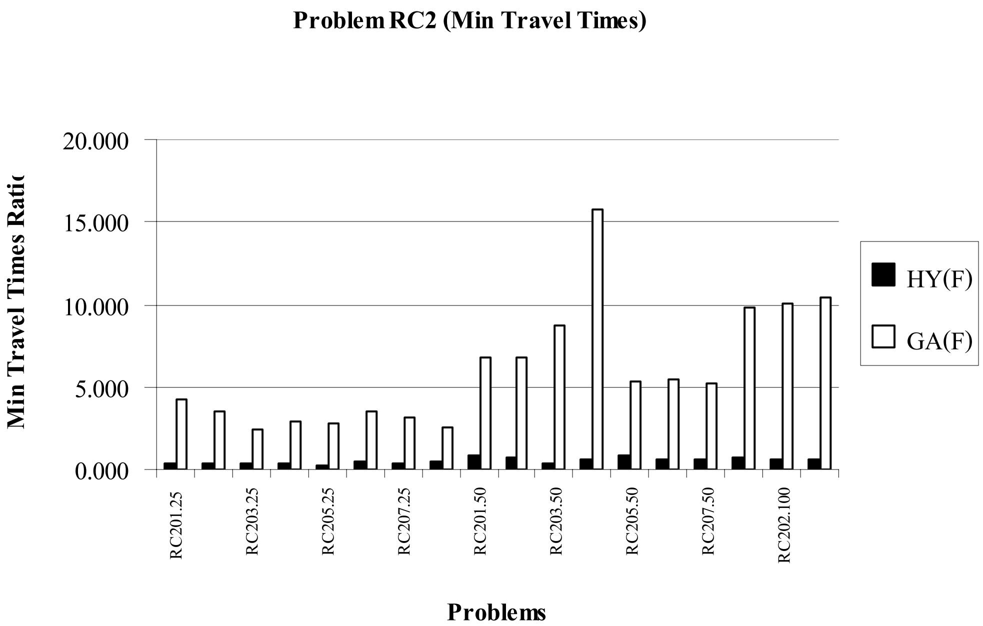
ภาพที่ 55 กราฟเปรียบเทียบสมรรถนะระหว่าง HY และ GA สำหรับปัญหา RC1 (จำนวนรถขนส่งต่ำสุด, ความสามารถในการบรรทุกสินค้า = 200 หน่วย)



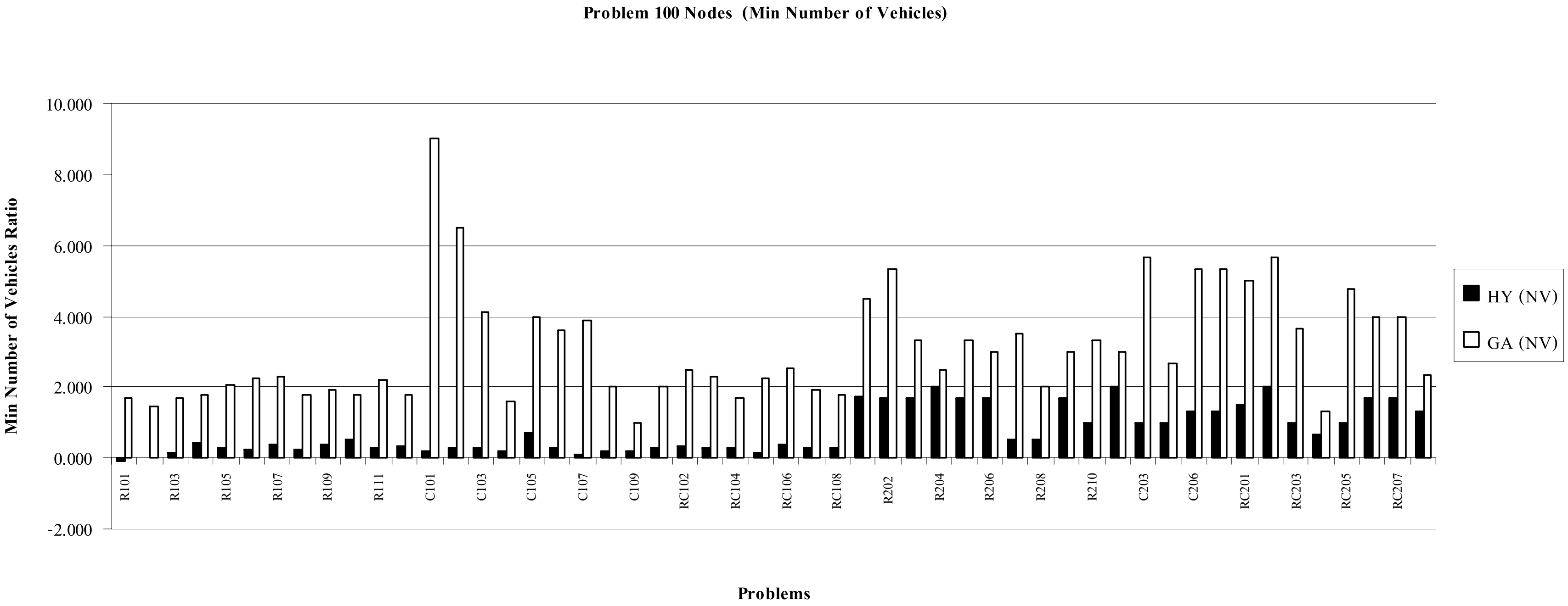
ภาพที่ 56 กราฟเปรียบเทียบสมรรถนะระหว่าง HY และ GA สำหรับปัญหา RC1 (เวลาเดินทางต่ำสุด)



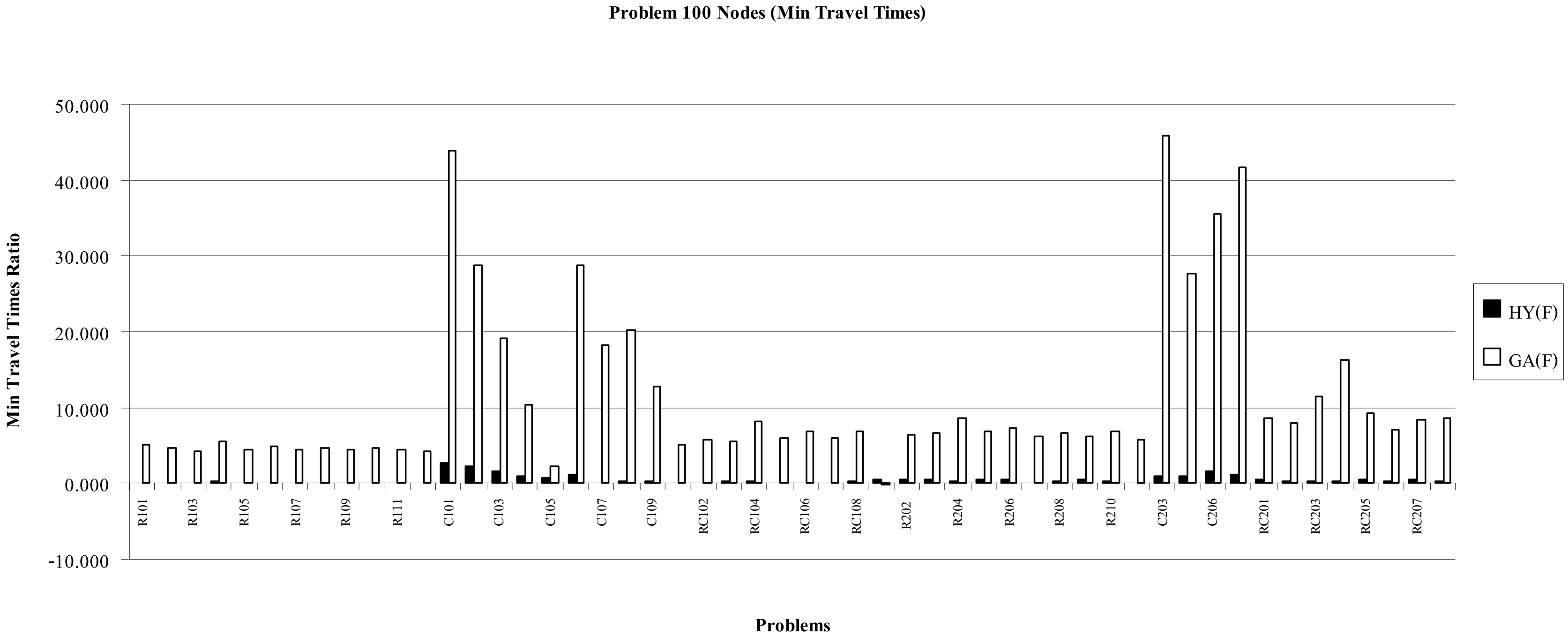
ภาพที่ 57 กราฟเปรียบเทียบสมรรถนะระหว่าง HY และ GA สำหรับปัญหา RC2 (จำนวนรถขนส่งต่ำสุด, ความสามารถในการบรรทุกสินค้า = 1000 หน่วย)



ภาพที่ 58 กราฟเปรียบเทียบสมรรถนะระหว่าง HY และ GA สำหรับปัญหา RC2 (เวลาเดินทางต่ำสุด)



ภาพที่ 59 กราฟเปรียบเทียบสมรรถนะระหว่าง HY และ GA สำหรับปัญหาขนาด 100 เมือง (จำนวนรถขนส่งต่ำสุด)



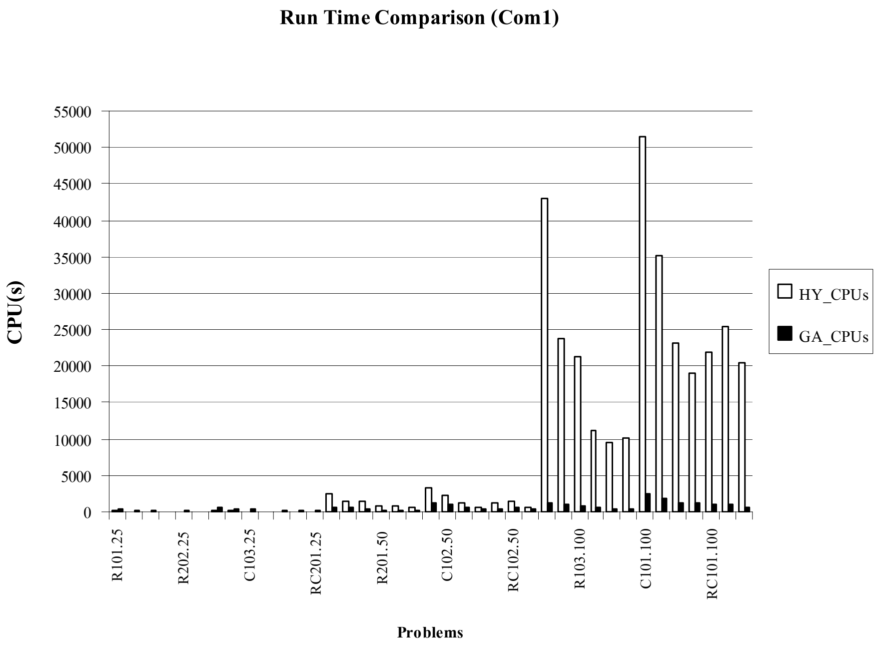
ภาพที่ 60 กราฟเปรียบเทียบสมรรถนะระหว่าง HY และ GA สำหรับปัญหาขนาด 100 เมือง (เวลาเดินทางต่ำสุด)

การเปรียบเทียบคุณภาพของผลเฉลยแสดงดังภาพที่ 47-60 โดยที่แกน x คือ ปัญหาเทียบเคียง เช่น R101.25 หมายถึง ปัญหาประเภท R1 ตัวอย่างที่ 01 มีจำนวนลูกค้า 25 เมือง ส่วนแกน y คือ สัดส่วนสัมพัทธ์ R ของผลเฉลย ระหว่างค่าที่ได้จากขั้นตอนวิธีการกับค่าที่ดีที่สุดจากปัญหาเทียบเคียงของ Solomon โดยที่ HY (NV) หมายถึง การหาจำนวนรถขนส่งที่น้อยที่สุดด้วยวิธีแบบผสมผสาน GA (F) หมายถึง การหาเวลาเดินทางรวมน้อยที่สุดด้วยวิธี GA

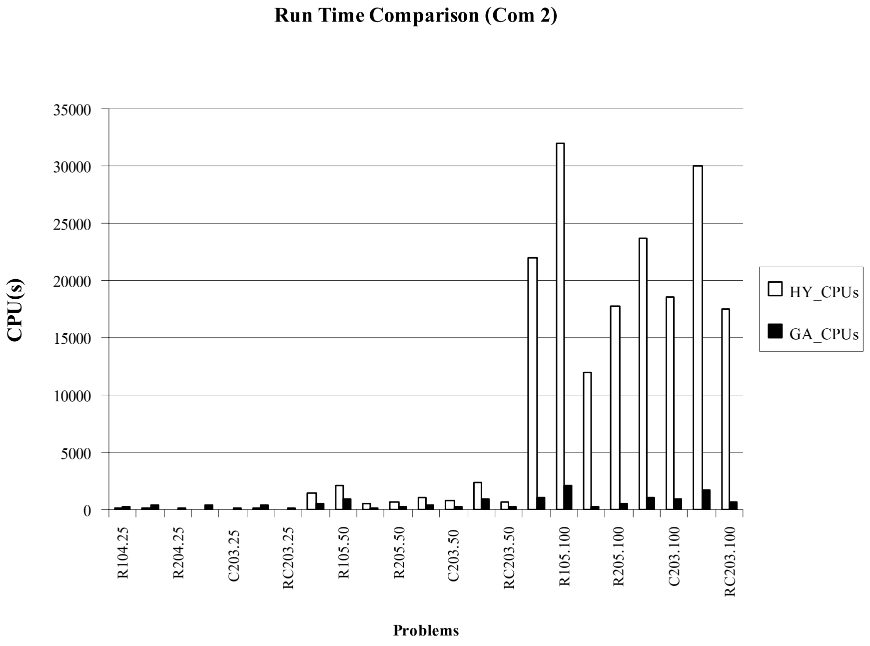
ภาพที่ 47-58 สำหรับปัญหาเทียบเคียงทุกประเภท ผลการเปรียบเทียบสมรรถนะแสดงให้เห็นอย่างชัดเจนว่าวิธี HY ให้คุณภาพของผลเฉลยที่ดีกว่า GA และในบางตัวอย่างให้ค่าผลเฉลยที่ดีกว่าปัญหาเทียบเคียง สังเกตได้จากค่าสัดส่วนสัมพัทธ์ที่ต่ำกว่าศูนย์ (ค่าติดลบ) หมายความว่า ค่าที่ได้จากขั้นตอนวิธีการมีค่าน้อยกว่าค่าที่ดีที่สุดของปัญหาเทียบเคียง จากกราฟยังแสดงให้เห็นว่าเมื่อจำนวนลูกค้าที่เพิ่มมากขึ้น ไม่มีผลกระทบต่อคุณภาพผลเฉลยที่ได้จากวิธี HY ในขณะที่คุณภาพของผลเฉลยที่ได้จากวิธี GA มีลักษณะไม่สม่ำเสมอ บางตัวอย่างแสดงค่าสัดส่วนสัมพัทธ์ที่ดี (ค่า R เข้าใกล้ค่าศูนย์) บางตัวอย่างแสดงค่าที่แย่ (ค่า R มีค่าสูงมาก)

ภาพที่ 59-60 แสดงการเปรียบเทียบสมรรถนะของวิธี HY และ GA ด้วยปัญหาเทียบเคียงทุกประเภท ที่มีจำนวนลูกค้า 100 เมือง ซึ่งเป็นการยืนยันด้วยผลการทดลองอย่างชัดเจนว่าวิธี HY ให้ผลเฉลยที่ดีกว่าวิธี GA โดยเฉพาะการหาเวลาเดินทางต่ำสุดสำหรับปัญหา VRPTW

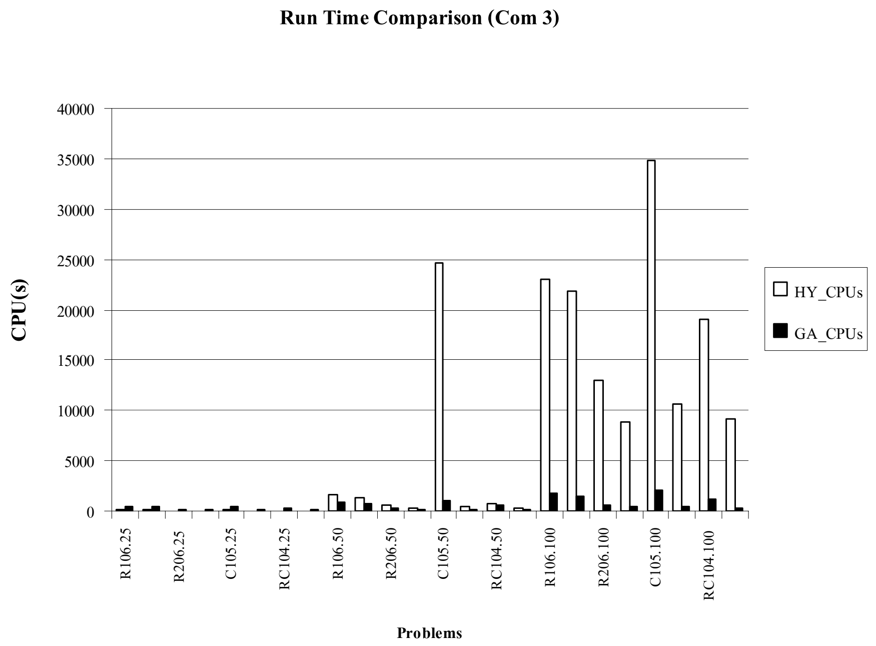
3.2 เวลาในการประมวลผลด้วยคอมพิวเตอร์ ผลการเปรียบเทียบเวลาในการประมวลผลระหว่างขั้นตอนวิธีการ HY และ GA โดยทดลองประมวลผลปัญหาเทียบเคียงของ Solomon จำนวน 56 ตัวอย่าง โดยแต่ละตัวอย่างแบ่งเป็นจำนวนลูกค้า 25, 50 และ 100 เมือง ใช้คอมพิวเตอร์จำนวน 6 เครื่องในการประมวลผล โดยกำหนดให้แต่ละเครื่องต้องประมวลผลปัญหาตัวอย่างให้ครบทั้ง 6 ประเภท ได้แก่ R1, R2, C1, C2, RC1 และ RC2 เพื่อให้สามารถวิเคราะห์เปรียบเทียบเวลาในการประมวลผลสำหรับปัญหาตัวอย่างแต่ละประเภทได้ เวลาที่ใช้ในการประมวลผลเริ่มจับเวลาการทำงานของ CPU ตั้งแต่ขั้นตอนวิธีการสร้างเส้นทางเริ่มต้น จนกระทั่งได้ผลเฉลยสุดท้าย โดยไม่รวมขั้นตอนการกำหนดตัวแปร หรือการสร้างกราฟแสดงเส้นทางการเดินรถขนส่ง กำหนดให้หน่วยเป็นวินาที ผลการทดลองแสดงดังกราฟต่อไปนี้



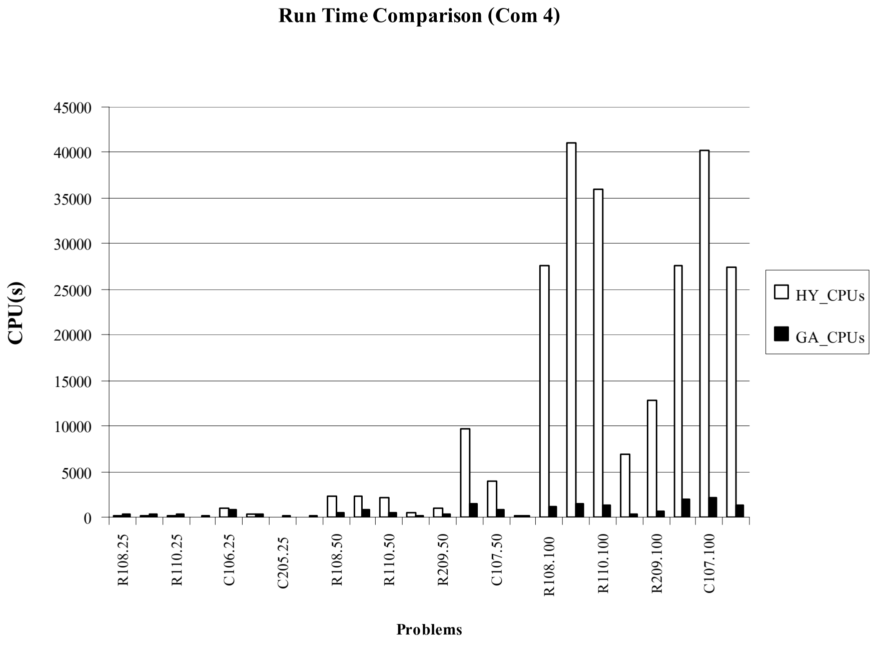
ภาพที่ 61 กราฟเปรียบเทียบเวลาในการประมวลผลระหว่างวิธี HY และ GA (Com 1)



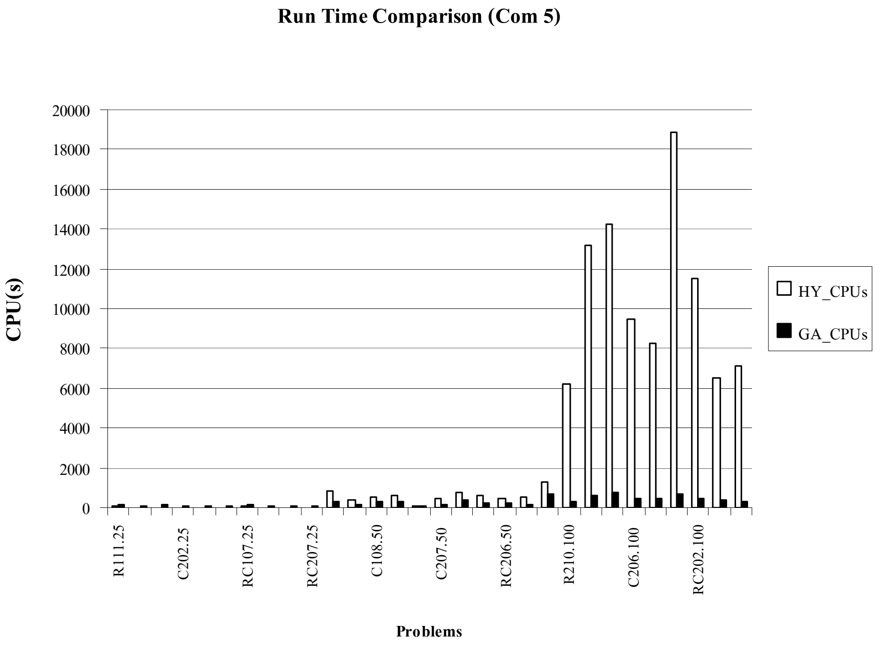
ภาพที่ 62 กราฟเปรียบเทียบเวลาในการประมวลผลระหว่างวิธี HY และ GA (Com 2)



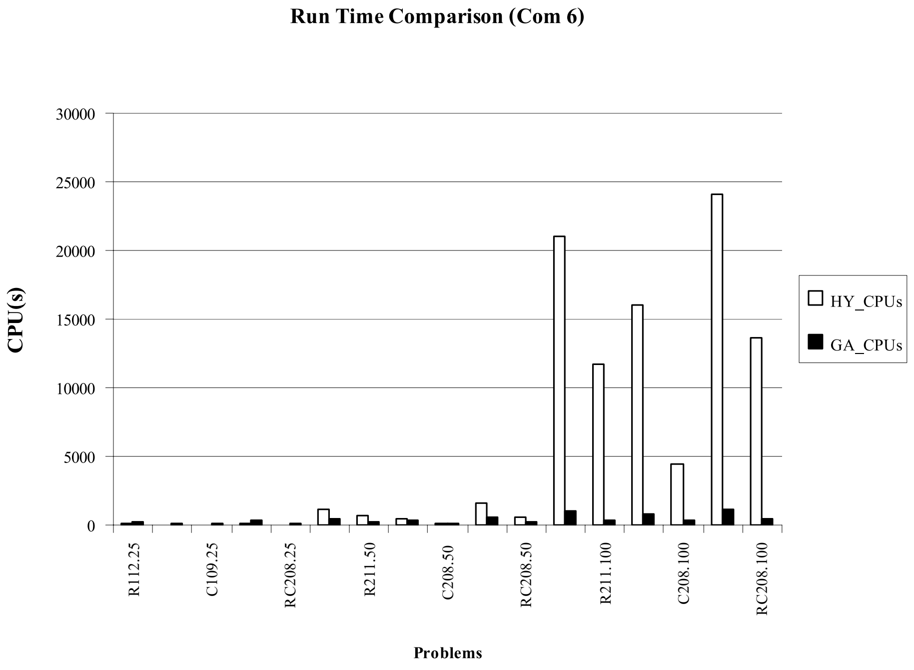
ภาพที่ 63 กราฟเปรียบเทียบเวลาในการประมวลผลระหว่างวิธี HY และ GA (Com 3)



ภาพที่ 64 กราฟเปรียบเทียบเวลาในการประมวลผลระหว่างวิธี HY และ GA (Com 4)



ภาพที่ 65 กราฟเปรียบเทียบเวลาในการประมวลผลระหว่างวิธี HY และ GA (Com 5)

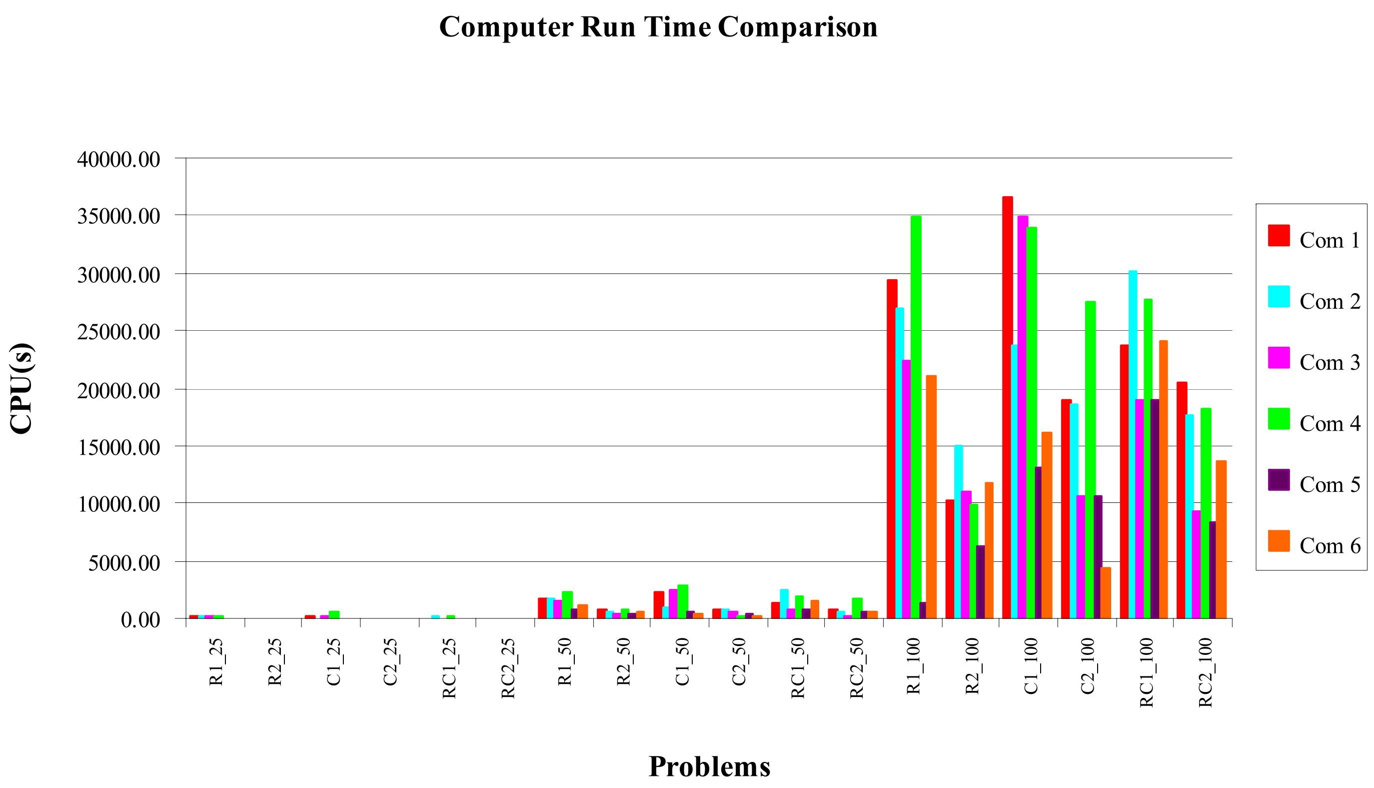


ภาพที่ 66 กราฟเปรียบเทียบเวลาในการประมวลผลระหว่างวิธี HY และ GA (Com 6)

จากภาพที่ 61-66 เป็นการเปรียบเทียบโดยใช้คอมพิวเตอร์จำนวน 6 เครื่อง ที่มีความสามารถในการประมวลผลที่แตกต่างกันไป แต่ผลการทดลองที่ได้เป็นไปในทางเดียวกัน คือ จำนวนลูกค้าน้อยกว่า 25 เมือง วิธี HY ใช้เวลาในการประมวลผลน้อยกว่าวิธี GA เมื่อจำนวนลูกค้าเพิ่มขึ้นแต่ไม่เกิน 50 เมือง วิธี HY ใช้เวลาในการประมวลผลใกล้เคียงกับวิธี GA และเมื่อลูกค้ามีจำนวนเพิ่มขึ้นมากกว่า 50 เมือง วิธี HY ใช้เวลาในการประมวลผลยาวนานกว่าวิธี GA เป็นอย่างมาก แสดงจากกราฟที่มีลักษณะเพิ่มขึ้นของเวลาในการประมวลผลอย่างรวดเร็ว ในขณะที่วิธี GA ยังคงใช้เวลาเพิ่มขึ้นจากเดิมไม่มากนัก

นอกจากนี้ จากกราฟยังแสดงให้เห็นว่า ทั้งจำนวนลูกค้า และประเภทของปัญหาเทียบเคียงมีผลต่อเวลาที่ใช้ในการประมวลผลด้วยคอมพิวเตอร์ด้วยวิธี HY นั่นคือ เมื่อจำนวนลูกค้าเพิ่มขึ้นเวลาที่ใช้ในการประมวลผลเพิ่มมากขึ้นแบบเลขชี้กำลัง และปัญหาประเภทที่ 1 ได้แก่ R1, C1 และ RC1 ที่มีลักษณะเป็นการจัดตารางเดินรถแบบสั้น ใช้เวลาในการประมวลผลยาวนานกว่าปัญหาประเภทที่ 2 ซึ่งได้แก่ R2, C2 และ RC2

ภาพที่ 67 เป็นการเปรียบเทียบเวลาในการประมวลผลสำหรับขั้นตอนวิธีการ HY ด้วยเครื่องคอมพิวเตอร์ทั้ง 6 เครื่อง ซึ่งในปัญหาเดียวกัน เวลาที่ใช้ในการประมวลอาจต่างกัน ทั้งนี้ขึ้นอยู่กับความสามารถของเครื่องคอมพิวเตอร์ จากผลการเปรียบเทียบแสดงให้เห็นว่าคอมพิวเตอร์เครื่องที่ 5 (Com 5) มีความสามารถสูงสุด หรือใช้เวลาในการประมวลผลน้อยที่สุด สำหรับปัญหาเทียบเคียงทุกประเภท



ภาพที่ 67 กราฟเปรียบเทียบเวลาในการประมวลผลสำหรับขั้นตอนวิธีการ HY

4. ตัวอย่างปัญหา (Problem Example)

ในหัวข้อที่ 4 เป็นการประยุกต์ใช้ขั้นตอนวิธีการกับตัวอย่างปัญหาในภาคอุตสาหกรรม โดยกำหนดตัวอย่างปัญหาเป็นบริษัทผลิตน้ำดื่มแห่งหนึ่ง ในเขตกรุงเทพมหานคร ที่มีลูกค้าจำนวน 25 เมือง กระจายอยู่ตามจุดต่างๆ ในเขตกรุงเทพมหานคร ปัจจุบันบริษัทประสบปัญหาการจัดเส้นทางเดินรถขนส่ง เนื่องจากสภาวะการจราจรที่คับคั่ง ทำให้เวลาเดินทางไม่แน่นอน ผู้จัดการบริษัทไม่ต้องการให้มีการเปลี่ยนแปลงเส้นทางบ่อยๆ จึงต้องการแผนการจัดเส้นทางเดินรถขนส่ง โดยมีวัตถุประสงค์ให้เกิดเวลาเดินทางรวมที่สั้นที่สุดและมีความทนทานต่อความไม่แน่นอนมากที่สุด และจำนวนรถขนส่งที่น้อยที่สุด และมีเงื่อนไขสำคัญ ได้แก่ รถขนส่งออกเดินทางและกลับมายังบริษัททุกครั้ง ลูกค้ามีความต้องการสินค้าไม่เท่ากัน รถขนส่งทุกคันเป็นรถประเภทเดียวกัน รถขนส่งมีความสามารถในการบรรทุกน้ำดื่มได้ครั้งละจำนวน 1,000 ขวด ลูกค้าแต่ละจุดมีการกำหนดกรอบเวลา และเวลาเดินทางประมาณเป็นช่วงเวลา โดยมีรายละเอียดดังตารางที่ 18

วิธีการแก้ปัญหา

4.1 สร้างรูปแบบของข้อมูลที่มีความไม่แน่นอน จากประสบการณ์และความชำนาญของวิศวกรฝ่ายโลจิสติกส์ของบริษัท สามารถแบ่งสถานการณ์ที่เป็นปัจจุบัน และที่คาดว่าจะเกิดในอนาคตได้ 4 เหตุการณ์ด้วยกัน, $|S| = 4$ และประมาณเปอร์เซ็นต์ความไม่แน่นอนของสภาพการจราจรด้วยค่าพารามิเตอร์, $\beta = 0.6$

4.2 เลือกเกณฑ์ที่เหมาะสมในการตัดสินผลเฉลยเชิงทนทาน สำหรับตัวอย่างปัญหานี้ เลือกใช้เกณฑ์ตัดสินผลเฉลย 3 เกณฑ์ด้วยกัน ประกอบด้วย สภาพทนทานสัมบูรณ์, Z_A ความแตกต่างทนทาน, Z_D และสภาพทนทานแบบสัมพัทธ์, Z_R

4.3 สร้างรูปแบบเชิงคณิตศาสตร์ ฟังก์ชั่นวัตถุประสงค์ และฟังก์ชั่นเงื่อนไข ดังสมการที่ (5) และ (6)

ตารางที่ 18 ตัวอย่างปัญหา RVRPTW

ลำดับที่	ตำแหน่งที่ตั้ง		ความต้องการสินค้า	กรอบเวลา		เวลาในการให้บริการ
	X	Y		เวลามาถึงเร็วสุด	เวลามาถึงช้าสุด	
1	35	35	0	0	1000	0
2	41	49	10	707	848	10
3	35	17	7	143	282	10
4	55	45	13	527	584	10
5	55	20	19	678	801	10
6	15	30	26	34	209	10
7	25	30	3	415	514	10
8	20	50	5	331	410	10
9	10	43	9	404	481	10
10	55	60	16	400	497	10
11	30	60	16	577	632	10
12	20	65	12	206	325	10
13	50	35	19	228	345	10
14	30	25	23	690	827	10
15	15	10	20	32	243	10
16	30	5	8	175	300	10
17	10	20	19	272	373	10
18	5	30	2	733	870	10
19	20	40	12	377	434	10
20	15	60	17	269	378	10
21	45	65	9	581	666	10
22	45	20	11	214	331	10
23	45	10	18	409	494	10
24	55	5	29	206	325	10
25	65	35	3	704	847	10
26	65	20	6	817	956	10

4.4 สร้างผลเฉลยเริ่มต้นด้วยวิธี MPFIH หาค่าผลเฉลยต่ำสุดในแต่ละสถานการณ์, s $g_s(X) = \min\{F(x,t_s) : x \in X\}$ โดยวิธี λ-LSD และ TS จากการประมวลผลด้วยขั้นตอนวิธีการแบบผสมผสานบนโปรแกรม MATLAB ผลเฉลยที่ได้แสดงดังตารางที่ 19

ตารางที่ 19 ผลเฉลยเวลาเดินทางต่ำสุดสำหรับตัวอย่างปัญหา, $|S| = 4$

ขั้นตอนวิธีการ	สถานการณ์			
	s_1	s_2	s_3	s_4
MPFIH	1188.00	1023.40	1052.70	1030.60
λ-LSD และ TS	442.54	499.47	524.34	455.85

4.5 เลือกค่ามากที่สุดจากทุกๆ สถานการณ์, S โดยที่ $h_x(S) = \max\{F(x,t_s) : s \in S\}$ จากตารางที่ 19 แสดงค่าเวลาเดินทางต่ำสุดที่เกิดขึ้นทั้ง 4 สถานการณ์ที่กำหนด จากนั้นหาผลเฉลยเชิงทนทานด้วยเกณฑ์การตัดสินใจสภาพทนทานสัมบูรณ์ (Z_A), ความแตกต่างทนทาน (Z_D) และสภาพทนทานแบบสัมพัทธ์ (Z_R) แสดงผลการคำนวณดังต่อไปนี้

ตารางที่ 20 ผลเฉลยสภาพทนทานสัมบูรณ์, Z_A คำนวณจากสมการที่ (2)

ผลเฉลย	สถานการณ์				$h_x(S)$ สูงสุด
	s_1	s_2	s_3	s_4	
x_1	442.54	761.81	766.07	724.35	766.07
x_2	597.60	499.47	734.91	736.63	736.63
x_3	689.34	726.00	524.34	742.30	742.30
x_4	704.19	715.50	660.89	455.85	715.50
$g_s(X)$ ต่ำสุด	442.54	499.47	524.34	455.85	715.50

ตารางที่ 21 ผลเฉลยความแตกต่างทนทาน, Z_D คำนวณจากสมการที่ (3)

ผลเฉลย	สถานการณ์				$h_x(S)$ สูงสุด
	s_1	s_2	s_3	s_4	
x_1	0.00	262.34	241.73	268.50	268.50
x_2	155.06	0.00	210.57	280.78	280.78
x_3	246.80	226.53	0.00	286.45	286.45
x_4	261.65	216.03	136.55	0.00	261.65
$g_s(X)$ ต่ำสุด	0.00	0.00	0.00	0.00	261.65

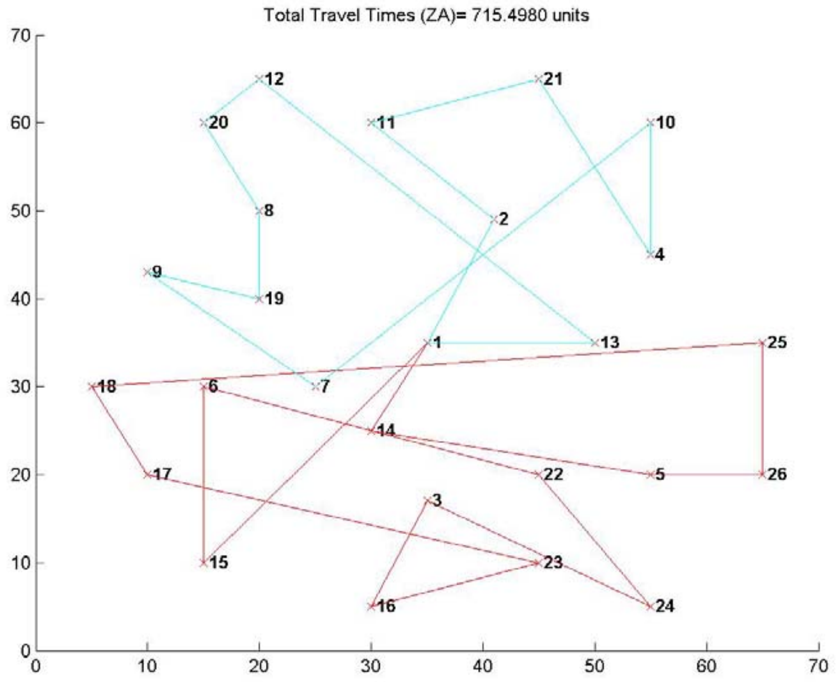
ตารางที่ 22 ผลเฉลยสภาพทนทานแบบสัมพัทธ์, Z_R คำนวณจากสมการที่ (4)

ผลเฉลย	สถานการณ์				$h_x(S)$ สูงสุด
	s_1	s_2	s_3	s_4	
x_1	0.000	0.525	0.461	0.589	0.589
x_2	0.350	0.000	0.402	0.616	0.616
x_3	0.558	0.454	0.000	0.628	0.628
x_4	0.591	0.433	0.260	0.000	0.591
$g_s(X)$ ต่ำสุด	0.000	0.000	0.000	0.000	0.589

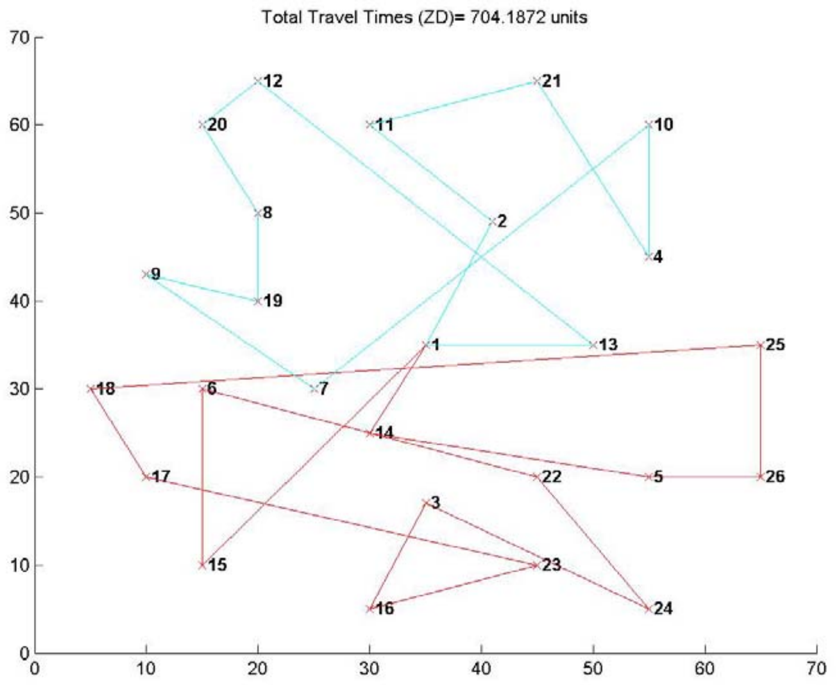
4.6 หาผลเฉลย x_R ซึ่งเป็นผลเฉลยเชิงทนทาน และให้ค่า $\min h_x(S)$ ซึ่งเป็นไปตามฟังก์ชั่นวัตถุประสงค์ (5) และเงื่อนไขข้อจำกัด (6) จากตารางที่ 20-22 ค่าเหมาะที่สุดเชิงทนทาน, Z ที่ได้จากการคำนวณ คือ Z_A =715.50, Z_D = 261.65 และ Z_R = 0.589 โดยผลเฉลยเชิงทนทานได้แก่ x_4 , x_4 และ x_1 ตามลำดับ ซึ่งมีเวลาเดินทางโดยรวมที่สั้นที่สุดและทนทานต่อความไม่แน่นอนมากที่สุด คือ 715.50, 704.19 และ 724.35 จำนวนรถขนส่งน้อยที่สุด คือ 2, 2 และ 3 คันตามลำดับ สรุปผลการคำนวณดังตารางที่ 23 และภาพที่ 68-70

ตารางที่ 23 ผลการคำนวณผลเฉลยเชิงทนทานสำหรับปัญหาตัวอย่าง

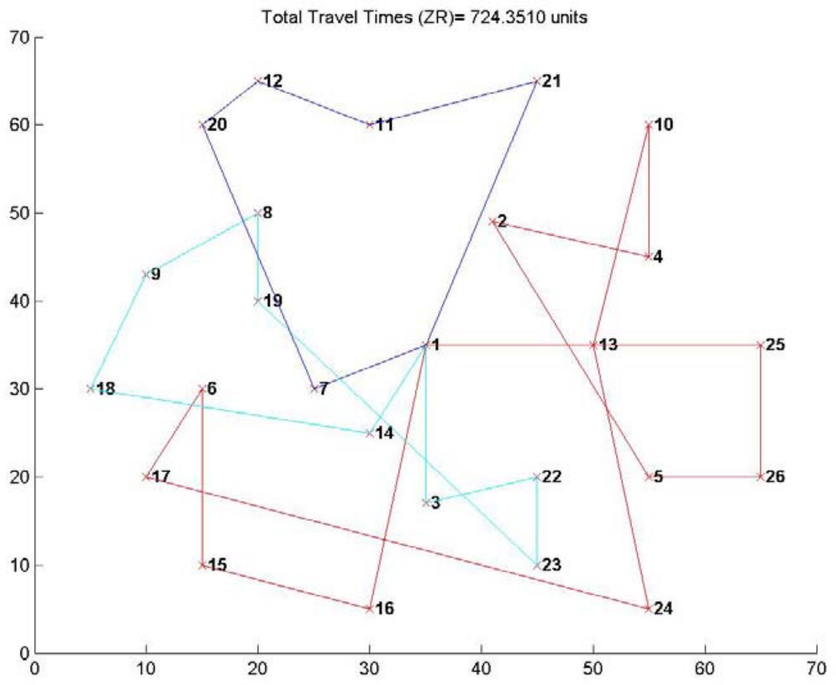
เกณฑ์การตัดสินใจ	ค่าเหมาะที่สุดเชิงทนทาน	ผลเฉลยเชิงทนทาน	เวลาเดินทาง	จำนวนรถขนส่ง
Z_A	715.50	x_4	715.50	2
Z_D	261.65	x_4	704.19	2
Z_R	0.589	x_1	724.35	3



ภาพที่ 68 ผลเฉลยสภาพทนทานสัมบูรณ์สำหรับปัญหาตัวอย่าง



ภาพที่ 69 ผลเฉลยความแตกต่างทนทานสำหรับปัญหาตัวอย่าง



ภาพที่ 70 ผลเฉลยสภาพทนทานแบบสัมพัทธ์สำหรับปัญหาตัวอย่าง

วิจารณ์

จากผลการทดลองแสดงให้เห็นว่าขั้นตอนวิธีการแบบผสมผสาน สามารถลดความซับซ้อนของปัญหาจากแบบแฟคทรอเรียล $O(n!)$ เป็นแบบเลขชี้กำลัง $O(\exp n)$ และสามารถหาค่าเหมาะที่สุดใกล้เคียงได้ ภายในเวลาที่ยอมรับได้ อย่างไรก็ตามการผลที่ได้เป็นเพียงการทดลองในสภาวะต่างๆ ที่งานวิจัยนี้ได้กำหนดขึ้น และยังไม่มีการพิสูจน์ทางทฤษฎีของความยากและความซับซ้อนของปัญหา เป็นเพียงประมาณการที่ได้จากผลการทดลองในงานวิจัยนี้เท่านั้น โดยวัตถุประสงค์หลักของงานวิจัย คือ ต้องการพัฒนาขั้นตอนวิธีการสำหรับหาผลเฉลยเชิงทนทานของปัญหา VRPTW แบบมีเวลาเดินทางไม่แน่นอน ซึ่งการวิจารณ์ผลมีรายละเอียดดังต่อไปนี้

1. วิจารณ์ผลการหาผลเฉลยเชิงทนทาน

ผลเฉลยเชิงทนทานซึ่งแสดงในรูปของเส้นทางการเดินรถขนส่งจากคลังสินค้า ไปยังลูกค้าและจากลูกค้าหนึ่งไปยังอีกลูกค้าหนึ่งที่กระจายอยู่ตามเมืองต่างๆ ดังภาพที่ 68-70 จะเห็นได้ว่าเมื่อเลือกใช้เกณฑ์ตัดสินที่แตกต่างกัน ทำให้ผลเฉลยที่ได้แตกต่างกันไปด้วย ทั้งนี้ความเหมาะสมขึ้นอยู่กับวัตถุประสงค์ของผู้ใช้งานเป็นสำคัญ และจากภาพที่ 68-70 แสดงการเปรียบเทียบระหว่างผลเฉลยที่ได้จากเกณฑ์ตัดสินแบบ Z_A , Z_D และ Z_R สำหรับปัญหาเทียบเคียงแต่ละประเภทของ Solomon พบว่าทุกปัญหาแสดงแนวโน้มไปในทิศทางเดียวกัน โดยที่ในกรณีของการหาจำนวนรถขนส่ง (Number of Vehicle) ค่า R ที่ได้มีค่าใกล้เคียงกัน และเข้าใกล้ศูนย์ บางปัญหาให้ค่าผลเฉลยที่ดีกว่าผลเฉลยเชิงกำหนดในทุกๆ การเปลี่ยนแปลงของค่าพารามิเตอร์ เช่น R101 และ RC201 ที่ให้ค่า R ที่ต่ำกว่าศูนย์ทั้งหมด และสำหรับปัญหา C201 ให้ค่าผลเฉลยเท่ากับผลเฉลยเชิงกำหนดในทุกๆ การเปลี่ยนแปลงของพารามิเตอร์ แสดงให้เห็นว่าผลเฉลยเชิงทนทานไม่จำเป็นต้องมีค่ามากกว่าผลเฉลยเชิงกำหนดเสมอไป ในบางครั้งอาจให้ค่าที่ดีกว่าหรือใกล้เคียงก็เป็นได้ ส่วนปัญหา C101 มีค่า R ที่สูงกว่าศูนย์ทั้งหมด แสดงให้เห็นว่าปัญหา C101 ผลเฉลยเชิงทนทานที่ได้ค่อนข้างมีค่าสูงกว่าผลเฉลยเชิงกำหนด

เมื่อเปรียบเทียบค่าจำนวนรถขนส่งต่ำสุดที่ได้จากเกณฑ์การตัดสินใจ พบว่าค่าที่ได้ไม่แตกต่างกันสำหรับทุกๆ ปัญหาเทียบเคียง ส่วนในกรณีของเวลาเดินทาง ค่า R ที่ได้สำหรับทุกปัญหาเทียบเคียง มีแนวโน้มที่ใกล้เคียงกันมาก สังเกตได้ว่าค่าเหมาะที่สุดเชิงทนทานสำหรับเกณฑ์ตัดสินแบบ Z_A ให้ค่าที่สูงกว่า Z_D และ Z_R ทั้งนี้สาเหตุเนื่องมาจากค่าเหมาะที่สุดเชิงทนทานที่ได้ถูกเลือกจากค่ามากที่สุดจากทุกสถานการณ์ที่เกิดขึ้น ข้อดีของวิธีนี้ คือ การคำนวณไม่ยุ่งยาก

ซับซ้อน ไม่ต้องเปรียบเทียบค่าเพื่อหาความแตกต่าง หรือความสัมพัทธ์กับค่าเหมาะที่สุดที่เกิดขึ้นในแต่ละสถานการณ์ แต่ข้อเสียที่สำคัญ คือ ผลเฉลยที่ได้ค่อนข้างห่างจากค่าเหมาะที่สุดมากเกินไป เมื่อเปรียบเทียบกับเกณฑ์ตัดสินอื่นๆ สำหรับเกณฑ์ตัดสินแบบ Z_D และ Z_R ส่วนใหญ่ค่า R ที่ได้ให้ผลที่มีแนวโน้มไปในทิศทางเดียวกันและใกล้เคียงกันมาก คือ มีค่าใกล้เคียงศูนย์ ถึงแม้ว่าการคำนวณจะมีความยุ่งยากมากกว่าแบบ Z_A แต่ผลเฉลยที่ได้มีความทนทานในขณะที่ไม่ห่างจากค่าเหมาะที่สุดสำหรับแต่ละสถานการณ์มากจนเกินไป

นอกจากนี้พารามิเตอร์ที่สำคัญที่มีผลต่อผลเฉลย คือ เปอร์เซ็นต์ความไม่แน่นอน จากผลการคำนวณที่ได้เมื่อเปอร์เซ็นต์ความไม่แน่นอนเพิ่มขึ้น ส่งผลให้ค่าเหมาะที่สุดเชิงทนทานมีค่าสูงขึ้นตามไปด้วย สังเกตได้จากกราฟที่แสดงแนวโน้มเพิ่มขึ้นอย่างต่อเนื่องเมื่อค่า β เพิ่มขึ้นจาก 0.2 ไปจนถึง 0.8 ทั้งนี้เกิดจากเวลาเดินทางในแต่ละเส้นทางที่ถูกกำหนดด้วยช่วงเวลาต่ำสุดและสูงสุด $[(1-\beta)t_{s_b},(1+\beta)t_{s_b}]$ เมื่อค่า β มากขึ้น ส่งผลให้ช่วงเวลากว้างขึ้นเช่นกัน ในขณะที่ขนาดของสถานการณ์ $|S|$ ไม่มีผลกระทบต่อค่าเหมาะที่สุดเชิงทนทานแต่อย่างใด จะเห็นได้ว่าเมื่อขนาดของสถานการณ์เพิ่มมากขึ้น ค่าเหมาะที่สุดมีการเปลี่ยนแปลงเพียงเล็กน้อย และไม่มีแนวโน้มที่จะเพิ่มสูงขึ้น เพราะฉะนั้นการกำหนดขนาดของสถานการณ์มากน้อย จึงไม่ใช่ปัจจัยสำคัญที่ส่งผลต่อคุณภาพของผลเฉลย ข้อสำคัญ คือ การกำหนดขนาดของสถานการ์ให้ครอบคลุมความไม่แน่นอนที่ต้องการพิจารณาทั้งหมดเท่านั้น

ตารางที่ 24 เปรียบเทียบข้อมูลนำเข้าและออกของวิธีการเข้าสู่ผลเฉลย

วิธีการเข้าสู่ผลเฉลย	ข้อมูลนำเข้า (Input)	ข้อมูลออก (Output)
กำหนดการเชิงกำหนด	ชุดข้อมูลเดี่ยว (Single Data Set)	ผลเฉลยเชิงกำหนด
กำหนดการเชิงสโทแคสติก	หลายชุดข้อมูล (Multiple Data Set), ทราบการแจกแจงหรือความน่าจะเป็นของข้อมูลในอดีต	ผลเฉลยเชิงสโทแคสติก
กำหนดการเชิงทนทาน	หลายชุดข้อมูล, ทราบสถานการณ์ที่เกิดขึ้นในปัจจุบัน, ประมาณการสถานการณ์ที่อาจเกิดขึ้นในอนาคตในรูปแบบของช่วงเวลาเดินทาง	ผลเฉลยเชิงทนทาน

ตารางที่ 25 เปรียบเทียบข้อดีและข้อเสียของวิธีการเข้าสู่ผลเฉลย

วิธีการเข้าสู่ผลเฉลย	ข้อดี	ข้อเสีย	การประยุกต์ใช้งาน
กำหนดการเชิงกำหนด	ใช้เวลาในการประมวลผลน้อย	ผลเฉลยไม่ครอบคลุมกรณีเกิดความไม่แน่นอน	- เหมาะสำหรับธุรกิจที่มีความแน่นอนของข้อมูลนำเข้า - ธุรกิจที่ต้องการค่าต้นทุนต่ำสุดและมีความเสี่ยงน้อย
กำหนดการเชิงสโทแคสติก	- ผลเฉลยครอบคลุมความไม่แน่นอนเป็นส่วนมาก - ผลเฉลยให้ค่าเหมาะต่ำสุด	- ใช้เวลาในการประมวลผลยาวนาน - รับประกันผลเฉลยเมื่อระบบมีความเสถียร - ความยุ่งยากซับซ้อนในขั้นตอนการวิเคราะห์ข้อมูล อาจจำเป็นต้องใช้ระดับวิศวกร หรือนักวิจัยที่มีความรู้เชิงวิชาการด้านสโทแคสติก - ความซับซ้อนในการออกแบบขั้นตอนวิธีการแก้ปัญหา	- เหมาะสำหรับธุรกิจที่มีความไม่แน่นอนของข้อมูล แต่สามารถประมาณการแจกแจง หรือความน่าจะเป็นได้ จากข้อมูลในอดีตได้ - ธุรกิจที่ต้องการค่าต้นทุนต่ำสุดจากความไม่แน่นอนทั้งหมด โดยยอมให้เกิดความเสี่ยงได้เล็กน้อยในกรณีที่เกิดสถานการณ์ที่เลวร้ายกว่าค่าคาดหวัง - ธุรกิจที่ต้องการวางแผนระยะสั้น (Short-Term Planning)
กำหนดการเชิงทนทาน	- ผลเฉลยครอบคลุมความไม่แน่นอนทั้งหมด - การวิเคราะห์การเกิดสถานการณ์ต่างๆ กระทำได้โดยง่าย โดยเจ้าหน้าที่เทคนิค หรือวิศวกรทั่วไป - รับประกันผลเฉลยสำหรับทุกสถานการณ์ที่กำหนดขึ้น - การออกแบบขั้นตอนวิธีการ และการนำไปใช้ไม่ยุ่งยากซับซ้อน	- ใช้เวลาในการประมวลผลยาวนาน ขึ้นอยู่กับจำนวนลูกค้า จำนวนเส้นทาง และจำนวนสถานการณ์ที่กำหนด - ให้ผลเฉลยที่เป็นค่าสูงสุดต่ำสุด (ส่วนมากให้ค่าที่สูงกว่ากำหนดการเชิงกำหนด และกำหนดการสโทแคสติก)	- เหมาะสำหรับธุรกิจที่มีความไม่แน่นอนของข้อมูล และสามารถสร้างสถานการณ์ความไม่แน่นอนที่เกิดขึ้นในปัจจุบัน และอนาคตได้ - ธุรกิจที่ไม่ต้องการต้นทุนต่ำสุด แต่ยอมรับต้นทุนสูงสุดต่ำสุด ที่รับประกันได้ว่าจะไม่มีความเสี่ยงใดๆ เกิดขึ้น ถึงแม้จะพบกับสถานการณ์ที่เลวร้ายที่สุด -ธุรกิจที่ต้องการวางแผนระยะยาว (Long-Term Planning)

2. วิจารณ์ผลการวัดสมรรถนะของวิธีการเข้าสู่ผลเฉลย

จากผลการวัดสมรรถนะของวิธีการเข้าสู่ผลเฉลย ซึ่งเปรียบเทียบระหว่างวิธีการสร้างสถานการณ์ กับวิธีการเข้าสู่ผลเฉลยแบบกรณีแย่ที่สุด และวิธีการเข้าสู่ผลเฉลยแบบคาดหวัง จากผลการทดลองแสดงให้เห็นว่า วิธีการเข้าสู่ผลเฉลยแบบการสร้างสถานการณ์มีสมรรถนะที่ดีและทนทานต่อความไม่แน่นอนมากที่สุด เมื่อเปรียบเทียบกับวิธีการอื่นๆ จากกราฟแสดงให้เห็นอย่างชัดเจนว่า ในช่วงเปอร์เซ็นต์ความไม่แน่นอนต่ำที่ β =0.2 สมรรถนะของทุกวิธีค่อนข้างดี โดยไม่มีความแตกต่าง และเมื่อเปอร์เซ็นต์ความไม่แน่นอนเพิ่มมากขึ้น หรือที่ค่า β มากกว่า 0.6 หรือ 60% ของค่าเวลาเดินทางพื้นฐาน (Travel Times Based-Case Value) หรือค่ากลางของช่วงเวลา $[l_{ij}, u_{ij}]$ มีผลกระทบทำให้วิธีการแบบกรณีแย่ที่สุด และแบบคาดหวัง มีสมรรถนะด้อยลง สังเกตได้จากค่า R ที่มีค่าสูงขึ้นเรื่อยๆ แสดงให้เห็นว่าผลเฉลยที่ได้มีความแตกต่างจากผลเฉลยเหมาะที่สุดที่เกิดขึ้นในแต่ละสถานการณ์ค่อนข้างมาก ซึ่งค่า R ที่สูงขึ้นเป็นการบ่งบอกถึงสมรรถนะของวิธีการที่ด้อยลง ในทางตรงกันข้ามในขณะที่เปอร์เซ็นต์ความไม่แน่นอนเพิ่มสูงขึ้น วิธีการเข้าสู่ผลเฉลยแบบการสร้างสถานการณ์ยังคงมีสมรรถนะที่ดี และทานทานต่อความไม่แน่นอน สังเกตได้จากกราฟที่ค่อนข้างมีระดับคงที่ ไม่มีการเปลี่ยนแปลงเมื่อเปอร์เซ็นต์ความไม่แน่นอนเพิ่มขึ้น

นอกจากนี้จากกราฟยังแสดงให้เห็นว่าขนาดของจำนวนสถานการณ์ $|S|$ ที่ใช้ในการทดลอง ไม่มีผลกระทบต่อสมรรถนะของวิธีการเข้าสู่ผลเฉลยแบบการสร้างสถานการณ์ โดยเมื่อทดลองเปลี่ยน $|S|$ = 10, 20, 40 และ 80 สมรรถนะของวิธีการยังคงไปในทิศทางเดียวกัน ซึ่งค่าเฉลี่ย R_{10} = 0.30, R_{20} = 0.31, R_{40} = 0.36, และ R_{80} = 0.32

3. วิจารณ์ผลการวัดสมรรถนะของขั้นตอนวิธีการ

3.1 คุณภาพของผลเฉลย ผลการเปรียบเทียบขั้นตอนวิธีการ HY และ GA ในการหาจำนวนรถขนส่งต่ำสุด พบว่าวิธี HY มีสมรรถนะดีกว่าวิธี GA ในทุกๆ ปัญหาเทียบเคียง โดยวิธี HY มีค่า $\overline{R} < 1$ แต่มีค่าเบี่ยงเบนค่อนข้างสูง สังเกตได้จากกราฟที่มีค่าขึ้นลงอยู่บริเวณศูนย์ ในขณะที่ GA มีค่า $\overline{R} > 1$ ในปัญหาประเภท R1, C1, C2 และ RC1 และในกรณีการหาเวลาขนส่งต่ำสุด เมื่อเปรียบเทียบผลเฉลยกับปัญหาเทียบเคียง พบว่าวิธี HY ยังคงให้ผลเฉลยที่ดีกว่าวิธี GA ในปัญหาเทียบเคียงทุกประเภท สังเกตได้จากกราฟที่มีค่าอยู่ใกล้ศูนย์ และยังมีค่าเบี่ยงเบนต่ำกว่าในกรณีหาจำนวนรถขนส่งต่ำสุด สำหรับวิธี HY มีค่า $\overline{R} < 1$ ในปัญหาทุกประเภท ในขณะที่ GA มีค่า $\overline{R} > 1$ และมีความเบี่ยงเบนสูง

จากผลการทดลองแสดงให้เห็นว่าวิธี HY มีสมรรถนะที่ดีสำหรับปัญหาประเภท R1, R2, RC1 และ RC2 ในขณะที่สมรรถนะด้อยลงเล็กน้อยสำหรับปัญหาประเภท C1 และ C2 อาจมาจากรูปแบบของปัญหาที่มีจัดกลุ่มลูกค้าไว้ก่อนแล้ว เมื่อขั้นตอนวิธีการ HY เข้าไปจัดลำดับลูกค้าใหม่ อาจส่งผลกระทบต่อผลเฉลยที่ด้อยคุณภาพลง จากกราฟแสดงให้เห็นชัดเจนว่า ขนาดของปัญหาที่แตกต่างกัน ไม่มีผลกระทบต่อคุณภาพของผลเฉลยทั้งในกรณีหาจำนวนรถขนส่งต่ำสุด และเวลาเดินทางต่ำสุด

3.2 เวลาในการประมวลผลด้วยคอมพิวเตอร์ จากกราฟแสดงผลการเปรียบเทียบเวลาในการประมวลผลระหว่างขั้นตอนวิธีการ HY และ GA พบว่าเมื่อปัญหามีขนาดเล็ก หรือมีจำนวนลูกค้า 25 เมือง วิธี HY ใช้เวลาในการประมวลผลน้อยกว่า GA เนื่องจากในช่วงเริ่มต้นของวิธี GA จะใช้เวลาส่วนมากในการสร้างกลุ่มประชากรของผลเฉลยเริ่มต้น ซึ่งอาจมีจำนวนมากน้อยแล้วแต่การกำหนดค่าเริ่มต้น สำหรับการทดลองนี้กำหนดให้ประชากรเริ่มต้นเท่ากับ 100 ผลเฉลย ในขณะที่วิธี HY เริ่มจากผลเฉลยเริ่มต้นเพียงอันเดียว เมื่อจำนวนลูกค้าเพิ่มขึ้นเป็น 50 เมือง วิธี GA ใช้เวลาน้อยกว่าวิธี HY ประมาณ 3 เท่า และเมื่อจำนวนลูกค้าเพิ่มขึ้นถึง 100 เมือง วิธี HY ใช้เวลาในการประมวลยาวนานกว่าวิธี HY ถึง 20 เท่า

นอกจากนี้ปัญหาเทียบเคียงแต่ละประเภท ยังใช้เวลาในการประมวลผลที่แตกต่างกัน จากผลการเปรียบเทียบการประมวลผลเฉพาะวิธีการ HY ที่เกิดจากคอมพิวเตอร์จำนวน 6 เครื่อง ซึ่งมีแนวโน้มไปในทิศทางเดียวกัน คือ เวลาที่ใช้ในการประมวลจะเพิ่มมากขึ้น เมื่อจำนวนลูกค้ามากกว่า 50 เมือง และเวลาที่ใช้ในการประมวลผลสำหรับปัญหาเทียบเคียงประเภทที่ 1 (R1, C1 และ RC 1) จะใช้เวลาในการประมวลผลยาวนานกว่าประเภทที่ 2 (R2, C2 และ RC 2) โดยประมาณ 3 เท่า ทั้งนี้เนื่องจากปัญหาประเภทที่ 1 เป็นลักษณะการจัดเส้นทางเดินรถขนส่งแบบสั้น ทำให้ต้องใช้จำนวนรถหลายคัน หรือเกิดเส้นทางจำนวนมาก ทำให้ขั้นตอนวิธีการในช่วงการค้นหาผลเฉลยเมื่อมีการสลับลูกค้าระหว่างเส้นทางทุกคู่การจัดที่เป็นไปได้ ใช้เวลาในการประมวลผลนาน ในขณะที่ประเภทที่ 2 เป็นการจัดเส้นทางแบบยาว คือ มีจำนวนลูกค้าในแต่เส้นทางมาก ทำให้ใช้รถขนส่งน้อยคัน และแต่ละคันต้องให้บริการลูกค้าจำนวนมากกว่า 30 เมือง ในกรณีนี้จึงเกิดเส้นทางน้อยกว่าประเภทที่ 1 ทำให้ขั้นตอนวิธีการไม่เสียเวลาในการค้นหาผลเฉลยระหว่างเส้นทางแต่ละคู่

สรุปและข้อเสนอแนะ

สรุป

จากผลและวิจารณ์ผลการทดลอง สามารถสรุปได้ว่าขั้นตอนวิธีการแบบผสมผสานที่พัฒนาขึ้นภายใต้วิธีการเข้าสู่ผลเฉลยเชิงทนทาน สามารถหาผลเฉลยที่มีความทนทานต่อความไม่แน่นอนในระดับต่างๆได้ อันนำไปสู่วัตถุประสงค์หลักของงานวิจัย โดยผลเฉลยนี้ประกอบด้วย จำนวนรถขนส่งต่ำสุด และเวลาเดินทางรวมต่ำสุด ที่สามารถนำไปใช้ในการจัดเส้นทางเดินรถขนส่งแบบมีคลังสินค้าเดียว และมีลูกค้าจำนวนมาก ซึ่งบรรลุตามวัตถุประสงค์ย่อยที่ได้กล่าวไว้ในข้างต้น นอกจากนี้งานวิจัยได้มีการเปรียบเทียบสมรรถนะอันประกอบด้วย เกณฑ์การตัดสินใจเชิงทนทาน วิธีการเข้าสู่ผลเฉลย และขั้นตอนวิธีการ ซึ่งผลที่ได้สรุปให้เห็นอย่างชัดเจนว่า เกณฑ์การตัดสินใจที่เหมาะสมงานวิจัยนี้มากที่สุด คือ Z_D และ Z_R ที่แสดงสมรรถนะใกล้เคียงกับผลเฉลยเชิงกำหนด โดยให้ค่า R ที่ต่ำกว่า Z_A

ส่วนวิธีการเข้าสู่ผลเฉลยเชิงทนทาน ในงานวิจัยนี้ได้นำเสนอวิธีการเข้าสู่ผลเฉลยแบบการสร้างสถานการณ์ จากผลการทดลองแสดงให้เห็นว่าเมื่อเปรียบเทียบกับวิธีการเข้าสู่ผลเฉลยแบบอื่นๆ ที่เลือกนำมาเปรียบเทียบ พบว่าวิธีการเข้าสู่ผลเฉลยแบบการสร้างสถานการณ์ มีความทนทานต่อเปอร์เซ็นต์ความไม่แน่นอนที่มีการเปลี่ยนแปลงเพิ่มขึ้น ในขณะที่วิธีการอื่นๆ มีสมรรถนะด้อยลงอย่างรวดเร็วเมื่อเปอร์เซ็นต์ความไม่แน่นอนมีค่าสูงขึ้น และส่วนสุดท้ายเป็นการเปรียบเทียบขั้นตอนวิธีการแบบผสมผสานกับขั้นตอนวิธีการ GA ซึ่งพิจารณาสมรรถนะใน 2 ด้านด้วยกัน คือ คุณภาพของผลเฉลย และเวลาที่ใช้ในการประมวลผล ผลที่ได้แสดงให้เห็นอย่างชัดเจนว่า ผลเฉลยที่ได้จากขั้นตอนวิธีการแบบผสมผสาน มีคุณภาพที่ดีกว่าในทุกปัญหาเทียบเคียงของ Solomon ในขณะที่เวลาในการประมวลผลด้อยกว่าขั้นตอนวิธีการ GA เมื่อจำนวนลูกค้ามีค่ามากกว่า 50 เมืองขึ้นไป ข้อด้อยในจุดนี้ มิได้สร้างปัญหาให้แก่ขั้นตอนวิธีการแต่อย่างใด เนื่องจากวิธีการเข้าสู่ผลเฉลยเชิงทนทานนี้ เป้าหมายหลัก คือ ใช้สำหรับการวางแผนระยะยาวที่ครอบคลุมสถานการณ์ที่เกิดขึ้นในปัจจุบัน และคาดว่าจะเกิดขึ้นในอนาคต โดยการประมวลผลครั้งหนึ่งอาจให้ผลเฉลยที่คงทนถาวรตราบเท่าที่เหตุการณ์ในอนาคตยังคงอยู่ในกรอบที่ผู้ตัดสินใจได้วางแผนสถานการณ์รองรับไว้ ซึ่งไม่จำเป็นต้องมีการประมวลผลบ่อยครั้งเช่นเดียวกับการหาผลเฉลยเชิงกำหนด หรือเชิงสโทแคสติก ซึ่งเป็นเพียงการวางแผนล่วงหน้าระยะสั้น เมื่อสถานการณ์ใดๆ เกิดขึ้นจริง อาจจำเป็นต้องมีการประมวลผลใหม่เพื่อให้ได้ผลเฉลยที่สอดคล้องกับสถานการณ์จริงมากขึ้น หรือที่เรียกว่า ขั้นตอนวิธีการแบบ 2 เฟส (Two Phase Algorithm) ซึ่งแบ่งเป็นการวางแผนล่วงหน้า (Pre-

Planning) และการวางแผนภายหลัง (Post-Planning) ขั้นตอนวิธีการจึงจำเป็นต้องถูกสร้างให้มีความรวดเร็วในการประมวลผล

ข้อเสนอแนะ

ในงานวิจัยนี้มุ่งเน้นการหาผลเฉลยเชิงทนทาน ซึ่งอาจให้ผลเฉลยที่แตกต่างไปจากวิธีการเข้าสู่ผลเฉลยแบบอื่นๆ เพราะฉะนั้นการนำผลเฉลยไปทำเป็นปัญหาเทียบเคียง ควรมีการพิจารณาให้ถี่ถ้วนในแง่ของวัตถุประสงค์การใช้งาน ขอบเขตของปัญหา และค่าพารามิเตอร์ที่ใช้ในการสร้างขั้นตอนวิธีการ ซึ่งได้อธิบายโดยละเอียดไว้ในส่วนเนื้อหาข้างต้น โดยเฉพาะการกำหนดค่าน้ำหนักโทษสำหรับกรณีเกิดเวลารอคอย กรณีเกิดเวลาล่าช้า และกรณีเกิดเวลาล่วงเวลา ซึ่งอาจมีผลต่อผลเฉลยที่ได้ ในการเปรียบเทียบจึงควรใช้ค่าที่เท่ากัน หรือใกล้เคียงกันมากที่สุด

สำหรับงานวิจัยนี้ได้นำเสนอสมรรถนะในการแก้ปัญหา VRPTW ที่มีจำนวนลูกค้าขนาด 100 เมือง ในกรณีที่พบปัญหาที่มีจำนวนลูกค้ามากกว่า 100 เมือง ยังคงสามารถใช้ขั้นตอนวิธีการที่พัฒนาขึ้นได้ หากมีคอมพิวเตอร์ที่มีกำลังมากพอที่สามารถประมวลผลได้ในเวลาอันรวดเร็ว หรือในเวลาที่ยอมรับได้ เนื่องจากขั้นตอนวิธีการดังกล่าวไม่มีข้อจำกัดจำนวนลูกค้า ผู้ตัดสินใจสามารถเปลี่ยนข้อมูลนำเข้าได้ตามปัญหาจริง ทั้งนี้ควรพิจารณาในเรื่องของเวลาในการประมวลผลเป็นสำคัญ เพราะเมื่อมีจำนวนลูกค้ามากขึ้นเวลาในการประมวลผลยิ่งมากขึ้นเป็นเท่าทวีคูณ เนื่องจากขั้นตอนวิธีการถูกออกแบบให้มุ่งเน้นหาผลเฉลยที่ดีที่สุด โดยมิได้จำกัดเรื่องเวลาในการค้นหา ซึ่งเป็นปัญหาหนึ่งซึ่งผู้วิจัยสนใจที่จะพัฒนาขั้นตอนวิธีการนี้ต่อไปในอนาคต โดยมีแนวคิดที่จะนำเอาข้อดีของขั้นตอนวิธีการแบบผสมผสาน อันได้แก่ ผลเฉลยที่ให้คุณภาพดี ร่วมกับการประมวลผลที่รวดเร็วของขั้นตอนวิธีการ GA หรือ SA ดังที่ได้นำเสนอไว้ในการตรวจเอกสาร เพื่อให้ขั้นตอนวิธีการมีประสิทธิภาพสูงทั้งในด้านคุณภาพของผลเฉลย และประหยัดเวลาในการประมวลผลด้วยคอมพิวเตอร์

ถึงแม้ในงานวิจัยนี้จะมีการกำหนดขอบข่ายที่ชัดเจน แต่ในทางปฏิบัติขอบข่าย หรือเงื่อนไขต่างๆ สามารถปรับปรุงเปลี่ยนแปลง หรือเพิ่มเติมเข้าไปในขั้นตอนวิธีการที่พัฒนาขึ้น ทั้งนี้เพื่อให้สอดคล้องกับระบบงานจริงที่เกิดขึ้น ตัวอย่างเช่น ในกรณีที่ปัญหามีจำนวนความต้องการสินค้ามากกว่าความสามารถในการบรรทุกสินค้าของรถขนส่ง 1 คัน ขั้นตอนวิธีการสามารถเพิ่มเติมขั้นตอนในการแบ่งจำนวนความต้องการออกเป็น 2 ส่วน (หรือมากกว่า 2 ส่วนแล้วแต่กรณี) โดยให้ส่วนใดส่วนหนึ่งเท่ากับความสามารถในการบรรทุกของรถขนส่ง 1 คัน และความต้องการส่วนที่เหลือจะถูกจัดเข้าสู่กระบวนการของขั้นตอนวิธีการตามปกติ หมายความว่า ลูกค้ารายที่มีความ

ต้องการมากกว่าความสามารถในการบรรทุกสินค้า จะต้องใช้รถขนส่งจำนวนมากกว่า 1 คัน ในการให้บริการ

นอกจากนี้การขยายงานวิจัยในอนาคต เพื่อให้สอดคล้องกับปัญหาจริง คือ การเพิ่มความหลากหลายของรถขนส่ง หรือการกำหนดให้มีความสามารถในการบรรทุกสินค้าที่แตกต่างกันไป เช่น ความสามารถของรถบรรทุกใหญ่ = 1,000 หน่วย รถบรรทุกเล็ก = 100 หน่วย หรือรถกระบะ = 50 หน่วย เป็นต้น ซึ่งในกรณีนี้จะต้องเข้าไปแก้ไขการกำหนดค่าพารามิเตอร์ที่เกี่ยวกับความสามารถในการบรรทุกสินค้า และเงื่อนไขเกี่ยวกับจำนวนความต้องการของลูกค้าในแต่ละเส้นทาง จะต้องไม่เกินความสามารถในการบรรทุกสินค้าของรถขนส่งประเภทนั้นๆ หรืออาจใช้รถบรรทุกใหญ่สำหรับปัญหาที่มีการจัดตารางเดินรถแบบยาว (มีจำนวนลูกค้ามากกว่า 30 เมืองในเส้นทาง) และใช้รถบรรทุกเล็ก หรือรถกระบะสำหรับปัญหาที่มีการจัดตารางเดินรถแบบสั้น (มีจำนวนลูกค้า 3-5 เมืองในเส้นทาง) อย่างไรก็ดีในการแก้ปัญหาต่างๆ อาจต้องมีการปรับเปลี่ยนขั้นตอนวิธีการที่พัฒนาขึ้น และส่งผลต่อเวลาในการประมวลผลโดยคอมพิวเตอร์ ซึ่งจำเป็นต้องมีการทดลองเพื่อให้ได้ผลเฉลยเชิงทนทานต่อไป

เอกสารและสิ่งอ้างอิง

ณกร อินทร์พยุง. 2548. **การแก้ปัญหาการตัดสินใจในอุตสาหกรรมการขนส่งและลอจิสติกส์.** ครั้งที่ 1. ซีเอ็ดยูเคชั่น, กรุงเทพฯ.

Aarts, E. and J.K. Lenstra. 1997. **Local Search in Combinatorial Optimization.** Wiley and Sons, Inc.

Agarwal, Y., K. Mathur and H.M. Salkin. 1989. A set-partitioning-based exact algorithm for the vehicle routing problem. **Networks** 19 (7): 731-749.

Beatrice, O., J. R. Brian and H. Franklin. 2006. Multi-objective genetic algorithms for vehicle routing problem with time windows. **Applied Intelligence** 24 (1): 17-30.

Ben-Tal, A. and A. Nemirovski. 1998. Robust convex optimization. **Mathematics of Operations Research** 23 (4): 769-805.

_______ . 1999. Robust solutions to uncertain programs. **Operations Research Letters** 25 (1): 1-13.

_______ . 2000. Robust solutions of linear programming problem contaminated with uncertain data. **Mathematical Programming** 88 (3): 411-424.

Bertsimas, D.J. 1992. A vehicle routing problem with stochastic demand. **Operations Research** 40 (3): 574-585.

Carraway, R.L., T.L. Morin and H. Moskowitz. 1989. Generalized dynamic programming for stochastic combinatorial optimization. **Operations Research** 37 (5): 819-829.

Charnes, A. and W. Cooper. 1959. Chance-constrained programming. **Management Science** 6 (1): 73-79.

Christofides, N. 1985. Vehicle routing, pp. 431-448. *In* The traveling salesman problem, Lawler, Lenstra, R. Kan and Shmoys, eds. **A Guided Tour of Combinatorial Optimization.** John Wiley.

_______, A. Mingozzi and P. Toth. 1981. State space relaxation procedures for the computation of bounds to routing problems. **Networks** 11 : 145-164.

Clarke, G. and J. Wright. 1964. Scheduling of vehicles from a central depot to a number of delivery points. **Operations Research** 12 (4): 568-581.

Committee on Stochastic Programming (COSP). 2008. **Stochastic Programming Community Home Page.** Available Source: http://stoprog.org, May 7, 2008.

Dantzig, G. B. 1955. Linear programming under uncertainty. **Management Science** 1(3-4): 197-206.

_______and J. H. Ramser. 1959. The truck dispatching problem. **Management Science** 6 (1): 80–91.

Dorigo, M., V. Maniezzo and A. Colorni. 1996. The ant system: Optimization by a colony of cooperating agents. **IEEE Trans. on Systems** 26 (1): 1-13.

El-Ghaoui, L. and H. Lebret. 1997. Robust solutions to least-square problems to uncertain data matrices. **SIAM Journal on Matrix Analysis and Applications** 18 (4): 1035-1064.

Fisher, M.L. 1993. Vehicle routing, pp. 1-33. *In* M.O. Ball, T.L. Magnanti, C.L. Monma and G.L. Nemhauser, eds. **Handbooks in Operations Research and Management Science.** Elsevier Science.

_______and R. Jaikumar. 1981. A generalized assignment heuristic for vehicle routing. **Networks** 11 (2): 109-124.

Gendreau M., G. Laporte and J.Y. Potvin. 1997. Vehicle routing, pp. 311-336. *In* Modern heuristics. **Local Search in Combinatorial Optimization.** Wiley and Sons.

Gillett, B. and L. Miller. 1974. A heuristic algorithm for the vehicle-dispatch problem. **Operations Research** 22 (2): 340-349.

Glover, F. 1989. Tabu Search-Part I. **ORSA Journal on Computing** 1 (3): 190-206.

Golden, B.L., A.A. Assad and M.O. Ball. 1983. Routing and scheduling of vehicles and crews. **Computer and Operations Research** 10 (2): 63-211.

Herrmann, J.W. 1999. A genetic algorithm for minimax optimization problems, pp. 1099-1103. *In* P.J. Angeline, Z. Michalewicz, M. Schoenauer, X. Yao and A. Zalzala, eds. **The 1999 Congress on Evolutionary Computation.** IEEE Press, Washington DC.

Jula, H., M. Dessouky and P. Ioannou. 2005. Truck route planning in non-stationary stochastic networks with time-windows at customer locations. **IEEE Trans. on Intelligent Transportation Systems.**

Kao, E. 1978. A preference order dynamic program for a stochastic travelling salesman problem. **Operations Research** 26 (6): 1033-1045.

Kouvelis, P. and G. Yu. 1977. Robust discrete optimization and its applications. **Kluwer Academic Publishers,** Netherlands.

Laporte G., F.V. Louveaux and H. Mecure. 1992. The vehicle routing problem with stochastic travel times. **Transportation Science** 26 (3): 161-170.

Metropolis, N., A.W. Rosenbluth, M.N. Rosenbluth, A.H. Teller and E. Teller. 1953. Equations of state calculations by fast computing machine. **Journal of Chemical Physic** 21 (6): 1087-1092.

Muhlenbein, H. 1997. Genetic algorithm, pp. 137-172. *In* E. Aarts and J.K. Lenstra, eds. **Local Search in Combinatorial Optimization.** Wiley and Sons, Inc.

Montemanni, R., J. Barta, M. Mastrolilli and L.M. Gambardella. 2007. The robust traveling salesman problem with interval data. **Transportation Science** 41 (3): 366-381.

Ordonez, F., Z. Shen and M. Dessouky. 2006. The minimum unmet demand stochastic vehicle routing problem. **USC ISE Working paper** #2006-07.

Osman, I. H. 1993. Vehicle routing and scheduling: Applications, algorithms and developments. **Proceedings of the International Conference on Industrial Logistics**, Université de Rennes, France, paper 2.3, pp. 1-37.

Paletta, P. 2002. The period traveling salesman problem: A new heuristic algorithm. **Computer and Operations Research** 29 (10): 1343-1352.

Potvin, J.Y. and J.M. Rousseau. 1995. An exchange heuristic for routing problems with time windows. **Journal of the Operations Research Society** 46 (12): 1433–1446.

Savelsbergh, M.W.P. 1985. Local search for routing problems with time windows. **Annals of Operations Research** 4 (1-4): 285-305.

Solomon, M.M. 1987. Algorithms for the vehicle routing and scheduling problems with time windows constraints. **Operations Research** 35 (2): 254–265.

_______ . 2005. **VRPTW benchmark problems**. Available Source: http://web.cba.neu.edu/~msolomon/problems.htm, March 24, 2005.

Taillard, E., P. Badeau, M. Gendreau, F. Guertin and J.Y. Potvin. 1997. A tabu search heuristic for the vehicle routing problem with soft time windows. **Transportation Science** 31 (2): 170-186.

Thangiah, S.R. 1993. Vehicle routing with time windows using genetic algorithms**. Technical Report.** 1-23.

_______. 1999. Complex Structures, pp. 347-381. *In* L. Chambers, ed. **Practical Handbook of Genetic Algorithms.** CRC Press, Australia.

_______, I.H. Osman, R. Vinayagamoorthy and T. Sun. 1995. Algorithms for the vehicle routing problems with deadlines. **American Journal of Mathematical and Management Science** 13 (3-4): 323-355.

_______, J.Y. Potvin and _______. 1996. Heuristics approaches to vehicle routing with backhauls and time windows. **Computer and Operations Research** 23 (11): 1043-1057.

Tillman, F. 1969. The multiple terminal delivery problems with probabilistic demands. **Transportation Science** 3 (3): 192-204.

Titiporn, T. 2001. **Bi-criteria vehicle routing problem**. M.S. Thesis, Kasetsart University.

Toth, P. and D. Vigo. 2001. **The vehicle routing problem: SIAM monographs on discrete mathematics and applications.** SIAM, Philadelphia.

ภาคผนวก

ภาคผนวก ก

ผลเฉลยเชิงทนทาน

ตารางผนวกที่ ก1 ผลเฉลยเชิงทนทานสำหรับปัญหา R101

Problems	$\|S\|$	β	Criterion	Number of Vehicles	Travel Times	R_NV	R_TT	CPU(s)
R101.25	10	0.2	Z_A	6	600.44	0.20	0.21	1957.00
			Z_D	5	557.35	0.00	0.12	
			Z_R	5	557.35	0.00	0.12	
R101.25	10	0.4	Z_A	6	675.37	0.20	0.36	1762.00
			Z_D	6	612.55	0.20	0.23	
			Z_R	6	612.55	0.20	0.23	
R101.25	10	0.6	Z_A	5	826.03	0.00	0.66	1731.00
			Z_D	6	690.59	0.20	0.39	
			Z_R	6	690.59	0.20	0.39	
R101.25	10	0.8	Z_A	5	1056.31	0.00	1.12	1751.00
			Z_D	5	906.47	0.00	0.82	
			Z_R	5	906.47	0.00	0.82	
R101.25	20	0.2	Z_A	5	612.32	0.00	0.23	3807.96
			Z_D	5	582.17	0.00	0.17	
			Z_R	5	582.17	0.00	0.17	

ตารางผนวกที่ ก1 (ต่อ)

Problems	$\|S\|$	β	Criterion	Number of Vehicles	Travel Times	R_NV	R_TT	CPU(s)
R101.25	20	0.4	Z_A	5	688.34	0.00	0.38	3616.14
			Z_D	5	577.16	0.00	0.16	
			Z_R	5	577.16	0.00	0.16	
R101.25	20	0.6	Z_A	5	1136.08	0.00	1.28	3513.06
			Z_D	6	698.62	0.20	0.40	
			Z_R	6	698.62	0.20	0.40	
R101.25	20	0.8	Z_A	5	1178.89	0.00	1.37	3303.07
			Z_D	5	805.43	0.00	0.62	
			Z_R	5	805.43	0.00	0.62	
R101.25	40	0.2	Z_A	5	616.00	0.00	0.24	7604.91
			Z_D	5	591.74	0.00	0.19	
			Z_R	5	591.74	0.00	0.19	
R101.25	40	0.4	Z_A	7	667.18	0.40	0.34	7283.21
			Z_D	5	596.71	0.00	0.20	
			Z_R	5	596.71	0.00	0.20	

ตารางผนวกที่ ก1 (ต่อ)

Problems	$\|S\|$	β	Criterion	Number of Vehicles	Travel Times	R_NV	R_TT	CPU(s)
R101.25	40	0.6	Z_A	7	771.23	0.40	0.55	7078.69
			Z_D	5	642.53	0.00	0.29	
			Z_R	5	642.53	0.00	0.29	
R101.25	40	0.8	Z_A	5	1273.10	0.00	1.56	6795.05
			Z_D	5	728.07	0.00	0.46	
			Z_R	5	878.36	0.00	0.77	
R101.25	80	0.2	Z_A	6	605.42	0.20	0.22	15128.58
			Z_D	5	581.86	0.00	0.17	
			Z_R	5	581.86	0.00	0.17	
R101.25	80	0.4	Z_A	6	682.57	0.20	0.37	13899.18
			Z_D	5	667.33	0.00	0.34	
			Z_R	5	667.33	0.00	0.34	
R101.25	80	0.6	Z_A	6	717.535	0.20	0.44	13947.738
			Z_D	5	676.6968	0.00	0.36	
			Z_R	5	602.5032	0.00	0.21	

ตารางผนวกที่ ก1 (ต่อ)

Problems	$\|S\|$	β	Criterion	Number of Vehicles	Travel Times	R_NV	R_TT	CPU(s)
R101.25	80	0.8	Z_A	6	860.1663	0.20	0.73	13761.068
			Z_D	5	898.1133	0.00	0.81	
			Z_R	5	898.1133	0.00	0.81	

หมายเหตุ คำอธิบายตาราง

Problems = ตัวอย่างปัญหา RVRPTW, $|S|$ = ขนาดของสถานการณ์, β = เปอร์เซ็นต์ความไม่แน่นอน, Criterion = เกณฑ์การตัดสินใจ

Number of Vehicles = จำนวนรถขนส่งที่ได้จากผลเฉลยเชิงทนทาน, Travel Times = เวลาเดินทางรวมที่ได้จากผลเฉลยเชิงทนทาน

R_NV = สัดส่วนจำนวนรถขนส่งเมื่อเทียบกับผลเฉลยเชิงกำหนด (Based-Case Number of Vehicles = 5)

R_TT = สัดส่วนเวลาเดินทางโดยรวมเมื่อเทียบกับผลเฉลยเชิงกำหนด (Based-Case Travel Times = 497.30)

CPU(s) = เวลาประมวลผลด้วยเครื่องคอมพิวเตอร์ (วินาที)

ตารางผนวกที่ ก2 ผลเฉลยเชิงทนทานสำหรับปัญหา R201

Problems	$\|S\|$	β	Criterion	Number of Vehicles	Travel Times	R_NV	R_TT	CPU(s)
R201.25	10	0.2	Z_A	3	578.40	0.00	0.21	584.34
			Z_D	3	571.22	0.00	0.19	
			Z_R	3	571.22	0.00	0.19	
R201.25	10	0.4	Z_A	3	729.80	0.00	0.52	573.25
			Z_D	3	629.14	0.00	0.31	
			Z_R	3	629.14	0.00	0.31	
R201.25	10	0.6	Z_A	3	828.64	0.00	0.73	628.86
			Z_D	2	680.78	-0.33	0.42	
			Z_R	3	708.60	0.00	0.48	
R201.25	10	0.8	Z_A	3	1217.45	0.00	1.54	614.58
			Z_D	4	807.62	0.33	0.68	
			Z_R	4	807.62	0.33	0.68	
R201.25	20	0.2	Z_A	3	828.64	0.00	0.73	631.84
			Z_D	2	680.78	-0.33	0.42	
			Z_R	3	708.60	0.00	0.48	

ตารางผนวกที่ ก2 (ต่อ)

Problems	$\|S\|$	β	Criterion	Number of Vehicles	Travel Times	R_NV	R_TT	CPU(s)
R201.25	20	0.4	Z_A	3	693.39	0.00	0.45	1239.01
			Z_D	3	536.02	0.00	0.12	
			Z_R	3	536.02	0.00	0.12	
R201.25	20	0.6	Z_A	3	865.45	0.00	0.80	1311.33
			Z_D	2	554.55	-0.33	0.16	
			Z_R	2	554.55	-0.33	0.16	
R201.25	20	0.8	Z_A	3	1089.65	0.00	1.27	1373.99
			Z_D	3	684.25	0.00	0.43	
			Z_R	4	768.58	0.33	0.60	
R201.25	40	0.2	Z_A	3	616.58	0.00	0.29	2587.53
			Z_D	3	552.54	0.00	0.15	
			Z_R	3	545.84	0.00	0.14	
R201.25	40	0.4	Z_A	3	668.15	0.00	0.39	2534.14
			Z_D	3	567.20	0.00	0.18	
			Z_R	3	567.20	0.00	0.18	

ตารางผนวกที่ ก2 (ต่อ)

Problems	$\|S\|$	β	Criterion	Number of Vehicles	Travel Times	R_NV	R_TT	CPU(s)
R201.25	40	0.6	Z_A	3	788.58	0.00	0.64	2606.34
			Z_D	3	584.25	0.00	0.22	
			Z_R	3	584.25	0.00	0.22	
R201.25	40	0.8	Z_A	3	1200.35	0.00	1.50	2614.09
			Z_D	3	627.49	0.00	0.31	
			Z_R	4	739.35	0.33	0.54	
R201.25	80	0.2	Z_A	3	603.41	0.00	0.26	4613.20
			Z_D	3	535.16	0.00	0.12	
			Z_R	3	535.16	0.00	0.12	
R201.25	80	0.4	Z_A	3	641.38	0.00	0.34	5239.99
			Z_D	3	539.77	0.00	0.13	
			Z_R	3	539.77	0.00	0.13	
R201.25	80	0.6	Z_A	3	876.26	0.00	0.83	5269.93
			Z_D	3	540.59	0.00	0.13	
			Z_R	3	540.59	0.00	0.13	

ตารางผนวกที่ ก2 (ต่อ)

Problems	$\lvert S \rvert$	β	Criterion	Number of Vehicles	Travel Times	R_NV	R_TT	CPU(s)
R201.25	80	0.8	Z_A	3	953.32	0.00	0.99	5431.77
			Z_D	3	540.41	0.00	0.13	
			Z_R	3	540.41	0.00	0.13	

หมายเหตุ คำอธิบายตาราง

Problems = ตัวอย่างปัญหา RVRPTW, $\lvert S \rvert$ = ขนาดของสถานการณ์, β = เปอร์เซ็นต์ความไม่แน่นอน, Criterion = เกณฑ์การตัดสินใจ

Number of Vehicles = จำนวนรถขนส่งที่ได้จากผลเฉลยเชิงทนทาน, Travel Times = เวลาเดินทางรวมที่ได้จากผลเฉลยเชิงทนทาน

R_NV = สัดส่วนจำนวนรถขนส่งเมื่อเทียบกับผลเฉลยเชิงกำหนด (Based-Case Number of Vehicles = 3)

R_TT = สัดส่วนเวลาเดินทางโดยรวมเมื่อเทียบกับผลเฉลยเชิงกำหนด (Based-Case Travel Times = 479.56)

CPU(s) = เวลาประมวลผลด้วยเครื่องคอมพิวเตอร์ (วินาที)

ตารางผนวกที่ ก3 ผลเฉลยเชิงทนทานสำหรับปัญหา C101

Problems	$\|S\|$	β	Criterion	Number of Vehicles	Travel Times	R_NV	R_TT	CPU(s)
C101.25	10	0.2	Z_A	4	406.98	0.33	0.24	1976.92
			Z_D	3	367.34	0.00	0.12	
			Z_R	3	367.34	0.00	0.12	
C101.25	10	0.4	Z_A	5	462.58	0.67	0.41	1895.83
			Z_D	3	360.37	0.00	0.09	
			Z_R	3	360.37	0.00	0.09	
C101.25	10	0.6	Z_A	5	521.83	0.67	0.59	1739.66
			Z_D	4	389.26	0.33	0.18	
			Z_R	4	389.26	0.33	0.18	
C101.25	10	0.8	Z_A	6	798.76	1.00	1.43	1717.29
			Z_D	5	530.26	0.67	0.61	
			Z_R	5	530.26	0.67	0.61	
C101.25	20	0.2	Z_A	5	454.41	0.67	0.38	6774.08
			Z_D	3	325.20	0.00	-0.01	
			Z_R	3	325.20	0.00	-0.01	

ตารางผนวกที่ ก3 (ต่อ)

Problems	$\|S\|$	β	Criterion	Number of Vehicles	Travel Times	R_NV	R_TT	CPU(s)
C101.25	20	0.4	Z_A	5	488.21	0.67	0.48	6403.94
			Z_D	3	328.51	0.00	0.00	
			Z_R	3	328.51	0.00	0.00	
C101.25	20	0.6	Z_A	5	517.45	0.67	0.57	3398.96
			Z_D	4	377.62	0.33	0.15	
			Z_R	4	377.62	0.33	0.15	
C101.25	20	0.8	Z_A	6	721.94	1.00	1.19	3400.74
			Z_D	4	523.09	0.33	0.59	
			Z_R	4	523.09	0.33	0.59	
C101.25	40	0.2	Z_A	4	442.52	0.33	0.34	8027.78
			Z_D	3	348.01	0.00	0.06	
			Z_R	3	348.01	0.00	0.06	
C101.25	40	0.4	Z_A	5	447.51	0.67	0.36	12054.31
			Z_D	3	363.51	0.00	0.10	
			Z_R	3	363.51	0.00	0.10	

ตารางผนวกที่ ก3 (ต่อ)

Problems	$\|S\|$	β	Criterion	Number of Vehicles	Travel Times	R_NV	R_TT	CPU(s)
C101.25	40	0.6	Z_A	5	564.59	0.67	0.72	7283.61
			Z_D	3	383.84	0.00	0.17	
			Z_R	3	383.84	0.00	0.17	
C101.25	40	0.8	Z_A	5	821.41	0.67	1.50	6597.21
			Z_D	4	444.63	0.33	0.35	
			Z_R	4	444.63	0.33	0.35	
C101.25	80	0.2	Z_A	4	435.60	0.33	0.32	29075.40
			Z_D	3	354.67	0.00	0.08	
			Z_R	3	354.67	0.00	0.08	
C101.25	80	0.4	Z_A	5	523.98	0.67	0.59	15192.85
			Z_D	3	384.32	0.00	0.17	
			Z_R	3	384.32	0.00	0.17	
C101.25	80	0.6	Z_A	5	529.28	0.67	0.61	14212.15
			Z_D	4	437.53	0.33	0.33	
			Z_R	3	401.27	0.00	0.22	

ตารางผนวกที่ ก3 (ต่อ)

Problems	$\|S\|$	β	Criterion	Number of Vehicles	Travel Times	R_NV	R_TT	CPU(s)
C101.25	80	0.8	Z_A	5	560.52	0.67	0.70	13202.93
			Z_D	4	442.86	0.33	0.35	
			Z_R	4	442.86	0.33	0.35	

หมายเหตุ คำอธิบายตาราง

Problems = ตัวอย่างปัญหา RVRPTW, $|S|$ = ขนาดของสถานการณ์, β = เปอร์เซ็นต์ความไม่แน่นอน, Criterion = เกณฑ์การตัดสินใจ

Number of Vehicles = จำนวนรถขนส่งที่ได้จากผลเฉลยเชิงทนทาน, Travel Times = เวลาเดินทางรวมที่ได้จากผลเฉลยเชิงทนทาน

R_NV = สัดส่วนจำนวนรถขนส่งเมื่อเทียบกับผลเฉลยเชิงกำหนด (Based-Case Number of Vehicles = 3)

R_TT = สัดส่วนเวลาเดินทางโดยรวมเมื่อเทียบกับผลเฉลยเชิงกำหนด (Based-Case Travel Times = 329.15)

CPU(s) = เวลาประมวลผลด้วยเครื่องคอมพิวเตอร์ (วินาที)

ตารางผนวกที่ ก4 ผลเฉลยเชิงทนทานสำหรับปัญหา C201

Problems	$\|S\|$	β	Criterion	Number of Vehicles	Travel Times	R_NV	R_TT	CPU(s)
C201.25	10	0.2	Z_A	2	257.12	0.00	0.02	156.03
			Z_D	2	257.12	0.00	0.02	
			Z_R	2	257.12	0.00	0.02	
C201.25	10	0.4	Z_A	2	241.50	0.00	-0.04	157.42
			Z_D	2	241.50	0.00	-0.04	
			Z_R	2	241.50	0.00	-0.04	
C201.25	10	0.6	Z_A	2	328.55	0.00	0.31	161.94
			Z_D	2	233.97	0.00	-0.07	
			Z_R	2	233.97	0.00	-0.07	
C201.25	10	0.8	Z_A	2	308.11	0.00	0.23	168.68
			Z_D	2	221.64	0.00	-0.12	
			Z_R	2	221.64	0.00	-0.12	
C201.25	20	0.2	Z_A	2	279.46	0.00	0.11	331.69
			Z_D	2	279.46	0.00	0.11	
			Z_R	2	279.46	0.00	0.11	

ตารางผนวกที่ ก4 (ต่อ)

Problems	$\|S\|$	β	Criterion	Number of Vehicles	Travel Times	R_NV	R_TT	CPU(s)
C201.25	20	0.4	Z_A	2	294.01	0.00	0.17	318.85
			Z_D	2	294.01	0.00	0.17	
			Z_R	2	294.01	0.00	0.17	
C201.25	20	0.6	Z_A	2	386.22	0.00	0.54	325.68
			Z_D	2	301.76	0.00	0.20	
			Z_R	2	301.76	0.00	0.20	
C201.25	20	0.8	Z_A	2	320.52	0.00	0.28	337.62
			Z_D	2	311.77	0.00	0.24	
			Z_R	2	311.77	0.00	0.24	
C201.25	40	0.2	Z_A	2	264.15	0.00	0.05	626.58
			Z_D	2	264.15	0.00	0.05	
			Z_R	2	264.15	0.00	0.05	
C201.25	40	0.4	Z_A	2	257.55	0.00	0.02	636.96
			Z_D	2	257.55	0.00	0.02	
			Z_R	2	257.55	0.00	0.02	

ตารางผนวกที่ ก4 (ต่อ)

Problems	$\|S\|$	β	Criterion	Number of Vehicles	Travel Times	R_NV	R_TT	CPU(s)
C201.25	40	0.6	Z_A	2	345.88	0.00	0.38	650.75
			Z_D	2	250.16	0.00	0.00	
			Z_R	2	250.16	0.00	0.00	
C201.25	40	0.8	Z_A	2	279.46	0.00	0.11	614.05
			Z_D	2	279.46	0.00	0.11	
			Z_R	2	279.46	0.00	0.11	
C201.25	80	0.2	Z_A	2	262.79	0.00	0.05	1257.91
			Z_D	2	262.79	0.00	0.05	
			Z_R	2	262.79	0.00	0.05	
C201.25	80	0.4	Z_A	2	257.49	0.00	0.02	1281.45
			Z_D	2	257.49	0.00	0.02	
			Z_R	2	257.49	0.00	0.02	
C201.25	80	0.6	Z_A	2	244.07	0.00	-0.03	1300.38
			Z_D	2	244.07	0.00	-0.03	
			Z_R	2	244.07	0.00	-0.03	

ตารางผนวกที่ ก4 (ต่อ)

Problems	$\|S\|$	β	Criterion	Number of Vehicles	Travel Times	R_NV	R_TT	CPU(s)
C201.25	80	0.8	Z_A	2	267.80	0.00	0.07	1369.50
			Z_D	2	240.73	0.00	-0.04	
			Z_R	2	240.73	0.00	-0.04	

หมายเหตุ คำอธิบายตาราง

Problems = ตัวอย่างปัญหา RVRPTW, $|S|$ = ขนาดของสถานการณ์, β = เปอร์เซ็นต์ความไม่แน่นอน, Criterion = เกณฑ์การตัดสินใจ

Number of Vehicles = จำนวนรถขนส่งที่ได้จากผลเฉลยเชิงทนทาน, Travel Times = เวลาเดินทางรวมที่ได้จากผลเฉลยเชิงทนทาน

R_NV = สัดส่วนจำนวนรถขนส่งเมื่อเทียบกับผลเฉลยเชิงกำหนด (Based-Case Number of Vehicles = 2)

R_TT = สัดส่วนเวลาเดินทางโดยรวมเมื่อเทียบกับผลเฉลยเชิงกำหนด (Based-Case Travel Times = 251.38)

CPU(s) = เวลาประมวลผลด้วยเครื่องคอมพิวเตอร์ (วินาที)

ตารางผนวกที่ ก5 ผลเฉลยเชิงทนทานสำหรับปัญหา RC101

Problems	$\|S\|$	β	Criterion	Number of Vehicles	Travel Times	R_NV	R_TT	CPU(s)
RC101.25	10	0.2	Z_A	5	510.20	0.67	0.41	1395.81
			Z_D	4	454.99	0.33	0.26	
			Z_R	4	454.99	0.33	0.26	
RC101.25	10	0.4	Z_A	4	518.42	0.33	0.44	1222.20
			Z_D	3	403.57	0.00	0.12	
			Z_R	3	403.57	0.00	0.12	
RC101.25	10	0.6	Z_A	4	830.20	0.33	1.30	1246.50
			Z_D	3	415.44	0.00	0.15	
			Z_R	3	415.44	0.00	0.15	
RC101.25	10	0.8	Z_A	4	1395.09	0.33	2.87	1121.07
			Z_D	4	498.40	0.33	0.38	
			Z_R	4	498.40	0.33	0.38	
RC101.25	20	0.2	Z_A	5	519.09	0.67	0.44	2681.73
			Z_D	4	443.62	0.33	0.23	
			Z_R	4	443.62	0.33	0.23	

ตารางผนวกที่ ก5 (ต่อ)

Problems	$\|S\|$	β	Criterion	Number of Vehicles	Travel Times	R_NV	R_TT	CPU(s)
RC101.25	20	0.4	Z_A	4	492.06	0.33	0.36	2418.65
			Z_D	3	429.35	0.00	0.19	
			Z_R	3	429.35	0.00	0.19	
RC101.25	20	0.6	Z_A	4	713.82	0.33	0.98	2591.75
			Z_D	3	351.15	0.00	-0.03	
			Z_R	3	351.15	0.00	-0.03	
RC101.25	20	0.8	Z_A	5	662.66	0.67	0.84	2343.27
			Z_D	4	532.73	0.33	0.48	
			Z_R	4	618.34	0.33	0.71	
RC101.25	40	0.2	Z_A	4	460.12	0.33	0.28	5188.18
			Z_D	3	377.45	0.00	0.05	
			Z_R	3	377.45	0.00	0.05	
RC101.25	40	0.4	Z_A	4	481.30	0.33	0.33	4696.53
			Z_D	3	396.73	0.00	0.10	
			Z_R	3	396.73	0.00	0.10	

ตารางผนวกที่ ก5 (ต่อ)

Problems	$\|S\|$	β	Criterion	Number of Vehicles	Travel Times	R_NV	R_TT	CPU(s)
RC101.25	40	0.6	Z_A	4	522.01	0.33	0.45	4973.02
			Z_D	3	451.93	0.00	0.25	
			Z_R	3	430.11	0.00	0.19	
RC101.25	40	0.8	Z_A	3	1049.21	0.00	1.91	4687.92
			Z_D	4	518.49	0.33	0.44	
			Z_R	4	518.49	0.33	0.44	
RC101.25	80	0.2	Z_A	4	445.44	0.33	0.23	10268.96
			Z_D	3	377.50	0.00	0.05	
			Z_R	3	377.50	0.00	0.05	
RC101.25	80	0.4	Z_A	4	503.78	0.33	0.40	9873.78
			Z_D	3	378.75	0.00	0.05	
			Z_R	3	378.75	0.00	0.05	
RC101.25	80	0.6	Z_A	5	489.25	0.67	0.36	9881.94
			Z_D	3	422.49	0.00	0.17	
			Z_R	3	422.49	0.00	0.17	

ตารางผนวกที่ ก5 (ต่อ)

Problems	$\|S\|$	β	Criterion	Number of Vehicles	Travel Times	R_NV	R_TT	CPU(s)
RC101.25	80	0.8	Z_A	3	1020.85	0.00	1.83	9550.67
			Z_D	4	478.19	0.33	0.33	
			Z_R	4	478.19	0.33	0.33	

หมายเหตุ คำอธิบายตาราง

Problems = ตัวอย่างปัญหา RVRPTW, $|S|$ = ขนาดของสถานการณ์, β = เปอร์เซ็นต์ความไม่แน่นอน, Criterion = เกณฑ์การตัดสินใจ

Number of Vehicles = จำนวนรถขนส่งที่ได้จากผลเฉลยเชิงทนทาน, Travel Times = เวลาเดินทางรวมที่ได้จากผลเฉลยเชิงทนทาน

R_NV = สัดส่วนจำนวนรถขนส่งเมื่อเทียบกับผลเฉลยเชิงกำหนด (Based-Case Number of Vehicles = 3)

R_TT = สัดส่วนเวลาเดินทางโดยรวมเมื่อเทียบกับผลเฉลยเชิงกำหนด (Based-Case Travel Times = 360.68)

CPU(s) = เวลาประมวลผลด้วยเครื่องคอมพิวเตอร์ (วินาที)

ตารางผนวกที่ ก6 ผลเฉลยเชิงทนทานสำหรับปัญหา RC201

Problems	$\|S\|$	β	Criterion	Number of Vehicles	Travel Times	R_NV	R_TT	CPU(s)
RC201.25	10	0.2	Z_A	3	602.67	0.00	0.58	486.91
			Z_D	3	423.09	0.00	0.11	
			Z_R	3	423.09	0.00	0.11	
RC201.25	10	0.4	Z_A	3	754.12	0.00	0.98	551.91
			Z_D	3	485.20	0.00	0.28	
			Z_R	3	485.20	0.00	0.28	
RC201.25	10	0.6	Z_A	3	754.49	0.00	0.98	515.50
			Z_D	3	396.08	0.00	0.04	
			Z_R	3	396.08	0.00	0.04	
RC201.25	10	0.8	Z_A	2	1300.36	-0.33	2.42	591.31
			Z_D	3	467.68	0.00	0.23	
			Z_R	3	467.68	0.00	0.23	
RC201.25	20	0.2	Z_A	3	627.48	0.00	0.65	996.91
			Z_D	3	431.29	0.00	0.13	
			Z_R	3	431.29	0.00	0.13	

ตารางผนวกที่ ก6 (ต่อ)

Problems	$\|S\|$	β	Criterion	Number of Vehicles	Travel Times	R_NV	R_TT	CPU(s)
RC201.25	20	0.4	Z_A	2	611.75	-0.33	0.61	1028.26
			Z_D	3	471.99	0.00	0.24	
			Z_R	3	471.99	0.00	0.24	
RC201.25	20	0.6	Z_A	3	896.78	0.00	1.36	916.04
			Z_D	3	538.59	0.00	0.42	
			Z_R	3	538.59	0.00	0.42	
RC201.25	20	0.8	Z_A	2	1113.84	-0.33	1.93	1002.49
			Z_D	3	507.25	0.00	0.33	
			Z_R	3	507.25	0.00	0.33	
RC201.25	40	0.2	Z_A	3	473.56	0.00	0.24	2003.19
			Z_D	3	417.45	0.00	0.10	
			Z_R	3	417.45	0.00	0.10	
RC201.25	40	0.4	Z_A	3	682.96	0.00	0.80	2065.49
			Z_D	3	458.01	0.00	0.20	
			Z_R	3	458.01	0.00	0.20	

ตารางผนวกที่ ก6 (ต่อ)

Problems	$\|S\|$	β	Criterion	Number of Vehicles	Travel Times	R_NV	R_TT	CPU(s)
RC201.25	40	0.6	Z_A	3	609.14	0.00	0.60	1964.31
			Z_D	3	512.45	0.00	0.35	
			Z_R	3	512.45	0.00	0.35	
RC201.25	40	0.8	Z_A	4	735.19	0.33	0.93	2309.03
			Z_D	3	543.86	0.00	0.43	
			Z_R	2	748.31	-0.33	0.97	
RC201.25	80	0.2	Z_A	3	550.92	0.00	0.45	4186.57
			Z_D	3	421.19	0.00	0.11	
			Z_R	3	421.19	0.00	0.11	
RC201.25	80	0.4	Z_A	3	661.76	0.00	0.74	4158.84
			Z_D	3	407.20	0.00	0.07	
			Z_R	3	407.20	0.00	0.07	
RC201.25	80	0.6	Z_A	4	669.41	0.33	0.76	4055.30
			Z_D	3	446.30	0.00	0.17	
			Z_R	3	446.30	0.00	0.17	

ตารางผนวกที่ ก6 (ต่อ)

Problems	$\|S\|$	β	Criterion	Number of Vehicles	Travel Times	R_NV	R_TT	CPU(s)
RC201.25	80	0.8	Z_A	4	701.22	0.33	0.84	4148.35
			Z_D	3	476.53	0.00	0.25	
			Z_R	3	476.53	0.00	0.25	

หมายเหตุ คำอธิบายตาราง

Problems = ตัวอย่างปัญหา RVRPTW, $|S|$ = ขนาดของสถานการณ์, β = เปอร์เซ็นต์ความไม่แน่นอน, Criterion = เกณฑ์การตัดสินใจ

Number of Vehicles = จำนวนรถขนส่งที่ได้จากผลเฉลยเชิงทนทาน, Travel Times = เวลาเดินทางรวมที่ได้จากผลเฉลยเชิงทนทาน

R_NV = สัดส่วนจำนวนรถขนส่งเมื่อเทียบกับผลเฉลยเชิงกำหนด (Based-Case Number of Vehicles = 3)

R_TT = สัดส่วนเวลาเดินทางโดยรวมเมื่อเทียบกับผลเฉลยเชิงกำหนด (Based-Case Travel Times = 380.46)

CPU(s) = เวลาประมวลผลด้วยเครื่องคอมพิวเตอร์ (วินาที)

ภาคผนวก ข

ผลการเปรียบเทียบขั้นตอนวิธีการ

ตารางผนวกที่ ข1 ผลการเปรียบเทียบขั้นตอนวิธีการสำหรับปัญหา R1 และ R2

Problems	Best Solution				Hybrid Algorithm (HY)			Genetic Algorithm (GA)			Computer
	NV	Travel Times	Authors	Capacity	NV	Travel Times	CPU(s)	NV	Travel Times	CPU(s)	
R101.25	8	617.1	KDMSS	200	7	613.2	123	14	1203.0	355	C1
R101.50	12	1044	KDMSS	200	12	1155.0	2508	26	3438.0	649	C1
R101.100	20	1637.7	KDMSS	200	17	1915.5	43079	51	10150	1300	C1
R102.25	7	547.1	KDMSS	200	5	494.7	86	11	1163.5	272	C1
R102.50	11	909	KDMSS	200	9	932.9	1349	22	2969	550	C1
R102.100	18	1466.6	KDMSS	200	17	1694.3	23716	42	8486.1	1074	C1
R103.25	5	454.6	KDMSS	200	5	498.9	96	9	1060.4	225	C1
R103.50	9	772.9	KDMSS	200	8	823.3	1384	18	2672	451	C1
R103.100	14	1208.7	CR+L	200	15	1467.4	21222	35	6756.6	920	C1
R104.25	4	416.9	KDMSS	200	4	417.6	90	7	911.7	278	C2
R104.50	6	625.4	KDMSS	200	7	698.2	1406	12	2189.2	469	C2
R104.100	11	971.5	IV	200	13	1199.3	21948	25	6542	1065	C2
R105.25	6	530.5	KDMSS	200	6	553.5	124	10	1073.5	415	C2
R105.50	9	899.3	KDMSS	200	10	957	2053	23	3042.6	884	C2
R105.100	15	1355.3	KDMSS	200	18	1607.9	31908	43	7594.5	2101	C2

ตารางผนวกที่ ข1 (ต่อ)

Problems	Best Solution				Hybrid Algorithm (HY)			Genetic Algorithm (GA)			Computer
	NV	Travel Times	Authors	Capacity	NV	Travel Times	CPU(s)	NV	Travel Times	CPU(s)	
R106.25	3	465.4	KDMSS	200	5	494.7	127	10	1067.5	430	C3
R106.50	5	793	KDMSS	200	8	872	1624	21	2744.5	899	C3
R106.100	13	1234.6	CR+KLM	200	15	1452	22965	39	7373.3	1787	C3
R107.25	4	424.3	KDMSS	200	5	487.2	100	9	988.5	374	C3
R107.50	7	711.1	KDMSS	200	8	793.5	1260	17	2703	707	C3
R107.100	11	1064.6	CR+KLM	200	14	1298.4	21865	33	5989.8	1426	C3
R108.25	4	397.3	KDMSS	200	4	427.8	148	7	894.6	308	C4
R108.50	6	617.7	CR+KLM	200	7	679.1	2358	11	2426.4	490	C4
R108.100	N/A	N/A	N/A	200	11	1054.3	27517	25	5425.4	1147	C4
R109.25	5	441.3	KDMSS	200	5	486.8	222	9	1049.7	396	C4
R109.50	8	786.8	KDMSS	200	8	867.6	2293	17	2742.8	745	C4
R109.100	13	1146.9	CR+KLM	200	15	1341.4	41049	32	6407.4	1468	C4
R110.25	4	444.1	KDMSS	200	4	433.5	243	8	973.2	357	C4
R110.50	7	697	KDMSS	200	8	789	2097	13	2815.9	562	C4
R110.100	12	1068	CR+KLM	200	15	1305.1	35938	28	6449.4	1243	C4

ตารางผนวกที่ ข1 (ต่อ)

Problems	Best Solution				Hybrid Algorithm (HY)			Genetic Algorithm (GA)			Computer
	NV	Travel Times	Authors	Capacity	NV	Travel Times	CPU(s)	NV	Travel Times	CPU(s)	
R111.25	5	428.8	KDMSS	200	4	471.3	47	8	997.3	168	C5
R111.50	7	707.2	CR+KLM	200	8	755.1	812	16	2474.6	338	C5
R111.100	12	1048.7	CR+KLM	200	13	1217.6	1289	32	6063.8	671.8	C5
R112.25	4	393	KDMSS	200	4	406.4	76	6	882.9	229	C6
R112.50	6	630.2	CR+KLM	200	7	767.1	1138	12	2186	472	C6
R112.100	N/A	N/A	N/A	200	12	1143.9	21033	25	5106.4	991	C6
R201.25	4	463.3	CR+KLM	1000	3	577.1	39	3	1594.1	85	C1
R201.50	6	791.9	CR+KLM	1000	7	1164.3	865	11	3894	285	C1
R201.100	8	1143.2	KLM	1000	11	1967.6	11186	22	1023	594	C1
R202.25	4	410.5	CR+KLM	1000	4	535.9	45	4	1258.9	106	C1
R202.50	5	698.5	CR+KLM	1000	7	1083.3	771	9	3365.6	241	C1
R202.100	N/A	N/A	N/A	1000	8	1683.1	9427	19	8802.1	511	C1
R203.25	3	391.4	CR+KLM	1000	2	468.3	40	3	1128.8	85	C1
R203.50	5	605.3	IV+C	1000	4	822.2	602	6	3005.4	165	C1
R203.100	N/A	N/A	N/A	1000	8	1408.4	10167	13	7137	361	C1

ตารางผนวกที่ ข1 (ต่อ)

Problems	Best Solution				Hybrid Algorithm (HY)			Genetic Algorithm (GA)			Computer
	NV	Travel Times	Authors	Capacity	NV	Travel Times	CPU(s)	NV	Travel Times	CPU(s)	
R204.25	2	355.0	IV+C	1000	2	442.5	30	2	960.6	94	C2
R204.50	2	506.4	IV	1000	2	619.1	489	4	2250.2	178	C2
R204.100	N/A	N/A	N/A	1000	6	1080.4	11997	7	7965.8	321	C2
R205.25	3	393	CR+KLM	1000	N/A	N/A	N/A	2	1570.4	94	C2
R205.50	4	690.1	IV+C	1000	5	896.1	658	6	3666.8	251	C2
R205.100	N/A	N/A	N/A	1000	8	1468.2	17797	13	7856.8	557	C2
R206.25	3	374.4	CR+KLM	1000	3	514.1	43	3	978.1	173	C3
R206.50	4	632.4	IV+C	1000	4	896.7	543	6	3031.8	303	C3
R206.100	N/A	N/A	N/A	1000	8	1369.9	13054	12	7616.5	570	C3
R207.25	3	361.6	KLM	1000	2	457	31	2	1080.3	95	C3
R207.50	N/A	N/A	N/A	1000	2	696.4	352	4	2442.4	189	C3
R207.100	N/A	N/A	N/A	1000	3	1025.8	8900	9	6393.8	428	C3
R208.25	1	328.2	IV+C	1000	N/A	N/A	N/A	2	786	104	C4
R208.50	N/A	N/A	N/A	1000	3	644.2	475	3	2324	159	C4
R208.100	N/A	N/A	N/A	1000	3	862.6	6951	6	5608.4	323	C4

ตารางผนวกที่ ข1 (ต่อ)

Problems	Best Solution				Hybrid Algorithm (HY)			Genetic Algorithm (GA)			Computer
	NV	Travel Times	Authors	Capacity	NV	Travel Times	CPU(s)	NV	Travel Times	CPU(s)	
R209.25	2	370.7	KLM	1000	2	452.7	44	3	989	141	C4
R209.50	4	600.6	IV+C	1000	4	751.2	968	6	2678.8	289	C4
R209.100	N/A	N/A	N/A	1000	8	1359.3	12834	12	6466.2	577	C4
R210.25	3	404.6	CR+KLM	1000	2	513.1	25	3	865.8	71	C5
R210.50	4	645.6	IV+C	1000	3	767.7	351	7	2377	161	C5
R210.100	N/A	N/A	N/A	1000	6	1283.6	6242	13	7325	296	C5
R211.25	2	350.9	KLM	1000	2	424	40	2	939.9	93	C6
R211.50	3	535.5	IV+DLP	1000	3	636.9	662	4	2782.7	183	C6
R211.100	N/A	N/A	N/A	1000	6	1028.5	11721	8	6113.3	360	C6

หมายเหตุ คำอธิบายตาราง

Problems = ตัวอย่างปัญหา VRPTW ของ Solomon, Best Solution = ผลเฉลยที่ดีที่สุด, NV = จำนวนรถขนส่ง,

Travel Times = เวลาเดินทางโดยรวม, CPU(s) = เวลาประมวลผลด้วยเครื่องคอมพิวเตอร์, Capacity = ขีดความสามารถในการบรรทุกสินค้า

Computer = รหัสเครื่องคอมพิวเตอร์ที่ใช้ในการประมวลผล, N/A = ไม่ปรากฏข้อมูล

Authors = ชื่อย่อนักวิจัยผู้ค้นหาผลเฉลยเหมาะที่สุดสำหรับปัญหาเทียบเคียง ประกอบด้วย

C - A. Chabrier, "Vehicle Routing Problem with Elementary Shortest Path based Column Generation." Forthcoming in: Computers and Operations Research (2005).

CR - W. Cook and J. L. Rich, "A parallel cutting plane algorithm for the vehicle routing problem with time windows," Working Paper, Computational and Applied Mathematics, Rice University, Houston, TX, 1999.

DLP - E. Danna and C. Le Pape, "Accelerating branch-and-price with local search: A case study on the vehicle routing problem with time windows," In: Column Generation, G. Desaulniers, J. Desrosiers, and M. M. Solomon (eds.), 99-130, Kluwer Academic Publishers (2005).

IV - S. Irnich and D. Villeneuve, "The shortest path problem with k-cycle elimination ($k \geq 3$): Improving a branch-and-price algorithm for the VRPTW." Forthcoming in: INFORMS Journal of Computing (2005).

KDMSS - N. Kohl, J. Desrosiers, O. B. G. Madsen, M. M. Solomon, and F. Soumis, "2-Path Cuts for the Vehicle Routing Problem with Time Windows,"Transportation Science, Vol. 33 (1), 101-116 (1999).

KLM - B. Kallehauge, J. Larsen, and O.B.G. Madsen. "Lagrangean duality and non-differentiable optimization applied on routing with time windows – experimental results." Internal report IMM-REP-2000-8, Department of Mathematical Modelling, Technical University of Denmark, Lyngby, Denmark, 2000.

L - J. Larsen. "Parallelization of the vehicle routing problem with time windows." Ph.D. Thesis IMM-PHD-1999-62, Department of Mathematical Modelling, Technical University of Denmark, Lyngby, Denmark, 1999.

ตารางผนวกที่ ข2 ผลการเปรียบเทียบขั้นตอนวิธีการสำหรับปัญหา C1 และ C2

Problems	Best Solution				Hybrid Algorithm (HY)			Genetic Algorithm (GA)			Computer
	NV	Travel Times	Authors	Capacity	NV	Travel Times	CPU(s)	NV	Travel Times	CPU(s)	
C101.25	3	191.3	KDMSS	200	3	330.7	209	18	1969.8	641	C1
C101.50	5	362.4	KDMSS	200	5	882	3223	34	8967.6	1292	C1
C101.100	10	827.3	KDMSS	200	12	3108.8	51477	100	37188	2489	C1
C102.25	3	190.3	KDMSS	200	4	366.5	157	14	1459.4	467	C1
C102.50	5	361.4	KDMSS	200	7	912	2226	38	5888.5	938	C1
C102.100	10	827.3	KDMSS	200	13	2676.9	35189	75	24674	1892	C1
C103.25	3	190.3	KDMSS	200	3	316.4	90	11	1098.6	316	C1
C103.50	5	361.4	KDMSS	200	6	687.7	1296	26	4014.2	655	C1
C103.100	10	826.3	KDMSS	200	13	2176.5	23140	51	16638	1269	C1
C104.25	3	186.9	KDMSS	200	3	266	60	9	844.5	426	C2
C104.50	5	358	KDMSS	200	6	636.5	1015	11	2131.8	440	C2
C104.100	10	822.9	KDMSS	200	12	1536.9	23643	26	9454.3	1114	C2
C105.25	3	191.3	KDMSS	200	3	254	122	11	1038.7	452	C3
C105.50	5	362.4	KDMSS	200	7	547	24652	25	5376.7	1030	C3
C105.100	10	827.3	KDMSS	200	17	1417.4	34810	50	2695.9	2104	C3

ตารางผนวกที่ ข2 (ต่อ)

Problems	Best Solution				Hybrid Algorithm (HY)			Genetic Algorithm (GA)			Computer
	NV	Travel Times	Authors	Capacity	NV	Travel Times	CPU(s)	NV	Travel Times	CPU(s)	
C106.25	3	191.3	KDMSS	200	4	364.1	924.4	17	1932.5	893	C4
C106.50	5	362.4	KDMSS	200	5	750.6	9629	34	7504.5	1456	C4
C106.100	10	827.3	KDMSS	200	13	1793.1	27552	46	24582	2048	C4
C107.25	3	191.3	KDMSS	200	4	263.7	295	9	973.6	395	C4
C107.50	5	362.4	KDMSS	200	5	371	4022	21	4348	896	C4
C107.100	10	827.3	KDMSS	200	11	970.4	40316	49	15902	2141	C4
C108.25	3	191.3	KDMSS	200	3	199.2	31	6	905	144	C5
C108.50	5	362.4	KDMSS	200	6	489.6	555	15	3447.4	322	C5
C108.100	10	827.3	KDMSS	200	12	1005.8	13174	30	17627.2	635	C5
C109.25	3	191.3	KDMSS	200	3	244.2	44	3	546	127	C6
C109.50	5	362.4	KDMSS	200	5	457	402	8	2966.6	327	C6
C109.100	10	827.3	KDMSS	200	12	1115.3	16049	20	11435.1	811	C6
C201.25	2	214.7	CR+L	700	2	286.4	12	2	8014.2	62	C1
C201.50	3	360.2	CR+L	700	4	576.2	690	18	10797	487	C1
C201.100	3	589.1	CR+KLM	700	10	1931.1	18922	42	50358	1177	C1

ตารางผนวกที่ ข2 (ต่อ)

Problems	Best Solution				Hybrid Algorithm (HY)			Genetic Algorithm (GA)			Computer
	NV	Travel Times	Authors	Capacity	NV	Travel Times	CPU(s)	NV	Travel Times	CPU(s)	
C202.25	2	214.7	CR+L	700	2	260.9	12	2	4695.8	50	C5
C202.50	3	360.2	CR+KLM	700	5	846.8	606	13	9663.3	280	C5
C202.100	3	589.1	CR+KLM	700	8	1512.3	14254	34	40823.9	725	C5
C203.25	2	214.7	CR+L	700	2	316.8	38	2	2010.8	93	C2
C203.50	3	359.8	CR+KLM	700	4	721.9	816	7	7790	291	C2
C203.100	3	588.7	KLM	700	6	1199.2	18601	20	27702.6	962	C2
C204.25	2	213.1	CR+KLM	700	2	314.2	25	2	936.6	78	C3
C204.50	2	350.1	KLM	700	2	503.7	491	3	3027.7	119	C3
C204.100	3	588.1	IV	700	6	1190.4	10678	11	16971	506	C3
C205.25	2	214.7	CR+L	700	2	262.1	21	2	5573.5	105	C4
C205.50	3	359.8	CR+KLM	700	2	493.8	153	2	44314	117	C4
C205.100	3	586.4	CR+KLM	700	6	1019.7	27468	29	35711	1292	C4
C206.25	2	214.7	CR+L	700	2	259.6	11	2	3809.3	49	C5
C206.50	3	359.8	CR+KLM	700	2	574.4	103	2	41342.8	57	C5
C206.100	3	586	CR+KLM	700	7	1469.4	9489	19	21487.6	421	C5

ตารางผนวกที่ ข2 (ต่อ)

Problems	Best Solution				Hybrid Algorithm (HY)			Genetic Algorithm (GA)			Computer
	NV	Travel Times	Authors	Capacity	NV	Travel Times	CPU(s)	NV	Travel Times	CPU(s)	
C207.25	2	214.5	CR+L	700	2	255	18	3	2918.4	72	C5
C207.50	3	359.6	CR+KLM	700	4	656.6	479	7	8311.3	157	C5
C207.100	3	585.8	CR+KLM	700	7	1273.8	8264	19	25086.8	444	C5
C208.25	2	214.5	CR+L	700	N/A	N/A	N/A	N/A	N/A	N/A	C6
C208.50	2	350.5	CR+KLM	700	2	389	135	2	29812.5	105	C6
C208.100	3	585.8	KLM	700	5	1063.5	4413	9	103078.2	395	C6

หมายเหตุ คำอธิบายตาราง

Problems = ตัวอย่างปัญหา VRPTW ของ Solomon, Best Solution = ผลเฉลยที่ดีที่สุด, NV = จำนวนรถขนส่ง,

Travel Times = เวลาเดินทางโดยรวม, CPU(s) = เวลาประมวลผลด้วยเครื่องคอมพิวเตอร์, Capacity = ขีดความสามารถในการบรรทุกสินค้า

Computer = รหัสเครื่องคอมพิวเตอร์ที่ใช้ในการประมวลผล, N/A = ไม่ปรากฏข้อมูล

Authors = ชื่อย่อนักวิจัยผู้ค้นหาผลเฉลยเหมาะที่สุดสำหรับปัญหาเทียบเคียง ประกอบด้วย

C - A. Chabrier, “Vehicle Routing Problem with Elementary Shortest Path based Column Generation.” Forthcoming in: Computers and Operations Research (2005).

CR - W. Cook and J. L. Rich, “A parallel cutting plane algorithm for the vehicle routing problem with time windows,” Working Paper,

Computational and Applied Mathematics, Rice University, Houston, TX, 1999.

DLP - E. Danna and C. Le Pape, "Accelerating branch-and-price with local search: A case study on the vehicle routing problem with time windows," In: Column Generation, G. Desaulniers, J. Desrosiers, and M. M. Solomon (eds.), 99-130, Kluwer Academic Publishers (2005).

IV - S. Irnich and D. Villeneuve, "The shortest path problem with k-cycle elimination ($k \geq 3$): Improving a branch-and-price algorithm for the VRPTW." Forthcoming in: INFORMS Journal of Computing (2005).

KDMSS - N. Kohl, J. Desrosiers, O. B. G. Madsen, M. M. Solomon, and F. Soumis, "2-Path Cuts for the Vehicle Routing Problem with Time Windows," Transportation Science, Vol. 33 (1), 101-116 (1999).

KLM - B. Kallehauge, J. Larsen, and O.B.G. Madsen. "Lagrangean duality and non-differentiable optimization applied on routing with time windows – experimental results." Internal report IMM-REP-2000-8, Department of Mathematical Modelling, Technical University of Denmark, Lyngby, Denmark, 2000.

L - J. Larsen. "Parallelization of the vehicle routing problem with time windows." Ph.D. Thesis IMM-PHD-1999-62, Department of Mathematical Modelling, Technical University of Denmark, Lyngby, Denmark, 1999.

ตารางผนวกที่ ข3 ผลการเปรียบเทียบขั้นตอนวิธีการสำหรับปัญหา RC1 และ RC2

Problems	Best Solution				Hybrid Algorithm (HY)			Genetic Algorithm (GA)			Computer
	NV	Travel Times	Authors	Capacity	NV	Travel Times	CPU(s)	NV	Travel Times	CPU(s)	
RC101.25	4	461.1	KDMSS	200	4	439.4	65	11	1387.5	268	C1
RC101.50	8	944	KDMSS	200	9	987.7	1315	21	4509.2	487	C1
RC101.100	15	1619.8	KDMSS	200	18	1960.1	21956	42	10380	1069	C1
RC102.25	3	351.8	KDMSS	200	4	407.4	87	11	1397.8	275	C1
RC102.50	7	822.5	KDMSS	200	10	1006.7	1442	22	4659.3	567	C1
RC102.100	14	1457.4	CR+KLM	200	16	1687.2	25346	42	10580	1064	C1
RC103.25	3	332.8	KDMSS	200	4	402.4	102	9	1261	435	C2
RC103.50	6	710.9	KDMSS	200	10	968.7	2387	19	3807.4	884	C2
RC103.100	11	1258	CR+KLM	200	14	1509.4	30065	36	8368.4	1726	C2
RC104.25	3	306.6	KDMSS	200	3	316.6	71	8	1177.9	336	C3
RC104.50	5	545.8	KDMSS	200	7	763.1	716	13	2856.8	541	C3
RC104.100	N/A	N/A	N/A	200	13	1368.1	19027	27	10537	1140	C3
RC105.25	4	411.3	KDMSS	200	4	422.9	104	11	1389.9	522	C4
RC105.50	8	855.3	KDMSS	200	9	976.8	1839	20	4701.6	1036	C4
RC105.100	15	1513.7	KDMSS	200	15	1755.8	24417	42	11222	2031	C4

ตารางผนวกที่ ข3 (ต่อ)

Problems	Best Solution				Hybrid Algorithm (HY)			Genetic Algorithm (GA)			Computer
	NV	Travel Times	Authors	Capacity	NV	Travel Times	CPU(s)	NV	Travel Times	CPU(s)	
RC106.25	3	345.5	KDMSS	200	4	401.9	107	10	1315	449	C4
RC106.50	6	723.2	KDMSS	200	9	918	2082	23	4160.6	1029	C4
RC106.100	N/A	N/A	N/A	200	15	1619.4	30894	39	11239	1884	C4
RC107.25	3	298.3	KDMSS	200	4	369.3	43	9	1197.6	182	C5
RC107.50	6	642.7	KDMSS	200	6	673.9	764	17	4140.3	355	C5
RC107.100	12	1207.8	IV	200	14	1394.2	18864	32	8671.4	695	C5
RC108.25	3	294.5	KDMSS	200	3	396.6	62	8	1056.2	306	C6
RC108.50	6	598.1	KDMSS	200	6	623	1587	16	3602.2	616	C6
RC108.100	11	1114.2	IV	200	13	1384.8	24067	28	9035.6	1115	C6
RC201.25	3	360.2	CR+L	1000	4	501.1	38	5	1881.6	136	C1
RC201.50	5	684.8	L+KLM	1000	7	1252.3	720	14	5328	371	C1
RC201.100	9	1261.8	KLM	1000	10	2117.9	20432	24	13601	690	C1
RC202.25	3	338	CR+KLM	1000	3	473.2	35	4	1519.2	88	C5
RC202.50	5	613.6	IV+C	1000	6	1063.7	574	10	4772.4	217	C5
RC202.100	8	1092.3	IV+C	1000	9	1820.2	11508	20	12105	443	C5

ตารางผนวกที่ ข3 (ต่อ)

Problems	Best Solution				Hybrid Algorithm (HY)			Genetic Algorithm (GA)			Computer
	NV	Travel Times	Authors	Capacity	NV	Travel Times	CPU(s)	NV	Travel Times	CPU(s)	
RC203.25	3	326.9	IV+C	1000	2	462.2	49	3	1116.9	131	C2
RC203.50	4	555.3	IV+C	1000	3	780.3	594	6	5403.1	266	C2
RC203.100	N/A	N/A	N/A	1000	6	1383.7	17555	14	13143	598	C2
RC204.25	3	299.7	C	1000	2	406.5	53	2	1153.6	99	C3
RC204.50	3	444.2	DLP	1000	3	710.2	233	3	7468.5	156	C3
RC204.100	N/A	N/A	N/A	1000	5	1114.4	9206	7	13708	346	C3
RC205.25	3	338.0	L+KLM	1000	3	474	69	5	1464	229	C4
RC205.50	5	630.2	IV+C	1000	8	1145.1	1614	14	4014.5	634	C4
RC205.100	7	1154.0	IV+C	1000	8	1840.2	18231	23	13196	1096	C4
RC206.25	3	324.0	KLM	1000	2	488.8	16	3	1476.6	71	C5
RC206.50	5	610.0	IV+C	1000	5	981.8	422	9	3918.5	198	C5
RC206.100	N/A	N/A	N/A	1000	8	1531.3	6525	15	9331.3	344	C5
RC207.25	3	298.3	KLM	1000	2	403.2	22	3	1241.7	70	C5
RC207.50	4	558.6	C	1000	5	876.5	561	8	3478.4	177	C5
RC207.100	N/A	N/A	N/A	1000	8	1512.5	7158	15	9970.8	336	C5

ตารางผนวกที่ ข3 (ต่อ)

Problems	Best Solution				Hybrid Algorithm (HY)			Genetic Algorithm (GA)			Computer
	NV	Travel Times	Authors	Capacity	NV	Travel Times	CPU(s)	NV	Travel Times	CPU(s)	
RC208.25	2	269.1	C	1000	2	384.1	27	3	943.4	127	C6
RC208.50	N/A	N/A	N/A	1000	3	730.8	526	5	3368.1	218	C6
RC208.100	N/A	N/A	N/A	1000	7	1121.1	13638	10	7874.8	424	C6

หมายเหตุ คำอธิบายตาราง

Problems = ตัวอย่างปัญหา VRPTW ของ Solomon, Best Solution = ผลเฉลยที่ดีที่สุด, NV = จำนวนรถขนส่ง,

Travel Times = เวลาเดินทางโดยรวม, CPU(s) = เวลาประมวลผลด้วยเครื่องคอมพิวเตอร์, Capacity = ขีดความสามารถในการบรรทุกสินค้า

Computer = รหัสเครื่องคอมพิวเตอร์ที่ใช้ในการประมวลผล, N/A = ไม่ปรากฏข้อมูล

Authors = ชื่อย่อนักวิจัยผู้ค้นหาผลเฉลยเหมาะที่สุดสำหรับปัญหาเทียบเคียง ประกอบด้วย

C - A. Chabrier, “Vehicle Routing Problem with Elementary Shortest Path based Column Generation.” Forthcoming in: Computers and Operations Research (2005).

CR - W. Cook and J. L. Rich, “A parallel cutting plane algorithm for the vehicle routing problem with time windows,” Working Paper, Computational and Applied Mathematics, Rice University, Houston, TX, 1999.

DLP - E. Danna and C. Le Pape, “Accelerating branch-and-price with local search: A case study on the vehicle routing problem with time windows,” In: Column Generation, G. Desaulniers, J. Desrosiers, and M. M. Solomon (eds.), 99-130, Kluwer Academic Publishers (2005).

IV - S. Irnich and D. Villeneuve, "The shortest path problem with k-cycle elimination (k $\geq$ 3): Improving a branch-and-price algorithm for the VRPTW." Forthcoming in: INFORMS Journal of Computing (2005).

KDMSS - N. Kohl, J. Desrosiers, O. B. G. Madsen, M. M. Solomon, and F. Soumis, "2-Path Cuts for the Vehicle Routing Problem with Time Windows," Transportation Science, Vol. 33 (1), 101-116 (1999).

KLM - B. Kallehauge, J. Larsen, and O.B.G. Madsen. "Lagrangean duality and non-differentiable optimization applied on routing with time windows - experimental results." Internal report IMM-REP-2000-8, Department of Mathematical Modelling, Technical University of Denmark, Lyngby, Denmark, 2000.

L - J. Larsen. "Parallelization of the vehicle routing problem with time windows." Ph.D. Thesis IMM-PHD-1999-62, Department of Mathematical Modelling, Technical University of Denmark, Lyngby, Denmark, 1999.

ตารางผนวกที่ ข4 ผลการเปรียบเทียบขั้นตอนวิธีการสำหรับปัญหาขนาด 100 เมือง

Problems	Best Solution				Hybrid Algorithm (HY)			Genetic Algorithm (GA)			Computer
	NV	Travel Times	Authors	Capacity	NV	Travel Times	CPU(s)	NV	Travel Times	CPU(s)	
R101	19	1645.79	H	200	17	1915.5	43079	51	10150	1300	C1
R102	17	1486.12	RT	200	17	1694.3	23716	42	8486.1	1074	C1
R103	13	1292.68	LLH	200	15	1467.4	21222	35	6756.6	920	C1
R104	9	1007.24	M	200	13	1199.3	21948	25	6542	1065	C2
R105	14	1377.11	RT	200	18	1607.9	31908	43	7594.5	2101	C2
R106	12	1251.98	M	200	15	1452	22965	39	7373.3	1787	C3
R107	10	1104.66	S97	200	14	1298.4	21865	33	5989.8	1426	C3
R108	9	960.88	BBB	200	11	1054.3	27517	25	5425.4	1147	C4
R109	11	1194.73	HG	200	15	1341.4	41049	32	6407.4	1468	C4
R110	10	1118.59	M	200	15	1305.1	35938	28	6449.4	1243	C4
R111	10	1096.72	RGP	200	13	1217.6	1289	32	6063.8	671.8	C5
R112	9	982.14	GTA	200	12	1143.9	21033	25	5106.4	991	C6
C101	10	828.94	RT	200	12	3108.8	51477	100	37188	2489	C1
C102	10	828.94	RT	200	13	2676.9	35189	75	24674	1892	C1
C103	10	828.06	RT	200	13	2176.5	23140	51	16638	1269	C1

ตารางผนวกที่ ข4 (ต่อ)

Problems	Best Solution				Hybrid Algorithm (HY)			Genetic Algorithm (GA)			Computer
	NV	Travel Times	Authors	Capacity	NV	Travel Times	CPU(s)	NV	Travel Times	CPU(s)	
C104	10	824.78	RT	200	12	1536.9	23643	26	9454.3	1114	C2
C105	10	828.94	RT	200	17	1417.4	34810	50	2695.9	2104	C3
C106	10	828.94	RT	200	13	1793.1	27552	46	24582	2048	C4
C107	10	828.94	RT	200	11	970.4	40316	49	15902	2141	C4
C108	10	828.94	RT	200	12	1005.8	13174	30	17627.2	635	C5
C109	10	828.94	RT	200	12	1115.3	16049	20	11435.1	811	C6
RC101	14	1696.94	TBGGP	200	18	1960.1	21956	42	10380	1069	C1
RC102	12	1554.75	TBGGP	200	16	1687.2	25346	42	10580	1064	C1
RC103	11	1261.67	S98	200	14	1509.4	30065	36	8368.4	1726	C2
RC104	10	1135.48	CLM	200	13	1368.1	19027	27	10537	1140	C3
RC105	13	1629.44	BBB	200	15	1755.8	24417	42	11222	2031	C4
RC106	11	1424.73	BBB	200	15	1619.4	30894	39	11239	1884	C4
RC107	11	1230.48	S97	200	14	1394.2	18864	32	8671.4	695	C5
RC108	10	1139.82	TBGGP	200	13	1384.8	24067	28	9035.6	1115	C6
R201	4	1252.37	HG	1000	11	1967.6	11186	22	1023	594	C1

ตารางผนวกที่ ข4 (ต่อ)

Problems	Best Solution				Hybrid Algorithm (HY)			Genetic Algorithm (GA)			Computer
	NV	Travel Times	Authors	Capacity	NV	Travel Times	CPU(s)	NV	Travel Times	CPU(s)	
R202	3	1191.7	RGP	1000	8	1683.1	9427	19	8802.1	511	C1
R203	3	939.54	M	1000	8	1408.4	10167	13	7137	361	C1
R204	2	825.52	BVH	1000	6	1080.4	11997	7	7965.8	321	C2
R205	3	994.42	RGP	1000	8	1468.2	17797	13	7856.8	557	C2
R206	3	906.14	SSSD	1000	8	1369.9	13054	12	7616.5	570	C3
R207	2	893.33	BVH	1000	3	1025.8	8900	9	6393.8	428	C3
R208	2	726.75	M	1000	3	862.6	6951	6	5608.4	323	C4
R209	3	909.16	H	1000	8	1359.3	12834	12	6466.2	577	C4
R210	3	939.34	M	1000	6	1283.6	6242	13	7325	296	C5
R211	2	892.71	BVH	1000	6	1028.5	11721	8	6113.3	360	C6
C201	3	591.56	RT	700	10	1931.1	18922	42	50358	1177	C1
C202	3	591.56	RT	700	8	1512.3	14254	34	40823.9	725	C5
C203	3	591.17	RT	700	6	1199.2	18601	20	27702.6	962	C2
C204	3	590.6	RT	700	6	1190.4	10678	11	16971	506	C3
C205	3	588.88	RT	700	6	1019.7	27468	29	35711	1292	C4

ตารางผนวกที่ ข4 (ต่อ)

Problems	Best Solution				Hybrid Algorithm (HY)			Genetic Algorithm (GA)			Computer
	NV	Travel Times	Authors	Capacity	NV	Travel Times	CPU(s)	NV	Travel Times	CPU(s)	
C206	3	588.49	RT	700	7	1469.4	9489	19	21487.6	421	C5
C207	3	588.29	RT	700	7	1273.8	8264	19	25086.8	444	C5
C208	3	588.32	RT	700	5	1063.5	4413	9	103078.2	395	C6
RC201	4	1406.91	M	1000	10	2117.9	20432	24	13601	690	C1
RC202	3	1367.09	CC	1000	9	1820.2	11508	20	12105	443	C5
RC203	3	1049.62	CC	1000	6	1383.7	17555	14	13143	598	C2
RC204	3	798.41	M	1000	5	1114.4	9206	7	13708	346	C3
RC205	4	1297.19	M	1000	8	1840.2	18231	23	13196	1096	C4
RC206	3	1146.32	H	1000	8	1531.3	6525	15	9331.3	344	C5
RC207	3	1061.14	BVH	1000	8	1512.5	7158	15	9970.8	336	C5
RC208	3	828.14	IKMUY	1000	7	1121.1	13638	10	7874.8	424	C6

หมายเหตุ คำอธิบายตาราง

Problems = ตัวอย่างปัญหา VRPTW ของ Solomon, Best Solution = ผลเฉลยที่ดีที่สุด, NV = จำนวนรถขนส่ง,

Travel Times = เวลาเดินทางโดยรวม, CPU(s) = เวลาประมวลผลด้วยเครื่องคอมพิวเตอร์, Capacity = ขีดความสามารถในการบรรทุกสินค้า

Computer = รหัสเครื่องคอมพิวเตอร์ที่ใช้ในการประมวลผล, N/A = ไม่ปรากฏข้อมูล

Authors = ชื่อย่อนักวิจัยผู้ค้นหาผลเฉลยเหมาะที่สุดสำหรับปัญหาเทียบเคียง ประกอบด้วย

BVH - R. Bent and P. Van Hentenryck, "A Two-Stage Hybrid Local Search for the Vehicle Routing Problem with Time Windows," Technical Report CS-01-06, Department of Computer Science, Brown University, 2001.

BBB - J. Berger, M. Barkaoui and O. Bräysy, "A Parallel Hybrid Genetic Algorithm for the Vehicle Routing Problem with Time Windows," Working paper, Defense Research Establishment Valcartier, Canada, 2001.

CC - Z. J. Czech and P. Czarnas, "A Parallel Simulated Annealing for the Vehicle Routing Problem with Time Windows," Proc. 10th Euromicro Workshop on Parallel, Distributed and Network-based Processing, Canary Islands, Spain, (January 9--11, 2002), 376--383.

CLM - J.-F. Cordeau, G. Laporte, and A. Mercier, "A Unified Tabu Search Heuristic for Vehicle Routing Problems with Time Windows," Working Paper CRT-00-03, Centre for Research on Transportation, Montreal, Canada, 2000.

GTA - L. M. Gambardella, E. Taillard, and G. Agazzi, "MACS-VRPTW: A Multiple Ant Colony System for Vehicle Routing Problems with Time Windows," in New Ideas in Optimization, D. Corne, M. Dorigo and F. Glover (eds), 63-76, McGraw-Hill, London, 1999.

HG - J. Homberger and H. Gehring, "Two Evolutionary Metaheuristics for the Vehicle Routing Problem with Time Windows," INFOR, VOL. 37, 297-318, (1999).

H - J. Homberger, "Verteilt-parallele Metaheuristiken zur Tourenplanung," Gaber, Wiesbaden (2000).

IKMUY - T. Ibaraki, M. Kubo, T. Masuda, T. Uno and M. Yagiura, "Effective Local Search Algorithms for the Vehicle Routing Problem with General Time Windows," Working Paper, Department of Applied Mathematics and Physics, Kyoto University, Japan, 2001.

LLH - H. Li, A. Lim, and J. Huang, "Local Search with Annealing-like Restarts to Solve the VRPTW," Working Paper, Department of Computer Science, National University of Singapore, 2001.

M - D. Mester, "An Evolutionary Strategies Algorithm for Large Scale Vehicle Routing Problem with Capacitate and Time Windows Restrictions, "Working Paper, Institute of Evolution, University of Haifa, Israel (2002).

RT - Y. Rochat and E.D. Taillard, "Probabilistic Diversification and Intensification in Local Search for Vehicle Routing," Journal of Heuristics 1, 147-167, (1995).

RGP - L.M. Rousseau, M. Gendreau and G. Pesant, "Using Constraint-Based Operators to Solve the Vehicle Routing Problem with Time Windows," Journal of Heuristics, forthcoming.

SSSD - G. Schrimpf, J. Schneider, H. Stamm-Wilbrandt and G. Dueck, "Record Breaking Optimization Results Using the Ruin and Recreate Principle," Journal of Computational Physics 159, 139-171, (2000).

S97 - P. Shaw, "A New Local Search Algorithm Providing High Quality Solutions to Vehicle Routing Problems," Working Paper, University of Strathclyde, Glasgow, Scotland, 1997.

S98 - P. Shaw, "Using Constraint Programming and Local Search Methods to Solve Vehicle Routing Problems," in Principles and Practice of Constraint Programming - CP98, Lecture Notes in Computer Science, 417-431, M. Maher and J.-F. Puget (eds), Springer-Verlag, New York, 1998.

TBGGP - E. Taillard, P. Badeau, M. Gendreau, F. Geurtin, and J.Y. Potvin, "A Tabu Search Heuristic for the Vehicle Routing Problem with Time Windows,"Transportation Science, 31, 170-186, (1997).

ประวัติการศึกษา และการทำงาน

ชื่อ –นามสกุล	นางธรินี มณีศรี
วัน เดือน ปี ที่เกิด	วันที่ 29 มกราคม 2520
สถานที่เกิด	จังหวัดตราด
ประวัติการศึกษา	วศ.บ.วิศวกรรมอุตสาหการ ม.เทคโนโลยีสุรนารี วศ.ม.วิศวกรรมอุตสาหการ ม.เกษตรศาสตร์
ตำแหน่งหน้าที่การงานปัจจุบัน	อาจารย์ประจำ
สถานที่ทำงานปัจจุบัน	มหาวิทยาลัยศรีปทุม
ผลงานดีเด่นและรางวัลทางวิชาการ	
ทุนการศึกษาที่ได้รับ	ได้รับทุนสนับสนุนคุณภาพงานวิจัยระดับบัณฑิตศึกษา เพื่อการตีพิมพ์ในวารสารระดับชาติและนานาชาติ จาก บัณฑิตวิทยาลัย มหาวิทยาลัยเกษตรศาสตร์ (พ.ศ. 2551)