การจัดตารางพยาบาลแบบหลายวัตถุประสงค์ด้วยวิธีการฮิวริสติก : กรณีศึกษาของโรงพยาบาลเอกชนในประเทศไทย

นายนลธวัช สุคนธ์ภัทร

จุฬาลงกรณ์มหาวิทยาลัย
CHULALONGKORN UNIVERSITY

วิทยานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรมหาบัณฑิต
สาขาวิชาวิศวกรรมอุตสาหการ ภาควิชาวิศวกรรมอุตสาหการ
คณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย
ปีการศึกษา 2566

A heuristic approach to the multi-objective nurse scheduling problem : A case study of private hospital in Thailand

Mr. Nontawat Sukonpat

จุฬาลงกรณ์มหาวิทยาลัย
CHULALONGKORN UNIVERSITY

A Thesis Submitted in Partial Fulfillment of the Requirements
for the Degree of Master of Engineering in Industrial Engineering
Department of Industrial Engineering
Faculty Of Engineering
Chulalongkorn University
Academic Year 2023

หัวข้อวิทยานิพนธ์	การจัดตารางพยาบาลแบบหลายวัตถุประสงค์ด้วยวิธีการฮิวริสติก : กรณีศึกษาของโรงพยาบาลเอกชนในประเทศไทย
โดย	นายนลธวัช สุคนธ์ภัทร
สาขาวิชา	วิศวกรรมอุตสาหการ
อาจารย์ที่ปรึกษาวิทยานิพนธ์หลัก	รองศาสตราจารย์ ดร.พิศิษฎ์ จารุมณีโรจน์

คณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย อนุมัติให้นับวิทยานิพนธ์ฉบับนี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรมหาบัณฑิต

..... คณบดีคณะวิศวกรรมศาสตร์
(ศาสตราจารย์ ดร.สุพจน์ เตชวรสินสกุล)

คณะกรรมการสอบวิทยานิพนธ์

..... ประธานกรรมการ
(รองศาสตราจารย์ ดร.นระเกณฑ์ พุ่มชูศรี)

..... อาจารย์ที่ปรึกษาวิทยานิพนธ์หลัก
(รองศาสตราจารย์ ดร.พิศิษฎ์ จารุมณีโรจน์)

..... กรรมการ
(ดร.ภูวดล ดุษฎีรังสีกุล)

..... กรรมการภายนอกมหาวิทยาลัย
(รองศาสตราจารย์ ดร.ชูเวช ชาญสง่าเวช)

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

นลธวัช สุคนธ์ภัทร : การจัดตารางพยาบาลแบบหลายวัตถุประสงค์ด้วยวิธีการฮิวริสติก : กรณีศึกษาของโรงพยาบาลเอกชนในประเทศไทย. (A heuristic approach to the multi-objective nurse scheduling problem : A case study of private hospital in Thailand) อ.ที่ปรึกษาหลัก : รศ. ดร.พิศิษฐ์ จารุมณีโรจน์

งานวิจัยนี้มีวัตถุประสงค์หลักเพื่อพัฒนาแนวทางการแก้ปัญหาการจัดตารางการทำงานของพยาบาลแบบหลายวัตถุประสงค์ (Multi-objective Nurse Scheduling Problem : NSP) ด้วยแบบจำลองทางคณิตศาสตร์ ควบคู่ไปกับวิธีการฮิวริสติก ปัญหาการจัดตารางการทำงานพยาบาลแบบหลายวัตถุประสงค์นี้ เป็นปัญหาที่มีความซับซ้อน อันเป็นผลเนื่องมาจากความหลากหลายของข้อจำกัด และวัตถุประสงค์หลักของปัญหาต่างๆ ที่มีความขัดแย้งกัน นอกจากนี้ โรงพยาบาลส่วนใหญ่ยังขาดระบบสนับสนุนการตัดสินใจในการจัดตารางการทำงาน ส่งผลทำให้ผู้จัดตารางการทำงานจำเป็นต้องดำเนินการจัดตารางการทำงานด้วยตนเอง ซึ่งมักใช้เวลาในการจัดที่ยาวนาน และผิดพลาดได้ง่าย เพื่อแก้ปัญหาดังกล่าว ผู้วิจัยได้พัฒนาแบบจำลองทางคณิตศาสตร์ที่สะท้อนภาพการทำงานของผู้จัดตารางการทำงานพยาบาลแบบหลายวัตถุประสงค์ขึ้น โดยมุ่งเน้นไปที่แผนกฉุกเฉิน เนื่องจากแผนกดังกล่าวมีความซับซ้อนในการปฏิบัติงานมากกว่าแผนกอื่นๆ อย่างไรก็ดี ผู้วิจัยพบว่า เมื่อปัญหาดังกล่าวมีขนาดใหญ่ขึ้น ประสิทธิภาพของแบบจำลองทางคณิตศาสตร์มีแนวโน้มที่ลดถอยลง ทั้งในเชิงคุณภาพของคำตอบ และเวลาที่ใช้ในการหาคำตอบ อันเป็นผลเนื่องมาจากความซับซ้อนของปัญหาดังกล่าว ผู้วิจัยจึงได้ทำการพัฒนาฮิวริสติกขึ้น เพื่อเพิ่มประสิทธิภาพ และลดเวลาการหาคำตอบของปัญหาการจัดตารางการทำงานพยาบาลแบบหลายวัตถุประสงค์ โดยอ้างอิงจาก Adaptive Large Neighborhood Search (ALNS) ควบคู่ไปกับ Non-dominated Sorting Genetic Algorithm III (NSGA-III) จากผลการทดลอง ผู้วิจัยพบว่า วิธีการฮิวริสติกที่ถูกพัฒนาขึ้นมีคุณภาพเทียบเท่ากับวิธีแม่นตรงในปัญหาขนาดเล็ก โดยชุดคำตอบที่ได้จากวิธีการฮิวริสติกมีคุณภาพเทียบเท่ากับวิธีแม่นตรง หากแต่ใช้เวลาในการหาคำตอบที่สั้นกว่ามาก สำหรับปัญหาที่มีขนาดใหญ่ขึ้น วิธีการฮิวริสติกที่ถูกพัฒนาขึ้นก็สามารถสร้างชุดคำตอบที่มีความหลากหลายในระยะเวลาที่สั้นกว่าวิธีการแบบดั้งเดิมเป็นอย่างมาก

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

สาขาวิชา	วิศวกรรมอุตสาหการ	ลายมือชื่อนิสิต .....
ปีการศึกษา	2566	ลายมือชื่อ อ.ที่ปรึกษาหลัก .....

6470344021 : MAJOR INDUSTRIAL ENGINEERING

KEYWORD: Nurse Scheduling Problem, Adaptive Large Neighborhood Search (ALNS), Non-dominated Sorting Genetic Algorithm III (NSGA-III)

Nontawat Sukonpat : A heuristic approach to the multi-objective nurse scheduling problem : A case study of private hospital in Thailand . Advisor: Assoc. Prof. Pisit Jarumaneeroj, Ph.D.

This research aims to develop solution approaches to the Multi-Objective Nurse Scheduling Problem (NSP) — primarily based on mathematical modeling and heuristic approaches. The NSP could be regarded as one of complicated health care problems due to a wide range of constraints, stemming from both the hospitals and the operating nurses themselves, coupled with conflicts among different objectives. The NSP, in practice, is even worse off as there are normally no decision support systems that help generate efficient schedules. This, in turn, creates a huge burden on schedulers that need to manually generate the schedules on their own. Without decision support tools, schedule planning tends to be time consuming; yet, the resulting schedules are less effective and prone to errors. To address these issues, a mathematical model — reflecting the operation of nurse scheduling in an emergency department — is herein developed and solved using information of a case study hospital as a test case. We find that the mathematical modeling approach is capable of solving only small single objective NSP instances; and, it tends to terminate with a run-out-of-memory error in larger single objective NSPs. To better solve large and complicated NSPs, an Adaptive Large Neighborhood Search (ALNS) framework that has been combined with Non-dominated Sorting Genetic Algorithm III (NSGA-III) is thence developed and applied to the underlying NSPs. Based on our computational results, the proposed heuristic is comparably efficient as it could provide solutions that match the optimal solutions to small single NSPs, with less computational time. Furthermore, the proposed heuristic could generate a set of well-balanced solutions to the multi-objective NSPs that could be applied to the test case within an acceptable time period.

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

Field of Study:	Industrial Engineering	Student's Signature .....
Academic Year:	2023	Advisor's Signature .....

กิตติกรรมประกาศ

ในการจัดทำวิทยานิพนธ์ฉบับนี้ ข้าพเจ้าขอขอบพระคุณ รองศาสตราจารย์ ดร.พิศิษฏ์ จารุมณีโรจน์ อาจารย์ที่ปรึกษาวิทยานิพนธ์ ที่ท่านได้สละเวลาอันมีค่า เพื่อให้คำปรึกษา แนวความคิด รวมทั้งแนวทางการหาความรู้ ตลอดระยะเวลาในการดำเนินการทำวิจัย อันเป็นผลให้วิทยานิพนธ์ฉบับนี้มีความครบถ้วนสมบูรณ์ไปได้ด้วยดี

ขอขอบพระคุณ รองศาสตราจารย์ ดร.นระเกณฑ์ พุ่มชูศรี (ประธานกรรมการสอบวิทยานิพนธ์) รองศาสตราจารย์ ดร.ชูเวช ชาญสง่าเวช และ ดร.ภูวดล ดุษฎีรังสีกุล (กรรมการสอบวิทยานิพนธ์) ที่ได้สละเวลาในการมอบความรู้ และชี้แนะแนวทางในการพัฒนาวิทยานิพนธ์ ทำให้วิทยานิพนธ์ฉบับนี้มีความถูกต้องสมบูรณ์มากยิ่งขึ้น

ขอขอบพระคุณ บริษัทกรณีศึกษา คณะทำงาน และบุคลากรในแผนกฉุกเฉินทุกท่าน ที่ได้สละเวลาถ่ายทอดความรู้ และประสบการณ์ เป็นผลให้งานวิจัยได้เข้าไปเป็นส่วนหนึ่งของการแก้ปัญหา และเพิ่มประสิทธิภาพในการทำงานของบุคลากรภายในแผนกฉุกเฉินได้

สุดท้ายนี้ ข้าพเจ้าขอขอบพระคุณบิดา มารดา และครอบครัว ที่คอยให้กำลังใจและเป็นแรงกระตุ้นให้ผู้วิจัยทำวิทยานิพนธ์ฉบับนี้จนสำเร็จลุล่วงไปด้วยดี ข้าพเจ้าหวังเป็นอย่างยิ่งว่าวิทยานิพนธ์ฉบับนี้ จะสามารถนำไปใช้เพื่อก่อให้เกิดประโยชน์แก่สังคม และผู้ที่สนใจได้เป็นอย่างดี

นลธวัช สุคนธ์ภัทร

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

สารบัญ

หน้า

.....ค

บทคัดย่อภาษาไทย.....ค

.....ง

บทคัดย่อภาษาอังกฤษ.....ง

กิตติกรรมประกาศ.....จ

สารบัญ.....ฉ

สารบัญตาราง.....ฌ

สารบัญรูป.....ฎ

บทที่ 1 บทนำ.....1

1.1 ที่มาและความสำคัญของงานวิจัย.....1

1.2 วัตถุประสงค์ของงานวิจัย.....9

1.3 ขอบเขตของงานวิจัย.....10

1.4 ประโยชน์ที่ได้รับจากงานวิจัย.....10

บทที่ 2 บททบทวนวรรณกรรม.....11

2.1 ปัญหาการจัดตารางการทำงานพยาบาล (Nurse Scheduling Problem).....11

2.2 การประยุกต์ใช้แบบจำลองทางคณิตศาสตร์ (Mathematical Model).....12

2.3 การประยุกต์ใช้วิธีการฮิวริสติก (Heuristic).....14

2.4 Fast Non-Dominated Sorting (FNDS).....15

2.5 Adaptive Large Neighborhood Search (ALNS).....16

บทที่ 3 วิธีการดำเนินงานวิจัย.....20

3.1 ลักษณะของปัญหา.....20

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

3.2 แบบจำลองทางคณิตศาสตร์ 23
3.2.1 ค่าดัชนี (Indices) 23
3.2.2 ค่าคงที่ (Parameters) 23
3.2.3 ตัวแปรตัดสินใจ (Decision Variables) 24
3.2.4 ฟังก์ชันจุดประสงค์ (Objective Function) 26
3.2.5 ข้อจำกัด (Constraints) 27
3.3 การแก้ปัญหาการจัดตารางการทำงานของพยาบาลด้วยวิธีฮิวริสติก 32
3.3.1 การนำเข้าข้อมูล (Import Parameters) 33
3.3.2 การสร้างคำตอบเริ่มต้น (Initial Solution) 34
3.3.3 การสร้างคำตอบที่ดีที่สุด (Pareto Front) 34
3.3.4 Fast Non-Dominated Sorting (FNDS) 35
3.3.4.1 Front 36
3.3.4.2 Crowding Distance 37
3.3.4.3 Reference Points 39
3.3.4.4 Niche Count 41
3.3.5 การปรับปรุงคุณภาพคำตอบด้วย ALNS 42
3.3.5.1 การเลือกตัวทำลายคำตอบ (Destroy Operator Selection) 44
3.3.5.2 การเลือกทำลายคำตอบแบบ Random Removal 45
3.3.5.3 การเลือกทำลายคำตอบแบบ Shaw Removal 46
3.3.5.4 การเลือกทำลายคำตอบแบบ History-Based Removal 49
3.3.5.5 การเลือกตัวซ่อมแซมคำตอบ (Repair Operator Selection) 50
3.3.5.6 การเลือกซ่อมแซมคำตอบแบบ Greedy Insertion 50
3.3.5.7 การเลือกซ่อมแซมคำตอบแบบ Regret-Based Insertion 51
3.3.5.8 การเลือกซ่อมแซมคำตอบแบบ Least-Loaded Nurse Repair 53

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

3.3.6 การปรับปรุงน้ำหนักของตัวดำเนินการ (Operator Weights Updating) 53
3.3.7 การปรับปรุง - ตรวจสอบคุณภาพคำตอบด้วย Simulated Annealing (SA) 54
3.3.7.1 ขั้นตอนการยอมรับคำตอบด้วยวิธี Simulated Annealing (SA) 55
3.3.7.2 การปรับปรุงคำตอบในพื้นที่ใกล้เคียง (Local Search) 55
3.3.7.3 การปรับปรุงคำตอบโดยการสลับตำแหน่ง (Relocation) 55
บทที่ 4 ผลการดำเนินงานวิจัย 56
4.1 ข้อมูลนำเข้าสำหรับการหาคำตอบ 56
4.2 การนำเข้าข้อมูลที่ใช้ในการหาคำตอบของปัญหาขนาดเล็ก 58
4.3 การหาคำตอบแบบหลายวัตถุประสงค์ด้วย Preemptive Goal Programming กับปัญหาขนาดเล็ก 58
4.4 การหาคำตอบแบบหลายวัตถุประสงค์ด้วยฮิวริสติกกับปัญหาขนาดเล็ก 61
4.5 การหาคำตอบแบบหลายวัตถุประสงค์ด้วยฮิวริสติกกับปัญหาขนาดใหญ่ 63
บทที่ 5 สรุปผลการดำเนินงานวิจัย 72
5.1 สรุปผลการดำเนินงานวิจัย 72
5.2 ข้อเสนอแนะ 73
บรรณานุกรม 75
ประวัติผู้เขียน 80

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

สารบัญตาราง

หน้า

ตารางที่ 1-1 รายได้เฉลี่ยของบุคลากรทางการแพทย์แต่ละตำแหน่ง (หน่วย : บาท) 2
ตารางที่ 1-2 จำนวนพยาบาลที่ต้องการในแต่ละกะการทำงานของวันจันทร์-ศุกร์ (หน่วย : คน) 5
ตารางที่ 1-3 จำนวนพยาบาลที่ต้องการในแต่ละกะการทำงานของวันเสาร์-อาทิตย์ (หน่วย : คน) 6
ตารางที่ 1-4 ค่าจ้างพิเศษเมื่อพยาบาลแต่ละตำแหน่งเข้ากะการทำงาน (หน่วย : บาท) 7
ตารางที่ 2-1 โครงสร้างและเทคทิคของ ALNS 17
ตารางที่ 2-2 ตารางความเกี่ยวข้อง และความแตกต่างของงานวิจัย 18
ตารางที่ 3-1 การละเมิดกฎข้อจำกัดและบทลงโทษ 24
ตารางที่ 3-2 ตารางแสดงพิกัดของจุดอ้างอิง (Reference point) 40
ตารางที่ 4-1 รายละเอียดการดำเนินการหาคำตอบสำหรับแต่ละวิธี 56
ตารางที่ 4-2 พารามิเตอร์สำหรับการทำงานของ Adaptive Large Neighborhood Search 57
ตารางที่ 4-3 การเรียงลำดับความสำคัญของผลเฉลย 59
ตารางที่ 4-4 ผลเฉลยที่ 1 จากการหาคำตอบด้วยวิธี PGP กับปัญหาขนาดเล็ก 60
ตารางที่ 4-5 ผลเฉลยที่ 2 จากการหาคำตอบด้วยวิธี PGP กับปัญหาขนาดเล็ก 60
ตารางที่ 4-6 ผลเฉลยที่ 3 จากการหาคำตอบด้วยวิธี PGP กับปัญหาขนาดเล็ก 60
ตารางที่ 4-7 ผลเฉลยที่ 4 จากการหาคำตอบด้วยวิธี PGP กับปัญหาขนาดเล็ก 61
ตารางที่ 4-8 ผลเฉลยที่ 5 จากการหาคำตอบด้วยวิธี PGP กับปัญหาขนาดเล็ก 61
ตารางที่ 4-9 ผลเฉลยที่ 6 จากการหาคำตอบด้วยวิธี PGP กับปัญหาขนาดเล็ก 61
ตารางที่ 4-10 ผลเฉลยจากวิธีฮิวริสติกสำหรับปัญหาขนาดเล็ก 62
ตารางที่ 4-11 การเปรียบเทียบคุณภาพคำตอบระหว่าง IBM CPLEX Optimization Studio และฮิวริสติกของปัญหาขนาดเล็ก 62
ตารางที่ 4-12 ผลเฉลยจากวิธีฮิวริสติกสำหรับปัญหาขนาดใหญ่ 63

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

ตารางที่ 4-13 ตารางค่าวัตถุประสงค์ของผลเฉลยจำนวน 54 ผลเฉลยจากการประมวลผล 10 ครั้ง 65
ตารางที่ 4-14 ตารางการทำงานพยาบาลจากผลเฉลยที่มีค่าวัตถุประสงค์ที่ 1 ต่ำที่สุด..... 68
ตารางที่ 4-15 ตารางการทำงานพยาบาลจากผลเฉลยที่มีค่าวัตถุประสงค์ที่ 2 ต่ำที่สุด..... 69
ตารางที่ 4-16 ตารางการทำงานพยาบาลจากผลเฉลยที่มีค่าวัตถุประสงค์ที่ 3 ต่ำที่สุด..... 70
ตารางที่ 4-17 ตารางการทำงานของพยาบาลแผนกฉุกเฉินในเดือนกันยายน พ.ศ.2566..... 71

จุฬาลงกรณ์มหาวิทยาลัย
CHULALONGKORN UNIVERSITY

สารบัญรูป

หน้า

รูปที่ 1-1 จำนวนบุคลากรทางการแพทย์แต่ละตำแหน่งของกระทรวงสาธารณสุขทั้งประเทศ 1
รูปที่ 1-2 ค่าใช้จ่ายบุคลากรทางการแพทย์แต่ละตำแหน่งของกระทรวงสาธารณสุขทั้งประเทศ 2
รูปที่ 1-3 ชั่วโมงการทำงานของแต่ละชนิดกะการทำงาน 5
รูปที่ 3-1 ขั้นตอนการดำเนินงานการจัดตารางการทำงานของพยาบาล 21
รูปที่ 3-2 ขั้นตอนการทำงานของฮิวริสติกการจัดตารางการทำงานของพยาบาล 32
รูปที่ 3-3 Pseudocode แสดงขั้นตอนการนำเข้าข้อมูลพื้นฐาน 33
รูปที่ 3-4 Pseudocode แสดงขั้นตอนการสร้างชุดคำตอบเริ่มต้น (Initial solutions) 34
รูปที่ 3-5 Pseudocode แสดงขั้นตอน Fast Non-Dominated Sorting 35
รูปที่ 3-6 ตัวอย่างการแบ่ง Front ด้วยการตรวจสอบคำตอบที่ไม่ถูกครอบงำ 36
รูปที่ 3-7 Pseudocode แสดงขั้นตอน Fronts 37
รูปที่ 3-8 Pseudocode แสดงขั้นตอนคำนวณ Crowding distance 38
รูปที่ 3-9 จุดอ้างอิง (Reference point) บนพื้นที่ของ Objective space 39
รูปที่ 3-10 Pseudocode แสดงขั้นตอนการสร้างจุดอ้างอิง (Reference points) 40
รูปที่ 3-11 Pseudocode แสดงขั้นตอนคำนวณ Niche count 42
รูปที่ 3-12 ตัวอย่างความน่าจะเป็นในการสุ่มตัวทำลายด้วยวิธี Roulette wheel 43
รูปที่ 3-13 Pseudocode แสดงขั้นตอนการทำงานของ Adaptive Large Neighborhood Search (ALNS) 44
รูปที่ 3-14 วิธีในการทำลายคำตอบ (Destroy operator) 45
รูปที่ 3-15 Pseudocode แสดงขั้นตอนการทำงานของ Random removal 45
รูปที่ 3-16 ชนิดของตัวทำลายคำตอบแบบ Shaw's 46
รูปที่ 3-17 Pseudocode แสดงขั้นตอนการทำงานของ Shaw's min shift 46

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

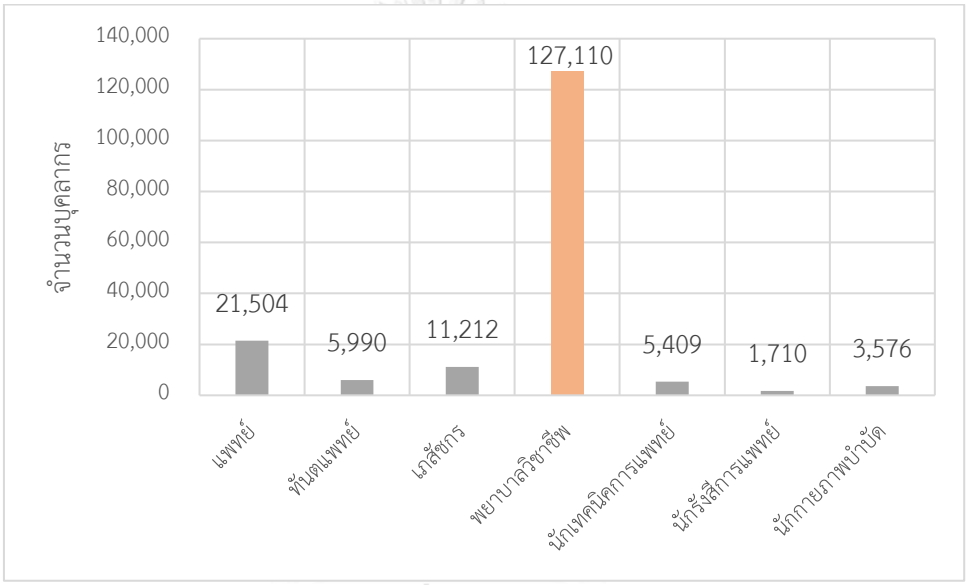
รูปที่ 3-18 Pseudocode แสดงขั้นตอนการทำงานของ Shaw’s max shift..... 47
รูปที่ 3-19 Pseudocode แสดงขั้นตอนการทำงานของ Shaw’s min working day 47
รูปที่ 3-20 Pseudocode แสดงขั้นตอนการทำงานของ Shaw’s max working day..... 48
รูปที่ 3-21 Pseudocode แสดงขั้นตอนการทำงานของ Shaw’s max violation..... 48
รูปที่ 3-22 Pseudocode แสดงขั้นตอนการทำงานของ History based removal..... 49
รูปที่ 3-23 วิธีในการซ่อมแซมคำตอบ (Repair operators) 50
รูปที่ 3-24 Pseudocode แสดงขั้นตอนการทำงานของ Greedy insertion..... 51
รูปที่ 3-25 ชนิดของตัวซ่อมแซมคำตอบแบบ Regret-based insertion..... 51
รูปที่ 3-26 Pseudocode แสดงขั้นตอนการทำงานของ 2 Regret-based insertion 52
รูปที่ 3-27 Pseudocode แสดงขั้นตอนการทำงานของ 3 Regret-based insertion 52
รูปที่ 3-28 Pseudocode แสดงขั้นตอนการทำงานของ Least-loaded nurse repair..... 53
รูปที่ 3-29 การปรับปรุง - ตรวจสอบคุณภาพคำตอบด้วย Simulated Annealing (SA) 54
รูปที่ 4-1 ขั้นตอนการหาคำตอบแบบหลายวัตถุประสงค์ด้วยวิธี Preemptive goal programming
..... 59
รูปที่ 4-2 การ Plot 3 มิติ จากผลเฉลยใน Front 1 ด้วยการประมวลผล 10 ครั้ง 67

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

บทที่ 1
บทนำ

1.1 ที่มาและความสำคัญของงานวิจัย

ตำแหน่งพยาบาลเป็นเสมือนเสาหลักของโรงพยาบาลที่เป็นกำลังขับเคลื่อนหลักในโรงพยาบาล ซึ่งทำหน้าที่ในทุกภาคส่วนตั้งแต่การสอบถามอาการของผู้ป่วย ประเมินอาการ คัดกรอง ให้คำปรึกษา และเป็นตัวกลางในการสื่อสารระหว่างแพทย์ และผู้ป่วย ด้วยเหตุดังกล่าวพยาบาลจึงเป็นตำแหน่งที่มีจำนวนมากที่สุดในโรงพยาบาล ดังแสดงในรูปที่ 1-1 (หน่วย : คน)



รูปที่ 1-1 จำนวนบุคลากรทางการแพทย์แต่ละตำแหน่งของกระทรวงสาธารณสุขทั้งประเทศ
(Drug And Medical Supply Information Center, 2564)

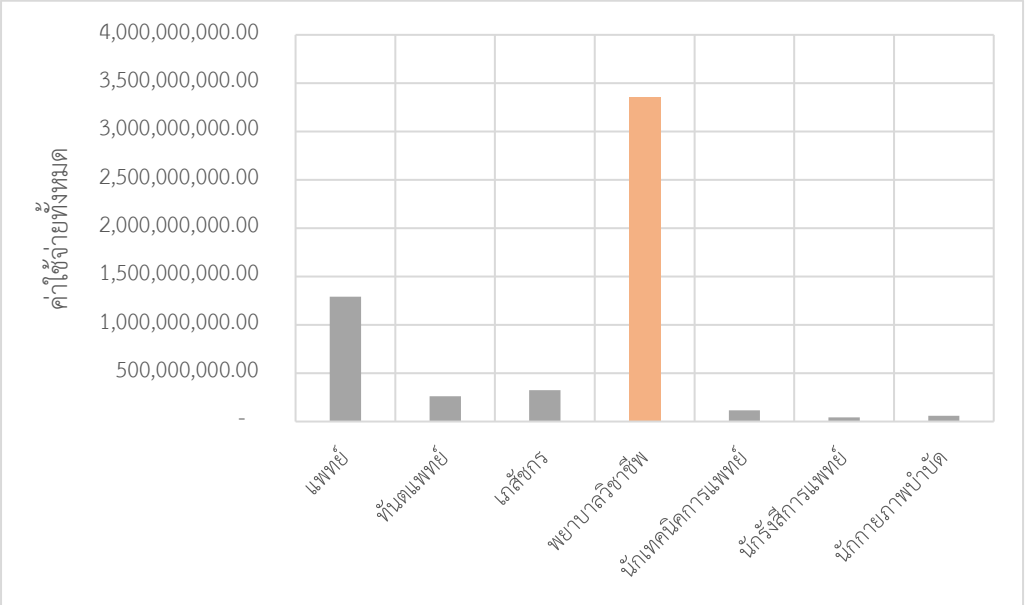
จากรูปที่ 1-1 จะสังเกตได้ว่า จำนวนบุคลากรทางการแพทย์ทั้งหมดจากกระทรวงสาธารณสุขจะมีจำนวนของพยาบาลเมื่อเทียบกับจำนวนบุคลากรทางการแพทย์อื่นมากกว่าอย่างเห็นได้ชัด ซึ่งหากคิดเป็นเปอร์เซ็นต์ จะมีสัดส่วนมากถึง 72 %ของจำนวนบุคลากรทั้งหมดซึ่งสอดคล้องกับปริมาณงาน และหน้าที่ ที่มีความหลากหลายของพยาบาล

นอกจากปริมาณบุคลากรทางการแพทย์ที่สามารถบอกอัตรากำลังของแต่ละตำแหน่งแล้ว ข้อมูลที่สำคัญอีกมิติหนึ่ง คือ รายได้เฉลี่ยของบุคลากรทางการแพทย์ ดังตารางที่ 1-1 (หน่วย : บาท) ที่จะทำให้เราทราบถึงค่าใช้จ่ายรวมทั้งหมดของโรงพยาบาลว่าในตำแหน่งใดมีค่าใช้จ่ายในโรงพยาบาลสูงที่สุด เพื่อที่จะเลือกใช้บุคลากรทางการแพทย์ตำแหน่งนั้นเป็นตัวนำร่องในการปรับปรุงคุณภาพในการทำงาน

ตารางที่ 1-1 รายได้เฉลี่ยของบุคลากรทางการแพทย์แต่ละตำแหน่ง (หน่วย : บาท)

ตำแหน่ง	เงินเดือนเฉลี่ย
แพทย์	60,000
ทันตแพทย์	43,800
เภสัชกร	28,800
พยาบาลวิชาชีพ	26,400
นักเทคนิคการแพทย์	21,600
นักรังสีการแพทย์	25,800
นักกายภาพบำบัด	16,800

ถึงแม้ว่าพยาบาลวิชาชีพจะไม่ได้เป็นตำแหน่งที่มีเงินเดือนสูงที่สุดในบรรดาบุคลากรทางการแพทย์ของโรงพยาบาล แต่ด้วยจำนวนของพยาบาลวิชาชีพที่มีจำนวนมากจึงส่งผลให้ค่าใช้จ่ายในการจ้างพยาบาลรวมทั้งหมดเป็นค่าใช้จ่ายหลักของโรงพยาบาล ซึ่งมีค่าสูงกว่าค่าใช้จ่ายของบุคลากรทางการแพทย์ตำแหน่งอื่นๆ เป็นอย่างมาก ดังแสดงในรูปที่ 1-2 (หน่วย : บาท)



รูปที่ 1-2 ค่าใช้จ่ายบุคลากรทางการแพทย์แต่ละตำแหน่งของกระทรวงสาธารณสุขทั้งประเทศ (Drug And Medical Supply Information Center, 2564)

เพื่อลดค่าใช้จ่ายที่เกินความจำเป็น เราอาจมุ่งเน้นไปที่การสร้างตารางการทำงานพยาบาลที่มีประสิทธิภาพ ซึ่งการดำเนินการดังกล่าว นอกจากจะช่วยลดความสูญเสียที่เกิดขึ้นจากการทำงานแล้ว ยังส่งผลทำให้การปฏิบัติหน้าที่ของพยาบาลเป็นไปอย่างมีประสิทธิภาพมากยิ่งขึ้นอีกด้วย

การจัดตารางการทำงานพยาบาลเป็นหนึ่งในกิจกรรมของการจัดเวรพยาบาล ถ้าหากอยู่ในอุตสาหกรรมอื่นจะถูกเรียกว่า การจัดตารางการทำงาน ถือเป็นหนึ่งในกิจกรรมที่ถูกใช้งานในหลากหลายอุตสาหกรรม เช่น การจัดตารางพนักงานในสายการผลิตเพื่อบริหารกำลังคนให้เพียงพอต่อภาระงาน หรือการจัดตารางการทำงานของเครื่องจักรเพื่อให้เวลาปิดงานของระบบ (Makespan) มีค่าต่ำที่สุด การจัดตารางที่ดีนอกจากจะส่งผลดีต่อประสิทธิภาพในการทำงาน และเกิดความคุ้มค่าในการใช้ทรัพยากรแล้ว ยังเป็นการเพิ่มความพึงพอใจของผู้ทำงานโดยตรง ซึ่งหลักการดังกล่าว สามารถนำไปประยุกต์ใช้กับการจัดตารางการทำงานพยาบาลได้ หากแต่จะมีความแตกต่างกันในแง่ของข้อจำกัดในการทำงานเท่านั้น

ในปัจจุบันโรงพยาบาลส่วนใหญ่ยังมักทำการจัดตารางด้วยมือ หรือที่นิยมเรียกในอีกชื่อหนึ่งว่า “การจัดตารางด้วยตนเอง” เนื่องจากขาดระบบช่วยสนับสนุนการตัดสินใจ (Decision support system) ส่งผลทำให้ใช้เวลาในการจัดตารางที่ยาวนาน และมักเกิดความผิดพลาดได้ง่าย ทั้งนี้ ตารางการทำงานที่ดี ควรแก้ไขได้ง่าย รองรับข้อจำกัดได้อย่างครอบคลุม และมีประสิทธิภาพ โดยในงานวิจัยนี้จะมุ่งเน้นไปที่การจัดตารางการทำงานของพยาบาลแบบหลายวัตถุประสงค์ (Multi-objective nurse scheduling problem) เพื่อทดแทนการจัดตารางด้วยตนเองและตอบสนองต่อความต้องการของพยาบาล

เนื่องจากปัญหาการจัดตารางการทำงานพยาบาลไม่เพียงแต่จะเกี่ยวข้องกับการจัดสรรทรัพยากรมนุษย์แต่ยังสะท้อนถึงประสิทธิภาพของการดูแลคนไข้ และคุณภาพชีวิตของพยาบาล (Boonprasert, 1978) การจัดตารางการทำงานของพยาบาลส่วนใหญ่จึงมีความซับซ้อน เนื่องจากต้องคำนึงถึงกฎเกณฑ์ต่างๆ ไม่ว่าจะเป็นกฎหมายแรงงาน กฎสภาพยาบาล และกฎเฉพาะของโรงพยาบาลแต่ละแห่ง นอกจากนี้ เรายังต้องคำนึงถึงรูปแบบการให้บริการที่หลากหลาย ซึ่งส่งผลทำให้ไม่มีอัลกอริทึมใดที่เหมาะสมกับการจัดตารางการทำงานของพยาบาลที่สามารถนำไปประยุกต์ใช้ได้กับทุกโรงพยาบาล เรายังคงต้องปรับข้อจำกัดต่างๆให้เหมาะสมกับบริบทของโรงพยาบาลที่จะนำไปใช้งานด้วย (Maenhout et al., 2010)

สังเกตว่า ในมุมของผู้จัดตารางการทำงาน เราจำเป็นต้องตระหนักถึง ทั้งภาระงานของผู้ปฏิบัติงาน และจำนวนผู้ป่วยที่เข้ารับบริการ เราจึงไม่ควรกำหนดกะการทำงานที่มุ่งเน้นเพียงการรองรับผู้ป่วยหากแต่สร้างภาระงานที่ไม่เป็นธรรม หรือการเข้ากะทำงานที่ติดต่อกันจนผู้ปฏิบัติงานอ่อนล้า นอกจากนี้เรายังต้องตระหนักถึงปริมาณงานของพยาบาลที่ได้รับมอบหมายภายในกรอบระยะเวลาวางแผนตารางการทำงานพยาบาล โดยตารางการทำงานพยาบาลจะต้องมีปริมาณงานไม่เกินที่กำหนด และมีความเท่าเทียมกันให้ได้มากที่สุด เพื่อไม่ให้พยาบาลได้รับภาระงานที่มากจนเกิดความเครียดสะสม ซึ่งจะเป็นปัญหาการลาออกของพยาบาลที่อาจเกิดขึ้นได้ในอนาคต

ถึงแม้ว่าในโรงพยาบาลจะมีหลายแผนก เช่น แผนกผู้ป่วยใน (Inpatient department) แผนกผู้ป่วยนอก (Outpatient department) แผนกรังสี (Radiology department) เป็นต้น แต่ในแผนกส่วนใหญ่แล้วจะมีเวลาทำ

การที่แน่นอน ไม่จำเป็นต้องมีพยาบาลเตรียมความพร้อมตลอด 24 ชั่วโมง พยาบาลมีหน้าที่ที่ชัดเจน เป็นงานที่ไม่เร่งด่วน ซึ่งสิ่งที่กล่าวมาเหล่านี้จะไม่เกิดขึ้นในแผนกฉุกเฉิน เนื่องจากแผนกฉุกเฉินมีความจำเป็นต้องเตรียมความพร้อมกับสถานการณ์ฉุกเฉินที่อาจจะเกิดขึ้นได้ทุกเมื่อ ทำให้แผนกฉุกเฉินมีความพิเศษในด้านข้อจำกัด และความยากในการกระจายงานให้มีความเหมาะสม

ด้วยความซับซ้อนของการจัดตารางการทำงานพยาบาลในแผนกฉุกเฉิน จะทำให้เราสามารถพัฒนาคุณภาพของตารางการทำงานพยาบาลโดยใช้สมการทางคณิตศาสตร์ควบคู่ไปกับวิธีการฮิวริสติกได้อย่างมาก เมื่อเทียบกับการจัดตารางด้วยมือ นอกจากนี้ยังสามารถประยุกต์วิธีการจัดตารางให้สามารถใช้กับแผนกอื่นๆในโรงพยาบาลง่ายขึ้น เนื่องจากแผนกอื่นๆในโรงพยาบาลมักมีความซับซ้อนน้อยกว่า สำหรับการสร้างตารางการทำงานพยาบาลของแผนกฉุกเฉินในงานวิจัยนี้ จะอ้างอิงข้อมูลจากแผนกฉุกเฉินของโรงพยาบาลเอกชนขนาดใหญ่แห่งหนึ่งในประเทศไทย

ทั้งนี้ข้อมูลพยาบาลของโรงพยาบาลกรณีศึกษามีข้อจำกัดด้านตำแหน่งของพยาบาลที่มีหลายระดับ ซึ่งขึ้นอยู่กับประสบการณ์ และความสามารถของพยาบาล เพื่อให้ผู้ป่วยผู้เข้ารับบริการได้รับการดูแลที่ถูกต้องเหมาะสม และปลอดภัย ตามนโยบายการบริหารจัดการกำลัง งานวิจัยนี้จะแบ่งระดับของพยาบาลออกเป็น 4 ตำแหน่ง ได้แก่

- หัวหน้าพยาบาล : ปฏิบัติหน้าที่พิจารณาจัดตารางการทำงานพยาบาล และตัดเวรกรณียอดผู้เข้ารับการบริการน้อย หรือเรียกกำลังเสริมเมื่อมีผู้เข้ารับบริการมาก ประสบการณ์ในการทำงาน 7 ปีขึ้นไป
- พยาบาลวิชาชีพระดับปฏิบัติการ : ปฏิบัติหน้าที่ตัดสินใจหรือแก้ปัญหาที่ยุ่งยากซับซ้อน หรือภาวะวิกฤตตามมาตรฐานวิชาชีพ ในบางกะการทำงานหากไม่มีหัวหน้าพยาบาลเข้าในกะนั้นๆ พยาบาลวิชาชีพระดับปฏิบัติการจะสามารถทำหน้าที่ตัดสินใจแทนได้ ประสบการณ์ในการทำงาน 0-7 ปี
- ผู้ช่วยพยาบาล : คัดกรองผู้ป่วยเบื้องต้น ตรวจวัดสัญญาณชีพ ประสานงานกับทีมแพทย์ และพยาบาล
- พนักงานฉุกเฉินการแพทย์ : ปฏิบัติการนอกพื้นที่ พร้อมประสานงานกับแผนกฉุกเฉิน สามารถปฐมพยาบาล และหัตถการเบื้องต้นเพื่อเตรียมการส่งผู้ป่วยเข้ารักษาในโรงพยาบาล

สำหรับรูปแบบกะการทำงานของแผนกฉุกเฉิน จะถูกแบ่งออกเป็น 6 กะ กะละ 8 – 12 ชั่วโมง ได้แก่ กะเช้า กะบ่าย กะดึก กะDay กะNight และกะ11 โดยกะเช้าจะเริ่มงานตั้งแต่เวลา 07:00น. - 15:00น. กะบ่ายจะเริ่มงานตั้งแต่เวลา 15:00น. – 23:00น. กะดึกจะเริ่มงานตั้งแต่เวลา 23:00น. – 07:00น. กะDayจะเริ่มงานตั้งแต่เวลา

07:00น. – 19:00น. กะNightจะเริ่มงานตั้งแต่เวลา 19:00น. – 07:00น. และกะสุดท้ายกะ11จะเริ่มงานตั้งแต่เวลา 11:00น. – 23:00น. โดยแต่ละกะจะครอบคลุมช่วงเวลาการทำงานเป็นไปตามรูปที่ 1-3

รูปที่ 1-3 ชั่วโมงการทำงานของแต่ละชนิดกะการทำงาน

ถึงแม้ว่าในแผนกฉุกเฉินจะมีชนิดของกะการทำงานมากถึง 6 ชนิด แต่ในการจัดอัตรากำลังฝ่ายการพยาบาลจะจัดให้มีบุคลากรด้านการพยาบาลเพียงพอสำหรับการดูแลผู้ป่วยให้ครอบคลุมเพียงแค่ 3 กะ คือ กะเช้า กะบ่าย และกะดึก (กะการทำงานละ 8 ชั่วโมง) ซึ่งการแบ่งเป็น 3 กะนั้นจะทำให้ง่ายต่อการตรวจสอบกำลังคน และง่ายหากมีเหตุการณ์ฉุกเฉินที่มีความจำเป็นจะต้องจัดตารางการทำงานใหม่ ในจุดนี้ก็จะมีความซับซ้อนในการจัดตารางเพิ่มขึ้นหากสังเกตในรูปที่ 1-3 กะDay จะครอบคลุมกะเช้า กะNight จะครอบคลุมกะดึก แต่ถ้ามีการเข้าทั้งกะDay และกะNight ภายในวันเดียวกันจะครอบคลุมทั้งกะเช้า กะบ่าย และกะดึก แม้ว่ากะการทำงานทั้งสองจะเป็นกะการทำงานที่มีระยะเวลาการทำงานที่ยาวนานถึง 12 ชั่วโมง แต่จะมีข้อดีที่สามารถใช้พยาบาลสองคนครอบคลุมการทำงานได้ตลอดวันช่วยให้แผนกฉุกเฉินมีกำลังคนในการรองรับผู้ป่วยได้อย่างเพียงพอ แต่จะต้องไม่ลืมคำนึงถึงการละเมิดกฎการทำงานด้วย นอกจากนี้ยังพบว่ามีกะทำงาน 11 ที่ครอบคลุมการทำงานช่วยเช้า 4 ชั่วโมง คือ ช่วงเวลา 11:00 น.- 15:00 น.และช่วงบ่ายทั้ง 8 ชั่วโมง สำหรับจุดประสงค์ในการมีกะ11 เกิดจากการทำงานของแผนกฉุกเฉินมักจะมีผู้ป่วยเข้ารับการรักษาหนาแน่นในช่วง 11:00 น. เป็นต้นไปตามสถิติของโรงพยาบาล จึงมีความจำเป็นที่จะต้องมีกะ11 เพื่อรับมือกับผู้ป่วยที่มากขึ้นในช่วงเวลานั้นๆ โรงพยาบาลจึงมีการทดลองจำนวนความต้องการพยาบาลแต่ละตำแหน่ง เพื่อหาความเหมาะสมจากปริมาณคนไข้ที่เข้าใช้บริการจึงได้มาเป็นจำนวนพยาบาลที่ต้องการในกะต่างๆ โดยจำนวนพยาบาลที่ต้องการในแต่ละกะจะเป็นไปตามตารางที่ 1-2 และตารางที่ 1-3

ตารางที่ 1-2 จำนวนพยาบาลที่ต้องการในแต่ละกะการทำงานของวันจันทร์-ศุกร์ (หน่วย : คน)

ตำแหน่ง	กะเช้า	กะบ่าย	กะดึก
หัวหน้าพยาบาล : HD	1	0	0
พยาบาลวิชาชีพ : RN	4	3	2
ผู้ช่วยพยาบาล : PN	1	1	1
พนักงานฉุกเฉินการแพทย์ : EMT	2	2	1

ตารางที่ 1-3 จำนวนพยาบาลที่ต้องการในแต่ละกะการทำงานของวันเสาร์-อาทิตย์ (หน่วย : คน)

ตำแหน่ง	กะเช้า	กะบ่าย	กะดึก
หัวหน้าพยาบาล : HD	0	0	0
พยาบาลวิชาชีพ : RN	4	3	2
ผู้ช่วยพยาบาล : PN	1	1	1
พนักงานฉุกเฉินการแพทย์ : EMT	2	2	1

ตารางที่ 1-2 และตารางที่ 1-3 แสดงจำนวนพยาบาลแต่ละตำแหน่งที่แผนกฉุกเฉินมีความต้องการขั้นต่ำ โดยหัวหน้าพยาบาลจะเข้ากะเช้าของวันจันทร์ - ศุกร์ ในวันเสาร์ - อาทิตย์หัวหน้าพยาบาลจะไม่ได้รับมอบหมายให้เข้ากะการทำงานแต่จะเป็นกะการทำงานแบบเผื่อเรียก หรือที่รู้จักกันในชื่อว่า “เวรตาม” สำหรับพยาบาลในตำแหน่งอื่นๆ จะมีความต้องการในทุกวันเท่ากันทั้งหมด อย่างเช่น พยาบาลวิชาชีพระดับปฏิบัติการจะต้องการพยาบาล ในกะเช้า 4 คน กะบ่าย 3 คน และกะดึก 2 คน ในทุกๆวันของสัปดาห์ ส่วนผู้ช่วยพยาบาลมีความต้องการพยาบาล ในกะเช้า 1 คน กะบ่าย 1 คน และกะดึก 1 คน ในทุกๆวันของสัปดาห์ แต่สำหรับในกะ Day กะNight และ กะ11 จะไม่ได้ระบุเป็นความต้องการพยาบาลในกะเหล่านี้ เนื่องจากเป็นกะการทำงานที่สามารถเลือกใช้เพื่อทดแทนการทำงานของกะเช้า กะบ่าย และกะดึก ได้ตามความเหมาะสม

การจัดตารางการทำงานพยาบาลนอกจากจะต้องจัดอัตรากำลังของพยาบาลให้มีความเพียงพอ โรงพยาบาลจะต้องคำนึงถึงคุณภาพชีวิตในการทำงานของพยาบาลด้วย อาจมีการกำหนดแนวทางร่วมกับพยาบาลในการดำเนินการจัดตารางการทำงานพยาบาล เพื่อให้พยาบาลได้เตรียมตัว และได้รับภาระในการทำงานได้อย่างเหมาะสม สำหรับแนวทางการจัดตารางการทำงานของพยาบาลจะมีรายละเอียดดังนี้

- หน่วยงานจัดตารางเวรให้แล้วเสร็จ ภายในวันที่ 25 ของทุกเดือน เพื่อให้พยาบาลได้มีเวลาในการเตรียมตัว และหากเกิดกรณีฉุกเฉิน เช่น มีพยาบาลคนใดได้รับเชื้อโควิด-19 จะได้มีเวลาในการจัดตารางการทำงานใหม่ได้ทันเวลา
- จัดกลุ่มพนักงานตามลักษณะงานเดียวกัน (แยกตามตำแหน่งหน้าที่ หากตำแหน่งใดมีกำลังคนไม่เพียงพอสามารถทดแทนได้ด้วยตำแหน่งที่สูงกว่าตามดุลยพินิจของหัวหน้าพยาบาล)
- ระบุเงื่อนไขในการจัดเวรของแต่ละวัน เช่น ต้องมีพนักงานขึ้นเวรเช้า 4 คน เวรบ่าย 3 คน และเวรดึก 2 คน เป็นต้น
- การจัดตารางการทำงานของพยาบาลต้องคำนึงถึงชั่วโมงรวมของการทำงาน คือ ทำงานได้ไม่เกิน 12 ชั่วโมง ใน 1 วัน หรือไม่ควรเกิน 48 ชั่วโมง ใน 1 สัปดาห์ (อ้างอิงจากประกาศสภาพยาบาล วันที่ 17 เมษายน 2560)

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

- ถ้าจำเป็นที่ต้องมีการทำงานเกินกว่าชั่วโมงที่กำหนดจะต้องมีแนวปฏิบัติที่ชัดเจนในการสนับสนุนให้พยาบาลมีเวลาพักผ่อนอย่างเพียงพอ และควรจัดให้มีเวลาพักระหว่างการปฏิบัติงานในแต่ละเวรอย่างเหมาะสม (อ้างอิงจากประกาศสภาพยาบาล วันที่ 17 เมษายน 2560)
- หลีกเลี่ยงการจัดตารางการทำงานที่จะทำให้พยาบาลต้องทำงาน 12ชั่วโมงต่อวัน ติดกันเกิน 3 วันใน 1 สัปดาห์
- ควรจัดให้มีกะ 8 ชั่วโมง (กะเช้า กะบ่าย กะดึก) สลับกับกะ 12 ชั่วโมง (กะDay กะNight กะ11) หากเป็นไปได้
- หลีกเลี่ยงการจัดกะการทำงานดึก (กะดึก กะNight) ติดต่อกัน 2 วัน
- หลีกเลี่ยงการจัดกะการทำงานดึก (กะดึก กะNight) ติดกับกะบ่าย
- หลีกเลี่ยงการจัดกะการทำงานบ่าย ติดกับกะเช้า หรือกะDay
- หยุดติดต่อกันได้ไม่เกิน 5 วันใน 1 เดือน หากเป็นการขอใช้วันหยุดร่วมด้วย สามารถขอหยุดได้เฉพาะในเดือนที่มีความจำเป็นเท่านั้น

สำหรับค่าจ้างของพยาบาล จะถูกแบ่งออกเป็น 2 ส่วน ประกอบด้วย 1. ค่าจ้างที่เป็นฐานเงินเดือน ซึ่งขึ้นอยู่กับตำแหน่ง และประสบการณ์ (ในงานวิจัยนี้จะไม่มีการนำฐานเงินเดือนของพยาบาลเข้ามาเป็นส่วนในการตัดสินใจ) 2. ค่าจ้างพิเศษที่ได้จากการเข้ากะการทำงานที่นอกเหนือจากกะเช้า โดยค่าจ้างพิเศษของพยาบาลแต่ละตำแหน่งจะแสดงดังตารางที่ 1-4

ตารางที่ 1-4 ค่าจ้างพิเศษเมื่อพยาบาลแต่ละตำแหน่งเข้ากะการทำงาน (หน่วย : บาท)

ตำแหน่ง	กะเช้า	กะบ่าย	กะดึก	กะDay	กะNight	กะ11
หัวหน้าพยาบาล : HD	0	0	0	0	0	0
พยาบาลวิชาชีพ : RN	0	280	330	140	471	280
ผู้ช่วยพยาบาล : PN	0	100	100	50	150	100
พนักงานฉุกเฉินการแพทย์ : EMT	0	75	75	38	113	75

จะสังเกตได้ว่า ค่าจ้างพิเศษที่พยาบาลได้รับเมื่อเข้ากะที่แตกต่างกัน ในขั้นตอนการจัดตารางการทำงาน มีความสำคัญอย่างยิ่งที่พยาบาลในตำแหน่งเดียวกันควรได้รับมอบหมายกะการทำงานแต่ละชนิดที่เท่าเทียมกันเพื่อให้ลดความเลื่อมล้ำของค่าจ้างพิเศษ

ในส่วนของการดำเนินงานวิจัย ผู้วิจัยได้ทำการรวบรวมข้อจำกัดจากพยาบาลที่ทำงานอยู่ในแผนกฉุกเฉินเพื่อนำมาสร้างเป็นแบบจำลองทางคณิตศาสตร์ โดยมุ่งหวังให้แบบจำลองดังกล่าวเป็นต้นแบบในการสร้างระบบสนับสนุนการตัดสินใจช่วยจัดตารางการทำงานของพยาบาลที่เหมาะสมสำหรับการใช้งานในอนาคต ทั้งนี้ ผู้วิจัยคาด

ว่า ด้วยแบบจำลองทางคณิตศาสตร์ที่สร้างขึ้น ผู้จัดตารางจะสามารถสร้างตารางการทำงานที่มีประสิทธิภาพ และมีความเท่าเทียมกันมากขึ้น ทั้งในมุมของจำนวนกะการทำงาน และวันปฏิบัติงานที่ได้รับมอบหมาย ภายใต้ระยะเวลาการจัดตารางที่ลดลง

สำหรับวัตถุประสงค์ของการจัดตาราง จะมาจากการเก็บข้อมูลจากพยาบาลผู้มีประสบการณ์ ซึ่งสามารถสรุปวัตถุประสงค์สำหรับสร้างตารางการทำงานของพยาบาล ได้เป็น 3 วัตถุประสงค์ ได้แก่

- วัตถุประสงค์ที่ 1 เกิดค่าใช้จ่ายในการจ้างพยาบาลที่เหมาะสมที่สุด คือ จ้างพยาบาลให้เพียงพอกับความต้องการ เพื่อลดปัญหาพยาบาลในกะการทำงานขาด หรือเกิน
- วัตถุประสงค์ที่ 2 มีความเท่าเทียมในการได้รับค่าจ้าง คือ พยาบาลทุกคนจะได้รับค่าจ้างในการทำงานใกล้เคียงกันมากที่สุดเท่าที่เป็นไปได้ (ภายในตำแหน่งเดียวกัน)
- วัตถุประสงค์ที่ 3 เกิดการละเมิดข้อจำกัดให้น้อยที่สุดเท่าที่เป็นไปได้ คือ เกิดการละเมิดข้อจำกัดรองให้น้อยที่สุดเท่าที่เป็นไปได้ และในส่วนของข้อจำกัดหลักจะไม่เกิดการละเมิดเกิดขึ้น เนื่องจากไม่สามารถละเมิดข้อจำกัดหลักได้

เพื่อให้สามารถตอบสนองทั้ง 3 วัตถุประสงค์ของการสร้างตารางการทำงานของพยาบาล เราจึงกำหนดให้แบบจำลองทางคณิตศาสตร์ในงานวิจัยนี้ให้เป็นแบบหลายวัตถุประสงค์ (Multi-objective mathematical model) (Chang et al., 2015) และใช้เทคนิคการแก้ปัญหาแบบ Preemptive goal programming (PGP) ที่สามารถจัดการกับปัญหาแบบหลายวัตถุประสงค์ที่มีความขัดแย้งกันได้ โดยให้ความสำคัญกับวัตถุประสงค์ที่สนใจที่สุดก่อน แล้วจึงพิจารณาวัตถุประสงค์อื่นๆตามลำดับความสำคัญ (Rerkjirattika et al., 2020)

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

อย่างไรก็ดีปัญหาการจัดตารางการทำงานของพยาบาล หรือ Nurse Scheduling Problem (NSP) เป็นปัญหาเอ็นพีแบบยาก (NP Hard) (Haspeslagh et al., 2014) เนื่องจากการจัดตารางการทำงานของพยาบาลต้องพิจารณาปัจจัยหลายอย่าง เช่น จำนวนพยาบาลที่มีอยู่ภายในแผนก จำนวนความต้องการพยาบาลในแต่ละกะการทำงาน และข้อจำกัดด้านความสามารถที่มีไม่เท่ากัน ทำให้เกิดความน่าจะเป็นของการจัดตารางจำนวนมากจนความสามารถในการประมวลผลของคอมพิวเตอร์ไม่เพียงพอต่อการแก้ปัญหาด้วยวิธีแม่นตรง (Exact method) ภายในกรอบเวลาที่กำหนด เช่นเดียวกับแบบจำลองทางคณิตศาสตร์ที่ใช้ในการปัญหาการจัดตารางการทำงานพยาบาล ให้สามารถตอบสนองกับวัตถุประสงค์หลัก 3 ข้อ ได้แก่ 1. เกิดค่าใช้จ่ายในการจ้างพยาบาลที่เหมาะสมที่สุด 2. มีความเท่าเทียมในการได้รับค่าจ้าง 3. เกิดการละเมิดข้อจำกัดให้น้อยที่สุดเท่าที่เป็นไปได้ อีกทั้งต้องไม่ละเมิดข้อจำกัดหลักที่กำหนดไว้ เราจะพบว่าสมการทางคณิตศาสตร์สามารถสร้างตารางการทำงานของพยาบาลได้เพียงปัญหาขนาดเล็กเท่านั้น ทำให้ไม่สามารถสร้างตารางการทำงานพยาบาลสำหรับปัญหาขนาดใหญ่ ให้กับพยาบาลทุกคนในแผนกด้วยวิธีแม่นตรงได้ แต่กระนั้นตารางการทำงานของพยาบาลกลุ่มตัวอย่างที่ถูกสร้างจากแบบจำลองทางคณิตศาสตร์ไม่ใช่สิ่งที่สูญเปล่า เนื่องจากผลเฉลยแม่นตรง จะถูกใช้เป็นตัวเปรียบเทียบประสิทธิภาพ

กับวิธีการฮิวริสติก (Heuristic) ที่เป็นเทคนิคที่ได้รับความนิยมในการแก้ปัญหาเอ็นพีแบบยาก (NP Hard) และเป็นเทคนิคที่งานวิจัยนี้จะใช้ เพื่อหาคำตอบกับปัญหาการจัดตารางการทำงานขนาดเล็ก และขนาดใหญ่ โดยคำตอบที่ได้จากปัญหาขนาดเล็กจะต้องมีความใกล้เคียงกับผลเฉลยแม่นตรง ที่อยู่ในช่วงที่ยอมรับได้ ภายใต้เวลาในการหาคำตอบที่สั้นกว่าวิธีแบบจำลองทางคณิตศาสตร์ (Wang et al., 2013) แล้วจึงขยายขอบเขตในการแก้ปัญหาขนาดใหญ่ด้วยการเพิ่มจำนวนพยาบาลทั้งหมดภายในแผนก

ในการทำงานของฮิวริสติก ในงานวิจัยนี้จะเลือกใช้อัลกอริทึมชื่อ Adaptive Large Neighborhood Search (ALNS) ที่ทำงานโดยอาศัยหลักการการสำรวจในพื้นที่วัตถุประสงค์ (Objective space) โดยการปรับปรุงคำตอบจะเกิดขึ้นจากการตัวทำลายคำตอบ (Destroy operator) ตัวซ่อมแซมคำตอบ (Repair operator) และประเมินคุณภาพของคำตอบจากค่าวัตถุประสงค์เพื่อปรับเกณฑ์ในการเลือกใช้ตัวดำเนินการ (Operator)

ทว่าการนำตารางพยาบาลไปใช้งานจริงนั้นเราไม่อาจทราบได้เลยว่าตารางการทำงานควรสร้างตารางให้ตอบสนองกับวัตถุประสงค์ในมิติใด จึงจะสามารถรับมือกับผู้เข้าใช้บริการ ณ ขณะนั้นได้อย่างเหมาะสม และเกิดความพึงพอใจในการทำงานของพยาบาล จึงเป็นเหตุให้ตารางการทำงานพยาบาลควรมีตัวเลือกที่หลากหลายแตกต่างกันในด้านความสำคัญของวัตถุประสงค์ เพื่อให้พยาบาลสามารถตัดสินใจเลือกใช้ได้ ซึ่งการที่จะสร้างตารางที่มีความหลากหลายในแต่ละวัตถุประสงค์พร้อมๆกันทีละหลายตาราง คือ การทำ Pareto front แต่การจะสร้างตารางการทำงานจำนวนหลายตารางด้วยวิธี ALNS เพียงอย่างเดียวนั้นเป็นเรื่องที่ทำได้ยาก และอาจจะควบคุมการกระจายตัวของคำตอบได้ไม่เหมาะสม เพื่อลดความซับซ้อนในการสร้างตารางการทำงานพยาบาลให้เป็น Pareto front จะมีการนำอีกเทคนิคที่มีชื่อว่า Fast Non-Dominated Sorting (FNDS) ที่เป็นส่วนหนึ่งในการทำงานของ Non-dominated Sorting Genetic Algorithm III (NSGA-III) ที่ไม่เพียงจะง่ายต่อการสร้างตารางการทำงานพยาบาลพร้อมๆกันหลายตาราง แต่ยังทำให้ตารางการทำงานพยาบาลมีค่าแต่ละวัตถุประสงค์ที่กระจายตัวอยู่ในพื้นที่วัตถุประสงค์ (Objective space) ได้อย่างสม่ำเสมอ

Chulalongkorn University

1.2 วัตถุประสงค์ของงานวิจัย

1. เพื่อพัฒนาประสิทธิภาพตารางการทำงานพยาบาลในแผนกฉุกเฉิน
    1.1. เกิดค่าใช้จ่ายที่เหมาะสมที่สุด
    1.2. สร้างความเท่าเทียมในการได้รับค่าจ้าง ให้มากที่สุดเท่าที่จะเป็นไปได้
    1.3. เกิดการละเมิดข้อจำกัดในการทำงานให้น้อยที่สุดเท่าที่เป็นไปได้
2. เพื่อสร้างตารางการทำงานพยาบาลที่มีความหลากหลายในแต่ละวัตถุประสงค์ โดยการใช้ Pareto Front
3. สร้างโปรแกรมการจัดตารางการทำงานพยาบาลด้วยวิธีฮิวริสติก เพื่อทดแทนวิธีการจัดตารางด้วยตนเอง

1.3 ขอบเขตของงานวิจัย

1 ตารางการทำงานของพยาบาลจะถูกใช้สำหรับโรงพยาบาลเอกชนขนาดใหญ่ทางภาคตะวันออกเฉียงเหนือแห่งหนึ่งในประเทศไทย

2 ข้อมูล และข้อจำกัดเป็นของพยาบาลที่อยู่ในแผนกฉุกเฉิน

3 ตารางที่ถูกสร้างขึ้นจะเป็นตารางการทำงานของพยาบาลทุกคนภายในแผนกฉุกเฉิน

4 การจัดตารางล่วงหน้าจะไม่นำตารางการทำงานก่อนหน้าเข้ามาเป็นส่วนหนึ่งในการตัดสินใจ

1.4 ประโยชน์ที่ได้รับจากงานวิจัย

1 ลดภาระการจัดตารางการทำงานพยาบาลของหัวหน้าพยาบาล

2 มีทางเลือกในการจัดตารางการทำงานพยาบาลเพิ่มมากขึ้น

3 ควบคุมค่าใช้จ่ายในการจ้างพยาบาลได้อย่างเหมาะสม อีกทั้งยังเพิ่มความเท่าเทียมในการได้รับค่าจ้างและ เพิ่มความพึงพอใจในการทำงาน

4 ลดเวลาการจัดตารางการทำงานพยาบาล และสามารถแก้ไข ปรับเปลี่ยนให้สามารถรองรับกับเหตุการณ์ที่ไม่คาดคิดได้อย่างทันท่วงที

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

บทที่ 2
บททบทวนวรรณกรรม

2.1 ปัญหาการจัดตารางการทำงานพยาบาล (Nurse Scheduling Problem)

การจัดตารางการทำงานของบุคลากรเป็นงานทั่วไปสำหรับองค์กร หรือบริษัท ซึ่งถูกนำมาใช้กับหลากหลายอุตสาหกรรม เนื่องจากเป็นพื้นฐานของระบบการจัดการแรงงานให้เกิดประสิทธิภาพ ทั้งนี้การจัดตารางทำงานของบุคลากรหมายถึง กระบวนการจัดสรรทรัพยากรมนุษย์ เพื่อให้องค์กรสามารถตอบสนองต่อความต้องการสำหรับการผลิต หรือบริการของตน โดยปกติจะเป็นงานที่ซับซ้อนมากจึงเป็นเรื่องยากที่จะสร้างตารางทำงานของบุคลากรให้สามารถรองรับกับความต้องการในทุกด้านได้ โดยยังสามารถสร้างความพึงพอใจแก่บุคลากรได้ (Mansini et al., 2023)

จุดเริ่มต้นของการแก้ปัญหาการจัดตารางทำงานของบุคลากรเกิดในช่วงปีคริสต์ศักราชที่ 1954 ซึ่งใช้สำหรับการจัดการงานแสดงสินค้า ที่มีความจำเป็นต้องใช้พนักงานหลายตำแหน่ง รวมถึงจัดสรรพนักงานแต่ละคนภายใต้ข้อจำกัดที่แตกต่างกัน เพื่อให้งานแสดงสินค้าสามารถรองรับลูกค้าที่จะมาร่วมงานได้อย่างมีประสิทธิภาพ (Dantzig et al., 1954)

ปัญหาการจัดตารางทำงานของบุคลากรสามารถแบ่งประเภทได้เป็นสองประเภทหลักๆ คือ การจัดตารางภายในแผนก และการจัดตารางข้ามแผนก โดยการจัดตารางภายในแผนกจะเป็นการบริหารตารางการทำงานของบุคลากรภายในแผนกเท่านั้นจะไม่มีการจัดตารางร่วมกับบุคลากรภายนอกแผนก หรือบุคลากรภายในแผนกจะไม่ถูกกำหนดให้ทำงานนอกแผนก ซึ่งการกำหนดขอบเขตการทำงานจะเป็นอิสระจากความต้องการของแผนกอื่นๆ แต่ในประเภทที่สอง การจัดตารางข้ามแผนกเราสามารถแลกเปลี่ยนบุคลากรระหว่างแผนกได้ ซึ่งเป็นการใช้ทรัพยากรบุคคลร่วมกันของแต่ละแผนก (Bard et al., 2008) ซึ่งมีความคล้ายคลึงกับปัญหาการจัดตารางพยาบาล (Nurse Scheduling Problem) ที่มุ่งเน้นการมอบหมายงานให้พยาบาลเข้ากะการทำงานเพื่อรองรับกับการเข้ารับบริการของคนไข้ ภายใต้เป้าหมายการลดต้นทุนรวมของการจ้างพยาบาล และเพิ่มความพึงพอใจของพยาบาลให้ได้สูงที่สุด โดยต้องคำนึงถึงกฎเกณฑ์ต่างๆ (Svirsko et al., 2019) ซึ่งการจัดตารางข้ามแผนกจะมีความคล้ายคลึงกับงานวิจัยการจัดตารางการทำงานพยาบาลที่พยาบาลหนึ่งคนสามารถมีทักษะที่หลากหลาย จึงสามารถทำงานทดแทนในบางตำแหน่งที่เกิดความขาดแคลนบุคลากรได้

การจัดตารางการทำงานพยาบาลเป็นงานที่มีองค์ประกอบซับซ้อน หากแต่มีวัตถุประสงค์หลักเพื่อวางแผนให้มีจำนวนพยาบาลเพียงพอต่อการเข้ารับบริการของผู้เข้ารับการรักษา ซึ่งการบรรลุวัตถุประสงค์ดังกล่าว เราจำเป็นต้องคำนึงถึงข้อจำกัดต่างๆ ในการทำงานควบคู่ไปด้วย เช่น การไม่กำหนดให้พยาบาลเข้าทำงานในวันที่ขอลาหยุด หรือการไม่กำหนดให้พยาบาลเข้าทำงานในกะที่ทับซ้อนกัน (Glass et al., 2010)

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

เมื่อมีการมอบหมายงานจะต้องไม่ลืมคำนึงถึงความสามารถของพยาบาลที่เป็นส่วนหนึ่งของการตัดสินใจ เนื่องจากพยาบาลแต่ละคนไม่อาจสามารถครอบคลุมการทำงานได้ทั้งหมด ในปัญหาการจัดตารางการทำงานของพยาบาลจะต้องพิจารณาประสบการณ์ ความสามารถ และระดับของพยาบาล ซึ่งอาจเรียกแทนระดับเป็นชื่อตำแหน่ง หรือรหัสเพื่อจำแนกความสามารถของพยาบาล โดยการแบ่งระดับในแต่ละโรงพยาบาลอาจถูกกำหนดโดยสภาการพยาบาล หรือแบ่งตามมาตรฐานระบบการจัดการของแต่ละโรงพยาบาล (Kellogg et al., 2007 ; Legrain et al., 2015)

ด้วยเหตุดังกล่าวการจัดตารางพยาบาลเราจำเป็นต้องทำให้มั่นใจว่าตารางการทำงานที่จัดออกมามีความถูกต้อง และเหมาะสม เพราะหากตารางทำงานของพยาบาลถูกจัดอย่างไม่เหมาะสม ไม่เพียงแต่จะส่งผลกระทบเชิงลบต่อประสิทธิภาพการทำงานของพยาบาล และประสบการณ์เข้ารับการรักษาของผู้ป่วย แต่ยังส่งผลต่อความพึงพอใจของบุคลากรอีกด้วย (Franz et al., 1989) การรักษาสมดุลให้พยาบาลมีวันหยุดพักผ่อนที่เพียงพอ และมีปริมาณกะการทำงานที่เหมาะสมจึงเป็นเรื่องสำคัญ สำหรับปัญหาการจัดตารางการทำงานพยาบาลที่ผู้จัดตารางการทำงานทุกคนไม่ควรมองข้าม

2.2 การประยุกต์ใช้แบบจำลองทางคณิตศาสตร์ (Mathematical Model)

เพื่อให้สามารถแก้ปัญหาการจัดตารางการทำงานพยาบาลได้ง่ายขึ้นจึงมีการนำแบบจำลองทางคณิตศาสตร์เข้ามาประยุกต์ใช้ในการแก้ปัญหาดังกล่าวให้เป็นไปอย่างมีระเบียบแบบแผน ซึ่งเป็นประโยชน์อย่างยิ่งในแผนกที่มีความซับซ้อนในการจัดตาราง เช่น แผนกฉุกเฉินที่ผู้เข้ารับการรักษาต้องได้รับการรักษาอย่างเร่งด่วน และไม่สามารถระบุได้ว่า ในวันหนึ่งๆ จะมีผู้เข้ารับการรักษาจำนวนเท่าใด ซึ่งในกรณีดังกล่าว การเตรียมความพร้อมในการจัดการกำลังคน หรือการจัดตารางพยาบาลถือว่าเป็นงานที่มีความสำคัญเป็นอย่างยิ่ง (El Adoly et al., 2018) เนื่องจากแบบจำลองทางคณิตศาสตร์จะมีความยืดหยุ่นในการพิจารณาข้อจำกัดที่หลากหลาย ปัญหาการจัดตารางทำงานของบุคลากรทางการแพทย์ส่วนใหญ่จึงมักถูกออกแบบมาให้สามารถแก้ปัญหาได้ด้วยแบบจำลองทางคณิตศาสตร์ ปัญหาดังกล่าวยังได้รับความสนใจ และถูกดำเนินการวิจัยอย่างต่อเนื่องมายาวนาน หากแต่ส่วนใหญ่จะเป็นการวิจัยที่อยู่ในระดับสถาบันการศึกษา และมีงานวิจัยเพียงส่วนน้อย ประมาณ 30% เท่านั้น ที่ถูกนำไปประยุกต์ใช้งานจริง (Smet et al., 2014) โรงพยาบาลส่วนใหญ่จึงยังจำเป็นต้องดำเนินการจัดตารางด้วยตนเองอยู่ ส่งผลทำให้ไม่สามารถจัดการค่าใช้จ่ายในการจัดจ้างพยาบาลได้อย่างเหมาะสม และในบางครั้งการจัดตารางด้วยตนเองยังเกิดความผิดพลาดได้ง่ายอีกด้วย

นอกจากแบบจำลองทางคณิตศาสตร์จะสามารถเพิ่มประสิทธิภาพในการจัดตารางการทำงาน เรายังสามารถประยุกต์ใช้แบบจำลองทางคณิตศาสตร์ในการควบคุมค่าใช้จ่ายได้อย่างเหมาะสม ซึ่งค่าใช้จ่ายที่เกิดขึ้นจากการจ้างงานพยาบาลนั้น คิดเป็นสัดส่วนกว่า 44% ของค่าใช้จ่ายทั้งหมดของค่าจ้างบุคลากรทางการแพทย์ การปรับปรุงตารางปฏิบัติงานของพยาบาลจะช่วยให้โรงพยาบาลสามารถลดค่าใช้จ่ายที่เกินความจำเป็นในส่วนนี้ลงได้ นอกจากนี้ การจัดตารางการทำงานพยาบาลที่มีประสิทธิภาพยังถือเป็นการเตรียมรับมือกับปริมาณความต้องการของคนไข้ที่มีแนวโน้มเพิ่มสูงขึ้นในภายภาคหน้าอย่างยั่งยืนอีกด้วย (Vanden et al., 2002)

จุฬาลงกรณ์มหาวิทยาลัย
CHULALONGKORN UNIVERSITY

จากการศึกษาของ Nobil et al. (2022) ที่ได้ทำการพัฒนาการจัดตารางการทำงานของพยาบาลจากการประยุกต์ใช้แบบจำลองโปรแกรมเชิงเส้นจำนวนเต็มแบบผสม หรือMixed Integer Linear Programming (MILP) เพื่อลดต้นทุนในการจ้างพยาบาลของโรงพยาบาลแห่งหนึ่งในประเทศอิหร่าน โดยใช้ข้อมูล ข้อจำกัดในการทำงานของพยาบาลที่เป็นข้อมูลจริง พบว่าเราสามารถลดต้นทุนการจ้างพยาบาลลงได้กว่า 10% ในขณะที่สามารถจัดตารางการทำงานของพยาบาลให้รองรับกับการเข้าใช้บริการของคนไข้ได้อย่างมีประสิทธิภาพ

Burke et al. (2006) พบว่าเราสามารถกำหนดให้แบบจำลองทางคณิตศาสตร์สะท้อนปัญหาในโลกชีวิตจริงมากขึ้น เพื่อรับมือกับปัญหาที่อาจเกิดขึ้นได้ในบางสถานการณ์ที่การทำตามกฎระเบียบอาจไม่สามารถรับมือกับเหตุการณ์ที่ไม่คาดคิด ซึ่งในสถานการณ์ดังกล่าวเราอาจแบ่งข้อจำกัดออกเป็นสองแบบได้แก่

1. ข้อจำกัดหลัก : เป็นข้อจำกัดที่ไม่สามารถละเมิดได้ เช่น การมอบหมายจำนวนพยาบาลที่จะต้องเข้าในกะการทำงานในแต่ละวัน ไม่สามารถให้พยาบาลคนใดคนหนึ่งเข้าในกะเดียวกัน ในวันเดียวกันได้มากกว่าหนึ่งครั้ง
2. ข้อจำกัดรอง : เป็นข้อจำกัดที่อาจถูกละเมิดได้เป็นการผ่อนปรนข้อจำกัดเพื่อให้สามารถหาคำตอบได้ หรือในบางกรณีข้อจำกัดรองจะเป็นการนำข้อจำกัดที่สามารถละเมิดได้หากมีความจำเป็น แต่จะมีบทลงโทษสำหรับการละเมิดข้อจำกัดในแต่ละข้อจำกัดที่แตกต่างไปขึ้นอยู่กับความรุนแรงของการละเมิดข้อจำกัด เช่น การละเมิดข้อจำกัดด้านชั่วโมงการทำงาน การละเมิดข้อจำกัดเกี่ยวกับการได้รับวันหยุดเสาร์-อาทิตย์ สำหรับข้อจำกัดรองควรมีการเก็บข้อมูลจากแหล่งที่น่าเชื่อถือ เพื่อให้ได้ค่าน้ำหนักของบทลงโทษที่มีความเหมาะสม เพราะค่าน้ำหนักเหล่านี้จะถูกนำไปเป็นส่วนหนึ่งของค่าวัตถุประสงค์ (Objective values)

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

จุดเด่นอีกหนึ่งอย่างของการแก้ปัญหาโดยการใช้แบบจำลองทางคณิตศาสตร์ คือ สามารถมีได้หลายวัตถุประสงค์ (Multi-Objective) โดยการใช้แบบจำลองทางคณิตศาสตร์แบบหลายวัตถุประสงค์ (Multi-Objective Mathematical Model) เพื่อหาคำตอบที่เหมาะสมที่สุด ซึ่งวัตถุประสงค์มีเป้าหมายเป็นไปในทิศทางเดียวกัน หรือมีความขัดแย้งกันก็ได้ ขึ้นอยู่กับปัญหาที่จะทำการหาคำตอบ แต่ที่สำคัญผู้วิจัยจะต้องกำหนดความสำคัญของแต่ละวัตถุประสงค์เพื่อเป็นตัวช่วยให้สมการทางคณิตศาสตร์สามารถหาคำตอบโดยอ้างอิงจากความสำคัญของวัตถุประสงค์ที่เรากำหนดไว้ อย่างไรก็ดีการมีหลายวัตถุประสงค์มากเกินไปอาจทำให้แบบจำลองทางคณิตศาสตร์มีความซับซ้อนมากจนเกินขีดความสามารถในการหาคำตอบได้ หากจะทำการแก้ปัญหาโดยการใช้แบบจำลองทางคณิตศาสตร์เพียงอย่างเดียว จึงควรมีเกณฑ์ในการเลือกวัตถุประสงค์ที่มีความสำคัญสำหรับปัญหานั้นๆ (Berrada et al., 1996)

อย่างไรก็ดี การใช้แบบจำลองทางคณิตศาสตร์ในการแก้ปัญหาส่วนใหญ่ มักเจอกับปัญหาที่ไม่สามารถหาคำตอบได้เมื่อข้อมูลของปัญหาเริ่มมีขนาดใหญ่ขึ้น หรือวัตถุประสงค์มีความซับซ้อนมากขึ้น อย่างเช่นปัญหาการจัดตารางทำงานของพยาบาล ซึ่งเป็นที่ทราบกันดีว่าเป็นปัญหา เอ็นพีแบบยาก (NP Hard) ที่มีความซับซ้อนในการแก้ปัญหาอย่างมาก (Haspeslagh et al., 2014) การจัดตารางทำงานของพยาบาลที่มีชุดข้อมูลที่จะต้องทำการตัดสินใจเป็นจำนวนมาก มักเกินขีดจำกัดของหน่วยประมวลผล หรือหน่วยเก็บข้อมูลหลักของคอมพิวเตอร์เต็มก่อนที่จะได้คำตอบ (Burke et al., 2004) เราจึงอาจจะต้องหาวิธีการแก้ปัญหาสำหรับการจัดตารางของพยาบาลที่เหมาะสมกว่าการใช้แบบจำลองทางคณิตศาสตร์เพียงอย่างเดียว (Huang et al., 2014)

2.3 การประยุกต์ใช้วิธีการฮิวริสติก (Heuristic)

เพื่อลดเวลาการประมวลผลของปัญหาเอ็นพีแบบยาก (NP Hard) จึงมีการนำวิธีการฮิวริสติก ที่เป็นหนึ่งในวิธีที่ได้รับความนิยมในการแก้ปัญหาเอ็นพีแบบยาก โดยเฉพาะอย่างยิ่งปัญหาการจัดตารางการทำงานของพยาบาลด้วยจุดเด่นที่มีความยืดหยุ่น ใช้เวลาในการประมวลผลน้อยกว่าการหาคำตอบจากแบบจำลองทางคณิตศาสตร์ และสามารถปรับปรุงอัลกอริทึม (Algorithm) ให้รองรับกับปัญหาที่มีความซับซ้อนได้เป็นอย่างดี อย่างเช่นงานวิจัยที่ได้นำปัญหาการจัดตารางการทำงานพยาบาลมาทดลองแก้ด้วยวิธีแบบจำลองทางคณิตศาสตร์แล้วพบว่า ไม่สามารถหาคำตอบในกรอบเวลาที่กำหนดได้เนื่องจากข้อมูลที่ป้อนเข้าไป มีการกำหนดข้อจำกัดหลักและข้อจำกัดรองที่มีความซับซ้อน Resende et al. (2004) จึงได้พัฒนาฮิวริสติกมาแก้ปัญหา ซึ่งสามารถให้คำตอบได้ใกล้เคียงกับแบบจำลองทางคณิตศาสตร์ แต่ใช้เวลาในการหาคำตอบที่น้อยกว่ามาก

แต่ถึงกระนั้นวิธีการฮิวริสติก ก็มีจุดด้อย คือ คำตอบที่ได้จากการประมวลผลอาจไม่ใช่ผลเฉลยแม่นตรง (Exact solution) ที่ให้ค่าวัตถุประสงค์ที่ดีที่สุด อย่างไรก็ตามฮิวริสติกก็สามารถถูกพัฒนาให้มีประสิทธิภาพใกล้เคียงกับแบบจำลองทางคณิตศาสตร์ได้ และสามารถสร้างคำตอบที่มีความหลากหลายภายในการประมวลผลครั้งเดียว จึงเป็นเหตุผลที่วิธีการฮิวริสติกได้รับความนิยม และเป็นวิธีการแก้ปัญหาที่ได้รับการยอมรับอย่างกว้างขวาง (Lin et al., 1993 ; Hochba et al., 1997)

ในงานวิจัยการจัดตารางการทำงานพยาบาลครั้งนี้ ได้มีการนำเทคนิค Fast Non-Dominated Sorting (FNDS) ซึ่งเป็นส่วนหนึ่งของอัลกอริทึม Non-dominated Sorting Genetic Algorithm III (NSGA-III) มาร่วมเป็นส่วนหนึ่งในการทำงาน เพื่อประยุกต์ใช้ในการสร้างคำตอบแบบ Pareto front ที่ช่วยให้คำตอบมีความเหมาะสมในแต่ละมิติของวัตถุประสงค์ที่แตกต่างกัน การทำ Pareto front ไม่เพียงแต่จะเป็นการสร้างชุดคำตอบที่มีความแตกต่างกันเป็นจำนวนหลายชุดคำตอบ แต่ในการนำไปใช้งานจะช่วยให้พยาบาลสามารถเลือกตารางการทำงานที่ตอบสนองต่อข้อจำกัด และความต้องการที่หลากหลายในเวลาเดียวกันได้ โดยผ่านการตัดสินใจของพยาบาล ณ สถานการณ์ปัจจุบัน อย่างไรก็ตาม FNDS ไม่สามารถใช้เป็นอัลกอรีทึมหลักเพียงอย่างเดียวได้ เนื่องจากไม่มีขั้นตอนการปรับปรุงคุณภาพของคำตอบ ดังนั้นเราจึงต้องประยุกต์ใช้เทคทิคอื่นควบคู่ไปด้วย ซึ่งอีกเทคนิคที่งานวิจัยนี้จะ

เลือกใช้ได้แก่ Adaptive Large Neighborhood Search (ALNS) ที่มีโครงสร้างการทำงานแบบโมดูล (Module) ทำให้สามารถเพิ่มขั้นตอน FNDS ได้ง่าย สำหรับ ALNS จะทำหน้าที่ปรับปรุงประสิทธิภาพของคำตอบให้ดีขึ้น ขณะที่ FNDS จะช่วยในการแจกแจง และกำหนดทิศทางในการปรับปรุงคำตอบ โดยการทำงานร่วมกันของทั้งสองจะช่วยให้สามารถสร้างคำตอบที่มีคุณภาพใกล้เคียงกับแบบจำลองทางคณิตศาสตร์ได้ ภายในกรอบเวลาการประมวลผลที่ลดลงอย่างมาก (Burke et al., 2012)

2.4 Fast Non-Dominated Sorting (FNDS)

FNDS เป็นส่วนหนึ่งในขั้นตอนการคัดเลือกคำตอบที่มีอยู่ใน NSGA-II และ NSGA-III โดย FNDS ของ NSGA-III มีการปรับปรุงขั้นตอนการทำงานด้วยการเพิ่มขั้นตอนการสร้าง Reference point และ Niche count แต่ยังคงมีขั้นตอนการหา Front และ Crowding distance ที่เดิมทีเป็นขั้นตอนที่อยู่ใน FNDS ของ NSGA-II ส่งผลให้ไม่เกิดการตัดสินใจคัดเลือกคำตอบโดยอาศัยการคำนวณค่า Crowding distance ที่บ่งบอกการกระจายตัวของคำตอบแค่เพียงปัจจัยเดียว แต่ยังมี Niche count ที่สามารถกำหนดทิศทางในการปรับปรุงคำตอบได้อย่างมีประสิทธิภาพ (Gu et al., 2020)

ในขั้นตอนการจัดกลุ่มของคำตอบ หรือ Front เป็นการคำนวณค่าวัตถุประสงค์ในแต่ละวัตถุประสงค์เพื่อมาเปรียบเทียบคำตอบกัน ซึ่งคำตอบที่ไม่ถูกครอบงำ (Non-dominate) โดยคำตอบอื่นเลยจะถูกจัดให้อยู่ที่ Front แรก ซึ่งคำตอบที่ถูกคัดเลือกให้อยู่ใน Front ก่อนหน้าจะไม่ถูกนำมาเปรียบเทียบกับคำตอบที่เหลือ เมื่อตรวจสอบคำตอบจนครบทุกคำตอบแล้วก็จะทำการปิด Front และเริ่มหาสมาชิกใน Front ถัดไป กระบวนการจัด Front จะดำเนินไปจนกระทั่งคำตอบทุกคำตอบได้กลุ่มของตนเอง ซึ่งวิธีการนี้ถูกคิดค้นขึ้นมาในปี ค.ศ. 2014 โดยการกำหนด M-Front เพื่อลดความซับซ้อนโดยอาศัยการเรียงลำดับแบบไม่ครอบงำ ส่งผลให้สามารถปรับปรุงประสิทธิภาพจากอัลกอริทึมดั้งเดิมให้ดีขึ้นได้ในระยะเวลาการประมวลผลที่สั้นลง (Drozdik et al., 2014 ; Fang et al., 2008)

จุฬาลงกรณ์มหาวิทยาลัย
CHULALONGKORN UNIVERSITY

โดยทั่วไปแล้วการใช้ Crowding distance มักจะนิยมใช้ใน NSGA-II หรือ NSGA-III ดังเช่น Mohan (2023) ที่มีการคำนวณค่า Crowding distance เพื่อเป็นการคัดเลือกคำตอบใน NSGA-II โดยดูจากค่าของ Crowding distance ที่อยู่ใน Front เดียวกัน คำตอบที่มีค่า Crowding distance มากแสดงว่ามีคำตอบนั้นคล้ายคลึงกับคำตอบอื่นน้อย เป็นคำตอบที่มีความหลากหลาย และมีโอกาสจะพัฒนาคำตอบให้ดีขึ้นได้ง่าย จึงมีโอกาสมากกว่าในการถูกเลือกนำไปเข้าขั้นตอน Crossover และ Mutation สำหรับสร้างประชากรรุ่นถัดไป

Reference point และ Niche count เป็นส่วนที่ถูกพัฒนาเพิ่มเติมใน FNDS ของ NSGA-III ที่จะเข้ามามีส่วนช่วย Crowding distance ในการแจกแจงชุดคำตอบได้อย่างทั่วถึง และรักษาความหลากหลายของคำตอบระหว่างที่ทำการปรับปรุงคำตอบในแต่ละรอบ Reference point เป็นจุดที่กระจายตัวอย่างสม่ำเสมอบนพื้นที่วัตถุประสงค์ (Objective space) เมื่อมีชุดคำตอบที่กระจายตัวอยู่รอบๆจุด Reference point เราจะคำนวณ

Niche count เพื่อเป็นการหาระยะห่างจากจุด Reference point ของแต่ละชุดคำตอบให้ถูกปรับปรุงไปในทิศทางที่ใกล้กับ Reference point ที่ใกล้ที่สุดของแต่ละคำตอบ (Deb et al., 2013)

2.5 Adaptive Large Neighborhood Search (ALNS)

ในปี ค.ศ.2006 Ropke et al. (2006) ได้พัฒนาวิธีการ Adaptive Large Neighborhood Search (ALNS) เพื่อแก้ปัญหาการจัดเส้นทางการขนส่ง และรับคืนสินค้าแบบมีเวลาเข้ามาเกี่ยวข้อง (Pickup and delivery problem with time windows) ซึ่ง ALNS ได้นำหลักการของ Large Neighborhood Search (LNS) ของ Shaw (1998) มาพัฒนา โดยใช้การทำลายคำตอบ และซ่อมแซมคำตอบเช่นเดียวกับ LNS หากแต่ ALNS มีการเพิ่มขั้นตอนการให้คะแนนกับตัวดำเนินการ (Operator) ที่สามารถปรับค่าได้ และนำไปทดสอบกับปัญหาที่ได้รับการปรับปรุงจากวิธี LNS พบว่าจากคำตอบทั้งหมด 338 คำตอบที่ผ่านกระบวนการปรับปรุงด้วย LNS ได้รับการปรับปรุงให้ดีขึ้นได้ถึง 227 คำตอบ หรือคิดเป็นเปอร์เซ็นต์การปรับปรุงคำตอบจาก ALNS ถึง 67% จากคำตอบทั้งหมด ผลลัพธ์ที่ได้ชี้ให้เห็นว่าการพัฒนา ALNS โดยเพิ่มขั้นตอนการให้คะแนนตัวดำเนินการ จะสามารถสร้างคำตอบให้มีประสิทธิภาพที่ดีขึ้นได้

จากการค้นคว้าทำให้เราพบว่ามีงานวิจัยที่มีความคล้ายคลึงกับปัญหาการจัดตารางการทำงานพยาบาล คืองานของ Gutjahr et al., (2023) ที่ทำการวิจัยเกี่ยวกับการจัดตารางการทำงานของพนักงานสำหรับจัดแข่งกีฬา ซึ่งมีความซับซ้อนในด้านทักษะเฉพาะของพนักงาน กฎหมายแรงงาน และเวลาที่พนักงานแต่ละคนสามารถทำงาน การแก้ปัญหาเบื้องต้นได้เสนอให้ทำการแก้ปัญหาขนาดเล็กด้วยโปรแกรมเชิงเส้นแบบผสม (Mixed integer linear programming) สำหรับหาคำตอบแม่นตรง (Exact solution) จากนั้นจึงเริ่มพัฒนา ALNS เพื่อแก้ปัญหาขนาดใหญ่ในระยะเวลาที่สั้นลงซึ่งให้ผลที่ใกล้เคียงกับคำตอบแม่นตรง (Exact solution) ผู้วิจัยยังได้ให้ข้อเสนอว่า การใช้ ALNS ยังมีความเหมาะสมที่จะนำไปใช้กับปัญหาหลากหลาย ไม่ว่าจะเป็นปัญหาการจัดตารางงานพนักงาน การจัดตารางการบิน หรือตารางเดินรถไฟ รวมไปถึงการจัดตารางการทำงานของบุคลากรทางการแพทย์อีกด้วย

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

ในส่วนของการทำงานของ ALNS เราสามารถจำแนกออกเป็นขั้นตอนหลักๆ 4 ขั้นตอน ได้แก่ 1. กลไกการเลือกตัวดำเนินการ (Operator) 2. เกณฑ์ในการยอมรับคำตอบใหม่ที่ได้จากการปรับปรุงคำตอบ (Acceptance criterion) 3. เกณฑ์การหยุดปรับปรุงคำตอบ (Stopping criterion) 4. การออกแบบตัวทำลายคำตอบ และตัวปรับปรุงคำตอบ (Design destroy and repair operator) โดยขั้นตอนทั้ง 4 จะสามารถแบ่งย่อยตามวิธีที่ใช้ในการแก้ปัญหาได้อีกหลายวิธี ซึ่งในขั้นตอนการเลือกใช้จะขึ้นอยู่กับความเหมาะสมในการนำมาประยุกต์ใช้กับแต่ละงานวิจัย แสดงความนิยมใช้งานในแต่ละวิธีได้จากตารางที่ 2-1 (Mara et al., 2022)

ตารางที่ 2-1 โครงสร้างและเทคทิคของ ALNS

โครงสร้างของ ALNS	วิธีการ	จำนวนบทความวิจัยที่นำไปใช้ (หน่วย : บทความ)	สัดส่วนการนำไปใช้ (หน่วย : %)
กลไกการเลือกตัวดำเนินการ	Roulette Wheel	250	99.21 %
	Other Mechanism	1	0.40 %
เกณฑ์ในการยอมรับคำตอบ	Greedy Mechanism	20	7.94 %
	Metropolis Criterion	205	81.35 %
	Record-to-Record	20	7.94 %
	Threshold Acceptance	3	1.19 %
	Pareto Domination	3	1.19 %
	Other Mechanism	6	2.38 %
เกณฑ์การหยุดปรับปรุงคำตอบ	Number of Iterations	188	74.60 %
	Number of Non-Improving Iterations	48	19.05 %
	Running Time Limit	49	19.44 %
	Annealing Temperature	25	9.92 %
	Other Mechanism	3	1.19 %

ขั้นตอนการทำลายคำตอบ และซ่อมแซมคำตอบได้มีการปรับปรุง โดยมีการปรับค่าโอกาสในการเลือกใช้ตัวดำเนินการจากการเก็บสถิติในระหว่างการค้นหาคำตอบ เพื่อให้สามารถปรับปรุงไปในทิศทางที่ถูกต้องได้มากขึ้น แต่นอกเหนือจากการปรับปรุงโดยการเก็บสถิติแล้วเราควรมีการเลือกตัวทำลายแบบสุ่มเข้าไปด้วยจะช่วยให้คำตอบเกิดความหลากหลาย และในบางครั้งอาจสร้างโอกาสให้คำตอบเข้าใกล้ Optimal solution ได้ในระยะเวลาที่สั้นลง

เมื่อได้วิธีการเลือกตัวทำลายคำตอบแล้ว เราจะต้องกำหนดจำนวนคำตอบที่เราจะทำลายในแต่ละครั้ง (Neighborhood size) ที่จะมีส่วนทำให้ ALNS ทำงานได้อย่างมีประสิทธิภาพ การเลือกจำนวนคำตอบที่จะทำลายในแต่ละครั้ง หากเลือกจำนวนมาก ALNS จะใช้เวลาในการหาคำตอบนาน และเพิ่มความซับซ้อนให้กับขั้นตอนการซ่อมแซมคำตอบ แต่จะเป็นการเพิ่มโอกาสที่คำตอบจะไม่ติดอยู่ใน Local minima กลับกันหากเราเลือกจำนวนการคำตอบสำหรับการทำลายน้อยเกินไปจะทำให้คำตอบอาจติดอยู่ใน Local minima และไม่สามารถซ่อมแซมคำตอบให้มีประสิทธิภาพดีขึ้นได้ การเลือกจำนวนคำตอบที่ที่จะทำลายจึงมีความสำคัญอย่างยิ่ง (Sacramento et al., 2019)

ภายหลังที่คำตอบผ่านขบวนการปรับปรุง เราไม่อาจทราบได้ว่าคำตอบที่ผ่านขบวนการปรับปรุงมานั้นมีประสิทธิภาพเพียงพอต่อการนำไปใช้ หรือนำไปปรับปรุงต่อหรือไม่ เพื่อหลีกเลี่ยงการยอมรับคำตอบทุกคำตอบ จากขั้นตอนการปรับปรุงคำตอบที่จะส่งผลให้คำตอบมีประสิทธิภาพต่ำลงได้ จึงจำเป็นจะต้องควบคุม และมีเกณฑ์ในการยอมรับคำตอบให้กับ ALNS ซึ่งวิธีที่นิยมนำมาใช้ก็คือ Simulated Annealing (SA) (Kirkpatrick et al., 1983 ; Černý, 1985) โดยหลักการของ SA ใช้วิธีการเลียนแบบพฤติกรรมของขบวนการหลอมเหล็ก ซึ่งจะใช้อุณหภูมิเป็นเกณฑ์ควบคุมโอกาสในการยอมรับคำตอบ (Acceptance probability) ถ้าคำตอบที่ได้จากขั้นตอนการทำลาย และซ่อมแซมมีค่าวัตถุประสงค์ที่ดีขึ้นกว่าคำตอบในปัจจุบัน SA จะยอมรับคำตอบในทุกกรณี แต่ถ้าคำตอบไม่ดีกว่าคำตอบในปัจจุบัน SA จะใช้การตัดสินใจยอมรับคำตอบ จากการคำนวณโอกาสในการยอมรับคำตอบ (Acceptance probability)

จากที่ได้กล่าวมาถึงปัญหา และเทคนิคที่ใช้ในการแก้ปัญหาการจัดตารางการทำงานของพยาบาล เราสามารถอ้างอิงจากงานวิจัยที่มีความใกล้เคียงเพื่อเป็นแนวทางในงานวิจัยได้ แต่ด้วยปัญหาการจัดตารางการทำงานของพยาบาลในโรงพยาบาลแต่ละแห่งมีความจำเพาะในหลากหลายด้าน เช่น ด้านกฎระเบียบของโรงพยาบาล กฎสภาพยาบาล หรือแม้กระทั่งวัฒนธรรมในแต่ละประเทศ จึงทำให้ไม่สามารถนำงานวิจัยอื่นๆ มาใช้สร้างตารางการทำงานของพยาบาลได้อย่างเหมาะสม แต่เราสามารถนำมาใช้เป็นแนวทางการแก้ปัญหาเพื่อบรรลุวัตถุประสงค์ในการจัดตารางการทำงานของพยาบาลดัง ตารางที่ 2-2 ความเกี่ยวข้อง และความแตกต่างของงานวิจัย

ตารางที่ 2-2 ตารางความเกี่ยวข้อง และความแตกต่างของงานวิจัย

Research	Objective characteristics			Constraints		Pareto Fronts	Technique	Real Life Data
	Cost	Fairness	Preferences	HC	SC			
Building better nurse scheduling algorithms (Aickelin et al., 2004)			✓	✓			IP&GA	
Preference scheduling for nurses using column generation (Bard et al., 2005)			✓	✓	✓		IP&CG	
An ACO algorithm for a dynamic regional nurse scheduling problem in Austria (Gutjahr et al., 1997)	✓		✓	✓		✓	AOC	
Nurse scheduling in a hospital emergency department: a case study at a Thai university hospital (Dumrongsiri et al., 2018)		✓	✓	✓			MOA	✓

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

ตารางที่ 2-2 ตารางความเกี่ยวข้อง และความแตกต่างของงานวิจัย (ต่อ)

Research	Objective characteristics			Constraints		Pareto Fronts	Technique	Real Life Data
	Cost	Fairness	Preferences	HC	SC			
Local search neighborhoods for dealing with a novel nurse rostering model (Bilgin et al., 2012)			✓	✓	✓		VNS-ALNS	✓
Cooperative search for fair nurse rosters (Martin et al., 2013)		✓	✓	✓	✓		Agent-based	✓
An integrated nurse staffing and scheduling analysis for longer-term nursing staff allocation problems (Maenout et al., 2013)	✓	✓	✓	✓	✓		MILP	✓
A heuristic approach to the multi-objective nurse scheduling problem: A case study of private hospital in Thailand	✓	✓	✓	✓	✓	✓	**MILP&ALNS with FNDS**	✓

การศึกษางานวิจัยที่เกี่ยวกับปัญหาการจัดตารางการทำงานพยาบาลที่มีทั้งงานวิจัยในประเทศไทย และต่างประเทศ โดยแต่ละงานวิจัยจะมีความแตกต่างในด้าน ข้อมูลของพยาบาล แผนกในโรงพยาบาล วัตถุประสงค์ ข้อจำกัด และเทคนิคที่ใช้ในการแก้ปัญหา ดังเช่นในงานวิจัยของเราที่มุ่งเน้นให้ตารางการทำงานของพยาบาลสามารถบรรลุวัตถุประสงค์ในด้านค่าใช้จ่าย ความเท่าเทียมในการได้รับค่าจ้าง และความพึงพอใจในการทำงาน โดยข้อจำกัดจะมีทั้งข้อจำกัดที่ไม่สามารถละเมิดได้ และสามารถละเมิดได้ เพื่อให้ใกล้เคียงกับสถานการณ์การทำงานในโรงพยาบาล ซึ่งข้อมูลพยาบาลภายในแผนกจะถูกแก้ปัญหาด้วยแบบจำลองทางคณิตศาสตร์ และวิธีการฮิวริสติกในการสร้างตารางการทำงานของพยาบาลที่มีความเหมาะสม และมีประสิทธิภาพ

จุฬาลงกรณ์มหาวิทยาลัย
CHULALONGKORN UNIVERSITY

บทที่ 3
วิธีการดำเนินงานวิจัย

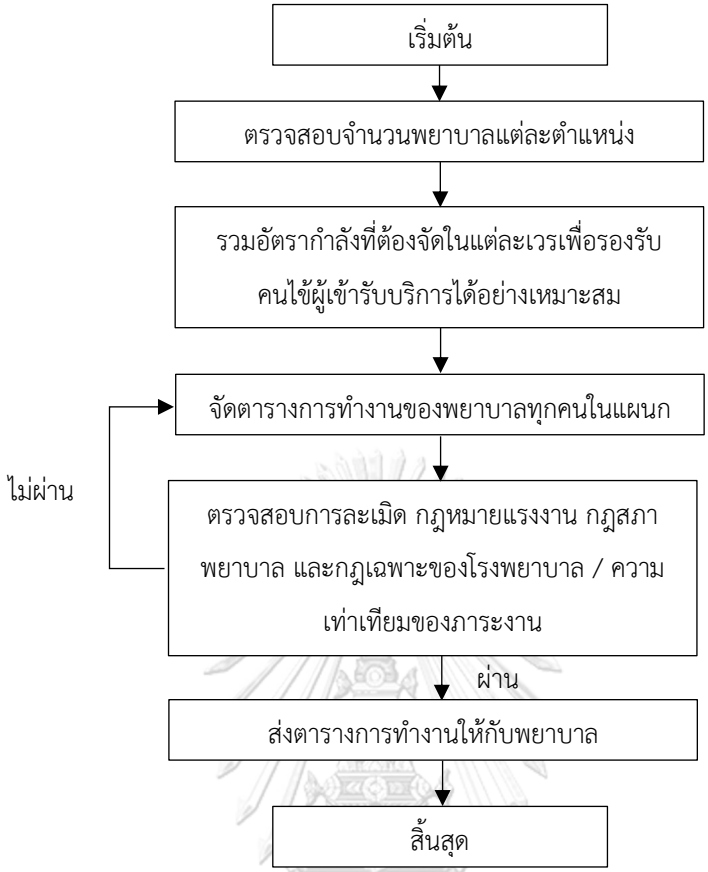
3.1 ลักษณะของปัญหา

การจัดตารางการทำงานพยาบาลในโรงพยาบาลทุกวันนี้ส่วนใหญ่แล้วยังคงนิยมใช้วิธีจัดตารางด้วยตนเอง เนื่องจากเป็นวิธีจัดตารางที่ถูกใช้มาเป็นเวลานาน และสามารถทำได้ง่าย หากแต่การจัดตารางด้วยพยาบาลจะเป็นที่ทราบกันดีว่าเป็นงานที่มีความซับซ้อน และมักจะเป็นงานที่ใช้เวลามาก ซึ่งความซับซ้อนในการจัดตารางการทำงานพยาบาลเกิดจากความต้องการที่หลากหลาย เช่น ความต้องการของโรงพยาบาล ความต้องการของพยาบาล ที่ถูกกำหนดมาเพื่อช่วยให้ตารางการทำงานพยาบาลมีความเป็นธรรมกับพยาบาลผู้ปฏิบัติงาน และสร้างมาตรฐานในการดูแลผู้ป่วย

สิ่งที่ควรระวังในการจัดตารางการทำงานพยาบาล คือ ตารางการทำงานพยาบาลต้องมีความเหมาะสม เพราะหากตารางการทำงานไม่เหมาะสมอาจนำไปสู่การได้รับภาระงานที่มากเกินไป ตัวอย่างเช่น การได้รับชั่วโมงการทำงานเกินกว่ากำหนด และความเหนื่อยล้าสะสมจากการทำงานหนัก ปัญหาเหล่านี้จะส่งผลกระทบต่อความสามารถในการดูแลผู้ป่วยของพยาบาล ซึ่งผลกระทบจากตารางที่มีความไม่เหมาะสมจะไม่เพียงเกิดปัญหากับพยาบาลผู้ปฏิบัติงานเท่านั้น แต่ยังเกิดปัญหากับผู้ป่วยได้ด้วย เช่น มีพยาบาลไม่เพียงพอต่อผู้ป่วยที่เข้ารับบริการทำให้เกิดความล่าช้าในการเข้ารักษา หรือการรักษาที่ไม่มีประสิทธิภาพ ส่งผลเสียต่อภาพลักษณ์ของโรงพยาบาล ปัญหาเหล่านี้อาจเกิดขึ้นได้หากเราใช้วิธีการจัดตารางด้วยตนเอง ที่แม้จะเป็นวิธีการจัดตารางที่ทำได้ง่ายที่สุด แต่ไม่สามารถการันตีได้ว่าตารางที่ได้มามีประสิทธิภาพ และความถูกต้องมากน้อยเพียงใด

ดังเช่นการจัดตารางการทำงานของพยาบาลในแผนกฉุกเฉินของโรงพยาบาล ที่ปัจจุบันยังจำเป็นต้องใช้หัวหน้าพยาบาลจัดสรรพยาบาลทุกตำแหน่งให้สอดคล้องกับปริมาณผู้ป่วยที่เข้ารับบริการ หัวหน้าพยาบาลจำเป็นต้องจัดตารางการทำงานภายใต้ข้อจำกัดในการทำงาน และกฎระเบียบต่างๆ ทำให้ต้องใช้ความละเอียดรอบคอบ และใช้เวลาในการจัดตารางนาน แต่ถึงอย่างนั้นการจัดตารางด้วยมือยังคงมีข้อจำกัดทั้งในด้านประสิทธิภาพ และความถูกต้องของตาราง ทำให้พยาบาลในแผนกต้องเผชิญปัญหาตารางการทำงานมีความผิดพลาด และขาดความเท่าเทียม ซึ่งการจะลดเวลาในการจัดตาราง หรือเพิ่มประสิทธิภาพ ด้วยวิธีการจัดตารางแบบดั้งเดิมนั้นเป็นถือเรื่องที่ทำได้ยาก เนื่องจากขั้นตอนในการจัดตาราง และองค์ประกอบที่จะต้องคำนึงถึงมีจำนวนมาก เช่น การตอบสนองความต้องการพยาบาลในแต่ละวัน การตรวจสอบตาราง และการแก้ไขตารางให้เกิดความถูกต้อง เพียงแค่ขั้นตอนเหล่านี้อาจทำให้ต้องใช้หัวหน้าพยาบาล 1 คน เข้ามาทำหน้าที่จัดตารางถึง 2 วัน โดยจะสามารถสังเกตขั้นตอนการจัดตารางของหัวหน้าพยาบาลที่จะต้องปฏิบัติในการจัดตารางการทำงานพยาบาลแต่ละครั้งจะมีขั้นตอน แสดงดังรูปที่ 3-1

จุฬาลงกรณ์มหาวิทยาลัย
CHULALONGKORN UNIVERSITY



รูปที่ 3-1 ขั้นตอนการดำเนินงานการจัดตารางการทำงานของพยาบาล

จากรูปที่ 3-1 เราจะสังเกตได้ว่าทุกครั้งที่จะทำการจัดตารางการทำงานพยาบาลหัวหน้าพยาบาลจะทำการตรวจสอบจำนวนพยาบาลในแต่ละตำแหน่งเพื่อตรวจสอบว่าในการจัดตารางมีอัตรากำลังที่เพียงพอหรือไม่ จากนั้นหัวหน้าพยาบาลจะเริ่มจัดตารางโดยทยอยกรอกพยาบาลแต่ละคนลงในตารางจนได้อัตรากำลังที่เพียงพอในแต่ละกะการทำงาน เพื่อให้ได้ตารางการทำงานล่วงหน้าจำนวน 28 วัน แต่หลังจากที่จัดตารางการทำงานเสร็จ จะยังไม่สามารถนำไปใช้งานได้ทันที หัวหน้าพยาบาลยังจะต้องตรวจสอบว่าตารางการทำงานนั้นได้มีการละเมิดกฎหมายแรงงาน กฎสภาพยาบาล และกฎเฉพาะของโรงพยาบาลอยู่ในตารางการทำงานที่ถูกจัดหรือไม่ ซึ่งในกรณีที่อัตรากำลังคนขาดการละเมิดกฎบางข้อมีความจำเป็นที่จะต้องละเมิดจะยิ่งเพิ่มความซับซ้อนให้กับหัวหน้าพยาบาล การแก้ไขจนตารางการทำงานพยาบาลแล้วเสร็จ จะใช้เวลาในการจัดตารางประมาณ 2 วันทำการ (อ้างอิงจากเวลาโดยเฉลี่ยของการจัดตารางการทำงานของแผนกฉุกเฉินของโรงพยาบาลเอกชนแห่งหนึ่งในประเทศไทย) การใช้เวลาจัดตารางการทำงานถึง 2 วัน สะท้อนให้เห็นว่า หากสามารถลดภาระงานในส่วนของหัวหน้าพยาบาลไปได้ หัวหน้าพยาบาลจะมีเวลาเพิ่มสำหรับปฏิบัติหน้าที่ที่มีความสำคัญในส่วนอื่นๆได้อีกด้วย

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

ในงานวิจัยฉบับนี้จะเป็นการศึกษาปัญหาการจัดตารางการทำงานของพยาบาล (Nurse Scheduling Problem, NSP) เพื่อพัฒนาโปรแกรมการจัดตารางการทำงานพยาบาลมาทดแทนหน้าที่ของหัวหน้าพยาบาลที่ดั้งเดิมจะต้องจัดตารางการทำงานของพยาบาลด้วยมือ โดยโปรแกรมดังกล่าวจะถูกพัฒนาให้สามารถตอบสนองข้อมูลที่ใช้ในการจัดตารางการทำงานพยาบาลชุดเดียวกับการจัดตารางแบบดั้งเดิม เพื่อแสดงให้เห็นถึงประสิทธิภาพทั้งในด้านคุณภาพของคำตอบ และด้านระยะเวลาในการหาคำตอบ

จากการศึกษาข้อมูลระดับของพยาบาลในแผนกมีสิ่งที่น่าสนใจ คือ การจัดระดับของพยาบาลที่แตกต่างกันมาทำงานร่วมกัน หรือการทดแทนระดับกันได้ โดยในงานวิจัยนี้ โรงพยาบาลมีข้อกำหนดสำหรับพยาบาลที่มีระดับสูงกว่าสามารถทำงานแทนพยาบาลที่มีระดับต่ำกว่าได้ (ในสถานการณ์ปกติการจัดตารางการทำงานภายในแผนกฉุกเฉินจะไม่มีการจัดตารางให้พยาบาลทำงานข้ามระดับ จะทำได้เมื่อมีเหตุจำเป็นเท่านั้น) แต่พยาบาลที่มีระดับต่ำกว่าจะไม่สามารถทำงานแทนพยาบาลที่มีระดับสูงกว่าได้ โดยสามารถแบ่งระดับออกเป็น 4 ระดับ ได้แก่

- หัวหน้าพยาบาล : สามารถทำงานได้ทุกระดับแต่ควรหลีกเลี่ยงการให้หัวหน้าพยาบาลไปทำงานทดแทนระดับอื่นๆ ซึ่งภายในแผนกจะมีหัวหน้าพยาบาลจำนวน 1 คน
- พยาบาลวิชาชีพระดับปฏิบัติการ : สามารถทำงานได้ทุกระดับยกเว้นหัวหน้าพยาบาล ซึ่งภายในแผนกจะมีพยาบาลวิชาชีพระดับปฏิบัติการจำนวน 16 คน
- ผู้ช่วยพยาบาล : สามารถทำงานของผู้ช่วยพยาบาล และพนักงานฉุกเฉินการแพทย์ ซึ่งภายในแผนกจะมีผู้ช่วยพยาบาลจำนวน 2 คน
- พนักงานฉุกเฉินการแพทย์ : สามารถทำงานได้เฉพาะระดับพนักงานฉุกเฉินการแพทย์เท่านั้น ซึ่งภายในแผนกจะมีพนักงานฉุกเฉินการแพทย์จำนวน 6 คน

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

แม้ว่าจะสามารถให้พยาบาลที่ตำแหน่งสูงกว่าเข้ากะการทำงานแทนพยาบาลที่ตำแหน่งต่ำกว่าได้ แต่จะต้องแลกมากับค่าใช้จ่ายในการจ้างพยาบาลในกะนั้นที่สูงขึ้น ซึ่งในกรณีโรงพยาบาลประสบปัญหาพยาบาลบางตำแหน่งขาดแคลน การทดแทนตำแหน่งมีความจำเป็นอย่างยิ่ง เพราะจะช่วยให้โรงพยาบาลสามารถรับมือกับปริมาณคนไข้ในแต่ละวันได้อย่างมีประสิทธิภาพ และสามารถตอบสนองความต้องการของทางโรงพยาบาล และพยาบาลผู้ปฏิบัติงานได้ดีขึ้น

จากการพิจารณาลักษณะของปัญหาการจัดตารางการทำงานพยาบาลที่ได้กล่าวมาเราสามารถนำเทคนิคที่จะช่วยให้เกิดการแก้ปัญหาที่มีประสิทธิภาพได้ โดยการนำแบบจำลองทางคณิตศาสตร์ตัวแบบชนิดผสม Mixed Integer Linear Programming (MILP) ที่ช่วยจัดสรรทั้งปริมาณภาระงาน และทรัพยากรได้อย่างเป็นระบบ ผ่านการแปลงข้อมูลให้อยู่ในรูปของตัวแปร และสมการ ที่การันตีคุณภาพของคำตอบมีความน่าเชื่อถือ โดยแบบจำลองทางคณิตศาสตร์ตัวแบบชนิดผสมที่จะใช้ในการสร้างตารางการทำงานพยาบาลล่วงหน้าเป็นเวลา 28 วัน สำหรับพยาบาลจำนวน 25 คน 4 ตำแหน่ง มีรายละเอียดดังต่อไปนี้

3.2 แบบจำลองทางคณิตศาสตร์

3.2.1 ค่าดัชนี (Indices)

N = พยาบาลทั้งหมดภายในแผนก (n ∈ N)

HD = หัวหน้าพยาบาล (hd ∈ N)

RN = พยาบาลวิชาชีพระดับปฏิบัติการ (rn ∈ RN)

PN = ผู้ช่วยพยาบาล (pn ∈ PN)

EMT = พนักงานฉุกเฉินการแพทย์ (emt ∈ EMT)

E = พยาบาลที่สามารถทำงานในกะดึกได้ (e ∈ E)

D = ลำดับวัน {1,2,3,4,5,6,7} ∈ D (โดยดัชนีที่ 1 คือ วันจันทร์, 2 คือ วันอังคาร, 3 คือ วันพุธ, 4 คือ วันพฤหัสบดี, 5 คือ วันศุกร์, 6 คือ วันเสาร์, 7 คือ วันอาทิตย์)

S = ชนิดของกะการทำงาน {1,2,3} ∈ S (โดยดัชนีที่ 1 คือ กะการทำงานเช้า, 2 คือ กะการทำงานบ่าย, 3 คือ กะการทำงานดึก, 4 คือ กะการทำงานDay, 5 คือ กะการทำงานNight, 6 คือ กะการทำงาน11)

K = ระดับของพยาบาล {1,2,3,4} ∈ K (โดยดัชนีที่ 1 คือ พนักงานฉุกเฉินการแพทย์, 2 คือ ผู้ช่วยพยาบาล, 3 คือ พยาบาลวิชาชีพระดับปฏิบัติการ, 4 คือ หัวหน้าพยาบาล)

Q = จำนวนสัปดาห์การวางแผนทั้งหมด (Planning Horizon) (1,2,3,4 ∈ Q)

V = ชนิดของการละเมิดข้อจำกัด (v ∈ V)

3.2.2 ค่าคงที่ (Parameters)

C_{ns} = ค่าจ้างของพยาบาล n เมื่อ กำหนดให้เข้ากะการทำงาน s

R_{dsk} = จำนวนพยาบาลที่ต้องการในวันที่ d ของกะการทำงาน s ระดับที่ k

WR_s = จำนวนชั่วโมงการทำงานของกะการทำงาน s

NA_{ndq} = พยาบาล n ขอลาหยุด หรือออกอบรม ในวันที่ d ของสัปดาห์ที่ q

I_{nk} = ค่าคงที่การจัดระดับของพยาบาลคนที่ n ระดับที่ k

W_{max_week} = จำนวนกะการทำงานรวมสูงสุดที่พยาบาลทุกคนสามารถทำได้ไม่เกินในหนึ่งสัปดาห์

W_{min_week} = จำนวนกะการทำงานรวมขั้นต่ำที่พยาบาลทุกคนต้องทำในหนึ่งสัปดาห์

A_{max} = จำนวนวันที่สามารถเข้าทำงานติดต่อกันสูงสุด

A_{min} = จำนวนวันขั้นต่ำที่ต้องทำงานติดต่อกันต่ำสุด

LNS = จำนวนสูงสุดของการเข้ากะการทำงานดึกของพยาบาลทุกคนในการวางแผนทั้งหมด (Planning Horizon)

3.2.3 ตัวแปรตัดสินใจ (Decision Variables)

X_{ndsqk} = 1 เมื่อพยาบาล n ถูกกำหนดให้เข้าทำงานในวันที่ d กะการทำงานที่ s ในสัปดาห์ที่ q ในระดับที่ k
= 0 กรณีอื่นๆ

H_{ndq} = 1 เมื่อพยาบาล n ถูกกำหนดให้เข้าทำงานในวันที่ d ในสัปดาห์ที่ q
= 0 กรณีอื่นๆ

Y_{dqk} = 1 เมื่อมีพยาบาลเข้าในกะการทำงาน Day มากกว่า Night ในวันที่ d สัปดาห์ที่ q ในระดับที่ k
= 0 กรณีอื่นๆ

$UNFAIRNESS_WageRN_{np}$ = ค่าจ้างของพยาบาลวิชาชีพระดับปฏิบัติการ ที่เหลื่อมล้ำของพยาบาลลำดับที่ n และ p

$UNFAIRNESS_WagePN_{np}$ = ค่าจ้างของผู้ช่วยพยาบาล ที่เหลื่อมล้ำของพยาบาลลำดับที่ n และ p

$UNFAIRNESS_WageEMT_{np}$ = ค่าจ้างของพนักงานฉุกเฉินการแพทย์ ที่เหลื่อมล้ำของพยาบาลลำดับที่ n และ p

DN_{dqk} = จำนวนกะบ่ายของวันที่ d ในสัปดาห์ที่ q ระดับที่ k เมื่อในวันนั้นๆ มีการเข้ากะการทำงาน Day และNight ในวันเดียวกัน

V_{vnq} = จำนวนการละเมิดข้อจำกัดของดัชนีที่ v ของพยาบาลลำดับที่ n ในสัปดาห์ที่ q

V_{vndq} = จำนวนการละเมิดข้อจำกัดของดัชนีที่ v ของพยาบาลลำดับที่ n ในวันที่ d ในสัปดาห์ที่ q

V_{vds} = จำนวนการละเมิดข้อจำกัดของดัชนีที่ v ในวันที่ d กะการทำงานที่ s

V_{vsq} = จำนวนการละเมิดข้อจำกัดของดัชนีที่ v ของกะการทำงานที่ s ในสัปดาห์ที่ q

V_{vd} = จำนวนการละเมิดข้อจำกัดของดัชนีที่ v ในวันที่ d

V_{vns} = จำนวนการละเมิดข้อจำกัดของดัชนีที่ v ของพยาบาลลำดับที่ n กะการทำงานที่ s

V_{v} = ผลรวมของบทลงโทษสำหรับการละเมิดข้อจำกัดของดัชนีที่ v (ดูตารางที่ 3-1 และสมการ 1 – 10 ด้านล่างประกอบ)

ตารางที่ 3-1 การละเมิดกฎข้อจำกัดและบทลงโทษ

ลำดับ	คำอธิบาย	ดัชนี	น้ำหนักการลงโทษ	ข้อจำกัด	ตัวแปรการละเมิด
1.	การละเมิดกฎจำนวนการทำงานที่ทำได้ใน 1 สัปดาห์				
1.1	จำนวนกะการทำงานสูงสุดถูกละเมิด	1	w_1 = 2,000	3	V_1
1.2	จำนวนกะการทำงานขั้นต่ำถูกละเมิด	2	w_2 = 500	4	V_2

ตารางที่ 3-1 การละเมิดกฎข้อจำกัดและบทลงโทษ (ต่อ)

ลำดับ	คำอธิบาย	ดัชนี	น้ำหนักการลงโทษ	ข้อจำกัด	ตัวแปรการละเมิด
2.	การละเมิดกฎการทำงานติดต่อกัน				
2.1	จำนวนวันเข้าทำงานติดต่อกันสูงสุดในสัปดาห์ถูกละเมิด	3	w_3 = 2,400	5	V_3
2.2	จำนวนวันเข้าทำงานติดต่อกันขั้นต่ำในสัปดาห์ถูกละเมิด	4	w_4 = 2,400	6	V_4
2.3	จำนวนชั่วโมงทำงานรวมในสัปดาห์ถูกละเมิด	5	w_5 = 200	8	V_5
3.	การละเมิดกฎรูปแบบการทำงาน				
3.1	การทำงานในกะดึกติดต่อกันเกิน 1 วัน ในสัปดาห์ถูกละเมิด	6	w_6 = 800	16	V_6
3.2	การทำงานในกะดึกติดต่อกันเกิน 1 วัน ข้ามสัปดาห์ถูกละเมิด	7	w_7 = 800	17	V_7
3.3	การทำงานในกะบ่ายแล้วห้ามต่อด้วยกะเช้า หรือกะDay ในสัปดาห์ถูกละเมิด	8	w_8 = 1,500	18	V_8
3.4	การทำงานในกะบ่ายแล้วห้ามต่อด้วยกะเช้า หรือกะDay ข้ามสัปดาห์ถูกละเมิด	9	w_9 = 1,500	19	V_9
4.	การละเมิดกฎจำนวนการทำงานในกะดึก				
4.1	จำนวนกะดึกสูงสุดถูกละเมิด	10	w_{10} = 2,200	20	V_{10}

$$V_1 = \sum_{n \in N} \sum_{q \in Q} w_1 * V_{1nq} \quad (1)$$

$$V_2 = \sum_{n \in N} \sum_{q \in Q} w_2 * V_{2nq} \quad (2)$$

$$V_3 = \sum_{n \in N} \sum_{1}^{|D|-1} \sum_{q \in Q} w_3 * V_{3ndq} \quad (3)$$

$$V_4 = \sum_{n \in N} \sum_{1}^{|D|-1} \sum_{q \in Q} w_4 * V_{4ndq} \quad (4)$$

$$V_5 = \sum_{n \in N} \sum_{q \in Q} w_5 * V_{5nq} \quad (5)$$

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

$$V_6 = \sum_{n \in N} \sum_{d \in D/\{7\}} \sum_{s \in 3,5} \sum_{q \in Q} w_6 * V_{6ndsq} \quad (6)$$

$$V_7 = \sum_{n \in N} \sum_{s \in 3,5} \sum_{q \in Q/\{4\}} w_7 * V_{7nsq} \quad (7)$$

$$V_8 = \sum_{n \in N} \sum_{d \in D/\{7\}} \sum_{q \in Q} w_8 * V_{8ndq} \quad (8)$$

$$V_9 = \sum_{n \in N} \sum_{q \in Q/\{4\}} w_9 * V_{9nq} \quad (9)$$

$$V_{10} = \sum_{n \in E} \sum_{s \in 3,5} w_{10} * V_{10ns} \quad (10)$$

3.2.4 ฟังก์ชันจุดประสงค์ (Objective Function)

การจัดตารางการทำงานของพยาบาลในงานวิจัยฉบับนี้ มีวัตถุประสงค์หลักเพื่อลดปัญหาที่เกิดขึ้นจากการสร้างตารางแบบดั้งเดิมที่มีความซับซ้อน และมีข้อจำกัดที่หลากหลาย ซึ่งหากทำการจัดตารางด้วยตนเอง อาจใช้เวลานาน และอาจเกิดความผิดพลาดได้ง่าย

ฟังก์ชันจุดประสงค์ของแบบจำลองที่สร้างขึ้นถูกอ้างอิงจากความต้องการของโรงพยาบาล และพยาบาลผู้ปฏิบัติงาน ซึ่งประกอบไปด้วย 3 จุดประสงค์หลัก ได้แก่

1 การลดค่าใช้จ่ายในการจ้างพยาบาล (Minimize cost) เพื่อให้เกิดการใช้ทรัพยากรที่มีอยู่อย่างมีประสิทธิภาพ (Obj.1)

2 การเพิ่มความเท่าเทียมกันในมุมของค่าจ้างในการเข้ากะที่พยาบาลแต่ละคนในแผนกได้รับ (Minimize unfairness wages) เพื่อให้การทำงานของพยาบาลมีความเป็นธรรม และเท่าเทียมกัน โดยจัดตารางการทำงานให้มีจำนวนชนิดกะการทำงานเท่ากัน หรือใกล้เคียงกันมากที่สุด เพื่อป้องกันไม่ให้พยาบาลได้รับเงินค่าจ้างพิเศษไม่เป็นธรรม (Obj.2)

3 มุ่งเน้นให้เกิดการละเมิดข้อจำกัดให้น้อยที่สุดเท่าที่เป็นไปได้ (Minimize the violation) เพื่อให้พยาบาลรู้สึกพึงพอใจ และมีความสุขในการทำงาน (Obj.3)

ดังแสดงในสมการ (Obj. 1) – (Obj. 3) ตามลำดับ

$$\text{Minimize} \quad \sum_{n \in N} \sum_{d \in D} \sum_{s \in S} \sum_{q \in Q} \sum_{k \in K} X_{ndsqk} * C_{ns} \quad (\text{Obj.1})$$

$$\text{Minimize} \quad \Big(\sum_{n \in RN} \sum_{p \in RN} \text{UNFAIRNESS}_{\text{Wage}} \text{RN}_{np} + \sum_{n \in PN} \sum_{p \in PN} \text{UNFAIRNESS}_{\text{Wage}} \text{PN}_{np} + \sum_{n \in EMT} \sum_{p \in EMT} \text{UNFAIRNESS}_{\text{Wage}} \text{EMT}_{np} \Big) \qquad \text{(Obj.2)}$$

$$\text{Minimize} \quad \sum_{v \in V} V_v \qquad \text{(Obj.3)}$$

สังเกตว่าทั้ง 3 วัตถุประสงค์มีความขัดแย้งกันกล่าวคือ หากต้องการลดค่าใช้จ่ายให้ต่ำที่สุดพยาบาลอาจต้องทำงานติดต่อกันนานขึ้น หรือได้รับวันหยุดพักผ่อนน้อย ซึ่งทำให้เกิดไม่พึงพอใจในการทำงาน ดังนั้นเราจึงต้องคำนึงถึงความสมดุลของแต่ละวัตถุประสงค์ให้มีความสำคัญอย่างเท่าเทียมกัน เพื่อให้สามารถตอบสนองกับความต้องการของโรงพยาบาล และพยาบาลผู้ปฏิบัติงานของโรงพยาบาล

3.2.5 ข้อจำกัด (Constraints)

$$\sum_{k \in K} X_{ndsqk} \leq 1 \qquad ; \forall n \in N, d \in D, s \in S, q \in Q \qquad (1)$$

$$\sum_{s \in S} \sum_{k \in K} X_{ndsqk} \leq H_{ndq} \qquad ; \forall n \in N, d \in D, q \in Q \qquad (2)$$

$$\sum_{d \in D} \sum_{s \in S} \sum_{k \in K} X_{ndsqk} - V_{1nq} \leq W_{max_week} \qquad ; \forall n \in N, q \in Q \qquad (3)$$

$$\sum_{d \in D} \sum_{s \in S} \sum_{k \in K} X_{ndsqk} + V_{2nq} \geq W_{min_week} \qquad ; \forall n \in N, q \in Q \qquad (4)$$

$$\sum_{d}^{|D|} H_{ndq} - V_{3ndq} \leq A_{max} \qquad ; \forall n \in N, d \in (1,2,\dots,(|D|-1)), q \in Q \qquad (5)$$

$$\sum_{d}^{|D|} H_{ndq} + V_{4ndq} \geq A_{min} \qquad ; \forall n \in N, d \in (1,2,\dots,(|D|-1)), q \in Q \qquad (6)$$

$$(k - I_{nk}) * X_{ndsqk} = 0 \qquad ; \forall n \in N, d \in D, q \in Q, k \in K \qquad (7)$$

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

$$\sum_{d \in D} \sum_{s \in S} \sum_{k \in K} WR_s * X_{ndsqk} - V_{5nq} \leq 48$$

$$; \forall n \in N, q \in Q \quad (8)$$

$$\sum_{n \in N} (X_{nd1qk} + X_{nd4qk}) \geq R_{d1k} \quad ; \forall d \in D, q \in Q, k \in K \quad (9)$$

$$\sum_{n \in N} (X_{nd2qk} + X_{nd6qk}) + DN_{dqk} \geq R_{d2k}$$

$$; \forall d \in D, q \in Q, k \in K \quad (10)$$

$$\sum_{n \in N} (X_{nd3qk} + X_{nd5qk}) \geq R_{d3k} \quad ; \forall d \in D, q \in Q, k \in K \quad (11)$$

$$\sum_{n \in N} X_{nd4qk} - \sum_{n \in N} X_{nd5qk} \leq M * Y_{dqk}$$

$$; \forall d \in D, q \in Q, k \in K \quad (12)$$

$$\sum_{n \in N} X_{nd5qk} - \sum_{n \in N} X_{nd4qk} \leq M * (1 - Y_{dqk})$$

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

$$; \forall d \in D, q \in Q, k \in K \quad (13)$$

$$DN_{dqk} \geq \sum_{n \in N} X_{nd5qk} - M * (1 - Y_{dqk})$$

$$; \forall d \in D, q \in Q, k \in K \quad (14)$$

$$DN_{dqk} \geq \sum_{n \in N} X_{nd4qk} - M * Y_{dqk} \quad ; \forall d \in D, q \in Q, k \in K \quad (15)$$

$$X_{ndsqk} + X_{n(d+1)sqk} - V_{6ndsq} \leq 1 \quad ; \forall n \in N, d \in D/\{7\}, s \in \{3,5\}, q \in Q \quad (16)$$

$$X_{n7sqk} + X_{n1s(q+1)k} - V_{7nsq} \leq 1 \quad ; \forall n \in N, s \in \{3,5\}, q \in Q/\{4\} \quad (17)$$

$$X_{nd2qk} + X_{n(d+1)1qk} + X_{n(d+1)4qk} - V_{8ndq} \leq 1$$

$$; \forall n \in N, d \in D/\{7\}, q \in Q \quad (18)$$

$$X_{n72qk} + X_{n11(q+1)k} + X_{n14(q+1)k} - V_{9nq} \leq 1$$

$$; \forall n \in N, q \in Q/\{4\} \quad (19)$$

$$\sum_{d \in D} \sum_{q \in Q} \sum_{k \in K} X_{ndsqk} - V_{10ns} \leq LNS \quad ; \forall n \in E, s \in \{3,5\} \quad (20)$$

$$X_{ndsqk} \leq 0 \quad ; \forall n \in N/E, d \in D, s \in \{3,5\}, q \in Q, k \in K \quad (21)$$

$$\sum_{s \in S} \sum_{k \in K} X_{ndsqk} + NA_{ndq} \leq 1 \quad ; \forall n \in N, d \in D, q \in Q \quad (22)$$

$$UNFAIRNESS_WageRN_{np} \geq \sum_{d \in D} \sum_{s \in S} \sum_{q \in Q} \sum_{k \in K} \left((C_{ns} * X_{ndsqk}) - (C_{ps} * X_{pdsqk})\right)$$

$$; \forall n \in RN, p \in RN \quad (23)$$

$$UNFAIRNESS_WageRN_{np} \geq \sum_{d \in D} \sum_{s \in S} \sum_{q \in Q} \sum_{k \in K} \left((C_{ps} * X_{pdsqk}) - (C_{ns} * X_{ndsqk})\right)$$

$$; \forall n \in RN, p \in RN \quad (24)$$

$$UNFAIRNESS_WagePN_{np} \geq \sum_{d \in D} \sum_{s \in S} \sum_{q \in Q} \sum_{k \in K} \left((C_{ns} * X_{ndsqk}) - (C_{ps} * X_{pdsqk})\right)$$

$$; \forall n \in PN, p \in PN \quad (25)$$

$$UNFAIRNESS_WagePN_{np} \geq \sum_{d \in D} \sum_{s \in S} \sum_{q \in Q} \sum_{k \in K} \left((C_{ps} * X_{pdsqk}) - (C_{ns} * X_{ndsqk})\right)$$

$$; \forall n \in PN, p \in PN \quad (26)$$

$$UNFAIRNESS_WageEMT_{np} \geq \sum_{d \in D} \sum_{s \in S} \sum_{q \in Q} \sum_{k \in K} \left((C_{ns} * X_{ndsqk}) - (C_{ps} * X_{pdsqk})\right)$$

$$; \forall n \in EMT, p \in EMT \quad (27)$$

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

$$UNFAIRNESS_WageEMT_{np} \geq \sum_{d \in D} \sum_{s \in S} \sum_{q \in Q} \sum_{k \in K} \left((C_{ps} * X_{pdsqk}) - (C_{ns} * X_{ndsqk}) \right) \quad ; \forall n \in EMT, p \in EMT \tag{28}$$

$$X_{ndsqk} \in \{ 0,1 \} \quad ; \forall n \in N, d \in D, s \in S, q \in Q, k \in K \tag{29}$$

$$H_{ndq} \in \{ 0,1 \} \quad ; \forall n \in N, d \in D, q \in Q \tag{30}$$

$$Y_{dqk} \in \{ 0,1 \} \quad ; \forall d \in D, q \in Q, k \in K \tag{31}$$

$$UNFAIRNESS_{np} \geq 0 \quad ; \forall n \in N, p \in N \tag{32}$$

$$V_{vnq} \geq 0 \quad ; \forall v \in V, n \in N, q \in Q \tag{33}$$

$$V_{vndq} \geq 0 \quad ; \forall v \in V, n \in N, d \in D, q \in Q \tag{34}$$

$$V_{vq} \geq 0 \quad ; \forall v \in V, q \in Q \tag{35}$$

$$V_{vds} \geq 0 \quad ; \forall v \in V, d \in D, s \in S \tag{36}$$

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

$$V_{vsq} \geq 0 \quad ; \forall v \in V, s \in S, q \in Q \tag{37}$$

$$V_{vd} \geq 0 \quad ; \forall v \in V, d \in D \tag{38}$$

$$V_{vns} \geq 0 \quad ; \forall v \in V, n \in N, s \in S \tag{39}$$

$$V_{v} \geq 0 \quad ; \forall v \in V \tag{40}$$

I. ข้อจำกัดที่ 1 : พยาบาลแต่ละคนสามารถเข้ากะการทำงานได้เพียง 1 ระดับต่อกะการทำงาน

II. ข้อจำกัดที่ 2 : พยาบาลแต่ละคนสามารถเข้ากะการทำงานได้สูงสุด 1 กะการทำงานต่อวัน

III. ข้อจำกัดที่ 3 : กำหนดให้กะการทำงานรวมสูงสุดที่พยาบาลแต่ละคนได้รับในหนึ่งสัปดาห์ไม่ควรเกินค่าสูงสุด

IV. ข้อจำกัดที่ 4 : กำหนดให้กะการทำงานรวมอย่างน้อยที่สุดที่พยาบาลแต่ละคนได้รับในหนึ่งสัปดาห์ควรมากกว่าขั้นต่ำที่กำหนด

V. ข้อจำกัดที่ 5 : กำหนดให้พยาบาลแต่ละคนไม่ถูกกำหนดให้ทำงานติดต่อกันเกินจำนวนวันที่กำหนดเพื่อไม่ให้พยาบาลเกิดความล้าสะสมจากการทำงาน

VI. ข้อจำกัดที่ 6 : ถูกสร้างขึ้นเพื่อกำหนดขอบล่างสำหรับการเข้าทำงานของพยาบาลที่ติดต่อกันขั้นต่ำ

VII. ข้อจำกัดที่ 7 : ทำให้มั่นใจได้ว่าจะไม่มีการกำหนดกะการทำงานของพยาบาลที่มีระดับต่ำกว่าที่ต้องการ

VIII. ข้อจำกัดที่ 8 : ช่วยจำกัดจำนวนชั่วโมงการทำงานของพยาบาลในแต่ละสัปดาห์ให้ไม่เกิน 48 ชั่วโมง

IX. ข้อจำกัดที่ 9 : ทำให้มั่นใจได้ว่าจะมีพยาบาลเข้ากะการทำงานเช้า ในแต่ละระดับ ของแต่ละวันทำการมีเพียงพอต่อความต้องการ

X. ข้อจำกัดที่ 10 : ทำให้มั่นใจได้ว่าจะมีพยาบาลเข้ากะการทำงานบ่าย ในแต่ละระดับ ของแต่ละวันทำการมีเพียงพอต่อความต้องการ

XI. ข้อจำกัดที่ 11 : ทำให้มั่นใจได้ว่าจะมีพยาบาลเข้ากะการทำงานดึก ในแต่ละระดับ ของแต่ละวันทำการมีเพียงพอต่อความต้องการ

XII. ข้อจำกัดที่ 12 – 15 : เป็นข้อจำกัดที่จะช่วยนับจำนวนกะการทำงานบ่ายของวันที่ d สัปดาห์ที่ q ระดับที่ k เมื่อในวันนั้นๆ มีการเข้ากะDay และกะNight พร้อมกัน

XIII. ข้อจำกัดที่ 16 – 17 : ทำให้มั่นใจได้ว่าจะไม่มีพยาบาลคนใดได้รับกะการทำงานดึก และกะการทำงานNight ติดต่อกันมากกว่า 1 วัน โดยไม่มีเหตุจำเป็น

XIV. ข้อจำกัดที่ 18 – 19 : ทำให้มั่นใจได้ว่าจะไม่มีพยาบาลคนใดได้รับกะการทำงานบ่าย แล้วต่อด้วยกะการทำงานเช้า หรือกะการทำงานDay โดยไม่มีเหตุจำเป็น

XV. ข้อจำกัดที่ 20 : ป้องกันไม่ให้พยาบาลทุกคนภายในแผนกได้รับกะการทำงานดึก หรือกะการทำงานNight ตลอด 4 สัปดาห์เกินกว่าค่าสูงสุด

XVI. ข้อจำกัดที่ 21 : เป็นข้อจำกัดที่จะไม่มอบหมายกะการทำงานดึก หรือกะการทำงานNight ให้กับพยาบาลที่ไม่สามารถเข้างานในกะการทำงานดึกได้

XVII. ข้อจำกัดที่ 22 : ทำให้มั่นใจได้กว่าจะไม่มีการมอบหมายงานใดๆ ในวันที่พยาบาลขอลาหยุด

XVIII. ข้อจำกัดที่ 23 – 28 : เป็นข้อจำกัดที่จะช่วยสร้างความเท่าเทียมในการได้รับค่าจ้าง ของพยาบาลทุกคน

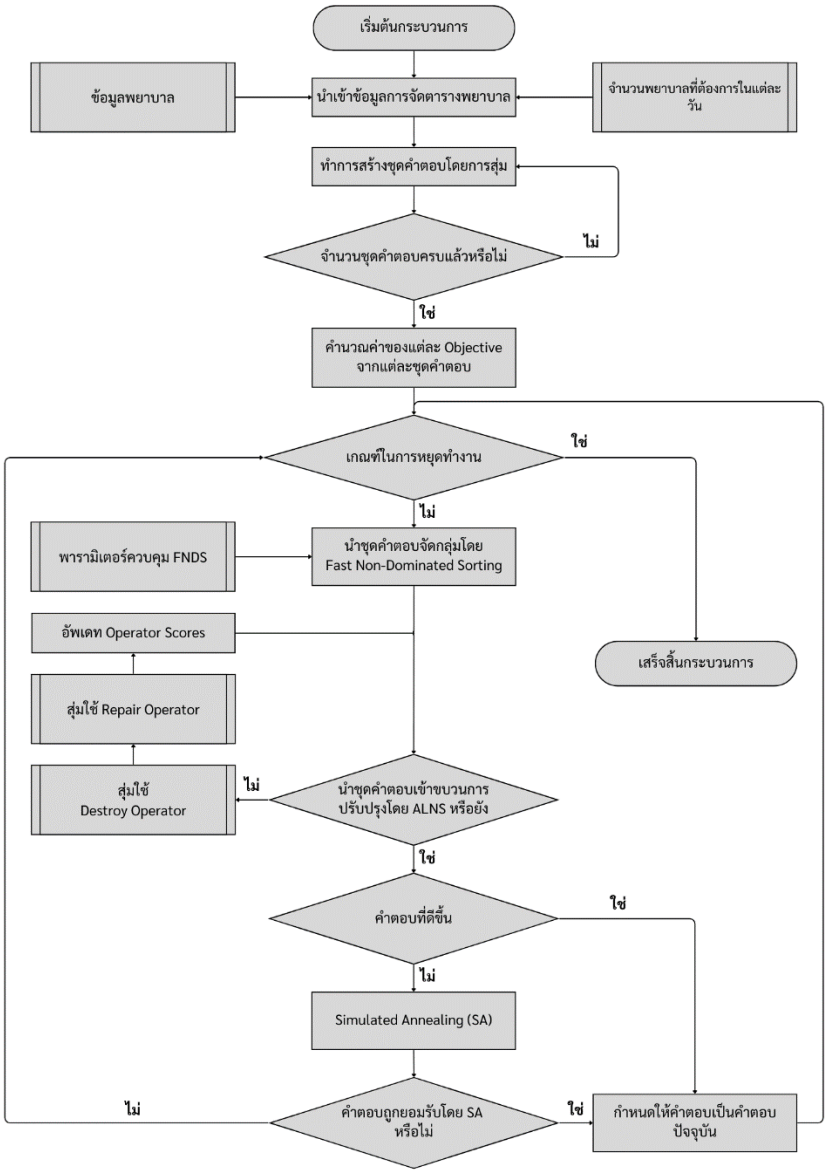
XIX. ข้อจำกัดที่ 29 – 31 : เป็นข้อจำกัดของตัวแปรฐานสอง (Binary variables)

XX. ข้อจำกัดที่ 32 – 40 : เป็นข้อจำกัดของตัวแปรจำนวนเต็มบวก (Positive integer variables)

จุฬาลงกรณ์มหาวิทยาลัย
CHULALONGKORN UNIVERSITY

3.3 การแก้ปัญหาการจัดตารางการทำงานของพยาบาลด้วยวิธีฮิวริสติก

จากที่ได้กล่าวไปก่อนหน้านี้ ปัญหาการจัดตารางทำงานของพยาบาลเป็นที่ทราบกันดีว่าเป็นปัญหาเอ็นพีแบบยาก (NP Hard) ที่มีความซับซ้อนในการแก้ปัญหาอย่างมาก (Haspeslagh et al., 2014) การใช้แบบจำลองทางคณิตศาสตร์เป็นหนึ่งในวิธีแก้ปัญหาที่เป็นที่นิยม แต่เนื่องจากความซับซ้อนของปัญหาทำให้แบบจำลองทางคณิตศาสตร์มีข้อจำกัดทั้งด้านคุณภาพของคำตอบ และเวลาที่ใช้ในการประมวลผล ผู้วิจัยจึงได้เลือกวิธีฮิวริสติกสำหรับปรับปรุงคุณภาพของคำตอบในระยะเวลาที่สั้นลง ซึ่งในงานวิจัยนี้ผู้วิจัยให้ความสนใจไปที่การพัฒนา Adaptive Large Neighborhood Search (ALNS) ซึ่งเป็นอัลกอริทึมในการปรับปรุงคำตอบ และ Fast Non-Dominated Sorting (FNDS) ที่เป็นส่วนหนึ่งของ Non-dominated sorting genetic Algorithm III (NSGA-III) ที่จะเข้ามาช่วยให้สามารถสร้างตารางการทำงานที่มีความหลากหลายตามมิติต่างๆ ของแต่ละวัตถุประสงค์หลักในการจัดตาราง โดยการสร้างตารางการทำงานของพยาบาลจะมีขั้นตอนการทำงานดังรูปที่ 3-2



รูปที่ 3-2 ขั้นตอนการทำงานของฮิวริสติกการจัดตารางการทำงานของพยาบาล

การทำงานของโปรแกรมจะเริ่มจากนำเข้าข้อมูลการจัดตารางพยาบาล ประกอบไปด้วย จำนวนพยาบาลที่ต้องการในแต่ละวัน ของแต่ละตำแหน่ง ข้อมูลค่าจ้างพยาบาลในแต่ละกะ จำนวนพยาบาลแต่ละตำแหน่ง รวมถึงพารามิเตอร์ต่างๆที่ถูกใช้ในการดำเนินการของโปรแกรม จากนั้นจะทำการสร้างชุดคำตอบจำนวน n ชุด ซึ่งแต่ละชุดคำตอบจะได้มาจากการสุ่มพยาบาลในแต่ละตำแหน่งให้เพียงพอกับความต้องการพยาบาลในแต่ละ กะ ของแต่ละวัน โดยในขั้นตอนนี้จะไม่ให้ความสำคัญกับค่าวัตถุประสงค์ เพียงแต่ต้องการสร้างชุดคำตอบเบื้องต้นเท่านั้น หลังจากได้ชุดคำตอบเบื้องต้นชุดคำตอบทั้งหมดจะถูกนำไปเข้าขั้นตอนจัดกลุ่มของคำตอบตามค่าวัตถุประสงค์ในปัจจุบัน โดยจะผ่านขั้นตอน *FNDS* ที่ประกอบไปด้วยฟังก์ชัน *Front, Crowding distance, Reference points* และ *Niche count* เพื่อเป็นตัวกำหนดทิศทางในการปรับปรุงคำตอบให้กับ *ALNS* ที่จะดำเนินการโดยผ่านขั้นตอนหลักๆ 3 ขั้นตอน คือ การสุ่มใช้ตัวทำลาย, การสุ่มใช้ตัวซ่อมแซม และอัพเดทคะแนนจากวิธีการ Performance feedback ที่จะเป็นการเพิ่ม-ลด โอกาสในการเลือกใช้ตัวดำเนินการ หากคำตอบที่ผ่านการปรับปรุงในขั้นตอน *ALNS* เป็นคำตอบที่ดีกว่าเดิมคำตอบนั้นจะถูกยอมรับ และกำหนดให้เป็นชุดคำตอบใหม่ แต่หากไม่เป็นคำตอบที่ดีกว่าเดิมชุดคำตอบนั้นจะต้องผ่านขั้นตอนการยอมรับคำตอบด้วย *Simulated annealing (SA)* ก่อนถ้าคำตอบถูกยอมรับด้วย *SA* จะทำการกำหนดให้เป็นคำตอบใหม่ แทนชุดคำตอบเดิม ในทางกลับกันถ้าคำตอบไม่ถูกยอมรับเราจะยังใช้คำตอบเดิม ที่เป็นคำตอบก่อนหน้ากระบวนการปรับปรุงในรอบปัจจุบัน โดยคำตอบจาก *SA* ไม่ว่าจะถูกยอมรับหรือไม่ถูกยอมรับก็ตาม จะต้องเข้าสู่ขั้นตอนการตรวจสอบว่าเข้าข่ายเกณฑ์ในการหยุดทำงานหรือไม่ เพื่อคืนคำตอบทั้งหมดออกมา จึงเป็นอันเสร็จสิ้นกระบวนการ

3.3.1 การนำเข้าข้อมูล (Import Parameters)

กระบวนการนำเข้าข้อมูลเป็นขั้นตอนเตรียมการป้อนข้อมูลเบื้องต้นทั้งหมดที่จะใช้ในการสร้างตารางการทำงานของพยาบาล โดยข้อมูลเบื้องต้นจะประกอบไปด้วย จำนวนพยาบาลภายในแผนก จำนวนวันในสัปดาห์ ชนิดของกะการทำงาน จำนวนสัปดาห์ในกรอบการจัดตารางการทำงานพยาบาล จำนวนพยาบาลที่ต้องการในแต่ละวันและแต่ละกะการทำงาน อัตราค่าจ้างของพยาบาลในแต่ละกะ และสุดท้ายบทลงโทษเมื่อเกิดการละเมิดกฎการทำงาน แสดงขั้นตอนการนำเข้าข้อมูลดังภาพที่ 3-3

Pseudocode for Parameter Initialization

```
Import Basic Parameters:
    N ← Set of nurses
    D ← Set of days
    S ← Set of shifts
    Q ← Set of weeks
    R ← Required nurses
    C ← Nurse wage rates
    V ← Penalty values for violations
End import
```

รูปที่ 3-3 Pseudocode แสดงขั้นตอนการนำเข้าข้อมูลพื้นฐาน

3.3.2 การสร้างคำตอบเริ่มต้น (Initial Solution)

ในขั้นตอนการสร้างคำตอบเริ่มต้นจะเริ่มจากการมอบหมายกะการทำงานของพยาบาลแบบสุ่มมาเติมให้แต่ละกะการทำงานให้ได้จำนวนตามความต้องการพยาบาลในแต่ละวัน (num_nurses) แต่ละกะ (num_shifts) ของแต่ละสัปดาห์ (num_weeks) ซึ่งจำนวนชุดคำตอบที่ได้จะขึ้นอยู่กับจำนวนชุดคำตอบ (num_solutions) ที่กำหนดไว้ เช่น กำหนด num_solutions = 10 ก็จะได้ชุดคำตอบเริ่มต้นจำนวน 10 ชุดคำตอบ สำหรับชุดคำตอบเริ่มต้นที่ได้มาจะถูกนำมาคำนวณค่าของแต่ละวัตถุประสงค์ได้แก่ ค่าใช้จ่ายในการจ้างพยาบาลทั้งหมด ค่าความแตกต่างระหว่างค่าจ้างที่พยาบาลแต่ละคนได้รับ และค่าของบทลงโทษที่ละเมิดกฎการทำงาน ผ่านฟังก์ชัน *"evaluate"* โดยค่าของวัตถุประสงค์จะถูกนำไปใช้ต่อในขั้นตอนถัดไป นอกจากนี้ในการสร้างคำตอบเริ่มต้นจะถูกกำหนดว่าอย่างน้อยภายในแผนกจะต้องมีพยาบาลเพียงพอต่อความต้องการ มิเช่นนั้นจะไม่สามารถสร้างคำตอบเริ่มต้นได้ โดยในการสร้างคำตอบเริ่มต้นจะมีขั้นตอนการทำงานแสดงดังรูปที่ 3-4

```
Pseudocode for Generate Initial Solutions
Initialization: solutions, costs, balances, penalties as empty lists
for solution in 0 to num_solutions do
    for day in 0 to num_days do
        for shift in 0 to num_shifts do
            for week in 0 to num_weeks do
                required_nurses = num_reqs[day][shift][week]
                if required_nurses >0 then
                    available_nurses = list from 0 to num_nurses
                    if length of available_nurses >required_nurses then
                        Raise an exception "Insufficient nurses"
                    end
                    Random select "required_nurses" from available_nurses
                    for each selected nurse do
                        if nurse is already assigned a shift on the current
                          day and week then
                            continue
                        end
                        Assign the nurse to the current day, shift, and
                          week in the X array
                    end
                end
            end
        end
    end
    Evaluate the schedule using a function "evaluate"
    Append schedule and its data to respective lists
end
Return solutions, costs, balances, penalties
```

รูปที่ 3-4 Pseudocode แสดงขั้นตอนการสร้างชุดคำตอบเริ่มต้น (Initial solutions)

3.3.3 การสร้างคำตอบที่ดีที่สุด (Pareto Front)

แนวทางการสร้างชุดของคำตอบที่เหมาะสมที่สุด สำหรับปัญหาจัดตารางรางการทำงานพยาบาลแบบหลายวัตถุประสงค์ที่มีความขัดแย้งกัน มักนิยมใช้วิธีที่เรียกว่า Pareto front ซึ่งวิธีนี้สามารถช่วยจัดการกลุ่มของคำตอบที่ดีที่สุดได้หลายชุดคำตอบ โดยคำว่าชุดคำตอบที่ดีที่สุดในที่นี้หมายถึง ไม่มีชุดคำตอบคำตอบใดที่ดีกว่าชุดคำตอบอื่นในทุกด้าน (คำตอบไม่ถูกครอบงำโดยคำตอบอื่น) วิธีดังกล่าวจึงมีความเหมาะสมอย่างยิ่งสำหรับปัญหาการจัดตารางการทำงานพยาบาล ที่ต้องจัดการกับวัตถุประสงค์ที่มีความขัดแย้ง และช่วยสร้างทางเลือกในการ

เลือกใช้ตารางการทำงานที่เหมาะสมกับสถานการณ์ของแผนกฉุกเฉิน โดยการสร้าง Pareto front ในงานวิจัยนี้จะเลือกใช้วิธี Fast Non-Dominated Sorting (FNDS) เข้ามาช่วยจัดการให้ชุดคำตอบแตกต่างกันอย่างมีประสิทธิภาพ

3.3.4 Fast Non-Dominated Sorting (FNDS)

ในขั้นตอนการสร้างกลุ่มคำตอบที่ดีที่สุดในงานวิจัยนี้ ผู้วิจัยได้นำเอาวิธีการ Fast Non-Dominated Sorting (FNDS) ที่มีหน้าที่คัดเลือก และจัดอันดับของชุดคำตอบสำหรับปัญหาที่มีหลายวัตถุประสงค์ และเป็นส่วนหนึ่งของ Non-dominated Sorting Genetic Algorithm III (NSGA-III) ที่มีหลักการหาคำตอบที่ไม่ถูกครอบงำ (Non-dominated solution) ในพื้นที่หลายมิติ (Multi-dimensional space) โดยขั้นตอนในกระบวนการ *FNDS* จะเริ่มจากการนำชุดคำตอบทั้งหมดมาเปรียบเทียบกันเพื่อสร้าง *Fronts* สำหรับการแยกชุดคำตอบที่ไม่ถูกครอบงำ จากนั้นจะใช้เทคนิค *Crowding distance* ที่เป็นการคำนวณระยะห่างของคำตอบภายใน Front เดียวกันเพื่อทำให้ทราบถึงความห่างของแต่ละคำตอบ ส่วน *Reference points* จะเป็นการสร้างจุดที่กระจายตัวอยู่อย่างสม่ำเสมอเพื่อใช้อ้างอิงตำแหน่ง และสุดท้าย *Niche count* จะเป็นการหาว่าคำตอบที่กำลังคำนวณ ณ ปัจจุบันนั้นควรจะเลือกให้คำตอบนี้ถูกปรับปรุงไปในทิศทางใดจึงจะมีแนวโน้มที่คำตอบจะถูกปรับปรุงได้รวดเร็วที่สุด โดยขั้นตอนการทำงานของ *FNDS* จะมีขั้นตอนแสดงดังรูปที่ 3-5

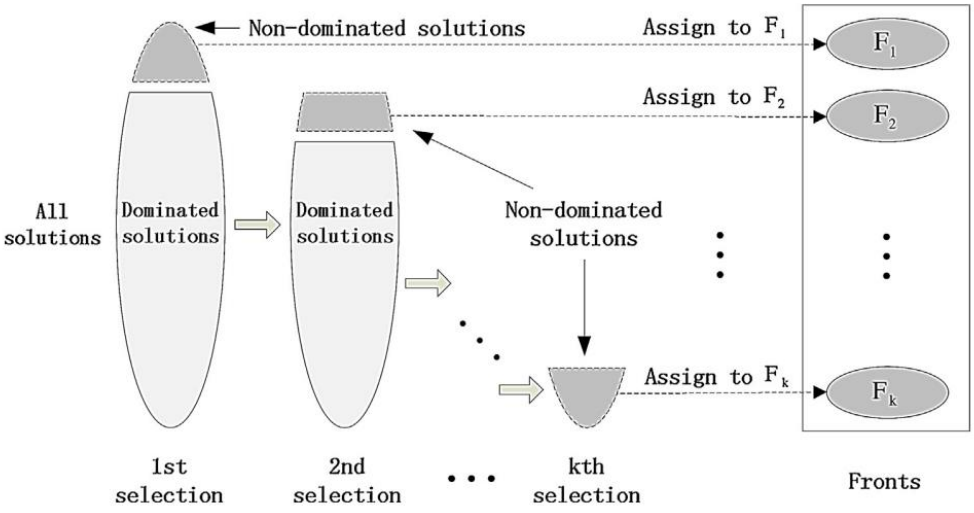
Pseudocode for Fast Non-Dominated Sorting

```
Function dominates(solution1, solution2)
    not_worse ← ALL((a ≤ b) for a, b in zip(solution1, solution2))
    better ← ANY((a < b) for a, b in zip(solution1, solution2))
    return not_worse and better
Function Front(solutions)
    fronts ← Initialize an empty list to store the sorted fronts.
    foreach solution1 in solutions do
        foreach solution2 in solutions do
            if dominates(solution1, solution2) then
                Add solution1 to the current front.
    return fronts
Function Calculate_Crowding_Distance(fronts, objectives)
    for each front in fronts do
        Calculate the crowding distance for each solution in the front.
    return Crowding distance
Function generate_reference_points(num_objectives, num_divisions)
    reference_points ← Create the reference points based on the number
      of objectives and divisions.
    return Reference points
Function Calculate_Niche_Count(solutions, reference_points, objectives)
    niche_count ← Calculate the niche count for each solution based on
      reference points.
    return Niche count for each solution
Function FastNonDominatedSort(solutions)
    Fronts ← Front(solutions)
    Crowding_distance ← Calculate_Crowding_Distance(fronts,
      objectives)
    Reference_points ← generate_reference_points(number_of_objectives,
      number_of_divisions)
    Niche_count ← Calculate_Niche_Count(solutions, reference_points,
      objectives)
    return Fronts, Crowding_distance, Niche_count
```

รูปที่ 3-5 Pseudocode แสดงขั้นตอน Fast Non-Dominated Sorting

3.3.4.1 Front

หลักการที่สำคัญในการทำงานของ Front คือการจัดกลุ่มของคำตอบตามลำดับความสำคัญ โดย Front แรกจะเป็นชุดคำตอบที่ดีที่สุดจากคำตอบที่มีทั้งหมดในปัจจุบัน จะประกอบด้วยคำตอบที่ไม่ถูกครอบงำจากคำตอบใดๆ และ Front ถัดไปจะประกอบด้วยคำตอบที่ถูกครอบงำเพียงคำตอบที่ถูกจัดอยู่ใน Front ก่อนหน้าทั้งหมด แต่ไม่ถูกครอบงำจากคำตอบที่เหลือ ในการแบ่ง Front จะดำเนินการไปจนถึง K-Front (ทุกคำตอบแบ่งให้อยู่ใน Front จนครบ) การมี Front จะช่วยให้สามารถระบุ และเลือกคำตอบที่ดีที่สุดสำหรับปัญหาที่มีหลายวัตถุประสงค์ (Multi-objective) ได้อย่างมีประสิทธิภาพ ตัวอย่างการแบ่ง Front แสดงดังรูปที่ 3-6



รูปที่ 3-6 ตัวอย่างการแบ่ง Front ด้วยการตรวจสอบคำตอบที่ไม่ถูกครอบงำ (Bao et al., 2017)

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

ในการทำงานของ Front ประกอบด้วย 2 ฟังก์ชัน ได้แก่ ฟังก์ชัน *Dominate* ที่ทำหน้าที่เปรียบเทียบสองคำตอบว่าคำตอบแรกดีกว่าคำตอบที่สองหรือไม่ โดยมีเงื่อนไขคือ คำตอบแรกจะต้องมีประสิทธิภาพของวัตถุประสงค์ไม่แย่ไปกว่าคำตอบที่สองในทุกมิติ และต้องดีกว่าอย่างน้อยมิติใดมิติหนึ่ง ส่วนที่สองคือ ฟังก์ชัน *Sorted front* ฟังก์ชันนี้เป็นเสมือนหัวใจของ *Fronts* ที่จะทำหน้าที่จัดเรียงคำตอบให้เป็นหลายๆ Front เริ่มต้นด้วยการสร้างลิสต์ว่าง (Empty list) เพื่อเก็บข้อมูลการเปรียบเทียบของทุกคำตอบ เมื่อทุกคำตอบถูกเปรียบเทียบครบทั้งหมดแล้ว โปรแกรมจะสร้างกลุ่มของคำตอบที่เรียกว่า Fronts คำตอบที่ไม่ถูกครอบงำโดยคำตอบอื่นๆเลยจะถูกจัดเป็นกลุ่มแรกเมื่อไม่มีสมาชิกของ Front แรกเพิ่มแล้วโปรแกรมจะสร้าง Front ที่สองและหาคำตอบที่ไม่ถูกครอบงำ และทำการเพิ่มเข้าไปใน Front ที่สอง ขั้นตอนนี้จะทำซ้ำจนกระทั่ง solution ทั้งหมดถูกจัดให้เข้ากลุ่มต่างๆตามลำดับของการไม่ถูกครอบงำได้ครบถ้วน และท้ายที่สุดจะทำการคืนค่า fronts ออกมาเพื่อใช้ในการดำเนินการขั้นตอนต่อๆไป แสดงขั้นตอนดังรูปที่ 3-7

```
Pseudocode for Fronts
Function dominates(solution1, solution2)
    not_worse ← ALL(a ≤ b for a, b in zip(solution1, solution2))
    better ← ANY(a < b for a, b in zip(solution1, solution2))
    return not_worse and better
Function sorted_fronts(solutions)
    n_solutions ← length(solutions)
    dominates_counter ← [0] * n_solutions
    dominated_by ← [[] for i in range(n_solutions)]
    fronts ← []
    for i in range(n_solutions) do
        for j in range(i+1, n_solutions) do
            if dominates(solutions[i], solutions[j]) then
                append j to dominated_by[i]
                increment dominates_counter[j]
            end
            else if dominates(solutions[j], solutions[i]) then
                append i to dominated_by[j]
                increment dominates_counter[i]
            end
        end
    end
    current_front ← [i for i, count in enumerate(dominates_counter) if
     count == 0]
    while current_front do
        next_front ← []
        for i in current_front do
            for j in dominated_by[i] do
                decrement dominates_counter[j]
                if dominates_counter[j] == 0 then
                    append j to next_front
                end
            end
        end
        append current_front to fronts
        current_front ← next_front
    end
    return fronts
```

รูปที่ 3-7 Pseudocode แสดงขั้นตอน Fronts

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

3.3.4.2 Crowding Distance

Crowding distance เป็นการประเมินความหนาแน่นของคำตอบที่ได้ในพื้นที่คำตอบทั้งหมด (Solution space) เพื่อทำให้เกิดการกระจายตัวของคำตอบทั้งหมดบน Pareto front ได้อย่างสม่ำเสมอ ส่งผลให้คำตอบไม่อยู่ติดกัน หรือกระจายตัวห่างจากันมากเกินไป ซึ่งค่าของ Crowding distance จะสามารถคำนวณได้จากระยะห่างของแต่ละคำตอบที่อยู่ภายใน Front เดียวกันหากคำตอบใดมีค่า Crowding distance มากจะแสดงว่าคำตอบนั้นมีความคล้ายคลึงกับคำตอบอื่นน้อย ในทางตรงกันข้าม หากมีค่าน้อยแสดงว่าคำตอบนั้นมีความคล้ายคลึงกับคำตอบอื่นมากควรได้รับการปรับปรุง กระบวนการเหล่านี้จะช่วยให้กลุ่มคำตอบใน Pareto front มีคำตอบที่มีความหลากหลาย และเกิดประสิทธิภาพที่ดีขึ้น สำหรับการคำนวณ Crowding distance สามารถคำนวณได้จากสมการที่ (11)

$$\mathbf{CD_{New}} = \mathbf{CD_{Current}} + \left(\frac{\mathbf{Value^{m}_{i+1}} - \mathbf{Value^{m}_{i-1}}}{\mathbf{Value^{m}_{Max}} - \mathbf{Value^{m}_{Min}}}\right) \qquad (11)$$

$\mathbf{CD_{New}}$ = ค่า Crowding Distance จากการคำนวณล่าสุด

$\mathbf{CD_{Current}}$ = ค่า Crowding Distance ของปัจจุบัน

$\mathbf{Value^{m}_{i+1}}$ = ค่าวัตถุประสงค์ที่สูงกว่าค่าปัจจุบันหนึ่งลำดับภายใน Front ของวัตถุประสงค์ที่ m

$\mathbf{Value^{m}_{i-1}}$ = ค่าวัตถุประสงค์ที่ต่ำกว่าค่าปัจจุบันหนึ่งลำดับภายใน Front ของวัตถุประสงค์ที่ m

$\mathbf{Value^{m}_{Max}}$ = ค่าวัตถุประสงค์ที่สูงที่สุดในปัจจุบันภายใน Front ของวัตถุประสงค์ที่ m

$\mathbf{Value^{m}_{Min}}$ = ค่าวัตถุประสงค์ที่ต่ำที่สุดในปัจจุบันภายใน Front ของวัตถุประสงค์ที่ m

การคำนวณ Crowding distance จะเริ่มจากการกำหนดให้ Crowding distance มีค่าเท่ากับ 0 สำหรับทุกคำตอบ จากนั้นคำนวณค่าวัตถุประสงค์ในแต่ละมิติ และทำการจัดเรียงตามค่าวัตถุประสงค์ภายใน Front เดียวกัน เพื่อทำการเรียงลำดับค่าวัตถุประสงค์จากมากไปน้อย โดยกำหนดค่า Crowding distance ให้กับค่าวัตถุประสงค์ที่มีค่าต่ำสุด และสูงสุดให้เป็นค่าอนันต์ สำหรับคำตอบที่ไม่ใช่คำตอบที่มีค่าวัตถุประสงค์ต่ำสุด และสูงสุดจะนำมาคำนวณหาค่า Crowding distance ตามปกติจนครบทุกคำตอบ จากนั้นฟังก์ชัน *Crowding distance* จะส่งคืนค่า Crowding distance ออกไปใช้งานขั้นตอนถัดไป ในส่วนของขั้นตอนการทำงาน โดยละเอียดของฟังก์ชัน *Crowding distance* สามารถดูได้จากรูปที่ 3-8

Pseudocode for Calculate Crowding Distance

```
   Data: fronts, objectives
 1 num_objs ← length(objectives)
 2 distances ← [0.0] * length(front)
 3 for obj_index, obj in enumerate(objectives) do
 4     sorted_indices ← sorted(0 to length(front)-1, key =
         front[k][obj_index])
 5     distances[sorted_indices[0]] ← ∞
 6     distances[sorted_indices[length(front)-1]] ← ∞
 7     objective_range ← front[sorted_indices[length(front)-1]][obj_index] -
         front[sorted_indices[0]][obj_index]
 8     for i in 1 to length(front)-2 do
 9         if objective_range == 0 then
10             distances[sorted_indices[i]] ← distances[sorted_indices[i]] + 0
11         end
12         else
13             distances[sorted_indices[i]] ← distances[sorted_indices[i]] +
                 (front[sorted_indices[i+1]][obj_index] -
                 front[sorted_indices[i-1]][obj_index]) / objective_range
14         end
15     end
16 end
17 return Crowding distance
```

รูปที่ 3-8 Pseudocode แสดงขั้นตอนคำนวณ Crowding distance

3.3.4.3 Reference Points

การสร้างจุดอ้างอิง (Reference points) เป็นการกำหนดค่า หรือจุดสำหรับใช้เป็นตำแหน่งอ้างอิงในการแก้ปัญหาที่มีหลายวัตถุประสงค์ โดยจุดอ้างอิงทุกจุดจะถูกสร้างขึ้นมาล่วงหน้า และถูกใช้ตั้งแต่เริ่มต้นกระบวนการจนจบขั้นตอนการปรับปรุงคำตอบ สำหรับในการวิจัยนี้จะนำวิธีการเชิงระบบของ Das and Dennis (1998) ในการกำหนด Hyperplane ที่จะมีกลุ่มของจุดอ้างอิงอยู่บนจุดตัดของ Hyperplane นั้นๆ โดยแต่ละจุดจะประกอบไปด้วยเวกเตอร์ (Vector) ที่เริ่มจากจุดกำเนิดเชื่อมต่อไปยังแต่ละจุดอ้างอิง โดยวิธีการคำนวณหาจุดอ้างอิง (Reference points) นั้นสามารถคำนวณได้จากสูตร (12) (Krairiksh et al., 2022)

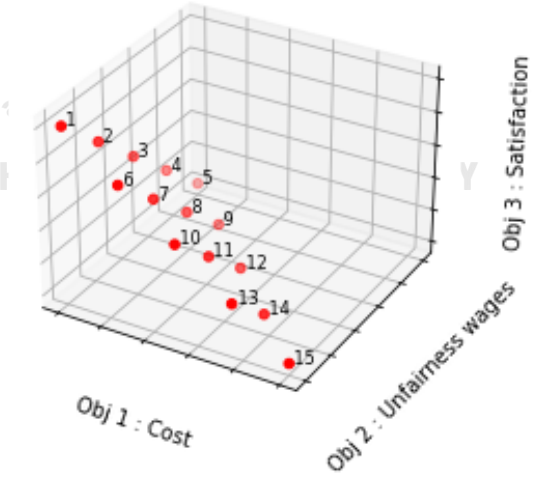
$$N = \frac{(m + p - 1)!}{(m - 1)!\, p!} \qquad (12)$$

N = จำนวนจุดอ้างอิง (Number of reference point)

m = จำนวนวัตถุประสงค์ (Number of objective)

p = จำนวน Division plane

ในงานวิจัยนี้จะเป็นปัญหาแบบ 3 วัตถุประสงค์ (m = 3) และได้เลือกให้มีจำนวน Division = 4 (p = 4) เพราะฉะนั้นจุดอ้างอิงจะกระจายตัวอยู่ใน Objective space ซึ่งจะกระจายตัวแบบ Uniform distribution อยู่ภายในแกน x y z พิกัด (0-1,0-1,0-1) จำนวน 15 จุดดังรูปที่ 3-9 และตารางที่ 3-2



รูปที่ 3-9 จุดอ้างอิง (Reference point) บนพื้นที่ของ Objective space

ตารางที่ 3-2 ตารางแสดงพิกัดของจุดอ้างอิง (Reference point)

จุดอ้างอิง	พิกัดแกน x	พิกัดแกน y	พิกัดแกน z
Point 1	0.00	0.00	1.00
Point 2	0.00	0.25	0.75
Point 3	0.00	0.50	0.50
Point 4	0.00	0.75	0.25
Point 5	0.00	1.00	0.00
Point 6	0.25	0.00	0.75
Point 7	0.25	0.25	0.50
Point 8	0.25	0.50	0.25
Point 9	0.25	0.75	0.00
Point 10	0.50	0.00	0.50
Point 11	0.50	0.25	0.25
Point 12	0.50	0.50	0.00
Point 13	0.75	0.00	0.25
Point 14	0.75	0.25	0.00
Point 15	1.00	0.00	0.00

ขั้นตอนการสร้างจุดอ้างอิงจะเริ่มจากการนำเข้ามูลเข้าหลักๆ 2 ค่าคงที่ คือ จำนวนวัตถุประสงค์ และจำนวน Division เข้ามาคำนวณด้วยสูตรที่ (12) เพื่อหาจำนวนจุดอ้างอิง (Reference point) และบ่งบอกค่าพิกัดแต่ละจุดที่กระจายตัวอยู่ในพื้นที่วัตถุประสงค์ (Objective space) โดยสามารถดูขั้นตอนการทำงานได้จากรูปที่ 3-10

จุฬาลงกรณ์มหาวิทยาลัย

Pseudocode for Generate Reference Points

```
1  Function generate_reference_points(num_objectives,
   num_divisions)
2      Function recursive(points, num_obj, num_div, left, point, index)
3          if index == num_obj - 1 then
4              point[index] = left / num_div
5              points.append(point.copy())
6              return
7          end
8          for i in 0 to left do
9              point[index] = i / num_div
10             recursive(points, num_obj, num_div, left - i, point, index +
                1)
11         end
12     points = []
13     point = [0] * num_objectives
14     recursive(points, num_objectives, num_divisions, num_divisions,
        point, 0)
15     points = Reference points
16     return Reference points
```

รูปที่ 3-10 Pseudocode แสดงขั้นตอนการสร้างจุดอ้างอิง (Reference points)

3.3.4.4 Niche Count

ก่อนหน้านี้เราได้ทำการกำหนดจุด Reference points ที่กระจายตัวอยู่ใน Objective space หากแต่ยังไม่ระบุวิธีที่จะแจกแจงแต่ละชุดคำตอบให้กับแต่ละจุดใน Reference points ซึ่งขั้นตอนที่จะแจกแจงคำตอบจะสามารถทำได้ใน *Niche count* ซึ่งมีขั้นตอนที่ประกอบไปด้วย

- การปรับอัตราส่วนของค่าวัตถุประสงค์ (Normalized objective function) : เป็นการปรับอัตราส่วนของค่าวัตถุประสงค์ให้อยู่ในค่าระหว่าง 0 – 1 เพื่อให้ค่าในแต่ละวัตถุประสงค์อยู่ในอัตราส่วนที่เท่ากัน ในการปรับอัตราส่วนของงานวิจัยนี้จะใช้วิธีที่เรียกว่า Min – Max normalization ซึ่งสามารถคำนวณได้จากสมการที่ (13)

$$\text{Normalized Objective Value} = \frac{\text{Value}_{\text{Current}} - \text{Value}_{\text{Min}}}{\text{Value}_{\text{Max}} - \text{Value}_{\text{Min}}} \qquad (13)$$

Normalized Objective Value = ค่าวัตถุประสงค์ในอัตราส่วน 0 – 1

$\text{Value}_{\text{Current}}$ = ค่าวัตถุประสงค์ปัจจุบัน

$\text{Value}_{\text{Min}}$ = ค่าวัตถุประสงค์ที่ต่ำที่สุดในปัจจุบัน

$\text{Value}_{\text{Max}}$ = ค่าวัตถุประสงค์ที่สูงที่สุดในปัจจุบัน

- ระยะทางยูคลิด (Euclidean distance) : การหาระยะทางยูคลิดสำหรับแต่ละคำตอบจะทำให้ทราบระยะทางระหว่างคำตอบ และ Reference point ซึ่งการหาระยะห่างนี้จะช่วยให้สามารถแจกแจงคำตอบให้กับแต่ละ Reference point ได้อย่างมีประสิทธิภาพจากการนำระยะทางยูคลิดมาเลือกกำหนด Reference point ให้แต่ละคำตอบ โดยจะสามารถคำนวณระยะทางยูคลิดได้จากสมการที่ (14)

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

$$D(S_i, R_j) = \sqrt{(x_{R_j} - x_{S_i})^2 + (y_{R_j} - y_{S_i})^2 + (z_{R_j} - z_{S_i})^2} \qquad (14)$$

$D(S_i,R_j)$ = ระยะห่างระหว่างคำตอบที่ i และจุด Reference point ที่ j

X_{Si} = พิกัดแกน X ของคำตอบที่ i

Y_{Si} = พิกัดแกน Y ของคำตอบที่ i

Z_{Si} = พิกัดแกน Z ของคำตอบที่ i

X_{Rj} = พิกัดแกน X ของ Reference point ที่ j

Y_{Rj} = พิกัดแกน Y ของ Reference point ที่ j

Z_{Rj} = พิกัดแกน Z ของ Reference point ที่ j

ฟังก์ชันการทำงานของ *Niche count* จะมีขั้นตอนดังรูปที่ 3-11 ซึ่งจะเป็นการนำจุดอ้างอิง และค่าวัตถุประสงค์ของคำตอบแต่ละชุดในปัจจุบันเข้ามาคำนวณเพื่อหาทิศทางในการปรับปรุงแต่ละคำตอบ ในขั้นตอนนี้จะมีการปรับอัตราส่วนของค่าวัตถุประสงค์ โดยทำการหาขอบล่าง และขอบบนของแต่ละวัตถุประสงค์ เพื่อนำมาใช้เป็นข้อมูลในการคำนวณตามสมการที่ (13) หลังจากทำการการปรับอัตราส่วนของค่าวัตถุประสงค์ (Normalized objective function) แล้วจะเริ่มนำข้อมูลของแต่ละชุดคำตอบที่ถูกปรับอัตราส่วนมาคำนวณระยะทางยูคลิดจาก (14) โดยแต่ละตัวจะมีค่าระยะทางยูคลิด 15 ข้อมูล เนื่องจากการคำนวณระยะทางจะเป็นการหาระยะระหว่างคำตอบที่สนใจไปยังจุดอ้างอิงทั้ง 15 จุด ภายหลังที่คำนวณค่า Niche count ครบทุกตัวแล้วจะทำการคืนค่า Niche count เพื่อใช้ในขั้นตอนถัดไป

Pseudocode for Calculate Niche Count

```
Data: solutions, reference_points, objectives
objective_names = ["Cost", "Unfairwages", "Satisfaction"]
min_objectives = np.min(objectives, axis=0)
max_objectives = np.max(objectives, axis=0)
norm_solutions = []
for sol in solutions do
    norm_sol = [(sol[i] - min_objectives[i]) / (max_objectives[i] -
      min_objectives[i]) for i in range(len(sol))]
    norm_solutions.append(norm_sol)
end
niche_count_matrix = []
for norm_sol in norm_solutions do
    niche_counts_for_this_solution = []
    for j, ref_point in enumerate(reference_points) do
        differences = np.array(norm_sol) - np.array(ref_point)
        x_diff = differences[0]
        y_diff = differences[1]
        z_diff = differences[2]
        distance = np.sqrt(x_diff^2 + y_diff^2 + z_diff^2)
        niche counts for each solution.append(distance)
    end
    Niche count.append(niche counts for each solution)
end
return Niche count
```

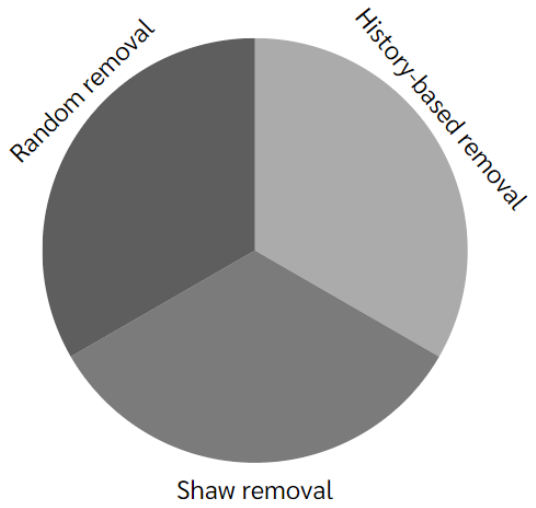
รูปที่ 3-11 Pseudocode แสดงขั้นตอนคำนวณ Niche count

CHULALONGKORN UNIVERSITY

3.3.5 การปรับปรุงคุณภาพคำตอบด้วย ALNS

Adaptive Large Neighborhood Search เป็นอัลกอริทึมแบบสำรวจจากความเป็นไปได้ในการพัฒนาคำตอบในพื้นที่ค้นหา (Search space) โดยจะเลือกทำลายคำตอบบางส่วนของคำตอบปัจจุบัน เพื่อสร้าง Neighborhood ของคำตอบที่เป็นไปได้ในรอบถัดไป จากนั้นจะเลือกซ่อมแซมคำตอบใน Neighborhood เพื่อปรับปรุงให้คำตอบให้มีค่าวัตถุประสงค์ที่ดีขึ้น หากคำตอบที่ได้ถูกปรับปรุงให้ให้ดีขึ้น หรือแย่ลงจะมีการปรับปรุงค่าน้ำหนักของตัวดำเนินการ (Operator weights updating) ทั้งตัวทำลายคำตอบ (Destroy operator) และตัวซ่อมแซมคำตอบ (Repair operator) เพื่อเพิ่ม-ลดโอกาสในการเลือกใช้ในครั้งต่อไป ซึ่ง ALNS จะทำขั้นตอนต่างๆ ซ้ำไปจนกระทั่งเข้าเกณฑ์ในการหยุดทำงาน (Stopping criteria)

การทำงานของ ALNS ในงานวิจัยนี้เป็นวิธีที่ใช้ในการปรับปรุงคุณภาพของคำตอบ โดยทำงานร่วมกับฟังก์ชัน *FNDS* เพื่อจำแนกคำตอบให้ถูกปรับปรุงไปในทิศทางที่เหมาะสมของแต่ละชุดคำตอบในปัจจุบัน ซึ่งจะใช้ข้อมูลจากฟังก์ชัน *FNDS* ได้แค่ Crowding distance และ Niche count ในตอนเริ่มแรกของการปรับปรุงคำตอบจะเริ่มจากนำคำตอบผ่านการทำลายคำตอบบางส่วน ซึ่งการเลือกตัวทำลายคำตอบนั้นจะอาศัยการสุ่มด้วยวิธี Roulette wheel โดยตัวทำลายคำตอบไม่ว่าจะเป็น *Random removal, Shaw removal* หรือ *History-based removal* จะมีโอกาสที่จะถูกเลือกใช้งานอย่างเท่าเทียมกัน แสดงดังรูปที่ 3-12 หลังจากนั้นคำตอบที่ผ่านขั้นตอนการทำลายคำตอบ จะถูกนำเข้ามาซ่อมแซมคำตอบ เพื่อซ่อมแซมคำตอบที่ถูกทำลายไป ซึ่งวิธีในการเลือกตัวซ่อมแซมคำตอบจะใช้เทคนิค และรูปแบบเหมือนกับตัวทำลายคำตอบ คือ การสุ่มด้วยวิธี Roulette wheel ถึงแม้ว่าในรอบการปรับปรุงรอบแรกเรากำหนดให้ตัวทำลายคำตอบ และตัวซ่อมแซมคำตอบ จะมีโอกาสที่จะถูกเลือกใช้ที่เท่ากัน แต่ภายหลังจากผ่านการการทำลาย และซ่อมแซมคำตอบแล้ว ประสิทธิภาพของตัวดำเนินการจะถูกสะท้อนออกมาในรูปคะแนนสะสม และถูกนำไปคำนวณค่าความน่าจะเป็น หากตัวดำเนินการตัวใดมีประสิทธิภาพในการปรับปรุงคุณภาพของคำตอบไปในทิศทางที่ดี ก็จะเป็นการเพิ่มโอกาสที่จะถูกเลือกใช้สูงขึ้น ยกตัวอย่างเช่น ตัวทำลาย Shaw removal สามารถทำลายคำตอบ แล้วประสิทธิภาพดีขึ้นจะส่งผลให้ Shaw removal ได้คะแนนสะสมเพิ่มขึ้น และจะมีสัดส่วนในวงกลมมากขึ้น ซึ่งทำให้มีโอกาสในการถูกเลือกสูงขึ้นจากเดิม แต่เพื่อป้องกันไม่ให้ตัวดำเนินการบางตัวมีคะแนนสะสมมากเกินไปจนทำให้ตัวดำเนินการตัวอื่นๆ ไม่ถูกเลือกใช้ หรือมีโอกาสที่จะถูกเลือกใช้น้อย จึงจะต้องกำหนดเกณฑ์ในการรีเซ็ต (Reset) คะแนนสะสม ซึ่งสามารถทำได้ด้วยการกำหนดจำนวนรอบที่จะทำการรีเซ็ต (Reset) ในที่นี้ คือ Segment ที่เมื่อจำนวนรอบในการปรับปรุงครบจำนวน Segment โปรแกรมจะทำการรีเซ็ตคะแนนให้กลับไปเท่ากับ 0 ในส่วนสุดท้ายของโปรแกรม คือ การกำหนดเกณฑ์ในการหยุดทำงาน (Stopping criterion) จากนั้นฟังก์ชันจะส่งคืนค่าคำตอบที่ถูกปรับปรุงออกไปเข้าขั้นตอนยอมรับคำตอบต่อไป โดยการทำงานทั้งหมดของ ALNS สามารถดู Pseudocode ได้จากรูปที่ 3-13



รูปที่ 3-12 ตัวอย่างความน่าจะเป็นในการสุ่มตัวทำลายด้วยวิธี Roulette wheel

Pseudocode for Adaptive Large Neighborhood Search (ALNS)

Input: Fronts, Crowding distance, reference points, niche count
Initialize: Solutions set with 15 solutions
Set initial weights = 0 for each operator
Number of Iteration: Maximum number of search iterations for ALNS
Max Unimproved Iteration: Maximum number of search iterations without improvement before stopping ALNS
Segment: Number of iterations in each search segment (weights are reset after each segment)
for each solution in Solutions:
    Evaluate the solution's priority using Crowding distance and Niche count
end for
Sort Solutions based on their priority
Initialize: currentIteration = 0, unimprovedIteration = 0
while currentIteration < Number of Iteration **and** unimprovedIteration < Max Unimproved Iteration **do**
    if currentIteration % Segment == 0 **then**
        Reset operator weights
    end if
    Select a *destroy operator* based on roulette wheel selection using probabilities from:
        - Random removal
        - Shaw removal
        - History-based removal
    Apply destroy operator on current solution
    Select a *repair operator* based on roulette wheel selection using probabilities from:
        - Greedy insertion
        - Regret based insertion
        - Least loaded nurse repair
    Apply repair operator on destroyed solution
    Evaluate the solution using Fronts, Crowding distance, reference points, niche count
    Update the weights of operators using:

$$W_{i(t+1)} = W_{it} + \alpha(P_{it} - P_{avg_t})$$

where,

$$P_i = \frac{W_i}{\sum_{j \in J} W_j}$$

    Increase the probability of selecting an operator based on its updated weight
    currentIteration++
    if solution improved **then** unimprovedIteration = 0
    else unimprovedIteration++
end while
Output: Pareto front

รูปที่ 3-13 Pseudocode แสดงขั้นตอนการทำงานของ Adaptive Large Neighborhood Search (ALNS)

3.3.5.1 การเลือกตัวทำลายคำตอบ (Destroy Operator Selection)

ขั้นตอนการเลือกตัวทำลายคำตอบเป็นขั้นตอนที่มุ่งเน้นการทำลาย หรือแก้ไขคำตอบในปัจจุบันเพื่อขยายกรอบพื้นที่ในการค้นหาคำตอบใหม่ๆ ที่ทำให้สามารถปรับปรุงค่าวัตถุประสงค์ (Objective function) ได้สำหรับงานวิจัยนี้จะอาศัยหลักการในการเลือกรูปแบบการทำลายคำตอบ คือ 1.สร้างความหลากหลาย และขยายพื้นที่การสำรวจ (Diversity and exploration) 2.เพิ่มความสมดุล และประสิทธิภาพ (Balance and efficiency)

สำหรับวิธีการทำลายคำตอบในการปรับปรุงคำตอบของ ALNS จะสามารถแบ่งได้ 3 วิธีหลักๆ ได้แก่ Random removal, Shaw removal และ History-based removal แสดงดังรูปที่ 3-14 โดยการทำลายของทั้ง 3 วิธี จะทำการเลือกจำนวนคำตอบที่จะเลือกทำลาย ในที่นี้จำนวนคำตอบที่จะทำลายจะเป็นการทำลายจำนวน กะการทำงานจากพยาบาลที่อยู่ในแผนก เช่น ทำลายกะเช้า ในวันที่ d สัปดาห์ที่ k ของพยาบาลที่ n จะเท่ากับการทำลายคำตอบ 1 ครั้ง ซึ่งจำนวนการทำลายคำตอบ จะมีชื่อเป็นทางการว่า “Neighborhood size” สำหรับในงานวิจัยนี้จะกำหนด Neighborhood size = 12

รูปที่ 3-14 วิธีในการทำลายคำตอบ (Destroy operator)

3.3.5.2 การเลือกทำลายคำตอบแบบ Random Removal

การเลือกทำลายคำตอบแบบสุ่ม (Random removal) เป็นหนึ่งในเทคนิคในการเลือกทำลายคำตอบใน Adaptive Large Neighborhood Search (ALNS) ซึ่งการทำลายคำตอบแบบสุ่มนั้นจะทำการสุ่มเลือกบางส่วนของคำตอบปัจจุบันโดยการลบออกจากคำตอบ เพื่อให้เกิดคำตอบใหม่ที่มีโครงสร้าง หรือลักษณะที่แตกต่างไปจากเดิม การสุ่มนี้มีความสำคัญในการสร้างความหลากหลายในการทำลายคำตอบ และเพิ่มโอกาสในการสร้างคำตอบที่มีคุณภาพในขั้นตอนซ่อมแซมในลำดับต่อไป นอกจากการทำลายคำตอบแบบสุ่มจะช่วยเพิ่มคุณภาพของคำตอบแล้วยังช่วยให้ขอบเขตการค้นหาคำตอบของอัลกอริทึมกว้างขึ้น และช่วยลดความน่าจะเป็นที่คำตอบจะติดอยู่ใน Local minimum ซึ่งวิธีการทำลายคำตอบแบบสุ่มนั้นเป็นวิธีที่ทำได้ง่าย และได้รับความนิยมในการค้นหาคำตอบใหม่ๆที่อยู่ใน Objective space โดยขั้นตอนการเลือกทำลายคำตอบแบบสุ่มสามารถดูได้จากรูปที่ 3-15

จุฬาลงกรณ์มหาวิทยาลัย
CHULALONGKORN UNIVERSITY

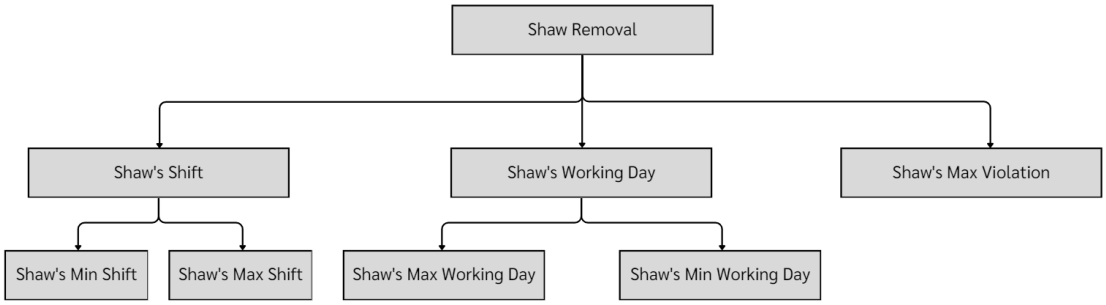
Pseudocode for Random Removal

```
1 Input: Solution, Neighborhood size (n)
2 RandomRemoval(Solution, n) begin
3     for i = 1 to n do
4         Select a random element from the solution.
5         Destroy the selected element from Solution
6     end
7     return Modified Solution
8 end
```

รูปที่ 3-15 Pseudocode แสดงขั้นตอนการทำงานของ Random removal

3.3.5.3 การเลือกทำลายคำตอบแบบ Shaw Removal

เทคนิค Shaw removal เป็นวิธีที่มุ่งเน้นไปที่การทำลายคำตอบ ที่เลือกเฉพาะส่วนของคำตอบที่มีคุณสมบัติคล้ายคลึงกันมาเป็นตัวเลือกในการทำลายคำตอบ เพื่อให้ในขั้นตอนการซ่อมแซมคำตอบสามารถทำทำได้ง่าย และแม่นยำมากขึ้น ซึ่งทำให้สามารถประหยัดเวลา และทรัพยากร ในขณะที่ยังคงได้คำตอบที่มีประสิทธิภาพ สำหรับในงานวิจัยนี้ได้นำวิธีการทำลายคำตอบแบบ Shaw ทั้งหมด 3 วิธี คือ Shaw's min-max - shift, Shaw's min-max - working day และ Shaw's max violation แสดงดังรูปที่ที่ 3-16 โดยการเลือกใช้ตัวทำลายคำตอบ Shaw จะใช้วิธี Roulette wheel เป็นเทคนิคเดียวกับการเลือกตัวทำลายคำตอบก่อนหน้า



รูปที่ 3-16 ชนิดของตัวทำลายคำตอบแบบ Shaw's

- การทำลายคำตอบแบบ Shaw's min shift : เป็นการเลือกทำลายคำตอบจากการคำนวณปริมาณกะการทำงานของพยาบาลแต่ละคนในชุดคำตอบเพื่อหาพยาบาลที่ได้รับจำนวนกะน้อยที่สุด โดยพยาบาลที่มีกะน้อยที่สุดจะถูกทำลายได้ไม่เกิน 1 กะการทำงานในที่นี้เพื่อป้องกันความไม่สมดุลในการทำลายกะการทำงานของพยาบาลที่มีจำนวนกะน้อยอยู่แล้ว จากนั้นจะทำการลบกะการทำงานจากพยาบาลคนอื่นๆ ที่มีจำนวนกะน้อยที่สุดรองลงมาจนครบจำนวน Neighborhood size และทำการคืนคำตอบที่ถูกทำลายกลับมา แสดงดังรูปที่ 3-17

CHULALONGKORN UNIVERSITY

Pseudocode for Shaw Removal with Min Shift

```
1  Input: Solution, Neighborhood size (n), Nurse selection counts
2  Initialize: nurseSelectionCount for each nurse in Solution to 0
3  ShawRemovalMinShift(Solution, n) begin
4      for i = 1 to n do
5          Determine element with minimum shift in Solution
6          if nurseSelectionCount[element] < 1 then
7              Remove the determined element from Solution
8              nurseSelectionCount[element] += 1
9          end
10     end
11     return Modified Solution
12 end
```

รูปที่ 3-17 Pseudocode แสดงขั้นตอนการทำงานของ Shaw's min shift

- การทำลายคำตอบแบบ Shaw's max shift : ทำได้โดยเลือกกะการทำงาน จากพยาบาลที่มีกะการทำงานมากที่สุดจากการคำนวณปริมาณกะการทำงานของพยาบาลแต่ละคน กระบวนการนั้นจะคล้ายคลึงกับ Shaw's min shift แต่จะแตกต่างตรงที่ไม่มีข้อจำกัดในเลือกทำลายคำตอบบางส่วนจากพยาบาลคนเดิม เนื่องจากการที่พยาบาลได้รับกะการทำงานที่มากกว่าคนอื่นการที่จะทำลายกะบางกะออกมีแนวโน้มที่จะส่งผลดีให้กับตารางการทำงานพยาบาลมากกว่าจะเกิดผลเสีย การทำลายด้วย Shaw's max shift จะวนซ้ำไปจนครบจำนวน Neighborhood size และทำการคืนคำตอบที่ถูกทำลายกลับมา แสดงดังรูปที่ 3-18

```
Pseudocode for Shaw Removal with Max Shift
1 Input: Solution, Neighborhood size (n)
2 Procedure: Solution, n begin
3     for i = 1 to n do
4         Determine element with maximum shift in Solution
5         Remove the determined element from Solution
6     end
7     return Modified Solution
8 end
```

รูปที่ 3-18 Pseudocode แสดงขั้นตอนการทำงานของ Shaw's max shift

- การทำลายคำตอบแบบ Shaw's min working day : จะมีหลักการคล้ายกับ การทำลายคำตอบแบบ Shaw's min-max shift แต่จะเป็นการคำนวณจำนวนวันทำงานของพยาบาลที่ได้รับจำนวนกะการทำงานที่น้อยที่สุดภายใต้ข้อจำกัดพยาบาลหนึ่งคนจะถูกทำลายได้เพียงหนึ่งครั้งเท่านั้น การทำลายจะทำลายจนครบจำนวน Neighborhood size และทำการคืนคำตอบที่ถูกทำลายกลับมาแสดงดังรูปที่ 3-19

```
Pseudocode for Shaw Removal with Min Working Day
1  Input: Solution, Neighborhood size (n), Nurse selection counts
2  Output: Modified Solution
3  Initialize: nurseSelectionCount for each nurse in Solution to 0
4  ShawRemovalMinWorkingDay(Solution, n) begin
5      for i = 1 to n do
6          Determine element with minimum working day in Solution
7          if nurseSelectionCount[element] < 1 then
8              Remove the determined element from Solution
9              nurseSelectionCount[element] += 1
10         end
11     end
12     return Modified Solution
13 end
```

รูปที่ 3-19 Pseudocode แสดงขั้นตอนการทำงานของ Shaw's min working day

- การทำลายคำตอบแบบ Shaw's max working day : ในการทำงานของ Shaw's max working day จะทำการหาพยาบาลที่ได้รับวันทำงานในกรอบเวลาจัดตาราง 28 วันสูงที่สุด โดยคำนวณวันทำงานของพยาบาลจากตารางการทำงาน เพื่อทำลายคำตอบโดยไม่มีข้อจำกัดเหมือนดัง Shaw's min working day ทำให้สามารถเลือกทำลายได้อย่างอิสระ การทำลายจะทำลาย

จนครบจำนวน Neighborhood size และทำการคืนคำตอบที่ถูกทำลายกลับมาแสดงดังรูปที่ 3-20

```
Pseudocode for Shaw Removal with Max Working Day
1 Input: Solution, Neighborhood size (n)
2 Output: Modified Solution
3 ShawRemovalMaxWorkingDay(Solution, n) begin
4     for i = 1 to n do
5         Determine element with maximum working day in Solution
6         Remove the determined element from Solution
7     end
8     return Modified Solution
9 end
```

รูปที่ 3-20 Pseudocode แสดงขั้นตอนการทำงานของ Shaw's max working day

- การทำลายคำตอบแบบ Shaw's max violation : ในส่วนของการทำลายคำตอบแบบ Shaw's max violation จะทำการนำเข้าข้อมูลสองอย่าง คือ solution และ Neighborhood size เพื่อใช้ในการตรวจสอบการละเมิดกฎการทำงานที่อยู่ภายในคำตอบปัจจุบัน ซึ่งการทำลายคำตอบนั้นจะเลือกทำลายกะการทำงานของพยาบาลที่จะช่วยลดค่าการละเมิดกฎได้สูงที่สุด (Obj.3) โดยการทำลายคำตอบจะสามารถทำลายกะการทำงานของพยาบาลคนเดิมได้สูงสุดเพียง 3 กะการทำงานเท่านั้น และเลือกทำลายคำตอบจำนวนเท่ากับ Neighborhood size แสดงดังรูปที่ 3-21

```
Pseudocode for Shaw Removal with Max Violation
1 Input: Solution, Neighborhood size (n)
2 Output: Modified Solution
3 ShawRemovalMaxViolation(Solution, n) begin
4     Initialize a dictionary to keep track of how many times each element
       has been selected
5     for i = 1 to n do
6         while True do
7             Determine element with maximum violation in Solution
8             If the element has been selected less than 3 times
9             Increase the count of that element in the dictionary by 1
10            Remove the determined element from Solution
11            Break
12        end
13        Exclude that element from being the maximum in the next
           determination
14    end
15 end
16 return Modified Solution
```

รูปที่ 3-21 Pseudocode แสดงขั้นตอนการทำงานของ Shaw's max violation

3.3.5.4 การเลือกทำลายคำตอบแบบ History-Based Removal

อีกหนึ่งวิธีในการเลือกทำลายคำตอบ History-based removal เป็นการจดจำประสิทธิภาพ โดยคำตอบบางส่วนถูกทำลายจะพิจารณาว่าการทำลายคำตอบนั้นๆ ช่วยให้ปรับปรุงฟังก์ชั่นจุดประสงค์ (Objective function) ในมิติต่างๆของการจัดตารางการทำงานของพยาบาลหรือไม่ หากการทำลายนั้นช่วยทำให้ประสิทธิภาพดีขึ้นจะถูกพิจารณา และนำเข้าไปในการตัดสินใจเลือกใช้ในรอบถัดๆไป ซึ่งวิธีการนี้จะแตกต่างกับวิธีแบบ Random removal และ Shaw removal ในแง่ของการจดจำข้อมูลที่เกิดขึ้นในอดีต และมีขั้นตอนการประมวลผลเพื่อหาคำตอบที่ซับซ้อนกว่า จะมีข้อดีเมื่อคำตอบปัจจุบันอยู่ใกล้กับ Optimal solution ซึ่ง History-based removal จะช่วยให้ขั้นตอนการปรับปรุงคุณภาพของคำตอบเข้าสู่ Optimal solution ได้ไวขึ้นทำให้ใช้เวลาในการปรับปรุงลดน้อยลง

ในส่วนของขั้นตอนการทำงานของ History-based removal นอกเหนือจากการนำเข้าข้อมูล Solutions และ Neighborhood size แล้วยังจำเป็นจะต้องเพิ่มจำนวนรอบสูงสุดในการสำรวจการทำลายแต่ละครั้ง (maxIter) Segment และจำนวนประวัติคำตอบที่ถูกทำลายในอดีต (m) โดยในทุกครั้งที่ทำการทำลายคำตอบจะโปรแกรมจะทำการเก็บข้อมูล และประเมินผลกระทบของการทำลายคำตอบนั้น เมื่อกระบวนการสุ่มสร็จสิ้นโปรแกรมจะเข้าสู่ช่วงที่สอง คือ การเลือกคำตอบที่จะทำการทำลาย โดยตัดสินใจจากคำตอบที่มีโอกาสจะนำไปสู่การปรับปรุงค่าวัตถุประสงค์ที่ดีขึ้นมากที่สุดร่วมกับข้อมูลในอดีต เมื่อสิ้นสุดกระบวนการจะทำการคืนคำตอบที่ถูกทำลายกลับมา แสดงดังรูปที่ 3-23

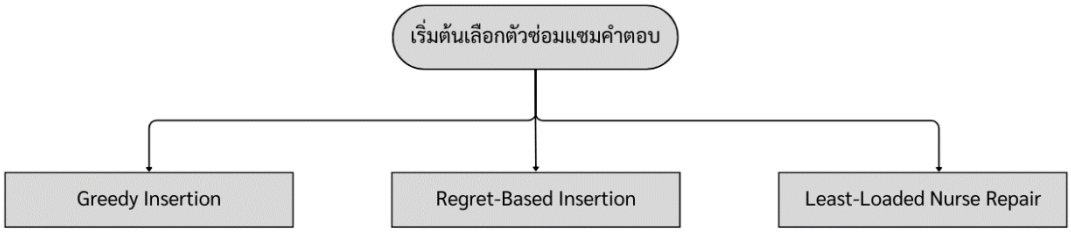
Pseudocode for History Based Removal

```
Input: Solutions, Neighborhood size n, Maximum random iterations
       maxIter, Segment
History Based Removal(Solutions, n, maxIter, segment, m) begin
    all_destroyed ← []
    for seg ← 1 to segment do
        destroy_selected ← []
        removal_history ← []
        iteration ← 0
        while iteration < maxIter do
            RemovedSolutions ← Randomly remove n solutions from
             Solutions
            improvement ← Evaluate the impact of RemovedSolutions
             on the quality of Solutions and select the best solution in
             RemovedSolutions
            Append improvement to removal_history
            iteration ← iteration + 1
        end
        for i ← 1 to n do
            improvement ← Select the best solution in removal_history
            Append improvement to destroy_selected
        end
        Destroy the selected elements from destroy_selected
        Append all elements from destroy_selected to all_destroyed
        if length(all_destroyed) > m then
            all_destroyed ← Select the best m solutions in all_destroyed
        end
    end
    return Modified Solution
end
```

รูปที่ 3-22 Pseudocode แสดงขั้นตอนการทำงานของ History based removal

3.3.5.5 การเลือกตัวซ่อมแซมคำตอบ (Repair Operator Selection)

การซ่อมแซมคำตอบเป็นขั้นตอนหลักใน ALNS ที่มีหน้าที่ซ่อมแซมคำตอบต่อจากขั้นตอนการทำลายคำตอบ โดยใช้วิธีการซ่อมแซมคำตอบที่เหมาะสม เพื่อให้คำตอบที่ได้มีคุณภาพที่ดีขึ้น ซึ่งในงานวิจัยนี้จะใช้วิธีการซ่อมแซมคำตอบ 3 วิธี ได้แก่ Greedy insertion, Regret-based insertion และ Least-loaded nurse repair แสดงดังรูปที่ 3-23 สำหรับการเลือกใช้วิธีการซ่อมแซมคำตอบจะใช้หลักการเดียวกับการเลือกใช้วิธีทำลายคำตอบ คือ Roulette wheel ที่โอกาสที่จะถูกเลือกใช้เท่ากันในรอบการซ่อมแซมแรก และเพิ่มโอกาสวิธีการซ่อมแซมที่มีประสิทธิภาพให้มีโอกาสในการถูกเลือกใช้มากขึ้น นอกจากนี้ในขั้นตอนการซ่อมแซมคำตอบ จะไม่เพียงซ่อมแซมให้ครบจำนวนที่ถูกทำลายไปก่อนหน้า แต่สามารถซ่อมแซมน้อยกว่า หรือมากกว่าจำนวนที่ถูกทำลายไป หากทำให้ประสิทธิภาพของคำตอบถูกปรับปรุงไปในทางที่ดีขึ้นได้



รูปที่ 3-23 วิธีในการซ่อมแซมคำตอบ (Repair operators)

3.3.5.6 การเลือกซ่อมแซมคำตอบแบบ Greedy Insertion

Greedy insertion เป็นวิธีการที่นำหลักการของ Greedy algorithm โดยจะเลือกการตัดสินใจที่ปรับปรุงคุณภาพของคำตอบในปัจจุบันให้ได้มากที่สุดเท่าที่จะเป็นไปได้ แต่จะไม่พิจารณาผลกระทบที่จะเกิดขึ้นในระยะยาว ซึ่ง Greedy insertion จะถูกนำมาใช้เป็นหนึ่งในขั้นตอนการปรับปรุงคำตอบสำหรับปัญหาการจัดตารางการทำงานของพยาบาล มีจุดประมีจุดประสงค์เพื่อปรับปรุงคำตอบที่ได้จากขั้นตอนการทำลายคำตอบ (Destroy) ให้มีคุณภาพดีขึ้น โดยการซ่อมแซมคำตอบที่เหมาะสม โดยขั้นตอนการทำงานของ Greedy insertion จะนำเข้าคำตอบที่ถูกทำลายมาคำนวณหาตำแหน่งที่จะซ่อมแซมคำตอบแล้วค่าวัตถุประสงค์ดีขึ้นให้ได้มากที่สุด จนกระทั่งจำนวนของคำตอบที่ถูกทำลายถูกซ่อมแซมจนครบทุกคำตอบที่ถูกทำลายจะทำการคืนค่าคำตอบที่ถูกซ่อมแซมกลับมา แสดงดังรูปที่ 3-24

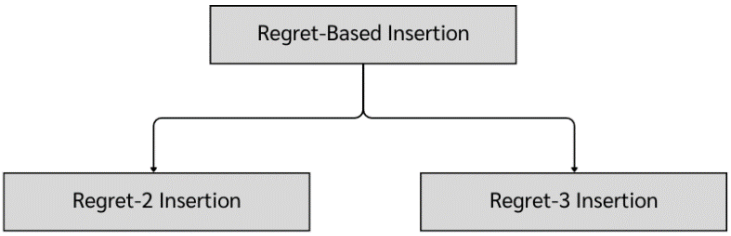
CHULALONGKORN UNIVERSITY

```
Pseudocode for Greedy Insertion
1  Input: Modified Solution (from Destroy Operator), List of Removed
     Elements
2  GreedyInsertion(ModifiedSolution, RemovedElements) begin
3      foreach element in RemovedElements do
4          foreach position in ModifiedSolution do
5              Compute the objectives for inserting element at position
6          end
7          Determine the position with the best improved objectives using
             the calculate objectives values
8          Insert the element into the determined position in
             ModifiedSolution
9      end
10     return Repaired Solution
11 end
```

รูปที่ 3-24 Pseudocode แสดงขั้นตอนการทำงานของ Greedy insertion

3.3.5.7 การเลือกซ่อมแซมคำตอบแบบ Regret-Based Insertion

Regret-based insertion เป็นกระบวนการซ่อมแซมคำตอบที่ใช้การประเมิน และเปรียบเทียบผลกระทบในการตัดสินใจเลือกซ่อมแซมคำตอบแต่ละครั้ง โดยการประเมินจะคำนวณค่า Regret หรือค่าเสียโอกาสหากไม่เกิดการแซ่มแซมคำตอบในตำแหน่งนั้นๆ ซึ่งค่า Regret จะสามารถคำนวณได้จากการเปลี่ยนแปลงของวัตถุประสงค์เมื่อทำการซ่อมแซมคำตอบในตำแหน่งที่แตกต่างกัน ซึ่งในงานวิจัยนี้จะใช้ Regret-based insertion 2 วิธีได้แก่ 2 Regret-based with random insertion และ3 Regret-based with random insertion แสดงดังรูปที่ 3-25



รูปที่ 3-25 ชนิดของตัวซ่อมแซมคำตอบแบบ Regret-based insertion

- 2 Regret-based with random insertion: ในการซ่อมแซมคำตอบด้วย 2 Regret จะเริ่มจากการนำเข้าคำตอบที่ผ่านการทำลายมาก่อนหน้า จากนั้นจะคำนวณหาตำแหน่ง ที่ซ่อมแซมคำตอบแล้วเกิดการปรับปรุงค่าวัตถุประสงค์ได้สูงสุด 2 อันดับ และจะทำการสุ่มซ่อมแซมคำตอบจาก 1 ใน 2 ของตำแหน่งที่เกิดการปรับปรุงค่าวัตถุประสงค์ได้สูงที่สุด ซึ่งการทำงานของ 2 Regret จะดำเนินการจนกระทั่งตารางการทำงานพยาบาลถูกซ่อมแซมจนสมบูรณ์ โดยมีขั้นตอนแสดงดังรูปที่ 3-26

```
Pseudocode for 2-Regret-Based Insertion
1  Input: Modified Solution (from Destroy Operator), List of Removed
     Elements
2  TwoRegretBasedInsertion(ModifiedSolution, RemovedElements)
     begin
3      foreach element in RemovedElements do
4          foreach position in ModifiedSolution do
5              Compute the objectives for inserting element at position
6              Determine the two best objective values for this position
7              Compute the regret value for this position as the difference
                between the two best objectives
8          end
9          Determine the two positions with the highest regret values
10         Randomly select one of the two positions
11         Insert the element into the selected position in ModifiedSolution
12     end
13     return Repaired Solution
14 end
```

รูปที่ 3-26 Pseudocode แสดงขั้นตอนการทำงานของ 2 Regret-based insertion

- 3 Regret-based with random insertion: การนำ 3 Regret มาประยุกต์ใช้ในการซ่อมแซมคำตอบจะมีการนำเข้าข้อมูล และขั้นตอนการคำนวณหาค่า Regret ที่เหมือนกับวิธี 2 Regret เพียงแต่จะคำนวณหาตำแหน่งที่ซ่อมแซมคำตอบแล้วสามารถปรับปรุงค่าวัตถุประสงค์ได้มากสุดจำนวน 3 ลำดับ และใช้เกณฑ์ในการเลือกซ่อมแซมที่แตกต่างกัน ซึ่งการเลือกการซ่อมแซมของวิธี 3 Regret จะมีการเปรียบเทียบคุณภาพของคำตอบที่ดีที่สุดในอันดับ 1 และอันดับ 2 หากแตกต่างกันเกิน 5% จะทำการเลือกซ่อมแซมคำตอบในตำแหน่งที่เกิดการปรับปรุงค่าวัตถุประสงค์สูงที่สุด แต่ถ้าแตกต่างกันน้อยกว่า 5% จะทำการเลือกซ่อมแซมคำตอบในตำแหน่งที่เกิดการปรับปรุงค่าวัตถุประสงค์สูงสุดอันดับ 3 แทน สำหรับขั้นตอนการซ่อมแซมจะดำเนินการจนทุกองค์ประกอบที่ถูกทำลาย ได้รับการซ่อมแซมอย่างสมบูรณ์คำตอบที่ถูกซ่อมแซมแล้วจะถูกคืนค่ากลับมาเพื่อตรวจสอบต่อไป แสดงขั้นตอนดังรูปที่ 3-27

```
Pseudocode for 3-Regret-Based Insertion
1  Input: Modified Solution (from Destroy Operator), List of Removed
     Elements
2  Output: Repaired Solution
3  ThreeRegretBasedInsertion(ModifiedSolution, RemovedElements)
     begin
4      foreach element in RemovedElements do
5          foreach position in ModifiedSolution do
6              Compute the objectives for inserting element at position
7              Determine the three best objective values for this position:
                Best1, Best2, and Best3
8              Compute the regret values between Best1 and Best2, and
                between Best1 and Best3
9          end
10         if The difference between Best1 and Best2 ≤ 5% then
11             Select the position corresponding to Best3
12         end
13         else
14             Select the position corresponding to Best1
15         end
16         Insert the element into the selected position in ModifiedSolution
17     end
18     return Repaired Solution
19 end
```

รูปที่ 3-27 Pseudocode แสดงขั้นตอนการทำงานของ 3 Regret-based insertion

3.3.5.8 การเลือกซ่อมแซมคำตอบแบบ Least-Loaded Nurse Repair

วิธีการปรับปรุงคำตอบแบบ Least-loaded nurse repair (LLNR) มุงเน้นไปที่การปรับปรุงคำตอบโดยพิจารณาภาระงานของพยาบาลในกรอบเวลาการจัดตะรางการทำงานทั้งหมด ซึ่งจะพยายามรักษาสมดุลภาระของพยาบาลทุกคนภายในแผนกให้เกิดความเท่าเทียมกันมากที่สุดเท่าที่จะเป็นไปได้ วิธีการนี้นอกจากจะสามารถกระจายภาระงานให้เท่าเทียมกันได้แล้ว ยังสามารถปรับปรุงค่าวัตถุประสงค์อื่นๆได้อีกด้วย หลักการทำงานของ LLNR จะเริ่มต้นจากการวิเคราะห์ภาระงานรวมของพยาบาลแต่ละคน แล้วจะทำการเลือกพยาบาลที่มีภาระงานจำนวนน้อยเข้ามาเติมเต็มตารางที่ถูกนำออกไปในขั้นตอนการทำลายคำตอบบางส่วน ซึ่งจะช่วยให้พยาบาลที่ได้รับภาระงานที่น้อยสามารถมาทดแทนในบางกะที่พยาบาลบางคนไม่สามารถเข้าทำงานได้ และสามารถปรับปรุงค่าวัตถุประสงค์ (Objective function) ในเวลาเดียวกันได้อีกด้วย โดยมีขั้นตอนทำงานโดยละเอียดแสดงดังรูป 3-28

Pseudocode for Least-Loaded Nurse Repair

```
1  Input: Modified Solution (from Destroy Operator), List of Removed
     Elements
2  Output: Repaired Solution
3  while List of Removed Elements is not empty do
4      Nurse N = Find the nurse with the least number of tasks in
         Modified Solution
5      Element E = Select an element from the List of Removed Elements
6      if Element E can be assigned to Nurse N then
7          Assign Element E to Nurse N in Modified Solution
8          Remove Element E from List of Removed Elements
9      end
10     else
11         Exclude Nurse N from further selection in this iteration
12     end
13     return Repaired Solution
14 end
```

รูปที่ 3-28 Pseudocode แสดงขั้นตอนการทำงานของ Least-loaded nurse repair

3.3.6 การปรับปรุงน้ำหนักของตัวดำเนินการ (Operator Weights Updating)

การปรับปรุงค่าน้ำหนักของตัวดำเนินการเป็นขั้นตอนที่จะสร้างโอกาสให้ ALNS เลือกขั้นตอนการทำลายคำตอบ และปรับปรุงคำตอบ ได้เหมาะสมโดยอาศัยการให้คะแนน และคำนวณคะแนนจากในอดีตจนถึงปัจจุบันจนกระทั่งครบจำนวน Segment คะแนนทั้งหมดจะถูก Reset ให้กลับไปเป็นค่าตั้งต้นเพื่อให้เกิดความหลากหลายและป้องกันการติดอยู่ใน Local minimum ในที่นี้ได้นำวิธีการที่มีชื่อว่า Performance Feedback ที่เป็นวิธีการปรับค่าน้ำหนักของวิธีการทำลาย และวิธีการปรับปรุงที่ถูกเลือกใช้ในรอบนั้นๆ เป็นการใช้การปรับค่าน้ำหนักที่ใช้ประสิทธิภาพของรอบการค้นหาที่ผ่านมาเพื่อปรับปรุงความน่าจะเป็นในการเลือกใช้ตัวทำลายคำตอบ และตัวปรับปรุงคำตอบในการค้นหารอบถัดไป ซึ่งวิธี Performance Feedback สามารถคำนวณได้จากสมการที่ (15)

$$W_{i(t+1)} = W_{it} + \alpha\left(P_{it} - P_t^{avg}\right) \qquad (15)$$

$W_{i(t+1)} =$ น้ำหนักรวมของตัวดำเนินการ (Operators) ที่ i ในรอบถัดไป

$W_{it} =$ น้ำหนักรวมของตัวดำเนินการ (Operators) ที่ i ในรอบปัจจุบัน

$P_{it} =$ คะแนนของตัวดำเนินการ (Operators) ที่ i ในรอบปัจจุบัน

P_t^{avg} = คะแนนเฉลี่ยของตัวดำเนินการ (Operators) ในรอบปัจจุบัน

หลังจากคำนวณหาค่าน้ำหนักของแต่ละตัวดำเนินการ (Operators) จะถูกนำมาใช้คำนวณค่าความน่าจะเป็นเพื่อเป็นตัวช่วยให้ ALNS มีโอกาสเลือกตัวดำเนินการ (Operators) ที่มีแนวโน้มที่จะปรับปรุงคุณภาพคำตอบได้มาก และลดโอกาสเลือกตัวเลือกที่ไม่มีประสิทธิภาพในแต่ละ Segment ซึ่งค่าความน่าจะเป็นจะสามารถคำนวณได้จากสมการที่ (16)

$$P_i = \left(\sum_{j \in J} W_j\right)^{-1} * W_i \qquad (16)$$

P_i = ความน่าจะเป็นในการเลือกตัวดำเนินการที่ i

W_i = น้ำหนักปัจจุบันของตัวดำเนินการที่ i

W_j = น้ำหนักปัจจุบันของตัวดำเนินการที่ j โดยที่ J เป็นเซตของตัวดำเนินการ (Operators) ทั้งหมด

3.3.7 การปรับปรุง - ตรวจสอบคุณภาพคำตอบด้วย Simulated Annealing (SA)

ในขั้นตอนนี้จะเป็นขั้นตอนตรวจสอบคุณภาพคำตอบ และทำการปรับปรุงคำตอบเป็นครั้งสุดท้ายก่อนที่โปรแกรมจะทำการวนคำตอบไปปรับปรุงคุณภาพใหม่ หรือเข้าเกณฑ์ในการหยุดทำงานในขั้นตอนนี้คำตอบจะถูกตรวจสอบด้วย Simulated annealing ก่อนเพื่อเช็คว่าเข้าเกณฑ์ในการยอมรับคำตอบหรือไม่ หากไม่เข้าเกณฑ์คำตอบจะถูกส่งไปปรับปรุงด้วยวิธี Local search หรือ Relocation จากนั้นจะกลับเข้ามาตรวจสอบคุณภาพอีกครั้งด้วย Simulated annealing โดยขั้นตอนทั้งหมดจะเป็นดัง Pseudocode ในรูปที่ 3-29

Pseudocode for Simulated Annealing

Input: Solution (Not optimal solution in Pareto front), $P_{initial}$, Cooling rate C

Function `SimulatedAnnealing`$(Solution, T_{initial}, C)$**:**
  $T \leftarrow -\frac{\Delta Sol}{\ln(0.5)}$
  $Solution_{current} \leftarrow Solution$
  if *Pass through SA with* $Solution_{current}$ **then**
    return Best Solution found
  for $i \leftarrow 1$ **to** n **do**
    $Solution_{temp} \leftarrow Solution_{current}$
    if *Random choice between Localsearch and Relocation* **then**
      Randomly remove one nurse from a shift in $Solution_{temp}$
      Randomly assign a nurse (possibly the same or different) to the shift
    else
      Swap two nurses' shifts in a different type of shift that can be substituted (e.g., Day to Night)
    if *Improvement check* **then**
      $Solution_{current} \leftarrow Solution_{temp}$
  if *Pass through SA with* $Solution_{current}$ **then**
    return Best Solution found
  return Best Solution found

รูปที่ 3-29 การปรับปรุง - ตรวจสอบคุณภาพคำตอบด้วย Simulated Annealing (SA)

3.3.7.1 ขั้นตอนการยอมรับคำตอบด้วยวิธี Simulated Annealing (SA)

หลักการทำงานของ Simulated annealing มีที่มาจากกระบวนการหลอมโลหะที่มีการให้ความร้อน และปล่อยให้โลหะมีการเย็นตัวด้วยอัตราการเย็นตัวที่เหมาะสมทำให้โครงสร้างภายในเนื้อโลหะมีคุณสมบัติความแข็งแรงและทนทานต่อการใช้งานมากขึ้น เมื่อนำมาประยุกค์ใช้กับการยอมรับคำตอบที่บางครั้งอาจมีการยอมรับคำตอบที่แย่กว่าเดิมในบางครั้ง ตามค่าอุณหภูมิที่ลดลงเป็นระยะๆ ซึ่งในช่วงแรกของการยอมรับคำตอบจะเป็นช่วงที่มีอุณหภูมิที่สูง SA จะยอมรับคำตอบในช่วงแรกนี้ได้ง่าย แต่เมื่ออุณหภูมิเริ่มเย็นตัว (ลดต่ำลง) โอกาสในการยอมรับคำตอบที่ไม่ค่อยมีประสิทธิภาพก็จะถูกยอมรับได้ยากขึ้นด้วย โดยสามารถคำนวณได้จากสมการที่ (17) สำหรับงานวิจัยนี้ในการหาคำตอบรอบแรกจะกำหนด $T_{initial}$ จากการคำนวณค่าความน่าจะเป็นที่จะยอมรับคำตอบ ($P_{accepted}$) ที่ 50 % และใช้อัตราการเย็นตัว (Cooling rate) ที่ 97 %

$$\mathrm{P_{accepted}} = \mathrm{e}^{-\frac{\Delta \mathrm{Sol}}{\mathrm{T_{Current}}}} \qquad (17)$$

$$\mathrm{T_{initial}} = \frac{-\Delta \mathrm{Sol}}{ln(P_{accepted})} \qquad (18)$$

$$\Delta \mathrm{Sol} = Sol_{new} - Sol_{old} \qquad (19)$$

$$\text{Geometric Cooling}: \ \mathrm{T_f} = \mathrm{T_{Current}} * \mathrm{c} \qquad (20)$$

3.3.7.2 การปรับปรุงคำตอบในพื้นที่ใกล้เคียง (Local Search)

ในการปรับปรุงคำตอบแบบ Local search จะทำการค้นหาคำตอบในบริเวณใกล้เคียงของคำตอบในปัจจุบันเพื่อหาคำตอบที่ดีขึ้น โดยขั้นตอนแรก Local search จะสุ่มเลือกพยาบาล 1 คน ออกจากกะการทำงาน 1 กะ และสุ่มเลือกพยาบาล 1 คน เข้ามาแทนที่ หากคำตอบใหม่ที่ได้จาก Local search นั้นเป็นคำตอบที่ดีขึ้นเราจะทำการยอมรับคำตอบใหม่ แต่ถ้าคำตอบใหม่เป็นคำตอบที่แย่ลงจะทำการเก็บคำตอบเดิมไว้ และทำการปรับปรุงต่อจนถึงรอบที่ n คำตอบจะถูกนำไปเข้ากระบวนการยอมรับคำตอบด้วย SA เป็นครั้งสุดท้ายก่อนสิ้นสุดขั้นตอนการยอมรับคำตอบ โดยแสดงขั้นตอนดังรูปที่ 3-29

3.3.7.3 การปรับปรุงคำตอบโดยการสลับตำแหน่ง (Relocation)

Relocation เป็นการปรับปรุงคำตอบโดยไม่เป็นการทำลายคำตอบเดิม แต่จะใช้วิธีการสลับสับเปลี่ยนกะการทำงาน หรือวันการทำงานที่ถูกจัดอยู่ในตารางการทำงานของพยาบาลเพื่อเพิ่มโอกาสในการปรับปรุงประสิทธิภาพจากตารางการทำงานเดิมให้สามารถลดความเหลื่อมล้ำของค่าจ้าง หรือลดการละเมิดกฎการทำงานได้ เพียงแค่การสลับตำแหน่งกะการทำงานของพยาบาลสองคน ในงานวิจัยนี้จะทำการสลับโดยการเปลี่ยนชนิดกะที่สามารถทดแทนกันได้ เช่น กะดั้งเดิมเป็นกะ Day ในการ Relocation จะสลับเป็นกะเช้าแทน การทำงานของ Relocation จะสิ้นสุดเมื่อครบจำนวนรอบที่ได้กำหนดไว้ และจะนำคำตอบเข้ากระบวนการตรวจสอบคำตอบด้วย SA เช่นเดียวกับขั้นตอน Local search โดยสามารถดูขั้นตอนการทำงานของ Relocation ได้จากรูปที่ 3-29

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

บทที่ 4
ผลการดำเนินงานวิจัย

4.1 ข้อมูลนำเข้าสำหรับการหาคำตอบ

ในการหาคำตอบของงานวิจัยนี้ผู้วิจัยมีจุดมุ่งหมายที่จะสร้างตารางการทำงานของพยาบาลให้กับทุกคนภายในแผนก ซึ่งจะประกอบด้วย

1 หัวหน้าพยาบาล จำนวน 1 คน
2 พยาบาลวิชาชีพระดับปฏิบัติการ จำนวน 16 คน
3 ผู้ช่วยพยาบาล จำนวน 2 คน
4 พนักงานฉุกเฉินการแพทย์ จำนวน 6 คน

การจัดตารางการทำงานพยาบาลให้กับพยาบาลทั้งแผนก (25 คน) นั้นผู้วิจัยจะทำการแก้ปัญหาด้วยวิธีแบบจำลองทางคณิตศาสตร์ และฮิวริสติก ซึ่งจะมีข้อแตกต่างระหว่างสองวิธี คือ การแก้ปัญหาด้วยวิธีแบบจำลองทางคณิตศาสตร์จะทำให้เราทราบถึงค่าวัตถุประสงค์ที่ดีที่สุดที่ตารางการทำงานพยาบาลสามารถทำได้ แต่ในการหาคำตอบด้วยวิธีแบบจำลองทางคณิตศาสตร์มีข้อจำกัดด้านเวลาในการประมวลผลเป็นอย่างมาก เนื่องจากปัญหาการจัดตารางการทำงานพยาบาล ในงานวิจัยนี้มีจำนวนพยาบาลมากเกินกว่าความสามารถของการประมวลผลในกรอบเวลาที่ได้ตั้งไว้ จึงมีความจำเป็นที่จะต้องลดขนาดของปัญหา เพื่อให้สามารถหาคำตอบที่ดีที่สุดได้ภายในระยะเวลาที่กำหนด สำหรับวิธีฮิวริสติก จะถูกใช้เพื่อลดเวลาในการประมวลผล และสร้างตารางการทำงานพยาบาลให้มีตัวเลือกในการจัดสินใจนำไปใช้ได้อย่างหลากหลากหลาย โดยจะถูกแบ่งให้จัดตารางการทำงานสำหรับปัญหาขนาดเล็กเพื่อเทียบประสิทธิภาพของอัลกอริทึม ถ้าหากสามารถให้คำตอบที่ใกล้เคียงกับวิธีแบบจำลองทางคณิตศาสตร์จะทำการใช้ในการแก้ปัญหาขนาดใหญ่ หรือว่าใช้ในการจัดตารางการทำงานพยาบาลให้กับทุกคนภายในแผนก โดยจะสามารถสรุปรายละเอียด วิธีการ ขนาดของปัญหา และวัตถุประสงค์ได้ดังตารางที่ 4-1

จุฬาลงกรณ์มหาวิทยาลัย
CHULALONGKORN UNIVERSITY

ตารางที่ 4-1 รายละเอียดการดำเนินการหาคำตอบสำหรับแต่ละวิธี

วิธีที่ใช้ในการหาคำตอบ	ขนาดของปัญหา	จำนวนพยาบาล (หน่วย : คน)	กรอบเวลาการสร้างตาราง (หน่วย : วัน)	วัตถุประสงค์
แบบจำลองทางคณิตศาสตร์	ปัญหาขนาดเล็ก	RN = 8 คน	28 วัน	หาคำตอบที่ดีที่สุด
ฮิวริสติก (Heuristic)	ปัญหาขนาดเล็ก	RN = 8 คน	28 วัน	ทดสอบประสิทธิภาพ
ฮิวริสติก (Heuristic)	ปัญหาขนาดใหญ่	HD = 1 คน RN = 16 คน PN = 2 คน EMT = 6 คน	28 วัน	สร้างตารางการทำงานจริง

สำหรับแบบจำลองทางคณิตศาสตร์เราจะแก้ปัญหา โดยมีโปรแกรม IBM ILOG CPLEX Optimization Studio เป็นการเขียนสมการทางคณิตศาสตร์ให้อยู่ในรูปแบบภาษาของโปรแกรม แต่ในการหาคำตอบด้วยฮิวริสติกนั้นนอกจากการสร้างอัลกอรึทึมที่เหมาะสมกับการแก้ปัญหาตารางการทำงานพยาบาลแล้วเราจะต้องกำหนดพารามิเตอร์ที่เหมาะสมกับการทำงานของโปรแกรมที่ไม่ว่าจะเป็นการแก้ปัญหาขนาดเล็ก หรือการแก้ปัญหาขนาดใหญ่ในการจัดตารางการทำงานพยาบาลจะถูกใช้พารามิเตอร์เหล่านี้ที่เหมือนกัน

เนื่องจากงานวิจัยนี้เลือกใช้ Adaptive Large Neighborhood Search (ALNS) ในการหาคำตอบจึงมีความจำเป็นต้องกำหนดเกณฑ์ในการให้คะแนน σ_1 ,σ_2, σ_3, σ_4 และ σ_5 เท่ากับ 0,10,7,4 และ 2 ตามลำดับ เพื่อใช้เป็นตัวสะท้อนประสิทธิภาพของตัวดำเนินการ (Operator) และถูกกำหนดให้รอบการค้นหาสูงสุดอยู่ที่ 6,600 รอบ โดยแบ่งเป็น Segment ทั้งหมด 66 Segment ทำให้แต่ละ Segment จะมีรอบในการดำเนินการ 100 รอบ ก่อนจะทำการ Reset ค่าคะแนนสะสม

นอกจากจำนวนรอบการค้นหาสูงสุดที่จะทำให้โปรแกรมหยุดทำงานเมื่อครบจำนวนรอบแล้ว การกำหนดเกณฑ์การหยุดทำงานด้วยจำนวนรอบการค้นหาสูงสุดที่ไม่ถูกปรับปรุงในการค้นหา จะส่งผลให้ ALNS หลีกเลี่ยงการค้นหาที่ไม่นำไปสู่การพัฒนา หรือปรับปรุงคุณภาพของคำตอบให้ดีขึ้น และในส่วนของกรอบเวลาสูงสุดที่ ALNS จะดำเนินการ จะเป็นกรอบเวลาที่การันตีได้ว่าตารางการทำงานพยาบาลจะสามารถจัดให้แล้วเสร็จได้ภายใน 1 ชั่วโมง ซึ่งหากเกิดเหตุด่วน หรือเหตุจำเป็นในแผนกแล้วมีความจำเป็นจะต้องจัดตารางการทำงานใหม่กรอบเวลา 1 ชั่วโมง สำหรับแผนกฉุกเฉินได้รับการยอมรับจากพยาบาลผู้ปฏิบัติงานว่า สามารถทำให้การจัดการกำลังคนภายในแผนกให้เกิดความต่อเนื่องในการทำงานโดย พารามิเตอร์ทั้งหมดจะแสดงดังตารางที่ 4-2

ตารางที่ 4-2 พารามิเตอร์สำหรับการทำงานของ Adaptive Large Neighborhood Search

พารามิเตอร์ (Parameter)	คำอธิบาย (Description)	ตัวแปรค่าคงที่ (Constant variable)
σ_1	คะแนนเริ่มต้นของทุกการทำลายและซ่อมแซม	0
σ_2	คะแนนเมื่อตัวเลือกการทำลายและซ่อมแซมในรอบเป็นคำตอบที่เป็นคำตอบที่ดีที่สุดเท่าที่เคยมีมาก่อน	10
σ_3	คะแนนเมื่อตัวเลือกการทำลายและซ่อมแซมในรอบเป็นคำตอบที่ดีกว่าคำตอบในปัจจุบันและไม่เคยพบคำตอบมาก่อน	7
σ_4	คะแนนเมื่อตัวเลือกการทำลายและซ่อมแซมในรอบเป็นคำตอบที่ดีกว่าคำตอบในปัจจุบันแต่เคยพบคำตอบมาก่อน	4
σ_5	คะแนนเมื่อตัวเลือกการทำลายและซ่อมแซมในรอบเป็นคำตอบที่แย่กว่าคำตอบในปัจจุบัน	2

ตารางที่ 4-2 พารามิเตอร์สำหรับการทำงานของ Adaptive Large Neighborhood Search (ต่อ)

พารามิเตอร์ (Parameter)	คำอธิบาย (Description)	ตัวแปรค่าคงที่ (Constant variable)
α	อัตราการปรับน้ำหนักของ Performance Feedback	0.5
Segment	จำนวนรอบในการค้นหาแต่ละช่วง	100
Number of Segment	จำนวน Segment ทั้งหมดในการทำงานของ ALNS	66
Number of Iteration	จำนวนรอบการค้นหาสูงสุดที่ ALNS จะดำเนินการ	6,600
Max Unimproved Iteration	จำนวนรอบการค้นหาสูงสุดที่คำตอบไม่ถูกปรับปรุงก่อนที่ ALNS จะหยุดดำเนินการ	1,320
Run Time Limit (Sec)	กรอบเวลาสูงสุดที่ ALNS จะดำเนินการ	3,600

การปรับแต่งค่าพารามิเตอร์ในการทำงานของอัลกอริทึมให้มีประสิทธิภาพไม่ได้มีวิธีที่แน่นอน หรือตายตัว เนื่องจากแต่ละปัญหามีลักษณะเฉพาะที่แตกต่างกัน การตั้งค่าพารามิเตอร์จึงต้องอาศัยการทดลอง และสังเกตพฤติกรรมที่เปลี่ยนแปลงเมื่อมีการปรับค่าพารามิเตอร์ ในกระบวนการปรับค่าให้เหมาะสมควรตรวจสอบให้เกิดความมั่นใจว่าค่าที่ถูกเลือกมาใช้สามารถทำให้อัลกิริทึมทำงานได้อย่างมีประสิทธิภาพตามความต้องการของการหาคำตอบที่เหมาะสม (Joy et al., 2023)

4.2 การนำเข้าข้อมูลที่ใช้ในการหาคำตอบของปัญหาขนาดเล็ก

ผู้วิจัยทำการทดลองหาขนาดปัญหาที่เหมาะสม ด้วยการลดจำนวนพยาบาลในตำแหน่งต่างๆ เริ่มจากนำหัวหน้าพยาบาล ผู้ช่วยพยาบาล พนักงานฉุกเฉินการแพทย์ ออกตามลำดับ เนื่องจากการจัดตารางการทำงานของแผนกฉุกเฉินปัจจุบันยังไม่มีการให้พยาบาลตำแหน่งสูงกว่าเข้ามาทำงานทดแทนตำแหน่งที่ต่ำกว่าได้ ด้วยเหตุนี้การนำพยาบาลบางตำแหน่งออกเพื่อลดขนาดของปัญหาจึงไม่เป็นผลกระทบต่อคำตอบ จากนั้นผู้วิจัยจึงทำการลดจำนวนพยาบาลวิชาชีพระดับปฏิบัติการ จาก 15 คน เป็น 8 คน แต่ยังคงใช้ข้อมูลความต้องการพยาบาลวิชาชีพระดับปฏิบัติการเท่าเดิม (กะเช้า 4 คน, กะบ่าย 3 คน, กะดึก 2 คน เพื่อหาคำตอบที่ดีที่สุดด้วยวิธี Preemptive goal programming ในระยะเวลาหาคำตอบ 3.5 ชั่วโมง ด้วยคอมพิวเตอร์ที่มี CPU Intel Core i7-8750H และแรม 8 GB ด้วยโปรแกรม IBM ILOG CPLEX Optimization Studio

4.3 การหาคำตอบแบบหลายวัตถุประสงค์ด้วย Preemptive Goal Programming กับปัญหาขนาดเล็ก

Preemptive goal programming (PGP) หรือ Multi-objective goal programming คือ วิธีการหาคำตอบของปัญหาแบบหลายวัตถุประสงค์ โดยทำการจัดเรียงความสำคัญของวัตถุประสงค์ที่สนใจ แล้วทำการแก้ปัญหาทีละขั้นตอน เพื่อให้ชุดคำตอบที่ได้เกิดการปรับปรุงตามตามลำดับความสำคัญ เช่น ปัญหาการจัดตารางการทำงานของพยาบาลในงานวิจัยนี้จะมีขั้นตอนการหาคำตอบแสดงดังรูปที่ 4-1

รูปที่ 4-1 ขั้นตอนการหาคำตอบแบบหลายวัตถุประสงค์ด้วยวิธี Preemptive goal programming

การหาคำตอบของงานวิจัยนี้จะทำทั้งหมด 18 รอบ เพื่อให้ได้กลุ่มผลเฉลยของการจัดตารางการทำงานพยาบาลจำนวน 6 ผลเฉลย โดยจะเรียงลำดับความสำคัญในมิติต่างๆ ดังตารางที่ 4-3

จุฬาลงกรณ์มหาวิทยาลัย
CHULALONGKORN UNIVERSITY

ตารางที่ 4-3 การเรียงลำดับความสำคัญของผลเฉลย

ผลเฉลยที่	ลำดับความสำคัญที่ 1	ลำดับความสำคัญที่ 2	ลำดับความสำคัญที่ 3
1	Minimize cost	Minimize unfairness wages	Minimize the violation
2	Minimize cost	Minimize the violation	Minimize unfairness wages
3	Minimize unfairness wages	Minimize cost	Minimize the violation
4	Minimize unfairness wages	Minimize the violation	Minimize cost
5	Minimize the violation	Minimize unfairness wages	Minimize cost
6	Minimize the violation	Minimize cost	Minimize unfairness wages

โดยปกติแล้วการแก้ปัญหาแบบหลายวัตถุประสงค์ด้วยวิธีแม่นตรงจะใช้วิธีที่เรียกว่า Scalarization method ซึ่งวิธีนี้มีข้อดี คือ ใช้เวลาประมวลผลรวดเร็ว เนื่องจากทำการแก้ปัญหาแบบหลายวัตถุประสงค์ในการประมวลผลครั้งเดียวจะมีข้อเสียตรงที่ไม่สามารถทราบถึงค่าที่ดีที่สุดในแต่ละวัตถุประสงค์ได้ สำหรับในงานวิจัยนี้ผู้วิจัยได้แก้ปัญหาของ Scalarization method ด้วยวิธี PGP กับปัญหาขนาดเล็ก (พยาบาลวิชาชีพระดับปฏิบัติการจำนวน 8 คน รองรับความต้องการพยาบาลในกะเช้า กะบ่าย และกะดึก เท่ากับ 4คน 3คน และ2คน ตามลำดับ ในกรอบการสร้างตารางการทำงานล่วงหน้า 28 วัน) ดังที่ได้กล่าวไปก่อนหน้า ผลการหาคำตอบแม่นตรงจะมีจำนวน 6 ผลเฉลย และแต่ละผลเฉลยจะใช้เวลานานกว่าปกติ 2-3 เท่าโดยประมาณ ซึ่งข้อมูลจากการประมวลผลด้วยวิธี PGP สามารถดูได้จากตารางที่ 4-4 ถึง ตารางที่ 4-9

ตารางที่ 4-4 ผลเฉลยที่ 1 จากการหาคำตอบด้วยวิธี PGP กับปัญหาขนาดเล็ก

ผลเฉลยที่ 1					
รอบการประมวลผล	**Minimize cost**	**Minimize unfairness wages**	**Minimize the violation**	**เวลาในการประมวลผล**	**เวลาในการประมวลผลทั้งหมด**
รอบที่ 1	41,558	-	-	00:32:42:88	02:54:00:84
รอบที่ 2	41,558	29,703	-	01:31:03:17	
รอบที่ 3	41,558	29,703	156,000	00:40:13:79	

ตารางที่ 4-5 ผลเฉลยที่ 2 จากการหาคำตอบด้วยวิธี PGP กับปัญหาขนาดเล็ก

ผลเฉลยที่ 2					
รอบการประมวลผล	**Minimize cost**	**Minimize unfairness wages**	**Minimize the violation**	**เวลาในการประมวลผล**	**เวลาในการประมวลผลทั้งหมด**
รอบที่ 1	41,558	-	-	00:32:42:88	02:47:00:67
รอบที่ 2	41,558	-	110,000	01:09:16:19	
รอบที่ 3	41,558	31,383	110,000	01:05:01:60	

ตารางที่ 4-6 ผลเฉลยที่ 3 จากการหาคำตอบด้วยวิธี PGP กับปัญหาขนาดเล็ก

ผลเฉลยที่ 3					
รอบการประมวลผล	**Minimize cost**	**Minimize unfairness wages**	**Minimize the violation**	**เวลาในการประมวลผล**	**เวลาในการประมวลผลทั้งหมด**
รอบที่ 1	-	3,038	-	01:54:30:30	02:56:22:41
รอบที่ 2	45,198	3,038	-	00:27:00:89	
รอบที่ 3	45,198	3,038	241,200	00:34:51:22	

ตารางที่ 4-7 ผลเฉลยที่ 4 จากการหาคำตอบด้วยวิธี PGP กับปัญหาขนาดเล็ก

ผลเฉลยที่ 4					
รอบการประมวลผล	**Minimize cost**	**Minimize unfairness wages**	**Minimize the violation**	**เวลาในการประมวลผล**	**เวลาในการประมวลผลทั้งหมด**
รอบที่ 1	-	3,038	-	01:54:30:30	03:07:42:71
รอบที่ 2	-	3,038	241,200	00:45:55:07	
รอบที่ 3	45,198	3,038	241,200	00:27:17:34	

ตารางที่ 4-8 ผลเฉลยที่ 5 จากการหาคำตอบด้วยวิธี PGP กับปัญหาขนาดเล็ก

ผลเฉลยที่ 5					
รอบการประมวลผล	**Minimize cost**	**Minimize unfairness wages**	**Minimize the violation**	**เวลาในการประมวลผล**	**เวลาในการประมวลผลทั้งหมด**
รอบที่ 1	-	-	76,800	01:15:41:44	03:08:30:46
รอบที่ 2	-	20,743	76,800	01:33:59:10	
รอบที่ 3	44,078	20,743	76,800	00:18:49:92	

ตารางที่ 4-9 ผลเฉลยที่ 6 จากการหาคำตอบด้วยวิธี PGP กับปัญหาขนาดเล็ก

ผลเฉลยที่ 6					
รอบการประมวลผล	**Minimize cost**	**Minimize unfairness wages**	**Minimize the violation**	**เวลาในการประมวลผล**	**เวลาในการประมวลผลทั้งหมด**
รอบที่ 1	-	-	76,800	01:15:41:44	02:46:02:74
รอบที่ 2	43,238	-	76,800	00:29:25:19	
รอบที่ 3	43,238	11,627	76,800	01:00:56:11	

CHULALONGKORN UNIVERSITY

4.4 การหาคำตอบแบบหลายวัตถุประสงค์ด้วยฮิวริสติกกับปัญหาขนาดเล็ก

การหาคำตอบสำหรับปัญหาขนาดเล็กด้วยวิธีฮิวริสติกเราจะทราบถึงประสิทธิภาพของฮิวริสติกที่ได้เลือกใช้ เนื่องจากคำตอบที่ได้จากวิธีฮิวริสติกจะถูกนำมาเปรียบเทียบกับการหาคำตอบด้วยวิธีแม่นตรง ที่มีการใช้ข้อมูลในการจัดตารางการทำงานของพยาบาลที่เป็นชุดเดียวกับวิธีแม่นตรง และจะทำการนำ 3 ผลเฉลยที่ให้ค่าวัตถุประสงค์ที่ต่ำที่สุดในแต่ละวัตถุประสงค์ คือ ผลเฉลยที่ 1 มีค่าใช้จ่ายรวมที่ต่ำที่สุด ผลเฉลยที่ 2 มีความเท่าเทียมในค่าจ้างพิเศษมากที่สุด และคำตอบที่ 3 มีการละเมิดกฎในการทำงานน้อยที่สุด แสดงดังตารางที่ 4-10

ตารางที่ 4-10 ผลเฉลยจากวิธีฮิวริสติกสำหรับปัญหาขนาดเล็ก

ผลเฉลยจากฮิวริสติก				
ผลเฉลย	**Minimize cost**	**Minimize unfairness wages**	**Minimize the violation**	**เวลาในการประมวลผลทั้งหมด**
ผลเฉลยที่ 1	41,698	26,980	107,200	00:21:35:11
ผลเฉลยที่ 2	43,202	3,095	198,000	
ผลเฉลยที่ 3	43,100	11,162	77,200	

จากนั้นจะทำการเปรียบเทียบผลเฉลยจากวิธีฮิวริสติก กับผลเฉลยจากวิธีแม่นตรง โดยเริ่มจากผลเฉลยที่ 1 ของวิธีฮิวริสติก จะนำไปเปรียบเทียบกับผลเฉลยที่ 1 และ 2 จากวิธีแม่นตรง จะสังเกตได้ว่าค่าวัตถุประสงค์ที่ 1 วิธีการแม่นตรงจะให้คำตอบที่ดีกว่า 0.336% แต่ในวัตถุประสงค์ที่ 2 และ 3 คำตอบจากวิธีฮิวริสติก สามารถให้คำตอบที่มีประสิทธิภาพมากกว่าถึง 10.093% สำหรับผลเฉลยที่ 1 , 16.319% สำหรับผลเฉลยที่ 2 และ 45.552% สำหรับผลเฉลยที่ 1 , 2.612% สำหรับผลเฉลยที่ 2 ตามลำดับ สำหรับประสิทธิภาพของผลเฉลยที่ 2 และ 3 สามารถดูได้จากตารางที่ 4-11

ตารางที่ 4-11 การเปรียบเทียบคุณภาพคำตอบระหว่าง IBM CPLEX Optimization Studio และฮิวริสติกของปัญหาขนาดเล็ก

ประสิทธิภาพคำตอบจากฮิวริสติกเทียบกับคำตอบจาก IBM CPLEX Optimization Studio										
ผลเฉลยจาก Heuristics	**Obj.1**	**Obj.2**	**Obj.3**	**ผลเฉลยจาก IBM CPLEX Optimization Studio**	**Obj.1**	**Obj.2**	**Obj.3**	**Obj 1 Gap (%)**	**Obj 2 Gap (%)**	**Obj 3 Gap (%)**
ผลเฉลยที่ 1	41,698	26,980	107,200	ผลเฉลยที่ 1	41,558	29,703	156,000	0.336%	-10.093%	-45.522%
				ผลเฉลยที่ 2	41,558	31,383	110,000	0.336%	-16.319%	-2.612%
ผลเฉลยที่ 2	43,202	3,095	198,000	ผลเฉลยที่ 3	45,198	3,038	241,200	-4.620%	1.842%	-21.818%
				ผลเฉลยที่ 4	45,198	3,038	241,000	-4.620%	1.842%	-21.717%
ผลเฉลยที่ 3	43,100	11,162	77,200	ผลเฉลยที่ 5	44,078	20,743	76,800	-2.269%	-85.836%	0.518%
				ผลเฉลยที่ 6	43,238	11,627	76,800	-0.320%	-4.166%	0.518%

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

จากการทดสอบประสิทธิภาพของคำตอบกับปัญหาขนาดเล็ก เราจะพบว่าวิธีฮิวริสติกไม่สามารถสร้างผลลัพธ์ของวัตถุประสงค์ในแต่มิติได้เหนือกว่าวิธีแม่นตรง แต่จะสามารถให้ค่าที่มีความใกล้เคียงในวัตถุประสงค์หลัก และให้คำตอบที่ดีกว่ามากในวัตถุประสงค์ที่เรากำหนดให้มีความสำคัญรองลงมา นอกจากนี้วิธีฮิวริสติกยังสามารถลดเวลาที่ใช้ในการประมวลผลได้ถึง 87.911% เมื่อเทียบกับเวลาเฉลี่ยที่ใช้ในการประมวลผลของวิธีแม่นตรง ซึ่งเป็นการแสดงถึงประสิทธิภาพของฮิวริสติกทั้งในด้านคุณภาพของคำตอบ และเวลาในการประมวลผล

4.5 การหาคำตอบแบบหลายวัตถุประสงค์ด้วยฮิวริสติกกับปัญหาขนาดใหญ่

จากที่ได้ทราบถึงประสิทธิภาพของฮิวริสติกจากการเปรียบเทียบกับวิธีแม่นตรงแล้วทำให้เรามั่นใจได้ว่า ฮิวริสติก ที่ได้เลือกใช้สามารถแก้ปัญหาการจัดตารางการทำงานของพยาบาลได้อย่างมีประสิทธิภาพ ในขั้นตอนต่อไปจะเป็นการนำข้อมูลที่ใช้ในการจัดตารางการทำงานพยาบาลทั้งหมด ประกอบไปด้วยจำนวนหัวหน้าพยาบาล 1 คน จำนวนพยาบาลวิชาชีพระดับปฏิบัติการ 16 คน จำนวนผู้ช่วยพยาบาล 2 คน และจำนวนพนักงานฉุกเฉินการแพทย์ 6 คน ภายในแผนกเข้ามาจัดตารางการทำงานตามความต้องการพยาบาลในแต่ละกะการทำงานของแต่ละตำแหน่งในแผนก เพื่อสร้างตารางการทำงานพยาบาลล่วงหน้า 28 วัน ผ่านการประมวลผลด้วย Adaptive Large Neighborhood Search (ALNS) ทำให้ได้ผลเฉลยจำนวน 15 ผลเฉลยที่ดีที่สุด ในระยะเวลาประมวลผลประมาณ 58 นาที โดยมีตัวอย่างของแต่ละผลเฉลยจะแสดงดังตารางที่ 4-12

ตารางที่ 4-12 ผลเฉลยจากวิธีฮิวริสติกสำหรับปัญหาขนาดใหญ่

ผลเฉลยจากฮิวริสติกสำหรับปัญหาขนาดใหญ่				
ผลเฉลย	**Minimize cost**	**Minimize unfairness wages**	**Minimize the violation**	**เวลาในการประมวลผลทั้งหมด**
ผลเฉลยที่ 1	56,284	156,244	168,000	00:58:49:12
ผลเฉลยที่ 2	56,742	123,967	175,700	
ผลเฉลยที่ 3	57,808	70,776	179,100	
ผลเฉลยที่ 4	59,207	23,710	196,500	
ผลเฉลยที่ 5	60,137	11,577	205,200	
ผลเฉลยที่ 6	55,354	139,704	175,200	
ผลเฉลยที่ 7	55,634	124,610	179,000	
ผลเฉลยที่ 8	56,130	67,902	184,300	
ผลเฉลยที่ 9	57,073	16,568	212,200	
ผลเฉลยที่ 10	55,075	176,642	179,600	
ผลเฉลยที่ 11	55,329	107,972	200,600	
ผลเฉลยที่ 12	55,303	58,624	213,500	
ผลเฉลยที่ 13	54,503	163,335	199,900	
ผลเฉลยที่ 14	54,643	117,513	213,800	
ผลเฉลยที่ 15	53,956	148,324	210,100	

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

เนื่องด้วยจำนวนผลเฉลยจากการประมวลผล 1 ครั้ง (1 Replicate) อาจส่งผลให้พยาบาลมีตัวเลือกตารางการทำงานไม่หลากหลาย และเพื่อเพิ่มประสิทธิภาพของคำตอบ ผู้วิจัยจึงได้ทำการประมวลผลจำนวน 10 ครั้ง (10 Replicate) ทำให้ได้ผลเฉลยจำนวน 150 ผลเฉลย จากนั้นนำคำตอบที่ได้ทั้งหมดมาผ่านการแบ่ง Front เพื่อ

คัดเลือกคำตอบที่มีผลเฉลยที่อยู่ใน Front 1 เท่านั้น ซึ่งจะเหลือผลเฉลยอยู่เพียง 54 ผลเฉลย จากทั้งหมด 150 ผลเฉลย ซึ่งการที่บางผลเฉลยไม่ถูกคัดเลือกเกิดจากผลเฉลยถูกครอบงำ (Dominate) จากผลเฉลยอื่นๆ ที่ดีกว่า จึงเป็นเหตุให้เรานำคำตอบเหล่านี้ออกไป เพื่อช่วยให้ตารางการทำงานพยาบาลมีประสิทธิภาพเพิ่มสูงขึ้น

จากตารางที่ 4-13 จะสังเกตได้ว่ามีผลเฉลยจำนวน 54 ผลเฉลยที่ให้ค่าวัตถุประสงค์ที่แตกต่างกันผลเฉลยทั้งหมดนี้จะเป็นผลเฉลยที่ไม่ถูกครอบงำ (Non-dominate) จากผลเฉลยอื่นๆ โดยตารางจะแสดงรายละเอียดของค่าวัตถุประสงค์ของแต่ละผลเฉลย และความแตกต่างเป็นเปอร์เซ็นต์ เมื่อผลเฉลยนั้นๆถูกเปรียบเทียบกับผลเฉลยที่มีค่าวัตถุประสงค์ต่ำที่สุด ซึ่งจะประกอบไปด้วย ผลเฉลยที่ 45 ที่มีค่าวัตถุประสงค์ที่ 1 ต่ำที่สุด มีตารางการทำงานพยาบาลแสดงดังตารางที่ 4-14 ถัดไปจะเป็น ผลเฉลยที่ 40 ที่มีค่าวัตถุประสงค์ที่ 2 ต่ำที่สุด มีตารางการทำงานพยาบาลแสดงดังตารางที่ 4-15 และสุดท้าย ผลเฉลยที่ 1 ที่มีค่าวัตถุประสงค์ที่ 3 ต่ำที่สุด มีตารางการทำงานพยาบาลแสดงดังตารางที่ 4-16 เพื่อแสดงให้เห็นว่าคำตอบแต่ละชุดมีประสิทธิภาพในแต่ละวัตถุประสงค์ที่ห่างจากคำตอบที่ดีที่สุดเป็นกี่เปอร์เซ็นต์

ถึงแม้ว่าผลการวิเคราะห์จากตารางทั้งสาม (ตารางที่ 4-14, ตารางที่ 4-15, ตารางที่ 4-16) จะสามารถตอบสนองต่อเป้าหมายหลักที่เราให้ความสนใจได้อย่างดีเยี่ยม แต่กระนั้นการที่เรามุ่งความสนใจไปที่วัตถุประสงค์เหล่านี้เพียงวัตถุประสงค์เดียว อาจทำให้เราละเลยความสำคัญด้านอื่นๆลง ส่งผลให้วัตถุประสงค์อื่นๆมีประสิทธิภาพที่ต่ำกว่าค่าเฉลี่ย ซึ่งนำไปสู่ความไม่สมดุลในการนำไปใช้กับสถานการณ์จริง โดยผลเฉลยที่ 45 ที่มีค่าวัตถุประสงค์ที่ 1 ต่ำที่สุด แต่มีค่าวัตถุประสงค์ที่ 2 สูงกว่าค่าเฉลี่ย 42.849% และค่าวัตถุประสงค์ที่ 3 สูงกว่าค่าเฉลี่ย 9.232% ถัดไปคือ ผลเฉลยที่ 40 ที่มีค่าวัตถุประสงค์ที่ 2 ต่ำที่สุด แต่มีค่าวัตถุประสงค์ที่ 1 สูงกว่าค่าเฉลี่ย 5.462% และค่าวัตถุประสงค์ที่ 3 สูงกว่าค่าเฉลี่ย 7.065% และสุดท้ายผลเฉลยที่ 1 ที่มีค่าวัตถุประสงค์ที่ 3 ต่ำที่สุด และมีค่าวัตถุประสงค์ที่ 1 ต่ำกว่าค่าเฉลี่ย 1.010% แต่มีค่าวัตถุประสงค์ที่ 2 สูงกว่าค่าเฉลี่ย 45.739% ด้วยความไม่สมดุลในการให้ความสำคัญกับวัตถุประสงค์ ในการเลือกนำไปใช้งานผลเฉลยอื่นที่อาจไม่ได้ให้ค่าวัตถุประสงค์ที่ดีที่สุด แต่สามารถกระจายความสำคัญให้กับแต่ละวัตถุประสงค์ได้มีความใกล้เคียงกัน จึงมีโอกาสที่จะถูกนำไปประยุกต์ใช้ได้มากกว่า

ในการนำไปประยุกต์ใช้เราอาจจะใช้เกณฑ์อื่นๆ เข้ามาร่วมตัดสินใจนำบางผลเฉลยออก เพราะถึงแม้ว่าผลเฉลยใน Front1 จะไม่ถูกครอบงำโดยผลเฉลยอื่นก็ตาม แต่จะต้องคำนึงถึงคุณภาพของวัตถุประสงค์โดยรวมด้วย หากสังเกตรูปที่ 4-2 ที่เป็นการพลอตคำตอบทั้ง 54 คำตอบจาก Front1 บทกราฟ 3 มิติ ในพื้นที่วัตถุประสงค์ (Objective space) จะพบว่ามีทั้งคำตอบที่อยู่ใกล้กับจุดกำเนิด (Origin) และอยู่ไกลออกจากจุดกำเนิด ทำให้เราเห็นว่าคำตอบที่มีความสมดุลในทุกวัตถุประสงค์จะมีพิกัดที่อยู่ใกล้กับจุดกำเนิด กลับกันคำตอบที่ให้ความสำคัญกับบางวัตถุประสงค์มาก แต่ลดความสำคัญกับวัตถุประสงค์อื่นๆลงไป จะมีพิกัดที่อยู่ไกลออกจากจุดกำเนิด เป็นเพราะว่าค่าวัตถุประสงค์ที่ถูกลดความสำคัญมีค่ามาก อาจทำให้ไม่ตรงกับเป้าหมายของการการปัญหาที่ต้องการให้

จุฬาลงกรณ์มหาวิทยาลัย
CHULALONGKORN UNIVERSITY

ค่าวัตถุประสงค์โดยรวมลดต่ำได้มากที่สุดเท่าที่จะทำได้ ซึ่งในการนำไปใช้งานเราอาจเลือกคำตอบเหล่านี้ออก เพื่อให้ตารางการทำงานมีความสมดุลของวัตถุประสงค์ได้ใกล้เคียงกันมากที่สุดเท่าที่จะเป็นไปได้

นอกจากการใช้ผลเฉลยที่ได้จากวิธีแม่นตรงมาใช้เป็นเกณฑ์ในการเปรียบเทียบกับผลเฉลยจากวิธีฮิวริสติก ในงานวิจัยนี้จะนำผลเฉลยที่ได้จากวิธีการฮิวริสติก มาเปรียบเทียบกับตารางการทำงานของพยาบาลในแผนกฉุกเฉินของเดือนกันยายน ที่มีองค์ประกอบของข้อมูลที่ใส่เข้าไป (Input data) ที่เหมือนกันเพื่อแสดงให้เห็นว่าหากมีการนำตารางการทำงานพยาบาลที่ได้จากวิธีฮิวริสติก ไปใช้งานไม่เพียงแต่ละประหยัดเวลา และลดภาระงานในการจัดตาราง แต่ยังสามารถเพิ่มประสิทธิภาพให้ไปในทิศทางที่ดีขึ้นได้อีกด้วย โดยตารางการทำงานของพยาบาลในแผนกฉุกเฉินของเดือนกันยายนจะแสดงดังตารางที่ 4-17 หากเรานำมาคิดค่าวัตถุประสงค์แต่ละมิติแล้วจะประกอบไปด้วย 1.วัตถุประสงค์ที่ 1 = 62,647 , 2.วัตถุประสงค์ที่ 2 = 185,894 ,3.วัตถุประสงค์ที่ 3 = 224,900 ซึ่งค่าเฉลี่ยของผลเฉลยจากวิธีฮิวริสติก จะอยู่ที่ 1.วัตถุประสงค์ที่ 1 = 56,853 , 2.วัตถุประสงค์ที่ 2 = 84,769 ,3.วัตถุประสงค์ที่ 3 = 190,703 ส่งผลให้ประสิทธิภาพโดยเฉลี่ยของผลเฉลยจากฮิวริสติก สามารถพัฒนาประสิทธิภาพจากการจัดตารางแบบดั้งเดิมไปได้ถึง 1.การเปลี่ยนแปลงของวัตถุประสงค์ที่ 1 = 9.249%, 2.การเปลี่ยนแปลงของวัตถุประสงค์ที่ 2 = 54.392%, 3.การเปลี่ยนแปลงของวัตถุประสงค์ที่ 3 = 15.206%

จากการตรวจสอบการใช้ฮิวริสติกในการแก้ปัญหาการจัดตารางการทำงานพยาบาลแสดงให้เห็นว่า ฮิวริสติกไม่เพียงแต่ช่วยให้ตารางการทำงานมีประสิทธิภาพ และมีความยืดหยุ่นในการนำไปใช้งานเท่านั้น แต่ยังเป็นการลดภาระงานในการจัดตารางได้อีกด้วย ซึ่งผลลัพธ์ในการประยุกต์ใช้ฮิวริสติกจะช่วยให้การทำงานของพยาบาลมีประสิทธิภาพสูงขึ้น ได้รับความเป็นธรรมในด้านค่าจ้างมากขึ้น และเป็นการยกระดับคุณภาพชีวิตในการทำงานให้กับพยาบาลทุกคนในแผนก

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

ตารางที่ 4-13 ตารางค่าวัตถุประสงค์ของผลเฉลยจำนวน 54 ผลเฉลยจากการประมวลผล 10 ครั้ง

ผลเฉลย	Minimize cost	Percentage Difference	Minimize unfairness wages	Percentage Difference	Minimize the violation	Percentage Difference
ผลเฉลยที่ 1	56,284	4.31%	156,224	1249.43%	168,000	0.00%
ผลเฉลยที่ 2	59,913	11.04%	79,177	583.92%	168,168	0.10%
ผลเฉลยที่ 3	59,125	9.58%	82,972	616.70%	168,776	0.46%
ผลเฉลยที่ 4	57,964	7.43%	86,936	650.94%	169,151	0.69%
ผลเฉลยที่ 5	58,590	8.59%	86,422	646.50%	170,286	1.36%
ผลเฉลยที่ 6	56,534	4.78%	133,462	1052.82%	170,831	1.69%
ผลเฉลยที่ 7	61,792	14.52%	45,297	291.27%	171,210	1.91%
ผลเฉลยที่ 8	57,399	6.38%	115,893	901.06%	172,595	2.74%

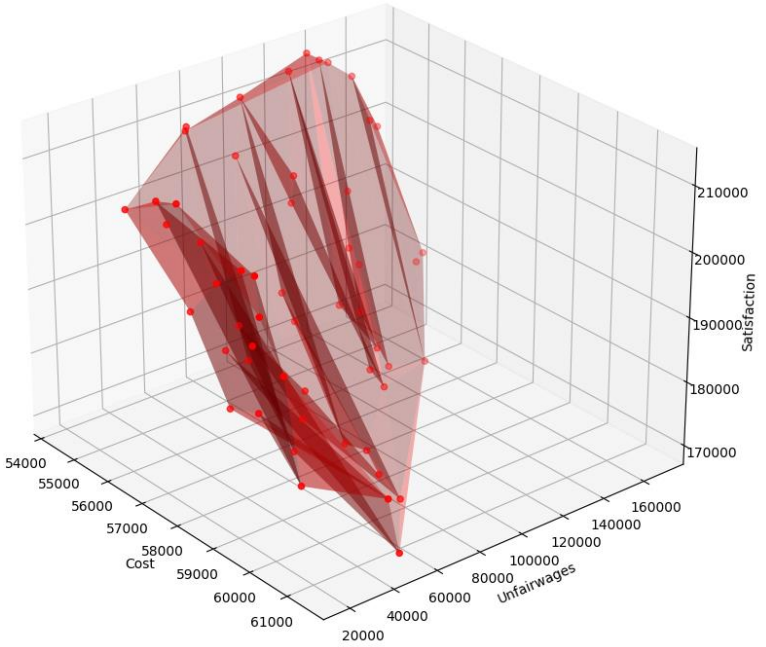
ตารางที่ 4-13 ตารางค่าวัตถุประสงค์ของผลเฉลยจำนวน 54 ผลเฉลยจากการประมวลผล 10 ครั้ง (ต่อ)

ผลเฉลย	Minimize cost	Percentage Difference	Minimize unfairness wages	Percentage Difference	Minimize the violation	Percentage Difference
ผลเฉลยที่ 9	58,307	8.06%	57,129	393.47%	173,063	3.01%
ผลเฉลยที่ 10	56,864	5.39%	118,363	922.40%	173,285	3.15%
ผลเฉลยที่ 11	57,604	6.76%	72,917	529.84%	173,759	3.43%
ผลเฉลยที่ 12	55,354	2.59%	139,704	1106.74%	175,200	4.29%
ผลเฉลยที่ 13	56,742	5.16%	123,967	970.80%	175,700	4.58%
ผลเฉลยที่ 14	55,634	3.11%	124,610	976.36%	179,000	6.55%
ผลเฉลยที่ 15	57,808	7.14%	70,776	511.35%	179,100	6.61%
ผลเฉลยที่ 16	55,192	2.29%	171,125	1378.15%	179,107	6.61%
ผลเฉลยที่ 17	55,662	3.16%	102,278	783.46%	179,241	6.69%
ผลเฉลยที่ 18	55,075	2.07%	176,642	1425.80%	179,600	6.90%
ผลเฉลยที่ 19	61,728	14.40%	41,180	255.71%	179,777	7.01%
ผลเฉลยที่ 20	60,550	12.22%	22,471	94.10%	180,167	7.24%
ผลเฉลยที่ 21	56,640	4.97%	63,680	450.06%	180,995	7.74%
ผลเฉลยที่ 22	55,025	1.98%	145,050	1152.92%	181,247	7.89%
ผลเฉลยที่ 23	57,977	7.45%	33,389	188.41%	181,756	8.19%
ผลเฉลยที่ 24	55,432	2.74%	100,054	764.25%	183,356	9.14%
ผลเฉลยที่ 25	54,834	1.63%	143,332	1138.08%	183,554	9.26%
ผลเฉลยที่ 26	56,130	4.03%	67,902	486.53%	184,300	9.70%
ผลเฉลยที่ 27	59,262	9.83%	25,208	117.74%	186,386	10.94%
ผลเฉลยที่ 28	57,211	6.03%	44,125	281.14%	186,862	11.23%
ผลเฉลยที่ 29	54,710	1.40%	144,853	1151.21%	192,202	14.41%
ผลเฉลยที่ 30	57,090	5.81%	30,451	163.03%	194,101	15.54%
ผลเฉลยที่ 31	54,962	1.86%	112,886	875.09%	194,688	15.89%
ผลเฉลยที่ 32	59,207	9.73%	23,710	104.80%	196,500	16.96%
ผลเฉลยที่ 33	60,740	12.57%	11,900	2.79%	198,487	18.15%
ผลเฉลยที่ 34	54,503	1.01%	163,335	1310.86%	199,900	18.99%
ผลเฉลยที่ 35	56,302	4.35%	48,252	316.79%	200,056	19.08%
ผลเฉลยที่ 36	55,329	2.54%	107,972	832.64%	200,600	19.40%
ผลเฉลยที่ 37	57,936	7.38%	28,897	149.61%	201,146	19.73%
ผลเฉลยที่ 38	54,704	1.39%	156,005	1247.54%	202,219	20.37%
ผลเฉลยที่ 39	56,411	4.55%	31,393	171.17%	205,078	22.07%
ผลเฉลยที่ 40	60,137	11.46%	11,577	0.00%	205,200	22.14%
ผลเฉลยที่ 41	55,074	2.07%	84,776	632.28%	205,552	22.35%
ผลเฉลยที่ 42	59,005	9.36%	22,509	94.43%	207,099	23.27%
ผลเฉลยที่ 43	54,110	0.29%	157,441	1259.95%	207,413	23.46%

จุฬาลงกรณ์มหาวิทยาลัย
CHULALONGKORN UNIVERSITY

ตารางที่ 4-13 ตารางค่าวัตถุประสงค์ของผลเฉลยจำนวน 54 ผลเฉลยจากการประมวลผล 10 ครั้ง (ต่อ)

ผลเฉลย	Minimize cost	Percentage Difference	Minimize unfairness wages	Percentage Difference	Minimize the violation	Percentage Difference
ผลเฉลยที่ 44	56,167	4.10%	16,968	46.57%	208,263	23.97%
ผลเฉลยที่ 45	53,956	0.00%	148,324	1181.20%	210,100	25.06%
ผลเฉลยที่ 46	59,894	11.01%	13,744	18.72%	210,114	25.07%
ผลเฉลยที่ 47	54,898	1.75%	64,529	457.39%	211,098	25.65%
ผลเฉลยที่ 48	54,188	0.43%	139,939	1108.77%	211,992	26.19%
ผลเฉลยที่ 49	57,073	5.78%	16,568	43.11%	212,200	26.31%
ผลเฉลยที่ 50	54,826	1.61%	91,368	689.22%	213,122	26.86%
ผลเฉลยที่ 51	55,303	2.50%	58,624	406.38%	213,500	27.08%
ผลเฉลยที่ 52	54,643	1.27%	117,513	915.06%	213,800	27.26%
ผลเฉลยที่ 53	57,831	7.18%	13,572	17.23%	214,377	27.61%
ผลเฉลยที่ 54	54,410	0.84%	130,117	1023.93%	214,667	27.78%



รูปที่ 4-2 การ Plot 3 มิติ จากผลเฉลยใน Front 1 ด้วยการประมวลผล 10 ครั้ง

ตารางที่ 4-14 ตารางการทำงานพยาบาลจากผลเฉลยที่มีค่าวัตถุประสงค์ที่ 1 ต่ำที่สุด

ตำแหน่ง	ลำดับ	ตารางการทำงานพยาบาลแผนกห้องฉุกเฉิน																											
		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28
		จ	อ	พ	พฤ	ศ	ส	อา	จ	อ	พ	พฤ	ศ	ส	อา	จ	อ	พ	พฤ	ศ	ส	อา	จ	อ	พ	พฤ	ศ	ส	อา
HOD	1	ข	ข	ข	ข	ข	X	X	ข	ข	ข	ข	ข	X	X	ข	ข	ข	ข	ข	X	X	ข	ข	ข	ข	ข	X	X
RN	2	X	X	X	D	X	ข	D	X	ข	X	N	D	X	X	X	D	X	X	D	X	บ	X	X	ข	N	X	X	บ
RN	3	X	N	D	D	บ	X	X	X	X	N	X	X	ข	ข	X	X	N	X	X	11	X	D	X	ข	X	D	บ	X
RN	4	N	X	D	X	D	X	N	ข	X	ข	ข	X	X	บ	X	X	D	X	N	X	X	D	D	บ	X	X	X	ข
RN	5	X	D	X	D	X	X	X	X	D	X	บ	X	บ	X	D	X	X	X	D	X	D	X	N	D	X	X	D	N
RN	6	ข	X	X	N	X	X	ข	N	บ	X	X	N	X	ข	N	บ	X	X	บ	X	X	X	ข	X	ข	X	ข	X
RN	7	D	บ	X	X	N	X	X	ข	X	11	X	บ	X	N	X	N	X	N	X	D	N	X	บ	X	X	N	X	X
RN	8	บ	X	ข	N	X	X	บ	D	X	X	D	X	X	D	X	D	บ	X	D	ข	X	N	X	N	X	X	X	D
RN	9	D	X	X	ข	X	N	ข	X	X	D	X	D	X	X	บ	X	D	X	N	X	D	X	D	X	D	X	ข	X
RN	10	X	N	X	บ	X	ข	N	X	X	D	X	X	ข	X	X	N	X	D	X	X	D	X	X	X	D	X	N	X
RN	11	X	X	X	X	D	X	D	X	D	X	N	ข	N	X	D	X	ข	บ	X	X	X	บ	X	X	N	X	X	N
RN	12	X	D	ข	X	X	D	X	N	ข	ข	X	ข	X	D	X	ข	D	ข	X	X	X	X	N	X	D	N	X	X
RN	13	D	X	N	X	D	X	X	บ	X	X	X	N	X	N	X	X	X	D	X	N	X	N	X	X	X	ข	N	ข
RN	14	X	D	N	X	N	11	X	X	N	X	X	X	D	X	N	X	X	X	X	D	X	D	X	D	X	X	X	D
RN	15	N	X	บ	X	X	X	X	D	X	X	D	X	ข	X	ข	X	X	N	D	X	D	X	D	X	X	D	X	X
RN	16	X	D	X	X	X	N	X	X	N	X	X	X	N	X	D	ข	X	ข	X	ข	N	X	X	X	11	บ	X	X
RN	17	X	X	X	X	ข	D	X	X	X	N	D	X	D	ข	X	X	N	X	X	N	X	ข	X	N	X	ข	D	X
PN	18	D	N	D	N	D	N	D	N	D	N	D	N	D	N	D	N	D	N	D	N	D	N	D	N	D	N	D	N
PN	19	N	D	N	D	N	D	N	D	N	D	N	D	N	D	N	D	N	D	N	D	N	D	N	D	N	D	N	D
EMT	20	ข	X	ข	D	X	D	D	D	ข	N	X	X	D	ข	D	ข	D	ข	X	X	บ	N	บ	X	X	ข	D	D
EMT	21	บ	N	N	X	X	D	X	D	บ	X	D	D	X	X	ข	X	X	D	N	ข	D	X	D	D	N	X	X	บ
EMT	22	X	D	D	N	ข	บ	X	N	X	บ	บ	X	บ	บ	N	X	N	บ	X	N	X	X	N	X	ข	บ	N	ข
EMT	23	D	D	X	X	D	N	ข	X	X	ข	N	ข	ข	N	บ	N	X	N	ข	บ	X	D	D	บ	X	X	บ	N
EMT	24	X	N	X	บ	N	X	บ	บ	N	X	ข	N	X	D	X	บ	บ	X	บ	X	N	บ	X	N	บ	N	ข	X
EMT	25	N	X	บ	ข	บ	X	N	X	D	D	X	บ	N	X	X	D	ข	X	D	D	ข	D	X	ข	D	D	X	X

Objectives Value		
Total Cost	Unfairness Wages	Violations
53,956	148,324	210,100

ตารางที่ 4-15 ตารางการทำงานพยาบาลจากผลเฉลยที่มีค่าวัตถุประสงค์ที่ 2 ต่ำที่สุด

ตำแหน่ง	ลำดับ	ตารางการทำงานพยาบาลแผนกห้องฉุกเฉิน																										
		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28
		จ	อ	พ	พฤ	ศ	ส	อา	จ	อ	พ	พฤ	ศ	ส	อา	จ	อ	พ	พฤ	ศ	ส	อา	จ	อ	พ	พฤ	ศ	ส	อา
HOD	1	ข	ข	ข	ข	ข	X	X	ข	ข	ข	ข	ข	X	X	ข	ข	ข	ข	ข	X	X	ข	ข	ข	ข	ข	X	X
RN	2	X	X	X	D	X	ข	D	X	ข	X	N	D	X	X	X	D	X	X	D	X	บ	X	X	ข	N	X	X	บ
RN	3	X	N	D	D	บ	X	X	X	X	N	X	X	ข	ข	X	X	N	X	X	11	X	D	X	ข	X	D	บ	X
RN	4	N	X	D	X	D	X	N	ข	X	ข	ข	X	X	บ	X	X	D	X	N	X	X	D	D	บ	X	X	X	ข
RN	5	X	D	X	D	X	X	X	X	D	X	บ	X	บ	X	D	X	X	X	D	X	D	X	N	D	X	X	D	N
RN	6	ข	X	X	N	X	X	ข	N	บ	X	X	N	X	ข	N	บ	X	X	บ	X	X	X	ข	X	ข	X	ข	X
RN	7	D	บ	X	X	N	X	X	ข	X	11	X	บ	X	N	X	N	X	N	X	D	N	X	บ	X	X	N	X	X
RN	8	บ	X	ข	N	X	X	บ	D	X	X	D	X	X	D	X	D	บ	X	D	ข	X	N	X	N	X	X	X	D
RN	9	D	X	X	ข	X	N	ข	X	X	D	X	D	X	X	บ	X	D	X	N	X	D	X	D	X	D	X	ข	X
RN	10	X	N	X	บ	X	ข	N	X	X	D	X	X	ข	X	X	N	X	D	X	X	D	X	X	X	D	X	N	X
RN	11	X	X	X	X	D	X	D	X	D	X	N	ข	N	X	D	X	ข	บ	X	X	X	บ	X	X	N	X	X	N
RN	12	X	D	ข	X	X	D	X	N	ข	ข	X	ข	X	D	X	ข	D	ข	X	X	X	X	N	X	D	N	X	X
RN	13	D	X	N	X	D	X	X	บ	X	X	X	N	X	N	X	X	X	D	X	N	X	N	X	X	X	ข	N	ข
RN	14	X	D	N	X	N	11	X	X	N	X	X	X	D	X	N	X	X	X	X	D	X	D	X	D	X	X	X	D
RN	15	N	X	บ	X	X	X	X	D	X	X	D	X	ข	X	ข	X	X	N	D	X	D	X	D	X	X	D	X	X
RN	16	X	D	X	X	X	N	X	X	N	X	X	X	N	X	D	ข	X	ข	X	ข	N	X	X	X	11	บ	X	X
RN	17	X	X	X	X	ข	D	X	X	X	N	D	X	D	ข	X	X	N	X	X	N	X	ข	X	N	X	ข	D	X
PN	18	D	N	D	N	D	N	D	N	D	N	D	N	D	N	D	N	D	N	D	N	D	N	D	N	D	N	D	N
PN	19	N	D	N	D	N	D	N	D	N	D	N	D	N	D	N	D	N	D	N	D	N	D	N	D	N	D	N	D
EMT	20	ข	X	ข	D	X	D	D	D	ข	N	X	X	D	ข	D	ข	D	ข	X	X	บ	N	บ	X	X	ข	D	D
EMT	21	บ	N	N	X	X	D	X	D	บ	X	D	D	X	X	ข	X	X	D	N	ข	D	X	D	D	N	X	X	บ
EMT	22	X	D	D	N	ข	บ	X	N	X	บ	บ	X	บ	บ	N	X	N	บ	X	N	X	X	N	X	ข	บ	N	ข
EMT	23	D	D	X	X	D	N	ข	X	X	ข	N	ข	ข	N	บ	N	X	N	ข	บ	X	D	D	บ	X	X	บ	N
EMT	24	X	N	X	บ	N	X	บ	บ	N	X	ข	N	X	D	X	บ	บ	X	บ	X	N	บ	X	N	บ	N	ข	X
EMT	25	N	X	บ	ข	บ	X	N	X	D	D	X	บ	N	X	X	D	ข	X	D	D	ข	D	X	ข	D	D	X	X

Objectives Value		
Total Cost	Unfairness Wages	Violations
60,137	11,577	205,200

ตารางที่ 4-16 ตารางการทำงานพยาบาลจากผลเฉลยที่มีค่าวัตถุประสงค์ที่ 3 ต่ำที่สุด

ตำแหน่ง	ลำดับ	ตารางการทำงานพยาบาลแผนกห้องฉุกเฉิน																											
		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28
		จ	อ	พ	พฤ	ศ	ส	อา	จ	อ	พ	พฤ	ศ	ส	อา	จ	อ	พ	พฤ	ศ	ส	อา	จ	อ	พ	พฤ	ศ	ส	อา
HOD	1	ข	ข	ข	ข	ข	X	X	ข	ข	ข	ข	ข	X	X	ข	ข	ข	ข	ข	X	X	ข	ข	ข	ข	ข	X	X
RN	2	X	X	X	D	X	ข	D	X	ข	X	N	D	X	X	X	D	X	X	D	X	บ	X	X	ข	N	X	X	บ
RN	3	X	N	D	D	บ	X	X	X	X	N	X	X	ข	ข	X	X	N	X	X	11	X	D	X	ข	X	D	บ	X
RN	4	N	X	D	X	D	X	N	ข	X	ข	ข	X	X	บ	X	X	D	X	N	X	X	D	D	บ	X	X	X	ข
RN	5	X	D	X	D	X	X	X	X	D	X	บ	X	บ	X	D	X	X	X	D	X	D	X	N	D	X	X	D	N
RN	6	ข	X	X	N	X	X	ข	N	บ	X	X	N	X	ข	N	บ	X	X	บ	X	X	X	ข	X	ข	X	ข	X
RN	7	D	บ	X	X	N	X	X	ข	X	11	X	บ	X	N	X	N	X	N	X	D	N	X	บ	X	X	N	X	X
RN	8	บ	X	ข	N	X	X	บ	D	X	X	D	X	X	D	X	D	บ	X	D	ข	X	N	X	N	X	X	X	D
RN	9	D	X	X	ข	X	N	ข	X	X	D	X	D	X	X	บ	X	D	X	N	X	D	X	D	X	D	X	ข	X
RN	10	X	N	X	บ	X	ข	N	X	X	D	X	X	ข	X	X	N	X	D	X	X	D	X	X	X	D	X	N	X
RN	11	X	X	X	X	D	X	D	X	D	X	N	ข	N	X	D	X	ข	บ	X	X	X	บ	X	X	N	X	X	N
RN	12	X	D	ข	X	X	D	X	N	ข	ข	X	ข	X	D	X	ข	D	ข	X	X	X	X	N	X	D	N	X	X
RN	13	D	X	N	X	D	X	X	บ	X	X	X	N	X	N	X	X	X	D	X	N	X	N	X	X	X	ข	N	ข
RN	14	X	D	N	X	N	11	X	X	N	X	X	X	D	X	N	X	X	X	X	D	X	D	X	D	X	X	X	D
RN	15	N	X	บ	X	X	X	X	D	X	X	D	X	ข	X	ข	X	X	N	D	X	D	X	D	X	X	D	X	X
RN	16	X	D	X	X	X	N	X	X	N	X	X	X	N	X	D	ข	X	ข	X	ข	N	X	X	X	11	บ	X	X
RN	17	X	X	X	X	ข	D	X	X	X	N	D	X	D	ข	X	X	N	X	X	N	X	ข	X	N	X	ข	D	X
PN	18	D	N	D	N	D	N	D	N	D	N	D	N	D	N	D	N	D	N	D	N	D	N	D	N	D	N	D	N
PN	19	N	D	N	D	N	D	N	D	N	D	N	D	N	D	N	D	N	D	N	D	N	D	N	D	N	D	N	D
EMT	20	ข	X	ข	D	X	D	D	D	ข	N	X	X	D	ข	D	ข	D	ข	X	X	บ	N	บ	X	X	ข	D	D
EMT	21	บ	N	N	X	X	D	X	D	บ	X	D	D	X	X	ข	X	X	D	N	ข	D	X	D	D	N	X	X	บ
EMT	22	X	D	D	N	ข	บ	X	N	X	บ	บ	X	บ	บ	N	X	N	บ	X	N	X	X	N	X	ข	บ	N	ข
EMT	23	D	D	X	X	D	N	ข	X	X	ข	N	ข	ข	N	บ	N	X	N	ข	บ	X	D	D	บ	X	X	บ	N
EMT	24	X	N	X	บ	N	X	บ	บ	N	X	ข	N	X	D	X	บ	บ	X	บ	X	N	บ	X	N	บ	N	ข	X
EMT	25	N	X	บ	ข	บ	X	N	X	D	D	X	บ	N	X	X	D	ข	X	D	D	ข	D	X	ข	D	D	X	X

Objectives Value		
Total Cost	Unfairness Wages	Violations
56,284	156,244	168,000

ตารางที่ 4-17 ตารางการทำงานของพยาบาลแผนกฉุกเฉินในเดือนกันยายน พ.ศ.2566

ตำแหน่ง	ลำดับ	ตารางการทำงานพยาบาลแผนกห้องฉุกเฉิน																											
		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28
		จ	อ	พ	พฤ	ศ	ส	อา	จ	อ	พ	พฤ	ศ	ส	อา	จ	อ	พ	พฤ	ศ	ส	อา	จ	อ	พ	พฤ	ศ	ส	อา
HOD	1	ช	ช	ช	ช	ช	X	X	ช	ช	ช	ช	ช	X	X	ช	ช	ช	ช	ช	X	X	ช	ช	ช	ช	ช	X	X
RN	2	D	N	ช	X	X	D	X	X	N	X	X	X	X	ช	X	X	ช	X	X	N	N	บ	ช	N	X	X	11	N
RN	3	X	D	X	X	ช	บ	X	บ	N	ศ	D	ช	X	X	บ	X	D	ช	X	X	X	X	บ	11	X	ช	D	X
RN	4	X	D	ช	D	D	ศ	X	X	D	X	D	D	X	X	N	D	X	N	ช	X	N	D	บ	ช	D	D	N	X
RN	5	ช	X	X	X	N	X	X	D	X	X	D	ศ	X	D	D	X	X	ช	X	ศ	บ	N	ช	ช	X	X	X	D
RN	6	X	ช	บ	N	X	ศ	X	X	D	บ	N	ช	X	ช	X	X	D	X	X	X	X	N	X	D	X	ช	X	11
RN	7	X	X	X	N	ศ	X	ช	11	X	D	X	X	D	X	X	X	X	X	N	X	ช	D	N	D	11	X	D	X
RN	8	X	X	X	X	X	N	N	D	ช	X	X	X	X	X	D	D	X	X	X	X	X	ช	D	X	ศ	11	X	X
RN	9	D	X	X	ช	X	X	N	X	X	X	X	N	ช	N	X	ช	X	บ	X	X	X	X	X	X	X	N	N	X
RN	10	N	X	X	X	X	D	X	ช	11	ช	N	X	X	X	N	ช	บ	X	X	D	D	X	D	X	D	D	X	X
RN	11	X	11	X	X	X	X	ช	X	X	X	X	ช	N	บ	X	X	N	D	D	บ	ช	X	N	X	X	X	X	X
RN	12	ช	N	D	11	บ	X	X	X	X	11	บ	X	X	D	ช	N	X	X	X	X	X	X	ศ	X	บ	X	ช	D
RN	13	11	X	N	N	N	X	X	11	X	N	ช	X	D	N	X	X	X	N	ช	X	D	D	N	N	ช	X	D	X
RN	14	N	X	X	X	D	ช	11	X	X	ช	X	บ	D	N	X	X	ศ	X	X	ช	N	11	X	X	X	X	X	X
RN	15	X	X	N	X	X	ช	X	ช	D	X	X	D	X	X	X	N	ช	D	บ	D	X	X	X	N	N	X	X	D
RN	16	X	ช	D	D	11	X	D	ศ	X	D	ช	11	บ	X	X	บ	บ	X	D	N	X	X	X	X	X	X	X	ช
RN	17	11	X	X	D	X	11	D	ศ	X	N	N	X	N	X	ช	X	X	X	N	D	X	D	X	X	ช	N	X	N
PN	18	D	N	D	N	D	N	D	N	D	N	D	N	D	N	D	N	D	N	D	N	D	N	D	N	D	N	D	N
PN	19	N	D	N	D	N	D	N	D	N	D	N	D	N	D	N	D	N	D	N	D	N	D	N	D	N	D	N	D
EMT	20	ช	N	11	X	D	X	ช	N	บ	D	X	ช	N	X	X	D	N	D	N	X	N	D	N	D	X	D	N	X
EMT	21	N	X	X	X	ช	N	X	X	D	ช	N	บ	X	บ	ช	N	D	X	D	N	X	11	D	N	N	X	ศ	ช
EMT	22	D	X	N	ช	11	X	D	ช	N	X	D	X	ช	N	11	D	X	N	D	D	N	X	N	D	D	N	X	บ
EMT	23	บ	D	X	ศ	X	บ	11	บ	ช	N	X	X	D	D	ช	X	N	D	N	X	D	X	X	X	X	ช	D	D
EMT	24	X	ช	D	บ	N	ช	X	D	X	X	ช	N	X	D	บ	X	X	X	X	N	X	N	D	N	D	X	บ	N
EMT	25	X	11	ช	D	X	D	N	X	X	11	บ	D	บ	X	N	N	D	N	X	D	D	ช	X	X	X	บ	D	X

Objectives Value		
Total Cost	Unfairness Wages	Violations
62,647	185,094	224,900

บทที่ 5

สรุปผลการดำเนินงานวิจัย

5.1 สรุปผลการดำเนินงานวิจัย

การจัดตารางการทำงานพยาบาล ถือเป็นหนึ่งในกิจกรรมที่มีความสำคัญสำหรับพยาบาลทุกคน เพราะการได้รับมอบหมายกะการทำงานที่มีความเหมาะสม มีความเท่าเทียมในค่าจ้าง และมีรูปแบบกะการทำงานที่พยาบาลพึงพอใจนั้นเป็นสิ่งที่ส่งผลให้การทำงานมีประสิทธิภาพ เกิดความสุขในการทำงาน ลดความเครียดจากความเหนื่อยล้าสะสม และลดปัญหาการลาออก แต่วิธีการที่สามารถเข้าถึงปัญหาและแก้ปัญหาได้สอดคล้องกับความต้องการต้องอาศัยข้อมูลจากพยาบาลผู้มีประสบการณ์ จากนั้นข้อมูลจะถูกทำให้อยู่ในรูปของสมการทางคณิตศาสตร์และนำมาวิเคราะห์ผ่านโปรแกรมเพื่อหาคำตอบ

ผลการวิเคราะห์หาคำตอบโดยใช้สมการทางคณิตศาสตร์และโปรแกรม IBM ILOG CPLEX Optimizer แสดงให้เห็นว่าเราสามารถแก้ปัญหาที่ซับซ้อนและใช้เวลานาน ได้อย่างรวดเร็วและแม่นยำ โดยสามารถสร้างตารางการทำงานของพยาบาลภายใต้วัตถุประสงค์ที่ได้กำหนดไว้ คือ วัตถุประสงค์ที่ 1 ที่สามารถหาคำตอบที่ดีที่สุดของค่าจ้างพยาบาลที่เหมาะสมที่สุด วัตถุประสงค์ที่ 2 ที่สร้างมาตรฐานความเท่าเทียมกัน และวัตถุประสงค์ที่ 3 มีบทลงโทษหากมีการละเมิดข้อจำกัดที่ได้กำหนดไว้ แต่สมการทางคณิตศาสตร์ก็มีข้อจำกัดเมื่อมีจำนวนพยาบาลเพิ่มสูงขึ้นจะใช้เวลาในการประมวลผลนาน ผู้วิจัยจึงได้นำวิธีฮิวริสติก (Heuristic) มาช่วยแก้ปัญหาให้ใช้เวลาในการประมวลผลสั้นลง สามารถสร้างตารางการทำงานที่มีความเหมาะสมในแต่ละวัตถุประสงค์ได้หลายตาราง และยังมีประสิทธิภาพใกล้เคียงกับการใช้สมการทางคณิตศาสตร์ ทำให้ตารางการทำงานที่ได้จากวิธีฮิวริสติก เป็นตัวเลือกในการนำไปใช้กับแผนกฉุกเฉินได้หลากหลายสถานการณ์

จุฬาลงกรณ์มหาวิทยาลัย
CHULALONGKORN UNIVERSITY

สำหรับการวิจัยนี้ในอนาคตจะเริ่มจากการพัฒนาให้สามารถรองรับข้อจำกัดต่างๆของโรงพยาบาลเอกชนแห่งหนึ่งในประเทศไทย และถูกนำไปใช้กับโรงพยาบาลอื่นๆในประเทศไทยอย่างแพร่หลาย เนื่องจากหลักการในการสร้างตารางนั้นสามารถประยุกต์ใช้ โดยการปรับเปลี่ยนข้อจำกัด และวัตถุประสงค์ในการจัดตารางให้เป็นไปตามความต้องการของโรงพยาบาลแต่ละที่ได้ เพื่อเป็นส่วนหนึ่งของการพัฒนาคุณภาพในการให้บริการทางการแพทย์ คุณภาพชีวิตของผู้ปฏิบัติงาน และยังเป็นการเตรียมรับมือกับเหตุการณ์ที่ประเทศไทยกำลังเป็นประเทศที่เข้าสู่สังคมสูงอายุ (Aging society) ฉะนั้นการเตรียมการรับมือกับความต้องการ (Demand) ของผู้เข้ารับบริการทางการแพทย์นั้นเป็นสิ่งสำคัญ

5.2 ข้อเสนอแนะ

สำหรับในการทำงานวิจัยนี้ผู้วิจัยได้มีโอกาสเก็บข้อมูลจากพยาบาลผู้ปฏิบัติงานจริงทั้งช่องทางออนไลน์ และที่แผนกฉุกเฉิน พบว่าการจัดตารางการทำงานพยาบาลเป็นกิจกรรมที่ใช้เวลานาน และมีความซับซ้อนมาก เนื่องจากโรงพยาบาลมีการปรับเปลี่ยนกฎตามสถานการณ์ตลอดเวลา พยาบาลมีการเพิ่ม-ลด และการลากะทันหัน อันเนื่องมาจาก ป่วย หรือออกเวรนอกสถานที่ด่วน จึงทำให้วิธีการสร้างตารางการทำงานในงานวิจัยนี้ยังไม่สามารถรอบรับกับเหตุการณ์ที่ไม่แน่นอนเหล่านี้ได้ หากมีการพัฒนาต่อยอดที่ทำให้ตารางการทำงานพยาบาลง่ายต่อการแก้ไข และสามารถปรับเปลี่ยนได้ตามสถานการณ์จะเป็นปัจจัยสำคัญที่ช่วยให้พยาบาลสามารถประยุกต์ใช้ตารางให้เกิดประสิทธิภาพที่ดีขึ้นได้

อีกหนึ่งมิติที่ควรคำนึงถึงในอนาคต คือ การคำนึงถึงครอนอไทป์ (Chronotype) หรือระดับการตื่นตัวเข้ามาเป็นส่วนหนึ่งในตัวแปรตัดสินใจมอบหมายกะการทำงานให้กับพยาบาล ด้วยลักษณะการปฏิบัติงานของพยาบาลที่มีกะการทำงานตลอด 24 ชั่วโมง พยาบาลแต่ละคนจะมีครอนอไทป์ในแต่ละช่วงที่แตกต่างกันไป จึงเป็นเหตุผลหนึ่งที่พยาบาลแต่ละคนมีความต้องการเข้าทำงานในแต่ละชนิดกะการทำงานที่แตกต่างกัน โดยในจังหวะทางชีวภาพของมนุษย์ (Circadian rhythm) จะมีช่วงเวลาของวงจรประมาณ 24 ชั่วโมง แบ่งออกเป็น 3 ประเภท คือ มีความตื่นตัวในช่วงเช้า (Morning type) มีความตื่นตัวในช่วงเย็น (Evening type) และ กลุ่มที่ไม่ได้ อยู่ในกลุ่ม Morning และ Evening type (Intermediate type) ซึ่งการคำนึงถึงครอนอไทป์จะช่วยให้การจัดตารางการทำงานพยาบาลสร้างความพึงพอใจ และเสริมสร้างความสุขให้กับพยาบาลได้เป็นอย่างดี (Daranee Pipatkulchai, 2021)

ถึงแม้ว่าในงานวิจัยนี้จะเป็นการสร้างตารางการทำงานพยาบาล แต่แนวทาง และหลักการที่ได้จากการวิจัยนี้สามารถถูกนำไปประยุกต์ใช้กับบุคลากรทางการแพทย์ในตำแหน่งอื่นๆ ได้เช่นกัน เพราะว่าการทำงานในสาขาสุขภาพมักต้องเผชิญกับสถานการณ์ที่เปลี่ยนแปลงอย่างรวดเร็วที่มีความคล้ายคลึงกับพยาบาล ดังนั้นการมีระบบการเข้ามาช่วย ไม่เพียงแต่จะช่วยให้การจัดสรรทรัพยากรมนุษย์ และการบริหารจัดการเวลามีประสิทธิภาพยิ่งขึ้นเท่านั้น แต่ยังช่วยในการปรับปรุงคุณภาพการดูแลผู้ป่วย และลดภาระงานของบุคลากรทางการแพทย์ในตำแหน่งอื่นๆอีกด้วย

นอกจากนี้โปรแกรมการจัดตารางการทำงานพยาบาลในงานวิจัยนี้ยังคงมีข้อจำกัดในด้านความง่ายต่อการใช้งานของโปรแกรม ซึ่งในการสร้างตารางการทำงานใหม่แต่ละครั้งผู้วิจัยยังมีความจำเป็นต้องเป็นคนสร้างตารางให้และส่งให้กับหัวหน้าพยาบาล ในข้อจำกัดส่วนนี้หากได้รับการพัฒนารูปแบบของโปรแกรมให้มีความใช้งานได้ง่ายขึ้นในระดับที่หัวหน้าพยาบาลสามารถจัดตารางการทำงานโดยใช้โปรแกรมได้ด้วยตนเอง โปรแกรมการจัดตารางการทำงานจะยิ่งมีความเหมาะสมต่อการนำไปงาน

อีกหนึ่งสิ่งที่มีความสำคัญในการพัฒนาโปรแกรมให้ดียิ่งขึ้น และยังเป็นข้อจำกัดของโปรแกรมจัดตาราง คือ โปรแกรมยังไม่สามารถรับมือกับการจัดตารางการทำงานใหม่ด้วยการใช้ข้อมูลตารางในอดีตเข้ามาเป็นส่วนหนึ่งในการจัดตาราง เช่น มีพยาบาลลาหยุดกะทันหัน หรือมีเหตุฉุกเฉิน เป็นเหตุให้ต้องจัดตารางการทำงานใหม่ โดยต้องใช้ข้อมูลตารางการทำงานก่อนหน้าเข้ามาตัดสินใจร่วมด้วย ด้วยข้อจำกัดในส่วนนี้โปรแกรมการจัดตารางในปัจจุบันจะสามารถสร้างตารางการทำงานพยาบาลได้เพียงตารางตอนต้นเดือนที่บอกกะการทำงานล่วงหน้าให้กับพยาบาลทุกคนในแผนกในเดือนนั้นๆ หากจะทำการแก้ไขตารางการทำงานยังคงจะต้องอาศัยการตัดสินใจ และการจัดตารางใหม่ด้วยตัวหัวหน้าพยาบาล

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

บรรณานุกรม

Aickelin, U., & White, P. (2004). Building better nurse scheduling algorithms. *Annals of Operations Research*, *128*, 159-177.

Bao, C., Xu, L., Goodman, E. D., & Cao, L. (2017). A novel non-dominated sorting algorithm for evolutionary multi-objective optimization. *Journal of Computational Science*, *23*, 31-43.

Bard, J. F., & Purnomo, H. W. (2005). Preference scheduling for nurses using column generation. *European Journal of Operational Research*, *164*(2), 510-534.

Bard, J. F., & Wan, L. (2008). Workforce design with movement restrictions between workstation groups. *Manufacturing & Service Operations Management*, *10*(1), 24-42.

Berrada, I., Ferland, J. A., & Michelon, P. (1996). A multi-objective approach to nurse scheduling with both hard and soft constraints. *Socio-economic planning sciences*, *30*(3), 183-193.

Bilgin, B., De Causmaecker, P., Rossie, B., & Vanden Berghe, G. (2012). Local search neighbourhoods for dealing with a novel nurse rostering model. *Annals of Operations Research*, *194*, 33-57.

Boonprasert, U. (1978). *A comprehensive model for higher education institutional planning: the case of Chulalongkorn university, Thailand*. University of Pittsburgh.

Burke, E. K., Causmaecker, P. D., Petrovic, S., & Berghe, G. V. (2006). Metaheuristics for handling time interval coverage constraints in nurse scheduling. *Applied Artificial Intelligence*, *20*(9), 743-766.

Burke, E. K., De Causmaecker, P., Berghe, G. V., & Van Landeghem, H. (2004). The state of the art of nurse rostering. *Journal of scheduling*, *7*, 441-499.

Burke, E. K., Li, J., & Qu, R. (2012). A Pareto-based search methodology for multi-objective nurse scheduling. *Annals of Operations Research*, *196*, 91-109.

Černý, V. (1985). Thermodynamical approach to the traveling salesman problem: An efficient simulation algorithm. *Journal of optimization theory and applications*, *45*, 41-51.

Chang, K.-H. (2015). Multiobjective optimization and advanced topics. *e-Design*, 1105-1173.

Dantzig, G. B. (1954). A comment on Edie's “Traffic delays at toll booths”. *Journal of the Operations Research Society of America*, *2*(3), 339-341.

Daranee Pipatkulchai, A. P., Ratree Chimchalong,Janthana Namthep. (2021). Chronotypes and

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

Scheduling Needs of Nurses Operating in University-Affiliated Hospitals. *The Journal of Thailand Nursing and Midwifery Council*.

Das, I., & Dennis, J. E. (1998). Normal-boundary intersection: A new method for generating the Pareto surface in nonlinear multicriteria optimization problems. *SIAM journal on optimization*, *8*(3), 631-657.

Deb, K., & Jain, H. (2013). An evolutionary many-objective optimization algorithm using reference-point-based nondominated sorting approach, part I: solving problems with box constraints. *IEEE Transactions on Evolutionary Computation*, *18*(4), 577-601.

Drozdik, M., Akimoto, Y., Aguirre, H., & Tanaka, K. (2014). Computational cost reduction of nondominated sorting using the M-front. *IEEE Transactions on Evolutionary Computation*, *19*(5), 659-678.

Drug And Medical Supply Information Center, M. o. P. H. (2564). https://ops.moph.go.th/public/

Dumrongsiri, A., & Chongphaisal, P. (2018). Nurse scheduling in a hospital emergency department: A case study at a Thai university hospital. *Songklanakarin Journal of Science & Technology*, *40*(1).

El Adoly, A. A., Gheith, M., & Fors, M. N. (2018). A new formulation and solution for the nurse scheduling problem: A case study in Egypt. *Alexandria engineering journal*, *57*(4), 2289-2298.

Fang, H., Wang, Q., Tu, Y.-C., & Horstemeyer, M. F. (2008). An efficient non-dominated sorting method for evolutionary algorithms. *Evolutionary computation*, *16*(3), 355-384.

Franz, L. S., Baker, H. M., Leong, G. K., & Rakes, T. R. (1989). A mathematical model for scheduling and staffing multiclinic health regions. *European Journal of Operational Research*, *41*(3), 277-289.

Glass, C. A., & Knight, R. A. (2010). The nurse rostering problem: A critical appraisal of the problem structure. *European Journal of Operational Research*, *202*(2), 379-389.

Gu, Z.-M., & Wang, G.-G. (2020). Improving NSGA-III algorithms with information feedback models for large-scale many-objective optimization. *Future Generation Computer Systems*, *107*, 49-69.

Gutjahr, M., Parragh, S. N., & Tricoire, F. (2023). Adaptive large neighborhood search for a personnel task scheduling problem with task selection and parallel task assignments. *arXiv*

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

preprint arXiv:2302.04494.

Gutjahr, W. J., & Rauner, M. S. (2007). An ACO algorithm for a dynamic regional nurse-scheduling problem in Austria. *Computers & Operations Research*, *34*(3), 642-666.

Haspeslagh, S., De Causmaecker, P., Schaerf, A., & Stølevik, M. (2014). The first international nurse rostering competition 2010. *Annals of Operations Research*, *218*, 221-236.

Hochba, D. S. (1997). Approximation algorithms for NP-hard problems. *ACM Sigact News*, *28*(2), 40-52.

Huang, H., Lin, W., Lin, Z., Hao, Z., & Lim, A. (2014). An evolutionary algorithm based on constraint set partitioning for nurse rostering problems. *Neural Computing and Applications*, *25*, 703-715.

Joy, G., Huyck, C., & Yang, X.-S. (2023). Review of Parameter Tuning Methods for Nature-Inspired Algorithms. *Benchmarks and Hybrid Algorithms in Optimization and Applications*, 33-47.

Kellogg, D. L., & Walczak, S. (2007). Nurse scheduling: from academia to implementation or not? *Interfaces*, *37*(4), 355-369.

Kirkpatrick, S., Gelatt Jr, C. D., & Vecchi, M. P. (1983). Optimization by simulated annealing. *science*, *220*(4598), 671-680.

Krairiksh, S., & Dusadeerungsikul, P. O. (2022). Multi-Compartment Vehicle Loading and Route Optimization: Case for Exporting Thai Fresh Fruits.

Legrain, A., Bouarab, H., & Lahrichi, N. (2015). The nurse scheduling problem in real-life. *Journal of medical systems*, *39*, 1-11.

Lin, F.-T., Kao, C.-Y., & Hsu, C.-C. (1993). Applying the genetic approach to simulated annealing in solving some NP-hard problems. *IEEE Transactions on systems, man, and cybernetics*, *23*(6), 1752-1767.

Maenhout, B., & Vanhoucke, M. (2010). Branching strategies in a branch-and-price approach for a multiple objective nurse scheduling problem. *Journal of scheduling*, *13*, 77-93.

Maenhout, B., & Vanhoucke, M. (2013). An integrated nurse staffing and scheduling analysis for longer-term nursing staff allocation problems. *Omega*, *41*(2), 485-499.

Mansini, R., Zanella, M., & Zanotti, R. (2023). Optimizing a complex multi-objective personnel scheduling problem jointly complying with requests from customers and staff. *Omega*, *114*,

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

102722.

Mara, S. T. W., Norcahyo, R., Jodiawan, P., Lusiantoro, L., & Rifai, A. P. (2022). A survey of adaptive large neighborhood search algorithms and applications. *Computers & Operations Research*, *146*, 105903.

Martin, S., Ouelhadj, D., Smet, P., Berghe, G. V., & Özcan, E. (2013). Cooperative search for fair nurse rosters. *Expert Systems with Applications*, *40*(16), 6674-6683.

Mohan, S., & Sinha, A. (2023). Elitist Non-dominated Sorting directional Bat algorithm (ENSdBA). *Expert Systems with Applications*, *227*, 120292.

Nobil, A. H., Sharifnia, S. M. E., & Cárdenas-Barrón, L. E. (2022). Mixed integer linear programming problem for personnel multi-day shift scheduling: A case study in an Iran hospital. *Alexandria engineering journal*, *61*(1), 419-426.

Rerkjirattikal, P., Huynh, V.-N., Olapiriyakul, S., & Supnithi, T. (2020). A goal programming approach to nurse scheduling with individual preference satisfaction. *Mathematical Problems in Engineering*, *2020*, 1-11.

Resende, M. G., de Sousa, J. P., Burke, E., De Causmaecker, P., Petrovic, S., & Berghe, G. V. (2004). Variable neighborhood search for nurse rostering problems. *Metaheuristics: computer decision-making*, 153-172.

Ropke, S., & Pisinger, D. (2006). A unified heuristic for a large class of vehicle routing problems with backhauls. *European Journal of Operational Research*, *171*(3), 750-775.

Sacramento, D., Pisinger, D., & Ropke, S. (2019). An adaptive large neighborhood search metaheuristic for the vehicle routing problem with drones. *Transportation Research Part C: Emerging Technologies*, *102*, 289-315.

Shaw, P. (1998). Using constraint programming and local search methods to solve vehicle routing problems. International conference on principles and practice of constraint programming,

Smet, P., Bilgin, B., De Causmaecker, P., & Vanden Berghe, G. (2014). Modelling and evaluation issues in nurse rostering. *Annals of Operations Research*, *218*, 303-326.

Svirsko, A. C., Norman, B. A., Rausch, D., & Woodring, J. (2019). Using mathematical modeling to improve the emergency department nurse-scheduling process. *Journal of emergency nursing*, *45*(4), 425-432.

Vanden Berghe, G. (2002). An advanced model and novel meta-heuristic solution methods to

จุฬาลงกรณ์มหาวิทยาลัย
CHULALONGKORN UNIVERSITY

personnel scheduling in healthcare.

Wang, F.-S., & Chen, L.-H. (2013). Heuristic Optimization. In W. Dubitzky, O. Wolkenhauer, K.-H. Cho, & H. Yokota (Eds.), *Encyclopedia of Systems Biology* (pp. 885-885). Springer New York. https://doi.org/10.1007/978-1-4419-9863-7_411

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

ประวัติผู้เขียน

ชื่อ-สกุล นายนลธวัช สุคนธ์ภัทร

วัน เดือน ปี เกิด 17 พฤศจิกายน 2539

สถานที่เกิด จังหวัดสมุทรปราการ

วุฒิการศึกษา วิศวกรรมศาสตรบัณฑิต สาขาวิชาวิศวกรรมเครื่องกล
มหาวิทยาลัยเทคโนโลยีพระจอมเกล้าพระนครเหนือ

ที่อยู่ปัจจุบัน 40 ซอย.30 ถนนท้ายบ้าน ตำบลปากน้ำ อำเภอเมือง สมุทรปราการ 10270

จุฬาลงกรณ์มหาวิทยาลัย
CHULALONGKORN UNIVERSITY