บทที่ 3
การพัฒนาวิธีการหาคำตอบ

ในบทนี้ผู้วิจัยได้นำเสนอการพัฒนาวิธีการหาคำตอบของปัญหา โดยกรณีศึกษาที่ผู้วิจัยเลือกใช้สำหรับเป็นแนวทางในการแก้ปัญหาการรับและส่งสินค้าให้ส่งทันภายในระยะเวลารับประกันนั้น คือปัญหาการรับและส่งเวชระเบียนในโรงพยาบาล สำหรับการพัฒนาวิธีการหาคำตอบผู้วิจัยจะนำเสนอแบบจำลองทางคณิตศาสตร์ของปัญหางานวิจัยก่อน ซึ่งการหาคำตอบที่เหมาะสมที่สุดจากแบบจำลองทางคณิตศาสตร์นั้นใช้เวลาค่อนข้างนานในการหาคำตอบ และสามารถประยุกต์ใช้ได้กับปัญหาขนาดเล็กเท่านั้น ผู้วิจัยจึงได้พัฒนาฮิวริสติกสำหรับแก้ปัญหาขึ้น ซึ่งสามารถหาคำตอบได้ใกล้เคียงกับคำตอบที่เหมาะสมที่สุดและใช้เวลาในการหาคำตอบรวดเร็วกว่าการหาคำตอบจากแบบจำลองทางคณิตศาสตร์

3.1 ลักษณะการรับและส่งเวชระเบียนสำหรับผู้ป่วยนอกภายในโรงพยาบาล

งานวิจัยนี้จะศึกษาการรับและส่งเวชระเบียนสำหรับผู้ป่วยนอกในกรณีที่ผู้ป่วยไม่มีใบนัดหรือมาผิดนัด เมื่อผู้ป่วยมาถึงโรงพยาบาลผู้ป่วยจะต้องทำการติดต่อที่หน่วยเวชระเบียนซึ่งเป็นหน่วยงานที่ทำหน้าที่จัดการเวชระเบียนของผู้ป่วยทั้งหมดเสียก่อน สำหรับผู้ป่วยใหม่ให้ทำการขอแบบฟอร์มเขียนประวัติและยื่นให้หน่วยเวชระเบียนพร้อมทั้งแจ้งอาการป่วย จากนั้นผู้ป่วยรอเรียกชื่อเพื่อรับเอกสารและใบแนะนำหน่วยตรวจโรค ซึ่งเป็นใบที่แจ้งให้ผู้ป่วยไปรอรับบริการที่หน้าหน่วยตรวจโรคที่สอดคล้องกับอาการของผู้ป่วย เช่นหน่วยตรวจโรคจักษุ หน่วยตรวจโรคหู คอ จมูก หน่วยตรวจโรคผิวหนัง หน่วยตรวจโรคอายุรศาสตร์ เป็นต้น สำหรับผู้ป่วยเก่าซึ่งมาผิดนัด ให้แจ้งชื่อและหน่วยตรวจโรคที่ต้องการตรวจเพื่อให้เจ้าหน้าที่ค้นประวัติ จากนั้นจึงไปรอที่หน้าหน่วยตรวจโรคเพื่อรอเข้าพบแพทย์ ในกรณีนี้เวชระเบียนของผู้ป่วยควรจะถูกนำไปส่งยังหน่วยตรวจโรคเร็วที่สุดเท่าที่จะเป็นไปได้เพื่อลดเวลารอคอยของผู้ป่วย และเพื่อเป็นการสร้างความพึงพอใจให้กับผู้ป่วย จึงควรทำการกำหนดเวลารอคอยที่มากที่สุดไว้เพื่อรับประกันว่าเวชระเบียนจะถูกส่งถึงหน่วยตรวจโรคของผู้ป่วยไม่เกินระยะเวลานี้ ตัวอย่างเช่นถ้าหากกำหนดให้ระยะเวลาที่ให้ผู้ป่วยคอยเวชระเบียนสูงสุดคือ 30 นาที สำหรับผู้ป่วยที่มาผิดนัดและแจ้งเข้าขอตรวจ ณ เวลา 9 โมงตรง เวชระเบียนของผู้ป่วยควรจะถูกส่งถึงหน่วยตรวจโรคภายในเวลา 9 โมงครึ่ง

นอกจากลักษณะของการขนส่งเวชระเบียนดังที่ได้กล่าวไปข้างต้นนั้น เวชระเบียนของผู้ป่วยอาจยังต้องถูกขนส่งระหว่างหน่วยตรวจโรคด้วยในกรณีที่ผู้ป่วยแต่ละรายอาจจำเป็นต้องได้รับการตรวจโรคมากกว่าหนึ่งหน่วย สำหรับการขนส่งเวชระเบียนไปยังหน่วยตรวจโรคต่าง ๆ นั้นจะทำการขนส่งโดยใช้รถเข็นในกรณีที่เวชระเบียนที่ต้องทำการขนส่งนั้นมีจำนวนมาก หรืออาจทำการขนส่งโดยให้เจ้าหน้าที่ถือไปส่งให้ที่หน่วยตรวจโรคในกรณีที่เวชระเบียนที่ต้องทำการขนส่งนั้นมีจำนวนน้อย หลังจากตรวจผู้ป่วยเสร็จเรียบร้อยแล้วเวชระเบียนจะถูกขนส่งกลับมาเก็บยังห้องเก็บเวชระเบียน

3.2 รูปแบบปัญหางานวิจัย

รูปแบบของปัญหางานวิจัย ถูกแทนในรูปของกราฟ (Graph) ซึ่งประกอบไปด้วย 3 ส่วน ได้แก่

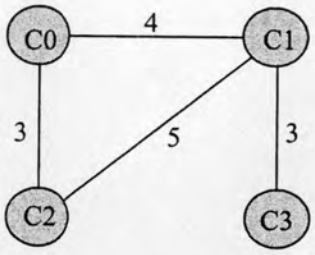
1) โหนด (Node) ใช้แทนหน่วยเวชระเบียนซึ่งเป็นที่ที่รถเข็นจอดคอยอยู่และหน่วยตรวจโรค

2) ด้าน (Edge) ใช้แทนเส้นทางเชื่อมระหว่างโหนด 2 โหนดใด ๆ

3) น้ำหนักบนด้าน (Weight) ใช้แทนระยะทางระหว่างโหนด 2 โหนดใด ๆ

เราสามารถเขียนแทนเป็นสัญลักษณ์ได้ว่ากราฟ $G=(N_0,E)$ โดยที่ $N_0=\{\text{C0}\}\cup N$ ซึ่ง $N=\{\text{C1},\ldots,\text{C}n\}$ ให้โหนด C0 ใช้แทนหน่วยเวชระเบียนซึ่งเป็นที่ที่รถเข็นจอดคอยอยู่และโหนด $i\in N$ ใช้แทนหน่วยตรวจโรค ด้าน $(i,j)\in E$ เมื่อ $i,j\in N_0$ คือระยะทางระหว่างโหนด i และโหนด j ให้ $R=\{1,\ldots,\bar{r}\}$ คือเซตของเวชระเบียนที่ต้องทำการขนส่งทั้ง $\bar{r}$ ชิ้น ซึ่งมีต้นทางและปลายทางอยู่ที่หน่วยเวชระเบียนหรือหน่วยตรวจโรคหน่วยใดหน่วยหนึ่ง พนักงานขนส่งเริ่มต้นอยู่ที่หน่วยเวชระเบียน ทำการรับเวชระเบียนถ้าหากมีเวชระเบียนที่ต้องทำการขนส่งไปยังหน่วยตรวจโรคใด ๆ หน่วยแต่ละหน่วยสามารถถูกเยี่ยมได้มากกว่า 1 ครั้งเพื่อทำการขนส่งเวชระเบียนทุกชิ้นให้ส่งทันถึงปลายทางภายในระยะเวลารับประกัน หลังจากทำการขนส่งเสร็จแล้ว พนักงานจึงนำรถเข็นกลับมาจอดที่หน่วยเวชระเบียน

ปัญหางานวิจัยนี้มีวัตถุประสงค์หลักคือต้องการหาจำนวนรถเข็นที่ใช้ในการขนส่งที่น้อยที่สุดและมีวัตถุประสงค์รองคือต้องการหาระยะทางในการเดินทางรวมที่ต่ำที่สุด โดยที่เวชระเบียนทุกชิ้นจะได้รับการขนส่งถึงปลายทางภายในระยะเวลารับประกัน เพื่อสร้างความพึงพอใจให้กับผู้ป่วยไม่ให้เสียเวลาคอยนาน ตัวอย่างของรูปแบบปัญหาเป็นดังรูปที่ 3.1



รูปที่ 3.1 กราฟ G แทนตัวอย่างปัญหาการรับและส่งเวชระเบียน ให้ส่งทันภายในระยะเวลารับประกัน

ตารางที่ 3.1 ต้นทางและปลายทางของเวชระเบียน

เวชระเบียนชิ้นที่	เวลาพร้อมในการขนส่ง	โหนดต้นทาง	โหนดปลายทาง
01	5	C2	C3
02	10	C1	C2
03	10	C0	C1
04	15	C0	C1
05	15	C0	C3

*กำหนดให้ระยะเวลารับประกันคือ 20 นาที

*กำหนดให้ระยะเวลาในการให้บริการรับและส่งเวชระเบียนคือ 1 นาทีต่อชิ้น

*กำหนดให้ระยะเวลาในการเดินทางคือ 1 นาทีต่อหน่วยระยะทาง

จากรูปที่ 3.1 และตารางที่ 3.1 เป็นตัวอย่างปัญหาที่มีหน่วยเวชระเบียนและหน่วยตรวจโรค 3 หน่วย โดยที่มีเวชระเบียนที่ต้องทำการขนส่งรวมทั้งสิ้น 5 ชิ้นด้วยกัน โหนด C0 ใช้แทนหน่วยเวชระเบียน โหนด C1 C2 และ C3 ใช้แทนหน่วยตรวจโรคที่ 1 2 และ 3 ตามลำดับ ข้อมูลของเวชระเบียนแต่ละชิ้นประกอบไปด้วยเวลาที่พร้อมได้รับการขนส่ง โหนดต้นทางและโหนดปลายทาง โดยกำหนดให้ระยะเวลารับประกันคือ 20 นาที ระยะเวลาในการให้บริการรับและส่งเวชระเบียนคือ 1 นาทีต่อชิ้น และ ระยะเวลาในการเดินทางคือ 1 นาทีต่อหน่วยระยะทาง

3.3 การเปลี่ยนรูปแบบปัญหางานวิจัย

จากรูปแบบปัญหาดังกล่าวพบว่าไม่ได้มีเส้นทางเชื่อมทุกโหนดครบทั้งหมด ดังนั้นผู้วิจัยจึงได้พัฒนาวิธีการหาคำตอบ โดยทำการคิดวิธีการเปลี่ยนรูปแบบปัญหาให้อยู่ในรูปแบบกราฟบริบูรณ์เพื่อทำการสร้างแบบจำลองทางคณิตศาสตร์

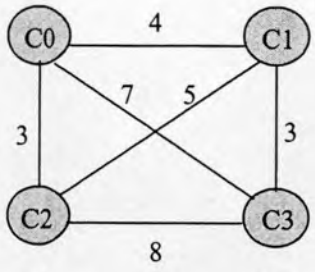
จากรูปแบบปัญหาดั้งเดิม จะเห็นว่าทุกโหนดไม่ได้มีเส้นทางเชื่อมโดยตรงเสมอไป การเดินทางระหว่างโหนดอาจต้องเดินทางผ่านโหนดอื่นด้วย อย่างเช่นถ้าหากต้องการเดินทางจากโหนด C0 ไปยังโหนด C3 สามารถเดินทางโดย C0 – C2 – C1 – C3 หรือ C0 – C1 – C3 ก็ได้ แต่ทั้ง 2 เส้นทางมีระยะเดินทางรวมที่ต่างกัน ดังนั้นในขั้นต้นผู้วิจัยจึงทำการสร้างกราฟบริบูรณ์ โดยให้ระยะทางระหว่าง 2 โหนดใด ๆ คือระยะทางของทางเดินที่สั้นที่สุด (Shortest Path) ระหว่าง 2 โหนดนั้น ๆ เสียก่อน ซึ่งสามารถหาทางเดินที่สั้นที่สุดระหว่าง 2 โหนดใด ๆ ได้โดยใช้ขั้นตอนวิธีของไดสตรา (Dijkstra's Algorithm) (วนิดา เหมะกุล, 2535)

กำหนดให้กราฟบริบูรณ์ $G' = (N_0, E')$ เป็นกราฟแทนปัญหางานวิจัย โดยที่ระยะทางบนด้าน $(i, j) \in E'$ เมื่อ $i, j \in N_0$ คือระยะทางของทางเดินที่สั้นที่สุดระหว่างโหนด i และโหนด j จากตัวอย่างในรูปที่ 3.1 จะได้ว่าทางเดินที่สั้นที่สุดระหว่าง 2 โหนดใด ๆ เป็นดังตารางที่ 3.2

ตารางที่ 3.2 ทางเดินที่สั้นที่สุดและระยะทางระหว่าง 2 โหนดใด ๆ ของกราฟ *G*

คู่ของโหนด	ทางเดินที่สั้นที่สุด (Shortest Path)	ระยะทาง
(C0 , C1)	C0 – C1	4
(C0 , C2)	C0 – C2	3
(C0 , C3)	C0 – C1 – C3	7
(C1 , C2)	C1 – C2	5
(C1 , C3)	C1 – C3	3
(C2 , C3)	C2 – C1 – C3	8

จากตารางที่ 3.2 นำไปสร้างกราฟบริบูรณ์ G' ได้ดังรูปที่ 3.2

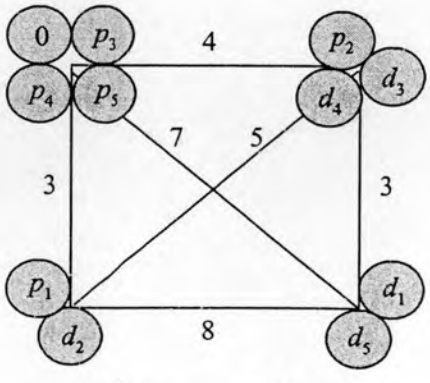


รูปที่ 3.2 กราฟบริบูรณ์ G'

อย่างไรก็ตามหลังจากทำการแปลงกราฟ G เป็นกราฟบริบูรณ์ G' แล้ว จะเห็นว่าแต่ละโหนดนั้นสามารถถูกเยี่ยมได้มากกว่าหนึ่งครั้ง เนื่องจากเวชระเบียนบางชิ้นสามารถถูกรับหรือส่งที่โหนดเดียวกันได้ เพื่อทำการสร้างแบบจำลองทางคณิตศาสตร์ ผู้วิจัยจึงดัดแปลงกราฟ G' อีกครั้ง โดยการใช้ dummy node แทนสำหรับการรับหรือส่งเวชระเบียนแต่ละชิ้น

สำหรับปัญหาที่มีเวชระเบียนที่ต้องทำการขนส่งทั้งสิ้น $\bar{r}$ ชิ้น ให้ $P = \{p_1, p_2, ..., p_{\bar{r}}\}$ คือเซตของโหนดที่ต้องทำการรับเวชระเบียนชิ้นที่ $r \in R$ และให้ $D = \{d_1, d_2, ..., d_{\bar{r}}\}$ คือเซตของโหนดที่ให้บริการส่งเวชระเบียนชิ้นที่ $r \in R$ และกำหนดให้ $N'' = P \cup D$ และ $N_0'' = \{0\} \cup N''$ โดยที่โหนด 0 ใช้แทนที่จอดรถซึ่งอยู่ที่หน่วยเวชระเบียน เราจะทำการสร้างรูปแบบปัญหาใหม่ให้อยู่ในรูปของกราฟแบบบริบูรณ์ $G'' = (N_0'', E'')$ สำหรับระยะทางของ $(i, j) \in E''$ โดยที่ $i, j \in N_0''$ คือระยะทางของทางเดินที่สั้นที่สุดระหว่างโหนด $i \in N''$ ไปยังโหนด $j \in N''$ สังเกตว่าถ้าหากโหนด i และ j อยู่ที่ตำแหน่งเดียวกัน ระยะทางระหว่าง 2 โหนดนี้จะมีค่าเป็นศูนย์

จากกราฟบริบูรณ์ G' ดังรูปที่ 3.2 สามารถเปลี่ยนรูปแบบปัญหาเป็นกราฟบริบูรณ์ G'' ได้ดังรูปที่ 3.3



รูปที่ 3.3 กราฟบริบูรณ์ G''

จากรูปที่ 3.3 โหนด 0 หมายถึงหน่วยเวชระเบียนซึ่งเป็นจุดเริ่มต้นและสิ้นสุดของการขนส่ง โหนด p_1 p_2 p_3 p_4 และ p_5 หมายถึงโหนดที่ต้องทำการรับเวชระเบียนชิ้นที่ 1 2 3 4 และ 5 ตามลำดับ โหนด d_1 d_2 d_3 d_4 และ d_5 หมายถึงโหนดที่ต้องทำการส่งเวชระเบียนชิ้นที่ 1 2 3 4 และ 5 ตามลำดับ โดยที่โหนด 0 p_3 p_4 และ p_5 อยู่ที่ตำแหน่งเวชระเบียน โหนด p_2 d_3 และ d_4 อยู่ที่ตำแหน่งหน่วยตรวจโรคที่ 1 โหนด p_1 และ d_2 อยู่ที่ตำแหน่งหน่วยตรวจโรคที่ 2 และโหนด d_1 และ d_5 อยู่ที่หน่วยตรวจโรคที่ 3

จากกราฟ G'' นี้ จะทำการหาเส้นทางการขนส่งโดยเริ่มต้นแล้วสิ้นสุดที่โหนด 0 และทุกโหนด $i \in N''$ จะถูกเยี่ยมโหนดละ 1 ครั้ง ซึ่งเส้นทางที่ได้นี้ก็คือการจัดเรียงลำดับการให้บริการรับและส่งเวชระเบียนเพื่อให้ส่งทันถึงปลายทางภายในระยะเวลารับประกัน

เมื่อพิจารณาขนาดของปัญหาจะเห็นว่า จำนวนโหนดของกราฟ G'' ขึ้นอยู่กับจำนวนเวชระเบียนที่ต้องทำการขนส่ง โดยจำนวนโหนดเท่ากับสองเท่าของจำนวนเวชระเบียนที่ต้องทำการขนส่ง (ไม่รวมโหนด 0) จากนี้ต่อไปจะขอเรียกขนาดของปัญหาโดยใช้จำนวนเวชระเบียนที่ต้องทำการขนส่งแทนจำนวนโหนดภายในกราฟ สำหรับตัวอย่างข้างต้นเรียกได้ว่าเป็นปัญหาที่มีเวชระเบียนที่ต้องทำการขนส่ง 5 ชิ้น

3.4 แบบจำลองทางคณิตศาสตร์ของปัญหางานวิจัย

จากกราฟบริบูรณ์ $G'' = (N_0'', E'')$ เราสามารถนำไปสร้างแบบจำลองทางคณิตศาสตร์ของปัญหางานวิจัยให้อยู่ในรูปของการโปรแกรมเชิงจำนวนเต็มแบบผสม (Mixed Integer Programming) ได้ดังนี้

เซต (Sets)

R = $\{1, 2, ..., \bar{r}\}$ คือ เซตของเวชระเบียน โดยที่ $\bar{r}$ คือจำนวนเวชระเบียนที่ต้องทำการขนส่ง

P = $\{p_1, p_2, \ldots, p_{\bar{r}}\}$ คือ เซตของโหนดที่ต้องทำการรับเวชระเบียนชิ้นที่ $r \in R$

D = $\{d_1, d_2, \ldots, d_{\bar{r}}\}$ คือ เซตของโหนดที่ต้องทำการส่งเวชระเบียนชิ้นที่ $r \in R$

N'' = $P \cup D$

N_0'' = $\{0\} \cup N''$ โดยที่ 0 หมายถึงที่จอดรถ

$Pairs = \{(p_1, d_1), (p_2, d_2), \ldots, (p_{\bar{r}}, d_{\bar{r}})\}$

$K = \{1, 2, \ldots, \bar{k}\}$ คือ เซตของรถเข็น โดยที่ $\bar{k}$ คือจำนวนรถเข็นที่ใช้สำหรับขนส่งเวชระเบียน

ตัวแปร (Variables)

$$x_{ij}^k = \begin{cases} 1 & \text{ถ้ารถคันที่ } k \in K \text{ เดินทางจากโหนด } i \in N_0'' \text{ ไปยังโหนด } j \in N_0'' \\ 0 & \text{อื่น ๆ} \end{cases}$$

tbs_i คือ เวลาเริ่มให้บริการที่โหนด $i \in N''$

beg^k คือ เวลาที่รถเข็นคันที่ $k \in K$ เริ่มต้นให้บริการการขนส่ง

end^k คือ เวลาที่รถเข็นคันที่ $k \in K$ สิ้นสุดการให้บริการการขนส่ง

lv_i คือ โหลดเวชระเบียนรวมบนรถเข็นหลังจากให้บริการที่โหนด $i \in N''$ แล้ว

y คือ จำนวนรถเข็นที่ใช้ในการขนส่ง

พารามิเตอร์ (Parameters)

$dist_{ij}$ คือ ระยะทางจากโหนด $i \in N_0''$ ไปยังโหนด $j \in N_0''$

t_{ij} คือ เวลาที่ใช้ในการเดินทาง (Travel Time) จากโหนด $i \in N_0''$ ไปโหนด $j \in N_0''$

q_j คือ ปริมาณหรือโหลดของเวชระเบียนสำหรับโหนด $j \in N''$ โดยที่

$q_j > 0$ หมายถึง ปริมาณหรือโหลดของเวชระเบียนที่ต้องไปรับ

$q_j < 0$ หมายถึง ปริมาณหรือโหลดของเวชระเบียนที่ต้องไปส่ง

rt_p คือ เวลาพร้อมให้บริการรับเวชระเบียนที่โหนด $p \in P$

ST คือ เวลาที่ใช้ในการบริการสำหรับรับหรือส่งเวชระเบียน 1 ชิ้น โดยที่ $ST > 0$

Q คือ ความจุของรถเข็น

GT คือ ระยะเวลารับประกัน (guaranteed time)

T คือ พารามิเตอร์ค่ามากที่ใหญ่กว่าเวลาเดินทางรวมของเส้นทางใด ๆ ที่เป็นไปได้

ค่า T ค่าหนึ่งที่สามารถใช้ได้คือ $T = \sum_{i \in N_0''} \sum_{j \in N_0''} t_{ij}$

C คือ พารามิเตอร์ค่ามากที่ใหญ่กว่าความจุของรถเข็น

ค่า C ค่าหนึ่งที่สามารถใช้ได้คือ $C = 2Q$

1239030 75

V คือ ค่าปรับสำหรับการใช้รถแต่ละคันในการขนส่ง ซึ่งกำหนดให้มีค่าสูงกว่าระยะทางในการเดินทางของเส้นทางใด ๆ ที่เป็นไปได้

ค่า V ค่าหนึ่งที่สามารถใช้เป็นค่าปรับได้คือ $V = \sum_{i \in N_0''} \sum_{j \in N_0''} dist_{ij}$

แบบจำลอง (Model)

minimize $TotalCost = yV + \sum_{i \in N_0''} \sum_{j \in N_0''} \sum_{k \in K} dist_{ij} x_{ij}^k \quad , i \neq j$ (1)

subject to

$$\sum_{j \in P} x_{0j}^k \leq 1 \quad , \forall k \in K \tag{2}$$

$$x_{0j}^k = 0 \quad , \forall j \in D, \forall k \in K \tag{3}$$

$$x_{i0}^k = 0 \quad , \forall i \in P, \forall k \in K \tag{4}$$

$$\sum_{i \in N_0''} \sum_{k \in K} x_{ij}^k = 1 \quad , \forall j \in N'', i \neq j \tag{5}$$

$$\sum_{j \in N_0''} \sum_{k \in K} x_{ij}^k = 1 \quad , \forall i \in N'', i \neq j \tag{6}$$

$$\sum_{l \in N_0''} x_{jl}^k = \sum_{i \in N_0''} x_{ij}^k \quad , \forall j \in N'', \forall k \in K, \ i \neq j, j \neq l \tag{7}$$

$$\sum_{q \in N_0''} x_{qp}^k = \sum_{s \in N''} x_{sd}^k \quad , \forall (p,d) \in Pairs, \forall k \in K, q \neq p, s \neq d \tag{8}$$

$$tbs_p \leq tbs_d \quad , \forall (p,d) \in Pairs \tag{9}$$

$$tbs_p \geq rt_p \quad , \forall p \in P \tag{10}$$

$$beg^k + t_{0j} - (1 - x_{0j}^k)T \leq tbs_j \quad , \forall j \in P, \forall k \in K \tag{11}$$

$$(tbs_i + ST + t_{ij}) - (1 - x_{ij}^k)T \leq tbs_j \quad , \forall i, j \in N'', \forall k \in K \tag{12}$$

$$(tbs_i + ST + t_{i0}) - (1 - x_{i0}^k)T \leq end^k \quad , \forall i \in D, \forall k \in K \tag{13}$$

$$(tbs_d - rt_p) \leq GT \quad , \forall (p,d) \in Pairs \tag{14}$$

$$lv_i \leq Q \quad , \forall i \in N'' \tag{15}$$

$$q_j - (1 - x_{0j}^k)C \leq lv_j \quad , \forall j \in P, \forall k \in K \tag{16}$$

$$(lv_i + q_j) - (1 - x_{ij}^k)C \leq lv_j \quad , \forall i, j \in N'', \forall k \in K \tag{17}$$

$$\sum_{j \in N''} \sum_{k \in K} x_{0j}^k = y \tag{18}$$

$$\sum_{j \in P} x_{0j}^{k_1} \geq \sum_{j \in P} x_{0j}^{k_2} \quad , \forall k_1, k_2 \in K, k_2 - k_1 = 1 \tag{19}$$

$$x_{ij}^{k} \in \{0,1\} \quad (20)$$

$$tbs_i \geq 0 \quad (21)$$

$$lv_i \geq 0 \quad (22)$$

$$y \text{ integer} \quad (23)$$

ฟังก์ชันวัตถุประสงค์ (1) มีวัตถุประสงค์เพื่อลดจำนวนรถเข็นที่ใช้ในการขนส่งและลดระยะทางในการเดินทางรวมให้ต่ำที่สุด เงื่อนไขบังคับ (2)-(9) ใช้สำหรับบังคับเส้นทางการเดินทางของรถ โดยที่เงื่อนไขบังคับ (2) ใช้สำหรับบังคับให้รถเข็นแต่ละคัน สามารถได้รับมอบหมายให้ขนส่งเวชระเบียนได้ไม่เกิน 1 เส้นทางหรืออาจไม่ได้รับการมอบหมายให้ทำการขนส่งก็ได้ เมื่อรถเข็นได้รับมอบหมายให้ทำการขนส่งเวชระเบียนแล้ว เริ่มต้นรถเข็นจะต้องทำการรับ (pick up) เวชระเบียนเท่านั้น เงื่อนไขบังคับ (3) ใช้สำหรับบังคับให้รถเข็นไม่สามารถเริ่มต้นเส้นทางการขนส่งด้วยการส่งเวชระเบียนได้ เงื่อนไขบังคับ (4) ใช้สำหรับบังคับให้รถเข็นไม่สามารถเสร็จสิ้นการขนส่งด้วยการรับเวชระเบียนได้ เงื่อนไขบังคับ (5) ใช้สำหรับบังคับให้รถเข็นสามารถเดินทางเข้าแต่ละโหนดได้เพียง 1 ครั้งเท่านั้น เงื่อนไขบังคับ (6) ใช้สำหรับบังคับให้รถเข็นสามารถเดินทางออกจากแต่ละโหนดได้เพียง 1 ครั้งเท่านั้น เงื่อนไขบังคับ (7) ใช้บังคับให้รถเข็นที่เดินทางเข้าโหนดใด ๆ ต้องเดินทางออกจากโหนดนั้นด้วย เงื่อนไขบังคับ (8) บังคับให้รถเข็นคันเดียวกันให้บริการทั้งรับและส่งเวชระเบียนชิ้นเดียวกัน เงื่อนไขบังคับ (9) บังคับลำดับการเดินทางของรถเข็นโดยใช้เวลาเป็นตัวกำหนด นั่นคือรถเข็นต้องเดินทางไปยังโหนดสำหรับรับเวชระเบียนก่อนที่จะไปยังโหนดสำหรับส่งเวชระเบียนแต่ละชิ้นเสมอ ดังนั้นเวลาที่เริ่มให้บริการรับเวชระเบียนจะต้องน้อยกว่าเวลาที่ให้บริการส่งเวชระเบียนเสมอ สำหรับเงื่อนไขบังคับ (10) บังคับให้เวลาเริ่มต้นบริการรับเวชระเบียนจะเริ่มต้นได้ก็ต่อเมื่อหลังจากที่เวชระเบียนพร้อมรับบริการแล้ว เงื่อนไขบังคับ (11) ใช้บังคับให้เวลาที่รถเข็นแต่ละคันเริ่มต้นบริการรับเวชระเบียนชิ้นแรก ต้องมากกว่าหรือเท่ากับเวลาที่รถเข็นคันนั้นเริ่มต้นการขนส่ง เงื่อนไขบังคับ (12) คือการกำหนดเวลาที่รถเข็นเริ่มให้บริการสำหรับแต่ละโหนด ต้องมีค่าน้อยกว่าเวลาที่รถเข็นคันนั้นเริ่มให้บริการสำหรับโหนดที่ถูกบริการทีหลังเสมอภายในเส้นทางเดียวกัน เนื่องจาก $ST > 0$ เงื่อนไขบังคับนี้จะช่วยป้องกันการเกิด subtours ด้วย เงื่อนไขบังคับ (13) ใช้บังคับให้เวลาที่เสร็จสิ้นการขนส่งของรถเข็นแต่ละคัน ต้องมากกว่าหรือเท่ากับเวลาที่ให้บริการส่งเวชระเบียนชิ้นสุดท้าย เงื่อนไขบังคับ (14) ใช้สำหรับบังคับให้รถเข็นส่งเวชระเบียนถึงปลายทางโดยไม่เกินระยะเวลารับประกัน เงื่อนไขบังคับ (15) ใช้สำหรับบังคับให้โหลดรวมบนรถเข็นหลังจากไปรับหรือส่งเวชระเบียนชิ้นใด ๆ แล้ว ต้องไม่เกินความจุของรถ เงื่อนไขบังคับ (16)-(17) คำนวณโหลด

รวมบนรถเข็นหลังจากไปรับหรือส่งเวชระเบียนชิ้นใด ๆ เงื่อนไขบังคับ (18) ใช้สำหรับนับจำนวนรถเข็นทั้งหมดที่ใช้ในการขนส่ง เงื่อนไขบังคับ (19) เป็นเงื่อนไขเพิ่มเติมเพื่อช่วยลดเวลาในการค้นหาคำตอบที่เหมาะสมที่สุด โดยกำหนดให้ใช้รถเข็นในการขนส่งอย่างเป็นลำดับ นั่นหมายความว่า รถเข็นคันที่ 2 จะไม่สามารถใช้ในการขนส่งได้ ถ้าหากรถเข็นคันที่ 1 ไม่ได้ถูกใช้สำหรับขนส่ง

จากแบบจำลองทางคณิตศาสตร์ เราสามารถคำนวณหาจำนวนตัวแปรของปัญหาได้ดังนี้ สำหรับปัญหาทดสอบที่มีเวชระเบียนที่ต้องทำการขนส่ง $\bar{r}$ ชิ้น โดยกำหนดให้รถเข็นที่มีอยู่ในปัญหาทดสอบมีจำนวน $\bar{k}$ คัน จะได้ว่าจำนวนตัวแปรของปัญหาทดสอบ คำนวณได้ดังนี้

จำนวนตัวแปรของปัญหาทดสอบ $= (2\bar{r}+1)^2 \cdot \bar{k} + 4\bar{r} + 2\bar{k} + 1$

3.5 ฮิวริสติกสำหรับแก้ปัญหา

เนื่องจากการแก้ปัญหาผ่านแบบจำลองทางคณิตศาสตร์นั้น ใช้เวลานานและประยุกต์ใช้ได้สำหรับปัญหาที่มีขนาดเล็กเท่านั้น ดังนั้นฮิวริสติกจึงถูกนำมาใช้สำหรับแก้ปัญหาที่มีขนาดใหญ่ ซึ่งใช้เวลาคำนวณน้อยกว่ามาก ฮิวริสติกที่ผู้วิจัยนำเสนอสำหรับแก้ปัญหามีรายละเอียดดังนี้

3.5.1 ตัวแทนคำตอบของปัญหา

คำตอบของปัญหาวิจัยคือการจัดเรียงลำดับการให้บริการรับและส่งเวชระเบียน ซึ่งจะถูกแทนด้วยอาร์เรย์ 2 มิติ ระบุถึงลำดับของการให้บริการในการขนส่งเวชระเบียนแต่ละเส้นทาง โดยใช้รถเข็นหนึ่งคันในการขนส่งหนึ่งเส้นทาง ลักษณะของคำตอบเป็นดังนี้

$$\mathbf{Route}[k][s] = l$$

โดยที่ k คือหมายเลขรถเข็น $k = 1, 2, \ldots$

s คือลำดับการให้บริการขนส่งของรถเข็น $s = 0, 1, 2, \ldots$

l คือการบริการรับหรือส่งเวชระเบียน

$l = 0$ หมายถึงรถเข็นทำการเริ่มต้นให้บริการหรือสิ้นสุดการให้บริการซึ่งรถเข็นจะจอดอยู่ที่หน่วยเวชระเบียน

$l = Px$ หมายถึงรถเข็นให้บริการรับเวชระเบียนชิ้นที่ x

$l = Dx$ หมายถึงรถเข็นให้บริการส่งเวชระเบียนชิ้นที่ x

ตัวอย่างคำตอบของปัญหาเป็นดังรูปที่ 3.4

k \ *s*	0	1	2	3	4	5	6	7
1	0	P1	P2	D2	D1	P5	D5	0
2	0	P3	D3	P4	D4	0		

รูปที่ 3.4 อาร์เรย์ 2 มิติ แสดงคำตอบของปัญหา

จากรูปที่ 3.4 แสดงลักษณะคำตอบของปัญหา โดยรถเข็นคันที่ 1 เริ่มต้นอยู่ที่หน่วยเวชระเบียน จากนั้นจึงเริ่มให้บริการอย่างแรกคือทำการเก็บเวชระเบียนชิ้นที่ 1 จากนั้นเก็บเวชระเบียนชิ้นที่ 2 ส่งเวชระเบียนชิ้นที่ 2 และชิ้นที่ 1 แล้วจึงเก็บและส่งเวชระเบียนชิ้นที่ 5 ตามลำดับ สำหรับรถเข็นคันที่ 2 เริ่มต้นให้บริการเก็บเวชระเบียนชิ้นที่ 3 แล้วนำไปส่ง จากนั้นจึงไปเก็บเวชระเบียนชิ้นที่ 4 แล้วจึงนำไปส่งตามลำดับ

ต่อไปนี้จะขอเขียนคำตอบของปัญหาให้อยู่ในรูปที่แสดงถึงเส้นทางการขนส่งที่เข้าใจง่ายขึ้น ดังรูปที่ 3.5

R1 0 – P1 – P2 – D2 – D1 – P5 – D5 – 0

R2 0 – P3 – D3 – P4 – D4 – 0

รูปที่ 3.5 เส้นทางการขนส่ง

จากรูปที่ 3.5 *R1* และ *R2* หมายถึงเส้นทางการขนส่งของรถเข็นคันที่ 1 และ 2 ตามลำดับ

3.5.2 ฮิวริสติก

ฮิวริสติกที่นำเสนอนั้น เริ่มต้นจะทำการสร้างคำตอบเริ่มต้น (Initial Solution) ก่อน จากนั้นจึงพยายามปรับปรุงคำตอบไปเรื่อย ๆ จนกระทั่งได้ตามเกณฑ์ที่กำหนดไว้เพื่อหยุดการปรับปรุงคำตอบ ซึ่งกระบวนการปรับปรุงคำตอบนั้น ใช้ทั้ง Intensification Strategy และ Diversification Strategy ฮิวริสติกที่นำเสนอเป็นดังรูปที่ 3.6

```
Procedure Heuristic
Begin
    S ← Generate_InitialSolution()
    Set parameter : α, β, L, MaxIT
    Set TabuList to be empty
    S ← LocalSearch(S, α, β)
    S_b ← S

    Iteration ← 1
    While Iteration <= MaxIT Do
        S' ← ChooseBestofNeighborSolution(N(S) – TabuList, α, β)
        Update(S',TabuList)
        S' ← RearrangewithinRoute_MinimalWaitingTime(S')
        S' ← LocalSearch(S', α, β)

        If Cost(S') < Cost(S_b) Then
            S_b ← S'
        End If

        S ← S'
        Iteration ← Iteration + 1
    End While

    Return S_b
End
```

รูปที่ 3.6 ฮิวริสติกสำหรับแก้ปัญหา

ฮิวริสติกที่นำเสนอสำหรับใช้ในการค้นหาคำตอบนั้นเป็นเมตาฮิวริสติก (Metaheuristic) ประเภทหนึ่งที่เรียกว่าตาบูเสริช (Tabu Search) กระบวนการทำงานของฮิวริสติกเริ่มต้นด้วยการใช้ฟังก์ชัน *Generate_InitialSolution*() เพื่อสร้างคำตอบ S ซึ่งใช้เป็นคำตอบเริ่มต้นของฮิวริสติก สำหรับวิธีการสร้างคำตอบเริ่มต้นผู้วิจัยได้ดัดแปลงมาจากฮิวริสติกการแทรกแบบที่หนึ่ง(Insertion Heuristic I) ของโซโลมอน (Solomon, 1987) หลังจากได้คำตอบเริ่มต้นแล้ว จึงปรับปรุงคำตอบเริ่มต้นก่อนโดยใช้ฟังก์ชัน *LocalSearch*() ซึ่งเป็นฟังก์ชันที่ปรับปรุงคำตอบให้ดีขึ้นกว่าคำตอบเดิม วิธีการนี้เป็นกลยุทธ์ในการค้นหาคำตอบที่เรียกว่า Intensification Strategy คำตอบที่ได้หลังจากเรียกใช้ฟังก์ชัน *LocalSearch*() จะถูกกำหนดให้เป็นคำตอบที่ดีที่สุด S_b (Best Solution) และกำหนดให้เป็นคำตอบ S เป็นคำตอบปัจจุบัน (Current Solution)

จากนั้นจะเข้าสู่กระบวนการหาคำตอบโดยทำการวนซ้ำ เพื่อปรับปรุงคำตอบปัจจุบัน S ต่อไปจนถึงเกณฑ์ที่กำหนดไว้ ภายใต้การปรับปรุงโดยการวนซ้ำนั้นจะเริ่มต้นด้วยการเรียกใช้ฟังก์ชัน *ChooseBestofNeighborSolution*() ซึ่งภายในฟังก์ชันนี้จะทำการสร้างคำตอบข้างเคียงของคำตอบปัจจุบัน S ขึ้นมาชุดหนึ่ง ($\mathcal{N}(S)$) แล้วเลือกคำตอบข้างเคียงที่ดีที่สุดแต่

ไม่อยู่ในตาบูลิสท์ (Tabu list) มาเป็นคำตอบปัจจุบันตัวใหม่ S' ต่อไป โดยวิธีการนี้เป็นกลยุทธ์ในการค้นหาคำตอบที่เรียกว่า Diversification Strategy ซึ่งคำตอบที่เลือกมาเป็นคำตอบปัจจุบันตัวใหม่ S' นั้น อาจไม่ใช่คำตอบที่ดีนัก แต่ที่เลือกมานั้นเพื่อหวังที่จะหลุดจาก Local Optima แล้วไปเจอ Global Optima ซึ่งเป็นหลักการของเมตาฮิวริสติก คำตอบปัจจุบันตัวใหม่ S' ที่เลือกมานั้นจะถูกบันทึกใส่ลงในตาบูลิสท์ เพื่อห้ามไม่ให้ถูกเลือกมาเป็นคำตอบปัจจุบัน S' อีก เป็นระยะเวลาเท่ากับจำนวน L รอบ ซึ่งเป็นค่าความยาวของตาบูลิสท์ที่กำหนดไว้ จากนั้นจึงเรียกใช้ฟังก์ชัน *RearrangewithinRoute_MinimalWaitingTime*() เพื่อจัดเรียงเส้นทางบางเส้นทางของคำตอบปัจจุบัน S' ใหม่ เพื่อให้ได้เส้นทางที่มีระยะเวลาการรอคอยน้อยที่สุด ซึ่งเป็นแนวความคิดที่หวังว่าจะสามารถทำให้แทรกเวชระเบียนลงในเส้นทางนี้เพิ่มอีกได้ สุดท้ายจึงเรียกใช้ฟังก์ชัน *LocalSearch*() เพื่อปรับปรุงคำตอบปัจจุบัน S' ให้ดีขึ้นกว่าเดิม จากนั้นจึงทำการเปรียบเทียบว่าคำตอบปัจจุบัน S' ดีกว่าคำตอบที่ดีที่สุด S_b หรือไม่ ถ้าหากคำตอบปัจจุบัน S' ดีกว่าคำตอบที่ดีที่สุด S_b ก็จะทำการกำหนดให้คำตอบ S' เป็นคำตอบที่ดีที่สุด S_b แทน เมื่อสิ้นสุดแต่ละรอบของการค้นหาคำตอบจะทำการเพิ่มค่าตัวแปร *Iteration* ทีละหนึ่ง

สำหรับการปรับปรุงคำตอบจะหยุดปรับปรุงเมื่อค่า *Iteration* มีค่าเกินกว่าค่า *MaxIT* ซึ่งหมายความว่าได้ทำการปรับปรุงคำตอบจนถึงรอบที่กำหนดไว้ใช้เป็นเกณฑ์สำหรับปรับปรุงคำตอบ แล้วจะได้ว่าคำตอบที่ดีที่สุด S_b ที่เก็บไว้ตัวล่าสุดจะเป็นคำตอบที่เหมาะสมของปัญหา

สำหรับรายละเอียดแต่ละส่วนของฮิวริสติก มีรายละเอียดดังนี้

3.5.2.1 ฟังก์ชัน *Generate_InitialSolution*()

ฟังก์ชันนี้จะทำการสร้างคำตอบเริ่มต้น โดยผู้วิจัยได้ประยุกต์ขั้นตอนวิธีการสร้างคำตอบเริ่มต้นมาจากฮิวริสติกการแทรกแบบที่หนึ่งของโซโลมอน (Solomon, 1987) ซึ่งเป็นฮิวริสติกเชิงสร้างที่ใช้สำหรับสร้างเส้นทางการเดินรถสำหรับปัญหาการจัดเส้นทางเดินรถภายในกรอบเวลา (VRPTW) ฮิวริสติกนี้ถือได้ว่าเป็นฮิวริสติกที่มีประสิทธิภาพมาก ได้รับการยอมรับและถูกนำไปประยุกต์ใช้อย่างกว้างขวาง

เนื่องจากฮิวริสติกการแทรกแบบที่หนึ่งของโซโลมอนนั้น เป็นฮิวริสติกสำหรับแก้ปัญหาการจัดเส้นทางเดินรถที่รับหรือส่งสินค้าอย่างใดอย่างหนึ่งเท่านั้น

แต่งานวิจัยนี้เป็นปัญหาที่พิจารณาทั้งรับและส่งสินค้า ดังนั้นผู้วิจัยจึงได้ทำการดัดแปลงเพื่อให้สามารถประยุกต์ใช้แก้ปัญหางานวิจัยนี้ได้ แต่ยังคงแนวความคิดในการสร้างเส้นทางที่โซโลมอนได้นำเสนอไว้เพื่อให้ได้จำนวนรถและระยะทางที่เหมาะสม

ในการอธิบายวิธีการสร้างคำตอบเริ่มต้นนั้น ผู้วิจัยจะอธิบายหลักการไปพร้อมกับตัวอย่างการสร้างเส้นทาง โดยใช้ตัวอย่างในรูปที่ 3.1 และตารางที่ 3.1 ในการอธิบาย

3.5.2.1.1 เกณฑ์การเลือกให้บริการขนส่งเวชระเบียนชิ้นแรกลงในเส้นทาง

สำหรับการเลือกให้การบริการรับและส่งเวชระเบียนชิ้นใดเป็นชิ้นแรกที่จะถูกแทรกลงในเส้นทางการขนส่ง ให้เลือกเวชระเบียนชิ้นที่พร้อมได้รับการขนส่งเร็วที่สุด (Minimum Ready Time) หลังจากแทรกการบริการรับและส่งเวชระเบียนชิ้นแรกลงในเส้นทางแล้ว จะทำให้ได้เส้นทางการขนส่งขึ้นมาโดยจะขอเรียกว่าเส้นทางเริ่มต้น แสดงดังรูปที่ 3.7

$$0 - Px - Dx - 0$$

รูปที่ 3.7 เส้นทางเริ่มต้น

จากปัญหาตัวอย่าง จะได้ว่าเวชระเบียนชิ้นที่ 1 มีเวลาพร้อมได้รับการขนส่งที่เร็วที่สุดคือนาทีที่ 5 ดังนั้นเส้นทางเริ่มต้นสำหรับปัญหาตัวอย่างเป็นดังรูปที่ 3.8

$$0 - P1 - D1 - 0$$

รูปที่ 3.8 เส้นทางเริ่มต้นสำหรับปัญหาตัวอย่าง

จากเส้นทางเริ่มต้นในรูปที่ 3.8 จะได้ว่าระยะทางรวมในการเดินทางคือ 18 หน่วย

เส้นทางเริ่มต้นถือได้ว่าเป็นเส้นทางบางส่วน (Partial Route) ซึ่งเป็นเส้นทางที่ยังไม่สมบูรณ์เนื่องจากยังแทรกการบริการของเวชระเบียนไม่ครบทุกชิ้น จากนี้ต่อไปจะค่อย ๆ ทำการแทรกการบริการรับและส่งเวชระเบียนทีละชิ้นลงในเส้นทางจนได้เส้นทางที่สมบูรณ์

3.5.2.1.2 เกณฑ์การแทรกการให้บริการขนส่งเวชระเบียนลงในเส้นทางบางส่วน

หลังจากได้เส้นทางเริ่มต้นแล้ว ขั้นตอนต่อไปคือทำการแทรกเวชระเบียนที่ยังไม่ถูกแทรกลงในเส้นทาง โดยมีหลักการคำนวณดังนี้

ให้ cpr เป็นเส้นทางบางส่วนปัจจุบัน (Current Partial Route) สำหรับคู่ลำดับการบริการรับและส่งเวชระเบียนแต่ละชิ้นที่ $r \in R$ ที่ยังไม่ได้ถูกแทรกลงในเส้นทาง ให้ทำการทดลองแทรก Pr และDr นั้น ลงในเส้นทางบางส่วนปัจจุบันทุกตำแหน่งที่เป็นไปได้ จะทำให้ได้เส้นทางทั้งหมดที่สามารถแทรก Pr และ Dr ลงไปในเส้นทางบางส่วนปัจจุบันได้

กำหนดให้ FR_r เป็นเซตของเส้นทางทั้งหมดที่สามารถแทรกเวชระเบียน r ลงในเส้นทางบางส่วนปัจจุบันได้ จากนั้นให้ทำการคำนวณหาตำแหน่งแทรกที่เป็นไปได้ที่ดีที่สุด ซึ่งเป็นตำแหน่งที่แทรกแล้วทำให้ระยะทางการขนส่งเพิ่มขึ้นน้อยที่สุด (Minimum Insertion Cost, $MinIC$) โดยคำนวณ ดังนี้

$$MinIC(r) = \min_{fr \in FR_r} \{Cost(fr) - Cost(cpr)\},$$

$r \in R$ ที่ยังไม่ได้ถูกแทรกลงในเส้นทาง

หลังจากทำการคำนวณค่าการแทรกที่ดีที่สุดของเวชระเบียนแต่ละชิ้นที่ยังไม่ถูกแทรกเสร็จแล้ว เราจะได้เส้นทางที่ดีที่สุดที่เป็นไปได้ของการแทรกเวชระเบียนแต่ละชิ้นลงในเส้นทางบางส่วนปัจจุบัน กำหนดให้ fr_r^* คือเส้นทางที่ดีที่สุดของเวชระเบียนแต่ละชิ้น r หลังทำการทดลองแทรกลงไปในเส้นทางบางส่วนปัจจุบัน และจะได้ว่าเส้นทาง fr_r^* เป็นเส้นทางที่ให้ค่า $MinIC(r)$ ต่ำที่สุด

จากปัญหาตัวอย่าง จะได้ว่าขณะนี้เวชระเบียนที่ยังไม่ได้ทำการแทรกลงในเส้นทางได้แก่ เวชระเบียนชิ้นที่ 2 3 และ 4 ดังนั้นจึงให้ทดลองแทรกเวชระเบียนเหล่านี้ลงในเส้นทางบางส่วนปัจจุบันเพื่อหาตำแหน่งที่ดีที่สุดที่ทำให้ระยะทางการขนส่งเพิ่มขึ้นน้อยที่สุด การคำนวณหาเส้นทางที่ทำให้ระยะทางการขนส่งเพิ่มขึ้นน้อยที่สุดเมื่อทำการแทรกเวชระเบียนชิ้นที่ 2 ลงในเส้นทางบางส่วน ระยะทางการขนส่งที่เพิ่มขึ้นเป็นดังตารางที่ 3.3

ตารางที่ 3.3 การคำนวณหาเส้นทางที่ทำให้ระยะทางการขนส่งเพิ่มขึ้นน้อยที่สุดหลังจากทำการแทรกเวชระเบียนชิ้นที่ 2 ลงในเส้นทางบางส่วนปัจจุบัน

เส้นทางหลังทำการทดลองแทรกลงในเส้นทางบางส่วน	ระยะทางการขนส่งเพิ่มขึ้น (Insertion Cost)
0 – **P2** – **D2** – P1 – D1 – 0	Infeasible Route
0 – **P2** – P1 – **D2** – D1 – 0	Infeasible Route
0 – **P2** – P1 – D1 – **D2** – 0	Infeasible Route
0 – P1 – **P2** – **D2** – D1 – 0	Infeasible Route
0 – P1 – **P2** – D1 – **D2** – 0	22 – 18 = 4*
0 – P1 – D1 – **P2** – **D2** – 0	22 – 18 = 4

จากตารางที่ 3.3 จะได้ว่า เซตของเส้นทางทั้งหมดที่สามารถแทรกเวชระเบียนชิ้นที่ 2 ลงในเส้นทางบางส่วนปัจจุบันได้ คือ

$$FR_2 = \{0 - P1 - \mathbf{P2} - D1 - \mathbf{D2} - 0 \ , \ 0 - P1 - D1 - \mathbf{P2} - \mathbf{D2} - 0\}$$

โดยเส้นทางที่ดีที่สุดสำหรับการแทรกเวชระเบียนชิ้นที่ 2 คือ

$$fr_2^* = 0 - P1 - \mathbf{P2} - D1 - \mathbf{D2} - 0$$

ซึ่งมีระยะทางการขนส่งเพิ่มขึ้นน้อยที่สุด คือ $MinIC(2) = 4$ หน่วย

โดยหลักการเดียวกันนี้ ให้ทำการทดลองแทรกเวชระเบียนชิ้นอื่น ๆ ที่เหลือลงในเส้นทางบางส่วนปัจจุบันด้วย หลังจากคำนวณทดลองการแทรก

เวชระเบียนแต่ละชิ้น จะได้ว่าเส้นทางที่ดีที่สุดและระยะทางการขนส่งที่เพิ่มขึ้นที่น้อยที่สุด เป็นดังตารางที่ 3.4

ตารางที่ 3.4 เส้นทางที่ดีที่สุดหลังการแทรกเวชระเบียนแต่ละชิ้นลงในเส้นทางบางส่วน และระยะทางการขนส่งที่เพิ่มขึ้นที่น้อยที่สุด

เวชระเบียนชิ้นที่ถูกแทรก	เส้นทางที่ดีที่สุด fr_r^*	ระยะทางรวม $Cost(fr_r^*)$	$MinIC(r)$
2	fr_2^* = 0 – P1 – **P2** – D1 – **D2** – 0	22	$MinIC(2) = 4$
3	fr_3^* = 0 – **P3** – P1 – **D3** – D1 – 0	18	$MinIC(3) = 0$
4	fr_4^* = 0 – P1 – **P4** – **D4** – D1 – 0	20	$MinIC(4) = 2$
5	fr_5^* = 0 – P1 – **P5** – **D5** – D1 – 0	20	$MinIC(5) = 2$

จากนั้นให้ทำการคำนวณค่าประหยัด (Saving Cost) ที่เกิดจากการทดลองแทรกเวชระเบียนนั้นลงในเส้นทางบางส่วนปัจจุบัน ณ ตำแหน่งที่ดีที่สุด (ทำให้ได้เส้นทาง fr_r^*) แทนที่จะนำเวชระเบียนชิ้นนั้นไปทดลองสร้างเส้นทางการขนส่งใหม่

ให้ nr_r เป็นเส้นทางการขนส่งที่ทดลองสร้างขึ้นมาใหม่เพื่อทำการขนส่งเวชระเบียนชิ้นที่ r มีลักษณะเป็นดังรูปที่ 3.9

$$nr_r = 0 - Pr - Dr - 0$$

รูปที่ 3.9 เส้นทางการขนส่งที่ทดลองสร้างขึ้นใหม่สำหรับขนส่งเวชระเบียนชิ้นที่ r

ค่าประหยัดที่เกิดจากการแทรกเวชระเบียนชิ้นที่ r ลงในเส้นทางเดิมแทนที่จะแทรกลงในเส้นทางใหม่ คำนวณได้ดังนี้

$$SC(r) = (Cost(cpr) + Cost(nr_r)) - Cost(fr_r^*)$$

ดังนั้น เราสามารถคำนวณค่าประหยัดที่มากที่สุด (Maximum Saving Cost, *MaxSC*) ได้ดังนี้

$$MaxSC = \max_{r}\{SC(r)\},$$

$r \in R$ ที่ยังไม่ได้ถูกแทรกลงในเส้นทาง

หลังจากคำนวณค่าประหยัดเสร็จแล้ว จะได้ว่าเวชระเบียนชิ้นที่ทำให้ได้ค่าประหยัดสูงสุด จะเป็นเวชระเบียนจริงที่ถูกเลือกแทรกลงในเส้นทางบางส่วนปัจจุบัน ณ ตำแหน่งที่ดีที่สุด และกำหนดให้เส้นทางที่ได้หลังจากแทรกเวชระเบียนชิ้นนั้นแล้ว เป็นเส้นทางบางส่วนปัจจุบันถัดไป

จากปัญหาตัวอย่าง วิธีการคำนวณค่าประหยัดสูงสุดเป็นดังตารางที่ 3.5

ตารางที่ 3.5 การคำนวณค่าประหยัดของปัญหาตัวอย่าง

เวชระเบียนชิ้นที่ถูกแทรก	เส้นทางที่เกิดจากการแทรกเวชระเบียนลงในเส้นทางใหม่	เส้นทางที่เกิดจากการแทรกเวชระเบียนลงในเส้นทางเดิม	SC(*r*)
2	0 – P1 – D1 – 0 $Cost(cpr) = 18$ 0 – **P2** – **D2** – 0 $Cost(nr_2) = 12$	0 – P1 – **P2** – **D2** – D1 – 0 $Cost\left(fr_2^*\right) = 22$	8
3	0 – P1 – D1 – 0 $Cost(cpr) = 18$ 0 – **P3** – **D3** – 0 $Cost(nr_2) = 8$	0 – **P3** – P1 – **D3** – D1 – 0 $Cost\left(fr_2^*\right) = 18$	8
4	0 – P1 – D1 – 0 $Cost(cpr) = 18$ 0 – **P4** – **D4** – 0 $Cost(nr_2) = 8$	0 – P1 – **P4** – **D4** – D1 – 0 $Cost\left(fr_2^*\right) = 20$	6
5	0 – P1 – D1 – 0 $Cost(cpr) = 18$ 0 – **P5** – **D5** – 0 $Cost(nr_2) = 14$	0 – P1 – **P5** – **D5** – D1 – 0 $Cost\left(fr_2^*\right) = 20$	12*

จากตารางที่ 3.5 จะเห็นว่าเวชระเบียนที่ให้ค่าประหยัดสูงสุดคือ เวชระเบียนชิ้นที่ 5 ซึ่งให้ค่าประหยัด 12 หน่วย ดังนั้นจึงได้ว่า เวชระเบียนชิ้นที่ 5 เป็นเวชระเบียนที่เหมาะสมที่สุดที่จะได้รับการเลือกให้แทรกลงในเส้นทาง

บางส่วนปัจจุบัน ดังนั้นจะได้ว่าเส้นทางบางส่วนปัจจุบันเส้นทางใหม่ เป็นดังรูปที่ 3.10

0 – P1 – P5 – D5 – D1 – 0

รูปที่ 3.10 เส้นทางบางส่วนหลังการคำนวณรอบที่ 1

จากนั้นให้วนคำนวณซ้ำ โดยรอบการคำนวณที่ 2 จะมีเส้นทางในรูปที่ 3.10 เป็นเส้นทางบางส่วนปัจจุบัน และเวชระเบียนที่ยังไม่ได้ถูกแทรกคือเวชระเบียนชิ้นที่ 2 3 และ 4 ทำการคำนวณวนซ้ำเช่นนี้เรื่อยไปจนสามารถแทรกเวชระเบียนได้จนครบทุกชิ้น ถ้าหากรอบการคำนวณใด ไม่สามารถแทรกเวชระเบียนลงในเส้นทางบางส่วนปัจจุบันได้เลย ให้ทำการสร้างเส้นทางขึ้นมาใหม่โดยใช้เกณฑ์การเลือกให้บริการขนส่งเวชระเบียนชิ้นแรกลงในเส้นทางตามวิธีในหัวข้อ 3.5.2.1.1

หลังจากแทรกเวชระเบียนได้จนครบทุกชิ้น จะได้เส้นทางที่สมบูรณ์ซึ่งจะใช้เป็นคำตอบเริ่มต้นของปัญหา ดังรูปที่ 3.11

R1 0 – P1 – P3 – P5 – D3 – D5 – D1 – 0

R2 0 – P4 – D4 – P2 – D2 – 0

รูปที่ 3.11 คำตอบเริ่มต้นของปัญหา

3.5.2.2 ฟังก์ชัน *LocalSearch*()

ฟังก์ชันนี้เป็นการค้นหาคำตอบโดยใช้หลักการของ Intensification Strategy ซึ่งจะทำการปรับปรุงคำตอบปัจจุบันให้ดีขึ้นกว่าเดิม โดยใช้วิธีการย้ายคู่ของการบริการรับและส่งเวชระเบียนทั้ง 3 วิธีของ Li and Lim (2003) โดยจากนี้จะขอเรียกคู่ของการบริการรับและส่งเวชระเบียนว่า "PD" วิธีการย้าย PD ทั้ง 3 วิธีมีรายละเอียดดังนี้

1) PD-Shift Operator

PD-Shift Operator คือวิธีการย้าย PD ของเวชระเบียนจากเส้นทางหนึ่งไปยังอีกเส้นทางหนึ่ง โดยทำการสุ่มเส้นทางขึ้นมา 2 เส้นทาง กำหนดให้เป็นเส้นทาง *R1* และ *R2* ตามลำดับการสุ่ม จากนั้นทำการสุ่มเวชระเบียนขึ้นมา 1 ชิ้นจากเส้นทาง *R1* แล้วนำไปแทรกลงในเส้นทาง *R2* หลังจากทำการแทรกแล้วเส้นทางทั้งสองต้องยังคงเป็นเส้นทางที่เป็นไปได้ โดยอยู่ภายใต้เงื่อนไขบังคับที่ว่าเวชระเบียนทุกชิ้นต้องได้รับการขนส่งภายในระยะเวลารับประกัน และโหลดรวมของเวชระเบียนบนรถเข็นตลอดเส้นทางการขนส่งนั้น ต้องไม่เกินความจุของรถ

โดยวิธีการย้ายแบบนี้จะเป็นทางหนึ่งที่จะช่วยลดจำนวนเส้นทางลงได้ ซึ่งเป็นการช่วยลดจำนวนรถที่จะใช้ในการขนส่งลงนั่นเอง ตัวอย่างของ PD-Shift Operator เป็นดังรูปที่ 3.12

R1 0 – P1 – **P2** – **D2** – D1 – 0 ⟹ ***R1*** 0 – P1 – D1 – 0

R2 0 – P3 – D3 – P4 – D4 – 0 ***R2*** 0 – **P2** – P3 – D3 – P4 – **D2** – D4 – 0

รูปที่ 3.12 ตัวอย่างของ PD-Shift Operator : P2-D2 ถูกย้ายจาก *R1* ไปยัง *R2*

2) PD-Exchange Operator

PD-Exchange Operator คือการสลับสับเปลี่ยน PD ระหว่างสองเส้นทาง โดยทำการสุ่มเส้นทางขึ้นมา 2 เส้นทาง กำหนดให้เป็นเส้นทาง *R1* และ *R2* ตามลำดับการสุ่ม จากนั้นทำการสุ่มเวชระเบียนขึ้นมา 1 ชิ้นจากเส้นทาง *R1* และสุ่มเวชระเบียนขึ้นมาอีก 1 ชิ้นจากเส้นทาง *R2* จากนั้นทำการย้ายเวชระเบียนที่สุ่มมาออกจากเส้นทางเดิม แล้วทำการแทรกเวชระเบียนที่สุ่มมาจากเส้นทาง *R1* ลงในเส้นทาง *R2* และทำการแทรกเวชระเบียนที่สุ่มจากเส้นทาง *R2* ลงในเส้นทาง *R1* หลังจากทำการแทรกแล้วเส้นทางทั้งสองต้องยังคงเป็นเส้นทางที่เป็นไปได้ โดยอยู่ภายใต้เงื่อนไขบังคับที่ว่าเวชระเบียนทุกชิ้นต้องได้รับการขนส่งภายในระยะเวลารับประกัน และโหลดรวมของเวชระเบียนบนรถเข็นตลอดเส้นทางการขนส่งนั้น ต้องไม่เกินความจุของรถ ตัวอย่างของ PD-Exchange Operator เป็นดังรูปที่ 3.13

R1 0 – P1 – **P2** – **D2** – D1 – 0 ⟹ ***R1*** 0 – P1 – D1 – **P3** – **D3** – 0

R2 0 – **P3** – **D3** – P4 – D4 – 0 ***R2*** 0 – **P2** – P4 – **D2** – D4 – 0

รูปที่ 3.13 ตัวอย่างของ PD-Exchange Operator:
P2-D2 และ P3-D3 ถูกสลับเปลี่ยนเส้นทางกัน

3) Rearrange Operator

Rearrange Operator คือการสลับลำดับภายในเส้นทาง โดยใช้ 2 วิธีในการสลับลำดับภายในเส้นทาง ได้แก่ วิธีที่หนึ่งคือการย้าย P หรือ D ของเวชระเบียนจากตำแหน่งหนึ่งไปยังอีกตำแหน่งหนึ่ง ส่วนวิธีที่สองคือทำการสลับ 3 ลำดับที่อยู่ติดกัน โดยวิธีที่หนึ่งจะใช้เมื่อคำตอบของปัญหามีเส้นทาง 2 เส้นทางขึ้นไป ส่วนอีกวิธีหนึ่งใช้เมื่อคำตอบของปัญหาเหลือเพียง 1 เส้นทาง เส้นทางหลังจากทำการสลับที่แล้วต้องยังคงเป็นเส้นทางที่เป็นไปได้ โดยอยู่ภายใต้เงื่อนไขบังคับที่ว่าเวชระเบียนทุกชิ้นต้องได้รับการขนส่งภายในระยะเวลารับประกัน และโหลดรวมของเวชระเบียนบนรถเข็นตลอดเส้นทางการขนส่งนั้น ต้องไม่เกินความจุของรถ ตัวอย่างของ Rearrange Operator เป็นดังรูปที่ 3.14

R1 0 – P1 – **P2** – **D2** – **D1** – 0 ⟹ ***R1*** 0 – P1 – **D1** – **P2** – **D2** – 0

รูปที่ 3.14 ตัวอย่างของ Rearrange Operator:
D1-P2-D2 ถูกสลับภายในเส้นทาง

สำหรับขั้นตอนวิธีของฟังก์ชัน *LocalSearch*() เป็นดังรูปที่ 3.15

```
Function LocalSearch(S, α, β)
Begin
    Do
        Do
            Do
                Do
                    HSC = PD-ShiftOperator(S, α)
                While HSC > 0

                HSC = PD-ExchangeOperator(S, β)
            While HSC > 0

            HSC = RearrangeOperator(S)
        While HSC > 0

        HSC = TrytoReduceRoute(S)
    While HSC > 0

    Return S
End
```

รูปที่ 3.15 ฟังก์ชัน *LocalSearch*()

จากรูปที่ 3.15 แสดงขั้นตอนวิธีของฟังก์ชัน *LocalSearch*() เริ่มต้นมีคำตอบปัจจุบัน *S* เป็นส่วนนำเข้าของฟังก์ชัน ภายในฟังก์ชันนี้จะทำการปรับปรุงคำตอบ *S* ให้ดีขึ้นกว่าเดิม ซึ่งทำการปรับปรุงคำตอบโดยใช้ทั้ง 3 วิธีดังกล่าวข้างต้น

แนวความคิดที่ผู้วิจัยใช้ในการจัดเรียงลำดับ Operator เพื่อปรับปรุงเส้นทางนั้น เริ่มแรกจะใช้ PD-Shift Operator ก่อน ซึ่งถือได้ว่าเป็น Operator ที่มีประสิทธิภาพในการปรับปรุงคำตอบสูงที่สุด เนื่องจากเป็นวิธีการปรับปรุงเส้นทางเพียงวิธีเดียวที่สามารถช่วยลดจำนวนเส้นทางลงได้ นั่นหมายความว่าช่วยลดจำนวนรถเข็นที่ใช้ในการขนส่งลงได้นั่นเอง หลังจากที่ไม่สามารถใช้ PD-Shift Operator ในการปรับปรุงเส้นทางได้แล้วนั้น แสดงว่าคำตอบของปัญหาที่ได้ขณะนี้เริ่มมีเส้นทางบางเส้นทางที่มีความหนาแน่นมากขึ้น ทำให้ไม่สามารถนำเวชระเบียนไปแทรกลงในเส้นทางได้อีก จึงเข้าสู่การปรับปรุงคำตอบโดยใช้ PD-Exchange Operator ซึ่งโดยวิธีการของ Operator นี้ หลังจากที่สุ่มเวชระเบียนขึ้นมาเส้นทางละหนึ่งชิ้นจากสองเส้นทางที่จะทำการสลับกัน เวชระเบียนที่ถูกสุ่มขึ้นมานั้นจะถูกลบออกจากเส้นทางเดิมก่อนเพื่อช่วยลดความหนาแน่นภายในเส้นทาง จากนั้นถึงค่อยนำเวชระเบียนลงไป

แทรกลงในอีกเส้นทางหนึ่ง เมื่อ PD-Exchange Operator ไม่สามารถใช้ในการปรับปรุงคำตอบได้ จึงใช้ Rearrange Operator เพื่อปรับปรุงคำตอบโดยการสลับตำแหน่งภายในเส้นทาง

วิธีการคำนวณการปรับปรุงเส้นทางโดยฟังก์ชัน *LocalSearch*() เริ่มแรกคำตอบ S จะถูกปรับปรุงโดยใช้ PD-Shift Operator โดยทำการทดลองสุ่มย้าย PD จากเส้นทางหนึ่งไปลงยังอีกเส้นทางหนึ่งในตำแหน่งที่เป็นไปได้ ซึ่งจะทำการสุ่มเป็นจำนวน $\lfloor \alpha \bar{r} \rfloor$ ครั้ง แล้วคำนวณค่าประหยัด (Saving Cost, SC) ที่เกิดจากการนำ PD ไปใส่ลงยังตำแหน่งต่าง ๆ ซึ่งสามารถคำนวณค่าประหยัดได้ดังนี้

ให้ r_1 และ r_2 คือเส้นทางสองเส้นทางก่อนทำการย้าย PD
ให้ r_1' และ r_2' คือเส้นทางสองเส้นทางหลังทำการย้าย PD จาก r_1 ไป r_2
ให้ $f(r)$ คือ ค่าขนส่งของเส้นทาง r

$$SC(r_1, r_2) = (f(r_1) + f(r_2)) - (f(r_1') + f(r_2'))$$

ตำแหน่งที่ PD ของ r_1 ไปแทรกลงใน r_2 แล้วให้ค่าประหยัดที่มากที่สุด (Highest Saving Cost, HSC) ที่เป็นบวก จะถือว่าเป็นตำแหน่งที่ดีที่สุด และจะเป็นตำแหน่งจริงที่ใช้สำหรับย้าย PD ของ r_1 ไปแทรกลงใน r_2

การย้ายคำตอบโดยใช้ PD-Shift Operator นี้จะถูกทำซ้ำเมื่อให้ค่าประหยัดที่มากที่สุดที่เป็นบวกเท่านั้น ซึ่งเป็นการรับรองว่าเป็นการย้ายคำตอบไปสู่คำตอบที่ดีขึ้นเสมอ และจะหยุดย้ายคำตอบโดยใช้ PD-Shift Operator เมื่อไม่สามารถหาตำแหน่งแทรกที่ให้ค่าประหยัดที่มากที่สุดที่เป็นบวกได้

หลังจากการย้ายคำตอบโดยใช้ PD-Shift Operator ไม่สามารถให้ค่าประหยัดที่เป็นบวกได้ จะทำการเปลี่ยนวิธีการย้ายคำตอบโดยใช้ PD-Exchange Operator โดยจะทำการสลับสับเปลี่ยน PD ระหว่างสองเส้นทางเพื่อไปแทรกยังตำแหน่งที่ให้ค่าประหยัดที่เป็นบวกที่มากที่สุดเช่นกัน ซึ่งจะทำการสุ่มเป็นจำนวน $\lfloor \beta \bar{r} \rfloor$ ครั้ง โดยมีวิธีการคำนวณค่าประหยัดเช่นเดียวกับวิธี PD-Shift Operator ถ้าหากการย้ายคำตอบโดยใช้ PD-Exchange Operator สามารถให้ค่าประหยัดที่เป็นบวกได้ เราก็จะทำการย้ายคำตอบไปยังตำแหน่งที่ให้ค่าประหยัดที่มากที่สุด แล้วจากนั้นจะกลับไปเริ่มต้นทำ

การย้ายคำตอบโดยใช้ PD-Shift Operator ใหม่อีกครั้ง แต่ถ้าหากการย้ายคำตอบโดยใช้ PD-Exchange Operator ยังไม่สามารถให้ค่าประหยัดที่เป็นบวกได้ จะทำการเปลี่ยนวิธีการย้ายย้ายคำตอบโดยใช้ Rearrange Operator ซึ่งเป็นการสลับลำดับภายในเส้นทางแต่ละเส้นทางซึ่งจะทำทุกเส้นทางโดยสุ่มทำเส้นทางละหนึ่งครั้ง

การย้ายคำตอบโดยใช้ Rearrange Operator จะทำการย้ายไปยังตำแหน่งที่ให้ค่าประหยัดที่มากที่สุดที่เป็นบวก ซึ่งมีวิธีการคำนวณค่าประหยัดดังนี้

ให้ r_1 คือเส้นทางสองเส้นทางก่อนทำการสลับ

ให้ r_1' คือเส้นทางสองเส้นทางหลังทำการสลับ

$$SC(r_1) = f(r_1) - f(r_1')$$

ถ้าหากการย้ายโดยวิธี Rearrange Operator สามารถให้ค่าประหยัดที่เป็นบวกได้ ก็จะทำการย้ายไปยังตำแหน่งที่ให้ค่าบวกที่มากที่สุด แล้วจากนั้นจะกลับไปเริ่มต้นทำการย้ายคำตอบโดยใช้ PD-Shift Operator ใหม่อีกครั้ง แต่ถ้าหากการย้ายคำตอบโดยใช้ Rearrange Operator ยังไม่สามารถให้ค่าประหยัดที่เป็นบวกได้ สุดท้ายจะทดลองทำการลดเส้นทาง

การลดเส้นทางทำโดยใช้ฟังก์ชัน *TrytoReduceRoute*() ซึ่งจะพยายามลดเส้นทางลงโดยสุ่มเส้นทางขึ้นมาสองเส้นทาง กำหนดให้เป็นเส้นทาง *R1* และ *R2* ตามลำดับการสุ่ม จากนั้นทำการแทรกเวชระเบียนทุกชิ้นที่อยู่ในเส้นทาง *R1* ลงในเส้นทาง *R2* ถ้าหากสามารถลดเส้นทางลงได้ ก็จะเริ่มต้นทำการย้ายคำตอบโดยใช้ PD-Shift Operator ใหม่อีกครั้ง แต่ถ้าหากไม่สามารถลดเส้นทางลงได้ ก็จะถือว่าการปรับปรุงคำตอบให้ดีขึ้นอยู่ในสภาวะที่นิ่งเพียงพอแล้ว เนื่องจากทำการวนซ้ำเพื่อปรับปรุงคำตอบจนกว่าจะไม่มีวิธีการย้ายใดที่สามารถปรับปรุงคำตอบให้ดีขึ้นได้เลย ก็จะถือว่าคำตอบที่ปรับปรุงล่าสุดเป็นคำตอบปัจจุบันที่ดีที่สุด (Current Best Solution)

จากวิธีการย้ายคำตอบของฟังก์ชัน *LocalSearch*() จะเห็นว่าเป็นการย้ายคำตอบไปสู่คำตอบที่ดีขึ้นเสมอ ดังนั้นจึงรับประกันได้ว่าคำตอบที่ได้หลังจากทำการเรียกใช้ฟังก์ชันนี้แล้ว จะเป็นคำตอบที่ดีขึ้นกว่าเดิม

3.5.2.3 ฟังก์ชัน *ChooseBestofNeighborSolution*()

ในฟังก์ชันนี้จะทำการสร้างคำตอบข้างเคียงจากคำตอบปัจจุบัน S โดยจะสร้างคำตอบข้างเคียงโดยใช้วิธีการย้าย 2 วิธีด้วยกัน ได้แก่ PD-Shift Operator และ PD-Exchange Operator คำตอบข้างเคียงที่สร้างขึ้นมาทั้งหมดนั้นต้องเป็นคำตอบที่เป็นไปได้ (Feasible Solution) จากนั้นจะทำการเลือกคำตอบที่ดีที่สุดจากคำตอบข้างเคียงที่สร้างขึ้นมาทั้งหมดและไม่ได้อยู่ในตาบูลิสท์ (Tabu List) ขึ้นมาเป็นคำตอบปัจจุบัน S' ตัวต่อไป

คำตอบ S' ซึ่งถูกเลือกโดยวิธีการที่กล่าวมาข้างต้นนั้น อาจเป็นคำตอบที่ไม่ดีนัก เพราะว่าคำตอบที่ดีกว่า S' อาจถูกห้ามไว้ในตาบูลิสท์ จึงทำให้คำตอบที่ดีรองลงมาแต่ไม่ถูกห้าม ถูกเลือกมาเป็นคำตอบ S' แทน วิธีการเช่นนี้เป็นหลักการของเมตาฮิวริสติก นั่นคือการยอมรับคำตอบที่แย่กว่า เพื่อหวังจะหลุดจาก Local Optima แล้วไปเจอ Global Optima

การค้นหาคำตอบด้วยหลักการของตาบูเสริชเป็นวิธีหนึ่งของเมตาฮิวริสติก โดยใช้ตาบูลิสต์เป็นเครื่องมือช่วยในการค้นหาคำตอบ ตาบูลิสท์นั้นเปรียบเสมือนเป็น หน่วยความจำระยะสั้น (Short Term Memory) ซึ่งเป็นเครื่องมือช่วยจำว่าคำตอบใดเคยถูกเลือกเป็นคำตอบ S' มาบ้างแล้ว หลักการของตาบูเสริชคือคำตอบ S' ที่เคยถูกเลือกมานั้นจะถูกห้ามไว้ในตาบูลิสท์เป็นระยะเวลาเท่ากับจำนวน L รอบ ซึ่งเป็นค่าความยาวของตาบูลิสท์เพื่อไม่ให้ถูกเลือกขึ้นมาอีก ดังนั้นคำตอบ S' ที่ถูกเลือกจะต้องไม่ใช่คำตอบที่ถูกเลือกมาแล้วซึ่งถูกบันทึกเอาไว้อยู่ในตาบูลิสท์ จึงอาจทำให้คำตอบ S' ที่ถูกเลือกนั้นมีค่าแย่กว่า

สำหรับการบันทึกคำตอบ S' ที่ถูกเลือกมาลงในตาบูลิสท์ จะทำการคำนึงถึงวิธีการที่จะใช้บันทึกคำตอบลงในตาบูลิสท์ด้วย เนื่องจากถ้าหากบันทึกคำตอบทั้งคำตอบลงไปในตาบูลิสท์ จะเห็นว่าถึงแม้จะมีส่วนดีคือคำตอบที่บันทึกนั้นเป็นตัวแทนของคำตอบนั้นจริง ๆ แต่มีข้อเสียคือใช้พื้นที่เยอะในการบันทึกและใช้เวลามากในการตรวจสอบคำตอบในตาบูลิสท์ ทำให้ไม่มีประสิทธิภาพ แต่ถ้าหากเลือกบันทึกบางคุณสมบัติ (Attribute) ของคำตอบเท่านั้น เช่นบันทึกเฉพาะระยะทางรวมของเส้นทาง จะเห็นว่าใช้พื้นที่ในการบันทึกเพียงเล็กน้อย ตรวจสอบคำตอบได้ง่าย แต่มีข้อเสียคือมีคำตอบหลายคำตอบที่อาจให้ค่าระยะทางรวมเท่ากัน ดังนั้นการ

บันทึกเพียงระยะทางรวมลงในตาบูลิสท์นั้น อาจทำให้คำตอบที่จะนำไปสู่ Global Optima ถูกห้ามไปด้วย ดังนั้นการเลือกว่าควรจะใช้คุณสมบัติใดของคำตอบบันทึกลงไปในตาบูลิสท์จึงเป็นสิ่งที่สำคัญอย่างหนึ่งเช่นกัน

สำหรับการเลือกคุณสมบัติที่ใช้เป็นตัวแทนของคำตอบ เพื่อบันทึกลงในตาบูลิสท์นั้น ผู้วิจัยเลือกใช้คุณสมบัติของคำตอบ 3 คุณสมบัติเหล่านี้ร่วมกัน ได้แก่

1. จำนวนรถที่ใช้ (The used vehicles)
2. ระยะทางรวม (Total distance)
3. ระยะเวลาคอยรวม (Total waiting time)

สำหรับระยะเวลาคอยคำนวณจากเวลาที่รถใช้ในการคอยเพื่อรอรับสินค้า คำนวณได้ดังนี้

ให้ $RT(p_r)$ คือเวลาที่เวชระเบียนชิ้นที่ r พร้อมได้รับการขนส่ง

ให้ $AT(p_r)$ คือเวลาที่รถมาถึงโหนดเพื่อรับเวชระเบียนชิ้นที่ r

$$WaitingTime(p_r) = \begin{cases} 0 & , \text{ if } AT(p_r) \geq RT(p_r) \\ RT(p_r) - AT(p_r) & , \text{ if } AT(p_r) < RT(p_r) \end{cases}$$

ดังนั้นระยะเวลาคอยรวม (Total waiting time) คำนวณได้จาก

$$TotalWaitingTime = \sum_{p_r \in P} WaitingTime(p_r)$$

ลักษณะของตัวอย่างตาบูลิสท์ เป็นดังรูปที่ 3.16

	จำนวนรถ	ระยะทางรวม	ระยะเวลาคอยรวม
1	2	175.54	35.24
2	2	168.69	30.54
L			

รูปที่ 3.16 ตัวอย่างตาบูลิสท์

ด้วยวิธีการใช้คุณสมบัติทั้ง 3 คุณสมบัตินี้แทนคำตอบ จะเห็นว่าความน่าจะเป็นที่คำตอบ 2 คำตอบใด ๆ ที่แตกต่างกัน จะมีคุณสมบัติทั้ง 3 คุณสมบัติเหมือนกันนั้น จะมีค่าน้อยมาก (Li and Lim, 2003)

ตาบูลิสท์จะต้องมีการกำหนดความยาวของรายการไว้ด้วยความยาว L ซึ่งนั่นหมายถึงคำตอบที่ถูกบันทึกลงในตาบูลิสท์ จะถูกห้ามเลือกมาเป็นคำตอบ S' เป็นจำนวน L รอบ

หลังจากเลือกคำตอบข้างเคียงที่ดีที่สุดที่ไม่ได้อยู่ในตาบูลิสท์มาเป็นคำตอบ S' แล้ว จากนั้นคำตอบ S' นี้จะถูกบันทึกลงในตาบูลิสท์ด้วยคุณสมบัติทั้ง 3 คุณสมบัติข้างต้น เพื่อไม่ให้คำตอบ S' นี้ถูกเลือกมาเป็นคำตอบ S' อีกในรอบของการค้นหาคำตอบรอบถัดไป หลักการดังกล่าวนี้เรียกว่า Diversification Strategy

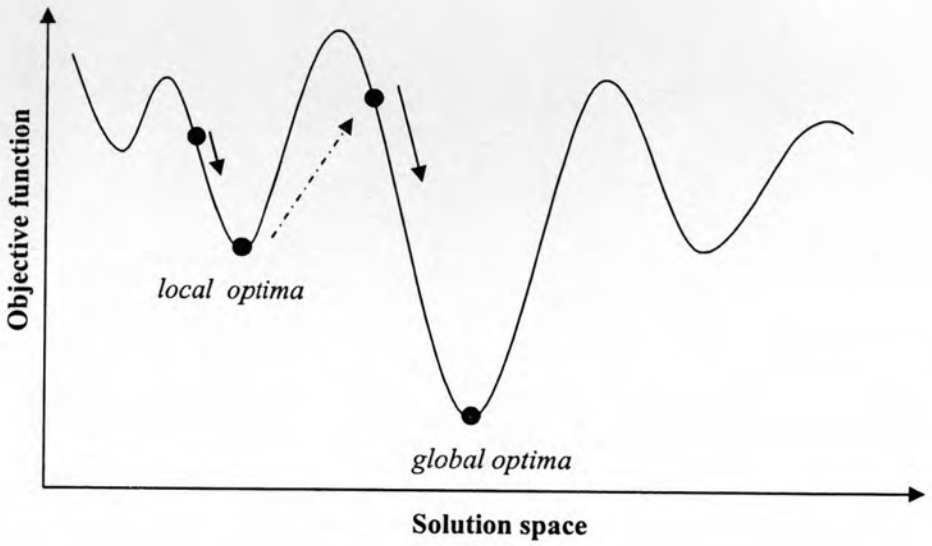
3.5.2.4 ฟังก์ชัน *RearrangewithinRoute_MinimalWaitingTime*()

ในฟังก์ชันนี้จะถูกเรียกใช้หลังจากที่ได้คำตอบ S' จากฟังก์ชัน *ChooseBestofNeighborSolution*() แล้ว ซึ่งจะทำการจัดเรียงเส้นทางของคำตอบ S' ใหม่ โดยทำการเรียงเฉพาะเส้นทางสองเส้นทางที่ถูกสลับตำแหล่ง PD มาจากฟังก์ชันข้างต้นให้มีเวลาคอยน้อยที่สุด ด้วยวิธีการเรียงเส้นทางเช่นนี้ อาจทำให้คำตอบแย่ขึ้นได้ (ระยะทางรวมมากขึ้น) แต่อาจจะทำให้สามารถนำ PD จากเส้นทางอื่นมาแทรกลงในเส้นทางเหล่านี้ได้ ซึ่งเป็นหลักการของ Diversification Strategy เช่นกัน

3.5.2.5 เกณฑ์การหยุดการค้นหาคำตอบ

สำหรับการปรับปรุงคำตอบจะหยุดปรับปรุงเมื่อค่า *Iteration* มีค่าเกินกว่าค่า *MaxIT* ซึ่งหมายความว่าได้ทำการปรับปรุงคำตอบจนถึงรอบที่กำหนดไว้ใช้เป็นเกณฑ์สำหรับปรับปรุงคำตอบ แล้วจะได้ว่าคำตอบที่ดีที่สุด S_b ที่เก็บไว้ตัวล่าสุดจะเป็นคำตอบที่เหมาะสมของปัญหา

สำหรับวิธีการค้นหาคำตอบของฮิวริสติกที่นำเสนอ ใช้หลักการการค้นหาคำตอบทั้ง Diversification Strategy และ Intensification Strategy ซึ่งสามารถจำลองได้ดังรูปที่ 3.17



รูปที่ 3.17 การค้นหาคำตอบโดยใช้หลักการ Diversification และ Intensification Strategy

จากรูปที่ 3.17 แสดงให้เห็นถึงการค้นหาคำตอบ ซึ่งบ่อยครั้งที่เจอคำตอบที่เป็น Local Optima แต่ถ้าหากเรายอมที่จะย้ายไปยังคำตอบที่แย่กว่าเดิม ก็อาจจะเป็นคำตอบที่นำไปสู่คำตอบที่เป็น Global Optima ได้

3.6 การสร้างเส้นทางการขนส่งโดยละเอียด

สำหรับการสร้างเส้นทางการขนส่งโดยละเอียด ผู้วิจัยใช้ปัญหาตัวอย่างข้างต้นเป็นตัวอย่างในการอธิบาย โดยผลการจัดเส้นทางการขนส่งสำหรับปัญหาตัวอย่างโดยใช้ฮิวริสติกที่นำเสนอเป็นดังรูปที่ 3.18

R1 0 – P1 – P3 – P5 – D3 – D5 – D1 – 0

R2 0 – P4 – D4 – P2 – D2 – 0

รูปที่ 3.18 เส้นทางการขนส่งเวชระเบียนสำหรับปัญหาตัวอย่าง

จากรูปที่ 3.18 คือคำตอบที่ได้จากฮิวริสติกแสดงเส้นทางการขนส่งเวชระเบียน โดยที่เส้นทางแต่ละเส้นทางนั้น จะต้องใช้รถเข็นหนึ่งคันเดินทางเพื่อขนส่งเวชระเบียน ดังนั้นจึงได้ว่าคำตอบของปัญหาตัวอย่างนี้ต้องใช้รถเข็น 2 คันในการขนส่งเวชระเบียนทุกชิ้นให้ส่งทันภายในระยะเวลารับประกัน

จากเส้นทางการขนส่งที่ได้คำนวณหาได้จากฮิวริสติกที่นำเสนอ แสดงถึงลำดับการรับและส่งเวชระเบียนโดยยังไม่ได้มีการระบุหน่วยที่ต้องเดินทางไปให้บริการ แต่เนื่องจากเราทราบจากข้อมูลนำเข้าอยู่แล้วว่าการรับและส่งเวชระเบียนแต่ละชิ้นนั้น มีต้นทางและปลายทางอยู่ที่หน่วยใดบ้าง ดังนั้นเราจึงสามารถสรุปเป็นเส้นทางระบุลำดับของหน่วยที่ต้องเดินทางไปพร้อมด้วยลำดับการขนส่งที่เกิดขึ้น ณ แต่ละหน่วยได้ดังรูปที่ 3.19

R1 C0 – C2(P1) – C0(P3,P5) – C1(D3) – C3(D5,D1) – C0

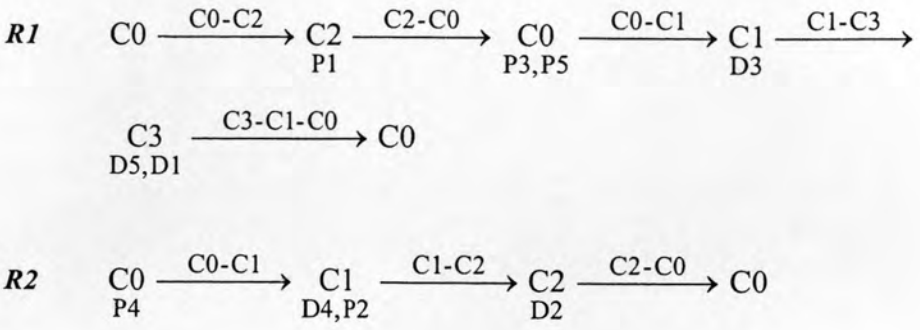
R2 C0(P4) – C1(D4,P2) – C2(D2) – C0

รูปที่ 3.19 เส้นทางระบุลำดับของหน่วยที่ต้องเดินทางไปพร้อมด้วยลำดับการขนส่งที่เกิดขึ้นแต่ละหน่วย

จากรูปที่ 3.19 ให้รถเข็นคันที่ 1 วิ่งตามเส้นทาง *R1* เริ่มต้นอยู่ที่หน่วยเวชระเบียน จากนั้นเดินทางไปยังหน่วยตรวจโรคที่ 2 เพื่อรับเวชระเบียนชิ้นที่ 1 แล้วเดินทางกลับไปที่หน่วยเวชระเบียนอีกครั้งเพื่อรับเวชระเบียนชิ้นที่ 3 และ 5 ตามลำดับ จากนั้นเดินทางไปยังหน่วยตรวจโรคที่ 1 เพื่อส่งเวชระเบียนชิ้นที่ 3 แล้วจึงเดินทางไปยังหน่วยเวชระเบียนที่ 3 เพื่อส่งเวชระเบียนชิ้นที่ 5 และ 1 ตามลำดับ สำหรับรถเข็นคันที่ 2 วิ่งตามเส้นทาง *R2* เริ่มต้นอยู่ที่หน่วยเวชระเบียน รอรับเวชระเบียนชิ้นที่ 4 แล้วเดินทางไปหน่วยตรวจโรคที่ 1 เพื่อส่งเวชระเบียนชิ้นที่ 4 และรับเวชระเบียนชิ้นที่ 2 จากนั้นจึงเดินทางไปที่หน่วยตรวจโรคที่ 2 เพื่อส่งเวชระเบียนชิ้นที่ 2 ตามลำดับ

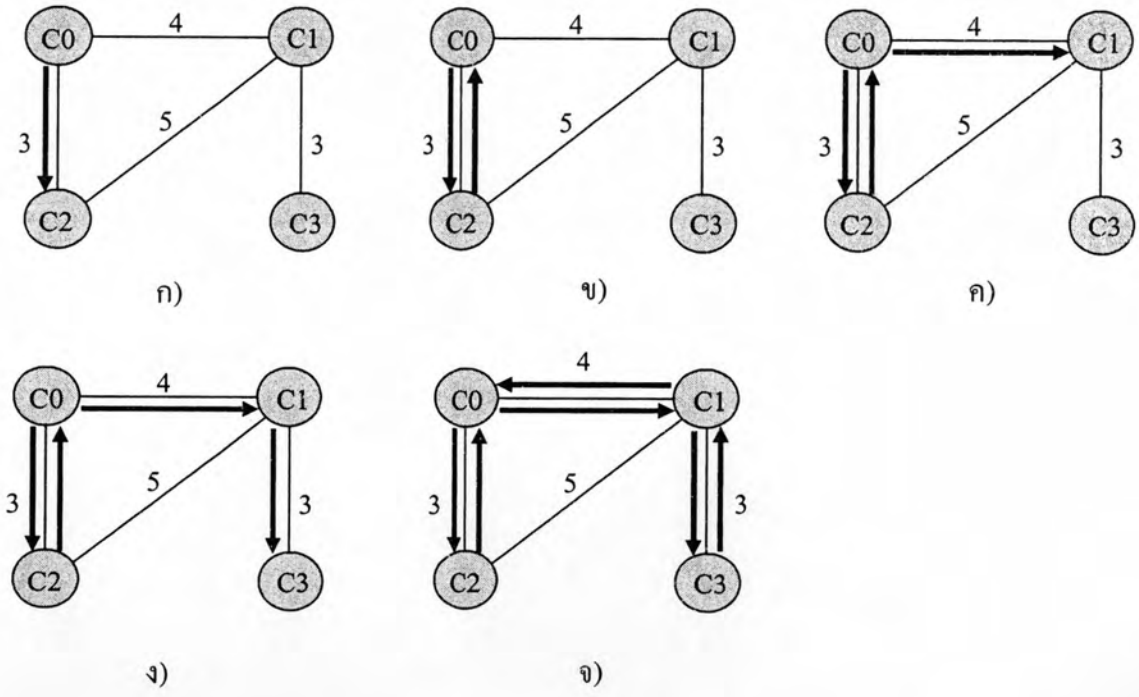
อย่างไรก็ตามการเดินทางจากหน่วยหนึ่งไปยังอีกหน่วยหนึ่งนั้น จะต้องเดินตามทางเดินที่สั้นที่สุดที่ได้คำนวณไว้ก่อนแล้ว จึงจะทำให้ได้ระยะทางที่สั้นที่สุดตามต้องการ ซึ่งสามารถระบุเป็นเส้นทางโดยละเอียดได้ดังรูปที่ 3.20

หอสมุดกลาง ศูนย์วิทยทรัพยากร จุฬาลงกรณ์มหาวิทยาลัย

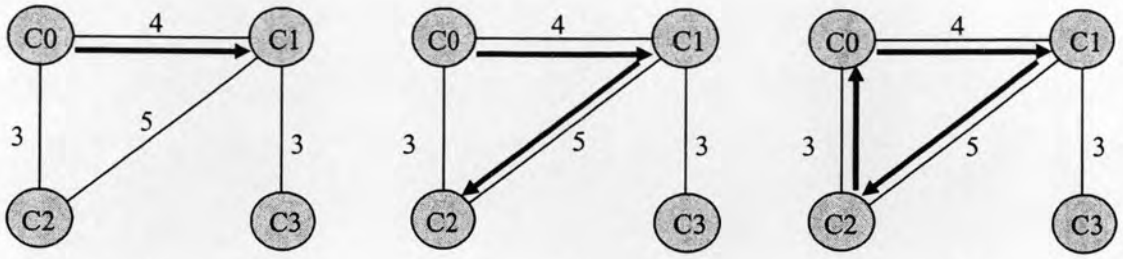


รูปที่ 3.20 เส้นทางการขนส่งเวชระเบียนของปัญหาตัวอย่างโดยละเอียด

จากเส้นทางการขนส่งในรูปที่ 3.20 สามารถวาดเป็นเส้นทางบนกราฟสำหรับรถเข็นคันที่ 1 และ 2 ได้ดังรูปที่ 3.21 และ 3.22 ตามลำดับ



รูปที่ 3.21 กราฟแสดงเส้นทางการขนส่งเวชระเบียนของรถเข็นคันที่ 1



รูปที่ 3.22 กราฟแสดงเส้นทางการขนส่งเวชระเบียนของรถเข็นคันที่ 2

จากรูปที่ 3.21 และ 3.22 คือเส้นทางการขนส่งที่เกิดขึ้นจริงตามปัญหาต้นแบบ จะได้ว่ารถเข็นคันที่ 1 วิ่งได้ระยะทาง 20 หน่วย รถเข็นคันที่ 2 วิ่งได้ระยะทาง 12 หน่วย ดังนั้นจึงสรุปได้ว่าฮิวริสติกที่นำเสนอสามารถหาคำตอบที่เหมาะสมได้โดยใช้รถเข็น 2 คัน วิ่งได้ระยะทางรวม 32 หน่วย

รายละเอียดของเส้นทางพร้อมตารางเวลาการเดินทางและการให้บริการในการรับและส่งเวชระเบียนของปัญหาตัวอย่าง แสดงดังตารางที่ 3.6

ตารางที่ 3.6 ตารางเวลาการรับส่งเวชระเบียนของปัญหาตัวอย่าง

เส้นทาง	หน่วย	การให้บริการรับหรือส่งเวชระเบียน		เวลาที่มาถึง	เวลาที่เริ่มบริการ	เวลารอคอย
		รับ	ส่ง			
R1	C0				0.00	
	C2	P1	-	3.00	5.00	2.00
	C0	P3	-	9.00	10.00	1.00
		P5	-	11.00	15.00	4.00
	C1	-	D3	20.00	20.00	0.00
	C3	-	D5	24.00	24.00	0.00
		-	D1	25.00	25.00	0.00
	C0				33.00	
R2	C0				0.00	
		P4	-	0.00	15.00	15.00
	C1	-	D4	20.00	20.00	0.00
		P2	-	21.00	21.00	0.00
	C2	-	D2	27.00	27.00	0.00
	C0				31.00	

จากตารางที่ 3.6 แสดงตารางเวลาการขนส่งเวชระเบียนทั้ง 5 ชิ้น โดยมีลำดับการขนส่งดังนี้ รถเข็นคันที่ 1 เริ่มต้นอยู่ที่หน่วยเวชระเบียน ณ นาทีที่ 0 จากนั้นเดินทางไปที่หน่วยตรวจโรคที่ 2 เพื่อรอรับเวชระเบียนชิ้นที่ 1 ณ นาทีที่ 5 (พร้อมได้รับการขนส่ง ณ นาทีที่ 5) แล้วเดินทางกลับไปยังหน่วยเวชระเบียนเพื่อทำการรอรับเวชระเบียนชิ้นที่ 3 ณ นาทีที่ 10 (พร้อมได้รับการขนส่ง ณ นาทีที่ 10) และชิ้นที่ 5 ณ นาทีที่ 15 (พร้อมได้รับการขนส่ง ณ นาทีที่ 15) จากนั้นเดินทางไปที่หน่วยตรวจโรคที่ 1 เพื่อส่งเวชระเบียนชิ้นที่ 3 ณ นาทีที่ 20 (จัดส่งภายใน 10 นาที) และเดินทางไปหน่วยตรวจโรคที่ 3 เพื่อส่งเวชระเบียนชิ้นที่ 5 ณ นาทีที่ 24 (จัดส่งภายใน 9 นาที) และส่งเวชระเบียนชิ้นที่ 1 ณ นาทีที่ 25 (จัดส่งภายใน 20 นาที) จากนั้นจึงเดินทางกลับมายังที่จอดรถ เสร็จสิ้นการขนส่ง ณ นาทีที่ 33

สำหรับรถเข็นคันที่ 2 เริ่มต้นอยู่ที่หน่วยเวชระเบียน ณ นาทีที่ 0 ทำการรอรับเวชระเบียนชิ้นที่ 4 ณ นาทีที่ 15 (พร้อมได้รับการขนส่ง ณ นาทีที่ 15) แล้วเดินทางไปยังหน่วยตรวจโรคที่ 1 เพื่อส่งเวชระเบียนชิ้นที่ 4 ณ นาทีที่ 20 (จัดส่งภายใน 5 นาที) และทำการรับเวชระเบียนชิ้นที่ 2 ณ นาทีที่ 21 (พร้อมได้รับการขนส่ง ณ นาทีที่ 10) จากนั้นเดินทางไปยังหน่วยตรวจโรคที่ 2 เพื่อส่งเวชระเบียนชิ้นที่ 2 ณ นาทีที่ 27 (จัดส่งภายใน 17 นาที) จากนั้นจึงเดินทางกลับมายังที่จอดรถ เสร็จสิ้นการขนส่ง ณ นาทีที่ 31