บทที่ 4
การประยุกต์ใช้วิธีโลคอลเสิร์จสำหรับปัญหาการจัดเส้นทาง และตารางเวลาการทำงานของยานพาหนะ

คำนำ

การแก้ปัญหาการตัดสินใจแบบไม่ต่อเนื่องที่มีซับซ้อนและมีขนาดใหญ่นั้น เนื่องจากไม่สามารถหาคำตอบที่ดีที่สุดได้ในเวลาที่จำกัด ดังนั้นวิธีการแก้ปัญหาดังกล่าวจึงนิยมใช้วิธีฮิวริสติกส์ในการแก้ปัญหา โดยวิธีฮิวริสติกส์นั้นเป็นวิธีที่ไม่สามารถรับประกันได้ว่าคำตอบที่หาได้นั้นเป็นคำตอบที่ดีที่สุด อย่างไรก็ตามนักวิจัยต่าง ๆ ก็พยายามคิดหาวิธีฮิวริสติกส์ต่าง ๆ ที่มีประสิทธิภาพในการหาคำตอบที่ใกล้เคียงกับคำตอบที่ดีที่สุด โดยทั่วไปแล้ววิธีฮิวริสติกส์เหล่านั้นเป็นวิธีที่เหมาะกับปัญหาเฉพาะปัญหานั้น เมื่อปัญหาเปลี่ยนแปลงไปวิธีดังกล่าวก็ไม่สามารถหาคำตอบได้ ถึงแม้ทำได้ก็ต้องปรับเปลี่ยนขั้นตอนการหาคำตอบเจากเดิมไปมาก วิธี Constrained local search (CLS) เป็นวิธีฮิวริสติกส์วิธีหนึ่งที่มีความยืดหยุ่นมาก สามารถเพิ่มลดเงื่อนไขของปัญหาได้ เพียงแค่เขียนเงื่อนไขดังกล่าวให้อยู่ในรูปของสมการก็สามารถเพิ่มเงื่อนไขเข้าไปได้ โดยไม่ต้องเปลี่ยนแปลงขั้นตอนการทำงาน

การประยุกต์ใช้วิธี CLS ในการแก้ปัญหานั้น จะคำนึงถึงปัญหาในโลกของความเป็นจริง คือ คำนึงถึงเงื่อนไขต่าง ๆ ที่เกิดขึ้นจริงในปัญหาดังกล่าวในปัจจุบันให้มากที่สุด ทั้งนี้เพื่อสามารถนำวิธีแก้ปัญหาดังกล่าวสามารถนำไปใช้งานได้จริงด้วย

กรอบแนวความคิด

ปัญหาการตัดสินใจแบบไม่ต่อเนื่องนั้น โดยส่วนใหญ่เป็นปัญหาที่มีตัวแปรการตัดสินใจมีค่าเป็น 0 หรือ 1 ดังนั้นการกำหนดปัญหาเพื่อใช้วิธี CLS ในการแก้ปัญหานั้น ต้องกำหนดปัญหาดังกล่าวให้อยู่ในรูปแบบตารางเมตริกซ์ 0/ 1 โดยตัวแปรการตัดสินใจนั้นมีความสัมพันธ์กันระหว่างแถวและคอลัมน์ และกำหนดเงื่อนไขขึ้นจากความสัมพันธ์ดังกล่าว จากนั้นเปลี่ยนเงื่อนไขเป็นค่าความขัดแย้ง โดยการกำหนดค่าความขัดแย้งนั้น ต้องคำนึงถึงวัตถุประสงค์ของการแก้ปัญหานั้น โดยปัญหาดังกล่าวนั้นอาจมีวัตถุประสงค์ของปัญหาหลายวัตถุประสงค์ วิธีที่ง่ายที่สุดคือการประเมินวัตถุประสงค์ในแต่ละข้อให้เป็นค่าใช้จ่าย ซึ่งวัตถุประสงค์ใดมีความสำคัญมากนั้นจะกำหนดค่าความขัดแย้งที่เกิดจากเงื่อนไขดังกล่าวนั้นมาก และวัตถุประสงค์ใดมีความสำคัญน้อยลงก็จะกำหนดค่าความขัดแย้งน้อยลงตามไปด้วย นอกจากนี้การเปลี่ยนเงื่อนไขต่าง ๆ ให้เป็นค่าความ

มหาวิทยาลัยบูรพา Burapha University

ขัดแย้งของเงื่อนไขหลักนั้น จะกำหนดเงื่อนไขที่หาคำตอบที่ถูกต้องตามเงื่อนไขดังกล่าวได้ยากให้มีค่าความขัดแย้งที่มาก และเงื่อนไขที่หาคำตอบที่ตรงตามเงื่อนไขได้ง่ายนั้นจะกำหนดให้มีค่าความขัดแย้งจากเงื่อนไขนั้นน้อยลงตามไปด้วย ทั้งนี้เพื่อ CLS จะหาคำตอบที่ตรงตามเงื่อนไขที่มีค่าความขัดแย้งมากให้ได้เสียก่อนจะปรับให้ตรงตามเงื่อนไขที่หาคำตอบที่ง่ายกว่า

การประยุกต์ใช้วิธีโลคอลเสิร์จในการแก้ปัญหาการจัดตารางยานพาหนะ

ปัญหาการจัดตารางยานพาหนะที่เรานำมาพิจารณาคือปัญหาการจัดตารางเวลารถประจำทาง ซึ่งเป็นปัญหาที่สำคัญของบริษัทผู้ให้บริการขนส่งสาธารณะ โดยทั่วไป การวางแผนการให้บริการรถประจำทางประกอบด้วย 4 ขั้นตอนได้แก่ 1) การกำหนดเส้นทางและความถี่ในการให้บริการ 2) การกำหนดตารางเวลาเดินรถ 3) การจัดตารางเวลาเดินรถประจำทาง และ 4) การจัดตารางเวลาการทำงานของพนักงาน เริ่มจากขั้นตอนการวางแผนเส้นทางเดินรถประจำทาง (Bus route หรือ Line) รวมทั้งการกำหนดความถี่ในการให้บริการ (Frequency, Headway) ต่อจากนั้น จะเป็นขั้นตอนการสร้างตารางเวลาเดินรถประจำทาง (Bus timetabling) โดยกำหนดจำนวนเที่ยวรถประจำทางทั้งหมดที่ต้องให้บริการต่อวัน หลังจากนั้น จะเป็นขั้นตอนการจัดตารางเดินรถประจำทาง (Bus scheduling) และจัดตารางเวลาการทำงานสำหรับพนักงาน (Crew scheduling) เพื่อให้บริการรถประจำทางตามตารางเวลาที่กำหนด

ในงานวิจัยนี้เสนอแนวคิดและวิธีแก้ปัญหาการจัดตารางเวลาเดินรถประจำทาง เพื่อให้ต้นทุนในการดำเนินงานน้อยที่สุด เราประเมินประสิทธิภาพของแบบจำลองและขั้นตอนวิธีทางคอมพิวเตอร์ที่พัฒนาขึ้นโดยใช้ข้อมูลจากองค์การขนส่งมวลชนกรุงเทพ

Solution method

เราทำการจำลองปัญหาการจัดตารางเวลาเดินรถประจำทางให้อยู่ในรูป Set partitioning ดังแสดงในภาพที่ 4.1 โดยกำหนดให้แถวแทนด้วยงาน (Job หรือ Timetabled trip) และคอลัมน์แทนด้วยรถประจำทาง งานวิจัยนี้พิจารณาปัญหาการจัดตารางเวลาเดินรถประจำทางจำนวนหลาย ๆ สายในครั้งเดียวโดยที่แต่ละสายอาจจะมีค่าใช้จ่ายคงที่ และสังกัดอยู่กองการเดินรถที่แตกต่างกัน

Timetabled trips		L1		L2	
		i1	i2	i3	i4
L1	8:00	0	1	0	0
	8:30	1	0	0	0
	9:00	0	1	0	0
L2	8:15	0	0	1	0
	8:35	0	0	0	1

ภาพที่ 4.1 ปัญหาการจัดตารางเวลาเดินรถประจำทางในรูป Set partitioning

มหาวิทยาลัยบูรพา Burapha University

ภาพที่ 4.1 แสดงตัวอย่างปัญหาโดยกำหนดให้มีงาน (Timetabled trip) ทั้งหมดจำนวน 5 งาน นั่นคือ เราจะต้องให้บริการรถประจำทางสาย L1 เริ่มต้นออกจากสถานีที่เวลา 8:00 8:30 และ 9:00 และให้บริการรถประจำทางสาย L2 เริ่มต้นออกจากสถานีที่เวลา 8:15 และ 8:35 รถประจำทาง i1, i2 สังกัดอยู่ในสาย L1 และ i3, i4 สังกัดอยู่ในสาย L2 จากตารางแสดงตัวอย่างคำตอบของปัญหา (ตัวแปรที่มีค่าเท่ากับ 1) คือ กำหนดให้รถประจำทาง i2 ให้บริการสาย L1 เริ่มทำงานที่เวลา 8:00 และ 9:00 ต่อเนื่องกัน รถประจำทาง i1 เริ่มทำงานที่เวลา 8:30 โดยให้บริการสาย L1 เช่นเดียวกัน ส่วน i3, i4 ให้บริการสาย L2 เริ่มทำงานที่เวลา 8:15 และ 8:35 ตามลำดับ เรากำหนดตัวแปรและพารามิเตอร์ในแบบจำลอง ดังนี้

$Z = \{1,...,z\}$	เซตของกองการเดินรถ หรือจุดจอดรถประจำทาง (Depots)
$L = \{1,...,l\}$	เซตของสายรถประจำทาง (Bus lines)
$L'_z = \{1,...,l_z\}$	เซตของสายรถประจำทางภายใต้กองการเดินรถ z
$W = \{1,...,w\}$	เซตของงาน (Starting jobs)
$W'_l = \{1,...,w_l\}$	เซตของงานสังกัดสาย l
$V = \{1,...,v\}$	เซตของรถประจำทาง
$V'_l = \{1,...,k_l\}$	เซตของรถประจำทางสังกัดสาย l
$D_i = \{1,...,d\}$	เซตลำดับการทำงานของรถประจำทาง i
Q_i	จำนวนงานที่รถประจำทาง i ทำงานได้
c_{ij}	ค่าใช้จ่ายของรถประจำทาง i ในการทำงาน j
x_{ij}	เท่ากับ 1 เมื่อรถประจำทาง i ทำงาน j ; 0 ในกรณีอื่น ๆ
e_{ij}	เวลาเริ่มงานลำดับที่ j ของรถประจำทาง i
t_{ij}	ระยะเวลาในการให้บริการ (Trip time) ของรถประจำทาง i ในงานลำดับ j
t_{pq}	ระยะเวลาในการเดินทาง (Transfer time) ของรถประจำทางระหว่าง Depot, p และ q
s_l	จำนวนรถประจำทางที่ใช้งานในสาย l
w_l	จำนวนงานทั้งหมดของสาย l
k_l	จำนวนรถประจำทางทั้งหมดที่สังกัดสาย l
FV_i	ค่าใช้จ่ายคงที่สำหรับการใช้งานรถประจำทาง i
LC_i	ค่าใช้จ่ายคงที่สำหรับการสลับสายรถประจำทาง i ต่อครั้งของงานก่อนหน้า j ไปยังงาน j

มหาวิทยาลัยบูรพา Burapha University

VT_j ค่าใช้จ่ายคงที่ในการใช้รถประจำทางระหว่าง Depot ของงานก่อนหน้า j ไปยังงาน j

f_i เท่ากับ 1 เมื่อรถประจำทาง i ทำงานอย่างน้อยหนึ่งงาน; 0 ในกรณีอื่น ๆ

a_{ij} เท่ากับ 1 เมื่อรถประจำทาง i มีการสลับสายเมื่อทำงาน j ; 0 ในกรณีอื่น ๆ

b_{ij} เท่ากับ 1 เมื่อรถประจำทาง i มีการใช้รถระหว่าง Depot ในการทำงาน j ; 0 ในกรณีอื่น ๆ

เราจำลองปัญหาการจัดตารางเวลาเดินรถประจำทางให้อยู่ในรูปเงื่อนไข (Constraint-based formulation) โดยกำหนดให้เงื่อนไขที่ไม่ยอมให้มีความขัดแย้งเกิดขึ้นเป็นเงื่อนไขหลัก (Hard constraint) และวัตถุประสงค์ของปัญหาเป็นเงื่อนไขรอง (Soft constraint) อัลกอริทึมจะพบคำตอบที่ดีที่สุดก็ต่อเมื่อ 1) เงื่อนไขหลักทุกตัวมีค่า Hard violation, $H = 0$ และ 2) เมื่อ Soft violation, S ของเงื่อนไขรองมีค่าน้อยที่สุด เราสามารถคำนวณหาค่าความขัดแย้งของเงื่อนไข (Constraint violation, v) ได้จากสมการ

$$Ax \le b \Rightarrow v = \max(0, Ax - b) \tag{4.1}$$

เมื่อ A คือ ค่าสัมประสิทธิ์, x คือ ตัวแปรการตัดสินใจ, และ b คือ ค่าขีดจำกัด (Bound)

Hard constraints

เราแบ่งเงื่อนไขหลักออกเป็น 2 ประเภทคือ 1) เงื่อนไขทั่วไป (Basic constraint) และเงื่อนไขอื่น ๆ (Side constraint) เราคำนวณหาค่าความขัดแย้งของเงื่อนไขได้ดังนี้

Basic constraints:

$$h_p = \max\left(0, \sum_{j \in W} \left| \sum_{i \in T} x_{ij} - 1 \right| \right) \tag{4.2}$$

เงื่อนไข (4.2) กำหนดให้งาน j ถูกให้บริการโดยรถประจำทาง i เพียงครั้งเดียวเท่านั้น นั่นคือ ผลรวมของรถประจำทางทุกคันที่ให้บริการงาน j ต้องมีค่าเท่ากับหนึ่ง และบังคับใช้กับทุกงาน, h_p คือค่าความขัดแย้งที่เกิดขึ้น

$$h_c = \max\left(0, \sum_{i \in T} \sum_{j \in D_i} \left[\left(e_{i,j-1} + t_{i,j-1} \right) - e_{ij} \right] \right) \tag{4.3}$$

เงื่อนไข (4.3) กำหนดให้เวลาในการทำงานก่อนหน้าเสร็จสิ้นก่อนเวลาเริ่มงานปัจจุบัน นั่นคือ เวลาเริ่มงานรวมกับเวลาในการทำงานก่อนหน้าต้องน้อยกว่าเวลาเริ่มงานปัจจุบัน, h_c คือค่าความขัดแย้งที่เกิดขึ้น

Side constraints

Route time constraints

เนื่องจากข้อจำกัดด้านน้ำมันเชื้อเพลิง การซ่อมบำรุง ชั่วโมงการทำงานของพนักงานขับรถ และข้อจำกัดด้านอื่น ๆ บริษัทผู้ให้บริการขนส่งสาธารณะอาจจะกำหนดเงื่อนไขเวลาการทำงานของรถประจำทาง (Feasible block time) เพื่อให้การใช้ทรัพยากรทั้งในส่วนของรถประจำทางและพนักงานเกิดความสมดุลและมีประสิทธิภาพ ในงานวิจัยนี้สามารถพิจารณา Timetabled trip 2 อันที่อยู่ระหว่าง Block time เป็น Trip เดียวกัน (A continuous trip) โดยกำหนดให้ Time break อยู่ในรูป Banned timeslots ที่รถประจำทางไม่สามารถทำงานได้ เพื่อทำให้บริษัทสามารถประเมินประสิทธิภาพของเที่ยวการให้บริการรถประจำทางได้อย่างเหมาะสม ข้อจำกัดของเวลาการให้บริการในเส้นทางแสดงดังสมการ

$$h_r = \min\left(0, \sum_{i \in I}\left(e_{i,m^*-1} - \left(e_{im^*} + t_{im^*} + bk\right)\right)\right) \tag{4.4}$$

เงื่อนไข (4.4) เมื่อ $m^* = j \bmod m \ \forall j \in D_r$, m คือ จำนวนเที่ยวการทำงานต่อเนื่องมากสุด (Max continuous trip). bk คือ ระยะเวลาใน Block time (หน่วย: นาที), และ h_r คือ ค่าความขัดแย้งที่เกิดขึ้น

$$h_c = \max\left(0, \sum_{i \in I}\left|\sum_{i \in I} x_{ij} - Q_i\right|\right) \tag{4.5}$$

เงื่อนไข (4.5) บังคับไม่ให้รถประจำทางทำงานเกินจำนวนที่กำหนดไว้ นั่นคือ ผลรวมของงานที่รถประจำทาง i ให้บริการจะต้องน้อยกว่าหรือเท่ากับจำนวนงานที่รถประจำทางสามารถทำงานได้, h_c คือค่าความขัดแย้งที่เกิดขึ้น

Line change and vehicle transfer

แม้ว่าแบบจำลองที่นำเสนอสามารถกำหนดค่าใช้จ่ายคงที่สำหรับการวิ่งสลับสาย (Line change) และการใช้รถประจำทางร่วมกันระหว่างกองการเดินรถ (Multi-depot operations) เพื่อให้ค่าใช้จ่ายในการดำเนินงานทั้งหมดน้อยที่สุด อย่างไรก็ตาม เนื่องจากเงื่อนไขในการบริหารจัดการ

มหาวิทยาลัยบูรพา Burapha University

บางอย่าง เราอาจจะทำการสลับสายหรือใช้รถประจำทางร่วมกันระหว่างกองการเดินรถ ได้เฉพาะรถประจำทางเพียงบางคันและบางช่วงเวลาเท่านั้น ยกตัวอย่างเช่น รถประจำทางที่ทำสัญญาติดป้ายโฆษณารอบคัน รถจะต้องวิ่งให้บริการเฉพาะสายนั้น ๆ สังเกตด้วยว่า การวิ่งสลับสายและการใช้รถประจำทางร่วมกันระหว่างกองการเดินรถ อาจก่อให้เกิดความไม่ชัดเจนในการวัดประสิทธิภาพรายได้และค่าใช้จ่ายของรถประจำทางแต่ละคันว่า รถประจำทางที่ถูกสลับสายไปวิ่งให้บริการสายอื่น ควรจะเป็นของสายการเดินรถใด เงื่อนไขการวิ่งสลับสายและการใช้รถประจำทางร่วมกันระหว่างกองการเดินรถ แสดงดังสมการที่ (4.6) และ (4.7) ตามลำดับ

$$h_m = \min\left(0, \sum_{l\in L}\left[\sum_{i\in I}\left(\sum_{j\in W_l} x_{ij}\right) m_{il}\right]\right) \tag{4.6}$$

$$h_t = \min\left(0, \sum_{z\in Z}\sum_{i\in I}\left[\sum_{l\in L_z}\left(\sum_{j\in W_l} x_{ij}\right) n_{iz}\right]\right) \tag{4.7}$$

เมื่อ $m_{il} = 1$ เมื่อรถประจำทาง i ในสาย l ไม่สามารถวิ่งสลับสายได้ และ $n_{iz} = 1$ เมื่อรถประจำทาง i ในกองการเดินรถ z ไม่สามารถใช้ร่วมกันระหว่างกองการเดินรถได้ h_m และ h_t คือค่าความขัดแย้งที่เกิดขึ้นจาก Line changes และ Vehicle transfers

Relaxed hard time windows

ในหลาย ๆ ครั้งเมื่อได้แผนการจัดตารางเวลาเดินรถประจำทางแล้ว นักวางแผนอาจต้องการทราบโดยทันทีว่า เมื่อเปลี่ยนแปลงเงื่อนไขของปัญหาบางตัวแล้วจะทำให้ตารางการเดินรถเปลี่ยนแปลงไปมากน้อยเพียงใด ในงานวิจัยนี้สามารถละเว้น (Relax) เงื่อนไข Late time-window ของงาน W' (เงื่อนไข 4.3) โดยที่ W'' คือเซตของงาน (Timetable trips) ที่ Late time-window ถูกเปลี่ยนจากเงื่อนไขหลักไปเป็นเงื่อนไขรองเพื่อก่อให้เกิดความยืดหยุ่นในการแก้ปัญหาความไม่แน่นอนของเวลาในการเดินทางของรถประจำทางในบางช่วงเวลา ซึ่งสามารถเขียนได้ดังสมการ

$$h_e = \max\left(0, \sum_{i\in I}\sum_{j\in D}\left[\left(e_{i,j-1} + t_{i,j-1}\right) - e_{ij}\right]\left(1 - y_j\right)\right) \tag{4.8}$$

$$s_e = \max\left(0, \sum_{i\in I}\sum_{j\in D_i}\left[\left(e_{i,j-1} + t_{i,j-1}\right) - e_{ij}\right] y_j\right) \tag{4.9}$$

มหาวิทยาลัยบูรพา Burapha University

เราแทนเงื่อนไข (4.3) ด้วย (4.8) และเพิ่มเงื่อนไข (4.9) ซึ่งเป็นค่าความขัดแย้งที่เกิดจากเงื่อนไขรอง s_c โดยที่ y_j เท่ากับ 1 เมื่อ $j \in W'$ และเท่ากับ 0 ในกรณีอื่น ๆ

Soft constraint

เราแปลงฟังก์ชันวัตถุประสงค์ของปัญหาให้อยู่ในรูปเงื่อนไขรอง และคำนวณค่าความขัดแย้ง (Soft violation, S) ได้ดังสมการ

$$S = \max\left(0, \sum_{i \in I}\left[\sum_{i \in H}\left(c_{ij}x_{ij} + LC_i a_{ij} + VT_j h_{ij}\right) + FV_i f_i\right] + S_c\right) \qquad (4.10)$$

สังเกตด้วยว่า เราพิจารณา LC_i และ VT_j เป็นค่าใช้จ่ายคงที่เหมือนกับ FV_i สำหรับ LC_i เป็น ค่าใช้จ่ายในการสลับสายรถประจำทางที่อยู่ในสายที่สังกัดไปวิ่งบริการสายอื่น ดังนั้น ค่าใช้จ่ายคงที่อาจจะพิจารณาจากค่าใช้จ่ายในการบริหารจัดการ ยกตัวอย่างเช่น การเปลี่ยนตัวเลขแสดงสายรถประจำทางแบบอิเล็กทรอนิกส์ เป็นต้น สำหรับ VT_j เป็นค่าใช้จ่ายคงที่ที่เกิดจากการใช้รถประจำทางระหว่างกองการเดินรถ โดยคิดแยกมาจากค่าใช้จ่ายที่เกิดจากการเดินทางระหว่างกองการเดินรถ นั่นคือค่าใช้จ่ายในการบริหารจัดการ หรือค่าใช้จ่ายในการดำเนินงานอื่น ๆ

จากสมการ (4.2) – (4.8) เราจะสามารถคำนวณและได้ค่าความขัดแย้งของเงื่อนไขหลัก คือ $H = h_p + h_r + h_c + h_m + h_t + h_s$ และคำนวณหาค่าความขัดแย้งจากเงื่อนไขทั้งหมด (Total constraint violation), $V = H + S$

เราประยุกต์ใช้วิธี CLS ในการแก้ปัญหาการจัดตารางเวลาเดินรถประจำทาง โดยอธิบายถึงขั้นตอนการประมวลผลเบื้องต้น และขั้นตอนการทำงานของวิธี CLS

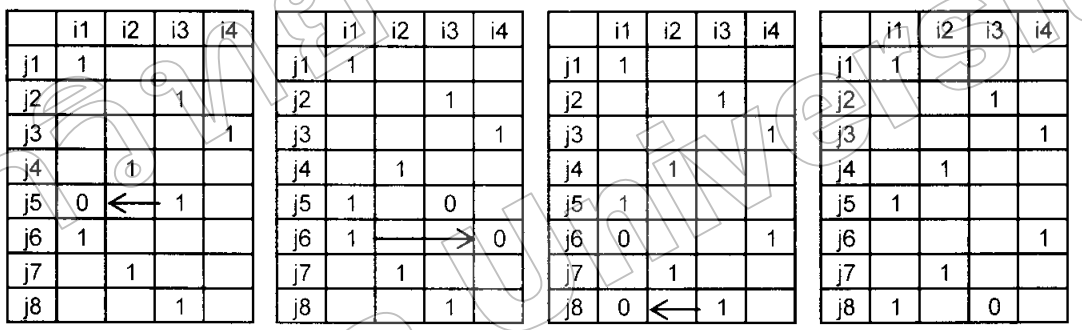
Sequencing feasible trips

ในการหาคำตอบของแบบจำลอง Bus scheduling ที่สร้างขึ้น สิ่งสำคัญอย่างหนึ่งคือการกำหนดลำดับการทำงาน ต้องบอกได้ว่ารถประจำทาง i ต้องทำงานใดก่อน และทำงานใดต่อจากนั้น เนื่องจากการหาคำตอบของปัญหาการจัดตารางเวลาเดินรถประจำทาง ตารางเวลาการเดินรถได้ถูกกำหนดไว้ล่วงหน้าแล้ว ดังนั้นเราจึงทราบว่าแต่ละงาน j รถประจำทาง i ต้องเริ่มให้บริการเวลาใด เราทำการ Pre-process โดยนำงานของรถประจำทางทุกสาย และทุกกองการเดินรถมาเรียงลำดับเวลาเริ่มต้นการให้บริการ (Departure time) จากน้อยไปหามาก ดังนั้น เราจะได้ $e'_{j-1} < e'_j \quad \forall j \in W$ เมื่อ e'_j คือเวลาเริ่มต้นการให้บริการ (หรือเวลาเริ่มต้นทำงาน j) ทั้งนี้เพื่อลดความซับซ้อนและลดพื้นที่ในการหาคำตอบของปัญหา

มหาวิทยาลัยบูรพา Burapha University

Constrained local search

วิธี CLS เริ่มแรกถูกพัฒนาโดย Indra-Payoong et al. (2005) โดยทำการพิจารณาปัญหาการตัดสินใจใด ๆ ประกอบไปด้วยเงื่อนไขหลักและเงื่อนไขรอง เรากำหนดให้ค่าความขัดแย้งเกิดขึ้นเมื่อกำหนดค่าตัวแปรในแบบจำลองแล้วขัดแย้งกับเงื่อนไขของปัญหา โดยอาศัยรูปแบบจำลอง Set partitioning เราสามารถแสดงเงื่อนไขหลักและเงื่อนไขรองให้อยู่ในรูปเมตริกซ์ของเงื่อนไข (Constraint matrix) เงื่อนไขอาจจะเป็นอิสระต่อกันหรือมีความสัมพันธ์กันภายในเมตริกซ์ ค่าความขัดแย้งของเงื่อนไขทุกตัวจะถูกรวมกันเป็นค่าความขัดแย้งของเงื่อนไขทั้งหมด V วิธี CLS ใช้ Variable flip เป็น Local move และอาศัยกลยุทธ์การเลือกตัวแปรแบบสุ่ม (Random variable selection) CLS ใช้ค่า V ในการประเมินประสิทธิภาพของการปรับปรุงคำตอบในแต่ละรอบการกระทำซ้ำ ภาพที่ 4.2 แสดงตัวอย่างการทำงานของ Variable flip operations



ภาพที่ 4.2 Variable flip operations

ขั้นตอนการทำงานของวิธี CLS สามารถสรุปได้ดังนี้

Step 1: Constraint selection

หลังจากกำหนดค่าของตัวแปรเริ่มต้น (Initial assignment, A) อัลกอริทึมจะสุ่มเลือกคอลัมน์ขึ้นมาจำนวน nc ซึ่งการเลือกนั้นจะให้ความสำคัญกับคอลัมน์ที่มีค่าความขัดแย้งกับเงื่อนไขหลัก (Violated hard constraints) ก่อนเป็นลำดับแรก

Step 2: Variable selection

จากขั้นตอน 1 อัลกอริทึมจะเลือกตัวแปร หรือแถวขึ้นมา 1 แถว สำหรับปัญหาการจัดตารางเวลาเดินรถประจำทาง (เนื่องจากจำนวนคอลัมน์มีไม่มากนัก) เราสุ่มเลือกแถว ที่มีค่าตัวแปรเท่ากับ 1 เพื่อนำมาทดลองสลับค่า (Trial flip) กับตัวแปรภายในแถวทั้งหมด

มหาวิทยาลัยบูรพา Burapha University

Step 3: Move acceptance

อัลกอริทึมเลือกค่า V' ที่ดีที่สุดภายในคอลัมน์และแถวที่ถูกทดลองสลับค่า แล้วกำหนดให้เป็นคำตอบปัจจุบัน $A_i \leftarrow A'$

ขั้นตอนทั้ง 3 นี้อาจจะถูกออกแบบให้มีรายละเอียดหรือมีลักษณะเฉพาะปัญหามากขึ้น (Domain-specific rules) เพื่อเพิ่มประสิทธิภาพการทำงานของวิธี CLS

ผลการทดลองและวิเคราะห์ผลการทดลอง

The BMTA case study

เราประเมินประสิทธิภาพของระบบจัดตารางเวลาเดินรถประจำทางที่พัฒนาขึ้นโดยใช้ข้อมูลจากองค์การขนส่งมวลชนกรุงเทพ (The Bangkok mass transit authority: BMTA). BMTA เป็นรัฐวิสาหกิจสังกัดกระทรวงคมนาคม ให้บริการรถประจำทางจำนวน 3,535 คัน 108 สายทั่วกรุงเทพมหานคร BMTA แบ่งเขตการให้บริการรถประจำทางออกเป็น 8 เขต (Zone) แต่ละเขตแบ่งออกเป็นกองการเดินรถประมาณ 3 กอง (Depot หรือ Division) แต่ละกองประกอบด้วยเส้นทางประมาณ 5-10 สาย ปัจจุบัน BMTA สร้างตารางเวลาเดินรถและจัดรถประจำทางเป็นรายเดือน โดยอาศัยนักวางแผนที่มีประสบการณ์ ซึ่งจำนวนเที่ยวให้บริการทั้งหมดในแต่ละวันจะถูกกำหนดในเชิงนโยบายจากสำนักงานใหญ่อีกทีหนึ่ง ขั้นตอนการจัดรถประจำทางของ BMTA จะขึ้นอยู่กับตารางการทำงานของพนักงานประจำรถ (พนักงานขับรถ และพนักงานเก็บสตางค์) เป็นหลัก ซึ่งโดยทั่วไป พนักงานขับรถและพนักงานเก็บสตางค์จะทำงานคู่กันและใช้รถประจำทางคันเดิมในการให้บริการ

เนื่องจากในปัจจุบัน BMTA ยังไม่มีระบบคอมพิวเตอร์ช่วยในการจัดตารางเวลาเดินรถ รวมทั้งเวลาการเดินทางของรถประจำทางที่ไม่แน่นอนในกรุงเทพฯ ในแต่ละวัน ผู้จัดการสายจะจัดรถประจำทางตามหน้างาน โดยอาศัยแผนจำนวนเที่ยวของรถประจำทางในแต่ละชั่วโมง (Hourly timeslot) ที่กำหนดไว้ล่วงหน้าเป็นแนวทางในการปฏิบัติ นั่นคือ เมื่อรถประจำทางกลับเข้าถึงกองการเดินรถในทุก ๆ Round trip พนักงานจะทำการบันทึกเวลา จำนวนเที่ยวการทำงาน และรายได้จากการให้บริการ ต่อจากนั้น ผู้จัดการสายจะพิจารณาทรัพยากร (รถประจำทางและพนักงาน) ที่มีอยู่ในขณะนั้น และทำการกำหนดเวลาการให้บริการของรถประจำทางในสายนั้น ๆ ใหม่ตามความต้องการในการเดินทาง สภาพการจราจร และทรัพยากรที่มีอยู่

Potential cost savings with line changes and multi-depot

จากข้อมูลการทำงานในปัจจุบัน แม้ว่าเงื่อนไขในเชิงนโยบายบางประการอาจจะไม่อนุญาตให้รถประจำทาง วิ่งสลับสาย อย่างไรก็ตาม หลาย ๆ ครั้งในทางปฏิบัติ BMTA มีการวิ่งสลับ

มหาวิทยาลัยบูรพา Burapha University

สายอยู่บ้าง ซึ่งจะกระทำขึ้นที่หน้างานโดยทันที โดยการสลับสายนั้น ผู้จัดการสายจะพิจารณาจากช่วงเวลา และบาง Section ของ Line ในแต่ละวัน (ในบางสายช่วงเวลาที่เป็น Peak demand อาจไม่เท่ากัน โดยเฉพาะอย่างยิ่งในช่วงเย็นหลังเลิกงาน ผู้โดยสารอาจจะเลิกงานพร้อมกัน แต่อาจจะทยอยกันกลับบ้านในช่วงเวลาที่แตกต่างกัน ในทางตรงกันข้าม ช่วง Morning peak hour รถประจำทางบางสายอาจจะติดอยู่ในกระแสการจราจรทั้งหมด) ดังนั้น จึงมีโอกาสในการจัดสรร (Re-allocate) รถประจำทางจากสายอื่นมาใช้ สังเกตอีกด้วยว่า ในปัจจุบัน BMTA ยังไม่มีการใช้รถประจำทางร่วมกันระหว่างกองการเดินรถ นั่นอาจเป็นเพราะว่า BMTA แบ่งโครงสร้างการบริหารในระดับกองการเดินรถเป็นอิสระต่อกัน

เราทดสอบแบบจำลองที่พัฒนาขึ้นโดยกำหนดค่า Headway ของรถประจำทางแต่ละสายให้แปรผันตามช่วงเวลา และกำหนด Trip time (โดยใช้ค่าเฉลี่ยของเวลาในการเดินทางในอดีตในแต่ละช่วงเวลาของวันต่าง ๆ) ของรถประจำทางปรับอากาศจำนวน 9 สาย 125 คัน สังกัดเขตการเดินรถที่ 4 ซึ่งประกอบด้วย 3 กองการเดินรถ เพื่อแสดงให้เห็นถึงศักยภาพในการลดต้นทุนในการดำเนินงาน เมื่อกองการเดินรถทั้ง 3 ใช้ทรัพยากรของรถประจำทางร่วมกัน เรากำหนดค่าพารามิเตอร์ที่ใช้ในแบบจำลองดังนี้ ค่าใช้จ่ายคงที่ของรถประจำทาง/ คัน/ วัน เท่ากับ 5,000 บาท (ซึ่งอาจจะประมาณจากค่าเช่า หรือ Asset cost ที่เกิดขึ้นจริง) ค่าใช้จ่ายในการสลับสายต่อครั้งเท่ากับ 100 บาท (× 2 คิดเปลี่ยนไปและเปลี่ยนกลับ) ค่าใช้จ่ายในการใช้รถประจำทางจากกองการเดินรถอื่นต่อครั้ง (โดยคิดแยกออกจากค่าใช้จ่ายที่เกิดจากระยะทางในการเดินทาง) เท่ากับ 500 บาท (× 2 คิดเรียกออกและเรียกเข้า) และค่าใช้จ่ายในการใช้งานรถประจำทางต่อกิโลเมตรเท่ากับ 30 บาท ผลการทดลองแสดงในตารางที่ 4.1

มหาวิทยาลัยบูรพา Burapha University

ตารางที่ 4.1 Multiple-depot with line change operations

Line/	Trips	Pure-line		Line change (LC)		Multi-depot with LC		Max. Savings %	
[#Bus]		#Bus	Cost	#Bus	Cost	#Bus	Cost	#Bus	Cost
Depot 1									
4 [18]	75	8	80687.5	7	82287.5	4	75687.5	50	6.2
72 [5]	78	8	80950	7	83450	7	86550	12.5	-6.92
205 [12]	76	11	94900	4	64700	4	75600	63.64	20.34
552 [15]	62	10	134630	5	115230	3	119630	70	11.14
Depot 2									
62 [14]	79	9	94770	10	104370	5	92870	44.44	2
77 [18]	101	12	155445	8	140445	4	131145	66.67	15.63
Depot 3									
12 [10]	66	7	69650	3	52450	3	64850	57.14	6.89
137 [18]	98	12	114880	8	101880	7	95780	41.67	16.63
551 [15]	53	8	91940	6	87240	4	84740	50	7.83
Total [125]	688	85	917853	58	832052.5	41	826852.5	51.76	9.91

จากตารางที่ 4.1 จะเห็นได้ว่า เมื่อใช้ระบบที่พัฒนาขึ้น (DSS) จัดตารางเดินรถประจำทางแบบสลับสาย (Line change) ทำให้ BMTA สามารถลดจำนวนรถประจำทางที่ใช้ทั้งหมดลงมากถึง 27 คัน และเมื่อมีการใช้รถประจำทางร่วมกันระหว่างกองการเดินรถ (Multi-depot) ด้วย สามารถลดจำนวนรถประจำทางลงไปได้อีก 17 คัน สังเกตด้วยว่า การใช้รถประจำทางระหว่างกองการเดินรถสามารถลดค่าใช้จ่ายในการดำเนินงานลงเพียงเล็กน้อยเมื่อเปรียบเทียบกับการวิ่งสลับสาย เนื่องจากต้องเสียค่าใช้จ่ายในการ Transfer รถระหว่างกอง ดังนั้น ในตารางที่ 1 ระบบ DSS สามารถช่วย BMTA ลดค่าใช้จ่ายในการดำเนินงานได้ 91,000 บาทต่อวัน (หรือ 2.73 ล้านบาทต่อเดือน)

เราตระหนักด้วยว่า จำนวนรถประจำทางที่ใช้งานทั้งหมดนั้นแปรผันโดยตรงกับความแตกต่างของการให้บริการรถประจำทางในแต่ละสาย (นั่นคือ ระยะเวลาของ Trip และ Headway ที่กำหนดในแต่ละช่วงเวลา) ยิ่งมีความแตกต่างกันมาก ก็จะทำให้มีโอกาสในการใช้ทรัพยากรร่วมกันได้มาก ซึ่งทำให้ BMTA สามารถใช้จำนวนรถประจำทางที่ลดลงได้ เราอาจจะตั้งข้อสังเกตด้วยว่าผลที่ได้จากตารางที่ 3 อาจจะไม่เป็นจริงทั้งหมดในทางปฏิบัติ ในลักษณะการทำงานที่เป็นอยู่ในปัจจุบัน อย่างไรก็ตาม เมื่อมีสายรถประจำทางจำนวนมาก วิ่งสลับสาย และใช้รถร่วมกันระหว่างกอง ระบบ DSS จะยังคงสามารถช่วยนักวางแผนในการจัดตารางเวลาเดินรถได้โดยง่ายและมีประสิทธิภาพ

มหาวิทยาลัยบูรพา Burapha University

การประยุกต์ใช้โลคอลเสิร์จในการแก้ปัญหาตารางทำงานของพนักงาน

ที่มาและความสำคัญของปัญหา

ปัญหาการจัดตารางเวลาการทำงานของพนักงานของระบบขนส่งมวลชน (Crew scheduling problem: CSP) เป็นปัญหาการตัดสินใจที่มีขนาดใหญ่และมีความซับซ้อนมากกว่าปัญหาการตัดสินใจในภาคอุตสาหกรรมขนส่งและโลจิสติกส์ทั่ว ๆ ไป เนื่องจากพนักงานแต่ละคนถูกพิจารณาเป็นทรัพยากร 1 หน่วย และมีเป็นจำนวนมาก ยกตัวอย่างเช่น เมื่อเปรียบเทียบกับปัญหาการจัดเส้นทางและตารางเวลาของยานพาหนะ ซึ่งทรัพยากรคือจำนวนยานพาหนะซึ่งอาจมีจำนวนไม่กี่สิบคันเท่านั้น นอกจากนี้ ปัญหา CSP ยังมีเงื่อนไขที่เกี่ยวกับความพึงพอใจของพนักงาน (Human factors) เข้ามาเกี่ยวข้องด้วย อาทิเช่น จำนวนชั่วโมงการทำงาน เวลาเริ่มงาน (กะการทำงาน) เวลาการหยุดพักระหว่างการทำงาน วันหยุด การจัดสรรค่าล่วงเวลา ทักษะและความเหมาะสมของงานที่ทำ ฯลฯ ซึ่งเงื่อนไขในการทำงานต่าง ๆ เหล่านี้จะต้องพิจารณาอย่างละเอียดรอบคอบ ไม่เพียงต้องการจัดสรรให้พนักงานทำงานให้เกิดประสิทธิภาพมากที่สุด ด้วยต้นทุนในการดำเนินการน้อยที่สุด แต่เรายังต้องคำนึงถึงระดับความพึงพอใจของพนักงานและความเสมอภาคในการทำงานให้มากที่สุดด้วยเช่นกัน

ในบทนี้ เราพิจารณาปัญหา CSP ขององค์การขนส่งมวลชนกรุงเทพ (ขสมก.) ในขั้นตอนการจัดตารางเวลาการทำงานของพนักงานจะถูกกระทำอย่างเป็นอิสระกับขั้นตอนการจัดตารางเวลาเดินรถประจำทาง และถูกกระทำประมาณ 1-2 เดือนล่วงหน้า ในปัจจุบันการจัดตารางเวลาการทำงานของพนักงานเดินรถเริ่มต้นจากการกำหนดจำนวนกะการทำงาน (Job timeslot) ให้รถประจำทางแต่ละสายในแต่ละวัน ภาพที่ 4.3 แสดงตัวอย่าง Job timeslot ของงานในแต่ละวัน

Morning shift			Afternoon shift			Night shift		
No.	Start	End	No.	Start	End	No.	Start	End
1	03:25	11:25	1	12:10	20:10	1	18:30	02:30
2	03:40	11:40	2	12:20	20:20	2	19:30	03:30
3	04:00	12:00	3	12:30	20:30	.		
.			.			.		
.			.			.		
N	05:25	13:25		14:10	22:10			

ภาพที่ 4.3 Job timeslot

จากภาพที่ 4.3 จะเห็นได้ว่า ขสมก. แบ่งกะในการทำงานออกเป็น 3 ประเภทได้แก่ กะเช้า กะบ่าย และกะสว่าง โดยในแต่ละกะจะมีเวลาเริ่มต้นและสิ้นสุดการทำงาน (หรือ Sign-in/ Sign-off time) ที่แตกต่างกัน พนักงานแต่ละคนจะมีชั่วโมงการทำงานเท่ากับ 8 ชั่วโมงต่อวัน ยกตัวอย่างเช่น

มหาวิทยาลัยบูรพา Burapha University

พนักงานเริ่มทำงานเวลา 3.25 น. และเลิกงานเวลา 11.25 น. เป็นต้น สังเกตด้วยว่าการกำหนดช่วงเวลาของกะที่แน่นอนสามารถช่วยให้นักวางแผนจัดสรรเวลาการทำงานของพนักงานได้ง่าย รวมทั้งยังช่วยให้พนักงานสามารถปฏิบัติตามแผนการทำงานได้ง่ายด้วย

เมื่อจำนวนกะการทำงาน (Job timeslot) ได้ถูกกำหนดแล้วเสร็จ ขั้นตอนต่อไปนักวางแผนจะกำหนดให้พนักงานทำงานตามกะการทำงาน ดังแสดงตัวอย่างในภาพที่ 4.4

Code	Name	Lot	Job timeslot		Break	Fri	Sat	Sun	.	Fri
			Start	End	(B)	1	2	3	.	29
425102	Crew 1	1	03:25	11:25	Sun	11	8	B		18
432099	Crew 2	2	03:40	11:40	Sat	12	B	6		1
439224	Crew 3	3	04:00	12:00	Fri	B	10	7		2
N										

ภาพที่ 4.4 Day shift

ภาพที่ 4.4 แสดงกะการทำงานของพนักงานในแต่ละวัน (Day shift) ยกตัวอย่างเช่น Lot 1 เริ่มงานเวลา 3.25 น. และเลิกงานเวลา 11.25 น. พนักงานคนที่ 1 (Crew 1) ถูกกำหนดให้ทำงานในวันศุกร์ที่ 1 ใน Lot 11 วันเสาร์ที่ 2 ทำงานใน Lot 8 ... และวันศุกร์ที่ 29 วันสุดท้ายของเดือนทำงานใน Lot 18 และมีวันหยุด (Break: B) ประจำสัปดาห์คือวันอาทิตย์

สังเกตด้วยว่าในขั้นตอนการกำหนด Day shift ให้พนักงานนั้นสามารถทำเป็นระบบอัตโนมัติ (Automated assignment) ได้ไม่ยากนัก อย่างไรก็ตามปัจจัยทางด้าน Human factors หลาย ๆ ตัวอาจจะทำให้อัลกอริทึมที่พัฒนาขึ้นไม่สามารถนำไปใช้ได้จริงในทางปฏิบัติ ทั้งนี้เนื่องจากไม่สามารถสร้างแบบจำลองทางคณิตศาสตร์ในรูปแบบที่ตายตัวได้ รวมทั้งการกำหนดเงื่อนไขในการทำงานบางอย่างอาจจะมีความไม่ชัดเจนโดยตั้งใจ หรือไม่ตั้งใจ อาทิเช่น พนักงานบางคนอาจจะมีข้อจำกัดในด้านครอบครัว ไม่เต็มใจที่รับงานในกะบ่าย หรือกะสว่าง พนักงานขับรถ B จะถูกกำหนดกะทำงานไม่ให้ต่อจากพนักงาน A เพราะพนักงาน B ขับรถเร็วกว่า A ทำให้ตารางเวลาการเดินแตกต่างจากที่วางแผนไว้มาก เป็นต้น เนื่องจากในแต่ละกองการเดินรถ (Depot) มีพนักงานเป็นจำนวนมาก (~ 800 คน) การสร้างตาราง Day shift ในแต่ละเดือนอาจจะต้องใช้เวลามาก ดังนั้น ในทางปฏิบัตินักวางแผนจึงใช้ตาราง Day shift ของเดือนที่ผ่านมา มาทำการปรับปรุง/เปลี่ยนแปลงแก้ไขบางส่วน เพื่อนำมาใช้ในเดือนปัจจุบัน

มหาวิทยาลัยบูรพา Burapha University

ภายหลังจากที่เราได้ทำการกำหนดตารางเวลาการทำงานของรถประจำทางแล้ว (Vanitchakornpong et al., 2008) เราจะได้ข้อมูลที่จะกำหนดการทำงานของพนักงาน ดังแสดงในภาพที่ 4.5

No.	Job	Line	Bus Code	Service Type
1	04:25 - 05:00	4	4-44089	Air conditioned
2	04:10 - 05:40	134	4-80227	Normal
3	04:20 - 05:20	4	4-40451	Normal
4	04:25 - 05:35	205	4-44086	Air conditioned
.				
N				

ภาพที่ 4.5 Scheduled trips (jobs)

ภาพที่ 4.5 แสดงข้อมูลการเดินรถประจำทาง ยกตัวอย่างเช่น Job no. 1 รถประจำทางสาย 4 หมายเลข 4-44089 ประเภท Air-conditioned เริ่มต้นงานออกจาก Depot เวลา 4.25 น. และกลับเข้า Depot เวลา 5.00 น.

ในบทนี้ เราพัฒนาขั้นตอนวิธีทางคอมพิวเตอร์เพื่อใช้ในการแก้ปัญหา CSP ของ ขสมก. โดยมีวัตถุประสงค์เพื่อจัดตารางเวลาการทำงานของพนักงานเดินรถให้มีต้นทุนในการดำเนินงานน้อยที่สุด (นั่นคือ จำนวนพนักงาน การจ่ายค่าล่วงเวลา) โดยอยู่ภายใต้เงื่อนไขในการทำงานต่าง ๆ ดังที่กล่าวมาข้างต้น

ขั้นตอนการแก้ปัญหา

เนื่องจาก CSP เป็นปัญหาการตัดสินใจที่มีความซับซ้อนและมีขนาดใหญ่ โดยเฉพาะอย่างเงื่อนไขการทำงานที่ปัจจัยที่เกี่ยวกับคน (Human factors) ซึ่งอาจจะถูกกำหนดเป็นทั้งเงื่อนไขรอง (Soft constraints) จำนวนหลาย ๆ ตัวโดยมีระดับความพอใจ หรือความสำคัญ (Preference) ที่แตกต่างกัน ดังนั้น การสร้างแบบจำลองและพัฒนาอัลกอริทึมเพื่อใช้ในการแก้ปัญหา จึงต้องถูกทำอย่างละเอียดรอบและพิถีพิถัน สำหรับปัญหา CSP ในขั้นตอนแรก เราอาจจะต้องตัดสินใจว่า การวางแผนในขั้นตอนใดควรจะให้เป็น Auto. Semi-auto หรือ Manual และอาจจะต้องพิจารณาว่าขั้นตอนใดควรจะแยก หรือว่ารวมกันยกตัวอย่างเช่น ปัญหา CSP อาจจะแบ่งโครงสร้างการตัดสินใจออกเป็น 2 ระดับได้แก่ Crew pairing และ Crew rostering เพื่อลดขนาดความซับซ้อนของปัญหา และเพื่อให้เกิดความยืดหยุ่นในทางปฏิบัติ อย่างมีประสิทธิภาพ ในงานวิจัยนี้ เราจำลองปัญหา CSP ให้อยู่ในรูป Set partitioning ดังแสดงในภาพที่ 4.6

มหาวิทยาลัยบูรพา Burapha University

Timetable trips			L1		L2	
			K11	K12	K21	K22
			c1	c2	c3	c4
L1	K11	8:00	1	0	0	0
L2	K21	8:15	0	0	1	0
L1	K12	8:30	0	1	0	0
L2	K22	8:35	0	0	0	1
L1	K13	9:00	1	0	0	0

ภาพที่ 4.6 แบบจำลอง Set partitioning

ภาพที่ 4.6 แสดงตัวอย่างปัญหา CSP โดยกำหนดให้มีงาน (Timetable trips) จำนวน 5 งาน โดยแบ่งออกเป็นสาย (Bus line) L1 และ L2 รถประจำทาง K11 ถูกกำหนดให้บริการที่เวลา 8:00 น. และรถประจำทาง K12 ให้บริการที่เวลา 8.30 น. เป็นต้น ในส่วนของพนักงานเดินรถแทนด้วยสัญลักษณ์ c1, c2, c3 และ c4 โดย c1, c2 สังกัดสาย L1 และ c1 นั้นมียานพาหนะประจำ (Priority bus) คือ K11 เป็นต้น เรากำหนดให้ ถ้าพนักงานทำงาน Timetable trip นั้นๆ มีค่าเท่ากับ 1 ยกตัวอย่างเช่น พนักงาน c1 ขับรถประจำทาง K11 (ซึ่งใช้ Priority bus ของตนเอง) ทำงานที่เวลา 8:00 น. และขับรถประจำทาง K13 ทำงานที่ 9:00 น. (โดยไม่ใช้ Priority bus ของตน) เป็นต้น เรากำหนดตัวแปรและพารามิเตอร์ในแบบจำลอง ดังนี้

$L = \{1,....,l\}$	เซตของสายรถประจำทาง
$W = \{1,...., w\}$	เซตของงาน
$W_l' = \{1,...., w_l\}$	เซตของงานสังกัดสาย l
$M = \{1,...., m\}$	เซตของพนักงานขับรถ
$M_l' = \{1,...., m_l\}$	เซตของพนักงานที่สังกัดสาย l
$O_i = \{1,...., o_i\}$	เซตลำดับการทำงานของพนักงาน i
m	จำนวนพนักงาน
Q_i	จำนวนงานที่พนักงานสามารถทำงานได้
c_{ij}	ค่าใช้จ่ายของพนักงาน i ในการทำงาน j
x_{ij}	เท่ากับ 1 เมื่อพนักงาน i ทำงาน j ; 0 ในกรณีอื่น ๆ
e_{ij}	เวลาเริ่มงานลำดับที่ j ของพนักงาน i
t_{ij}	ระยะเวลาในการทำงานของพนักงาน i ในลำดับงานที่ j
s_l	จำนวนพนักงานที่ใช้ใน l
$\{a_i, b_i\}$	พนักงาน i เริ่มงานที่เวลา a_i และเลิกงานที่เวลา b_i

มหาวิทยาลัยบูรพา Burapha University

Hard constraints

ในงานวิจัยนี้ เราพิจารณาเงื่อนไขหลัก (Hard constraints) ของพนักงานขับรถเท่านั้น เนื่องจากเรากำหนดให้พนักเก็บเสตางค์ทำงานคู่กับพนักงานขับรถคนใดคนหนึ่ง (Pre-specified crew pairs) เพื่อลดขนาดของปัญหา ดังนั้น เราพิจารณาเงื่อนไขหลักในการทำงานของพนักงานขับรถดังนี้

$$h_p = \max\left(0, \sum_{j \in W}\left|\sum_{i \subset M} x_{ij} - 1\right|\right) \tag{4.11}$$

เงื่อนไข (4.11) กำหนดให้พนักงาน i ทำงาน j เพียงครั้งเดียว นั่นคือ ผลรวมของพนักงานทุกคนที่ทำงาน j ต้องมีค่าเท่ากับ 1 และบังคับใช้กับทุกงาน

$$h_c = \sum_{i \in M}\left[\max\left(0, \sum_{j \in O_i}\left[\left(e_{i,j-1} + t_{i,j-1}\right) - e_{ij}\right]\right)\right] \tag{4.12}$$

เงื่อนไข (4.12) กำหนดให้เวลาการทำงานเสร็จสิ้นก่อนเวลาในการเริ่มงานปัจจุบัน นั่นคือ ผลรวมของเวลาเริ่มงานและเวลาในการทำงานก่อนหน้าจะต้องน้อยกว่าเวลาเริ่มงานปัจจุบัน

$$h_a = \sum_{i \in M}\left[\max\left(0, \sum_{j \in O_i}\left(a_i - e_{ij}\right)\right)\right] \tag{4.13}$$

เงื่อนไข (4.13) กำหนดให้พนักงานสามารถปฏิบัติงานใด ๆ ได้หลังจากเวลาเริ่มงานของพนักงาน

$$h_b = \sum_{i \in M}\left[\max\left(0, \sum_{j \in O_i}\left(e_{ij} - b_i\right)\right)\right] \tag{4.14}$$

เงื่อนไข (4.14) กำหนดให้พนักงานไม่สามารถปฏิบัติงานนอกเวลาทำงานของพนักงานได้

$$h_q = \sum_{i \in M}\left[\max\left(0, \sum_{j \in W} x_{ij} - Q_i\right)\right] \tag{4.15}$$

มหาวิทยาลัยบูรพา Burapha University

เงื่อนไข (4.15) กำหนดให้พนักงานแต่ละคนปฏิบัติงานได้ไม่เกินจำนวนงานที่กำหนดไว้

Soft constraints

เงื่อนไขรอง (Soft constraints) นั้นมีลักษณะเดียวกับค่าฟังก์ชันวัตถุประสงค์ (Objective function value) โดยเราคำนวณหาค่าความขัดแย้ง (Soft violation) เพื่อชี้วัดประสิทธิภาพของคำตอบที่ได้ ดังนี้

$$v_c = \max\left(0, \sum_{j\in H}\sum_{i\in M} c_{ij}x_{ij}\right) \tag{4.16}$$

สมการ (4.16) คือ ค่าความแย้งที่เกิดจากผลรวมของค่าใช้จ่ายของพนักงานทุกคนในการทำงานทั้งหมด

$$v_n = \sum_{i\in M}\left(\max\left(0, n_i\right)\right) \tag{4.17}$$

สมการ (4.17) คือ ค่าความขัดแย้งที่เกิดจากค่าใช้จ่ายของจำนวนพนักงานในการทำงาน, n_i เท่ากับ 1 เมื่อ $\sum_{j\in H} x_{ij} > 0$ หรือพนักงาน i ปฏิบัติงานตั้งแต่หนึ่งงานขึ้นไป และเท่ากับ 0 ในกรณีอื่น

$$v_r = \max\left(0, \sum_{j\in H}\sum_{i\in M} v_{ij}x_{ij}\right) \tag{4.18}$$

สมการ (4.18) คือ ค่าความขัดแย้งที่เกิดจากพนักงานทำงานโดยไม่ใช้รถประจำทางที่ตนพอใจ (Priority bus), v_{ij} ค่าความขัดแย้งของรถ (Priority bus) เมื่อพนักงาน i ขับรถที่ให้บริการงานที่ j

$$v_p = \max\left(0, \sum_{j\in H}\sum_{i\in M} p_{ij}x_{ij}\right) \tag{4.19}$$

สมการ (4.19) คือ ค่าความขัดแย้งที่เกิดจากความถนัดในเส้นทางการเดินรถ (Priority line) p_{ij} ค่าความขัดแย้งความถนัดสายรถ (Priority line) ของพนักงาน i สายรถของงานที่ j

$$v_h = \sum_{i\in M}\left[\max\left(0, t_{io} - h_i\right)\right] \tag{4.20}$$

มหาวิทยาลัยบูรพา Burapha University

สมการ (4.20) คือ ค่าความขัดแย้งที่เกิดจากค่าใช้จ่ายในการทำงานล่วงเวลาของพนักงาน เนื่องจากเงื่อนไขหลักบังคับให้พนักงานไม่สามารถเริ่มงานใหม่ได้เมื่อเกินเวลาทำงาน ค่าใช้จ่ายนี้จึงอาจเกิดขึ้นโดยที่การต่อช่วงเวลาทำงานนั้นอยู่ภายในเวลาทำงานมาตรฐาน (Standard working hours เช่น 8 ชั่วโมงต่อวัน) แต่มีเวลาสิ้นสุดงานเกินเวลาทำงานมาตรฐานที่กำหนด

$$v_a = \max\left(0, \sum_{i \in M} \left|\left(avg - k_i\right)\right|\right) \tag{4.21}$$

สมการ (4.21) คือ ค่าความขัดแย้งที่เกิดจากความแตกต่างของเวลาการทำงานของพนักงานแต่ละคน (นั่นคือ จำนวนชั่วโมงการทำงาน หรือจำนวนเที่ยวการเดินรถ) เพื่อทำให้พนักงานทุกคนมีเวลาทำงานใกล้เคียงกัน $avg = \left(\sum_{i \in K} t_i\right) / m'$ คือ ค่าเฉลี่ยเวลาทำงานของพนักงานแต่ละคน m' คือ จำนวนพนักงานที่ใช้งานจริง k_i ระยะเวลาทำงานจริง (Actual hours) ของพนักงาน i

จากสมการ (4.11) - (4.15) จะสามารถคำนวณหาค่าความขัดแย้งของเงื่อนไขหลักได้จาก $H = h_p + h_c + h_a + h_b + h_q$ และจากสมการ (4.16) - (4.21) จะได้ค่าความขัดแย้งของเงื่อนไขรอง $S = v_c + v_n + v_r + v_p + v_b + v_a$ และสามารถคำนวณหาค่าความขัดแย้งจากเงื่อนไขทั้งหมดได้จาก $V = H + S$

เราได้ประยุกต์ใช้วิธี Constrained local search (CLS) ในการแก้ปัญหาการจัดตารางเวลาการทำงานของพนักงาน โดยอธิบายถึงขั้นตอนของวิธีการหาคำตอบดังนี้

Sequencing feasible trip

ในการหาคำตอบของแบบจำลองของปัญหา CSP ที่สร้างขึ้น เริ่มต้นจากการกำหนดลำดับการทำงานของพนักงานเพื่อจะแสดงว่าพนักงานคนใดต้องทำงานใดก่อน ต่อจากนั้นจะต้องทำงานใดต่อไป เนื่องจากในงานวิจัยนี้ เราพิจารณาปัญหา CSP ซึ่งเกิดขึ้นหลังจากขั้นตอนการจัดตารางเดินรถประจำทางแล้ว (Bus scheduling process) เราจึงสามารถ Pre-process โดยเรียงลำดับงานจากเวลาเริ่มงานจากน้อยไปหามาก นั่นคือ $e'_{j-1} < e'_j \ \forall j \in W$ เมื่อ e'_j คือ เวลาเริ่มงานของงาน j เมื่อเรียงลำดับแล้ว

มหาวิทยาลัยบูรพา Burapha University

Constrained local search

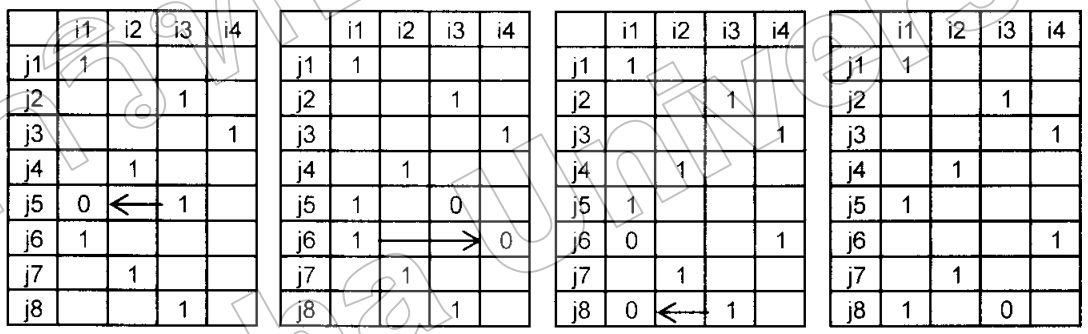
ขั้นตอนการทำงานของวิธี CLS สรุปได้ดังนี้

Step 1: Constraint selection

หลังจากกำหนดค่าเริ่มต้นให้กับตัวแปรทุกตัวในแบบจำลอง อัลกอริทึมจะสุ่มเลือกคอลัมน์ (พนักงาน)ขึ้นมา โดยในการเลือกจะให้ลำดับความสำคัญกับคอลัมน์ที่มีค่าความขัดแย้งของเงื่อนไขหลัก (Hard violation, H) ก่อน

Step 2: Variable selection

จากคอลัมน์หรือเงื่อนไข (Constraints) ที่ถูกเลือกในขั้นตอนที่ (1) อัลกอริทึมจะสุ่มเลือกตัวแปรที่มีค่าเท่ากับ 1 ขึ้นมา (นั่นคือ พนักงานถูกกำหนดให้ทำงานนั้น ๆ) จากนั้น อัลกอริทึมจะทำการทดลองสลับค่าตัวแปรในแถวเดียวกันกับคอลัมน์อื่น ตัวอย่างวิธีการทดลองสลับค่าตัวแปรแสดงดังภาพที่ 4.7



ภาพที่ 4.7 Variable flip operations

จากภาพที่ 4.7 ก. แสดงถึงตัวอย่างการสลับค่าตัวแปร โดยจากตัวอย่างสุ่มได้พนักงาน i3 และทำการสุ่มงานทั้งหมดที่พนักงาน i3 ได้ปฏิบัติอยู่ ได้งาน j5 จากนั้นทำการทดลองสลับค่ากับพนักงาน i1 จากงาน j5 ของพนักงาน i3 จากเดิมเป็น 1 ให้เป็น 0 และงาน j5 ของพนักงาน i1 เดิมจาก 0 ให้เป็น 1 นั่นคือการย้ายงาน j5 จากเดิมที่พนักงาน i3 เป็นผู้ปฏิบัติไปเป็นพนักงาน i1 เป็นผู้ปฏิบัติแทนจากนั้นลองสลับไปกับพนักงานอื่นทั้งหมด

Step 3: Move acceptance

เลือกการทดลองสลับค่า (Trial flips) ที่มีค่าความขัดแย้ง (Total violation, V) น้อยที่สุดและกำหนดให้เป็นคำตอบปัจจุบัน ดังภาพที่ 4.7 ก. การสลับงาน j5 จากพนักงาน i3 ไปยัง i1 มีค่า

มหาวิทยาลัยบูรพา Burapha University

ความขัดแย้งน้อยที่สุดจะทำการกำหนดเป็นคำตอบปัจจุบัน ดังภาพ 4.7 ข. โดยรอบทำงานต่อไปจะทำซ้ำเช่นเดียว ดังภาพ 4.7 ข. ค. ง. แสดงการย้ายคำตอบในแต่ละรอบการทำงาน

ผลการทดลองและวิเคราะห์ผลการทดลอง

ในหัวข้อนี้ เป็นการทดสอบแบบจำลองและอัลกอริทึมที่พัฒนาขึ้นโดยประยุกต์ใช้กับปัญหา CSP ขององค์การขนส่งมวลชนกรุงเทพ (BMTA) เพื่อความสะดวกในการอธิบาย เราแบ่งเงื่อนไขรอง (หรือฟังก์ชันวัตถุประสงค์) ออกเป็น 3 กลุ่มคือ 1) Pre-specified 2) Crew preference และ 3) Schedule cost ดังแสดงในตารางที่ 4.2

ตารางที่ 4.2 Soft violation scheme

No.	Soft constraints	Vilolation
	Pre-specified	
A-1	Crew pair	0
A-2	#Break	0
A-3	Break day	0
A-4	Shift type	0
A-5	Sing-in/Sign-off time	0
	Crew preference	
B-1	Priority bus	5
B-2	Priority line (4 levels)	0,5,15,30
B-3	Duty spreadover	1
	Schedule cost	
C-1	#Crew	500
C-2	Duty operating cost (DOC)/min	1.05
C-3	Overtime (OT)/min	5

ตารางที่ 4.2 เงื่อนไขรองทุกตัวในกลุ่ม Pre-specified มีค่าความขัดแย้ง (Violation) เท่ากับ 0 หรือกล่าวอีกนัยหนึ่ง เงื่อนไขในกลุ่มนี้ได้แก่ การจับคู่ในการทำงานระหว่างพนักงานขับรถและพนักงานเก็บสตางค์ (Crew pair) จำนวนวันหยุดประจำเดือน (#Break) วันหยุดประจำสัปดาห์ (Break day) ประเภทของกะการทำงาน (Shift type) และเวลาเริ่มต้นและสิ้นสุดการทำงานในแต่ละวัน (Sign-in/ Sign-off time) ได้ถูกกำหนดค่าไว้ล่วงหน้าแล้วโดยนักวางแผน เพื่อให้เกิดความยืดหยุ่นในการปฏิบัติงาน และลดพื้นที่ในการค้นหาคำตอบของอัลกอริทึม เงื่อนไขในกลุ่มที่ 2 แสดงค่าความขัดแย้งที่เกิดจากความชอบในการใช้รถประจำทางที่กำหนด (Priority bus) เส้นทางที่พนักงานมีทักษะในการขับขี่ หรือเป็นสายประจำ (Priority line) โดยที่ค่าความขัดแย้งจาก Priority line จะถูกแบ่งออกเป็น 4 ระดับ นั่นคือ รถประจำทางสายแรกนั้นจะมีค่าความขัดแย้งเป็นศูนย์ และสายต่อมามีค่าความขัดแย้งเป็น 5 และ 15 ตามลำดับ และนอกเหนือ 3 สายแรกแล้วทั้งหมดจะมีค่าความขัดแย้งเท่ากันทั้งหมดคือ 30 เงื่อนไขรองในกลุ่มที่ 2 ยังแสดงค่าความขัดแย้งที่เกิดจากความเท่าเทียมกันของจำนวนเที่ยวการทำงาน หรือจำนวนชั่วโมงการทำงานของพนักงานในแต่ละวัน ซึ่ง

มหาวิทยาลัยบูรพา Burapha University

เงื่อนไขรองแต่ละตัวในกลุ่มที่ 2 นี้อาจจะถูกกำหนดระดับความสำคัญ (ค่าความขัดแย้ง) ที่แตกต่างกันตามนโยบายของบริษัท

เงื่อนไขรองในกลุ่มสุดท้าย (Schedule cost) แสดงค่าความขัดแย้งที่เกิดจากค่าใช้จ่ายในการทำงานของพนักงาน และสามารถกำหนดค่าได้อย่างตรงไปตรงมา ได้แก่ ค่าใช้จ่ายที่เกิดจากการจ้างพนักงานต่อวัน (#Crew) ค่าใช้จ่ายที่เกิดจากการทำงานของพนักงานต่อนาที (Duty operation cost: DOC) และค่าใช้จ่ายที่เกิดจากค่าล่วงเวลาต่อนาที (Overtime)

จากผลการรันโปรแกรม เราแสดงตัวอย่างค่าดัชนีชี้วัดประสิทธิภาพของตารางการทำงานของพนักงานของรถประจำทางสาย 4 (Scheduling KPI) ดังตารางที่ 4.3 ผู้ให้บริการขนส่งรถประจำทางสาธารณะ (BMTA) สามารถปรับเปลี่ยนค่าพารามิเตอร์ในแบบจำลองเพื่อประเมินผลกระทบที่เกิดขึ้นได้

ตารางที่ 4.3 Crew scheduling KPI

Scheduling Plan	KPI
Crews	25
Line (s)	1
Trips	72
Preference	
> Priority bus	330
> Priority line	0
> Duty spreadover	1548
Operations	
> Idle time (hh:mm)	110:00
> Actual time (hh:mm)	66:55
> Extra time (hh:mm)	0:10
Cost (Baht)	
> Duty	11,057.29
> Overtime	10.42

เราประเมินประสิทธิภาพของแบบจำลองที่พัฒนาขึ้น โดยเปรียบเทียบกับตารางการทำงานของพนักงานซึ่งอาศัยประสบการณ์และความเคยชินของนักวางแผน (Current) เราแสดงตัวอย่างการทดสอบการจัดตารางการทำงานของพนักงานของรถประจำทางจำนวน 3 สาย (ดูตารางที่ 4.4 ประกอบ) ซึ่งในปัจจุบัน (Current) มีจำนวนพนักงานที่สังกัด 3 สายดังกล่าวทั้งหมด 120 คน โดยปราศจากดัชนีชี้วัดประสิทธิภาพใด ๆ จากนั้น เราใช้อัลกอริทึมที่พัฒนาขึ้นในการหาคำตอบ โดยกำหนดให้พนักงานทุกคนอยู่ในสายรถประจำทางที่ตนสังกัด (Single Line) จะสามารถลดจำนวนพนักงาน (#Crew) จากเดิม 120 คน เหลือ 84 คน รวมทั้งแสดงค่าความพอใจของพนักงาน (Preference) เวลาที่ใช้ในการทำงาน (Operations) และค่าใช้จ่าย (Cost) ในการดำเนินงาน (อ้างอิง

ตารางที่ 4.3) ตัวเลขในวงเล็บ แสดงค่าเบี่ยงเบนมาตรฐาน (SD) ของการรันโปรแกรมจำนวน 10 ครั้ง และ Time เป็นเวลาที่อัลกอริทึมใช้ในการหาคำตอบ มีหน่วยเป็นวินาที (sec)

ตารางที่ 4.4 Multi-skilled policy (test for 10 runs)

Scenario	Trips	#Crew	Preference			Cost		Time
			1	2	3	OT	DOC	(sec)
Current		120						
Single Line	253	84	1235	0	4836	957	38,024	19.218
			(86)	(0)	(67.76)	(13.46)	(160.52)	
2-Line	253	84	1255	840	4904	618	38,305	29.781
			(44)	(8.57)	(50.53)	(13.63)	(132.31)	
3-Line	253	82	1255	2490	3922	577	37,726	39.421
			(52)	(20.26)	(34.76)	(11.33)	(105.38)	
	Saving (%)	2.38				39.72	0.78	

นอกจากนี้ เรายังทดสอบนโยบายในการใช้พนักงานร่วมกันในแต่สาย หรือกล่าวอีกนัยหนึ่ง พนักงานสามารถขับรถประจำทางได้มากกว่า 1 สาย (Multi-skilled policy) จากผลการทดสอบชี้ให้เห็นว่า เราสามารถลดจำนวนพนักงานลงได้อีก 2 คน และลดค่าใช้จ่ายในการดำเนินงานลงได้อีกเพียงเล็กน้อย เนื่องจากว่าอัลกอริทึมหาคำตอบของกรณี Single Line ด้วยต้นทุนต่ำที่สุดอยู่แล้ว รวมทั้งในขั้นตอนการจัดตารางเดินรถ (Bus scheduling process) นโยบายการวิ่งสลับสาย (Mixed line operations) เมื่อลดจำนวนรถประจำทางที่ใช้งาน ก็จะส่งผลให้ลดจำนวนพนักงานด้วยเช่นกัน

ในบทนี้ เราประยุกต์ใช้วิธี CLS ในการจัดตารางการทำงานของพนักงานขับรถประจำทาง โดยเปรียบเทียบกับวิธีที่ใช้ในปัจจุบัน ซึ่งจากผลการทดสอบแสดงให้เห็นว่าวิธีที่นำเสนอสามารถช่วยให้บริษัทลดจำนวนพนักงาน และต้นทุนในการดำเนินการลงได้ อย่างไรก็ตาม การพิจารณาปัจจัยทางด้าน Human factors โดยการกำหนดค่าความขัดแย้งสำหรับ เงื่อนไขรองยังกระทำได้ยาก เพราะในบางกรณี คำตอบที่ได้อาจจะมีเงื่อนไขรองบางตัวที่ให้คำตอบที่มีคุณภาพไม่ดี แล้วอาจทำให้ตารางเวลาการทำงานที่สร้างขึ้นไม่สามารถนำไปใช้งานได้โดยทันที โดยนักวางแผนอาจจะต้องปรับเปลี่ยนค่าพารามิเตอร์ (Violation scheme) และรันโปรแกรมทดสอบอีกจำนวนหลายครั้งเพื่อเปรียบเทียบผล และเลือกคำตอบที่ต้องการ

มหาวิทยาลัยบูรพา Burapha University

สรุป

วิธี CLS นั้นเป็นวิธีการแก้ปัญหาการตัดสินใจไม่ต่อเนื่องที่มีความยืดหยุ่น โดยการกำหนดฟังก์ชันวัตถุประสงค์และเงื่อนไขให้อยู่ในรูปค่าความขัดแย้ง และทำการเคลื่อนย้ายพื้นที่ในการหาคำตอบเพื่อเปรียบเทียบค่าความขัดแย้งระหว่างคำตอบ เมื่อสามารถกำหนดปัญหาให้อยู่ในรูปของ set partitioning ได้ อย่างไรก็ตามวิธี CLS ถือว่าเป็นวิธีการแก้ปัญหาหนึ่งในกลุ่มของวิธีฮิวริสติกส์ ซึ่งไม่สามารถรับประกันคำตอบ หรือว่าเสถียรภาพของคำตอบได้ แต่จากการทดลองข้างต้นจะเห็นว่าการวิธี CLS นั้นใช้เวลาในการหาคำตอบได้อย่างรวดเร็ว ดังนั้นวิธี CLS นั้นจึงเหมาะสำหรับปัญหาที่ต้องหาคำตอบที่ดีพอในระยะเวลาที่จำกัด และปัญหาที่มีการเปลี่ยนแปลงเงื่อนไขเสมอ และจากการทดลองข้างต้นนั้น จะเห็นได้ว่าสามารถประยุกต์ใช้วิธี CLS แก้ปัญหาในโลกความเป็นจริงได้

มหาวิทยาลัยบูรพา Burapha University