

## A dynamic set partitioning-based model for the vehicle routing problem

Kriangsak Vanitchakornpong<sup>1\*</sup>, Nakorn Indra-Payoong<sup>2</sup>, Krisana Chinnasarn<sup>1</sup>

<sup>1</sup> Department of Computer Science, Faculty of Science, Burapha University

{\*kriangsv, krisana}@buu.ac.th

<sup>2</sup> College of Transport and Logistics, Burapha University

nakorn@buu.ac.th

### Abstract

This paper aims to present model and algorithm for a classical set partitioning problem. The vehicle routing problem can be formulated as a dynamic set partitioning based model in which the lower bound on the objective function value cannot be known in advance. We propose the branch-and-bound and local search methods for solving this class of problem. The computational results have demonstrated that the proposed model and methods are simple, practical and efficient by solving a real-world vehicle routing problem for car carrier company.

**Key-Words:** Set partitioning model; Branch-and-bound method; Local search

### 1. Introduction

การจัดกลุ่มของตัวแปรให้สอดคล้องกับเงื่อนไขแบบเซต (Set partitioning constraints) นั้นเป็นปัญหาการตัดสินใจที่มีความสำคัญซึ่งมีอยู่ในภาคอุตสาหกรรมต่างๆจำนวนมาก ยกตัวอย่างเช่น ปัญหาการมอบหมายงานให้กับพนักงาน (Job assignment problem) วัตถุประสงค์ของปัญหาคือ การกำหนดให้พนักงานทำงานเพื่อให้มีประสิทธิภาพ นั่นคือ การเลือกงานใดงานหนึ่งให้คนๆหนึ่งเพื่อจะให้เสียค่าใช้จ่ายน้อยที่สุด ภายใต้เงื่อนไขว่า พนักงาน 1 คนสามารถทำงาน 1 ชิ้นเท่านั้น ภายในช่วงเวลาที่กำหนด

ปัญหาการจัดเส้นทางยานพาหนะ (Vehicle routing problem: VRP) นั้นเป็นปัญหาโลจิสติกส์ที่มีความสำคัญ โดยนักวางแผนการขนส่งจะต้องตัดสินใจว่าจะให้ยานพาหนะหรือรถบรรทุกคันใดไปให้บริการขนส่งสินค้ากับลูกค้ารายใด โดยที่ลูกค้าแต่ละรายมีความต้องการในการขนส่งที่แตกต่างกันไป อาทิเช่น เงื่อนไขของเวลาในการขนส่ง (Pickup and delivery time windows) เป็นต้น นักวางแผนจะต้องทำการกำหนดลำดับการขนส่งสินค้าเพื่อให้เสียค่าใช้จ่ายในการขนส่งน้อยที่สุด ในงานวิจัยนี้ได้นำลักษณะเงื่อนไขแบบเซต มาประยุกต์ใช้ในการสร้างแบบจำลองการตัดสินใจของปัญหาการจัดเส้นทางยานพาหนะของบริษัทขนส่งรถยนต์รายหนึ่ง และได้พัฒนาอัลกอริทึมเพื่อใช้ในการหาคำตอบของปัญหาดังกล่าว

### 2. Literature Review

ปัญหาการตัดสินใจที่มีเงื่อนไขแบบเซตมีความซับซ้อน เมื่อปัญหาประกอบด้วยตัวแปรในการตัดสินใจจำนวนมาก เราอาจไม่สามารถหาคำตอบที่ดีที่สุดได้ ซึ่งเราเรียกปัญหาในลักษณะนี้ว่า ปัญหา NP-hard [1] โดยทั่วไป เราสามารถแบ่งอัลกอริทึมที่ใช้ในการหาคำตอบของปัญหาการตัดสินใจออกเป็น 2 ประเภทคือ 1)

วิธีการหาคำตอบที่ดีที่สุด (Exact หรือ optimization method) เช่น วิธีการแก้ปัญหาที่อาศัยโครงสร้างการตัดสินใจแบบต้นไม้ (Branch and bound: BnB) วิธี BnB นี้สามารถประกันการหาคำตอบที่ดีที่สุด (Optimal solution) ได้ในทุกๆ ครั้งของการรันโปรแกรม อย่างไรก็ตาม เมื่อปัญหาที่มีความซับซ้อน มีเงื่อนไขและตัวแปรการตัดสินใจจำนวนมาก วิธีการหาคำตอบที่ดีที่สุด อาจจะไม่เหมาะสมเพราะอัลกอริทึมจะใช้เวลานานในการหาคำตอบนานเกินไป หรืออาจไม่สามารถหาคำตอบที่ดีที่สุดได้ ภายใต้ทรัพยากรของเครื่องคอมพิวเตอร์ที่จำกัด

2) วิธีฮิวริสติกส์ (Heuristics) วิธีนี้สามารถหาคำตอบที่ดี ภายในระยะเวลาอันรวดเร็ว อย่างไรก็ตาม ข้อเสียของวิธีนี้คือ อัลกอริทึมไม่สามารถประกันการหาคำตอบที่ดีที่สุดได้ เช่น วิธีทางพันธุกรรม (Genetic algorithm: GA) หรือวิธีอื่นๆที่อยู่ในสาขาปัญญาประดิษฐ์ (Artificial intelligence: AI) เป็นต้น

การจัดเส้นทางยานพาหนะเป็นปัญหาการตัดสินใจที่มีความซับซ้อนและยากต่อการหาคำตอบที่ดีที่สุด โดยทั่วไป ปัญหาการจัดเส้นทางยานพาหนะอาศัยวิธีฮิวริสติกส์มาใช้แก้ปัญหา อาทิเช่น วิธี Saving ที่นำเสนอโดย Clarke และ Wright [2] หรือวิธีเมตาฮิวริสติกส์ (Metaheuristics) ที่มีโครงสร้างยืดหยุ่นกว่าวิธีฮิวริสติกส์ เช่น วิธีทาบู (Tabu search) ซึ่งนำเสนอโดย Osman [3] งานวิจัยของ Kelly และ Xu [4] นำเงื่อนไขแบบเซตมาสร้างแบบจำลองการตัดสินใจของปัญหาการจัดเส้นทางยานพาหนะ โดยแบบจำลองพิจารณาว่า จะต้องเลือกสินค้าหรือลูกค้ารายใดไปกับรถบรรทุกคันใด นอกจากนี้ ในทางปฏิบัติ เมื่อเราพิจารณาเงื่อนไขของลำดับการส่งสินค้าจะทำให้ปัญหาการตัดสินใจมีความซับซ้อนมากขึ้น

โดยทั่วไป การพัฒนาอัลกอริทึมเพื่อใช้ในการแก้ปัญหาการจัดเส้นทางยานพาหนะจะใช้วิธีฮิวริสติกส์ วิธีดังกล่าวนี้อาจจะให้คำตอบในการจัดกลุ่มของลูกค้าที่มีระยะทางไกลกันมารวมอยู่ในกลุ่มเดียวกันได้ ยกตัวอย่างเช่น รถบรรทุกคันหนึ่งไปรับสินค้าจากลูกค้าที่อยู่ในจังหวัดเชียงใหม่และสงขลา ซึ่งไม่สามารถเป็นจริงได้ในทางปฏิบัติ ดังนั้น จึงทำให้วิธีการหาคำตอบที่ดีที่สุดยังคงเป็นที่ต้องการจากภาคอุตสาหกรรมการขนส่งอยู่ โดยที่มันักวิจัยอีกกลุ่มหนึ่งได้กำลังพยายามที่จะปรับปรุงและพัฒนาอัลกอริทึมที่ใช้เวลาในการค้นหาคำตอบที่เร็วขึ้น ขณะเดียวกัน วิธีฮิวริสติกส์ก็ได้ถูกพัฒนาให้มีคุณภาพของคำตอบให้ใกล้เคียงคำตอบที่ดีที่สุดมากขึ้น ดังนั้น ถ้าสินค้าที่จะทำการขนส่งมีจำนวนไม่มาก เราอาจใช้วิธี Exact ในการหาคำตอบที่ดีที่สุด ในทางตรงข้าม เมื่อสินค้าที่จะทำการส่งมีเป็นจำนวนมาก เราอาจไม่สามารถใช้วิธี Exact ได้ เนื่องจากข้อจำกัดด้านเวลาและทรัพยากรคอมพิวเตอร์ที่ใช้ในการหาคำตอบ ดังนั้น วิธีฮิวริสติกส์จึงเป็นทางเลือกหนึ่งที่ได้รับคามนิยม

### 3. Problem Formulation

เราพิจารณาปัญหาการจัดเส้นทางยานพาหนะที่เกิดขึ้นจริงในอุตสาหกรรมการขนส่งรถยนต์ในประเทศไทย การตัดสินใจในการจัดเส้นทางยานพาหนะที่มีประสิทธิภาพช่วยให้บริษัทผู้ประกอบการขนส่งรถยนต์ (Car carrier) สามารถลดต้นทุนการขนส่งได้

#### 3.1 Case Study

เนื่องจากประเทศไทยเป็นแหล่งผลิตรถยนต์ที่มากที่สุดในภูมิภาคเอเชียตะวันออกเฉียงใต้ รวมทั้งมีการผลักดันทางด้านนโยบายจากรัฐบาลทำให้อุตสาหกรรมการผลิตและขนส่งรถยนต์เติบโตขึ้นอย่างรวดเร็ว โดยทั่วไป ปัญหาการจัดเส้นทางรถบรรทุกขนส่งรถยนต์ รถบรรทุกสามารถบรรทุกรถยนต์ (สินค้า) ได้ประมาณ 3-7 คัน ซึ่งจะขึ้นอยู่กับขนาดของรถยนต์ที่ทำการขนส่ง โดยผู้ประกอบการขนส่งจะทำการขนส่งรถยนต์จากโรงงานประกอบรถยนต์ อาทิเช่น บริษัทโตโยต้า นิสสัน และ BMW ไปยังผู้แทนจำหน่ายทั่ว

ประเทศ (Dealer) ปัญหาในการตัดสินใจคือ การจัดเส้นทางขนส่งว่า ในรถบรรทุก (Trailer) แต่ละคันนั้น จะบรรทุกรถยนต์ไปส่งที่ตัวแทนจำหน่าย ไต่บ้าง เพื่อให้ค่าใช้จ่ายในการขนส่งทั้งหมดน้อยที่สุด ในทางปฏิบัติจะมีนักวางแผนการขนส่งที่มีประสบการณ์เป็นผู้รวมกลุ่มรถยนต์ภายในรถบรรทุกแต่ละคัน และกำหนดเส้นทางรถบรรทุกที่จะเดินทางไปยังตัวแทนจำหน่ายรถยนต์ทั่วประเทศ ซึ่งในการทำงานในปัจจุบัน นักวางแผนไม่สามารถบอกได้ว่าผลที่เกิดจากการจัดเส้นทางที่ทำขึ้นนั้น มีประสิทธิภาพมากน้อยเพียงใด เพราะคำตอบที่เป็นไปได้มีเป็นจำนวนมาก นั่นก็คือ ไม่สามารถบอกได้ว่า คำตอบที่ได้จากการตัดสินใจนั้นมีประสิทธิภาพ หรือเป็นคำตอบที่ดีที่สุดได้

ในบทความนี้ เราได้สร้างแบบจำลองการตัดสินใจ และพัฒนาอัลกอริทึมเพื่อใช้ในการหาคำตอบของปัญหาดังกล่าว ซึ่งเราได้แบ่งปัญหาการตัดสินใจออกเป็น 2 ขั้นตอน ได้แก่ 1) ขั้นตอนการรวมกลุ่มสินค้า (รถยนต์) ภายในรถบรรทุก (Item batching) และ 2) ขั้นตอนการจัดลำดับการขนส่งรถยนต์ของรถบรรทุกแต่ละคัน (Discharge sequencing) เนื่องจากรถบรรทุกสามารถบรรทุกรถยนต์ได้มากที่สุด 7 คันเท่านั้น เราจึงสามารถใช้วิธี BnB หรือแม้กระทั่งวิธี Enumerative search มาใช้ในการหาคำตอบที่ดีที่สุดได้อย่างมีประสิทธิภาพ ดังนั้น ในบทความนี้ เราจึงไม่พิจารณาและกล่าวถึงรายละเอียดของขั้นตอนการจัดลำดับการขนส่งสินค้า (รถยนต์)

### 3.2 Mathematical Model

เราสร้างตารางจุดกำเนิด/จุดปลายทาง (O/D matrix) สำหรับเก็บข้อมูลระยะทางและเวลาจากจุดรับและจุดส่งสินค้าเพื่อใช้ในการคำนวณหาระยะทาง และตรวจสอบเงื่อนไขด้านเวลาในการขนส่ง เราสร้างแบบจำลองทางคณิตศาสตร์เพื่อจำลองลักษณะของปัญหา การรวมกลุ่มรถยนต์ภายในรถบรรทุก เราได้นำลักษณะเงื่อนไขแบบเซตมาสร้างแบบจำลองในการตัดสินใจดังนี้

ฟังก์ชันวัตถุประสงค์

$$\text{Min} \sum_{j=1}^n c_j k_j \quad (1)$$

ภายใต้เงื่อนไข

$$\sum_{i=1}^m x_{ij} = 1; \quad i = 1, 2, 3, \dots, m \quad (2)$$

$$\sum_{j=1}^n x_{ij} \leq b_j; \quad j = 1, 2, 3, \dots, n \quad (3)$$

$$x_{ij} \in \{0, 1\} \quad (4)$$

โดยที่  $x_{ij} = 1$  ถ้าสินค้า  $i$  ถูกจัดให้อยู่ในรถบรรทุก  $j$

$= 0$  ในกรณีอื่นๆ

$k_j$  เป็นสมาชิกของคำตอบที่ไม่ขัดแย้งกับเงื่อนไข,  $b_j$  คือ ความสามารถในการบรรทุกสินค้าของรถบรรทุก  $j$  และ  $c_j$  คือ ระยะทาง (หรือค่าใช้จ่าย) ในการขนส่งของรถบรรทุก  $j$  ซึ่งไม่สามารถทราบค่าได้ล่วงหน้า โดย

ระยะทางในการขนส่งของรถบรรทุก  $j$  สามารถหาได้จากสมการ (5) ภายหลังจากตัวแปร  $x_{ij}$  ทั้งหมดได้ถูกกำหนดค่าแล้ว

$$\sum_{p=1}^{n_j} \sum_{q=1}^{n_j} c_{pq} \quad (5)$$

$c_{pq}$  คือระยะทางจากจุดส่งสินค้า  $p$  ไปยังจุดส่งสินค้า  $q$  โดย  $p$  และ  $q$  เป็นสมาชิกของรถบรรทุก  $j$  ซึ่งก็คือ ระยะทางในการขนส่งของรถบรรทุก  $j$  ดังนั้น จากสมการ (1) เราสามารถเขียนฟังก์ชันวัตถุประสงค์ใหม่ได้ดังสมการ (6)

$$\text{Min} \sum_{j=1}^n \sum_{p=1}^{n_j} \sum_{q=1}^{n_j} c_{pq} k_j \quad (6)$$

#### 4. Solution Method

ในส่วนนี้จะอธิบายถึงขั้นตอนวิธีการหาคำตอบของปัญหาการรวมกลุ่มสินค้า (รถยนต์) ภายในรถบรรทุก ซึ่งขั้นตอนในการหาคำตอบจะถูกแบ่งออกเป็น 3 ส่วนคือ 1) การประมวลผลเบื้องต้น (Preprocessing) 2) โครงสร้างของวิธี BnB และ 3) วิธี Local search

##### 4.1 Preprocessing

เนื่องจากปัญหาการจัดเส้นทางยานพาหนะของบริษัทขนส่งรถยนต์มีตัวแปร  $x_{ij}$  จำนวนมาก ลักษณะของปัญหาอาจประกอบด้วยจุดส่งสินค้ามากถึง 200 จุดในแต่ละวัน (บริษัทผู้แทนจำหน่ายรถยนต์ทั่วประเทศ) โดยทั่วไปแล้ว วิธีการรวมกลุ่มสินค้า จะนำสินค้า (รถยนต์) ที่มีจุดส่งที่เดียวกันบรรทุกไปในรถบรรทุกคันเดียวกัน และจะนำจำนวนรถยนต์ที่เหลือที่มีจุดส่งต่างกันเข้าสู่ขั้นตอนต่อไป)

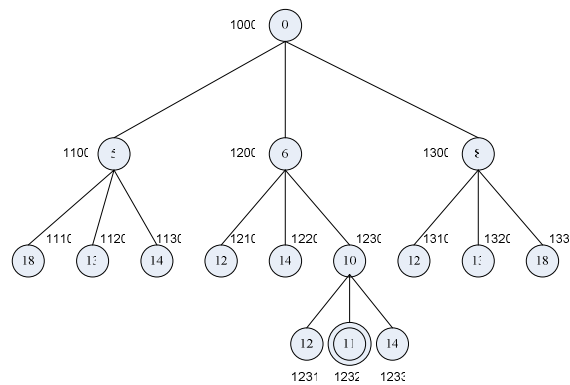
##### 4.2 Branch and bound framework

เราประยุกต์ใช้วิธี BnB โดยอาศัยขีดจำกัดล่าง (Lower bound) ในการตัดกิ่งของต้นไม้ (Node) ที่ไม่มีประสิทธิภาพเพื่อลดพื้นที่ของคำตอบที่เป็นไปได้ (Search space) อย่างไรก็ตาม ในการประยุกต์ใช้เงื่อนไขแบบเซตมาจำลองการตัดสินใจของปัญหาการจัดเส้นทางยานพาหนะนั้น เราไม่สามารถคำนวณหาค่าขีดจำกัดล่างที่ดีโดยใช้ Linear program (LP) relaxation ได้ล่วงหน้าและมีประสิทธิภาพ เพราะว่า  $c_j$  จะเปลี่ยนแปลงไปตามกลุ่มของสินค้า (รถยนต์) ที่ถูกขนส่งโดยรถบรรทุก  $j$  ซึ่งทำให้ปัญหาที่เรากำลังพิจารณานั้นมีความยากมากกว่า ปัญหาการตัดสินใจที่มีเงื่อนไขแบบเซตทั่วไป (Set partitioning problem : SPP) ดังนั้น ในงานวิจัยนี้ เราจึงหาค่าขีดจำกัดล่างของปัญหาโดยการกำหนดค่าเริ่มต้นให้กับตัวแปรตัวแรกก่อน และยังไม่กำหนดค่าให้กับตัวแปรอื่นๆ (Partial assignment) หลังจากนั้น อัลกอริทึมจะค่อยๆ เพิ่มตัวแปร  $x_{ij}$  เข้าไปที่ละตัวในแต่ละรอบการแตกกิ่งของต้นไม้ และคำนวณหาค่าขีดจำกัดล่างได้ ตามขั้นตอนดังนี้

- Node selection: เลือก Node ที่มีค่าขีดจำกัดล่างน้อยที่สุดในทุกๆ ชั้นของต้นไม้
- Variable selection: เลือกตัวแปร  $x_{ij}$  เรียงตามลำดับจุดส่งสินค้า (รถยนต์)  $i = 1, 2, 3, \dots, m$

- Bounding function: สำหรับสินค้า (รถยนต์)ที่ยังไม่ได้ถูกกำหนดให้อยู่ในรถบรรทุก  $j$  เราจะไม่นำมาคิดค่าขีดจำกัดล่าง
- Fathoming: เมื่อ Node ที่กำลังพิจารณาอยู่ให้คำตอบที่ไม่ขัดแย้งกับเงื่อนไข (Feasible node) และสินค้า (รถยนต์)  $i$  ทุกคันต้องถูกกำหนดว่าจะถูกขนส่งโดยรถบรรทุกคันใดลักษณะการทำงานของวิธี BnB แสดงดังรูปที่ 1

ขนาดตารางควรอยู่ในกรอบที่เว้นจากขอบกระดาษแล้ว เพื่อเป็นการง่ายต่อการอ้างถึงตารางต่างๆ ควรใส่หมายเลขและคำอธิบายของตารางทุกๆ ตารางด้วย โดยใช้ตัวเลขโรมันและคำบรรยายเหนือตาราง ให้ใส่เครื่องหมาย “:” ระหว่างเลขที่ตารางกับคำอธิบายด้วย ควรจัดตารางและคำอธิบายไว้กึ่งกลางหน้ากระดาษดังนี้



รูปที่ 1: การแตกกิ่งของวิธี BnB

วิธี BnB มีลำดับการทำงานของดังนี้

- ขั้นที่ 1 กำหนด Root node โดยให้รถบรรทุกคันที่หนึ่งขนส่งสินค้า (รถยนต์) คันที่หนึ่งทันที และเลือก Node นี้เป็น Node ที่ใช้ในการแตกกิ่งต่อไป ในงานวิจัยนี้ เรากำหนดให้รถบรรทุก  $j$  บรรทุกสินค้า (รถยนต์)  $i$  มีค่าใช้จ่ายในการขนส่งเท่ากัน
- ขั้นที่ 2 แตกกิ่งออกจาก Node ที่เลือกโดยกำหนดให้สินค้าอันแรกสุดที่ยังไม่ได้ถูกกำหนดให้อยู่ในรถบรรทุก  $j$
- ขั้นที่ 3 คำนวณค่าขีดจำกัดล่างของ Node ที่แตกออกไปทุก Node โดยใช้สมการ (5) เลือก Node ที่มีค่าขีดจำกัดล่างน้อยที่สุดในทุกๆ ชั้นของต้นไม้
- ขั้นที่ 4 ตรวจสอบเงื่อนไขความสามารถในการบรรทุกสินค้าของรถบรรทุก  $b_j$  และตรวจสอบว่าสินค้า  $i$  ถูกกำหนดให้ขนส่งโดยรถบรรทุก  $j$  แล้วหรือยัง ถ้าคำตอบที่ได้ยังขัดแย้งกับเงื่อนไข ให้ย้อนไปทำขั้นที่ 2 ถ้าคำตอบที่ได้สอดคล้องกับเงื่อนไขจะได้ Node ที่ดีที่สุด

### 4.3 Local search

เมื่อปัญหามีขนาดใหญ่ขึ้น(จุดส่งสินค้าจำนวนมาก) การใช้วิธี BnB ในการหาคำตอบนั้นอาจจะต้องใช้เวลานานมาก ในงานวิจัยนี้เราประยุกต์ใช้วิธี Constraint-based local search (CLS) โดย Indra-Payoong และคณะ [5] สำหรับปัญหาการจัดเส้นทางยานพาหนะที่มีขนาดใหญ่ นั่นคือ จุดส่งสินค้าสามารถมีได้มากถึง 200 จุด วิธี CLS มีขั้นตอนการทำงานแสดงดังรูปที่ 2

```

proc CLS
    input soft and hard constraints
    A := initial assignment
    while not stopping criterion do
        C := select-violated-hard-constraint (A)
        P := select-two-variables (C, A)
        V := total constraint violation
        h := hard violation
        s := soft violation
        A1, A2 := flip (A, P)
        if (V1 < V2) then (A ← A1)
            else (A ← A2)
                if h = 0 then A is feasible, record A
            end if
        end while
    output a feasible solution found, s
end proc
    
```

รูปที่ 2: ขั้นตอนการทำงานของวิธี CLS

รูปที่ 2 เงื่อนไขจะถูกแบ่งออกเป็นสองประเภทคือ เงื่อนไขหลัก (Hard constraint) และเงื่อนไขรอง (Soft constraint) โดยที่เงื่อนไขหลักจะต้องมีค่าความขัดแย้ง (Hard violation)  $h = 0$  และเงื่อนไขรอง อาจจะมีค่าความขัดแย้ง (Soft violation)  $s$  และอัลกอริทึมพยายามลดค่า  $s$  ให้น้อยที่สุด (หรือกล่าวอีกนัยหนึ่ง  $s$  คือระยะทางในการขนส่งทั้งหมด) วิธี CLS เริ่มการทำงานโดยการสุ่มเลือกเงื่อนไขจำนวน 1 ที่ยังมีค่าความขัดแย้งอยู่ (Violated hard constraint,  $h > 0$ ) จากนั้นอัลกอริทึมจะเลือกตัวแปร  $x_{ij}$  จำนวน 2 ตัวใดๆ จากเงื่อนไขที่ถูกเลือกในข้างต้น ซึ่งการเลือกตัวแปรจะเป็นในลักษณะสุ่มแบบไม่หยิบใส่กลับคืน หลังจากนั้นอัลกอริทึมจะทำการสลับค่าของตัวแปรที่ถูกเลือกมา 2 ตัวดังกล่าว (Trial flip) และเลือกตัวแปรที่สลับค่าแล้วมีค่าความขัดแย้งรวม ( $V = h + s$ ) น้อยกว่า เมื่อ  $h = 0$  อัลกอริทึมจะทำการคำนวณและบันทึกค่า  $s$  หลังจากนั้น อัลกอริทึมจะอนุญาตให้คำตอบที่หาได้ในรอบถัดไปสามารถผิดเงื่อนไขอีกได้ หรือทำการเริ่มต้น

การหาคำตอบใหม่โดยการกำหนดค่าให้ตัวแปร (Random assignment,  $A$ ) อัลกอริทึมสิ้นสุดการทำงาน เมื่อครบรอบการกระทำซ้ำ (No improving iteration) ที่กำหนด

## 5. Computation Results

ในส่วนนี้จะแสดงคำตอบของปัญหาการจัดเส้นทางยานพาหนะ โดยใช้อัลกอริทึมที่ได้พัฒนาขึ้น เราได้ทดสอบบนเครื่องคอมพิวเตอร์ Pentium IV ความถี่ในการประมวลผล 2.8 GHz และอัลกอริทึมถูกเขียนด้วยภาษา JAVA ตารางที่ 1 แสดงคำตอบที่ดีที่สุดที่ได้จากวิธี BnB

ตารางที่ 1: คำตอบที่ได้จากวิธี BnB

Customer	Lower bound (Distance KM)	CPU (s)
15	992	0.797
16	1051	1.219
17	1206	3.437
18	1584	53

ตารางที่ 2 (ในภาคผนวก) แสดงคำตอบที่ได้จากวิธี CLS โดยใช้ปัญหาเดียวกันกับตารางที่ 1 จากตารางจะเห็นได้ว่าประสิทธิภาพของวิธี CLS นั้นใกล้เคียงกับวิธี BnB มาก และวิธี CLS ใช้ระยะเวลาในการหาคำตอบน้อยกว่า

ตารางที่ 3 แสดงตัวอย่างการนำเอาวิธี CLS ไปใช้ในการแก้ปัญหาการจัดเส้นทางยานพาหนะที่มีจุดส่งสินค้าจำนวนมากถึง 200 จุด โดยการรันโปรแกรมซ้ำกัน 10 ครั้ง และค่าที่ได้ในตารางนั้นเป็นระยะทางในการขนส่งทั้งหมดที่ได้จากสมการ (5) หรือกล่าวอีกนัยหนึ่งคือ ค่าความขัดแย้ง  $s$  สังเกตด้วยว่า ค่า  $s$  เป็นตัวชี้วัดประสิทธิภาพ (ระยะทางทั้งหมด) ในการรวมกลุ่มสินค้าเท่านั้น ซึ่งคิดจากผลรวมของจุดกำเนิด 1 จุดไปยังจุดปลายทางหลายๆ จุดใดๆ ดังนั้น ค่า  $s$  จึงไม่ใช่ระยะทางในการขนส่งจริง ภายหลังจากขั้นตอนการกลุ่มสินค้า (Transport order batching) นี้แล้ว ผลลัพธ์ดังกล่าวจะถูกนำไปจัดลำดับการขนส่ง (Discharge sequencing) เพื่อที่จะได้ระยะทางที่สั้นที่สุดในการส่งจริงต่อไป

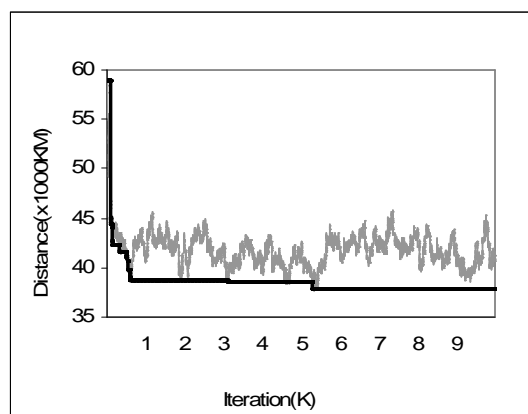
จากตาราง ค่าน้อยที่สุด (Min) ค่าที่มากที่สุด (Max) และค่าเฉลี่ย (Avg.) ของระยะทางในการขนส่งนั้นมีค่าที่ใกล้เคียงกัน ซึ่งชี้ให้เห็นว่า อัลกอริทึมที่พัฒนาขึ้นมีประสิทธิภาพและมีความเสถียรภาพในการหาคำตอบ

ตารางที่ 2: คำตอบที่ได้จากวิธี CLS สำหรับปัญหาที่มีขนาดใหญ่

	Customer	CPU (s)	Avg.	Min	Max
P1	50	0.344	8,173.7	7,795	8,350
P2	100	0.968	18,215	17,460	18,566
P3	150	1.797	28,735	28,208	29,244
P4	200	3.219	37,968	37,014	38,615

ตารางที่ 3: คำตอบที่ได้จากวิธี CLS

Run	15 customer		16 customer		17 customer		18 customer	
No.	S	CPU (s)	S	CPU (s)	S	CPU (s)	S	CPU (s)
1	992	0.141	1051	0.328	1210	0.204	1635	0.265
2	993	0.125	1051	0.140	1210	0.218	1624	0.141
3	992	0.125	1150	0.140	1206	0.172	1645	0.203
4	992	0.187	1051	0.203	1210	0.156	1656	0.203
5	992	0.203	1105	0.171	1206	0.188	1624	0.281
6	1018	0.157	1150	0.141	1210	0.156	1656	0.141
7	992	0.141	1051	0.172	1206	0.125	1635	0.219
8	992	0.156	1107	0.188	1239	0.234	1635	0.297
9	992	0.141	1051	0.234	1206	0.407	1660	0.157
10	992	0.188	1051	0.156	1206	0.188	1635	0.203
Avg.	994.8	0.156	1081.8	0.187	1210.9	0.205	1640.5	0.211



รูปที่ 3: ลักษณะการหาคำตอบของวิธี CLS (ปัญหา P4)



รูปที่ 3 แสดงลักษณะการหาคำตอบของวิธี CLS สำหรับปัญหา P4 ซึ่งมีจุดส่งสินค้าจำนวน 200 จุด โดยเส้นทึบแสดงคำตอบที่ดีที่สุด (Current best) และเส้นบางแสดงคำตอบที่หาได้ทีหาได้ ณ รอบการคำนวณนั้นๆ

สำหรับปัญหาการจัดเส้นทางยานพาหนะของบริษัทขนส่งรถยนต์ เราอาจใช้ทั้งสองวิธีที่นำเสนอในการแก้ปัญหา โดยปัญหาที่มีจุดส่งสินค้าไม่เกิน 20 จุด ผู้ใช้สามารถเลือกใช้วิธี BnB ในการหาคำตอบที่ดีที่สุดได้ (Optimal solution) เพื่อช่วยให้บริษัทสามารถประหยัดค่าใช้จ่ายในการขนส่งมากที่สุด อย่างไรก็ตาม เมื่อปัญหามีขนาดที่ใหญ่ขึ้น และมีข้อจำกัดด้านเวลาในการหาคำตอบและทรัพยากรคอมพิวเตอร์ ผู้ใช้อาจจะเลือกวิธี CLS ในการหาคำตอบที่มีคุณภาพดี ภายในระยะเวลาอันรวดเร็ว (Good enough fast enough solution) สังเกตด้วยว่า เนื่องจากวิธี CLS ไม่สามารถประกันการค้นหาคำตอบที่ดีที่สุดได้ ระบบช่วยในการตัดสินใจที่พัฒนาขึ้นจะต้องมีความยืดหยุ่นเพียงพอ เพื่ออนุญาตให้ผู้ใช้สามารถปรับแผนเส้นทางขนส่งได้

## 6. Conclusions

งานวิจัยนี้ได้ประยุกต์ใช้เงื่อนไขแบบเซตมาจำลองปัญหาการจัดเส้นทางยานพาหนะของบริษัทขนส่งรถยนต์ และพัฒนาอัลกอริทึมเพื่อใช้ในการหาคำตอบของปัญหาดังกล่าว เราได้นำเสนอการประยุกต์ใช้วิธี BnB มาหาคำตอบที่ดีที่สุด รวมทั้งการนำวิธี CLS ที่มีที่มีขั้นตอนการทำงานไม่ซับซ้อนมาใช้ในการแก้ปัญหาการจัดเส้นทางยานพาหนะเมื่อมีจุดส่งสินค้าจำนวนมาก ผลจากการศึกษาชี้ให้เห็นว่า บริษัทขนส่งรถยนต์สามารถนำอัลกอริทึมที่พัฒนาขึ้นมาช่วยในตัดสินใจวางแผนเส้นทางขนส่งสินค้าได้

## 7. Reference

- [1] Balas, E. Padberg, MW. (1976) "Set partitioning: a survey," SIAM Review, vol. 18, pp. 710–60.
- [2] Clarke, G. and Wright, J.W. (1964) "Scheduling of vehicles from a central depot to a number of delivery points," Operations Research, vol. 12, pp. 568–581.
- [3] Osman, I.H. (1993) "Metastrategy simulated annealing and tabu search algorithms for the vehicle routing problem," Annals of Operations Research, vol. 41, pp. 421–451.
- [4] Kelly, J.P. and Xu, J. (1999) "A set-partitioning-based heuristic for the vehicle routing problem," Journal of Computing, vol. 11, no. 2, pp. 161-172.
- [5] Indra-Payoong, N., Kwan, R.S.K. and Proll, L. (2005) "Rail container service planning: a constraint-based approach," in: Kendall, G, Petrovic, S, Burke, E & Gendreau, M (editors) Multidisciplinary Scheduling: Theory and Applications, pp. 343-365 Springer-Verlag.