

## การปรับปรุงระบบการเดินทางพนักงานขายแบบเพื่อนบ้านใกล้สุดด้วยเทคนิคการเคลื่อนย้ายแบบ 2-opt

### Improvement of Nearest Neighbor Travel Salesman Problem by using 2-opt technique

ชวลิต โค้วระวงศ์<sup>1</sup>

#### บทคัดย่อ

การขนส่งของที่ต้องจอดแวะแต่ละสถานที่เป็นจำนวนมากในแต่ละเที่ยว ลำดับการจอดแวะที่ดีมีผลต่อระยะทาง ในทางคอมพิวเตอร์กิจกรรมสอดคล้องกับทฤษฎีในทางคอมพิวเตอร์เรื่องปัญหาการเดินทางของพนักงานขาย หรือที่รู้จักกันดีในชื่อ Travelling Salesman Problem (TSP) ซึ่งเป็นปัญหาเกี่ยวกับกราฟที่ใช้หาวงจร Hamiltonian ที่สั้นที่สุด TSP ถูกจัดอยู่ในกลุ่ม NP-Hard หรือปัญหาไม่สามารถหาคำตอบที่ดีที่สุดได้ในเวลาอันรวดเร็ว การแก้ปัญหา Travelling Salesman Problem ในปัจจุบันนิยมใช้ Heuristic Algorithm มากขึ้น ถึงแม้ว่าจะเป็นที่ไม่ได้ให้ผลลัพธ์ที่ดีที่สุด แต่เป็นเพราะ Heuristic Algorithm นั้นสามารถให้คำตอบในช่วงเวลาที่รวดเร็วและไม่ต้องรอกอนาน ทำซ้ำได้ Nearest Neighbor Algorithm เป็นหนึ่งใน Heuristic Algorithm ที่ง่ายที่สุดและตรงไปตรงมาที่สุด แต่ตัวของ Nearest Neighbor Algorithm เองก็ยังมีปัญหาที่ต้องแก้ไขตามมามากขึ้นคือการตัดกันของเส้นทาง ดังนั้นในงานวิจัยนี้จึงเป็นการปรับปรุงระบบการเดินทางพนักงานขายแบบเพื่อนบ้านใกล้สุดด้วยเทคนิคการเคลื่อนย้ายแบบ 2-opt ซึ่งจะเป็นการจัดการข้อเสียของ Nearest Neighbor Algorithm ผลการทดสอบของระบบในงานวิจัยนี้แบ่งออกเป็น 2 ด้าน คือ ด้านระยะเวลาในการประมวลผลของระบบ Nearest Neighbor Algorithm บนตำแหน่งต่างๆ จำนวน 1,000 ตำแหน่งได้ภายในระยะเวลาไม่เกิน 5 – 6 วินาที แต่เมื่อรวมเทคนิค 2-opt ไปด้วยก็จะใช้เวลาเยอะขึ้นสำหรับในด้าน การตัดกันของเส้นทางเมื่อใช้เทคนิค 2-opt การตัดกันของเส้นทางจะลดลงหรือไม่มีเลยในบางกรณี

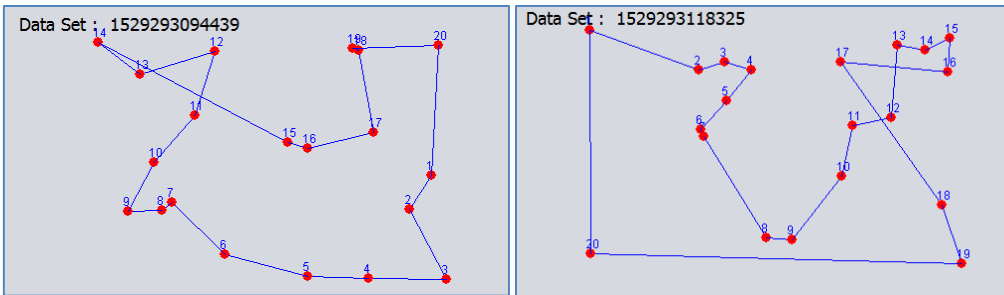
**คำสำคัญ:** ปัญหาการเดินทางของพนักงานขาย, เพื่อนบ้านใกล้สุด, ขั้นตอนวิธี, การตัดกันของเส้นทางวงจร, 2-opt

#### Abstract

Transportation that have a lot of visited points in each trip. Ordering visited points affects the distance. In the computer science, this problem is called as travel salesman problem, which is a problem with the graph used to find the shortest Hamiltonian circuit. The TSP is classified as NP-Hard problem cannot be answered in the best possible time. Today, Heuristic algorithms are used, even though they are not the best algorithms, but the Heuristic algorithm can provide answers in a acceptable time and without long waiting, we can repeat this process as often as we can. Nearest neighbor algorithm is one of the most simple and straightforward heuristic algorithm, but the nearest

<sup>1</sup>คณะวิทยาศาสตร์และเทคโนโลยี มหาวิทยาลัยราชภัฏวไลยอลงกรณ์ ในพระบรมราชูปถัมภ์ Emails: chavalit@vru.ac.th





ภาพที่ 1 ตัวอย่างผลลัพธ์ที่เกิดจากการใช้ Nearest Neighbor TSP

จากภาพที่ 1 จะพบว่าเมื่อตำแหน่งต่างเปลี่ยนแปลงไป มีโอกาสที่ Nearest Neighbor จะให้ผลลัพธ์เป็นเส้นทางที่ตัดกันโดยไม่ตั้งใจ ซึ่งรูปแบบการตัดกันของเส้นทางเป็นสิ่งที่ต้องกำจัดออกไปเพราะ จะทำให้ระยะทางโดยรวมสูงขึ้นได้และวิธีที่จะนำมาใช้จัดการการตัดกันของเส้นทางก็คือ เทคนิคการเคลื่อนย้ายแบบ 2-opt ซึ่งการเคลื่อนย้ายแบบ 2-opt คือ การตรวจสอบวงจรในเส้นทางว่ามีการตัดกันที่ใดบ้าง และถ้าตรวจพบการตัดกันจะต้องมีการปรับปรุงเส้นทางใหม่ให้ดีกว่าเดิม

ดังนั้นงานวิจัยนี้จะเป็นเรื่องเกี่ยวกับการปรับปรุงระบบการเดินทางพนักงานขายแบบเพื่อนบ้านใกล้ที่สุดด้วยเทคนิคการเคลื่อนย้ายแบบ 2-opt

### วัตถุประสงค์ของงานวิจัย

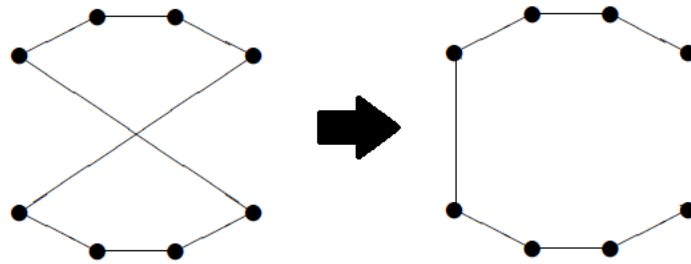
1. เพื่อลดจุดตัดที่เกิดขึ้นจากการคำนวณการเส้นทางจากเทคนิค Nearest Neighbor
2. เพื่อประยุกต์ใช้เทคนิคการเคลื่อนย้ายแบบ 2 opt

### เนื้อหาที่เกี่ยวข้อง

อัลกอริทึมที่ใช้จัดการปัญหา TSP ที่ได้รับการยอมรับว่าดีที่สุดนั้น อยู่ในกลุ่ม Ant Colony หรือ Genetic Algorithm (Nilsson, 2003) แม้ว่าจะสามารถให้ผลลัพธ์ที่ดีที่สุดในการหาคำตอบใกล้เคียงกับค่าความจริงมากที่สุด แต่ใช้เวลาในการประมวลผลค่อนข้างนานเมื่อมีจำนวนโหนดในกราฟมากขึ้นและคำนวณมีการคำนวณซ้ำผลลัพธ์กลับไม่เหมือนเดิม เนื่องจากระบบดังกล่าวนี้อาศัยเทคนิคการสุ่มเพื่อป้องกันการค้นพบ local maxima แล้วเข้าใจว่าเป็น Global Maxima ทำให้ในทางปฏิบัติแล้วจึงไม่ค่อยพบว่ามีกรนำอัลกอริทึมที่ได้รับการยอมรับเหล่านี้มาใช้ในงานจริง ในทางตรงข้ามบนเส้นทางอีกสายหนึ่ง Approximation algorithm หรือ Heuristics Algorithm แม้ว่าจะสร้างผลลัพธ์ได้ห่างไกลจากค่าความจริงมากกว่ากลุ่ม Ant Colony หรือ Genetic Algorithm แต่ได้เปรียบในเรื่องของการคำนวณที่รวดเร็วกว่าและทำซ้ำได้ผลลัพธ์เท่าเดิม ทำให้การคำนวณแบบ Heuristics Algorithm นั้นถูกนำมาใช้ใน ชีวิตประจำวันมากกว่า

Nearest neighbor เป็นหนึ่งใน Heuristics Algorithm ที่มีระบบการคิดที่เรียบง่ายที่สุด สำหรับวิธีเพื่อนบ้านใกล้ที่สุด (Nearest Neighbor) (Nilsson, 2003) จะมีขั้นตอนการคำนวณดังต่อไปนี้

- ขั้นตอนที่ 1. เลือกจุดแวะที่จะไปแบบสุ่มเป็นจุดแรก
- ขั้นตอนที่ 2. เลือกจุดแวะถัดไป (ที่ยังไม่เคยแวะไป) ที่ใกล้ที่สุด
- ขั้นตอนที่ 3. ตรวจสอบว่ามีจุดที่ยังไม่เคยแวะไปหรือไม่ ถ้าใช่, ทำซ้ำขั้นตอนที่ 2.
- ขั้นตอนที่ 4. กลับไปยังจุดแวะที่ 1



ภาพที่ 2 การเคลื่อนย้ายแบบ 2-opt

การเคลื่อนย้ายแบบ 2-opt คือ การตรวจสอบวงจรในเส้นทางว่ามีการตัดกันที่ใดบ้าง และถ้าตรวจพบการตัดกันจะต้องมีการปรับปรุงเส้นทางใหม่ให้ดีกว่าเดิม จากรูปที่ 2 จะเห็นว่าการเคลื่อนย้ายแบบ 2-opt จะช่วยปรับปรุงการตัดกันของเส้นทางได้

การประยุกต์เทคนิคการการเคลื่อนย้ายแบบ 2-opt มาจัดการการตัดกันของเส้นทางหลังจากได้ผลลัพธ์จาก Nearest Neighbor สามารถทำได้โดยด้วยขั้นตอนดังนี้

- ขั้นตอนที่ 1 ค้นหาการตัดกันของเส้นทางด้วยสมการทางคณิตศาสตร์ภายในเส้นทางทั้งหมด
- ขั้นตอนที่ 2 หากพบเส้นทางตัดกัน ถ้าไม่พบในจบการทำงาน
- ขั้นตอนที่ 3 ยกเลิกการลากเส้น 2 เส้น จากจุด 4 จุด
- ขั้นตอนที่ 4 สลับสับเปลี่ยนการเชื่อมต่อระหว่างจุดทั้ง 4 เพื่อให้การตัดกันของเส้นทางหายไป
- ขั้นตอนที่ 5. กลับไปยังขั้นตอนที่ 1

### วิธีดำเนินการวิจัย

งานวิจัยนี้ถูกพัฒนาโดยภาษาจาวา (Java) ในลักษณะของ Java Application Standalone เพื่อทำการทดสอบในเครื่องคอมพิวเตอร์ตั้งโต๊ะทั่วไป สำหรับวิธีทดลองขั้นตอนแรกเลือกชุดข้อมูลที่ใช้ทดลองพร้อมตั้งค่าตามวัตถุประสงค์ที่จะทดลอง ขั้นตอนที่สองกดปุ่ม “optimization” เพื่อหาคำตอบ บันทึกผลข้อมูลนำเข้า และข้อมูลนำออกเอาไว้เพื่อมาวิเคราะห์ โดยพิจารณา 2 ประเด็นที่สนใจ คือ ระยะเวลาในการประมวลผล และการตัดกันของเส้นทาง ตำแหน่งข้อมูลที่ใช้จะอยู่ในระบบ XY ที่มีตำแหน่ง (0,0) จนถึง (600,400) โดยเปรียบเทียบความแตกต่างระหว่างเทคนิค Nearest Neighbor และเทคนิค Nearest Neighbor + 2-opt

ในการทดสอบระยะเวลาในการประมวลผล ข้อมูลที่ใช้จะเป็นข้อมูล 1,000 ตำแหน่ง โดยทำการสุ่มข้อมูลตำแหน่ง 5 ชุด กดปุ่ม optimization เพื่อสั่งให้ระบบทำงานแล้วบันทึกระยะเวลาที่ใช้ในการประมวลผลของแต่ละชุดข้อมูล จากนั้นนำค่าทั้งหมดมาหาค่าเฉลี่ยกัน

ในด้านการตัดกันของเส้นทาง ชุดข้อมูลที่ใช้จะเป็นของข้อมูล 20 ตำแหน่ง ดังภาพที่ 1 โดยทำการสุ่มข้อมูลตำแหน่ง 5 ชุด กดปุ่ม optimization เพื่อสั่งให้ระบบทำงานแล้วพิจารณาการตัดกันของเส้นทาง

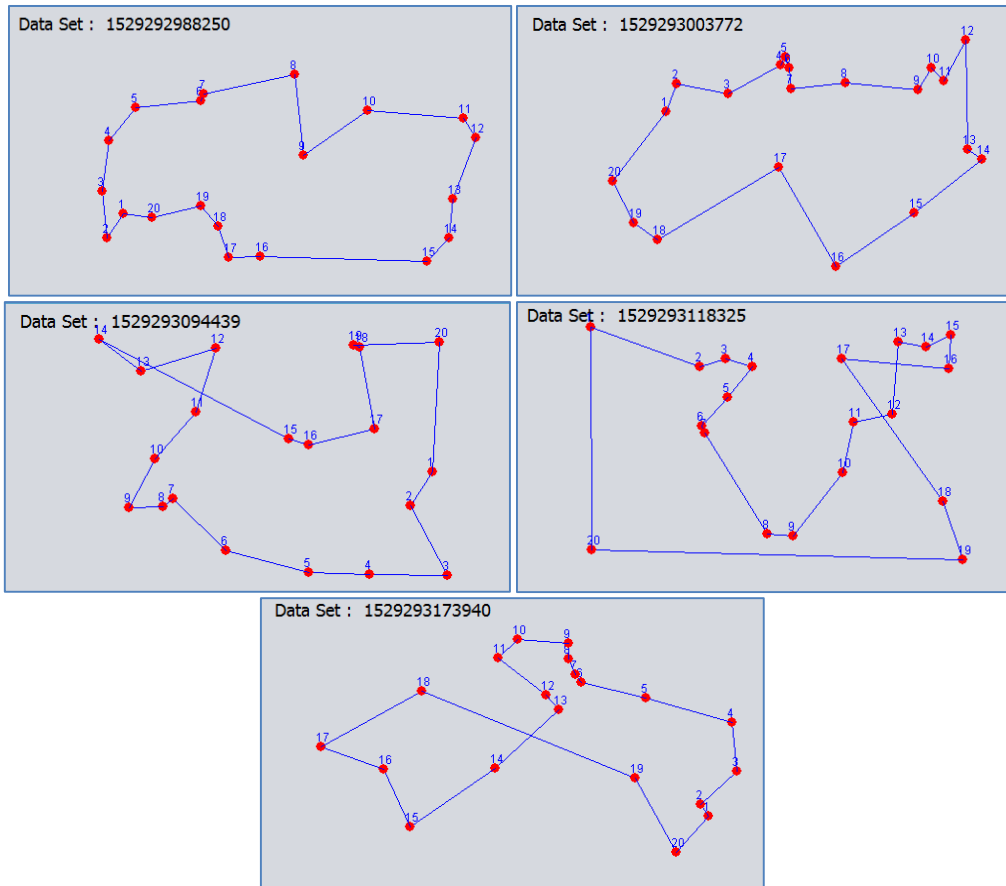
### สรุปผลการวิจัย

หากพิจารณาจากระยะเวลาในการคำนวณ ดังตารางที่ 1 คือ ข้อมูลทุกชุดสามารถสามารถคำนวณเสร็จและได้คำตอบอย่างรวดเร็วในระดับวินาที โดยมีระยะเวลาในการคำนวณตั้งชุดข้อมูลที่ 1 – 5 เมื่อไม่มี 2-opt ตามลำดับดังต่อไปนี้ ได้แก่ 5.794, 5.782, 5.741, 5.735, 5.799 วินาที แต่เมื่อมี 2-opt move จะใช้เวลามากขึ้นเกือบสองเท่า

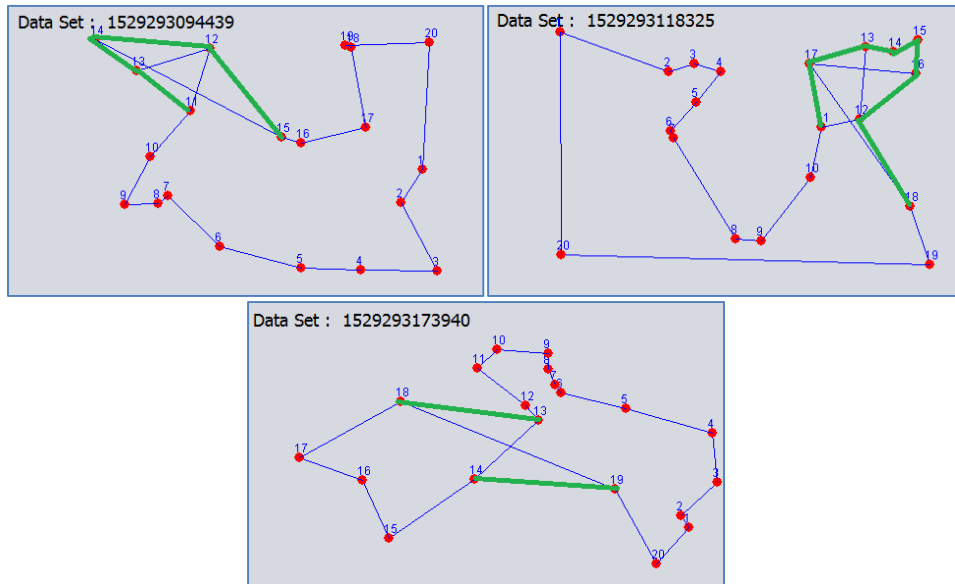
**ตารางที่ 1** ผลการทดลองด้านระยะเวลาในการคำนวณ TSP 1000 ตำแหน่ง

ชุดข้อมูล	ระยะเวลาในการคำนวณ (Second) แบบไม่มี 2-opt move	ระยะเวลาในการคำนวณ (Second) แบบมี 2-opt move
ชุดข้อมูลที่ 1	5.794	9.761
ชุดข้อมูลที่ 2	5.782	9.315
ชุดข้อมูลที่ 3	5.741	9.463
ชุดข้อมูลที่ 4	5.735	9.487
ชุดข้อมูลที่ 5	5.799	9.751
ค่าเฉลี่ย	5.772	9.324

หากพิจารณาจากการตัดกันของเส้นทางภาพที่ 2 คือ ข้อมูลชุดที่ 1 และ 2 นั้นจะได้คำตอบที่ดีและน่าเชื่อถือ ซึ่งลักษณะของวงจรไม่มีกันตัดกัน ในขณะที่ข้อมูลชุดที่ 3 – 5 นั้นจะได้วงจรที่มีการตัดกันมากบ้าง น้อยบ้าง หรือไม่ตัดกันเลย ซึ่งหากนำผลลัพธ์ของข้อมูลชุดที่ 3 – 5 มาเข้ากระบวนการปรับปรุงวงจรเพื่อไม่ให้เส้นตัดกันแล้วจะช่วยให้ระยะทางลดลง 4.14%, 5.83%, และ 3.03% ตามลำดับ โดยเส้นทางที่เปลี่ยนแปลงสามารถดูได้จากภาพที่ 3 และผลเปรียบเทียบสามารถดูได้จากตารางที่ 2



ภาพที่ 2 ผลการทดลอง Nearest Neighbor Algorithm



ภาพที่ 3 ผลการลดการตัดกันของวงจรที่ได้จาก Heuristic Algorithm

ตารางที่ 2 ผลการทดลองด้านความสมเหตุสมผลในการคำนวณ TSP 20 ตำแหน่ง

ชุดข้อมูลที่	รหัสชุดข้อมูล	Nearest Neighbor (px)	Nearest Neighbor + 2-opt move (px)	การแปรผลด้าน ระยะทาง
ชุดข้อมูลที่ 1	1529292988250	1129	1129	เท่าเดิม
ชุดข้อมูลที่ 2	1529293003772	1160	1160	เท่าเดิม
ชุดข้อมูลที่ 3	1529293094439	1391	1333	ลดลง 4.14%
ชุดข้อมูลที่ 4	1529293118325	1687	1589	ลดลง 5.83%
ชุดข้อมูลที่ 5	1529293173940	1296	1257	ลดลง 3.03%

### อภิปรายผล

เรื่องการตัดกันของเส้นในวงจรจะพบว่าการตัดกันของเส้นในวงจรมีผลทำให้ระยะทางเพิ่มมากขึ้น ดังนั้นในเบื้องต้นหากลดการตัดกันของเส้นในวงจรมีผลทำให้ระยะทางลดลงได้ และทำให้วงจรดูน่าเชื่อถือขึ้นแม้ว่าจะคำตอบที่ได้อาจจะยังไม่ใช่วงจรที่มีระยะทางสั้นที่สุดตาม แต่ก็พิสูจน์ได้ยากด้วยตาเปล่าว่าวิธีที่ดีกว่าจะเป็นแบบใด ซึ่งข้อนี้จะสามารถทำให้ตอบโจทย์ในเรื่องความน่าเชื่อถือได้ระดับหนึ่ง ดังนั้นกระบวนการปรับปรุงวงจรที่ลดการตัดกันของวงจรจึงถูกพัฒนาขึ้นโดยใช้เทคนิคที่เรียกว่าการเคลื่อนย้ายแบบ 2-opt เข้ามาประมวผลต่อจากเทคนิค Nearest neighbor และผลลัพธ์ที่ได้ก็คือ จุดตัดที่เคยเกิดขึ้นมีจำนวนลดลงหรือหายไป แต่แน่นอนว่าระยะเวลาในการประมวผลก็จะนานขึ้นเล็กน้อย

### ข้อเสนอแนะ

#### ข้อเสนอแนะในการนำผลวิจัยไปใช้

(1) ระบบนี้สามารถนำไปพัฒนากับงาน Server ได้เนื่องจากข้อดีในเรื่องของระยะเวลาและความแน่นอนในคำตอบที่สามารถทำซ้ำได้

#### ข้อเสนอแนะในการวิจัยครั้งต่อไป

- (1) พัฒนา Algorithm ที่รวดเร็วขึ้น โดยทำให้อยู่ระดับการประมวลไม่เกิน 5 วินาที
- (2) การพัฒนาชุดโปรแกรมร่วมกับงาน server

### เอกสารอ้างอิง

- Beardwood, J.; Halton, J.H.; Hammersley, J.M. (1959), "The Shortest Path through Many Points", *Proceedings of the Cambridge Philosophical Society*, 55: 299–327,
- Rego, César; Gamboa, Dorabela; Glover, Fred; Osterman, Colin (2011), "Traveling salesman problem heuristics: leading methods, implementations and latest advances", *European Journal of Operational Research*, 211 (3): 427–441

Johnson, D. S.; McGeoch, L. A. (1997). "The Traveling Salesman Problem: A Case Study in Local optimization" (PDF). In **Aarts, E. H. L.; Lenstra, J. K. Local Search in Combinatorial optimisation**. London: John Wiley and Sons Ltd. pp. 215–310.

C.Nilsson.(2003), Heuristics for the traveling salesman problem. **Tech. Report, Linköping University**,  
<http://web.tuke.sk/fei-cit/butka/hop/htsp.pdf>.