ขั้นตอนวิธีการจำลองการอบเหนียวสำหรับการวางแผนแผนการเดินทางท่องเที่ยวภายใต้เงื่อนไขบังคับด้านเวลา

A Simulated Annealing Algorithm for Travel Itinerary Planning under Time Constraints

ปิยรัตน์ งามสนิท[1], ธรา อั่งสกุล[2], จิติมนต์ อั่งสกุล[3]

Piyarat Ngamsanit, Thara Angskun, Jitimon Angskun

Received: 23 January 2017 ; Accepted: 19 April 2017

บทคัดย่อ

ปัญหาด้านการวางแผนการเดินทางท่องเที่ยว เกี่ยวข้องกับการจัดเส้นทางท่องเที่ยวให้เหมาะสมภายใต้เงื่อนไขบังคับที่แตกต่างกัน โดยเงื่อนไขบังคับที่นักท่องเที่ยวต่างมีเหมือนกันคือ ด้านเวลา เพราะเป็นสิ่งที่มีค่าและมีจำกัด ส่งผลให้ไม่สามารถไปเที่ยวได้ครบทุกสถานที่ตามที่อยากไปได้ บทความนี้จึงนำเสนอขั้นตอนวิธีการวางแผนการเดินทางที่มุ่งเน้นการวางแผนแผนการเดินทางตามสถานที่ที่ผู้ใช้ระบุว่าอยากไปให้ได้ครบหรือหลายแห่งมากที่สุดภายในระยะเวลาที่ผู้ใช้กำหนด โดยใช้แนวคิดจากขั้นตอนวิธีการจำลองการอบเหนียวเป็นพื้นฐานในการพัฒนา และเพื่อเพิ่มประสิทธิภาพของขั้นตอนวิธีจึงได้เสนอการใช้เทคนิคการเลือกเส้นทางที่พิจารณาจากระยะเวลาการเดินทางที่น้อยที่สุด (เอสพีเอส) มาใช้ในการสร้างคำตอบเบื้องต้นและการปรับปรุงแผนการเดินทาง นอกจากนี้ ยังได้แทรกส่วนพิเศษเพื่อการเพิ่มสถานที่ที่อาจจะยังเหลือให้ได้อยู่ในแผนการเดินทางซึ่งจะทำให้มีโอกาสได้แผนการเดินทางที่อาจไปเยือนสถานที่ได้หลายแห่งมากขึ้นในการทดสอบประสิทธิภาพทำโดยการเปรียบเทียบกับขั้นตอนวิธีอื่นในการทดสอบประสิทธิภาพด้านเวลาในการประมวลผล ผลการทดลองพบว่า ขั้นตอนวิธีการจำลองการอบเหนียวที่มีพื้นฐานจากการค้นหาตามระยะเวลาการเดินทางที่น้อยที่สุด (เอสเอ_เอสพีเอส) ที่นำเสนอใช้เวลาในการประมวลผลน้อยกว่าทุกขั้นตอนวิธีที่นำมาเปรียบเทียบ ส่วนในการทดสอบประสิทธิภาพด้านคุณภาพของคำตอบเมื่อเทียบกับขั้นตอนวิธีค้นหาทุกเส้นทางที่เป็นไปได้ ขั้นตอนวิธีที่นำเสนอให้คุณภาพของคำตอบที่แย่กว่าเพียงเล็กน้อยเท่านั้น แต่อย่างไรก็ตาม ขั้นตอนวิธีที่นำเสนอให้คุณภาพของคำตอบที่ดีกว่าเมื่อเทียบกับขั้นตอนวิธีการจำลองการอบเหนียวแบบดั้งเดิม

คำสำคัญ: ขั้นตอนวิธีการจำลองการอบเหนียว การค้นหาตามระยะเวลาการเดินทางที่น้อยที่สุด การวางแผนการเดินทางท่องเที่ยวเงื่อนไขบังคับด้านเวลา

Abstract

A travel planning problem was invented for the purpose of determining a workable itinerary under different travel constraints. The typical constraint most travelers encounter is the limited time constraint. It often means a traveler cannot reach all desired destinations. In this article, we propose a travel itinerary planning algorithm focused on facilitating travelers to reach their destinations as often as possible under given time constraints. The proposed algorithm is based on a simulated annealing algorithm and applies a shortest path search technique (SPS) to increase the efficiency of the algorithm. The SPS technique was used to find a minimized travel time scenario for an initial solution and then improve the travel itinerary. In addition, a special search procedure was used to add the remaining destinations into itineraries. With this procedure, we will have the opportunity to add even more destinations into the itinerary.The performance of the proposed algorithm (SA_SPS) was compared with other travel itinerary planning algorithms in two aspects. The first aspect was to test CPU time to run the algorithms. The experimental results reveal that the proposed SA_SPS spends

[1] *นักศึกษาปริญญาเอก,* [2, 3]*ผู้ช่วยศาสตราจารย์, สำนักวิชาเทคโนโลยีสังคม มหาวิทยาลัยเทคโนโลยีสุรนารี อำเภอเมือง จังหวัดนครราชสีมา 30000*

[1] *doctoral degree student,* [2, 3]*Assist. Prof., Faculty of Social Technology, Suranaree University of Technology, Mueng District, Nakhon Ratchasima 30000, Thailand.*

less CPU time than other methods. The second aspect was to test the quality of the solution. The results indicate that the quality of the solution of the SA_SPS algorithm is slightly worse than an exhaustive routing algorithm. However, the SA_SPS provided more quality in the solution than a traditional simulated annealing algorithm.

Keywords: Simulated Annealing Algorithm, Shortest Path Search, Travel Itinerary Planning, Time Constraints

บทนำ

ในการนำเสนอสารสนเทศเพื่อตอบสนองความต้องการด้านการท่องเที่ยวได้ส่งผลให้เกิดการพัฒนาระบบที่ทำหน้าที่ช่วยสนับสนุนการให้ข้อมูลตลอดจนช่วยวางแผนการเดินทางแก่นักท่องเที่ยว โดยคำนึงถึงเงื่อนไขบังคับที่หลากหลายและแตกต่างกันในการวางแผน แต่เงื่อนไขบังคับหนึ่งที่นักท่องเที่ยวมีเหมือนกันทุกคน คือเงื่อนไขบังคับด้านเวลา[1] เนื่องจากเป็นความจริงที่ว่า เวลาเป็นสิ่งที่มีค่าและนักท่องเที่ยวล้วนมีเวลาสำหรับการเดินทางที่จำกัด ซึ่งทำให้ไม่สามารถเดินทางไปเที่ยวครบทุกแห่งตามที่อยากไปได้

จากเหตุผลดังกล่าวข้างต้น ในกระบวนการวางแผนการเดินทางของระบบจึงควรต้องมีการพิจารณาถึงการใช้เวลาในการเดินทางให้คุ้มค่า กล่าวคือ เมื่อนักท่องเที่ยวได้เลือกสถานที่ที่อยากไปแล้ว ระบบควรมีการพิจารณาถึงแผนการเดินทางว่ามีสถานที่ท่องเที่ยวแห่งใดบ้าง ใช้ระยะเวลาในการเดินทางเท่าไหร่ สามารถเดินทางครอบคลุมทุกสถานที่ตามที่ผู้ใช้อยากไปได้หรือไม่ หรือหากไปได้ไม่ครบก็ควรจะไปให้ได้หลายแห่งมากที่สุดเท่าที่จะเป็นไปได้ภายใต้เวลาที่มีอยู่อย่างจำกัด เพราะนั่นหมายถึงการตอบสนองให้ตรงตามความต้องการของผู้ใช้อย่างหนึ่ง ซึ่งปัจจัยหลักที่ทำให้ระบบวางแผนสำเร็จนั่นก็คือ ขั้นตอนวิธี (Algorithm) ที่ใช้ในการวางแผนการเดินทาง[2]

สำหรับปัญหาการวางแผนการเดินทาง จัดอยู่ในกลุ่มของปัญหาที่ซับซ้อนยากต่อการหาคำตอบที่ดี-ที่สุด[3] (NP-Hard Problems) เพราะต้องมีการตัดสินใจเลือกเส้นทางที่เหมาะสมจากรูปแบบที่เป็นไปได้ทั้งหมดจำนวนมาก ตัวอย่างของรูปแบบการเดินทางที่เป็นไปได้ทั้งหมดของการท่องเที่ยว 3 สถานที่แสดงดัง(Table1) คำนวณได้จากสมการที่ 1

Table 1 An example of travel patterns in order to reach all destinations

Destinations	Travel patterns
1	(0-1-0), (0-2-0), (0-3-0)
2	(0-1-2-0), (0-1-3-0), (0-2-1-0), (0-2-3-0), (0-3-1-0), (0-3-2-0)
3	(0-1-2-3-0), (0-1-3-2-0), (0-2-1-3-0), (0-2-3-1-0), (0-3-1-2-0), (0-3-2-1-0)

$$\sum_{l=0}^{n-1} \frac{n!}{l!} \quad (1)$$

n คือ จำนวนสถานที่ท่องเที่ยว

l มีค่าเริ่มต้นตั้งแต่ 0 ถึง *n*– 1

จากสมการที่ 1 ถ้ามีสถานที่ท่องเที่ยวจำนวน 7 แห่ง รูปแบบการเดินทางที่เป็นไปได้คือ 13,699 รูปแบบและถ้ามี 15 แห่ง จะมีรูปแบบการเดินทางที่เป็นไปได้ถึง 3 ล้านล้านรูปแบบ จะเห็นได้ว่า หากต้องการแผนการเดินทางที่ดีที่สุดในสถานที่ 7 แห่ง อาจมีความเป็นไปได้ แต่อาจเป็นไปไม่ได้ในสถานที่ 15 แห่ง เนื่องจากปริภูมิการค้นหา (Search Space) มีขนาดใหญ่มาก ส่งผลให้ใช้เวลาประมวลผลนานจนเป็นไปไม่ได้ในทางปฏิบัติยิ่งไปกว่านั้น ในการวางแผนการเดินทางท่องเที่ยวแต่ละครั้ง เป็นที่ทราบกันโดยทั่วไปว่าย่อมมีเงื่อนไขบังคับประกอบอื่นที่ต้องพิจารณาในการวางแผน ซึ่งล้วนส่งผลให้รูปแบบการคำนวณของขั้นตอนวิธีทวีความซับซ้อนมากยิ่งขึ้นจนไม่สามารถใช้ขั้นตอนวิธีแบบค้นหาทุกเส้นทางที่เป็นไปได้ (Exhaustive Routing Algorithm: ER) ในการค้นหาคำตอบที่ดีที่สุด (Global Optimum)

จากปัญหาดังกล่าว เมื่อศึกษาเทคนิคต่าง ๆ ที่นำมาประยุกต์ใช้ในปัญหาการวางแผนการเดินทางจากหลากหลายแขนงพบว่า วิธีแบบเมตาฮิวริสติก (Metaheuristic Methods) สามารถนำมาใช้แก้ปัญหาลักษณะนี้ได้เป็นอย่างดี เนื่องจากเป็นวิธีการที่คำนวณหาเพียงบางส่วนของรูปแบบที่คาดว่าเป็นคำตอบ จึงลดเวลาในการคำนวณลงได้ อีกทั้งยังมีเทคนิคช่วยในการพิจารณาคุณภาพของคำตอบ ทำให้คำตอบที่ได้มีคุณภาพดีเพียงพอที่จะใช้นำไปใช้ภายในระยะเวลาการคำนวณที่เหมาะสม

ขั้นตอนวิธีการจำลองการอบเหนียว (Simulated Annealing Algorithm: SA) เป็นหนึ่งในขั้นตอนวิธีแบบเมตาฮิวริสติกที่นิยมนำมาเป็นพื้นฐาน (Base) ในการพัฒนาต่อยอดเพื่อแก้ไขปัญหาในกลุ่มของการวางแผนการเดินทางอย่างกว้างขวาง ในปัญหาการวางแผนการเดินทางของยานพาหนะ (Vehicle Routing Problem: VRP) ก็เช่นเดียวกัน ได้มีงานวิจัยที่ยืนยันว่าสามารถประยุกต์ขั้นตอนวิธีการจำลองการอบเหนียวเพื่อใช้ในปัญหาการวางแผนการขนส่งภายใต้กรอบเวลา[4,5,6] (VRP with Time Windows)ได้เป็นอย่างดี แต่เนื่องจากขั้นตอนวิธีนี้มีกระบวนการค้นหาที่ไม่มีความเฉพาะเจาะจงกับ

แบบจำลองหรือลักษณะเฉพาะของข้อมูลแบบใดแบบหนึ่ง จึงทำให้ไม่สามารถนำไปใช้หาผลเฉลยของปัญหาใด ๆ ได้โดยตรงซึ่งจะเห็นได้จากในหลาย ๆ งานวิจัยที่ได้มีการปรับปรุงขั้นตอนวิธีดังกล่าวให้มีประสิทธิภาพมากขึ้นด้วยวิธีการที่หลากหลาย ตัวอย่างเช่น การใช้ขั้นตอนวิธีการจำลองการอบเหนียวร่วมกับขั้นตอนวิธีหาค่าเหมาะสมที่สุดแบบกลุ่มอนุภาค[7] (Particle Swarm Optimization: PSO) การใช้ขั้นตอนวิธีการจำลองการอบเหนียวร่วมกับขั้นตอนวิธีแบบละโมบ (Greedy-Algorithm[8] การใช้ขั้นตอนวิธีการจำลองการอบเหนียวร่วมกับการค้นหาเฉพาะที่[9] (Local Search) ฯลฯ)

จากรูปแบบปัญหาการวิจัยในครั้งนี้ที่ต้องการแผนการเดินทางท่องเที่ยวที่เน้นการบริหารเวลากับจุดหมายปลายทางให้เหมาะสม คือไปเที่ยวในสถานที่ที่ผู้ใช้เป็นผู้ระบุว่าอยากไปให้ได้ครบหรือหลายแห่งมากที่สุดภายในระยะเวลาที่ผู้ใช้เองเป็นผู้กำหนดซึ่งในการเลือกเยือนสถานที่หากนำแนวคิดการค้นหาจากระยะเวลาการเดินทางที่น้อยที่สุดมาใช้ร่วมกับขั้นตอนวิธีการจำลองการอบเหนียวในการวางแผน ก็มีความน่าจะเป็นที่จะทำให้ได้แผนการเดินทางท่องเที่ยวที่ประกอบด้วยสถานที่ท่องเที่ยวครบหรือหลายแห่งมากที่สุดภายในระยะเวลาการคำนวณที่เหมาะสม จึงนำมาสู่บทความนี้ ที่นำเสนอขั้นตอนวิธีการจำลองการอบ-เหนียวที่มีพื้นฐานจากการค้นหาตามระยะเวลาการเดินทางที่น้อยที่สุดสำหรับการวางแผนการเดินทางท่องเที่ยวภายใต้เงื่อนไขบังคับด้านเวลาหรือเรียกชื่อย่อว่า เอสเอ_เอสพีเอส (Simulated Annealing Algorithm Based on Shortest Path Search: SA_SPS)

วิธีการวิจัย

สำหรับวิธีการวิจัยประกอบด้วยรูปแบบปัญหา การออกแบบและพัฒนาขั้นตอนวิธีและสภาพแวดล้อมในการทดลอง ซึ่งมีรายละเอียดดังต่อไปนี้

รูปแบบปัญหา

สำหรับปัญหาในการวิจัยครั้งนี้ คือปัญหาการวางแผนการเดินทางท่องเที่ยวที่ต้องการเดินทางไปเยือนสถานที่หลายแห่งมากที่สุดภายในระยะเวลาจำกัด ซึ่งหมายถึงปัญหาการตัดสินใจหนึ่งที่ต้องการเดินทางไปเยือนสถานที่ที่กำหนดสถานที่ละหนึ่งครั้ง โดยที่ระยะเวลาในการเดินทางไปและกลับระหว่างสองสถานที่ใช้เวลาไม่เท่ากัน (Asymmetrical Distance) ผลลัพธ์ที่ได้จากการแก้ไขปัญหานี้ คือแผนการเดินทางที่อาจเยือนสถานที่ปลายทางได้ครบหรือไม่ครบทุกแห่งก็ได้โดยที่จุดเริ่มต้นและจุดสิ้นสุดเป็นจุดเดียวกัน แต่ทั้งนี้ แผนการเดินทางจะต้องคำนึงถึงการเยือนสถานที่ให้ได้หลายแห่งมากที่สุดเท่าที่จะเป็นไปได้ภายใต้เงื่อนไขบังคับด้านเวลาและแผนการเดินทางต้องสามารถแสดงลำดับการผ่านสถานที่ได้โดยในการประมวลผลแต่ละครั้ง แผนการเดินทางที่ได้อาจมีจำนวนสถานที่ปลายทางมากน้อยต่างกัน ในที่นี้ถือว่าจำนวนสถานที่ที่อยู่ในแผนการเดินทางคือคุณภาพของคำตอบ โดยแผนการเดินทางใดที่มีจำนวนสถานที่ปลายทางตามที่ผู้ใช้ระบุหลายแห่งมากกว่า จะถือว่าเป็นคำตอบที่มีคุณภาพดีกว่า

เป็นที่ทราบกันดีว่า มีเงื่อนไขบังคับด้านเวลามากมายที่มีผลต่อการวางแผนการเดินทางท่องเที่ยว ในบทความนี้ได้นำเงื่อนไขบังคับด้านเวลาบางส่วนตามงานวิจัยของศศิวิมล กอบัว[1] ที่ได้มีการศึกษาปัจจัยที่เกี่ยวข้องด้านเวลาในการท่องเที่ยวเชิงลึกมาใช้ประกอบการวางแผนการเดินทางได้แก่ *1) เงื่อนไขบังคับด้านระยะเวลาในการเดินทางระหว่างสถานที่ (Travelling Time between Locations)* หมายถึง ระยะเวลาในการเดินทางระหว่างสองสถานที่ *2) เงื่อนไขบังคับด้านเวลาที่ใช้ ณ สถานที่ท่องเที่ยว (Visiting Time)* หมายถึง ระยะเวลาที่แวะเที่ยว ณ สถานที่นั้น และ *3) เงื่อนไขบังคับด้านเวลาของแผนการเดินทาง (Time of Itinerary)* หมายถึง ระยะเวลาทั้งหมดในการท่องเที่ยวตั้งแต่เริ่มต้นจนสิ้นสุดการเดินทางท่องเที่ยว

โดยการวางแผนการเดินทางท่องเที่ยวนี้ เป็นการวางแผนแบบล่วงหน้าด้วยรถยนต์ส่วนบุคคล และผู้ใช้เป็นผู้ระบุจุดเริ่มต้น สถานที่ท่องเที่ยว เวลาที่ใช้ ณ สถานที่ และวันเวลาเริ่มต้น/สิ้นสุดการเดินทางด้วยตนเองสำหรับข้อมูลพิกัดของแต่ละสถานที่ ความเร็วของรถยนต์ที่สามารถเดินทางได้ ระยะทางและระยะเวลาในการเดินทางระหว่างสถานที่ ได้จากการคำนวณโดยการเดินทางด้วยรถยนต์ของบริการส่วนต่อประสานโปรแกรมประยุกต์แผนที่กูเกิล หรือที่เรียกว่า กูเกิลแมพ เอพีไอ (Google Maps API)

การออกแบบและพัฒนาขั้นตอนวิธี

ขั้นตอนวิธีการจำลองการอบเหนียว เป็นเทคนิคการค้นหาคำตอบจากการสุ่มจุดตามที่สำรวจซ้ำจากการพัฒนาคำตอบเบื้องต้นเพื่อหาผลเฉลยอื่นที่ดีกว่า[10, 11] กระบวนการทำงานพื้นฐานหลัก ๆ ได้แก่ การสร้างคำตอบเบื้องต้น (Initial Solution) โดยการวนสุ่มจุดขึ้นมาจนได้หนึ่งคำตอบ จากนั้นเข้าสู่กระบวนการปรับปรุงคำตอบด้วยการสุ่มจุด (Random Swap) เพื่อค้นหาคำตอบใหม่ (Candidate Solution) ที่คาดว่าน่าจะดีขึ้นด้วยวิธีการค้นหาคำตอบใกล้เคียง (Neighborhood Function) ทำซ้ำไปเรื่อย ๆ โดยใช้คำตอบที่ดีที่สุดที่รู้มาเป็นฐานในการคำนวณแต่อย่างไรก็ตาม วิธีการค้นหาคำตอบใกล้เคียงที่ใช้การสุ่มจุดอาจไม่สามารถให้คำตอบที่ดีที่สุด

เสมอไป ดังนั้น บทความนี้จึงได้เสนอการใช้เทคนิคเอสพีเอส (Shortest Path Search Technique: SPS)ในการปรับวิถีการค้นหาคำตอบแทนการสุ่ม

สำหรับภาพรวมของการทำงานของขั้นตอน-วิธีที่นำเสนอแสดงดัง Figure 1 และรหัสเทียม (Pseudo code) แสดงดัง Table 2

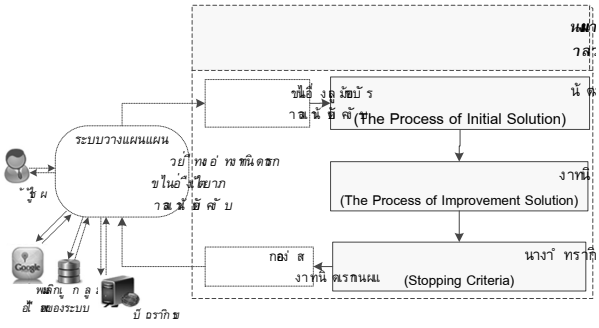

Figure 1 Overview of SA_SPS for the travel itinerary planner under time constraints

ในส่วนปฏิสัมพันธ์กับผู้ใช้เพื่อนำเข้าข้อมูล ดึงข้อมูลด้านเวลา และอธิบายแผนการเดินทาง (Figure 1) จะใช้ความสามารถจากระบบวางแผนแผนการเดินทางท่องเที่ยวใด ๆ ที่อยู่ภายใต้เงื่อนไขบังคับด้านเวลาจากนั้นนำข้อมูลเข้าสู่กระบวนการสร้างแผนการเดินทางเบื้องต้นและกระบวนการปรับปรุงแผนการเดินทางตามหลักการทำงานพื้นฐานของขั้นตอนวิธีจำลองการอบเหนียวโดยมีกระบวนการยุติการทำงานคอยตรวจสอบความเหมาะสมในการยุติ โดยรายละเอียดของกระบวนการทำงานมี 3 กระบวนการหลัก มีดังต่อไปนี้

1. กระบวนการสร้างแผนการเดินทางเบื้องต้น (The Process of Initial Solution)

เมื่อรับข้อมูลนำเข้าจากผู้ใช้แล้ว จะเข้าสู่การสร้างเงื่อนไขบังคับด้านเวลาทั้งหมดและกำหนดพารามิเตอร์ที่เกี่ยวข้องตามแนวคิดของขั้นตอนวิธีการจำลองการอบเหนียว (อธิบายเพิ่มเติมในหัวข้อการกำหนดพารามิเตอร์) จากนั้นเริ่มการค้นหาแผนการเดินทางเบื้องต้นจากการที่ปัญหาในงานวิจัยครั้งนี้มีปริภูมิการค้นหาขนาดใหญ่ การใช้วิธีสุ่มในการสร้างคำตอบเบื้องต้นไม่สามารถทำให้ได้คำตอบเบื้องต้นที่ใกล้เคียงกับคำตอบที่ดีที่สุด จึงได้เสนอการใช้เทคนิคเอสพีเอสในการหาลำดับการผ่านสถานที่แทนการสุ่ม

Table 2 The pseudo code of SA_SPS for travel itinerary planning under time constraints

```
SA_SPS(place demands, time constraints)
create constraints, initialize parameter
generate the initial solution as current solution (S) by using SPS_Technique ()
  while stopping criteria not met do //improve solution
    assign Layer for the places in S,based on the least position, start swap on the first Layer
    for i=1 to max Iter do
      while accept the S′ as Snotmet do
        generate a new candidate solution (S′) based on S, by using the priority queue in SPS_Technique () for selecting remaining places to swap with the place atthe current Layer
        S will be swapped as a new solution (S′)
        If check constraints f(S′) valid then
          get a feasible solution
          if S′ is better than the current one then
            accept S′ as S
            accept the f(S′) as current Cost of S, C(S)
            modified a heuristic f, h, g of places in priority queue
          else
            accept the new solution through the acceptance probability function
          end if
        else
          go to generate S′ again
        end if
      end while
      Add_remaining_place () by using SPS_Technique () for push remaining places into S as new solution (S″)
      If check constraints f(S″) valid then
        get a new feasible solution
        if S″ is better than the best one then
          replace the current solution with this new one
        end if
        update the best solution when the solution is better than it
      end if
    end for
  set cooling schedule, Iter=Iter+1, Layer= Layer+1
  endwhile
return the best solution
```

เนื่องจากคำตอบเบื้องต้นที่ดีนั้น มักจะเป็นกุญแจสำคัญในการเพิ่มประสิทธิภาพโดยรวมของวิธีการแบบเมตาฮิวริสติก[9]สำหรับการเลือกสถานที่ให้มาอยู่ในแผนการเดินทางจะมีการตรวจสอบให้อยู่ในเงื่อนไขและพิจารณาความเป็นไปได้ในการเดินทางกลับสู่จุดเริ่มต้นเสมอ ผลลัพธ์ที่ได้จากกระบวนการนี้คือ แผนการเดินทางเบื้องต้น1 แผนการเดินทางที่คาดว่าเยือนสถานที่ได้หลายแห่งมากที่สุดและใช้เวลาของแผนการเดินทางน้อยที่สุดภายในกรอบเวลาที่ผู้ใช้กำหนด

เทคนิคเอสพีเอส เป็นเทคนิคการหาลำดับการผ่านสถานที่ในแผนการเดินทางที่พิจารณาเลือกสถานที่จากระยะเวลาการเดินทางที่น้อยที่สุดจากสถานที่ปัจจุบันและต้องเป็นสถานที่ที่ยังไม่เคยไปมาก่อนให้มาอยู่ในแผนการเดินทาง เนื่องจากในปัญหาการวิจัยครั้งนี้ที่ต้องการเยี่ยมสถานที่ให้ได้หลายแห่งมากที่สุดภายในกรอบเวลาที่จำกัด ในการเลือกแผนการเดินทางเบื้องต้นที่ใช้ระยะเวลาในการเดินทางโดยรวมน้อยที่สุดก็มีความน่าจะเป็นที่ทำให้ได้แผนการเดินทางที่ประกอบไปด้วยสถานที่ท่องเที่ยวหลายแห่งมากที่สุด ดังนั้นการที่เลือกสถานที่ที่อยู่ใกล้ที่สุดกับสถานที่ที่อยู่ปัจจุบันแล้วเดินทางไปสถานที่นั้นก่อน เมื่อไปถึงสถานที่นั้นแล้วค่อยทำในทำนองเดียวกันอีกว่าจะเดินทางไปยังสถานที่ที่ใกล้ที่สุดสถานที่ถัดไป ทำเช่นนี้ก็น่าจะได้แผนการเดินทางเบื้องต้นที่มีสถานที่หลายแห่งมากที่สุด

โดยเทคนิคเอสพีเอสนี้ ประยุกต์มาจากแนวทางการค้นหาของขั้นตอนวิธีเอสตาร์ (A* Algorithm) และได้ปรับแต่งฟังก์ชันฮิวริสติกให้มีความเหมาะสมเข้ากับรูปแบบปัญหาดังแสดงในสมการที่ 2 ภาพรวมของเทคนิคเอสพีเอสแสดงดัง (Figure 2) และหลักการทำงานอธิบายด้วยรหัสเทียมดัง (Table 3)

$$f(x) = h(x_1,x_2,x_3,\ldots,x_n) + g(x_{1,n}) \quad (2)$$

เมื่อ f คือ ค่าฮิวริสติกของเวลาในการเดินทางระหว่างสถานที่จนกลับไปยังจุดเริ่มต้น

h คือ ค่าฮิวริสติกของเวลาที่ใช้ในการเดินทางจากสถานที่ปัจจุบันถึงสถานที่เป้าหมาย

g คือ ค่าฮิวริสติกของเวลาที่ใช้ในการเดินทางจากสถานที่สุดท้ายกลับไปยังสถานที่เริ่มต้น

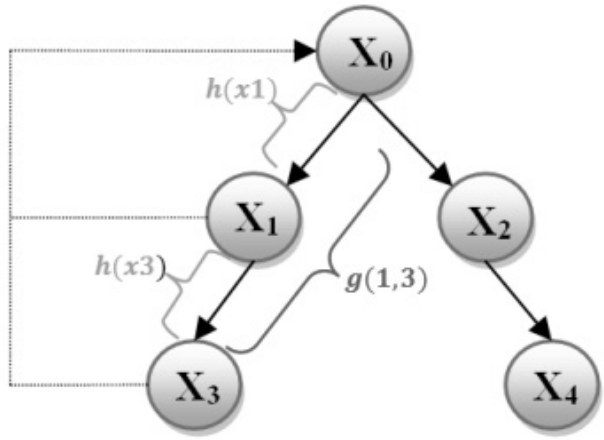

Figure 2 An overview of SPS technique.

Table 3 The pseudo code for SPS technique

```
SPS_Technique(Itinerary)
prioritize all places x in priority queue based on the least value of f(x),
sort ascending
  while stopping criteria not met do
    fori=1 to count_all_placesdo
      pop the first place x from priority queue which is not visited
      push the place x into the Route, modified Route
      f(route)=f(x) +Visiting Time
      If check constraints f(route) valid then
          accept the Route as current solution (S)
          accept the f(route) as current Cost of S, C(S)
          modified a heuristic f, h, g of all places in priority queue
        else
          go to pop the first place x from priority queue again
      end if
    end for
  endwhile
return generated solution, priority queue
```

Table 4 An example of assigning places into initial itinerary

Iteration	Candidate Solution	f(x)	S
1	0-1-0	205	~~0-1-0~~
	0-2-0	139	~~0-2-0~~
	0-3-0	178	~~0-3-0~~
	0-4-0	***74***	**0-4-0 P**
2	0-4-1-0	258	~~0-4-1-0~~
	0-4-2-0	***150***	**0-4-2-0 P**
	0-4-3-0	247	~~0-4-3-0~~
3	0-4-2-1-0	328	~~0-4-2-1-0~~
	0-4-2-3-0	***321***	**0-4-2-3-0 P**

ในการทำงานจะเริ่มต้นสร้างเส้นทางการเดินทางที่สถานที่ x_0 จากนั้นใช้แถวคอยลำดับความสำคัญ (Priority Queue) จัดอันดับให้สถานที่ x ที่เหลืออยู่ โดยสถานที่ x ใดมีค่า $f(x)$ น้อยกว่าจะถือว่ามีความสำคัญมากกว่า ในแต่ละรอบของการทำงาน จะเลือกสถานที่ x ที่ไม่เคยไปมาก่อนและมีค่า $f(x)$ ต่ำที่สุดให้มาอยู่ในเส้นทาง (Route) จากนั้นคำนวณค่า *f(route)* ซึ่งหมายถึงค่าของเวลาทั้งหมดของแผนการเดินทางในขณะนั้น โดยบวกระยะเวลาที่แวะเที่ยว ณ สถานที่นั้น (Visiting Time) เข้าไปด้วยในการยอมรับให้สถานที่ x มาอยู่ในเส้นทาง จะต้องตรวจสอบกับเงื่อนไขบังคับด้านเวลาของแผนการเดินทางหากอยู่ในเงื่อนไขบังคับก็จะยอมรับให้เป็นคำตอบปัจจุบัน (S) จากนั้นปรับค่าฮิวริสติก f, h และ g ของสถานที่ที่จะถูกพิจารณาต่อและปรับคิวในถอยคอย แต่หากไม่อยู่ใน

เงื่อนไขบังคับก็จะวนกลับไปสร้างเส้นทางใหม่ด้วยการเลือกสถานที่ *x* ที่อยู่ในแถวคอยในลำดับถัดไปขึ้นมาจนกว่าจะได้แผนการเดินทางที่เหมาะสม กล่าวคือ หยุดเมื่อค่า *f*(*route*) มีค่ามากกว่าค่าเวลาของแผนการเดินทาง หรือเมื่อแถวคอยไม่มีสถานที่แล้ว

ตัวอย่างการประมวลผลเพื่อค้นหาเส้นทางที่เป็นไปได้เบื้องต้นตามเทคนิคเอสพีเอสแสดงดัง (Table 4) ซึ่งหลังจากจบการประมวลผลจะได้แผนการเดินทางเบื้องต้น (*S*) ซึ่งอธิบายได้ว่า เริ่มต้นเดินทางออกจากสถานที่ 0 ไปยังสถานที่ 4 2 และ 3 ตามลำดับ จากนั้นสิ้นสุดการเดินทางและกลับมายังจุดเริ่มต้นสถานที่ 0เวลาของแผนการเดินทางเท่ากับ 321 นาที

2. กระบวนการปรับปรุงแผนการเดินทาง (The Process of Improvement Solution)

กระบวนการนี้เป็นกระบวนการหลักที่สำคัญที่จะทำให้ได้มาซึ่งคุณภาพของคำตอบที่ดีกว่าเดิม ประกอบด้วยการทำงาน3 ส่วนหลักดังนี้

1) ส่วนการค้นหาคำตอบใกล้เคียง

เป็นขั้นตอนการค้นหาคำตอบใหม่ (*S′*) ที่น่าจะดีกว่าเดิม ซึ่งใช้หลักในการสับเปลี่ยน[12](Swap) มาช่วยในการค้นหาซึ่งอาจเปลี่ยนทิศทาง (Vector) ในการค้นหาให้ชี้ไปสู่คำตอบที่ดีที่สุดได้ สำหรับการเลือกสถานที่ที่จะมาสับเปลี่ยนนั้น ได้เสนอการใช้เทคนิคเอสพีเอสแทนการสุ่ม โดยมีขั้นตอนการทำงานดังต่อไปนี้

ขั้นตอนที่ 1: *เริ่มสร้างคำตอบใหม่ S′* โดยใช้คำตอบปัจจุบัน *S* เป็นฐานในการคำนวณสับเปลี่ยน

ขั้นตอนที่ 2: *สับเปลี่ยนคำตอบใน S เพื่อค้นหา S′* ตามหลักการพื้นฐานของการค้นหาคำตอบใกล้เคียงที่ว่า คำตอบใหม่คือคำตอบใกล้เคียงที่อยู่รอบ ๆ คำตอบปัจจุบัน ดังนั้น สถานที่ที่จะสับเปลี่ยนกันจึงต้องเป็นสถานที่ที่บรรจุอยู่ใน *S* แล้วโดยเริ่มการสับเปลี่ยนจากสถานที่ *x* ในระดับที่ 1 หรือ *Layer 1* ก่อน (แทนด้วย *L1*) ในการเลือกว่าสถานที่ *x* นั้นจะสับเปลี่ยนกับสถานที่ใด จะใช้เทคนิคเอสพีเอสในการเลือก (ทำเหมือนกับกระบวนการสร้างแผนการเดินทางเบื้องต้น) กล่าวคือ เลือกสถานที่ที่ใช้เวลาเดินทางน้อยที่สุดจากสถานที่ที่อยู่ในปัจจุบันมาสับเปลี่ยน โดยพิจารณาค่า *h*(*x*) ที่น้อยที่สุดในขณะนั้น หากสถานที่ที่อยู่ในแถวคอยใน *L1* หมดให้ย้ายไปสับเปลี่ยนคำตอบที่ *L2, L3* จนถึง *Ln* ต่อไปสุดท้ายเมื่อการสับเปลี่ยนสำเร็จก็จะได้เส้นทางที่เป็นคำตอบใหม่ *S′*

ขั้นตอนที่ 3: *ตรวจสอบ S′ว่าอยู่ในเงื่อนไขบังคับด้านเวลาใช่หรือไม่* หากใช่ ให้ไปขั้นตอนที่ 4 แต่หากไม่ใช่ให้คงคำตอบไว้ที่ *S* เหมือนเดิม และทำซ้ำขั้นตอนที่ 2 ต่อไปจนกว่าจะได้ *S′* ที่อยู่ในเงื่อนไขบังคับ

ขั้นตอนที่ 4: *พิจารณาการยอมรับ S′ ให้เป็นคำตอบปัจจุบัน* ซึ่ง *S′*จะถูกยอมรับได้หากคำตอบใหม่ดีกว่าเดิมหรือมีค่าความน่าจะเป็นในการเคลื่อนย้ายที่เหมาะสม (อธิบายเพิ่มเติมในหัวข้อขั้นตอนการพิจารณาการยอมรับคำตอบ)หากยอมรับให้ *S′* เป็น S ให้ปรับปรุงค่าฮิวริสติก *f*, *h*, *g*และลำดับในแถวคอยและไปยังขั้นตอนที่ 5 แต่หากไม่ยอมรับให้คงคำตอบไว้ที่ *S* เหมือนเดิมและวนกลับไปทำขั้นตอนที่ 2 ใหม่

ขั้นตอนที่ 5: *ค้นหา S′′* ก่อนที่จะเคลื่อนย้ายคำตอบ ส่วนพิเศษเพื่อค้นหาเส้นทางที่อาจไปเยือนสถานที่ได้มากขึ้นจะทำงาน ซึ่งส่วนพิเศษนี้จะค้นหาเส้นทาง *S′′* ที่เยือนสถานที่ได้มากกว่าคำตอบเดิม (อธิบายเพิ่มเติมในหัวข้อส่วนพิเศษเพื่อค้นหาเส้นทางที่อาจไปเยือนสถานที่ได้มากขึ้น) จากนั้นตรวจสอบว่า *S′′* อยู่ในเงื่อนไขใช่หรือไม่ หากใช่ ก็จะยอมรับให้ *S′′* เป็น *S* แต่หากไม่ใช่ ให้คงคำตอบไว้ที่เดิม

ขั้นตอนที่ 6: *บันทึกเส้นทางคำตอบ*ในตัวแปรที่ใช้ในการจัดเก็บคำตอบ จากนั้นทำซ้ำขั้นตอนที่ 1-5 จนกว่าระบบจะเข้าสู่สภาวะสมดุลหรือครบจำนวนครั้งของการทำซ้ำที่ถูกกำหนดไว้ล่วงหน้า ขั้นตอนนี้จะทำให้มีคำตอบที่เข้าใกล้คำตอบที่เหมาะสมที่สุดเฉพาะที่ (Local Optimum)

ขั้นตอนที่ 7: ปรับลดค่าพารามิเตอร์ *T* ตามฟังก์ชัน $\alpha(T)$ ที่ได้กำหนดไว้ตามตารางการจัดการการอบเหนียว (Set Cooling Schedule) แล้วเริ่มทำซ้ำ ขั้นตอนที่ 1-5 ใหม่อีกครั้ง กระบวนการยุติเมื่อตรวจสอบแล้วพบว่าเงื่อนไขใดเงื่อนไขหนึ่งในกระบวนการยุติการทำงานเป็นจริง

2) ส่วนพิเศษเพื่อค้นหาเส้นทางที่อาจไปเยือนสถานที่ได้มากขึ้น

ในปัญหาวิจัยครั้งนี้ที่มีการคำนึงถึงเงื่อนไขบังคับด้านเวลา ส่งผลให้บางแผนการเดินทางไม่สามารถเยือนทุกสถานที่ตามที่ผู้ใช้ระบุได้เพราะใช้เวลาของแผนการเดินทางเกินกรอบที่กำหนด เพื่อให้แผนยังอยู่ในเงื่อนไขบังคับด้านเวลา ขั้นตอนวิธีจึงต้องพิจารณาตัดบางสถานที่ออกทำให้ในแต่ละรอบของการวนซ้ำมีบางสถานที่ที่ยังเหลืออยู่และยังไม่ถูกบรรจุในแผนการเดินทาง (แทนด้วยสถานที่ *y*) ดังนั้น จึงได้เสนอส่วนพิเศษเพื่อค้นหาเส้นทางที่อาจไปเยือนสถานที่ได้มากขึ้น ซึ่งอยู่ในขั้นตอนที่ 5 ของส่วนการค้นหาคำตอบใกล้เคียง (Table 2; procedure Add_remaining_place ()) วัตถุประสงค์คือ เพื่อเพิ่มคุณภาพของคำตอบจากการที่แผนการเดินทางสามารถเยือนสถานที่ได้หลายแห่งมากขึ้นนั่นเอง

หลักการทำงานใช้หลักการสำรวจว่า หากมีสถานที่*y* ที่ยังเหลืออยู่และยังไม่ถูกบรรจุใน *S* แล้ว สามารถ

เพิ่มสถานที่ *y* ต่อท้าย (Push) เข้าไปใน *S*ได้อีกหรือไม่ หลักในการพิจารณาเลือกสถานที่ *y* ใช้เทคนิคเอสพีเอสในการเลือก กล่าวคือ เลือกสถานที่ *y* จากค่า *f*(*y*) ที่มีค่าน้อยที่สุด ซึ่งส่วนพิเศษนี้จะสร้างเส้นทางคำตอบใหม่อีกหนึ่งเส้นทางนั่นคือ *S''* ที่เยื้องสถานที่ได้มากกว่าเส้นทางคำตอบเดิม

3) ส่วนการพิจารณายอมรับคำตอบ

ส่วนการทำงานนี้เป็นแนวคิดพื้นฐานของขั้นตอนวิธีการจำลองการอบเหนียวเป็นส่วนที่ใช้พิจารณาว่าควรจะใช้คำตอบใหม่เป็นคำตอบหรือสถานะใหม่ของระบบหรือไม่ โดยการเปรียบเทียบคำตอบเพื่อที่จะกระจายความน่าจะเป็นในการเปลี่ยนสถานะของระบบไป ซึ่งทำให้สามารถหลีกเลี่ยงปัญหาการตกอยู่ในคำตอบที่ไม่ขัดแย้งต่อเงื่อนไขแต่มีคุณภาพของคำตอบไม่ดีหรือที่เรียกว่าคำตอบที่เหมาะสมที่สุดเฉพาะที่ โดยส่วนการทำงานนี้จะแทรกอยู่ในทุก ๆ การสร้างเส้นทางคำตอบใหม่ที่เป็นไปได้ขึ้นมา (ทั้ง *S'*และ *S''*) หลักการทำงานคือ ในการที่จะพิจารณายอมรับให้ *S'*หรือ *S''*เป็นคำตอบ *S* ของระบบจะทำโดยเปรียบเทียบค่าความแตกต่างของต้นทุน (Cost) (ในที่นี้ต้นทุน คือจำนวนสถานที่ท่องเที่ยวที่อยู่ในแผนการเดินทาง) แทนด้วย Δ*E* ซึ่งหาได้จากสมการที่ 3

$$\Delta E = C(S) - C(S') \quad (3)$$

เมื่อ *C*(*S*) คือ จำนวนสถานที่ท่องเที่ยวที่อยู่ในแผนการเดินทางปัจจุบัน

(*S'*) คือ จำนวนสถานที่ท่องเที่ยวที่อยู่ในแผนการเดินทางใหม่

Δ*E* คือ ค่าความแตกต่างระหว่างจำนวนสถานที่

ถ้า Δ*E*< 0 ก็จะยอมรับให้คำตอบใหม่เป็นคำตอบปัจจุบันแทน และย้ายสถานะการค้นหาไปที่จุดที่คำตอบนั้นอยู่ แต่ถ้าคำตอบใหม่ไม่ดีกว่าคำตอบปัจจุบันก็อาจจะถูกยอมรับหรือไม่ยอมรับก็ได้โดยพิจารณาจากค่าความน่าจะเป็นในการยอมรับคำตอบใหม่ (Acceptance Probability Function: *prob*(*ΔE*))ซึ่งหาได้จากสมการที่ 4 เปรียบเทียบกับค่าความน่าจะเป็นของโบลทซ์มันน์ (Boltzmann's Probability: β) ที่ได้จากการสุ่มตัวเลข βในช่วง 0 ถึง 1 ขึ้นมา

$$prob(\Delta E) = e^{-\Delta E/kT} \quad (4)$$

เมื่อ *e* คือ มีค่าประมาณ 2.718

k คือ ค่าคงที่ของโบลท์ซมันน์

T คือ ค่าอุณหภูมิของระบบ

ถ้า *prob*(*ΔE*) >β[0,1] แล้ว จะยอมรับให้ *S'* เป็นคำตอบปัจจุบัน แต่หากไม่ใช่ให้คงคำตอบไว้ที่ *S*เหมือนเดิม โดยที่ *prob*(*ΔE*) ขึ้นอยู่กับ *ΔE* ในขณะนั้นกับค่าพารามิเตอร์อุณหภูมิ *T* ในช่วงแรกของการค้นหา ความน่าจะเป็นในการยอมรับคำตอบที่แย่กว่าจะมีมาก แต่เมื่อค้นหาไประยะหนึ่งค่า *T*ถูกปรับลดลงความน่าจะเป็นในการยอมรับคำตอบใหม่ที่แย่กว่าคำตอบปัจจุบันที่สร้างขึ้นในระยะหลังมักจะไม่ถูกยอมรับจนนำไปสู่การยุติการทำงานต่อไป

3. กระบวนการยุติการทำงาน (Stopping Criteria)

ในกระบวนการนี้ จะมีการพิจารณาการยุติการทำงานซึ่งจะยุติเมื่อตรวจสอบแล้วพบว่าอุณหภูมิสุดท้ายเท่ากับ 0.01 หรือจำนวนครั้งของการค้นหาคำตอบที่ไม่มีการปรับปรุงค่าคำตอบให้ดีขึ้นเท่ากับ 200 ครั้ง หรือเมื่อค้นหาทุกเส้นทางแล้ว (อธิบายเพิ่มเติมในหัวข้อ พารามิเตอร์กำหนดการหยุดทำงาน)

การออกแบบการทดลอง

สำหรับการออกแบบการทดลองประกอบด้วยการกำหนดพารามิเตอร์ที่ใช้ในการทดลอง วิธีการทดลอง และเครื่องมือที่ใช้ในการทดลองซึ่งมีรายละเอียดดังต่อไปนี้

การกำหนดพารามิเตอร์ที่ใช้ในการทดลอง

ในการทดลองกับเทคนิคที่พัฒนามาจากพื้นฐานของวิธีการจำลองการอบเหนียวจำเป็นต้องมีการกำหนดค่าพารามิเตอร์ที่เกี่ยวข้องซึ่งจะใช้สำหรับขั้นตอนวิธีที่นำเสนอและขั้นตอนวิธีการจำลองการอบเหนียวดั้งเดิม การกำหนดพารามิเตอร์นี้ประกอบด้วย

1. พารามิเตอร์กำหนดการลดลงของอุณหภูมิ α (T)

โดยที่ α เป็นค่าคงที่ที่มีค่าใกล้เคียงและน้อยกว่า 1 และ *T* คืออุณหภูมิ ซึ่ง α นี้ใช้กำหนดการปรับลดลงของอุณหภูมิและจำนวนครั้งของการค้นหาคำตอบ ถ้า α มีค่าน้อย การลดลงของ *T*จะไวขึ้น ทำให้การลู่เข้าสู่คำตอบเร็วในทางตรงข้ามเมื่อ α มีค่ามาก การลดลงของ *T*ก็จะช้าลง ส่งผลให้เวลาการค้นหาคำตอบก็จะเพิ่มมากขึ้นจากการสังเกตการทดลองเบื้องต้น พบว่า เมื่อพื้นที่การค้นหามีขนาดใหญ่ขึ้นแต่ยังใช้เวลาเพียงไม่กี่ไมโครวินาที ดังนั้นเวลาไม่กี่ไมโครวินาทีจึงไม่เป็นปัญหา จึงกำหนดค่า α เท่ากับ0.9[13]หรือ $\alpha(T) = 0.9 * \alpha(T_{-1})$

2. พารามิเตอร์อุณหภูมิเริ่มต้น T_0

ใช้กำหนดความน่าจะเป็นของการยอมรับคำตอบที่ได้จากการเริ่มแก้ไขปัญหาด้วยกระบวนการจำลองการอบเหนียว กำหนดT_0ที่ 0.8[14]

*3. พารามิเตอร์กำหนดการหยุดทำงาน*กำหนดใช้เพื่อพิจารณาการยุติการทำงานดังต่อไปนี้

1) ยุติการทำงานเมื่อพารามิเตอร์อุณหภูมิสุดท้าย T_{min} เท่ากับ 0.01เมื่อมีการปรับค่า *T*ตาม α (*T*) ค่า *T* จะถูกลดต่ำลงเรื่อย ๆ ในการทดลองนี้ จึงกำหนดใช้ T_{min} เพื่อเป็นเกณฑ์ในการหยุดการทำงานของระบบ โดยกำหนด T_{min} ที่ 0.01 ให้เป็นอุณหภูมิสุดท้ายซึ่งเป็นค่าที่เพียงพอแล้วก่อนที่ *T* ลดต่ำลงจนมีค่าเป็นศูนย์[14]

2) ยุติการทำงานเมื่อพารามิเตอร์ Max_noimprove เท่ากับ 200 ครั้งโดยปกติแล้วจำนวนครั้งของการค้นหาคำตอบที่ไม่มีการปรับปรุงค่าคำตอบให้ดีขึ้นหรือ *Max_noimprove* จะถูกกำหนดไว้ก่อนล่วงหน้าและต้องเพียงพอกับจำนวนครั้งในการวนรอบค้นหาคำตอบแต่ต้องไม่มากจนเกินไป[13] เพราะจะส่งผลต่อเวลาในการประมวลผลให้มากขึ้น ในการทดลองนี้ด้วยเวลาในการค้นหาคำตอบที่ไม่ได้แตกต่างกันมากนักแต่ยังสามารถให้คุณภาพคำตอบที่ดีที่สุดหรือใกล้เคียงที่ดีที่สุดได้ ดังนั้น จึงกำหนด *Max_noimprove* ที่ 200 ครั้ง กล่าวคือ หากการค้นหาไม่พบคำตอบที่ดีกว่าคำตอบที่ดีที่สุดที่รู้มา 200 ครั้ง ก็จะยุติกระบวนการทั้งหมด

3) ยุติการทำงานเมื่อค้นหาทุกเส้นทางที่เป็นไปได้แล้วโดยเมื่อค้นหาทุกเส้นทางที่เป็นไปได้แล้ว กระบวนการทั้งหมดก็สามารถยุติการทำงานทั้งหมดได้

วิธีการทดลอง

ในการทดลองเพื่อทดสอบประสิทธิภาพทำโดยเปรียบเทียบกับขั้นตอนวิธีค้นหาทุกเส้นทางที่เป็นไปได้และขั้นตอนวิธีการจำลองการอบเหนียวดั้งเดิม ในปัญหาการวางแผนการเดินทางท่องเที่ยวที่ต้องการเดินทางไปเยือนสถานที่หลายแห่งมากที่สุดภายในระยะเวลาจำกัด ซึ่งทำโดยการประมวลผลขั้นตอนวิธีละ 100 ครั้งตามข้อมูลและพารามิเตอร์ที่ภายใต้เงื่อนไขบังคับด้านเวลาเดียวกันในแต่ละสถานการณ์จำลองซึ่งสร้างจากจำนวนสถานที่ที่แตกต่างกันตั้งแต่ 3-25 แห่ง ตัวอย่างข้อมูลที่ใช้ในการทดลองในสถานการณ์ที่มีจำนวนสถานที่ 3 แห่งแสดงดัง (Table 5)

เนื่องจากแผนการเดินทางที่ได้จากการประมวลผล 100 ครั้งของแต่ละขั้นตอนวิธีมีความแตกต่างกันกล่าวคือ ในสถานการณ์จำลองเดียวกัน

การประมวลผลครั้งแรกใช้เวลาในการประมวลผล 0.05 วินาที โดยมีจำนวนสถานที่ที่ไปได้มากที่สุดคือ 15 แห่ง ในขณะที่การประมวลผลครั้งที่สองใช้เวลาในการประมวลผล 0.04 วินาที โดยมีจำนวนสถานที่ที่ไปได้มากที่สุดคือ 12 แห่ง ซึ่งถือว่าแผนการเดินทางดังกล่าวมีความแตกต่างกัน ดังนั้นเพื่อให้การวิเคราะห์ข้อมูลเป็นไปอย่างถูกต้อง ในการทดลองของงานวิจัยนี้จึงเก็บผลการทดลองเพื่อการวิเคราะห์จาก 3 รูปแบบ ได้แก่ จากกรณีเฉลี่ย (Average Case) จากกรณีที่ดีที่สุด (Best Case) และจากกรณีที่แย่ที่สุด (Worse Case) จากนั้นนำผลการทดลองไปวิเคราะห์ใน ด้านเวลาในการประมวลผล ซึ่งหากขั้นตอนวิธีใดใช้เวลาในการประมวลผลน้อยกว่าจะถือว่าขั้นตอนวิธีนั้นมีประสิทธิภาพดีกว่าและด้านคุณภาพของคำตอบที่หากขั้นตอนวิธีใดประมวลผลแล้วได้จำนวนสถานที่หลายแห่งมากกว่าจะถือว่าขั้นตอนวิธีนั้นมีประสิทธิภาพดีกว่า โดยกำหนดการทดลองในแต่ละรูปแบบดังนี้

Table 5 An example of dataset with 3 places.

	Travel Time between Place (Sec)				Visiting Time(Sec)
Place	0	1	2	3	
0	0	100	69	90	0
1	105	0	143	187	120
2	70	140	0	153	35
3	85	185	165	0	60

รูปแบบที่ 1 การวิเคราะห์จากกรณีเฉลี่ย

ด้านเวลาในการประมวลผล วิเคราะห์จากค่าเฉลี่ยของเวลาที่ใช้ในการประมวลผล*ด้านคุณภาพของคำตอบ* วิเคราะห์จากค่าเฉลี่ยของจำนวนสถานที่ที่อยู่ในแผนการเดินทาง ซึ่งทั้งสองด้านวิเคราะห์จากการประมวลผลของแต่ละขั้นตอนวิธีโดยเฉลี่ยจากการคำนวณ 100 ครั้ง ในแต่ละสถานการณ์จำลอง

รูปแบบที่ 2 การวิเคราะห์จากกรณีที่ดีที่สุด

*ด้านเวลาในการประมวลผล*วิเคราะห์จากเวลาที่ใช้ในการประมวลผลของแต่ละขั้นตอนวิธีเมื่อพบแผนการเดินทางที่ให้จำนวนสถานที่หลายแห่งมากที่สุดจากการประมวลผล 100 ครั้ง หากพบคำตอบที่ดีที่สุดหลายครั้งจะเลือกจากครั้งที่ใช้เวลาในการประมวลผลน้อยที่สุดสำหรับ*ด้านคุณภาพของคำตอบ*วิเคราะห์จากจำนวนสถานที่ที่ได้จากการประมวลผลของแต่ละขั้นตอนวิธีเมื่อพบแผนการเดินทางที่ให้จำนวนสถานที่หลายแห่งมากที่สุดจากการประมวลผล 100 ครั้ง

รูปแบบที่ 3 การวิเคราะห์จากกรณีที่แย่ที่สุด

*ด้านเวลาในการประมวลผล*วิเคราะห์จากเวลาที่ใช้ในการประมวลผลของแต่ละขั้นตอนวิธีเมื่อพบแผนการเดินทางที่ให้จำนวนสถานที่น้อยแห่งที่สุดจากการประมวลผล 100 ครั้ง หากพบคำตอบที่แย่ที่สุดหลายครั้งจะวิเคราะห์จากครั้งที่ใช้เวลาในการประมวลผลมากที่สุดสำหรับ*ด้านคุณภาพของคำตอบ* วิเคราะห์จากจำนวนสถานที่ที่ได้จากการประมวลผลของ

แต่ละขั้นตอนวิธีเมื่อพบแผนการเดินทางที่ให้จำนวนสถานที่น้อยแห่งที่สุดจากการประมวลผล 100 ครั้ง

เครื่องมือที่ใช้ในการทดลอง

ในการเขียนโปรแกรมเพื่อทดสอบประสิทธิภาพของขั้นตอนวิธี เขียนด้วยภาษาPHP3ประมวลผลการทดลองบนเครื่องคอมพิวเตอร์ที่มีหน่วยประมวลผลกลางชนิด Intel(R) Core(TM) i5 M560 ความถี่ 2.76 กิกะเฮิร์ทหน่วยความจำสำรองขนาด 4 กิกะไบท์ ภายใต้ระบบปฏิบัติการ Window 7 แบบ 32 บิต

ผลการทดลอง

ผลการทดลองเพื่อทดสอบประสิทธิภาพของขั้นตอนวิธีที่เสนอปรากฏใน (Table 6-8) และ (Figure 3-11) โดยที่ *place* หมายถึง จำนวนสถานที่ *SA_SPS* หมายถึง ขั้นตอนวิธีที่นำเสนอ *SA* หมายถึง ขั้นตอนวิธีการจำลองการอบเหนียวดั้งเดิม *ER* หมายถึง ขั้นตอนวิธีค้นหาทุกเส้นทางที่เป็นไปได้ *S*หมายถึง ผลลัพธ์ (จำนวนสถานที่ท่องเที่ยวที่มากที่สุดที่พบในแผนการเดินทางท่องเที่ยวที่แต่ละขั้นตอนวิธีคำนวณได้) Sec_{tot} หมายถึง เวลาในการประมวลผลของแต่ละขั้นตอนวิธี *%gap S* หมายถึง ร้อยละความแตกต่างของ *S* ระหว่างขั้นตอนวิธีที่นำเสนอกับขั้นตอนวิธีอื่นที่นำมาเปรียบเทียบ และ *%gap* Sec_{tot} หมายถึง ร้อยละความแตกต่างของ Sec_{tot} ระหว่างขั้นตอนวิธีที่นำเสนอกับขั้นตอนวิธีอื่นที่นำมาเปรียบเทียบ

ผลการวิเคราะห์จากกรณีเฉลี่ย

จากการทดลองเพื่อที่จะวิเคราะห์ข้อมูลจากกรณีเฉลี่ยทั้งหมดที่เกิดขึ้น ปรากฏผลการทดลองดัง (Table 6) และ (Figure 3-4) รายละเอียดดังต่อไปนี้

ด้านเวลาในการประมวลผล

ขั้นตอนวิธีที่นำเสนอ (SA_SPS) ใช้เวลาในการประมวลผลน้อยกว่าขั้นตอนวิธี SA และขั้นตอนวิธี ER โดยมีค่าเฉลี่ยของร้อยละความแตกต่างของเวลาที่ใช้ในการประมวลผล (*%gap* Sec_{tot}) น้อยกว่าโดยเฉลี่ยร้อยละ 13.83 และร้อยละ 99.95 ตามลำดับ (Table 6)

ด้านคุณภาพของคำตอบ

ขั้นตอนวิธี SA_SPS ให้คำตอบที่มีคุณภาพเท่ากับหรือดีกว่าขั้นตอนวิธี SA ในทุกสถานการณ์จำลองโดยมีค่าเฉลี่ยของร้อยละความแตกต่างของคุณภาพคำตอบ (*%gap S*) ที่ดีกว่าโดยเฉลี่ยร้อยละ 37.21 (Table 6)

สำหรับการเปรียบเทียบกับขั้นตอนวิธี ER พบว่าขั้นตอนวิธี SA_SPS ได้คำตอบที่มีคุณภาพแย่กว่าเล็กน้อย โดยมีค่า *%gap S* ที่น้อยกว่าโดยเฉลี่ยร้อยละ 9.48 (Table 6) สาเหตุเนื่องมาจาก ขั้นตอนวิธี ER เป็นการค้นหาทุกรูปแบบเส้นทางที่สามารถไปได้ จึงเป็นการค้นทั่วพื้นที่การค้นหาและให้คำตอบที่ดีที่สุดเสมอ ดังนั้น คำตอบที่ได้จากขั้นตอนวิธี ER จึงดีกว่า แต่ในทางกลับกัน การที่ขั้นตอนวิธี ER ต้องค้นหาทุกรูปแบบการเดินทางจึงใช้เวลาในการประมวลผลสูง และเมื่อมีการเพิ่มจำนวนสถานที่ รูปแบบการเดินทางที่เป็นไปได้ก็เพิ่มมากขึ้นเป็นหลายเท่าตัว ส่งผลให้เวลาที่ใช้ในการประมวลผลเพื่อสร้างรูปแบบที่เป็นไปได้ทั้งหมดสำหรับการค้นหาของขั้นตอนวิธี ER เพิ่มขึ้นเป็นอย่างมาก ด้วยข้อจำกัดของคอมพิวเตอร์ที่ใช้ในการทดลองที่ไม่สามารถประมวลผลได้ การทดลองของขั้นตอนวิธี ER จึงยุติการทดลองที่จำนวนสถานที่ 9 แห่ง

Table 6 Comparison results for the average case.

	S			*Sec*$_{tot}$			*%gap S*		*%gap Sec*$_{tot}$	
place	**SA_SPS**	**SA**	**ER**	**SA_SPS**	**SA**	**ER**	**SA**	**ER**	**SA**	**ER**
3	3	3	3	0.00378	0.00798	3.44355	0	0	52.69735	99.89035
4	4	4	4	0.00571	0.00971	5.34347	0	0	41.21134	99.89321
5	4.60	2.92	5	0.00552	0.00763	10.23446	36.52	-8.70	27.65979	99.94603
6	6	4.69	6	0.00865	0.01096	16.34231	21.83	0	21.07648	99.94706
7	6	4.75	7	0.00912	0.01057	54.34546	20.83	-16.67	13.75417	99.98322
8	6	3.82	8	0.01056	0.01300	208.45435	36.33	-33.33	18.73071	99.99493
9	8.36	4.89	9	0.01281	0.01373	685.34523	41.51	-7.66	6.69327	99.99813
10	9.69	7.5	-	0.01892	0.02001	-	22.60	-	5.43726	-
11	11	9.96	-	0.02145	0.03395	-	9.45	-	36.83880	-
12	10.95	5.94	-	0.03005	0.02510	-	45.75	-	-19.69004	-
13	12	7.12	-	0.03303	0.03132	-	40.67	-	-5.45971	-
14	14	11.34	-	0.04512	0.05751	-	19.00	-	21.53882	-
15	13	6.12	-	0.03694	0.03412	-	52.92	-	-8.24258	-
16	13.18	5.85	-	0.04259	0.03594	-	55.61	-	-18.51917	-
17	15.09	7.46	-	0.05050	0.05136	-	50.56	-	1.68204	-
18	15	7.48	-	0.04424	0.04961	-	50.13	-	10.82421	-
19	19	11.46	-	0.05190	0.08722	-	39.68	-	40.49529	-
20	19	8.94	-	0.06733	0.06721	-	52.95	-	-0.18363	-
21	21	12.48	-	0.07168	0.08350	-	40.57	-	14.16337	-
22	21	10.5	-	0.07561	0.07861	-	50.00	-	3.81536	-
23	21.50	9.929	-	0.07583	0.09550	-	53.82	-	20.59956	-
24	22.36	9.792	-	0.08867	0.12019	-	56.21	-	26.22071	-
25	23	9.46	-	0.12194	0.13063	-	58.87	-	6.65691	-
Avg.							**37.21**	**-9.48**	**13.82610**	**99.95042**

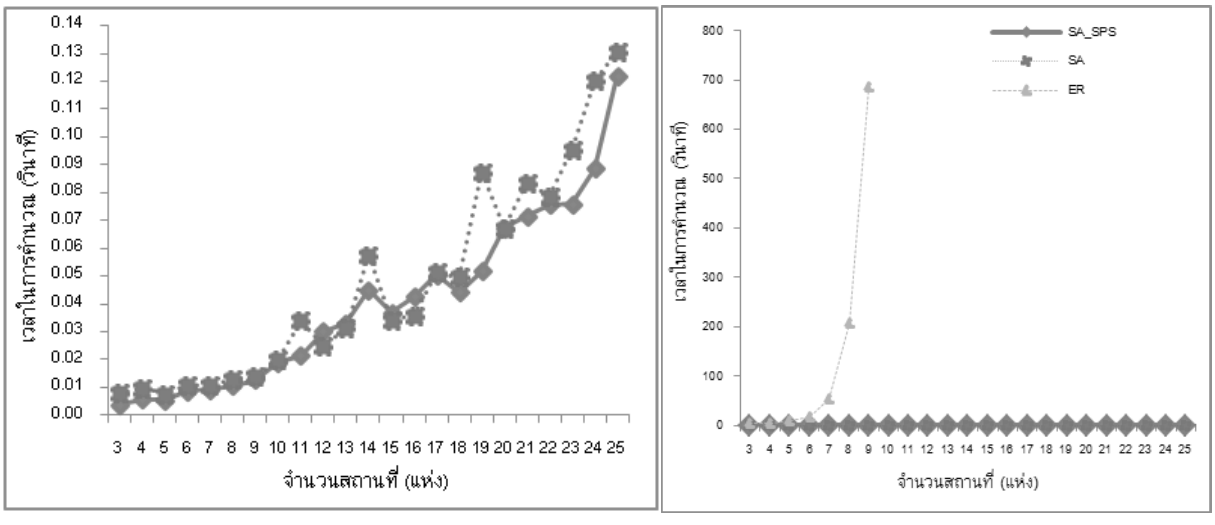

Figure 3 CPU time compare for the average case of SA_SPS, SA and ER.

ผลการวิเคราะห์จากกรณีที่ดีที่สุด

จากการทดลองเพื่อที่จะวิเคราะห์ข้อมูลจากกรณีที่พบแผนการเดินทางที่ให้คำตอบที่มีจำนวนสถานที่หลายแห่งมากที่สุด ปรากฏผลการทดลองดัง (Table 7) และ (Figure5-6) มีรายละเอียดดังต่อไปนี้

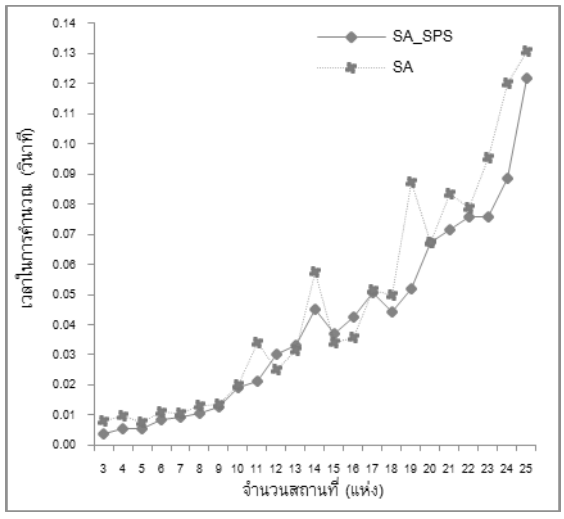

Figure 4 Extension fig.3for CPU time compare for the average case of SA_SPS and SA.

ด้านเวลาในการประมวลผล

ขั้นตอนวิธี SA_SPS ใช้เวลาในการประมวลผลน้อยกว่าขั้นตอนวิธี SA และขั้นตอนวิธี ER โดยมีค่าเฉลี่ยของร้อยละความแตกต่างของเวลาที่ใช้ในการประมวลผล ($\%gap\ Sec_{tot}$) น้อยกว่าโดยเฉลี่ยร้อยละ 38.58 และร้อยละ 99.97 ตามลำดับ (Table 7)

Table 7 Comparison results for the best case.

	S			***Sec_{tot}***			***%gap S***		***%gap Sec_{tot}***	
place	**SA_SPS**	**SA**	**ER**	**SA_SPS**	**SA**	**ER**	**SA**	**ER**	**SA**	**ER**
3	3	3	3	0.00164	0.00297	3.52845	0	0	44.75454	99.95344
4	4	4	4	0.00187	0.00265	5.74324	0	0	29.18715	99.96739
5	5	4	5	0.00220	0.00301	9.23436	20.00	0	26.94611	99.97622
6	6	5	6	0.00371	0.00454	15.23455	16.67	0	18.17981	99.97563
7	6	6	7	0.00434	0.00590	52.45657	0	-16.67	26.32740	99.99172
8	6	6	8	0.00569	0.01564	172.32535	0	-33.33	63.65090	99.99670
9	9	7	9	0.00739	0.01073	580.34230	22.22	0	31.14632	99.99873
10	10	10	-	0.01162	0.02540	-	0	-	54.25771	-
11	11	11	-	0.01236	0.13746	-	0	-	91.01193	-
12	11	9	-	0.01537	0.06286	-	18.18	-	75.54530	-
13	12	10	-	0.01660	0.03808	-	16.67	-	56.40985	-
14	14	14	-	0.02079	0.06387	-	0	-	67.45217	-
15	13	9	-	0.02153	0.02864	-	30.77	-	24.82369	-
16	14	9	-	0.02515	0.02141	-	35.71	-	-17.45202	-
17	16	11	-	0.03819	0.07022	-	31.25	-	45.60926	-
18	15	11	-	0.02926	0.04200	-	26.67	-	30.33593	-
19	19	15	-	0.03631	0.05935	-	21.05	-	38.82262	-
20	19	13	-	0.03887	0.05638	-	31.58	-	31.05412	-
21	21	17	-	0.04141	0.05269	-	19.05	-	21.39617	-
22	21	16	-	0.04607	0.22712	-	23.81	-	79.71645	-
23	22	14	-	0.05219	0.05850	-	36.36	-	10.78706	-
24	23	14	-	0.05771	0.08480	-	39.13	-	31.95052	-
25	23	15	-	0.05784	0.06114	-	34.78	-	5.39581	-
Avg.							**18.43**	**-7.14**	**38.57864**	**99.97997**

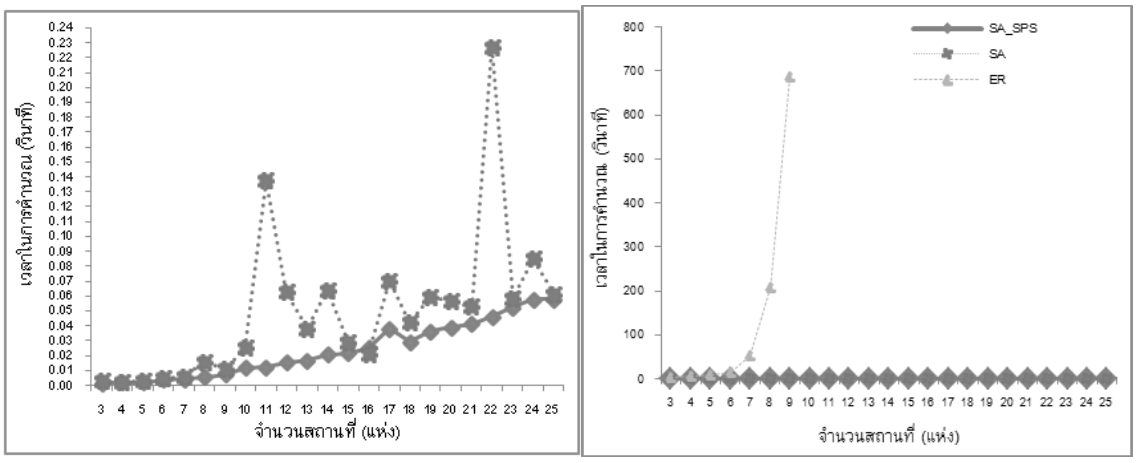

Figure 5 CPU time compare for the best case. of SA_SPS, SA and ER.

ด้านคุณภาพของคำตอบ

ขั้นตอนวิธี SA_SPS ให้คำตอบที่มีคุณภาพเท่ากับหรือดีกว่าขั้นตอนวิธี SA ในทุกสถานการณ์จำลองโดยมีค่าเฉลี่ยของร้อยละความแตกต่างของคุณภาพคำตอบ (*%gap S*) ที่ดีกว่าโดยเฉลี่ยร้อยละ 18.43 (Table 7) นอกจากนี้ ผลการทดลองยังชี้ให้เห็นว่าถึงแม้ขั้นตอนวิธี SA จะค้นเจอคำตอบที่ดีที่สุดใกล้เคียงกับขั้นตอนวิธีที่นำเสนอในหลายชุดข้อมูล แต่หากพิจารณาเวลาที่ใช้ในการประมวลผลร่วมด้วยจะพบว่า ขั้นตอนวิธีที่นำเสนอใช้เวลาในการประมวลผลเร็วกว่ามาก

สำหรับการเปรียบเทียบกับขั้นตอนวิธี ER ผลการทดลองเป็นไปในลักษณะเดิมคือ ขั้นตอนวิธี SA_SPS ให้คำตอบที่มีคุณภาพแย่กว่าโดยมีค่า *%gap S* ที่น้อยกว่าโดยเฉลี่ยร้อยละ 7.14 (Table 7)

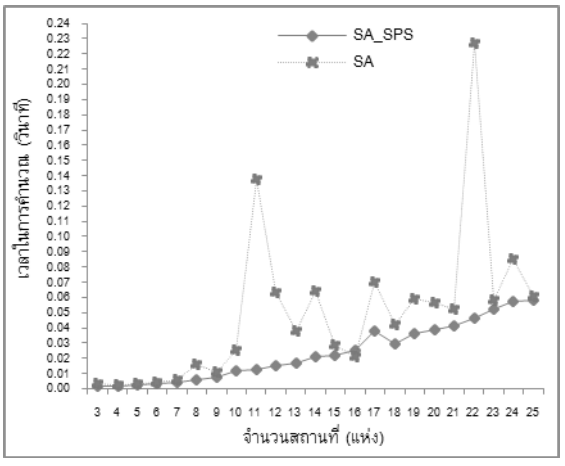

Figure 6 Extension fig.5for CPU time compare for the best case of SA_SPS and SA.

Table 8 Comparison results for the worse case.

	S			*Sec*$_{tot}$			*%gap S*		*%gap Sec*$_{tot}$	
place	**SA_SPS**	**SA**	**ER**	**SA_SPS**	**SA**	**ER**	**SA**	**ER**	**SA**	**ER**
3	3	3	3	0.00581	0.03895	5.34237	0	0	85.08601	99.89127
4	4	4	4	0.00943	0.02876	7.34236	0	0	67.19750	99.87151
5	4	2	5	0.01348	0.03436	12.34357	50.00	-25.00	60.75846	99.89077
6	6	4	5	0.01758	0.05030	18.34524	33.33	0	65.05437	99.90418
7	6	4	7	0.01893	0.02586	64.43253	33.33	-16.67	26.77701	99.97061
8	6	2	8	0.02024	0.02648	280.43523	66.67	-33.33	23.55363	99.99278
9	8	2	8	0.02990	0.01948	690.43523	75.00	-12.50	-53.49386	99.99567
10	9	5	-	0.02618	0.01003	-	44.44	-	-160.96710	-
11	11	7	-	0.03823	0.02040	-	36.36	-	-87.44423	-
12	10	2	-	0.03225	0.01689	-	80.00	-	-90.98662	-
13	12	5	-	0.06722	0.06212	-	58.33	-	-8.21475	-
14	14	8	-	0.09002	0.01980	-	42.86	-	-354.59274	-
15	13	4	-	0.08608	0.05980	-	69.23	-	-43.94629	-
16	13	4	-	0.08595	0.07798	-	69.23	-	-10.21235	-
17	15	5	-	0.09182	0.08465	-	66.67	-	-8.47037	-
18	15	5	-	0.07745	0.07671	-	66.67	-	-0.96467	-
19	19	8	-	0.19824	0.16750	-	57.89	-	-18.35577	-
20	19	5	-	0.11289	0.09447	-	73.68	-	-19.50437	-
21	21	7	-	0.20849	0.15286	-	66.67	-	-36.39807	-
22	21	6	-	0.19223	0.05221	-	71.43	-	-268.17912	-
23	21	5	-	0.19640	0.04095	-	76.19	-	-379.63465	-
24	22	6	-	0.19126	0.21158	-	72.73	-	9.60436	-
25	23	4	-	0.19852	0.03986	-	82.61	-	-398.04566	-
Avg.							**56.23**	**-12.50**	**-69.62519**	**99.93097**

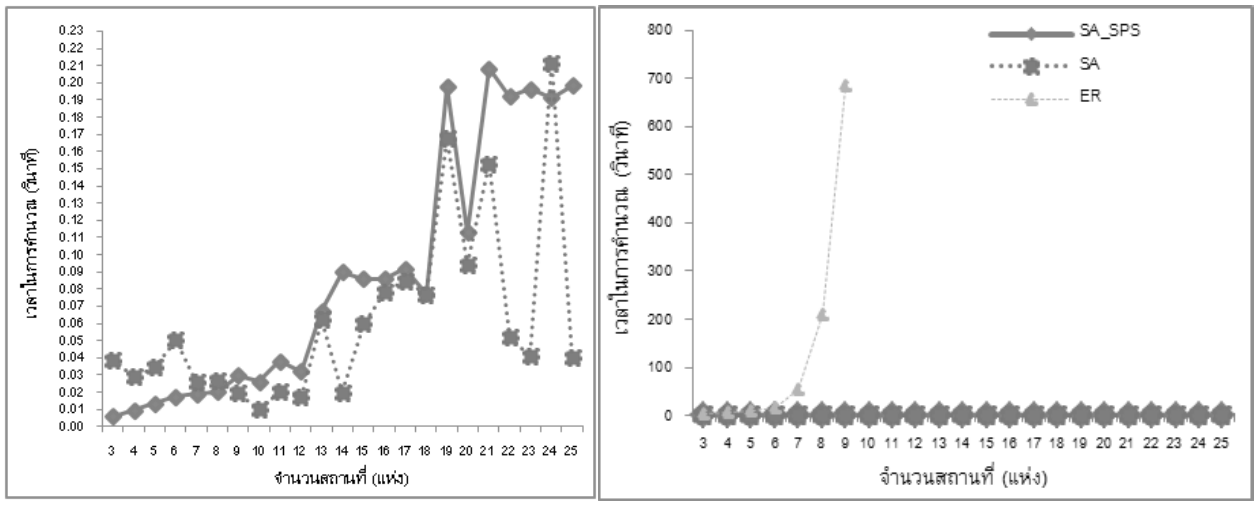

Figure 7 CPU time compare for the worse case of SA_SPS, SA and ER.

ผลการวิเคราะห์จากกรณีที่แย่ที่สุด

จากการทดลองโดยการประมวลผล 100 ครั้งของแต่ละขั้นตอนวิธี เพื่อที่จะวิเคราะห์ข้อมูลของแต่ละขั้นตอนวิธีเมื่อพบแผนการเดินทางที่ให้จำนวนสถานที่น้อยแห่งที่สุด ปรากฏผลการทดลองดัง Table 8 และ Figure 7-8 ซึ่งมีรายละเอียดดังต่อไปนี้

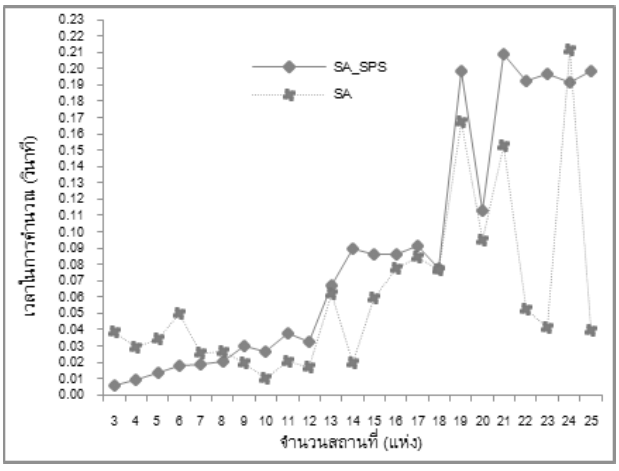

Figure 8 Extension fig.5for CPU time compare for the worse case of SA_SPS and SA.

ด้านเวลาในการประมวลผล

ขั้นตอนวิธี SA_SPS ใช้เวลาในการประมวลผลมากกว่าขั้นตอนวิธี SA โดยมีค่าเฉลี่ยของร้อยละความแตกต่างของเวลาที่ใช้ในการประมวลผล ($\%gap\ Sec_{tot}$) มากกว่าโดยเฉลี่ยร้อยละ 69.62 (Table 8) แต่ยังคงใช้เวลาในการประมวลผลน้อยกว่าขั้นตอนวิธี ER โดยมีค่าเฉลี่ยของร้อยละความแตกต่างของเวลาที่ใช้ในการประมวลผลน้อยกว่าโดยเฉลี่ยร้อยละ 99.93

ด้านคุณภาพของคำตอบ

ขั้นตอนวิธี SA_SPS ให้คำตอบที่มีคุณภาพเท่ากับหรือดีกว่าขั้นตอนวิธี SA ในทุกสถานการณ์จำลองโดยมีค่าเฉลี่ยของร้อยละความแตกต่างของคุณภาพคำตอบ (*%gap S*) ที่ดีกว่าโดยเฉลี่ยร้อยละ 56.23 (Table 8)

สำหรับการเปรียบเทียบกับขั้นตอนวิธี ER ให้ผลเหมือนเดิมคือขั้นตอนวิธีSA_SPS ให้คำตอบที่มีคุณภาพแย่กว่าโดยมีค่า *%gap S* ที่น้อยกว่าโดยเฉลี่ยร้อยละ 12.50 (Table 8)

จากผลการทดลองในกรณีนี้ที่เมื่อแต่ละขั้นตอนวิธีพบแผนการเดินทางที่ให้คำตอบที่แย่ที่สุดจะเห็นได้ว่า ขั้นตอนวิธีที่นำเสนอใช้เวลาในการประมวลผลมากกว่าขั้นตอนวิธี SA โดยเฉลี่ยร้อยละ 69.62 สาเหตุเนื่องจากถึงแม้จะเป็นกรณีคำตอบที่แย่ที่สุด แต่ขั้นตอนวิธี SA_SPS ยังสามารถให้แผนการเดินทางที่มีจำนวนสถานที่หลายแห่งกว่าขั้นตอนวิธี SA มาก โดยจะเห็นได้จากค่า *%gap S* ที่มีค่าสูงกว่าโดยเฉลี่ยถึงร้อยละ 56.23 นั่นหมายถึงจำนวนสถานที่ที่ขั้นตอนวิธีทั้งสองหาได้มีความแตกต่างกันมากนั่นเอง (Table 8) เมื่อมีจำนวนสถานที่ 24 แห่ง ขั้นตอนวิธีที่นำเสนอสามารถวางแผนการเดินทางให้ไปเยือนสถานที่ได้ถึง 22 แห่ง ในขณะที่ขั้นตอนวิธี SA สามารถวางแผนการเดินทางให้ไปเยือนได้เพียง 6 แห่ง ด้วยสาเหตุนี้ จึงส่งผลให้ขั้นตอนวิธีที่นำเสนอใช้เวลาในการประมวลผลมากกว่าขั้นตอนวิธี SA

การอภิปรายและสรุปผล

เพื่อทดสอบประสิทธิภาพของขั้นตอนวิธีที่นำเสนอในการทดลองจึงทำโดยเปรียบเทียบขั้นตอนวิธีที่นำเสนอกับขั้นตอนวิธีอื่น ได้แก่ ขั้นตอนวิธีการจำลองการอบเหนียวกับขั้นตอนวิธีค้นหาทุกเส้นทางที่เป็นไปได้ในการแก้ปัญหาการวางแผนการเดินทางท่องเที่ยวที่ต้องการเดินทางไปเยือนสถานที่หลายแห่งมากที่สุดภายในระยะเวลาจำกัดโดยออกแบบให้ทดลองกับสถานการณ์จำลองซึ่งสร้างจากจำนวนสถานที่ท่องเที่ยวที่แตกต่างกันตั้งแต่ 3 ถึง 25 แห่งทำให้ได้ทั้งสิ้น 23 สถานการณ์จำลอง จากนั้นทำการประมวลผล 100 ครั้งในแต่ละสถานการณ์จำลองในแต่ละขั้นตอนวิธี

สำหรับการวิเคราะห์ข้อมูลแบ่งออกเป็น 3 รูปแบบ ได้แก่ จากกรณีเฉลี่ยจากกรณีที่ดีที่สุด และจากกรณีที่แย่ที่สุด จากนั้นนำผลการทดลองจากทั้ง 3 รูปแบบไปวิเคราะห์ในด้านเวลาในการประมวลผลและด้านคุณภาพของคำตอบ

ในด้านเวลาในการประมวลผล ผลจากการวิเคราะห์ข้อมูลทั้ง 3 รูปแบบชี้ให้เห็นว่าขั้นตอนวิธีที่นำเสนอใช้เวลาในการประมวลผลน้อยกว่าทุกขั้นตอนวิธีที่นำมาเปรียบเทียบ โดยใช้เวลาในการประมวลผลน้อยกว่าขั้นตอนวิธีค้นหาทุกเส้นทางที่เป็นไปได้ค่อนข้างมาก เนื่องจากขั้นตอนวิธีที่นำเสนอได้ใช้เทคนิคเอสพีเอสแทนการสุ่มจุดในการหาลำดับการผ่านสถานที่ในกระบวนการค้นหาคำตอบเบื้องต้นและใช้สำหรับหลักการสับเปลี่ยนในกระบวนการปรับปรุงคำตอบ ทำให้สามารถจำกัดปริภูมิการค้นหาให้แคบลงกว่าเดิมได้จากการที่ได้เริ่มต้นค้นหาในจุดที่ถือว่าใกล้เคียงกับคำตอบที่ดีที่สุด ส่งผลให้ระยะเวลาในการคำนวณลดน้อยลง

ในด้านคุณภาพของคำตอบ ขั้นตอนวิธีที่นำเสนอให้คุณภาพแย่กว่าขั้นตอนวิธีค้นหาทุกเส้นทางที่เป็นไปได้เล็กน้อย เนื่องจากขั้นตอนวิธีค้นหาทุกเส้นทางที่เป็นไปได้เป็นการค้นหาทุกรูปแบบเส้นทางและประกันคำตอบที่ดีที่สุดทุกครั้งที่ประมวลผล ดังนั้น คำตอบที่ได้จากขั้นตอนวิธีนี้ย่อมดีกว่า แต่อย่างไรก็ตามขั้นตอนวิธีที่นำเสนอก็ยังให้คุณภาพคำตอบที่ดีกว่าขั้นตอนวิธีการจำลองการอบเหนียวแบบดั้งเดิม

นอกจากนี้ การเพิ่มส่วนพิเศษในการค้นหาเส้นทางที่อาจไปเยือนสถานที่ได้ขึ้น ที่ใช้หลักการสำรวจเพื่อหาโอกาสในการเพิ่มสถานที่ท่องเที่ยวที่ผู้ใช้ระบุเข้ามาแต่ยังไม่ถูกจัดลงในแผนการเดินทาง (เนื่องจากบางแผนการเดินทางใช้ระยะเวลาในการท่องเที่ยวเกินกรอบที่กำหนด) ที่ได้นำเสนอในบทความนี้นั้น ยังสามารถปรับปรุงคำตอบให้มีคุณภาพดีขึ้นกล่าวคือ ทำให้ได้แผนการเดินทางที่สามารถไปเยือนสถานที่ได้มากกว่าเดิม

ดังนั้นจึงสามารถสรุปได้ว่าขั้นตอนวิธีการจำลองการอบเหนียวที่มีพื้นฐานจากการค้นหาตามระยะเวลาการเดินทางที่น้อยที่สุดสำหรับการวางแผนการเดินทางท่องเที่ยวภายใต้เงื่อนไขบังคับด้านเวลาหรือเรียกชื่อย่อว่า เอสเอ_เอสพีเอส ที่ได้มีการดัดแปลงนำเทคนิคเอสพีเอสมาช่วยในการสร้างคำตอบเริ่มต้นและใช้ในหลักการสับเปลี่ยนเพื่อสร้างคำตอบใกล้เคียง โดยประยุกต์ใช้ร่วมกับขั้นตอนวิธีการจำลองการอบเหนียวที่นำเสนอในบทความนี้ เป็นวิธีที่พบว่ามีความเหมาะสมและใช้งานได้ดีสำหรับปัญหาวิจัยครั้งนี้ ดังผลการวิจัยที่รายงาน

เอกสารอ้างอิง

1. ศศิวิมล กอบัว. การพัฒนาระบบวางแผนแผนการเดินทางท่องเที่ยวภายใต้เงื่อนไขบังคับด้านเวลา [วิทยานิพนธ์ปริญญาวิทยาการสารสนเทศมหาบัณฑิต]. นครราชสีมา: มหาวิทยาลัยเทคโนโลยีสุรนารี; 2556.
2. จิติมนต์ อั่งสกุล และ ธรา อั่งสกุล. Online Trip Planners for the Tourism Industry.วารสารเทคโนโลยีสุรนารี 2551;2(1):33-45.
3. Blondel V, Tsitsiklis JN. NP-hardness of some linear control design problems. SIAM Journal on Control and Optimization 1997 Nov;35(6):2118-27.
4. Banos R, Ortega J, Gil C, Fernandez A, De Toro F. A simulated annealing-based parallel multi-objective approach to vehicle routing problems with time windows. Expert Systems with Applications 2013 Apr 30;40(5):1696-707.
5. Tavakkoli-Moghaddam R, Gazanfari M, Alinaghian M, Salamatbakhsh A, Norouzi N. A new mathematical model for a competitive vehicle routing problem with time windows solved by simulated annealing. Journal of manufacturing systems 2011 Apr 30;30(2):83-92.
6. สุพรรณ สุดสนธิ์, สรายุทธ กรวิรัตน์และนคร สุดสนธิ์. วิธีซิมมูเลเตดแอลเนลลิ่งสำหรับการจัดเส้นทางการขนส่งภายใต้กรอบเวลาที่ยืดหยุ่น. Engineering and Applied science Research 2557;41(4): 449-461.
7. Fang L, Chen P, Liu S. Particle swarm optimization with simulated annealing for TSP. In: Charles AL, Valeri MM, Zoran B, editors. AIKED 07.Proceedings of the 6th WSEAS International Conference on Artificial Intelligence, Knowledge Engineering and Data Bases; 2007 Feb 16-19; Corfu Island, Greece. Wisconsin USA: WSEAS; 2007. P.206-210.
8. Geng X, Chen Z, Yang W, Shi D, Zhao K. Solving the traveling salesman problem based on an adaptive simulated annealing algorithm with greedy search. Applied Soft Computing 2011 Jun 30;11(4):3680-9.
9. Leung SC, Zhang Z, Zhang D, Hua X, Lim MK. A meta-heuristic algorithm for heterogeneous fleet vehicle routing problems with two-dimensional loading constraints. European Journal of Operational Research 2013 Mar 1;225(2):199-210.
10. Cerny V. Thermodynamical approach to the traveling salesman problem: An efficient simulation algorithm. Journal of optimization theory and applications 1985 Jan 1;45(1):41-51
11. Kirkpatrick S, Gelatt CD, Vecchi MP. Optimization by simmulated annealing. Science 1983 May 13;220(4598):671-680.
12. Waters CD. A solution procedure for the vehicle-scheduling problem based on iterative route improvement. Journal of the Operational Research Society 1987 Sep 1;38(9):833-839.
13. กิตติภพ คนดี. การพัฒนาเทคนิคการกำหนดตำแหน่งติดตั้งโหนดอ้างอิงสำหรับระบบระบุตำแหน่งไร้สายภายในอาคารหลายชั้น [วิทยานิพนธ์ปริญญาวิศวกรรมศาสตรมหาบัณฑิต]. นครราชสีมา: มหาวิทยาลัยเทคโนโลยีสุรนารี; 2557.
14. Kirkpatrick S. Optimization by simulated annealing: Quantitative studies. Journal of statistical physics 1984 Mar 1;34(5-6):975-86.