

ซอฟต์แวร์แอนต์: ระบบมดในการแก้ปัญหาการเดินทางของพนักงานขาย

Soft-Ant: An Ant System for Traveling Saleman Problem

ธนาศศร์ สูดจิ่ง (Tanasat Sudjing)¹* ดร.วนิดา แก่นอากาศ (Dr.Wanida Kanarkard)**

บทคัดย่อ

งานวิจัยฉบับนี้นำเสนอการประยุกต์เทคนิคการคำนวณแบบซอฟต์แวร์ (Soft Computing) เพื่อพัฒนาขั้นตอนวิธีซอฟต์แวร์แอนต์ (Soft-Ant) ระบบมดในการแก้ปัญหาการเดินทางของพนักงานขาย ผู้วิจัยได้ทำการปรับปรุงขั้นตอนวิธีระบบอาณานิคมมด (Ant Colony System : ACS) ในส่วนของกฎการปรับปรุงฟีโรโมนแบบเฉพาะที่ (Local pheromone update rule) โดยการปรับปรุงค่าฟีโรโมนเริ่มต้น (τ_0) ด้วยการประยุกต์เทคนิคการคำนวณแบบซอฟต์แวร์ ผลการวิจัยสามารถประยุกต์เทคนิคการคำนวณแบบซอฟต์แวร์ เพื่อพัฒนาอัลกอริทึมซอฟต์แวร์แอนต์ ได้เป็นผลสำเร็จ อัลกอริทึมซอฟต์แวร์แอนต์ ที่ผู้วิจัยพัฒนาขึ้นสามารถแก้ปัญหาการเดินทางของพนักงานขายแบบสมมาตร (Symmetric) และอสมมาตร (Asymmetric) ได้ ผลการทดลองเปรียบเทียบอัลกอริทึมระบบอาณานิคมมด (ACS) ดั้งเดิม กับอัลกอริทึมซอฟต์แวร์แอนต์ (Soft-Ant) ในการแก้ปัญหาการเดินทางของพนักงานขาย โดยใช้ชุดข้อมูลจาก TSPLIB ทั้งหมด 6 ชุด พบว่าโดยเฉลี่ยอัลกอริทึมซอฟต์แวร์แอนต์ ให้ผลลัพธ์การทำงานดีกว่าอัลกอริทึมระบบอาณานิคมมด สามารถหาผลเฉลยของปัญหาที่ดีกว่า มีค่าเปอร์เซ็นต์ความผิดพลาดน้อยกว่า และใช้เวลาน้อยกว่า ดังนั้นการประยุกต์เทคนิคการคำนวณแบบซอฟต์แวร์สามารถที่จะเป็นทางเลือกใหม่ในการปรับปรุงขั้นตอนวิธีระบบอาณานิคมมดในการแก้ปัญหาการเดินทางของพนักงานขายได้

ABSTRACT

This research proposes a new approach which derived from soft computing, for the construction Soft-Ant : An ant system for traveling salesman problem. We are approved Ant Colony System (ACS) algorithm. In local pheromone update rule, the initial value of pheromone (τ_0) are determined by soft computing. This research can use soft computing technical for construction Soft-Ant : An ant system for traveling salesman problem. The proposes can solve symmetric traveling salesman problem (TSP) and asymmetric traveling salesman problem (ATSP) . Experimental result show that the proposed Soft-Ant: An ant system for traveling salesman problem outperform the conventional Ant Colony System (ACS). The Soft-ant can find the better solution of the problem, percentage of error less and take less time. Hence, soft computing is expected to be a new alternative way to the construction of ACS model for traveling salesman problem.

คำสำคัญ: ระบบอาณานิคมมด การคำนวณแบบซอฟต์แวร์ ปัญญาเชิงคำนวณ

Keywords: Ant colony system, Soft computing, Computational intelligence

¹Correspondent author: tanasat.s@kkumail.com

* นักศึกษา หลักสูตรวิศวกรรมศาสตรมหาบัณฑิต สาขาวิชาวิศวกรรมคอมพิวเตอร์ คณะวิศวกรรมศาสตร์ มหาวิทยาลัยขอนแก่น

** รองศาสตราจารย์ ภาควิชาวิศวกรรมคอมพิวเตอร์ คณะวิศวกรรมศาสตร์ มหาวิทยาลัยขอนแก่น



บทนำ

ปัญหาการเดินทางของพนักงานขาย(traveling salesman problem : TSP) เป็นปัญหาที่ยากและยังไม่มีวิธีการที่ใช้เวลาแบบโพลิโนเมียลในการแก้ปัญหาได้ ปัญหาการเดินทางของพนักงานขายได้รับความสนใจจากนักวิจัย เพื่อพัฒนาวิธีการหาคำตอบให้ดีและเร็ว วิธีการหาค่าที่เหมาะสมที่สุดด้วยอาณานิคมมด (Ant Colony Optimization : ACO) เป็นขั้นตอนวิธีทางคอมพิวเตอร์ที่ประสบความสำเร็จและสามารถนำไปประยุกต์ใช้เพื่อแก้ปัญหาการเดินทางของพนักงานขายได้อย่างมีประสิทธิภาพ Bonavear และ Dorigo [1] ได้ทำการเปรียบเทียบการหาคำตอบ แสดงให้เห็นว่าระบบอาณานิคมมด (Ant Colony System : ACS) ให้คำคำตอบที่ดีที่สุดเมื่อเปรียบเทียบกับวิธีการหาคำตอบด้วยขั้นตอนวิธีเชิงพันธุกรรม (Genetic Algorithm) การจำลองการอบเหนียว (Simulated Annealing) การค้นหาแบบตาบ (Tabu Search) และวิธีการโปรแกรมวิวัฒนาการ (Evolutionary programming)

วิธีการหาค่าที่เหมาะสมที่สุดด้วยวิธีอาณานิคมมดเป็นปัญญาประดิษฐ์ (Artificial Intelligence) ที่จัดอยู่ในกลุ่มของขั้นตอนวิธีเชิงวิวัฒนาการ (Evolutionary algorithm) ที่อาศัยฐานประชากร (Population based) ช่วยในการหาคำตอบที่เหมาะสมที่สุดในเวลาที่เหมาะสม เป็นวิธีการที่ได้รับแรงบันดาลใจจากธรรมชาติ โดยการเลียนแบบพฤติกรรมทางสังคมในการหาอาหารของมด ซึ่งมดสามารถเดินทางค้นหาแหล่งอาหาร และสร้างเส้นทางที่สั้นที่สุดระหว่างรังกับแหล่งอาหารได้โดยการปล่อยฟีโรโมนระหว่างเส้นทางเพื่อใช้ในการสื่อสารทางอ้อมกับมดตัวอื่นในฝูง แต่อย่างไรก็ตามวิธีการหาค่าที่เหมาะสมที่สุดด้วยอาณานิคมมด ยังมีข้อด้อยบางประการ เช่น พฤติกรรมความเมื่อยล้า (Stagnation behavior), การคำนวณที่ยาวนาน (Long computational time) และการลู่เข้าก่อนกำหนด (Premature convergence problem) เป็นต้น ดังนั้น จึงได้มีนักวิจัยให้ความสนใจปรับปรุงพัฒนาวิธีการหาค่าที่เหมาะสมที่สุดด้วยอาณานิคมมดให้มีประสิทธิภาพสูงขึ้น

การประมวลผลด้วยคอมพิวเตอร์ในปัจจุบัน หรือการคำนวณแบบดั้งเดิม เรียกว่า Hard Computing เป็นการประมวลผลที่มีขั้นตอนการแก้ปัญหาที่ชัดเจน (Deterministic) ได้ผลลัพธ์ที่มีความถูกต้อง (Accuracy) มีความแน่นอน (Certainty) มีความชัดเจน (Precise) และไม่ขึ้นอยู่กับเวลา (Timely Independent) ในทางตรงกันข้าม รูปแบบการคำนวณที่อยู่ภายใต้สภาวะที่มีความไม่แน่นอน (Uncertainty) ไม่มีความชัดเจน (Imprecise) คลุมเครือ (Fuzzy) หรือมีขอบเขตไม่แน่นอน (Vague) ซึ่งผลลัพธ์หรือคำตอบที่ได้จะเป็นค่าโดยประมาณ (Estimates) การคำนวณหรือการประมวลผลในลักษณะดังกล่าวเรียกว่า การคำนวณแบบซอฟต์ (Soft Computing) [2]

งานวิจัย [3] ได้นำเสนอวิธีการใหม่โดยการประยุกต์การคำนวณแบบซอฟต์(Soft Computing) เพื่อปรับปรุงโครงข่ายประสาทเทียมแบบ RBF (Radial Basis Function Neural Network : RBFN) ในการจดจำใบหน้าของมนุษย์ได้เป็นผลสำเร็จและให้ผลลัพธ์การทำงานที่ดีกว่า RBFN แบบดั้งเดิม เพราะการประยุกต์การคำนวณแบบซอฟต์ จะทำให้เกิดความไม่แน่นอนในการสร้างผลลัพธ์ที่ดีกว่าผลลัพธ์จากการจำแนก ทำให้ผลลัพธ์มีความยืดหยุ่น (Soften) มากกว่าผลลัพธ์ที่ตรงที่ได้มาจาก RBF และจะทำให้เกิดความเป็นไปได้ในการจำแนกที่ถูกต้องมากกว่า

งานวิจัยนี้จึงทำการศึกษาค้นคว้าเพื่อหาวิธีปรับปรุงขั้นตอนวิธีระบบอาณานิคมมด โดยเสนอปรับปรุงบางส่วนของแบบจำลองที่มีการคำนวณแบบเที่ยงตรง (Harden) ด้วยการประยุกต์เทคนิคการคำนวณแบบซอฟต์ (Soft Computing) เพื่อทำให้ผลลัพธ์มีความยืดหยุ่น (Soften) มากกว่าแบบเดิม จะทำให้มีความเป็นไปได้ที่จะได้ผลลัพธ์ในการค้นหาคำตอบที่ดีขึ้นเรียกวิธีการใหม่นี้ว่า ซอฟต์แอนท์ (Soft-Ant) ผลการทดลองอัลกอริทึมซอฟต์แอนท์เปรียบเทียบกับอัลกอริทึมระบบอาณานิคมมด (ACS)ดั้งเดิม ในการแก้ปัญหาการเดินทางของพนักงานขาย โดยใช้ชุดข้อมูลจาก TSPLIB ทั้งหมด 6 ชุด พบว่าโดยเฉลี่ยอัลกอริทึมซอฟต์แอนท์ ให้ผลลัพธ์การทำงานดีกว่าอัลกอริทึมระบบอาณานิคมมด สามารถหาผลเฉลยของปัญหาที่ดีกว่า มีค่าเปอร์เซ็นต์ความผิดพลาดน้อยกว่า และใช้เวลาน้อยกว่า

สำหรับปัญหาที่มีขนาดเล็ก และมีลักษณะการกระจายสม่ำเสมอเต็มพื้นที่ ดังนั้นแนวคิดในการประยุกต์เทคนิคการคำนวณแบบซอฟต์แวร์จึงสามารถนำมาใช้ในการปรับปรุงขั้นตอนวิธีระบบบอานานิคมมดเพื่อใช้ในการแก้ปัญหาการเดินทางของพนักงานขายให้ดีขึ้นได้

ระบบบอานานิคมมด (Ant Colony System)

ในปี 1991 Marco Dorigo และคณะ ได้ค้นพบวิธีการแก้ปัญหาโดยได้เลียนแบบพฤติกรรมทางธรรมชาติของมดในการหาเส้นทางที่สั้นที่สุดจากแหล่งอาหารสู่รัง[4] ซึ่งวิธีการในกลุ่มของความฉลาดแบบกลุ่ม (Swarm intelligence: SI) จะใช้หลักการจัดระบบด้วยตนเอง (Self-Organising Principles) ซึ่งเป็นพฤติกรรมการทำงานร่วมกันอย่างมีประสิทธิภาพของมด เช่น พฤติกรรมหาอาหาร (Foraging), การแบ่งประเภทงาน (Division of Labor), การช่วยกันขนย้าย (Cooperative Transport) เป็นต้น โดยมดจะสื่อสารระหว่างกัน โดยใช้สารเคมีชนิดหนึ่งที่เรียกกันว่า "สารฟีโรโมน" (Pheromone) [5-6] กล่าวคือ ขณะที่มดเดินทางหาอาหารอยู่นั้นมดจะทิ้งสารฟีโรโมนเอาไว้บนเส้นทางที่มดตัวนั้นเดิน เมื่อมดตัวอื่นที่อยู่บริเวณใกล้เคียงได้กลิ่นสารฟีโรโมนที่ทิ้งไว้แล้วก็จะมีความเป็นไปได้ (Probability) ที่จะเลือกเส้นทางดังกล่าวตามเส้นทางของมดตัวแรก มากกว่าที่จะเลือกเส้นทางอื่น หลังจากที่มีมดเลือกเส้นทางเดินได้แล้วก็จะทำการทิ้งสารฟีโรโมนทับลงบนเส้นทางที่มดตัวที่สองเลือกเดิน ทำให้ปริมาณของสารฟีโรโมนบนเส้นทางนั้นมีปริมาณเพิ่มสูงขึ้นซึ่งกลไกนี้เรียกว่า กระบวนการป้อนกลับแบบบวก (Positive Feedback or Autocatalytic Mechanism) ในขณะเดียวกันธรรมชาติของ สารฟีโรโมนก็ มีการระเหยอยู่ตลอดเวลา (Pheromone Decay) กล่าวคือถ้าเส้นทางใดที่มดไม่ มีการเลือกเดินบนเส้นทางนั้นก็จะไม่มีปริมาณของสารฟีโรโมนน้อยลง ความเป็นไปได้ที่มดจะเลือกเส้นทางดังกล่าว จึงมีน้อยลง จนในที่สุดปริมาณสารฟีโรโมนบนเส้นทางนั้นก็จะถูกระเหยไปจนหมด ซึ่งเรียกกลไกนี้ว่า กระบวนการป้อนกลับแบบลบ (Negative Feedback) [1, 4, 7] ซึ่งโดยทั่วไปแล้วจะสังเกตเห็นได้ว่าเมื่อกลุ่มมดเริ่มเดินทางหาอาหารในตอนแรก มดจะมีการเดินแบบสุ่ม และมีการกระจายของเส้นทางเดินของมด แต่เมื่อเวลาผ่านไปเส้นทางของมดที่เดินทางหาอาหารสู่รังจะเหลืออยู่เพียงเส้นทางเดียว และเส้นทางนั้นจะมีมดเดินเป็นจำนวนมากเมื่อเทียบกับตอนเริ่มต้น

การจำลองพฤติกรรมของมดในธรรมชาติ[8] ภาพที่ 1 อธิบายพฤติกรรมและความสำเร็จของมดในการเลือกเส้นทางที่สั้นที่สุดของมดในการเดินทางหาอาหาร การเดินปกติของมดจากจุด A ไปจุด E จนกระทั่งเมื่อมีสิ่งกีดขวางอยู่บนเส้นทางดังภาพที่ 1 (ข) ที่จุด B และจุด D มดพยายามหลบหลีกสิ่งกีดขวาง ทำให้เกิดเส้นทางใหม่สองเส้นทาง คือเส้นทาง BCD หรือ DCB และเส้นทาง BHD หรือ DHB ซึ่งทั้งสองเส้นทางมีระยะทางไม่เท่ากันและในขณะนั้นมดจะรู้ว่าเส้นทางใดคือเส้นทางที่สั้นกว่า ในช่วงแรกเนื่องจากยังไม่มดตัวใดเดินบนเส้นทางเหล่านั้นจึงไม่มีสารฟีโรโมนบนเส้นทางใหม่ มดจะตัดสินใจเลือกทางเดินแบบสุ่มเลือกไปในหนึ่งเส้นทางจากสองทางเลือก เมื่อมดจำนวนหนึ่งสุ่มเลือกเดินไปในทั้งสองเส้นทาง เป็นระยะเวลาหนึ่ง การเดินทางของฝูงมดจะเป็น ดังภาพที่ 1 (ค) ซึ่งจะพบว่าบนเส้นทางที่สั้นกว่าจะมีมดเดินบนเส้นทางนั้นมากกว่า และมดทุกตัวจะปล่อยสารฟีโรโมน ทำให้บนเส้นทางที่สั้นกว่านั้นมีปริมาณสารฟีโรโมนที่มากกว่านั่นเอง และเป็นผลทำให้การเลือกเส้นทางของมดในช่วงหลัง ๆ มดจะมีโอกาสมากกว่าที่จะเลือกเดินไปตามเส้นทางที่มีปริมาณสารฟีโรโมนที่มากกว่า ทำให้ฝูงมดสามารถสร้างเส้นทางที่สั้นที่สุดได้

วิธีระบบบอานานิคมมด (ACS) ถูกนำเสนอครั้งแรกในปี 1997 โดย Dorigo และ Gambardella [7] โดยการพัฒนาเพิ่มขึ้นเพื่อเพิ่มประสิทธิภาพจากระบบมด (Ant System :AS) ที่ไม่สามารถหาผลลัพธ์ที่ดีสำหรับปัญหามดเล็กได้ภายในเวลาที่เหมาะสมได้ ระบบบอานานิคมมดเป็นการปรับปรุงจากวิธีระบบมด ใน 4 ส่วน [1] ได้แก่ ส่วนแรกกฎ



การส่งผ่าน (Transition Rule) ส่วนที่สองการปรับปรุงฟีโรโมนวงกว้าง (Global pheromone update rule) ส่วนที่สามการปรับเส้นทางฟีโรโมนเฉพาะที่ (Local Updates of Pheromone Trail) และส่วนสุดท้ายการใช้รายการตำแหน่งที่ถูกเลือก (Candidate List) จากการปรับปรุงนี้ทำให้วิธีระบบอาณานิคมมดสามารถแก้ปัญหาการเดินทางของพนักงานขายได้ดีกว่าวิธีระบบมด

การพัฒนาซอฟต์แวร์แอนท์

งานวิจัยนี้ได้ทำการศึกษาอัลกอริทึมระบบอาณานิคมมด (Ant Colony System) ซึ่งส่วนสำคัญส่วนหนึ่งที่จะเพิ่มเข้ามาในอัลกอริทึมระบบอาณานิคมมด คือ กฎการปรับปรุงฟีโรโมนเฉพาะที่ (Local pheromone update rule) ดังสมการ

$$\tau_{ij}(t) \leftarrow (1 - \rho) \cdot \tau_{ij}(t) + \rho \cdot \tau_0 \quad (1)$$

สมการ (1) มีค่าอัตราการระเหย (ρ) หรือค่าควบคุมการสลายตัวของฟีโรโมน เป็นค่าคงที่มีค่าอยู่ในช่วง $0 < \rho < 1$ และค่าฟีโรโมนเริ่มต้น (τ_0) เป็นค่าคงที่มีค่าตามที่คำนวณในตอนกำหนดค่าเริ่มต้นครั้งเดียว ทำให้ส่วนของสมการกฎการปรับปรุงฟีโรโมนเฉพาะที่มีลักษณะเป็นแบบฮาร์ด (Hard Computing) ซึ่งความจริงแล้วในธรรมชาติยังมีปัจจัยอื่น ๆ ที่มีผลต่อการสลายตัวของฟีโรโมนเช่น มดที่มีขนาดไม่เท่ากัน, ความแรงของสายลม, ปริมาณแสงแดด เป็นต้น ดังนั้นผู้วิจัยจึงทำการปรับปรุงกฎการปรับปรุงฟีโรโมนเฉพาะที่ให้มีลักษณะเป็นการคำนวณแบบซอฟต์ (Soft Computing) ในสมการส่วนของการจำลองการปล่อย Pheromone ซึ่งเป็นการคูณของค่าคงที่ คือ ρ และ τ_0 ผู้วิจัยได้ทำการเปลี่ยนค่าคงที่ในสมการนี้ให้มีความยืดหยุ่น ไม่เป็นค่าคงที่เหมือนสมการเดิม ซึ่งสอดคล้องกับแนวคิดของการคำนวณแบบซอฟต์โดยเพิ่ม Sigmoid function หารเข้าไปในสมการส่วนของการจำลองปล่อย pheromone เพื่อขยายขอบเขตในการสร้างผลลัพธ์ ทำให้ผลลัพธ์มีความยืดหยุ่น (Soften) มากกว่าผลลัพธ์แบบที่แข็งตรง (Harden) จะให้มีความเป็นไปได้ที่จะได้ผลลัพธ์ในการค้นหาคำตอบที่ดีขึ้น สมการใหม่ในกฎการปรับปรุงฟีโรโมนเฉพาะที่แสดงดังสมการ (2)

ผู้วิจัยจึงทำการปรับปรุงกฎการปรับปรุงฟีโรโมนเฉพาะที่ให้มีลักษณะเป็นการคำนวณแบบซอฟต์ (Soft Computing) ในสมการส่วนของการจำลองการปล่อย Pheromone ซึ่งเป็นการคูณของค่าคงที่ คือ ρ และ τ_0 ผู้วิจัยได้เพิ่ม Sigmoid function หารเข้าไปในสมการส่วนของการจำลองปล่อย pheromone เพื่อขยายขอบเขตในการสร้างผลลัพธ์ ทำให้ผลลัพธ์มีความยืดหยุ่น (Soften) มากกว่าผลลัพธ์แบบที่แข็งตรง (Harden) จะให้มีความเป็นไปได้ที่จะได้ผลลัพธ์ในการค้นหาคำตอบที่ดีขึ้น

$$\tau_{ij}(t) \leftarrow (1 - \rho) \cdot \tau_{ij}(t) + \frac{\rho \cdot \tau_0}{\left(\frac{1}{1 + e^{-x}}\right)} \quad (2)$$

โดย x เป็นตัวเลขสุ่มที่มีค่าระหว่าง 0 ถึง 1

วิธีการวิจัย

ในการทดลองเปรียบเทียบการหาผลเฉลยของปัญหาการเดินทางของพนักงานขาย ของอัลกอริทึมซอฟต์แอนท์ เปรียบเทียบกับ อัลกอริทึมระบบอาณานิคมมด โดยเลือกข้อมูลปัญหาการเดินทางของพนักงานขายมาจาก

TSPLIB[9] เพื่อใช้ในการทดลอง ชุดข้อมูลที่เลือกมาเป็นชุดข้อมูลแบบสมมาตร และอสมมาตร ที่มีขนาดของปัญหาต่างกัน และมีลักษณะของการกระจายตัวของข้อมูลแตกต่างกัน (ตารางที่ 1)

การทดลองได้มีกำหนดพารามิเตอร์สำหรับอัลกอริทึมระบบอานานิคมมด และอัลกอริทึมซอฟต์แวร์แอนโทให้มีความเท่ากัน (ตารางที่ 2) และทำการทดลองกับข้อมูลที่ละชุดปัญหา ทำซ้ำจำนวน 3 ครั้ง เพื่อหาค่าเฉลี่ย และนำมาเปรียบเทียบกัน กำหนดเงื่อนไขการหยุดด้วยจำนวนรอบการสูงสุดไว้ดังนี้ 100, 300, 500, 700 และ 1000 รอบ

ผลการวิจัย

จากการทดลองกับชุดข้อมูลปัญหา *eil51* เปรียบเทียบอัลกอริทึมระบบอานานิคมมดกับ Soft-Ant ในการแก้ปัญหา พบว่าที่การกำหนดจำนวนรอบทำงาน 100 รอบ Soft-Ant มีผลการทำงานที่ดีกว่า ACS คือมีค่าเฉลี่ยของผลลัพธ์เท่ากับ 456.67 มีเปอร์เซ็นต์ความผิดพลาดเท่ากับ 7.20% เวลาที่ใช้ไปเท่ากับ 2.82 วินาที และที่การกำหนดจำนวนรอบทำงาน 300, 500, 700 และ 1000 รอบ ACS มีผลการทำงานที่ดีกว่า Soft-Ant คือมีค่าเฉลี่ยของผลลัพธ์เท่ากับ 446.33, 434.33, 439 และ 431 มีเปอร์เซ็นต์ความผิดพลาดเท่ากับ 4.77%, 2.66%, 3.05% และ 2.00% เวลาที่ใช้ไปเท่ากับ 8.69, 14.10, 18.47 และ 25.71 วินาที ตามลำดับ (ตารางที่ 3)

เมื่อทำการเปรียบเทียบเปอร์เซ็นต์ความผิดพลาด (%Err) ของอัลกอริทึมระบบอานานิคมมดกับ Soft-Ant ในการแก้ปัญหา *eil51* พบว่า ที่การกำหนดรอบการทำงาน 100 รอบ Soft-Ant จะมีเปอร์เซ็นต์ความผิดพลาดน้อยกว่า ACS แต่ที่การกำหนดรอบการทำงาน 300, 500, 700 และ 1000 รอบ อัลกอริทึม Soft-Ant จะมีเปอร์เซ็นต์ความผิดพลาดมากกว่าอัลกอริทึม ACS (ภาพที่ 2 ก)

จากการทดลองกับชุดข้อมูลปัญหา *berlin52* เปรียบเทียบอัลกอริทึมระบบอานานิคมมดกับ Soft-Ant ในการแก้ปัญหา พบว่าที่การกำหนดจำนวนรอบทำงาน 100, 500 และ 700 รอบ Soft-Ant มีผลการทำงานที่ดีกว่า ACS คือมีค่าเฉลี่ยของผลลัพธ์เท่ากับ 8090.67, 7776.33 และ 7678.00 มีเปอร์เซ็นต์ความผิดพลาดเท่ากับ 7.27%, 3.11% และ 1.80% เวลาที่ใช้ไปเท่ากับ 3.09, 14.37 และ 18.65 วินาที ตามลำดับในการกำหนดจำนวนรอบทำงาน 300 และ 1000 รอบ ACS มีผลการทำงานที่ดีกว่า Soft-Ant คือ มีค่าเฉลี่ยของผลลัพธ์เท่ากับ 7717 และ 7587.67 มีเปอร์เซ็นต์ความผิดพลาดเท่ากับ 2.32% และ 0.61% เวลาที่ใช้ไปเท่ากับ 8.69 และ 8.46 วินาที ตามลำดับ (ตารางที่ 4)

ผลการทดลองเปรียบเทียบเปอร์เซ็นต์ความผิดพลาด (%Err) ของอัลกอริทึมระบบอานานิคมมดกับ อัลกอริทึม Soft-Ant ในการแก้ปัญหา *berlin52* มาเปรียบเทียบกัน พบว่า ที่การกำหนดรอบการทำงาน 100, 500 และ 700 รอบ Soft-Ant จะมีเปอร์เซ็นต์ความผิดพลาดน้อยกว่าอัลกอริทึม ACS แต่ที่การกำหนดรอบการทำงาน 300 และ 1000 รอบ อัลกอริทึม Soft-Ant จะมีเปอร์เซ็นต์ความผิดพลาดมากกว่าอัลกอริทึม ACS (ภาพที่ 2 ข)

จากการทดลองกับชุดข้อมูลปัญหา *lin105* เปรียบเทียบอัลกอริทึมระบบอานานิคมมดกับอัลกอริทึม Soft-Ant ในการแก้ปัญหา พบว่า ที่การกำหนดจำนวนรอบทำงาน 100 และ 500 รอบ Soft-Ant มีผลการทำงานที่ดีกว่า อัลกอริทึม ACS คือมีค่าเฉลี่ยของผลลัพธ์เท่ากับ 16040.67 และ 15007.67 มีเปอร์เซ็นต์ความผิดพลาดเท่ากับ 11.56% และ 4.37% เวลาที่ใช้ไปเท่ากับ 11.73 และ 48.64 วินาทีตามลำดับในการกำหนดจำนวนรอบทำงาน 300, 700 และ 1000 รอบ ACS มีผลการทำงานที่ดีกว่า Soft-Ant คือ มีค่าเฉลี่ยของผลลัพธ์เท่ากับ 15481, 15202 และ 14868 มีเปอร์เซ็นต์ความผิดพลาดเท่ากับ 7.66%, 5.72% และ 3.40% เวลาที่ใช้ไปเท่ากับ 28.1, 62.99 และ 90.27 วินาที ตามลำดับ (ตารางที่ 5)

เมื่อนำเปอร์เซ็นต์ความผิดพลาด (%Err) ของอัลกอริทึมระบบอานานิคมมดกับอัลกอริทึม Soft-Ant ในการแก้ปัญหา *lin105* มาเปรียบเทียบกัน พบว่า ที่การกำหนดรอบการทำงาน 100 และ 500 รอบ Soft-Ant จะมีเปอร์เซ็นต์



ความผิดพลาดน้อยกว่าอัลกอริทึม ACS แต่ที่การกำหนดรอบการทำงาน 300, 700 และ 1000 รอบ อัลกอริทึม Soft-Ant จะมีเปอร์เซ็นต์ความผิดพลาดมากกว่าอัลกอริทึม ACS (ภาพที่ 2 ค)

จากการทดลองกับชุดข้อมูลปัญหา pcb442 เปรียบเทียบอัลกอริทึมระบบอานานิคมมดกับอัลกอริทึม Soft-Ant ในการแก้ปัญหา พบว่า ที่การกำหนดจำนวนรอบการทำงาน 100, 500, 700 และ 1000 รอบ Soft-Ant มีผลการทำงานที่ดีกว่า ACS คือมีค่าเฉลี่ยของผลลัพธ์เท่ากับ 59593.33, 59489.00, 59421.33 และ 58805.67 มีเปอร์เซ็นต์ความผิดพลาดเท่ากับ 17.36%, 17.16%, 17.02% และ 15.81% เวลาที่ใช้ไปเท่ากับ 145.97, 717.53, 967.81 และ 1445.95 วินาทีตามลำดับ ในการกำหนดจำนวนรอบการทำงาน 300 รอบ อัลกอริทึม ACS มีผลการทำงานที่ดีกว่าอัลกอริทึม Soft-Ant คือมีค่าเฉลี่ยของผลลัพธ์เท่ากับ 58678.00 มีเปอร์เซ็นต์ความผิดพลาดเท่ากับ 15.56% เวลาที่ใช้ไปเท่ากับ 427.08 วินาที (ตารางที่ 6)

เมื่อนำเปอร์เซ็นต์ความผิดพลาด (%Err) ของอัลกอริทึมระบบอานานิคมมดกับ อัลกอริทึม Soft-Ant ในการแก้ปัญหา pcb442 มาเปรียบเทียบ พบว่า ที่การกำหนดรอบการทำงาน ทำงาน 100, 500, 700 และ 1000 รอบ Soft-Ant จะมีเปอร์เซ็นต์ความผิดพลาดน้อยกว่า ACS แต่ที่การกำหนดรอบการทำงาน 300 รอบ อัลกอริทึม Soft-Ant จะมีเปอร์เซ็นต์ความผิดพลาดมากกว่า อัลกอริทึม ACS (ภาพที่ 2 ง)

จากการทดลองกับชุดข้อมูลปัญหา br17 เปรียบเทียบอัลกอริทึมระบบอานานิคมมดกับ Soft-Ant ในการแก้ปัญหา พบว่า ที่การกำหนดจำนวนรอบการทำงาน 100, 700 และ 1000 รอบ Soft-Ant มีผลการทำงานที่ดีกว่า ACS คือมีค่าเฉลี่ยของผลลัพธ์เท่ากับ 68.67, 74 และ 68 มีเปอร์เซ็นต์ความผิดพลาดเท่ากับ 76.07%, 79.84% และ 74.36% เวลาที่ใช้ไปเท่ากับ 0.53, 3.77 และ 5.21 วินาที ตามลำดับ ในการกำหนดจำนวนรอบการทำงาน 300 รอบ ACS มีผลการทำงานที่ดีกว่า Soft-Ant คือมีค่าเฉลี่ยของผลลัพธ์เท่ากับ 68.67 มีเปอร์เซ็นต์ความผิดพลาดเท่ากับ 76.07% เวลาที่ใช้ไปเท่ากับ 1.63 วินาที และในการกำหนดจำนวนรอบการทำงาน 500 รอบพบว่าอัลกอริทึมทั้งสองมีค่าเฉลี่ยของผลลัพธ์ และเปอร์เซ็นต์ความผิดพลาดเท่ากัน แต่เวลาที่ใช้ไป Soft-Ant ใช้เวลาน้อยกว่าในการทำงาน (ตารางที่ 7)

เมื่อนำเปอร์เซ็นต์ความผิดพลาด (%Err) ของอัลกอริทึมระบบอานานิคมมดกับ Soft-Ant ในการแก้ปัญหา br17 มาเปรียบเทียบกัน พบว่า ที่การกำหนดรอบการทำงาน ทำงาน 100, 700 และ 1000 รอบ อัลกอริทึม Soft-Ant จะมีเปอร์เซ็นต์ความผิดพลาดน้อยกว่า ACS แต่ที่การกำหนดรอบการทำงาน 300 รอบ อัลกอริทึม Soft-Ant จะมีเปอร์เซ็นต์ความผิดพลาดมากกว่าอัลกอริทึม ACS ส่วนที่การกำหนดค่า 500 รอบ มีเปอร์เซ็นต์ความผิดพลาดเท่ากัน (ภาพที่ 2 จ)

จากการทดลองกับชุดข้อมูลปัญหา ry48p เปรียบเทียบอัลกอริทึมระบบอานานิคมมดกับ Soft-Ant ในการแก้ปัญหา ry48p พบว่า ที่การกำหนดจำนวนรอบการทำงาน 100, 500 และ 700 รอบ Soft-Ant มีผลการทำงานที่ดีกว่า ACS คือมีค่าเฉลี่ยของผลลัพธ์เท่ากับ 15528.67, 15331 และ 14847 มีเปอร์เซ็นต์ความผิดพลาดเท่ากับ 7.67%, 6.30% และ 2.95% เวลาที่ใช้ไปเท่ากับ 2.56, 11.81 และ 16.52 วินาทีตามลำดับ ในการกำหนดจำนวนรอบการทำงาน 300 และ 1000 รอบ ACS มีผลการทำงานที่ดีกว่า Soft-Ant คือมีค่าเฉลี่ยของผลลัพธ์เท่ากับ 15391.33 และ 14766.33 มีเปอร์เซ็นต์ความผิดพลาดเท่ากับ 6.72% และ 2.39% เวลาที่ใช้ไปเท่ากับ 7.17 และ 23.59 วินาที ตามลำดับ (ตารางที่ 8)

เมื่อนำเปอร์เซ็นต์ความผิดพลาด (%Err) ของอัลกอริทึมระบบอานานิคมมดกับ Soft-Ant ในการแก้ปัญหา ry48p มาเปรียบเทียบกัน พบว่า ที่การกำหนดรอบการทำงาน ทำงาน 100, 500 และ 700 รอบ Soft-Ant จะมีเปอร์เซ็นต์ความผิดพลาดน้อยกว่า ACS แต่ที่การกำหนดรอบการทำงาน 300 และ 1000 รอบ อัลกอริทึม Soft-Ant จะมีเปอร์เซ็นต์ความผิดพลาดมากกว่าอัลกอริทึม ACS (ภาพที่ 2 ฉ)

ผลการทดลองเปรียบเทียบกับอัลกอริทึมอัลกอริทึมระบบอานานิคมมดกับอัลกอริทึมซอฟต์แวร์แอนต์ในการแก้ปัญหาทั้ง 6 ชุดข้อมูล ได้แก่ eil51, berlin52, lin105, pcb442, br17 และ ry48p โดยการกำหนดจำนวนรอบสูงสุดในการทำงานเป็น 100, 300, 500, 700 และ 1000 รอบ จะสามารถสรุปได้ดังนี้ (ตารางที่ 9)

การแก้ปัญหา eil51 อัลกอริทึม ACS มีค่าเฉลี่ยของผลเฉลยที่ดีกว่า มีค่าผิดพลาดที่น้อยกว่า และใช้เวลาน้อยกว่าอัลกอริทึม Soft-Ant แต่มีค่าเบี่ยงเบนมาตรฐานสูงกว่า การแก้ปัญหา berlin52 อัลกอริทึม Soft-Ant มีค่าเฉลี่ยของผลเฉลยที่ดีกว่า มีค่าผิดพลาดที่น้อยกว่า และมีค่าเบี่ยงเบนมาตรฐานต่ำกว่าอัลกอริทึม ACS แต่มีการใช้เวลาในการแก้ปัญหามากกว่า การแก้ปัญหา lin105 อัลกอริทึม Soft-Ant มีค่าเฉลี่ยของผลเฉลยที่ดีกว่า มีค่าผิดพลาดที่น้อยกว่า ใช้เวลาน้อยกว่า และมีค่าเบี่ยงเบนมาตรฐานต่ำกว่าอัลกอริทึม ACS การแก้ปัญหา pcb442 อัลกอริทึม ACS มีค่าเฉลี่ยของผลเฉลยที่ดีกว่า มีค่าผิดพลาดที่น้อยกว่า และใช้เวลาในการแก้ปัญหาน้อยกว่าอัลกอริทึม Soft-Ant แต่มีค่าเบี่ยงเบนมาตรฐานที่สูงกว่า การแก้ปัญหา br17 อัลกอริทึม Soft-Ant มีผลเฉลยที่ดีกว่า มีค่าผิดพลาดที่น้อยกว่า ใช้เวลาน้อยกว่า และมีค่าเบี่ยงเบนมาตรฐานต่ำกว่าอัลกอริทึม ACS การแก้ปัญหา ry48p อัลกอริทึม Soft-Ant มีผลเฉลยที่ดีกว่า มีค่าผิดพลาดที่น้อยกว่า ใช้เวลาน้อยกว่าอัลกอริทึม ACS แต่มีค่าเบี่ยงเบนมาตรฐานสูงกว่า

เมื่อทำการเปรียบเทียบค่าเฉลี่ยของผลเฉลย ค่าเบี่ยงเบนมาตรฐานและเวลาที่ใช้ในการแก้ปัญหาจะพบว่า อัลกอริทึม Soft-Ant สามารถให้ผลลัพธ์ที่ดีกว่าอัลกอริทึมระบบอาณานิคมมด(ACS) คือ สามารถหาผลเฉลยของปัญหาได้ดีกว่า โดยมีค่าเบี่ยงเบนมาตรฐานที่ต่ำกว่า และใช้เวลาในการแก้ปัญหาน้อยกว่า จำนวน 2 ชุด ได้แก่ lin105 และ br17 ส่วนชุดข้อมูลอื่นๆ อัลกอริทึม ACS ก็ไม่สามารถทำได้ดีกว่า อัลกอริทึม Soft-Ant

สรุปและอภิปรายผล

งานวิจัยนี้ผู้วิจัยได้ศึกษาหลักการทำงานของอัลกอริทึมการหาคำตอบด้วยอาณานิคมมด (Ant Colony Optimization) ในบริบทของปัญหาการเดินทางของพนักงานขายเดิน โดยทำการศึกษาอัลกอริทึมระบบอาณานิคมมด (Ant Colony System) ซึ่งส่วนสำคัญส่วนหนึ่งที่มีการปรับปรุงเพิ่มเติมเข้ามาในอัลกอริทึมระบบอาณานิคมมด คือ ได้มีการนำกฎการปรับปรุงฟีโรโมนเฉพาะที่ (Local pheromone update rule) เข้ามาใช้ในกฎการปรับปรุงฟีโรโมนในระดับท้องถิ่นนี้ขึ้นอยู่กับค่าอัตราการระเหย (ρ) หรือค่าควบคุมการสลายตัวของฟีโรโมน เป็นค่าคงที่มีค่าอยู่ในช่วง $0 < \rho < 1$ และค่าฟีโรโมนเริ่มต้น (τ_0) ซึ่งเป็นค่าคงที่มีค่าตามที่คำนวณในตอนกำหนดค่าเริ่มต้นครั้งเดียว ทำให้ส่วนของสมการกฎการปรับปรุงฟีโรโมนระดับท้องถิ่นมีลักษณะเป็นแบบฮาร์ด(Hard Computing) ซึ่งในธรรมชาติยังมีปัจจัยอื่นๆ ที่มีผลต่อการสลายตัวของฟีโรโมน เช่น มดที่มีขนาดไม่เท่ากัน ความแรงของลม ปริมาณแสงแดด เป็นต้น ผู้วิจัยจึงมีแนวในการปรับปรุงกฎการปรับปรุงฟีโรโมนในระดับท้องถิ่น โดยใช้เทคนิคการคำนวณแบบซอฟต์ (Soft Computing) ด้วยการทำให้ค่า $\rho\tau_0$ ในสมการปรับปรุงฟีโรโมนเฉพาะที่ ไม่เป็นค่าคงที่ ผู้วิจัยได้นำค่า $\rho\tau_0$ มาหารด้วยค่าที่ได้จากฟังก์ชัน Sigmoid ของตัวแปรสุ่มที่มีค่าระหว่าง 0 ถึง 1

ผู้วิจัยได้นำแนวคิดดังกล่าวมาพัฒนา ซอฟต์แวร์แอนท์: ระบบมดในการแก้ปัญหาการเดินทางของพนักงานขาย และพัฒนาเป็นโปรแกรมต้นแบบเพื่อทำการทดลอง โดยเลือกใช้ชุดข้อมูลปัญหาการเดินทางของพนักงานขายจาก TSPLIB จำนวน 6 ชุด ที่มีลักษณะต่างๆ ประกอบด้วยปัญหาการเดินทางของพนักงานขายแบบสมมาตร (Symmetric traveling salesman problem) ได้แก่ eil51, berlin52, lin105 และ pcb442 และปัญหาการเดินทางของพนักงานขายแบบอสมมาตร (Asymmetric traveling salesman problem) ได้แก่ br17 และ ry48p

ผลการทดลองเปรียบเทียบกับอัลกอริทึมระบบอาณานิคมมดกับ Soft-Ant ที่ผู้วิจัยพัฒนาขึ้นในการแก้ปัญหาการเดินทางของพนักงานขายทั้ง 6 ชุดข้อมูล ได้แก่ eil51, berlin52, lin105, pcb442, br17 และ ry48p โดยการกำหนดจำนวนรอบสูงสุดในการทำงานเป็น 100, 300, 500, 700 และ 1000 รอบ ผลสรุปพบว่าอัลกอริทึม Soft-Ant ให้ผลลัพธ์ที่ดีกว่าอัลกอริทึมระบบอาณานิคมมดคือสามารถหาผลเฉลยของปัญหาได้ดีกว่า โดยมีค่าเบี่ยงเบนมาตรฐานที่ต่ำกว่า และ

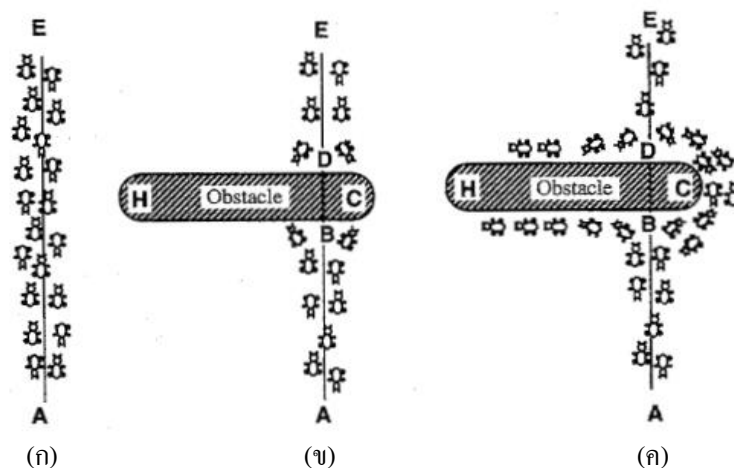


ใช้เวลาในการแก้ปัญหาที่น้อยกว่า จำนวน 2 ชุด ได้แก่ lin105 และ br17 และในชุดข้อมูลอื่นๆ ไม่พบว่าอัลกอริทึม ACS สามารถทำได้ดีกว่า อัลกอริทึม Soft-Ant

การปรับปรุงพัฒนาอัลกอริทึมซอฟต์แวร์แอนท์ สามารถพัฒนาต่อยอด เพื่อเพิ่มประสิทธิภาพให้กับอัลกอริทึม โดยทำการทดลองเพิ่มส่วนของการค้นหาเฉพาะที่ (local search) เช่น 2-opt, 3-opt เข้าไป และสามารถนำไปประยุกต์ใช้ในธุรกิจระบบการจัดการการขนส่งสินค้า (logistics) ที่มีการขนส่ง หรือเคลื่อนย้ายจากจุดต้นทางไปยังจุดบริโภครตามความต้องการของลูกค้า เพื่อช่วยในการกำหนดเส้นทางการจัดส่งสินค้าที่เหมาะสมที่สุด ได้อย่างมีประสิทธิภาพต่อไป

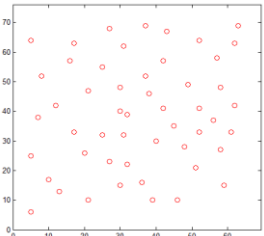
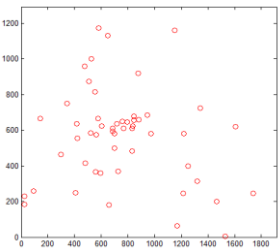
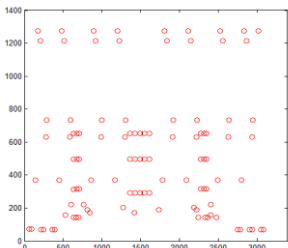
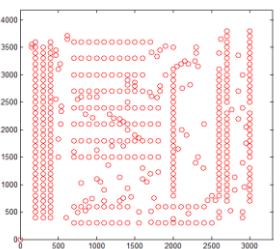
เอกสารอ้างอิง

1. Bonabeau E, Dorigo M, Theraulaz G. Swarm Intelligence: From Natural to Artificial Systems. Oxford University Press, 1999.
2. Zadeh LA. Soft computing and fuzzy logic. IEEE Softw. 1994; 11:48–56.
3. Pensuwon W, Adams RG, Davey N. Human Face Recognition using Soft Computing RBF. TENCON 2006 – 2006, IEEE Region 10 Conference; 2006 Nov 14-17. Hong Kong.
4. Dorigo M, Maniezzo V, Colomi A. Positive Feedback as a Search Strategy. Technical Report n.91-016; 1991. Politecnico di Milano, Italy.
5. Dorigo M, Birattari M, Stutzle T. Ant colony optimization. IEEE Comput. Intell. Mag. 2006; 1(4):28–39.
6. Dorigo M, Stützle T. Ant Colony Optimization. Cambridge. A Bradford Book; 2004.
7. Dorigo M, Gambardella LM. Ant colony system: a cooperative learning approach to the traveling salesman problem, IEEE Trans. Evol. Comput. 1997; 1(1):53–66.
8. Dorigo M, Maniezzo V, Colomi A. The Ant System: Optimization by a colony of cooperating agents. IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics). 1996; 26(1):29–41.
9. Reinelt G. TSPLIB—A Traveling Salesman Problem Library, ORSA J. Comput. 1991;3(4):376–84.



ภาพที่ 1 การจำลองพฤติกรรมของมดในธรรมชาติ (ก)เส้นทางการเดินของมดจากจุด A ไปจุด E (ข)เมื่อมีสิ่งกีดขวางเข้าบนเส้นทาง มดจะเลือกเดินอ้อมทำให้เกิดเส้นทางใหม่สองเส้นทาง (ค)บนเส้นทางที่สั้นกว่าจะมีมดเดินผ่านมากกว่าและมีการปล่อยสารฟีโรโมนบนเส้นทางนั้นมากกว่า [8]

ตารางที่ 1 ชุดข้อมูลจาก TSPLIB ที่ใช้ในการทดลอง ได้แก่ eil51, berlin52, lin105, pcb442, br17 และ ry48p

	ชุดข้อมูล (ผลเฉลย)	ขนาด ของปัญหา	ประเภทของ ปัญหา	ลักษณะข้อมูล	ลักษณะเด่น ของข้อมูล
1	eil51 (426)	51	TSP		ข้อมูลมีลักษณะ กระจายสม่ำเสมอเต็ม พื้นที่
2	berlin52 (7542)	52	TSP		ข้อมูลมีลักษณะ กระจายไม่สม่ำเสมอ บางจุดมีการรวมตัว เป็นกระจุก
3	lin105 (14379)	105	TSP		ข้อมูลมีลักษณะเรียง กันเป็นระเบียบ แต่ไม่ หนาแน่น
4	pcb442 (50778)	442	TSP		ข้อมูลมีลักษณะเรียง กันเป็นระเบียบ แต่มี ความหนาแน่นสูง
5	br17 (39)	17	ATSP	-	-
6	ry48p (14422)	48	ATSP	-	-



ตารางที่ 2 ค่าพารามิเตอร์สำหรับอัลกอริทึม ACS และอัลกอริทึม Soft-Ant ที่ใช้ในการทดลอง

พารามิเตอร์	อัลกอริทึม ACS	อัลกอริทึม Soft-Ant
จำนวนมด	10	10
α	1	1
β	2	2
ρ, ζ	0.1	0.1
τ_0	$(nL_{nn})^{-1}$	$(nL_{nn})^{-1}$
q_0	0.9	0.9
จำนวนรอบการทำงาน	100,300,500,700,1000	100,300,500,700,1000

ตารางที่ 3 ผลการทดลองเปรียบเทียบอัลกอริทึมระบบอาณานิคมมดกับ Soft-Ant ในการแก้ปัญหา eil51

จำนวน รอบ	ACS				Soft-Ant			
	AVG	SD	%Err	Time(s)	AVG	SD	%Err	Time(s)
100	461.33	11.24	8.29	3.08	456.67	12.34	7.20	2.82
300	446.33	15.63	4.77	8.69	447.00	10.58	4.93	8.12
500	437.33	2.08	2.66	14.62	439.67	8.02	3.21	14.09
700	439.00	3.61	3.05	18.47	442.33	3.51	3.83	18.09
1000	431.00	2.00	1.17	25.71	436.67	8.08	2.50	26.94
เฉลี่ยรวม	443.00	11.61	3.99	14.11	444.47	7.81	4.33	14.01

ตารางที่ 4 ผลการทดลองเปรียบเทียบ อัลกอริทึมระบบอาณานิคมมดกับ Soft-Ant ในการแก้ปัญหา Berlin52

จำนวนรอบ	ACS				Soft-Ant			
	AVG	SD	%Err	Time(s)	AVG	SD	%Err	Time(s)
100	8227.33	288.39	9.09	3.23	8090.67	345.88	7.27	3.09
300	7717.00	197.41	2.32	8.46	7801.00	34.00	3.43	8.40
500	7784.33	71.57	3.21	14.43	7776.33	191.89	3.11	14.37
700	8052.00	203.45	6.76	18.77	7678.00	205.17	1.80	18.65
1000	7587.67	74.81	0.61	26.55	7931.00	25.16	5.16	27.78
เฉลี่ยรวม	7873.67	260.39	4.40	14.29	7855.40	159.49	4.16	14.46

ตารางที่ 5 ผลการทดลองเปรียบเทียบ อัลกอริทึมระบบอาณานิคมมดกับ Soft-Ant ในการแก้ปัญหา Lin105

จำนวนรอบ	ACS				Soft-Ant			
	AVG	SD	%Err	Time(s)	AVG	SD	%Err	Time(s)
100	16244.67	649.22	12.97	9.59	16040.67	418.76	11.56	11.73
300	15481.00	390.85	7.66	28.10	15565.67	406.86	8.25	28.09
500	15270.33	141.99	6.20	60.40	15007.67	179.22	4.37	48.64
700	15202.00	323.21	5.72	62.99	15224.00	238.51	5.88	63.23
1000	14868.00	240.16	3.40	90.27	15121.00	283.88	5.16	94.00
เฉลี่ยรวม	15413.20	514.34	7.19	50.27	15391.80	418.42	7.04	49.14

ตารางที่ 6 ผลการทดลองเปรียบเทียบ อัลกอริทึมระบบอาณานิคมมดกับ Soft-Ant ในการแก้ปัญหา pcb442

จำนวนรอบ	ACS				Soft-Ant			
	AVG	SD	%Err	Time(s)	AVG	SD	%Err	Time(s)
100	60641.33	1815.74	19.42	143.98	59593.33	225.39	17.36	145.97
300	58678.00	1024.35	15.56	427.08	60820	865.32	19.78	430.71
500	59805.67	2253.72	17.78	742.35	59489.00	553.80	17.16	717.53
700	59641.33	111.43	17.46	968.01	59421.33	1094.84	17.02	967.81
1000	59141.67	2255.12	16.47	1379.49	58805.67	1463.80	15.81	1445.95
เฉลี่ยรวม	59581.60	739.40	17.34	732.18	59625.86	734.90	17.42	741.59

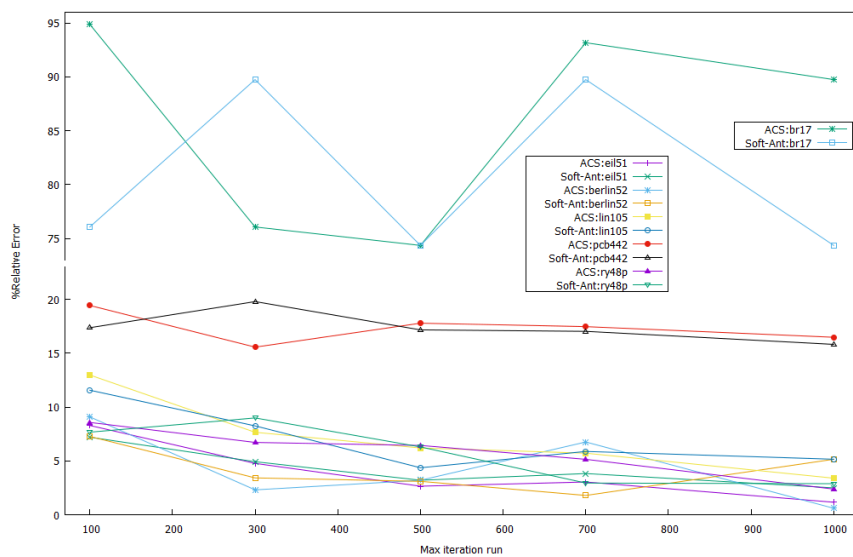
ตารางที่ 7 ผลการทดลองเปรียบเทียบ อัลกอริทึมระบบอาณานิคมมดกับ Soft-Ant ในการแก้ปัญหา br17

จำนวนรอบ	ACS				Soft-Ant			
	AVG	SD	%Err	Time(s)	AVG	SD	%Err	Time(s)
100	76.00	0.00	94.87	0.57	68.67	11.02	76.07	0.53
300	68.67	11.02	76.07	1.63	74.00	0.00	89.74	1.53
500	68.00	10.39	74.36	2.72	68.00	10.39	74.36	2.66
700	75.33	1.15	93.16	3.69	74.00	0.00	89.74	3.77
1000	74.00	0.00	89.74	5.62	68.00	10.39	74.36	5.21
เฉลี่ยรวม	72.40	3.79	85.64	2.85	70.53	3.18	80.85	2.74



ตารางที่ 8 ผลการทดลองเปรียบเทียบเกี่ยวกับอัลกอริทึมระบบอาณานิคมมดกับ Soft-Ant ในการแก้ปัญหา ry48p

จำนวน รอบ	ACS				Soft-Ant			
	AVG	SD	%Err	Time(s)	AVG	SD	%Err	Time(s)
100	15658.00	450.45	8.57	2.54	15528.67	68.82	7.67	2.56
300	15391.33	195.46	6.72	7.17	15718.33	173.42	8.99	6.95
500	15351.33	331.68	6.44	11.76	15331.00	476.47	6.30	11.81
700	15164.33	230.26	5.15	16.51	14847.00	368.39	2.95	16.52
1000	14766.33	260.54	2.39	23.59	14839.33	298.66	2.89	23.56
เฉลี่ยรวม	15266.27	330.40	5.85	12.31	15252.87	398.30	5.76	12.28



ภาพที่ 2 เปรียบเทียบเปอร์เซ็นต์ความผิดพลาดของอัลกอริทึมระบบอาณานิคมมดกับ Soft-Ant ในการแก้ปัญหา eil51, berlin52, lin105, pcb442, br17 และ ry48p

ตารางที่ 9 สรุปค่าเฉลี่ยรวมของผลลัพธ์, ค่าเบี่ยงเบนมาตรฐาน, เปอร์เซ็นต์ความผิดพลาดและเวลาที่ใช้ในการแก้ปัญหาของอัลกอริทึมระบบอาณานิคมมดและอัลกอริทึมซอฟต์แวร์แอนท์

ชุดข้อมูล	ACS				Soft-Ant			
	AVG	SD	%Err	Time(s)	AVG	SD	%Err	Time(s)
eil51	443.00	11.61	3.99	14.11	444.47	7.81	4.33	14.01
berlin52	7873.67	260.39	4.40	14.29	7855.40	159.49	4.16	14.46
lin105	15413.20	514.34	7.19	50.27	15391.80	418.42	7.04	49.14
pcb442	59581.60	739.40	17.34	732.18	59625.87	734.90	17.42	741.59
br17	72.40	3.79	85.64	2.85	70.53	3.18	80.85	2.74
ry48p	15266.27	330.40	5.85	12.31	15252.87	398.30	5.76	12.28