การเปรียบเทียบสมรรถนะของการค้นหาด้วยวิธีจีนเนติกอัลกอริทึมกับวิธีตาบู

กองพัน อารีรักษ์[1] และ สราวุฒิ สุจิตจร[2*]

Areerak, K-N.[1] and Sujitjorn, S.[2] (2002).* **Performance Comparison between Genetic Algorithm and Tabu Search Methods.** *Suranaree J. Sci. Technol. 9:61-68*

Abstract

This article reviews Genetic Algorithm (GA) and Tabu Search (TS) methods that have been applied to search for optimum solution under nonlinear constriant(s). The explanation of a new search method namely Adaptive Tabu Search (ATS) is given. The comparison of searching performance is conducted under an environment of having many local optimum solutions. It is found that the ATS method gives the best searching performance.

บทคัดย่อ

บทความนำเสนอการทบทวนวิธีจีนเนติกอัลกอริทึม วิธีตาบู ที่ใช้เพื่อการค้นหาคำตอบเหมาะที่สุด แก่ปัญหาที่มีเงื่อนไขบังคับแบบไม่เป็นเชิงเส้น จากนั้นนำเสนอวิธีการใหม่ที่เรียกว่า อะแดพทีฟตาบู ให้รายละเอียดการเปรียบเทียบสมรรถนะการค้นหาคำตอบในสภาพแวดล้อมที่เต็มไปด้วยคำตอบวงแคบเฉพาะถิ่น พบว่าวิธีอะแดพทีฟตาบูดำเนินงานได้ดีที่สุด

บทนำ

ปัญหาส่วนใหญ่ที่เกี่ยวข้องกับการหาค่าพารามิเตอร์ภายใต้เงื่อนไขบังคับ อาจดำเนินการแก้ไขปัญหาได้โดยอาศัยวิธีการหาค่าเหมาะที่สุด ขั้นตอนแรกในการหาค่าเหมาะที่สุด ต้องระบุถึงเซตของพารามิเตอร์, ฟังก์ชันวัตถุประสงค์ที่นำมาใช้ในการประเมินคุณภาพของคำตอบ และเงื่อนไขบังคับของปัญหาว่าเป็นอย่างไร จากนั้นการค้นหาคำตอบจึงเกิดขึ้นให้ได้เป็นคำตอบที่เหมาะที่สุด หมายความว่าคำตอบดังกล่าวจะ

[1] *นักวิจัย สาขาวิชาวิศวกรรมไฟฟ้า สำนักวิชาวิศวกรรมศาสตร์ มหาวิทยาลัยเทคโนโลยีสุรนารี อ.เมือง จ.นครราชสีมา 30000*

[2*] *Ph.D., รองศาสตราจารย์ หัวหน้าสาขาวิชาวิศวกรรมไฟฟ้า สำนักวิชาวิศกรรมศาสตร์ มหาวิทยาลัยเทคโนโลยีสุรนารี อ.เมือง จ.นครราชสีมา 30000*

* *ผู้เขียนที่ให้การติดต่อ*

ต้องให้ค่าที่ได้จากฟังก์ชันวัตถุประสงค์เป็นค่าที่ดีที่สุดเมื่อเปรียบเทียบกับคำตอบอื่น ๆ ที่เป็นไปได้ที่อยู่ภายใต้เงื่อนไขบังคับ อัลกอริทึมที่นำมาใช้สำหรับการแก้ปัญหาการหาค่าเหมาะที่สุดที่นิยมใช้กันมานานก็เช่น การโปรแกรมเชิงเส้นและไม่เชิงเส้น ซึ่งมีข้อจำกัดอยู่ที่ฟังก์ชันต่าง ๆ ที่เกี่ยวข้องต้องเรียบหรือโค้งมน (smooth) ในระยะหลัง ๆ มานี้ แนวทางที่เริ่มใช้งานกันแพร่หลายอาศัยการค่อยๆ ปรับเปลี่ยนคำตอบจากที่มีอยู่ไปเรื่อยๆ จนกว่าจะได้คำตอบที่ยอมรับได้ วิธีการปรับเปลี่ยนคำตอบดำเนินการได้หลายแบบ อาทิเช่น วิธีตาบู (Tabu Search หรือ TS) วิธีจีนเนติกอัลกอริทึม (Genetic Algorithm หรือ GA) วิธีโปรแกรมเชิงวิวัฒนาการ (Evolutionary Programming หรือ EP) เป็นต้น (Pham and Karaboga, 2001) ได้มีผู้วิจัยนำวิธีตาบูและวิธีจีนเนติกอัลกอริทึม ไปใช้งานเกี่ยวกับการแก้ไขปัญหาการหาค่าเหมาะที่สุดไว้มากมาย เช่น การนำจีนเนติกอัลกอริทึมมาทำการค้นหาสัมประสิทธิ์อนุกรมฟูริเยร์ แทนวิธีการอินทิเกรตโดยตรง ซึ่งในทางปฏิบัติมีความยุ่งยากซับซ้อน (Qi-Wen Yang *et al.*, 2000) การค้นหาพารามิเตอร์ของตัวควบคุมพีไอดี ด้วยวิธีจีนเนติกอัลกอริทึม (Wang, P. and Kwok, D. P., 1992) การนำจีนเนติกอัลกอริทึมมาใช้สำหรับการปรับค่าพารามิเตอร์ของตัวควบคุมพีไอดีให้ทันการณ์กับระบบที่มีการเปลี่ยนแปลงตลอดเวลา (Van Rensburg, P. J., Shaw, I. S., and Van Wyk, J. D., 1998) การนำจีนเนติกอัลกอริทึมมาประเมินความปลอดภัยในระบบไฟฟ้ากำลัง (Huang, S. J., 2001) การวางแผนการจ่ายไฟให้กับผู้บริโภคโดยเสียค่าใช้จ่ายน้อยที่สุด และได้ผลเป็นที่น่าพอใจด้วยวิธีจีนเนติกอัลกอริทึม (Aldridge, C. J., *et al.*, 2001) การนำวิธีตาบูมาใช้กับปัญหาการวางแผนการขยายเครือข่ายการส่ง (transmission network) (Da Silva, E.L. *et al.*, 2001) การนำวิธีตาบูมาดำเนินการจัดสรรการจ่ายพลังงานของเครื่องกำเนิดไฟฟ้าให้เกิดพลังงานสูญเสียน้อยที่สุด (Nara, K. *et al.*, 2001) การนำวิธีตาบูมาประยุกต์ใช้เพื่อระบุเอกลักษณ์ไม่เป็นเชิงเส้น ที่ปรากฏในระบบสองมวลความเฉื่อย (กองพัน อารีรักษ์, 2544) การนำวิธีตาบูมาใช้ สำหรับการคำนวณหาการไหลของกำลังไฟฟ้าที่เหมาะที่สุด (Sujitjorn, S. and Kulworawanichpong, T., 2001) เป็นต้น การนำวิธีการเหล่านี้ไปประยุกต์ใช้งาน จำเป็นต้องมีการประเมินความสามารถในการค้นหาคำตอบก่อนนำไปใช้งานจริงและยังได้พบข้อด้อยของวิธีการต่าง ๆ เช่น ดำเนินงานช้า การค้นหาคำตอบถูกล็อคด้วยคำตอบวงแคบเฉพาะถิ่นได้ง่าย เป็นต้น บทความนี้จึงนำเสนอการเปรียบเทียบสมรรถนะในการค้นหาคำตอบของวิธีตาบูและวิธีจีนเนติกอัลกอริทึม รวมถึงวิธีอะแดพทีฟตาบู (ATS) ที่ได้พัฒนาขึ้น เพื่อนำมาใช้ในการแก้ไขปัญหาการหาค่าเหมาะที่สุดได้อย่างมีประสิทธิภาพ

อุปกรณ์และวิธีการ

หลักการทำงานของจีนเนติกอัลกอริทึม

จีนเนติกอัลกอริทึม หรือ GA เป็นเครื่องมือหนึ่งที่ใช้เพื่อแก้ปัญหาการหาค่าเหมาะที่สุดได้อย่างมีประสิทธิภาพ แนวทางการดำเนินงานนั้นเลียนแบบกระบวนการทางพันธุกรรมในธรรมชาติ หลักการทำงานของ GA ประกอบด้วย 3 ส่วน ดังนี้

1. การคัดเลือก (selection) หมายถึง กระบวนการคัดเลือกพ่อพันธุ์แม่พันธุ์ (parents) จากการสุ่มประชากร (population) เริ่มต้น โดยพิจารณาถึงค่าความเหมาะสม (fitness) เป็นสิ่งสำคัญ จากนั้นจะนำพ่อพันธุ์แม่พันธุ์ที่ได้จากการเลือกไปดำเนินการทาง GA ต่อไป

2. ปฏิบัติการทางสายพันธุ์ (genetic operation) แบ่งออกเป็นสองส่วนใหญ่ ๆ คือ crossover และ mutation เป็นขั้นตอนที่ใช้สำหรับนำพ่อพันธุ์แม่พันธุ์ที่ผ่านการคัดเลือกตามขั้นตอนที่1 มาทำการขยายพันธุ์ให้เกิดเป็นโครโมโซมลูกหลาน (offspring) ออกมาชุดใหม่ การดำเนินการค้นหาด้วยวิธี GA นั้นอาจมีทั้งการทำ crossover และ mutation ควบคู่ไปด้วยกันอย่างเหมาะสม ต่อการแก้ปัญหาเป็นรายกรณี

3. การแทนที่ (replacement) หมายถึง กระบวนการนำโครโมโซมลูกหลานที่มีค่าความเหมาะสมมากนำไปแทนที่ในประชากรเริ่มต้นเพื่อจะดำเนินการตามปฏิบัติการทางสายพันธุ์ ในชั่วอายุต่อไป

ขั้นตอนการดำเนินงานค้นหาด้วยวิธี GA

ขั้นตอนที่ 1 ดำเนินการสุ่มประชากรเริ่มต้นเพื่อนำมาคัดเลือกพ่อพันธุ์แม่พันธุ์

ขั้นตอนที่ 2 ดำเนินการคัดเลือกพ่อพันธุ์แม่พันธุ์โดยดูจากค่าความเหมาะสมที่ได้จากฟังก์ชันกำหนดค่าความเหมาะสม (fitness function)

ขั้นตอนที่ 3 นำพ่อพันธุ์แม่พันธุ์ที่ผ่านการคัดเลือกจากขั้นตอนที่ 2 มาดำเนินการทาง GA เพื่อให้ได้โครโมโซมลูกหลานชุดใหม่ออกมา

ขั้นตอนที่ 4 ดำเนินการตรวจสอบว่าโครโมโซมลูกหลานชุดใดมีค่าความเหมาะสม เพื่อนำไปแทนที่เป็นประชากรเริ่มต้นในรอบต่อไป ในการเลือกเพื่อจะไปทำการแทนที่นี้จะพิจารณาทั้งพ่อพันธุ์แม่พันธุ์และลูกหลานควบคู่กันไป ถ้าใครมีค่าความเหมาะสมมากจะเข้ารอบต่อไป แต่ถ้าใครมีค่าความเหมาะสมน้อยจะตัดทิ้ง

ขั้นตอนที่ 5 ตัวที่อยู่รอดจนการแข่งขันมาถึงรอบสุดท้าย จะเป็นตัวแข็งแรงที่สุด ในที่นี้ก็คือจะเป็นคำตอบที่ดีที่สุด

หลักการทำงานของวิธีตาบู

การค้นหาแบบตาบู หรือ TS เป็นขั้นตอนวิธีการแบบตรรกที่นำมาประยุกต์เกี่ยวกับการหาคำตอบที่ดีที่สุดเชิงผสมผสาน (combinatorial optimization) ได้อย่างมีประสิทธิภาพ องค์ประกอบของวิธีการค้นหาแบบตาบูที่แตกต่างจากวิธีการค้นหาแบบอื่น ๆ คือ มีเกณฑ์ความเป็นตาบู (tabu list criterion) และเกณฑ์ความปรารถนา (aspiration criterion) ซึ่ง

"เกณฑ์ความเป็นตาบู" เป็นส่วนที่คอยเก็บข้อมูลของคำตอบในอดีตของกระบวนการค้นหานั้นๆ เพื่อเป็นตัวกำหนดการค้นหาคำตอบว่าจะมีทิศทางไปทางใด หลักการออกแบบเกณฑ์ความเป็นตาบู จะมีลักษณะแตกต่างกันออกไปขึ้นอยู่กับปัญหาแต่ละชนิด

"เกณฑ์ความปรารถนา" เป็นเงื่อนไขที่อาจจะต้องใช้ในบางครั้งเมื่อจำเป็นต้องเลือกคำตอบที่อยู่ในเกณฑ์ความเป็นตาบู งานบางชนิดที่ปัญหาไม่ซับซ้อนไม่จำเป็นต้องพึ่งส่วนนี้ก็ได้ เกณฑ์ความเป็นตาบูอย่างเดียวก็เพียงพอที่จะค้นหาคำตอบที่ดีที่สุด

ความหมายของคำศัพท์ที่ใช้ในการอธิบายการทำงาน

radius ขอบเขตของการสุ่มในแต่ละรอบการทำงาน

number_neighbor จำนวนค่าใกล้เคียงที่ได้จากการสุ่ม (neighborhood) ในแต่ละพื้นที่การค้นหา (search_space)

neighbor_list ส่วนที่เก็บค่าใกล้เคียงตามจำนวนที่กำหนด (tabu list)

best_local ค่าใกล้เคียงที่เป็นคำตอบวงแคบเฉพาะถิ่น (local optimal)

best_global ค่าใกล้เคียงที่เป็นค่าที่ดีที่สุดสำหรับคำตอบที่เป็นวงกว้าง (global optimal)

n จำนวนรอบในการค้นหาคำตอบ

xlimit ขอบเขตของพารามิเตอร์แต่ละตัว

S_0 ค่าเริ่มต้นในแต่ละพื้นที่การค้นหา

$S_1(r)$ เซตของจำนวนสมาชิกที่ได้จากการสุ่ม มีจำนวนเท่ากับ number_neighbor

$S_1, S_2, \ldots, S_n$ ค่าที่เก็บไว้ใน neighbor_list

cost ค่าที่ได้จากฟังก์ชันวัตถุประสงค์

ขั้นตอนการดำเนินงานค้นหาด้วยวิธี TS

ขั้นตอนที่ 1 หาค่าเริ่มต้น S_0 จากการสุ่มคำตอบ จากนั้นกำหนดให้ S_0 เป็นคำตอบที่ดีที่สุด best_global = S_0

ขั้นตอนที่ 2 ดำเนินการเคลื่อนย้ายในลักษณะสุ่มรอบ ๆ S_0 ภายในพื้นที่การค้นหา จนได้จำนวนค่าใกล้เคียงของ S_0 เท่ากับ number_neighbor ดังที่ได้อธิบายไว้ในรูปที่ 2 กำหนดให้เซตของสมาชิกที่ได้จากการสุ่ม คือ $S_1(r)$

ขั้นตอนที่ 3 คำนวณค่า cost จากฟังก์ชันวัตถุประสงค์ของสมาชิกแต่ละตัวที่อยู่ใน $S_1(r)$ และเลือกคำตอบที่ดีที่สุด กำหนดให้เป็น best_local

ขั้นตอนที่ 4 ถ้าค่า cost ของ best_local มีค่าน้อยกว่าค่า cost ของ best_global กำหนดให้ best_global = best_local

ถ้าไม่สามารถหาสมาชิกตัวใดใน $S_1(r)$ ที่

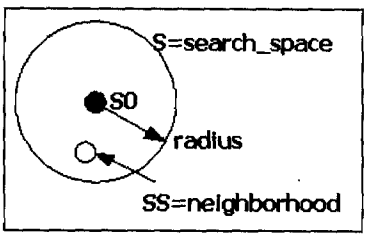

Figure 1. Search space.

ให้ค่า cost ดีกว่าค่า cost ของ S_0 ให้ไปทำในขั้นตอนที่ 6

ขั้นตอนที่ 5 กำหนดให้

S_0 = best_global

ขั้นตอนที่ 6 ถ้า best_local ไม่อยู่ใน neighbor_list ให้เก็บค่า best_local ไว้ใน neighbor_list

ขั้นตอนที่ 7 ตรวจสอบเงื่อนไขการยุติการค้นหา ถ้าเป็นไปตามเงื่อนไขจึงหยุดการค้นหา และคำตอบของการค้นหารอบล่าสุดจะถือว่าเป็นคำตอบที่ดีที่สุด แต่เมื่อตรวจสอบแล้วไม่เป็นไปตามเงื่อนไข ให้ไปเริ่มทำในขั้นตอนที่ 2 ใหม่

หลักการทำงานของวิธีอะแดพทีฟตาบู

การทำงานในส่วนของวิธีอะแดพทีฟตาบู หรือ วิธี ATS จะมีลักษณะเหมือนกับวิธี TS เพียงแต่เพิ่มการทำงานในส่วนของการย้อนรอยค้นหา (back_tracking) แทรกระหว่างขั้นตอนที่ 6 และ 7 และ adaptive radius ต่อท้ายขั้นตอนที่ 7 ก่อนจะเริ่มทำงานในขั้นตอนที่ 2 ใหม่ ส่วนที่เป็น back_tracking นั้น อัลกอริทึมจะยอมให้ย้อนกลับไปค้นหาคำตอบเก่าที่เคยผ่านมาแล้ว เพื่อนำไปเปรียบเทียบค่าที่มีอยู่ในปัจจุบันแล้วจึงเลือกค่าที่ดีกว่า ในส่วนของ adaptive radius อัลกอริทึมจะปรับรัศมีของพื้นที่ที่ทำการค้นหาให้แคบลง ๆ จนกว่าจะพบคำตอบที่เป็น global optimum หรือยุติการค้นหาตามเงื่อนไขที่กำหนด กระบวนการทำงานในส่วนที่เพิ่มขึ้น อธิบายได้ดังนี้

back_tracking

ดำเนินการกำหนดให้

S_0 = neighbor_list(3,:)

การทำงานในส่วนนี้แสดงไว้ดังรูปที่ 3 การกำหนดค่า S_0 = neighbor_list(3,:) หมายถึงการกำหนดค่า S_0 ให้เท่ากับค่าในแถวที่สามของ neighbor_list ซึ่งไม่จำเป็นต้องกำหนดค่าตามนี้เสมอไป อาจจะทำการกำหนดค่า S_0 ให้มีค่าเท่ากับตำแหน่งอื่นที่เก็บอยู่ใน neighbor_list ก็ได้

จากรูปที่ 3 จะสังเกตได้ว่าเมื่อมีการทำ back_tracking จะเกิดพื้นที่ใหม่เพื่อใช้ในการค้นหาคำตอบในส่วนนี้เองที่จะทำให้การค้นหาคำตอบสามารถให้คำตอบที่หลุดจากคำตอบวงแคบเฉพาะถิ่นได้

adaptive radius

การกำหนดขอบเขตพื้นที่การค้นหาเป็นสิ่งสำคัญ (ในที่นี้ค่า radius เป็นตัวกำหนดดังแสดงไว้ในรูปที่ 1) เนื่องจากถ้าทำการกำหนดพื้นที่ การค้นหากว้าง จะทำให้การดำเนินการค้นหาคำตอบมีความละเอียดน้อยด้วยเวลาค้นหาสั้น ๆ แต่ในทางกลับกัน ถ้าทำการกำหนดพื้นที่การค้นหาแคบเกินไป ก็จะเป็นการเพิ่มภาระการค้นหาให้ใช้เวลามากขึ้น ดังนั้นในบทความนี้จึงเสนอวิธีการปรับค่า radius ให้มีค่าที่เหมาะสมกับการค้นหาคำตอบ โดยดูที่ค่า best_cost เป็นเกณฑ์ การค้นหาคำตอบจะดำเนินไประยะหนึ่งจนใกล้จะพบคำตอบที่ดีที่สุด (best_cost มีค่าน้อยมากในกรณีของปัญหาการหาค่าต่ำสุด) จะทำการปรับค่า radius ให้มี

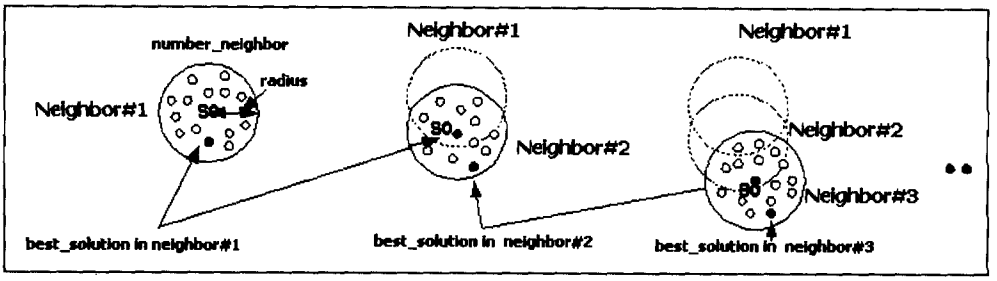

Figure 2. Searching procedures of the conventional TS method.

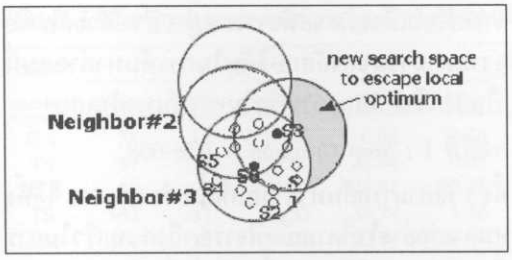

Figure 3. Back_tracking mechanism.

ค่าน้อยลง ดังความสัมพันธ์

$$radius_{new} = radius_{old} /10$$

เพื่อเพิ่มความละเอียดในการค้นหา การทำงานในส่วนนี้อธิบายได้ดังรูปที่ 4

ค่า best_cost ที่นำมาเป็นเกณฑ์ในแต่ละปัญหาไม่จำเป็นต้องเท่ากันขึ้นอยู่กับลักษณะของปัญหา ในบทความนี้จะใช้ค่าดังกล่าวตามรูปที่ 4 ซึ่งเป็นค่าที่ได้จากการคำนวณซ้ำ ๆ เพื่อศึกษาพฤติกรรมของระบบ

ปัญหาที่นำมาใช้ในการเปรียบเทียบสมรรถนะ

ปัญหาที่นำมาใช้ในการเปรียบเทียบสมรรถนะของวิธีการค้นหา มีรูปแบบแสดงดังความสัมพันธ์ (1) คือ การหาค่า x และ y ที่ทำให้ f(x,y) มีค่าต่ำสุด (minimization problem) ฟังก์ชัน f(x,y) ในที่นี้กำหนดให้เป็นฟังก์ชันวัตถุประสงค์คิดค้นโดย Bahachevsky *et al.* (1986) ฟังก์ชันดังกล่าวเป็นที่นิยมใช้กันทั่วไปในการนำมาเป็นฟังก์ชันทดสอบอัลกอริทึม เนื่องจากปรากฏคำตอบที่เป็นวงแคบเฉพาะถิ่น จำนวนมากตามแนวพื้นผิว ภาพ 3 มิติของฟังก์ชัน Bahachevsky แสดงไว้ในรูปที่ 5

```
if best_cost<=1.5e-08
   radius= radius/10;
end
if best_cost<=1.5e-10
   radius= radius/10;
end
if best_cost<=1.5e-14
   radius= radius/10;
end
```

Figure 4. Subalgorithum for adaptive radius.

การดำเนินการทดสอบ จะพิจารณาอยู่ 2 ส่วน คือความสามารถในการค้นหาคำตอบที่เป็นวงกว้าง (global optimal) และความเร็วในการค้นหาคำตอบ ซึ่งดูได้จากเวลาค้นหา (search time) หรือจำนวนรอบ (count) ที่ใช้ในการทดสอบ การทดสอบในบทความนี้ใช้โปรแกรม MATLABTM ทำงานบน Pentium III 933 MHz RAM 256 Mbytes HD 20 Gbytes

$$f(x,y) = x^2 + 2y^2 + -0.3\cos(3\pi x) - 0.4\cos(4\pi y) + 0.7 \quad (1)$$

ขั้นตอนการทดสอบอัลกอริทึม

ขั้นตอนที่ 1. ทำการสุ่มคำตอบที่ดีที่สุดที่เป็นไปได้เป็นค่าเริ่มต้น

ขั้นตอนที่ 2. นำ GA ทำการค้นหาค่า x และ y จนได้ค่า f(x,y) มีค่าน้อยกว่า stop criterion (ซึ่งมีค่าเท่ากับ 1.5e-008) หรือมีจำนวนรอบการค้นหาสูงสุดเท่ากับ 10,000 รอบ จึงหยุดการค้นหา

ขั้นตอนที่ 3. จัดเก็บคำตอบ x, y, f(x,y) เวลาค้นหา และจำนวนรอบ ลงบนคอมพิวเตอร์ (ค่า fitness function และค่า cost function ของวิธีการค้นหา ได้รวบรวมไว้ในภาคผนวก)

ขั้นตอนที่ 4. ดำเนินการซ้ำตามขั้นตอนที่ 1 ถึง 3 โดยเปลี่ยนค่า stop criterion เป็น 1.5e-010 และ 0 ตามลำดับ

ขั้นตอนที่ 5. ดำเนินการเหมือนขั้นตอนที่ 1 ถึง 4

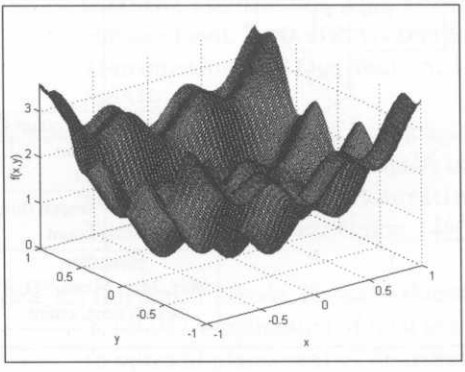

Figure 5. Bahachevsky surface.

แต่เปลี่ยนวิธีการค้นหาเป็นวิธีตาบูแบบดั้งเดิม และวิธีอะแดพทีฟตาบูที่ได้พัฒนาขึ้น ตามลำดับ

ขั้นตอนที่ 6. กำหนดค่า x = - 0.027 และ y = 0.008 เป็นค่าเริ่มต้นแบบตายตัวแทนการสุ่มคำตอบ และดำเนินการซ้ำตามขั้นตอนที่ 1 ถึง 4 ใหม่ ในขั้นตอนนี้จะใช้เฉพาะวิธี TS และ ATS เท่านั้น เนื่องจากวิธี GA ไม่สามารถกำหนดค่าเริ่มต้นแบบตายตัวได้

ขั้นตอนที่ 7. ในแต่ละกรณีจะทำทั้งหมด 15 ครั้ง เพื่อนำผลที่ได้มาหาค่าเฉลี่ย

ขั้นตอนที่ 8. นำค่าข้อมูลของเวลาค้นหา และจำนวนรอบในแต่ละกรณี มาทำการหาค่า max, min, mean และ standard deviation ภาพรวมของการดำเนินการทดสอบแสดงไว้ดังแผนภูมิในรูปที่ 6

ผลการทดลองและวิจารณ์

ผลการเปรียบเทียบสมรรถนะของการค้นหาระหว่างวิธี GA, TS และ ATS ได้รวบรวมไว้ในตารางที่ 1

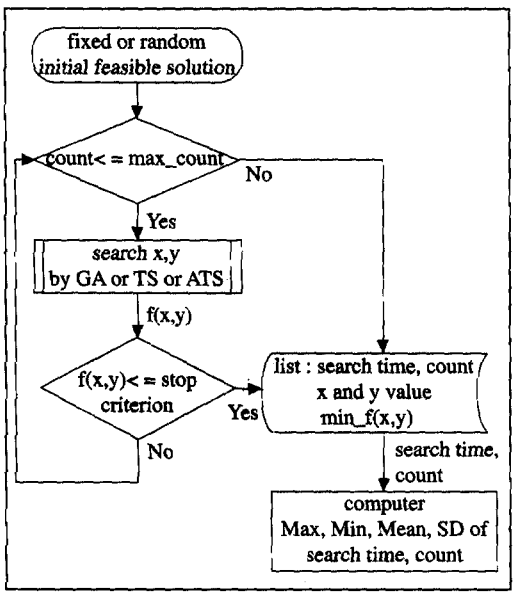

Figure 6. Flow chart for the tests of the GA, TS and ATS algorithums.

การเปรียบเทียบผลจะพิจารณาอยู่ 2 ส่วนด้วยกัน คือ ความสามารถของอัลกอริทึมในการค้นหาคำตอบที่เป็นวงกว้าง และความเร็วในการค้นหาคำตอบ

กรณีที่ 1 : Stop Criterion = 1.5e-008

ทั้ง 3 วิธีสามารถค้นหาคำตอบที่ต่ำกว่า 1.5e-008 ได้ทั้งหมด แต่อย่างไรก็ตามเมื่อพิจารณาถึงความเร็วในการค้นหาคำตอบ วิธี ATS ที่อาศัยค่าเริ่มต้นจากการสุ่มจะใช้เวลาในการค้นหาคำตอบเร็วที่สุด คือ 0.39 วินาที และวิธี GA จะใช้เวลาในการค้นหาคำตอบนานที่สุด คือ 0.54 วินาที

กรณีที่ 2 : Stop Criterion = 1.5e-010

ในกรณีนี้ วิธี GA ไม่สามารถหาคำตอบที่ต่ำกว่า 1.5e-010 ได้ แต่วิธีการของ TS และ ATS สามารถค้นหาคำตอบได้ต่ำกว่า 1.5e-010 เมื่อพิจารณาถึงเวลาที่ใช้ในการค้นหาคำตอบ พบว่า วิธี ATS ที่อาศัยค่าเริ่มต้นจากการสุ่มใช้เวลาเร็วที่สุด คือ 0.68 วินาที ซึ่งมีค่าเร็วกว่าวิธี ATS ที่อาศัยค่าเริ่มต้นจากการกำหนด ส่วนวิธี TS ที่อาศัยค่าเริ่มต้นจากการกำหนด ใช้เวลาในการค้นหาคำตอบนานที่สุด คือ 42.75 วินาที

กรณีที่ 3 : Stop Criterion = 0.0 (global)

สำหรับค่า Stop Criterion = 0.0 นี้เป็นการวัดความสามารถในการค้นหาคำตอบที่เป็นวงกว้าง โดยพารามิเตอร์ที่ปรับแต่งวิธี GA และ TS ต้องรักษาไว้ให้คงเดิม เหมือนกรณี stop criterion อื่น ๆ ซึ่งในกรณีนี้วิธี ATS สามารถทำการค้นหาคำตอบได้เพียงวิธีเดียวภายในจำนวนรอบการค้นหาจำกัด ส่วนวิธี TS และ GA ยุติการดำเนินการด้วยจำนวนรอบการค้นหาสูงสุด ก่อนที่จะค้นพบคำตอบ เมื่อเปรียบเทียบทางด้านเวลาที่ใช้ในการค้นหาคำตอบ วิธี ATS ที่อาศัยค่าเริ่มต้นจากการสุ่มใช้เวลาในการค้นหาคำตอบได้เร็วกว่าวิธี ATS ที่อาศัยค่าเริ่มต้นจากการกำหนด (1.43 > 1.57)

สรุป

จากผลการทดสอบอัลกอริทึม โดยอาศัยฟังก์ชัน Bahachevsky สำหรับการค้นหาคำตอบเหมาะที่สุด พบว่าวิธี ATS ที่ได้พัฒนาขึ้น โดยเพิ่มการทำงานในส่วน back_tracking และ adaptive radius ให้กับวิธี

Table 1. Performance comparison of various search methods.

Search	Counts				Search time (sec.)				Stop	Remarks
method	Max	Min	Mean	SD	Max	Min	Mean	SD	Criterion (≤)	
GA	38	14	27.07	6.98	0.76	0.33	0.54	0.11		random starting point
TS	87	18	59.80	23.16	0.61	0.17	0.42	0.16		
ATS	89	5	51.07	22.81	0.65	0.06	**0.39**	0.16	1.5e-008	
TS	141	32	66.53	34.29	1.16	0.27	0.52	0.28		fixed starting point x = - 0.027, y = 0.008
ATS	102	31	51.27	20.04	0.82	0.22	0.41	0.17		
GA	inf*	inf	inf	inf	inf	inf	inf	inf		random starting point
TS	7153	1012	3578.4	1939.4	49.55	7.08	26.33	13.66		
ATS	152	28	88.27	29.37	1.09	0.22	**0.68**	0.22	15e-101	
TS	13118	1400	5609.3	3645.8	90.96	9.72	42.75	27.26		fixed starting point x = - 0.027, y = 0.008
ATS	187	47	97.80	40.87	1.59	0.33	0.77	0.36		
GA	inf	inf	inf	inf	inf	inf	inf	inf		fixed starting point
TS	inf	inf	inf	inf	inf	inf	inf	inf		point
ATS	396	109	193.40	82.44	2.86	0.82	**1.43**	0.61	0.0	
TS	inf	inf	inf	inf	inf	inf	inf	inf		fixed starting point x = - 0.027, y = 0.008
ATS	369	71	212.47	85.25	2.69	0.60	1.57	0.60		

*inf = search stops with maximum count.

TS แบบธรรมดานั้น ให้สมรรถนะในการค้นหาคำตอบได้ดีกว่าวิธี TS และ GA ทั้งในด้านความสามารถในการค้นหาคำตอบที่เป็นวงกว้างและความเร็วในการค้นหาคำตอบ และเมื่อพิจารณาเปรียบเทียบผลระหว่างวิธี ATS ที่อาศัยค่าเริ่มต้นจากการสุ่มกับที่อาศัยค่าเริ่มต้นจากการกำหนด พบว่า วิธีการทั้งสองมีความสามารถในการค้นหาคำตอบที่เป็นวงกว้างทั้งคู่ เพียงแต่วิธีการ ATS ที่อาศัยค่าเริ่มต้นจากการสุ่มจะใช้เวลาในการค้นหาคำตอบได้เร็วกว่า เมื่อทำการทดสอบสมรรถนะของวิธีการค้นหาเรียบร้อยแล้ว ผลที่ได้จากการทดสอบเป็นสิ่งยืนยันว่า วิธี ATS นั้นเหมาะสมต่อการนำไปใช้กับปัญหาจริงในการหาค่าเหมาะที่สุด

เอกสารอ้างอิง

กองพัน อารีรักษ์. (2544). การระบุเอกลักษณ์ไม่เป็นเชิงเส้น และการตรวจสอบเสถียรภาพของระบบสองมวลความเฉื่อยที่ปรากฏการกำทอนเชิงกล. วิทยานิพนธ์ ปริญญาโท. มหาวิทยาลัยเทคโนโลยีสุรนารี, นครราชสีมา. 101น.

Aldridge, C. J., *et al.* (2001). Knowledge-based genetic algorithm for unit commitment. Proc. IEE Generation, Transmission, and Distribution. 148(2):146-152.

Bland, J. A. and Dawson, G. P. (1991). Tabu search and design optimization. IEEE Trans. Industrial Electronics. 23(3): 195-201.

David, B. F. (1995). Evolutionary Computation. IEEE Press., Inc., New York.

Da Silva, E. L., Areiza Ortiz, J. M., De Oliveira, G. C., and Binato, S. (2001). Transmission network expansion planning under a tabu search approach. IEEE Trans. Power Systems. 16(1):62-68.

David, G. B. (1989). Genetic Algorithms. Addison-Wesley Longman, Inc., Harlow.

Huang, S. J. (2001). Static security assessment of a power system using query-based learning approaches with genetic enhancement. Proc. IEE Generation, Transmission, and Distribution. 148 (4):319-325.

Mantawy, A. H., Abdel-Magid, Y. L. and Selim, S. Z. (1998). Unit commitment by tabu search. Proc. IEEE Generation, Transmission, and Distribution. 145(1): 56-64.

Nara, K., Hayashi, Y., Ikeda, K., and Ashizawa, T. (2001). Application of tabu search to optimal placement of distributed generators. IEEE Conf. Power Engineering Society Winter Meeting.

(pp. 928-923).

Pham, D. T. and Karaboga, D. (2000). Intelligent Optimisation Techniques. Springer., New York.

Qi-Wen, Y., Feng, L., Guo-Hong, Z., and Jing-Ping, J. (2000). The application of computational intelligence to fourier transformation. IEEE Conf. Electronics, Circuits and Systems 2000. (pp. 178-181).

Sujitjorn, S. and Kulworawanichpong, T. (2001). Opimal power flow using tabu search. IEEE Power Engineering Review. (in print)

Van Rensburg, P. J.,Shaw, I. S., and Van Wyk, J. D. (1998). Adaptive PID-control using a genetic algorithm. IEEE Conf. Knowledge-Based Intelligent Electronic Systems. Adelaide, Australia. (pp. 133-138).

Wang, P. and Kwok, D. P. (1992). Auto-tuning of classical PID controllers using an advanced genetic algorithm. Proc. IEEE Conf. Power Electronics and Motion Control. (pp. 1224-1229).

ภาคผนวก

fitness function สำหรับ GA :

$$fitness\ function = \frac{1}{min[f(x,y)]}$$

cost function สำหรับ GA TS และ ATS :

$$cost\ function = min[f(x,y)]$$