การใช้วิธีหาค่าเหมาะสมที่สุดแบบกลุ่มอนุภาคจัดการจราจรข้อมูลในเครือข่ายโอเอสพีเอฟ

Particle Swarm Optimization for Open Shortest Path First Network's Traffic Engineering

กายรัฐ เจริญราษฎร์ (Kairat Jaroenrat)*

บทคัดย่อ

ในปัจจุบันระบบเครือข่ายคอมพิวเตอร์มีบทบาทสำคัญมาก เพราะมีความต้องการรับส่งข้อมูลระหว่างเครื่องคอมพิวเตอร์อย่างแพร่หลาย ซึ่งโพรโทคอลชี้เส้นทางข้ามเครือข่ายในปัจจุบันที่นิยมใช้กันมากที่สุดใน คือ Open Shortest Path First (OSPF) เนื่องจากการที่เครือข่ายที่ใช้โพรโทคอลชี้เส้นทางโอเอสพีเอฟ หรือเรียกสั้นๆ ว่าเครือข่ายโอเอสพีเอฟนี้ มีการใช้อัลกอริทึมในการค้นหาเส้นทางด้วยตัวเอง จึงอาจทำให้การจัดเส้นทางการไหลของข้อมูลไม่ตรงกับความต้องการในการใช้งานเครือข่ายนั้นๆ ดังนั้นเพื่อเพิ่มความสามารถของระบบเครือข่ายให้สูงขึ้น และลดต้นทุนโดยรวมลง ผู้วิจัยจึงประยุกต์ใช้หลักการความฉลาดแบบกลุ่มด้วยวิธีแบบกลุ่มอนุภาค ซึ่งอาศัยรูปแบบวิธีเรียนรู้การหาอาหารของนกในธรรมชาติ เพื่อทดสอบหาค่าน้ำหนักของเส้นทางในเครือข่ายที่ต้องการ ซึ่งการแก้ไขปัญหาด้วยการหาค่าน้ำหนักช่องสัญญาณที่เหมาะสมด้วยวิธีการหาค่าเหมาะสมที่สุดแบบกลุ่มอนุภาคนี้ มีประสิทธิภาพเชิงสมรรถนะที่ดีกว่าการกำหนดค่าน้ำหนักแบบดั้งเดิมโดยเฉลี่ยที่ 1.24 เท่า และเมื่อเปรียบเทียบกับการหาคำตอบที่ดีที่สุดด้วยวิธีโปรแกรมเชิงเส้นพบว่า วิธีกลุ่มอนุภาคนี้ให้ประสิทธิภาพเชิงสมรรถนะได้ใกล้เคียงวิธีโปรแกรมเชิงเส้นที่ 94 % แต่มีประสิทธิภาพเชิงเวลาหรือเวลาในการประมวลผลที่ดีกว่าถึง 25.69 เท่า

คำสำคัญ: ความฉลาดแบบกลุ่ม การหาค่าเหมาะสมด้วยวิธีแบบกลุ่มอนุภาค เครือข่ายโอเอสพีเอฟ

Abstract

Nowadays, computer networks have more important roles and data exchange in networks has been widely used. The most famous Routing Protocols in the system is Open Shortest Path First (OSPF). Because the OSPF routers use Shortest-Path algorithm to find its routes, routing of traffic flow does not get together with the traffic requirement of such system. Therefore, in order to increase efficiency and reduce cost of the system. Applying The Particle Swarm Optimization (PSO) in swarm intelligence to solve this problem. Inspired by the schooling patterns of the way that birds forage determine weight of link in the appropriate network. The solution to find the appropriate links' weight by particle swarm optimization has the better performance than traditional weight set at 1.24 times in average. When Compared to the linear programming method, the performance of the Particle Swarm Optimization is 94% of linear programming method. However, the processing time of the Particle Swarm Optimization is 25.69 times better.

Keyword: Swarm Intelligence, Particle Swarm Optimization (PSO), Open Shortest Path First (OSPF).

1. บทนำ

ในปัจจุบันมีการใช้งานระบบคอมพิวเตอร์กันอย่างแพร่หลาย จึงเกิดความต้องการที่จะเชื่อมต่อคอมพิวเตอร์เหล่านั้นถึงกัน เพื่อเพิ่มความสามารถของระบบให้มี

* ภาควิชาวิศวกรรมคอมพิวเตอร์ คณะวิศวกรรมศาสตร์ กำแพงแสน มหาวิทยาลัยเกษตรศาสตร์

ประสิทธิภาพสูงขึ้นและลดต้นทุนของระบบโดยรวมลง ซึ่งโพรโทคอลชี้เส้นทางที่นิยมใช้กันอย่างแพร่หลายในปัจจุบันมากที่สุด คือ Open Shortest Path First (OSPF) [1] เป็นโพรโทคอลชี้เส้นทางตัวหนึ่งนิยมใช้กันอย่างแพร่หลายมากที่สุดในระบบเครือข่าย เนื่องจากมีจุดเด่นในหลายด้าน ทั้งการที่เป็นโพรโทคอลชี้เส้นทางแบบ Link State และการที่มีอัลกอริทึมในการค้นหาเส้นทางได้ด้วยตัวเอง ซึ่งอุปกรณ์จัดเส้นทาง (Router) ที่สื่อสารด้วยโพรโทคอลโอเอสพีเอฟทุกตัวจะเป็นรูท (Root) หรือจุดเริ่มต้นของระบบไปยังกิ่งย่อยๆ หรือโหนด (Node) ต่างๆ ซึ่งเป็นเทคนิคในการลดเส้นทางที่วนลูปของการค้นหาเส้นทางได้เป็นอย่างดี รวมถึงการเปลี่ยนแปลงเส้นทางในระบบเครือข่ายได้อย่างรวดเร็วเมื่อมีการเปลี่ยนแปลงเกิดขึ้นในระบบเครือข่าย และยังมีความสามารถในการรองรับการขยายของระบบได้เป็นอย่างดี จากข้อดีดังกล่าวทำให้ระบบเครือข่ายต่างๆ นิยมเลือกใช้ OSPF เป็นโพรโทคอลชี้เส้นทางแทนที่โพรโทคอลแบบ Distance Vector เช่น Routing Information Protocol (RIP) [2] Interior Gateway Routing Protocol (IGRP) [2] เป็นต้น ซึ่งในการคำนวณเส้นทางของโอเอสพีเอฟจะใช้ Dijkstra's Algorithm [3-4] ในการคำนวณเส้นทางที่สั้นที่สุด โดยพิจารณาจากเส้นทางที่มีค่าน้ำหนักรวมน้อยที่สุด ซึ่งหากกำหนดค่าน้ำหนักของช่องสัญญาณไว้ไม่เหมาะสมอาจจะทำให้เส้นทางการไหลของข้อมูลอาจไม่ตรงกับความต้องการในการใช้งานเครือข่ายนั้นๆ

การจัดเส้นทางให้แก่การไหลของข้อมูลโดยประยุกต์ใช้ความฉลาดแบบกลุ่ม (Swarm Intelligence) [5] ที่เป็นศาสตร์แขนงหรือกระบวนการหนึ่งที่ถูกรวมเข้าไปในปัญญาประดิษฐ์และเป็นแขนงหนึ่งในสาขา Biologically-Inspired Computing [6] ซึ่งศาสตร์แขนงนี้เป็นการเลียนแบบวิธีการทางธรรมชาติ เหมือนในกรณีของโครงข่ายประสาทเทียม (Artificial Neural Networks) [7] และกระบวนการทางพันธุกรรม (Genetic Algorithm) ซึ่งณัฐกุล และ กายรัฐ [8] ได้ทำการศึกษานี้ โดยนำไปตั้งค่าน้ำหนักให้กับเครือข่ายโอเอสพีเอฟ และยังมีกระบวนการของสิ่งมีชีวิตที่ดำรงชีวิตเป็นฝูงหรือกลุ่ม คือ ระบบอาณาจักรมด (Ant Colony System) ซึ่งเอกวุฒิ และ กายรัฐ [9] ได้ศึกษาหลักการทิ้งสารเคมีเพื่อหาอาหารของมด และนำมาทดสอบใช้กับการตั้งค่าน้ำหนักให้กับเครือข่ายโอเอสพีเอฟเช่นกัน โดยวิธีการหาค่าที่เหมาะสมแบบกลุ่มอนุภาค (Particle Swarm Optimization) [11-12] ก็ได้รับหลักการจากการหาอาหารของฝูงนก หรือฝูงปลา นอกจากนี้ยังมีการทำให้เหมาะสมโดยการผสมพันธุ์ของผึ้ง (Marriage in honey-bees Optimization) [10] ซึ่งเลียนแบบระบบการหาอาหารของผึ้ง จะสังเกตได้ว่าวิธีเหล่านี้อาศัยหลักการที่สิ่งมีชีวิตใช้การร่วมมือแทนที่จะแข่งขันกัน และมีการสื่อสารภายในกลุ่ม (Social Behavior) เพื่อแก้ไขปัญหาของมันในธรรมชาติ มาประยุกต์ในการแก้ไขปัญหาที่ซับซ้อนทำให้ประสิทธิภาพในการแก้ปัญหาและประสิทธิผลของคำตอบมีมากขึ้น ยิ่งไปกว่านั้นยังมีการใช้วิธีการโปรแกรมเชิงเส้น ซึ่ง กฤษณะ และกายรัฐ [16] ได้นำมาประยุกต์ใช้ในการหาค่าน้ำหนักที่เหมาะสมที่สุดของช่องสัญญาณในเครือข่ายโอเอสพีเอฟ ซึ่งเป็นวิธีที่ให้ประสิทธิภาพของเครือข่ายที่ดีมาก แต่ใช้เวลาในการประมวลผลนานมากเช่นกัน

เพื่อให้การจัดเส้นทางข้อมูลของเครือข่ายสามารถทำงานได้อย่างมีประสิทธิภาพ ผู้วิจัยจึงมีแนวคิดพัฒนาซอฟต์แวร์ที่ประยุกต์ใช้หลักการวิธีแบบกลุ่มอนุภาค (Particle Swarm Optimization) [11] [13] ที่ศึกษาพฤติกรรมทางสังคมของฝูงนก โดยการบินของนกหากมองเป็นพารามิเตอร์จะประกอบด้วยตำแหน่งที่นกบินอยู่ (Position) และความเร็วในการเคลื่อนที่ของนก (Velocity) ซึ่งนกแต่ละตัวทำหน้าที่บินค้นหาอาหารและอาจย้ายที่อยู่ไปด้วยพร้อมๆ กัน จุดใดที่มีอาหารอยู่ จุดใดสามารถพักอาศัยได้หรือเป็นจุดที่มีอันตราย นกจะมีการส่งสัญญาณเพื่อสื่อสารกันในกลุ่มเพื่อแลกเปลี่ยนข้อมูลระหว่างกันเพื่อให้ทราบถึงตำแหน่งที่มีอาหารอยู่และทำการเคลื่อนที่ไปยังแหล่งอาหารที่ได้รับข้อมูลมา ซึ่งเป็นวิธีการค้นหาคำตอบด้วยการใช้อนุภาค (Particles) จำนวนมากเคลื่อนที่ไปบนพื้นที่ที่ต้องการค้นหาเพื่อค้นหาคำตอบที่ดีที่สุด ซึ่งสามารถนำไปใช้ในการกำหนดค่าน้ำหนักให้แก่เครือข่ายที่ต้องการ และนำค่าน้ำหนักนั้นไปทดสอบเชิงสมรรถนะ วิเคราะห์การไหลของข้อมูลภายในเครือข่ายเพื่อหาชุดของคำตอบที่มีค่าน้ำหนักเหมาะสมที่สุด

2. ทฤษฎีที่เกี่ยวข้อง

ในการวิจัยเรื่องการหาค่าเหมาะสมด้วยวิธีแบบกลุ่มอนุภาคในการตั้งค่าน้ำหนักให้แก่เครือข่ายที่ใช้โพรโทคอลโอเอสพีเอฟในการชี้เส้นทาง หรือเรียกสั้นๆ ว่าเครือข่ายโอเอสพีเอฟ ผู้วิจัยได้ศึกษาหลักการของทฤษฎีและ

เทคโนโลยีต่างๆ ที่เกี่ยวข้องกับการพัฒนาระบบที่สามารถนำมาประยุกต์ใช้กับงานได้โดยแบ่งออกเป็นหัวข้อต่างๆ ดังต่อไปนี้

2.1 Open Shortest Path First

โอเอสพีเอฟเป็นโพรโทคอลชี้เส้นทางตัวหนึ่งนิยมใช้กันอย่างแพร่หลายมากที่สุดในระบบเครือข่าย เนื่องจากมีจุดเด่นในหลายด้าน เช่น การที่เป็นโพรโทคอลที่เรียนรู้โทโปโลยีของเครือข่ายในการค้นหาเส้นทางที่สั้นที่สุด หรือที่เรียกว่าโพรโทคอลชนิด Link State และการที่มีอัลกอริทึมในการค้นหาเส้นทางได้ด้วยตัวเอง ซึ่งในการคำนวณเส้นทางของโอเอสพีเอฟนี้จะใช้อัลกอริทึมของ Dijkstra ในการคำนวณเส้นทางที่สั้นที่สุด [1] ส่วนค่าแบบดั้งเดิม (Default Weight) คำนวณได้จากสมการที่ (1)

$$cost = \frac{10^8}{Bandwidth} \quad (1)$$

เมื่อ cost คือ ค่าน้ำหนักของสายสัญญาณ

Bandwidth คือ ค่าอัตราการรองรับการไหลของข้อมูล

2.2 การหาค่าเหมาะสมด้วยวิธีแบบกลุ่มอนุภาค

ในปี 1995 J. Kenndy and R. Eberthart [11] ได้นำเสนอการหาค่าที่เหมาะสมที่สุดด้วยวิธีแบบกลุ่มอนุภาค คือ การจำลองพฤติกรรมทางสังคมของฝูงนก มาเป็นขั้นตอนวิธีในการค้นหาโดยอาศัยความน่าจะเป็น โดยการจำลองพฤติกรรมของการออกไปหาอาหารของนก โดยนกในฝูงจะช่วยกันออกไปหาอาหาร ซึ่งแต่ละตัวจะไม่รู้ล่วงหน้าถึงสถานที่ที่มีอาหารอยู่ แต่ละครั้งของการบินออกไปหาอาหารนกจะเรียนรู้ด้วยการติดต่อสื่อสารกันภายในฝูงถึงระยะทางในการออกไปหาอาหาร ดังนั้นเมื่อตัวใดตัวหนึ่งพบอาหารจะเกิดการเปรียบเทียบเพื่อไปยังแหล่งอาหารที่อยู่ใกล้ที่สุดที่ค้นพบ จากพฤติกรรมดังกล่าวของฝูงนก นกแต่ละตัวจะถูกเรียกใหม่ด้วยคำว่า อนุภาค (Particle) ทุกๆ อนุภาคจะมี Fitness Values ซึ่งจะถูกประเมินด้วย Objective Function

การบินของนกหากมองเป็นพารามิเตอร์จะประกอบด้วยตำแหน่งนกบินอยู่ (Position) โดยนำมาแปลงเป็นค่าน้ำหนักที่ใช้ตั้งค่าให้กับเครือข่ายโอเอสพีเอฟและความเร็วของการเคลื่อนที่ (Velocity) ดังนั้นในหนึ่งอนุภาคเมื่อแทนด้วยพารามิเตอร์ของปัญหาหนึ่งจะประกอบด้วยตัวแปร ดังสมการที่ (2)

$$P_k = \{X, V, Fitness\ value, Gbest, Pbest\} \quad (2)$$

โดยที่ P คือ อนุภาคประกอบด้วย $P = \{P_1, P_2, ..., P_k\}$

K คือ จำนวนอนุภาค

X คือ ค่าน้ำหนักของสายสัญญาณ

$X_D = \{X_1, X_2, ..., X_D\}$

D คือ จำนวนของสายสัญญาณทั้งหมด

V คือ ค่าความเร็วในการเคลื่อนที่ของอนุภาค

$V_D = \{V_1, V_2, ..., V_D\}$

Fitness Value คือ ค่าความเหมาะสมของอนุภาคนั้นสามารถคำนวณได้จากฟังก์ชันเป้าหมายซึ่งได้จาก Fitness Value = Objective Function (X)

$Gbest_D$ คือ ค่าน้ำหนักของสายสัญญาณที่อนุภาคนั้นได้รับค่าความเหมาะสมสูงสุด โดย $Gbest_D = \{Gbest_1, Gbest_2, ..., Gbest_D\}$

$Pbest_D$ คือ ค่าความเหมาะสมที่ดีที่สุดที่กลุ่มอนุภาครุ่นปัจจุบันได้มา

การหาค่าที่เหมาะสมด้วยวิธีแบบกลุ่มอนุภาคสามารถแบ่งเป็น 7 ขั้นตอนดังนี้

2.2.1 Initial particle เป็นการสร้างประชากรเริ่มต้น *(X)* และค่าความเร็วการเคลื่อนที่ *(V)* ของอนุภาคแต่ละตัวโดยใช้การสุ่มค่า ดังสมการที่ (3) และ (4) โดยสุ่มค่าพารามิเตอร์ให้ตัวแปรภายของ X และ V ตามจำนวนของสายสัญญาณ

$$X_D = function\ random\ value(X_D) \quad (3)$$

$$V_D = function\ random\ value(V_D) \quad (4)$$

2.2.2 Evaluate fitness value เป็นการคำนวณค่าความเหมาะสมของอนุภาคด้วยฟังก์ชันความเหมาะสม ดังสมการที่ (5) ด้วยการส่งค่า X ให้แก่ฟังก์ชันเป้าหมาย

$$Fitness\ Value = Objective\ function(X_D) \quad (5)$$

2.2.3 Update *Pbest* เป็นขั้นตอนการเก็บค่าความเหมาะสมที่ดีสุดในรอบการทำงานปัจจุบัน โดยมีเงื่อนไขการปรับปรุงตามสมการที่ (6) และถ้าค่าความเหมาะสมของอนุภาคปัจจุบันมีค่าดีกว่า *Pbest* จะทำการปรับปรุงค่า *Pbest*

IF *fitness value* < *Pbest* THEN
Pbest = *fitness value* ENDIF (6)

2.2.4 Update *Gbest* เป็นการเก็บค่าน้ำหนักที่ให้ผลลัพธ์ดีที่สุด โดยพิจารณาจากการวนรอบตั้งแต่รอบแรกถึงปัจจุบันซึ่งแตกต่างจาก *Pbest* ที่พิจารณาการเก็บค่าที่ดีสุดเฉพาะรอบปัจจุบัน ดังนั้น *Gbest* จึงเหมือนการเก็บค่าความเหมาะสมที่ดีสุดตั้งแต่เริ่มต้นคำนวณ โดยมีเงื่อนไขการปรับปรุงข้อมูลดังสมการที่ (7)

IF *fitness value* < *Gbest* THEN
$Gbest = X_D$ ENDIF (7)

2.2.5 Update velocity เป็นการคำนวณความเร็วการเคลื่อนที่ของแต่ละอนุภาค ดังสมการที่ (8)

$$V'_D = Inertia\ weight \times V_D + c_1 \times random_1() \times (Pbest_D - X_D) + c_2 \times random_2() \times (Gbest_D - X_D)$$

$$V'_D = \begin{cases} -V_{\max} if\ V'_D \le -V_{\max} \\ V_{\max}\ if\ V'_D \ge V_{\max} \end{cases} \quad (8)$$

โดยที่ V'_D คือ ค่าความเร็วที่คำนวณได้จากอนุภาครุ่นปัจจุบัน ส่วน V_D คือ ค่าความเร็วที่คำนวณได้จากอนุภาครุ่นก่อน

Inertia Weight คือ ค่าถ่วงน้ำหนัก มีค่าอยู่ในช่วง [0.40-0.95]

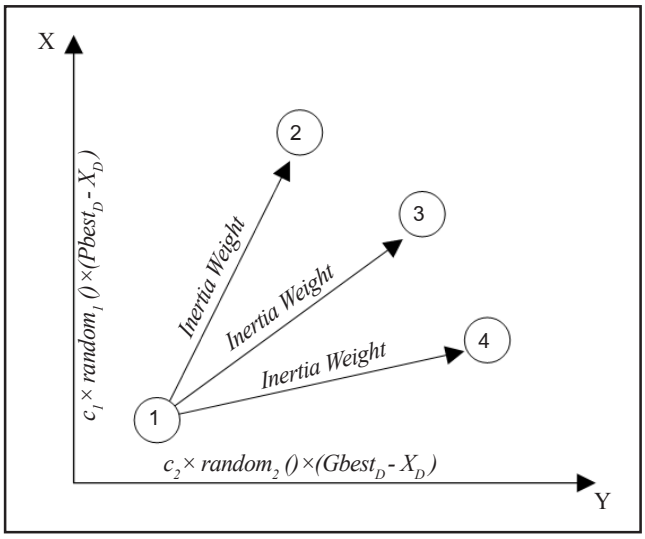
c_1 และ c_2 คือ ค่าแฟคเตอร์การเรียนรู้ (ค่าถ่วงน้ำหนักความจำ) มีค่าอยู่ในช่วง [0,2]

2.2.6 Update weight เป็นการปรับปรุงค่าน้ำหนักของแต่ละอนุภาค โดยใช้ผลลัพธ์จากการคำนวณในสมการที่ (7) ไปปรับปรุงค่าตามสมการที่ (8) ซึ่งการปรับปรุงด้วยการบวกค่าเพิ่มนั้นจำเป็นต้องตรวจสอบขอบเขตของปัญหาป้องกันไม่ให้ตำแหน่งเกินค่าขอบเขตของปัญหา

$$X'_D = X_D + V'_D \quad (9)$$

2.2.7 Terminate Check การตรวจสอบการสิ้นสุดการวนรอบตามเงื่อนไข เมื่อผลลัพธ์ที่ได้มีค่าต่ำสุดและดีกว่าคำตอบที่ต้องการให้หยุดทำงาน

การปรับค่าน้ำหนักของอนุภาคตามการคำนวณในขั้นตอนที่ 2.2.5 ด้วยสมการที่ (8) สามารถนำมาพิจารณาเขียนเป็นแผนภูมิแสดงการปรับค่าน้ำหนักได้ดังภาพที่ 1 จากสมการค่าถ่วงน้ำหนัก (Inertia Weight) และค่าแฟคเตอร์การเรียนรู้ที่ใช้กำหนดความเร็วในการปรับค่าน้ำหนัก สมการส่วนที่มี *Pbest* จะใช้สำหรับคำนวณทิศทางตามแนว



ภาพที่ 1 *การคำนวณการปรับค่าน้ำหนักของอนุภาค*

แกน y และสมการส่วนที่มี *Gbest* จะใช้คำนวณทิศทางตามแนวแกน x เมื่อนำสมการทั้งสามส่วนรวมกัน โดยสมมุติให้อนุภาคมีค่าน้ำหนักเป็น 1 หากพบว่าส่วนของ Pbest มีค่าที่ได้จากการสุ่มมากและเป็นทิศทางที่ได้ผลลัพธ์ดี หรือฟังก์ชันต้นทุนมีค่าลดลง จะทำให้อนุภาคทำการปรับน้ำหนักจาก 1 เป็น 2 ส่วนถ้าในกรณีของ Pbest ได้ผลลัพธ์ด้อยกว่าเดิม สมการในส่วนของ Gbest จะดึงอนุภาคให้ปรับน้ำหนักจาก 1 เป็น 4 จากปัญหาที่สมมุตินี้เปรียบเทียบได้กับการที่อนุภาคปรับค่าน้ำหนักจาก 1 เป็น 2 แสดงให้เห็นว่าอนุภาคตัวนี้พบฟังก์ชันต้นทุนที่ลดลงในค่าน้ำหนักที่ 2 และฟังก์ชันต้นทุนมีค่าน้อยกว่าที่นกในกลุ่มบอกกัน แต่หากอนุภาคปรับค่าน้ำหนักจาก 1 เป็น 4 แสดงให้เห็นว่าอนุภาคนี้ปรับเปลี่ยนค่าน้ำหนักจากที่อนุภาคในรุ่นชี้ไปยังค่าน้ำหนักเดิมเพราะค่าน้ำหนักใหม่มีฟังก์ชันต้นทุนมากทุนมากขึ้น

2.3 การโปรแกรมเชิงเส้น

การโปรแกรมเชิงเส้น (Linear Programming) [14] เป็นเทคนิคที่รู้จักกันแพร่หลายและเป็นส่วนหนึ่งของการวิจัยดำเนินงานในหลายๆ ด้าน ซึ่งถูกใช้ในการแก้ปัญหาทางการจัดสรรปัจจัยหรือทรัพยากร โดยที่ปัจจัยหรือทรัพยากรมีความหมายรวมถึงวัตถุดิบ กำลังคน เวลา สถานที่ เงินทุน หรือความรู้ความสามารถต่างๆ ปัญหาการจัดสรรปัจจัยและทรัพยากรเกิดขึ้นเมื่อเราต้องการจัดสรรทรัพยากรที่มีอยู่จำกัดทั้งขนาด ปริมาณ และขอบเขตของการใช้งาน เพื่อให้เกิดประโยชน์สูงสุด

การโปรแกรมเชิงเส้นประกอบไปด้วย 2 ส่วน คือ ส่วน

ของสมการกำหนดเป้าหมาย (Objective Function) ซึ่งเป็นสมการแสดงความสัมพันธ์ของต้นทุน กำไร เพื่อให้กำหนดเป้าหมายสูงสุดหรือต่ำสุด และส่วนที่สองคือสมการแสดงขอบข่าย (Constraints) ซึ่งแสดงข้อจำกัดต่าง ๆ ของปัจจัยหรือทรัพยากรในรูปสมการหรืออสมการ โดยที่สมการต่าง ๆ ทั้งหมดเป็นสมการเชิงเส้น

2.4 Objective Function

Bernard Fortz และ Mikkel Thorup [15] ได้พิจารณากราฟของระบบโครงข่าย $G=(N, A)$ ซึ่งมีเซ็ตของโหนดเป็น N และเซ็ตของเส้นเป็น A โดยกำหนดอัตราการไหลหรือความกว้างของช่องสายสัญญาณ (C_a) โดยที่ $a \in A$ โดยกำหนด d_{st} แทน *traffic flow* ระหว่าง s และ t และให้ f_a^{st} แสดงปริมาณ *traffic flow* จาก s ไป t บนช่องสัญญาณ a จึงสามารถคำนวณปริมาณ *traffic flow* l_a บนสายสัญญาณ $a \in A$ ซึ่งเป็นผลรวมของ f_a^{st} จากสมการที่ (10)

$$l_a = \sum_{(s,t)\in N\times N} f_a^{st} \tag{10}$$

และกำหนดวิธีการวัดประสิทธิภาพของระบบโครงข่ายไว้โดยใช้ฟังก์ชันต้นทุน ดังสมการที่ (11)

$$Minimize\Phi = \sum_{a\in A}\left(\Phi_a\left(l_a, C_a\right)\right) \tag{11}$$

ในขณะที่ $\phi_a\left(l_a, C_a\right)$ อิงกับ Delay ในตัวแบบ M/M/1 ในทฤษฎี Queuing ซึ่งได้จาก

$$\Phi_a\left(l_a, C_a\right) = \frac{l_a}{\left(C_a - l_a\right)} \tag{12}$$

แต่ต่อมาได้ทำการปรับแก้เป็นสมการที่ (13) ในรูปแบบ Piece-wise Linear เพื่อรองรับกรณีที่ $l_a = C_a$

$$\Phi_a\left(l_a, C_a\right) = Max\left\{\begin{array}{c} \Phi \geq l_a \\ \Phi_a \geq 3l_a - \frac{2}{3}c_a \\ \Phi_a \geq 10l_a - \frac{16}{3}c_a \\ \Phi_a \geq 70l_a - \frac{178}{3}c_a \\ \Phi_a \geq 500l_a - \frac{1468}{3}c_a \\ \Phi_a \geq 5000l_a - \frac{16318}{3}c_a \end{array}\right\} \tag{13}$$

เมื่อ Φ คือ ค่า Delay Cost ทั้งหมดในเครือข่าย

Φ_a คือ ค่า Delay Cost ที่ใช้ในสายสัญญาณ

l_a คือ ค่าภาระการไหลของสายสัญญาณ a

C_a คือ ค่าความกว้างของช่องสัญญาณ a

ภายใต้ Flow conservation ในสมการที่ (14)

$$\sum_{u(U,v)\in A} f_{u,v}^{s,t} - \sum_{u(U,v)\in A} f_{u,v}^{s,t} = \begin{cases} d_{st}, if\ v = t \\ -d_{st}, if\ v = t \\ 0, otherwise \end{cases} \tag{14}$$

$$f_a^{s,t} \geq 0 : d_{st},\ \ a \in A,\ \ t \in N,\ \ v,s,t, \in N$$

2.5 การวิเคราะห์การไหลของข้อมูล

การวิเคราะห์การไหลของข้อมูลเป็นการนำความต้องการของผู้ใช้ มาวิเคราะห์ร่วมกับข้อมูลตำแหน่งการติดตั้งของอุปกรณ์ในเครือข่าย การไหลของข้อมูลถือเป็นข้อมูลที่มีความสำคัญมากในการออกแบบระบบเครือข่าย เพราะใช้แสดงถึงภาระที่เครือข่ายจะต้องรองรับ ระบุทิศทางของข้อมูลพร้อมกับคุณสมบัติเชิงสมรรถนะของการไหล ในการวิเคราะห์ระบบเพื่อการออกแบบเครือข่ายจะใช้การไหลเป็นตัวแทนความต้องการในการใช้งานเครือข่ายของโปรแกรมหรือการส่งผ่านข้อมูลแต่ละชนิด [14]

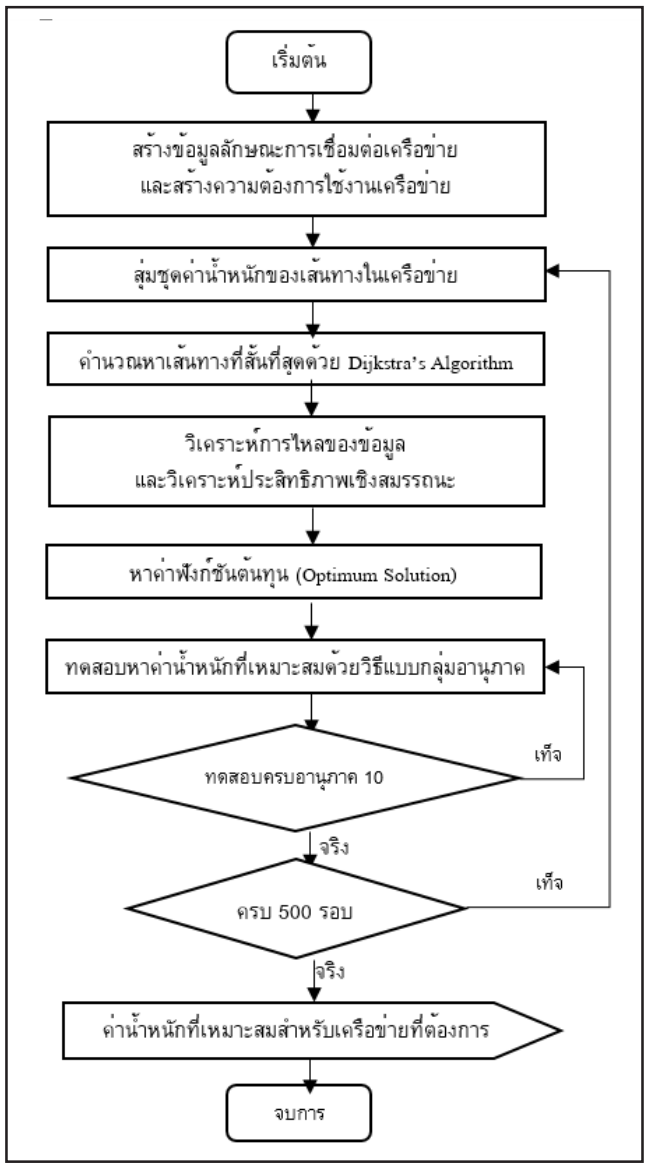
3.วิธีดำเนินการวิจัย

ผู้วิจัยใช้โปรแกรม Microsoft Visual Studio C# ในการพัฒนาระบบจำลองเครือข่าย และประมวลผลบนคอมพิวเตอร์ที่มีหน่วยประมวลผลกลาง Intel Core i5 430M (2.27 GHz) และ หน่วยความจำหลัก (RAM) 2.00 GB โดยแปรผันค่าพารามิเตอร์ของวิธีแบบกลุ่มอนุภาคอันประกอบไปด้วยค่าถ่วงน้ำหนัก *(W)*, ค่าโหนดดีกรีของเครือข่ายที่ทดสอบ และค่าความน่าจะเป็นของค่าแฟคเตอร์การเรียนรู้ *(C1, C2)* ที่ $(0 \leq C_1 \leq 2)$ และ $(0 \leq C_2 \leq 2)$ โดยทำการทดสอบกับเครือข่ายขนาด 10 โหนด, 25 โหนด และ 50 โหนด ที่สร้างโหนด, สายสัญญาณ และความต้องการการไหลข้อมูลขึ้นมาแบบสุ่ม โดยกำหนดให้มีโหนดดีกรี 1.2, 1.4, 1.6, 1.8 และ 2.0 ตามลำดับ

จากภาพที่ 2 แสดงขั้นตอนของการทดลองด้วยการใช้วิธีแบบกลุ่มอนุภาคในการคำนวณค่าน้ำหนักช่องสัญญาณในเครือข่ายโอเอสพีเอฟ โดยเริ่มต้นด้วยการสุ่มชุดค่าน้ำหนักนำไปคำนวณหาเส้นทางที่สั้นที่สุดแล้วนำมาวิเคราะห์

การไหลของข้อมูล วิเคราะห์ประสิทธิภาพเชิงสมรรถนะและเวลา

ขั้นตอนต่อมาเป็นการทดสอบชุดค่าน้ำหนักด้วยวิธีแบบกลุ่มอนุภาค โดยจะทำการทดสอบ 10 ครั้งต่อรอบการทำงาน ซึ่งในการทดสอบแต่ละครั้ง อนุภาคจะทำการปรับความเร็วบนชุดค่าน้ำหนักที่ทำให้ได้ค่าฟังก์ชันต้นทุนที่ต่ำที่สุดที่ได้จากการเคลื่อนที่ของกลุ่มอนุภาค แล้วทำการทดสอบจำนวน 500 รอบการทำงาน เมื่อทำการทดสอบเสร็จสิ้นจะทำการเลือกชุดของค่าน้ำหนักที่ทำให้ได้ค่าฟังก์ชันต้นทุนที่ต่ำที่สุด โดยทำการทดสอบด้วยการปรับพารามิเตอร์ C_1 และ C_2 ดังแสดงในตารางที่ 1 เพื่อทดสอบความสามารถในการคำนวณชุดค่าน้ำหนักของช่องสัญญาณที่โหนดดีกรีต่าง ๆ ของเครือข่าย 10 โหนด 25 โหนด และ 50 โหนด



ภาพที่ 2 *กระบวนการทดลองด้วยวิธีกลุ่มอนุภาค*

ตารางที่ 1 *ชุดของการปรับค่าพารามิเตอร์*

ชุดที่	C_1	C_2	ชุดที่	C_1	C_2
1	0.4	0.4	14	1.6	0.4
2	0.4	0.8	15	1.6	0.8
3	0.4	1.2	16	1.6	1.2
4	0.4	1.6	17	1.6	1.6
5	0.4	2.0	18	1.6	2.0
6	0.8	0.4	19	2.0	0.4
7	0.8	0.8	20	2.0	0.8
8	0.8	1.2	21	2.0	1.2
9	0.8	1.6	22	2.0	1.6
10	0.8	2.0	23	2.0	2.0
11	1.2	0.4	24	1.2	0.4
12	1.2	0.8	25	1.2	0.8
13	1.2	1.2			

นอกจากการปรับพารามิเตอร์ในแต่ละชุดการทดลองตามตารางที่ 1 แล้ว ผู้วิจัยยังได้ปรับค่า Inertia weight ที่ $W = 0.5$ และ $W = 0.7$ จากนั้นจึงนำผลลัพธ์ที่ได้ทั้งหมดมาทำการวิเคราะห์เพื่อหาประสิทธิภาพของค่าพารามิเตอร์ต่าง ๆ

การทดสอบหาชุดค่าน้ำหนักที่เหมาะสม เพื่อให้มีความสามารถในการส่งข้อมูลในเครือข่ายได้อย่างมีประสิทธิภาพ สามารถคำนวณประสิทธิภาพจากสมการที่ (15)

$$\text{ประสิทธิภาพเชิงสมรรถนะ} = \frac{C_{Default} \times 100}{C_{PSO}}\% \qquad (15)$$

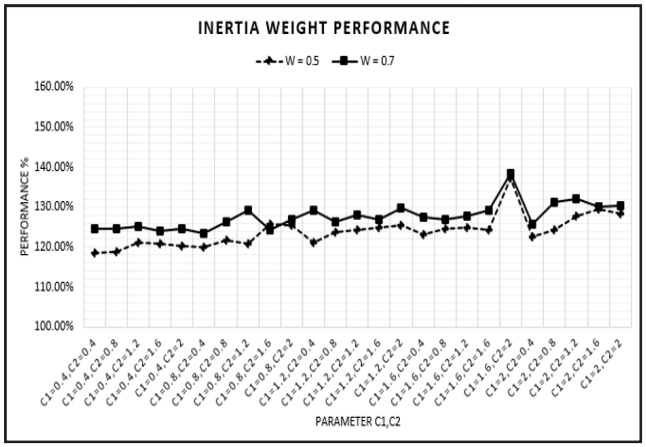
เมื่อ $C_{Default}$ คือ ค่า Delay Cost ในโครงข่ายโดยการกำหนดค่าน้ำหนักแบบดั้งเดิม ตามสมการที่ (1) และ C_{PSO} คือ ค่า Delay Cost ในโครงข่ายโดยการตั้งค่าน้ำหนักที่เหมาะสมด้วยวิธีแบบกลุ่มอนุภาค

4. ผลการดำเนินงาน

4.1 การวิเคราะห์ค่าถ่วงน้ำหนัก W (Interior Weight)

จากผลการทดลองในภาพที่ 3 พบว่าค่าประสิทธิภาพเชิงสมรรถนะของชุดข้อมูลค่าถ่วงน้ำหนักที่ $W=0.5$ และ $W=0.7$ ซึ่งค่าถ่วงน้ำหนักที่ $W = 0.7$ ให้ค่าประสิทธิภาพเชิงสมรรถนะ

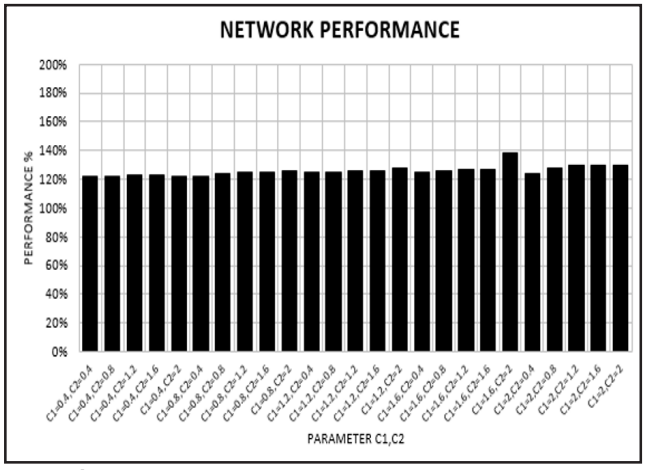
ที่ดีกว่าโดยเฉลี่ยและมีค่าต่างกันเพียงเล็กน้อยที่ส่วนเบี่ยงเบนมาตรฐานเท่ากับ 0.1193



ภาพที่ 3 *แผนภูมิเปรียบเทียบผลของค่าถ่วงน้ำหนักที่ w = 0.5 และ w = 0.7 โดยเฉลี่ยจากเครือข่ายขนาด 10 โหนด 25 โหนด และ 50 โหนด ที่โหนดดีกรี 1.2, 1.4, 1.6, 1.8 และ 2.0*

4.2 การวิเคราะห์ค่าพารามิเตอร์

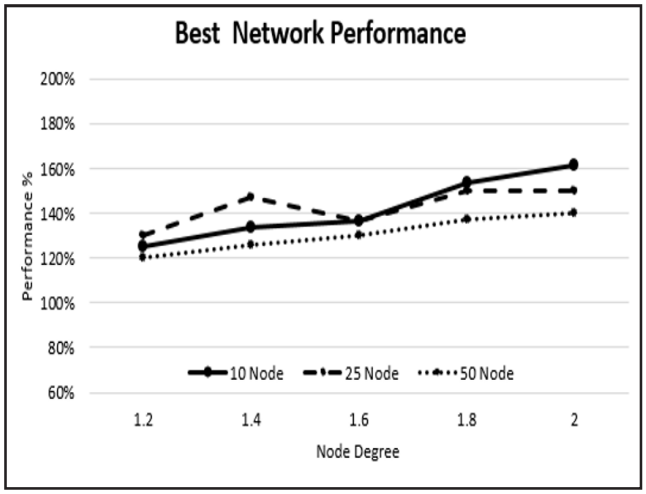
จากผลการทดลองดังแสดงในภาพที่ 4 พบว่าเครือข่ายที่คำนวณค่าน้ำหนักด้วยวิธีกลุ่มอนุภาคให้ค่าประสิทธิภาพที่ดีกว่าเครือข่ายที่กำหนดค่าน้ำหนักด้วยวิธีแบบดั้งเดิมโดยเฉลี่ยอยู่ในช่วงระหว่าง 6.36 ถึง 37.94 เปอร์เซ็นต์ โดยค่าพารามิเตอร์ที่ c_1 = 1.6 และ c_2 = 2 ให้ค่าประสิทธิภาพที่ดีที่สุดในการทดสอบ



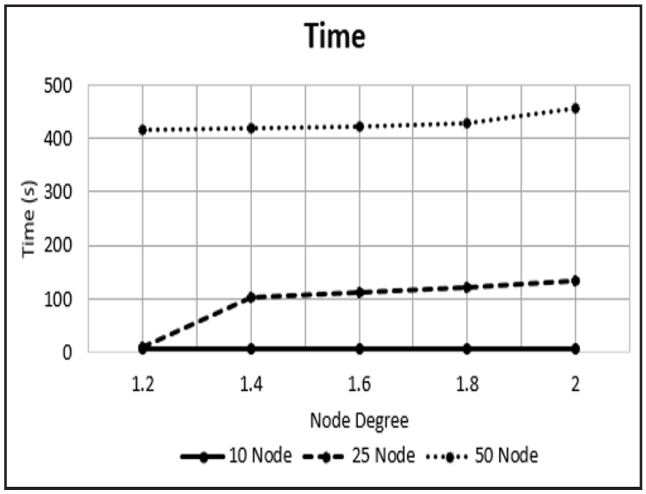
ภาพที่ 4 *แผนภูมิประสิทธิภาพโดยเฉลี่ยเมื่อปรับค่าพารามิเตอร์ C_1 และ C_2 ของเครือข่ายขนาด 10 โหนด 25 โหนด และ 50 โหนด ที่โหนดดีกรี 1.2, 1.4, 1.6, 1.8 และ 2.0*

4.3 การวิเคราะห์ค่าโหนดดีกรี

จากผลการทดลองดังแสดงในภาพที่ 5 พบว่าวิธีการแก้ปัญหาการหาค่าน้ำหนักช่องสัญญาณที่เหมาะสมด้วยวิธีแบบกลุ่มอนุภาค ให้ค่าประสิทธิที่ดีกว่าการกำหนดค่าน้ำหนักด้วยวิธีการดั้งเดิม โดยค่าประสิทธิภาพจะเพิ่มขึ้นเมื่อค่าโหนดดีกรีของเครือข่ายมากขึ้น หรือยิ่งเป็นเครือข่ายที่ซับซ้อนมากขึ้นเท่าไหร่ ยิ่งเหมาะสมสำหรับใช้วิธีแก้ปัญหาแบบกลุ่มอนุภาคนี้เท่านั้น



ภาพที่ 5 *แนวโน้มค่าประสิทธิภาพจากการทดสอบที่โหนดดีกรีต่างๆ ด้วยเครือข่าย 10 โหนด 25 โหนด และ 50 โหนด*



ภาพที่ 6 *เวลาที่ใช้ในการประมวลผลของเครือข่ายขนาดและโหนดดีกรีต่างๆ*

4.4 เวลาในการประมวลผลของวิธีกลุ่มอนุภาค

จากผลการทดลองในภาพที่ 6 แสดงถึงเวลาที่ใช้ในการประมวลผลคำนวณค่าน้ำหนักช่องสัญญาณด้วยวิธีกลุ่มอนุภาค พบว่าเมื่อเครือข่ายมีขนาดใหญ่ขึ้นหรือมีความซับซ้อนมากขึ้นก็จะใช้เวลาประมวลผลเพิ่มขึ้นตามไปด้วย

4.5 การเปรียบเทียบค่าประสิทธิภาพเชิงสมรรถนะและเชิงเวลา ระหว่างวิธีกลุ่มอนุภาคกับวิธีการโปรแกรมเชิงเส้น (Linear Programming)

การเปรียบเทียบประสิทธิภาพเชิงสมรรถนะระหว่างเครือข่ายที่คำนวณค่าน้ำหนักช่องสัญญาณด้วยวิธีกลุ่มอนุภาค กับเครือข่ายที่คำนวณค่าน้ำหนักช่องสัญญาณด้วยวิธีโปรแกรมเชิงเส้น [16] สามารถคำนวณได้จากสมการที่ (16) และการเปรียบเทียบประสิทธิภาพเชิงเวลาสามารถคำนวณได้จากสมการที่ (17)

$$\text{ประสิทธิภาพเชิงสมรรถนะ} = \frac{C_{LP} \times 100}{C_{PSO}}\% \qquad (16)$$

$$\text{ประสิทธิภาพเชิงเวลา} = \frac{T_{LP} \times 100}{T_{PSO}}\% \qquad (17)$$

เมื่อ C_{PSO} และ T_{PSO} คือ ค่า Delay Cost และค่าเวลาเฉลี่ยในโครงข่ายที่คำนวณค่าน้ำหนักช่องสัญญาณด้วยวิธีแบบกลุ่มอนุภาค

ส่วน C_{LP} และ T_{LP} คือ ค่า Delay Cost และค่าเวลาเฉลี่ยในโครงข่ายที่คำนวณค่าน้ำหนักช่องสัญญาณด้วยวิธีการโปรแกรมเชิงเส้น

ผู้วิจัยได้ทดลองเปรียบเทียบประสิทธิภาพของเครือข่ายขนาด 10 โหนด ที่คำนวณค่าน้ำหนักช่องสัญญาณด้วยวิธีกลุ่มอนุภาค กับเครือข่ายเดียวกันที่คำนวณค่าน้ำหนักช่องสัญญาณด้วยวิธีโปรแกรมเชิงเส้น [16] ดังแสดงในตารางที่ 2 พบว่าวิธีแบบกลุ่มอนุภาคนี้ให้ค่าประสิทธิภาพเชิงสมรรถนะโดยเฉลี่ยเท่ากับ 94% ของวิธีโปรแกรมเชิงเส้น และให้ค่าประสิทธิภาพเชิงเวลาดีกว่าวิธีโปรแกรมเชิงเส้นมาก โดยเฉลี่ย 2,569%

จากแผนภูมิผลการเปรียบเทียบค่าประสิทธิภาพเชิงสมรรถนะ (Delay Cost) ของเครือข่ายที่กำหนดค่าน้ำหนักด้วยวิธีแบบกลุ่มอนุภาค ที่ค่าพารามิเตอร์ต่างๆ ในภาพที่ 7 พบว่า ประสิทธิภาพเชิงสมรรถนะของวิธีกลุ่มอนุภาคมีค่าใกล้เคียงวิธีโปรแกรมเชิงเส้น ที่ในช่วง 88 ถึง 102% โดยมีแนวโน้มของสมรรถนะที่เพิ่มขึ้นเล็กน้อยเมื่อเพิ่มค่าพารามิเตอร์ C_1 และ C_2

ส่วนภาพที่ 8 เป็นแผนภูมิการเปรียบเทียบค่าประสิทธิภาพเชิงเวลาของวิธีแบบกลุ่มอนุภาค ซึ่งผู้วิจัยพบว่าวิธีแบบกลุ่มอนุภาคนี้ให้ประสิทธิภาพในด้านเวลาที่ดีกว่าวิธีโปรแกรมเชิงเส้นมาก กล่าวคือสามารถประมวลได้เร็วกว่า ในช่วงตั้งแต่ 25.52 ถึง 25.79 เท่า

ตารางที่ 2 *ค่าประสิทธิภาพเชิงสมรรถนะและเวลาของเครือข่ายขนาด 10 โหนด*

Parameter		Cost Perfoemance PSO and LP	Time Performance PSO and LP
C1	C2		
0.4	0.4	92%	2568%
0.4	0.8	89%	2569%
0.4	1.2	90%	2562%
0.4	1.6	88%	2562%
0.4	2	89%	2561%
0.8	0.4	94%	2576%
0.8	0.8	96%	2572%
0.8	1.2	91%	2578%
0.8	1.6	99%	2571%
0.8	2	94%	2574%
1.2	0.4	97%	2571%
1.2	0.8	95%	2574%
1.2	1.2	94%	2567%
1.2	1.6	94%	2568%
1.2	2	93%	2566%
1.6	0.4	95%	2578%
1.6	0.8	95%	2575%
1.6	1.2	94%	2579%
1.6	1.6	92%	2568%
1.6	2	102%	2564%
2	0.4	96%	2571%
2	0.8	95%	2569%
2	1.2	95%	2552%
2	1.6	95%	2572%
2	2	93%	2555%

5. สรุปผลและข้อเสนอแนะ

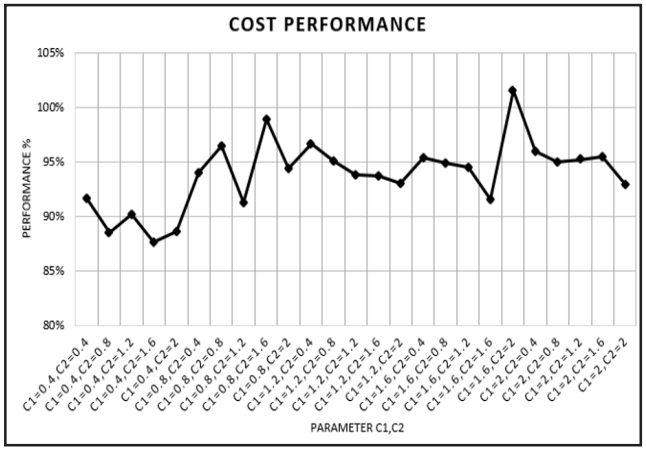
5.1 สรุปผลการดำเนินงาน

การประยุกต์ใช้วิธีกลุ่มอนุภาคในการคำนวณหาชุดค่าน้ำหนักช่องสัญญาณในเครือข่ายโอเอสพีเอฟนี้สามารถให้ค่าประสิทธิภาพเชิงสมรรถนะหรือดีเลย์ที่ดีกว่าการใช้ค่าน้ำหนักแบบดั้งเดิมโดยเฉลี่ยที่ 1.24 เท่า โดยมีแนวโน้มที่จะมีประสิทธิภาพเพิ่มขึ้นเมื่อเครือข่ายมีความซับซ้อนมากขึ้นอีกด้วย

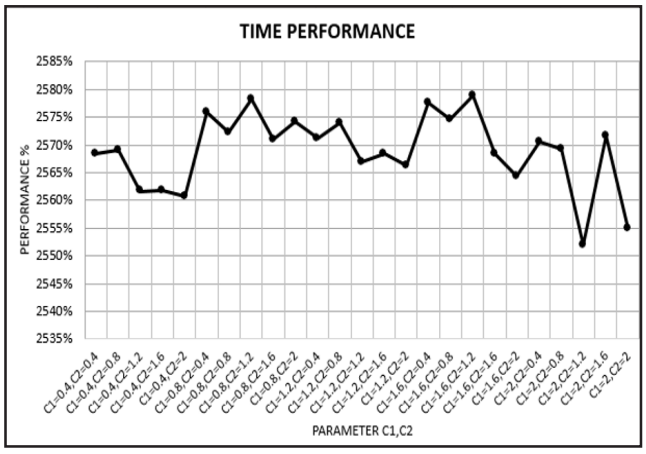
ส่วนการเปรียบเทียบประสิทธิภาพของวิธีกลุ่มอนุภาคนี้กับวิธีโปรแกรมเชิงเส้น พบว่าวิธีกลุ่มอนุภาคให้เครือข่ายที่มีประสิทธิภาพเชิงสมรรถนะหรือดีเลย์ที่ดีเกือบเท่ากับวิธีโปรแกรมเชิงเส้น ได้ประสิทธิภาพโดยเฉลี่ยถึง 94% ของวิธีโปรแกรมเชิงเส้น ในขณะที่ประสิทธิภาพเชิงเวลาหรือเวลาที่ใช้ในการประมวลผลของวิธีกลุ่มอนุภาคนี้ดีกว่าหรือเร็วกว่าวิธีโปรแกรมเชิงเส้นมากถึง 25.69 เท่า จึงสรุปได้ว่าวิธีแบบกลุ่มอนุภาคนี้เหมาะสมที่จะใช้แก้ปัญหาการหาค่าน้ำหนักของช่องสัญญาณในเครือข่ายโอเสพีเอฟเพราะสามารถหาคำตอบของปัญหาได้อย่างรวดเร็ว และได้ค่าประสิทธิภาพที่ดีใกล้เคียงกับ Optimum Solution ของวิธีการโปรแกรมเชิงเส้น เลยทีเดียว

5.2 ข้อเสนอแนะ

การวิจัยต่อในอนาคต สามารถให้ผลการทดสอบมีค่าประสิทธิภาพที่แม่นยำมากขึ้นได้ด้วยการปรับค่าพารามิเตอร์ให้มีความละเอียดมากยิ่งขึ้น ปรับจำนวนรอบการทำงานให้มากขึ้น หรือทำการคำนวณค่าน้ำหนักช่องสัญญาณด้วยอัลกอริทึมอื่นๆ [17] เช่น Memetic Algorithm (MA), Shuffled Frog Leaping Algorithm (SFL) เป็นต้น



ภาพที่ 7 *ค่าประสิทธิภาพเชิงสมรรถนะของวิธีแบบกลุ่มอนุภาคเทียบกับวิธีการโปรแกรมเชิงเส้น*



ภาพที่ 8 *ค่าสมรรถนะเชิงเวลาของวิธีแบบกลุ่มอนุภาคเทียบกับวิธีการโปรแกรมเชิงเส้น*

6. เอกสารอ้างอิง

[1] วิรินทร์ เมฆประดิษฐสิน. *คัมภีร์ระบบเครือข่ายแบบฉบับอาจารย์วิรินทร์.* เล่ม 1, ซีเอ็ดกรุงเทพฯ : ยูเคชั่น จำกัด (มหาชน), 2547.

[2] เอกลักษณ์ โสภาพันธ์. “เร้าติ้งโพรโทคอลประเภท Distance Vector.” Available online at http://www.gotoknow.org/posts/306786. (วันที่ค้นคว้าข้อมูล 5 กรกฎาคม 2556).

[3] E. W. Dijstra. “A note on two problems in connexion with graphs.” *Numerische Mathematik,* Vol. 1, pp. 269-271, 1959.

[4] T. H. Cormen, C. E. Leiserson, R. L. Rivest and S. Clifford. “Section 24.3: Dijkstra’s algorithm.” *Introduction to Alogorithm (Second ed.),* MIT Press and McGraw-Hill., pp. 595-601, 2001.

[5] G. Beni and J. Wang. “Swarm Intelligence in Cellular Robotic System.” *Proceed. NATO Advanced Workshop on Robots and Biological Systems,* Tuscany, Italy, June 26-30, 1989.

[6] N. Frobes. “Biologically Inspired Computing.” *Computing in Science and Engineering,* Vol. 2, No. 6, pp. 83-87, November-December 2000.

[7] G. V. Cybenko. “Approximation by Superposition of a Sigmoidal function.” *Mathematics of Control, Signal, and System,* Vol. 2, pp. 303-314, 1989.

[8] ณัฐกุล หาญกิตติกาญจนา และ กายรัฐ เจริญราษฎร์.

"การใช้วิธีเชิงพันธุกรรมในการตั้งค่าน้ำหนักให้แก่เครือข่ายโอเอสพีเอฟ." *งานประชุมวิชาการระดับชาติมหาวิทยาลัยราชภัฏนครปฐม,* ครั้งที่ 4, 12-13 กรกฎาคม 2555.

[9] เอกวุฒิ ไชยรัตน์ และ กายรัฐ เจริญราษฎร์. "การใช้วิธีอาณานิคมมดในการตั้งค่าน้ำหนักให้แก่เครือข่ายโอเอสพีเอฟ." *การประชุมวิชาการระดับชาติด้านคอมพิวเตอร์และเทคโนโลยีสารสนเทศ,* ครั้งที่ 8, 9-10 พฤษภาคม 2555.

[10] H. A. Abbass. "Marriage in Honey Bees Optimization: A haplometrosis polygynous swarming approach." *In Proceedings of Congress on Evolutionary Computation,* pp. 207-214, Seoul, Korea, 27-30 May 2001.

[11] J. Kennedy and R. Eberhart. "Particle Swarm Optimization." *In Proceedings of IEEE International Conference on Neural Networks IV,* pp. 1942-1948, 1995.

[12] Y. Shi and R .C. Eberhart. "A modified particle swarm optimizer." *In Proceedings of IEEE International Conference on Evolutionary Computation,* pp. 69-73, 1998.

[13] สุภกิจ นุตยะกุล. "การคำนวณเชิงวิวัฒนาการระหว่างเจเนติกอัลกอริทึมกับพาติเคิลสวอมออปติไมเซชั่น." *วารสารสถาบันเทคโนโลยีพระจอมเกล้าคุณทหารลาดกระบัง,* ฉบับที่ 2, กรกฎาคม-ธันวาคม, 2554.

[14] พิสิษฐ์ ชาญเกียรติก้อง. "การออกแบบโครงข่ายคอมพิวเตอร์." สำนักพิมพ์มหาวิทยาลัยรังสิต, กรุงเทพ, 2550.

[15] F. Bernard and T. Mikkel. "Internet traffic engineering by optimizing OSPF weights." *In Proceedings of IEEE INFOCOM,* Vol. 2, pp. 519-528, March 2000.

[16] กฤษณะ ลานทอง และ กายรัฐ เจริญราษฎร์. "การหาค่าน้ำหนักของเส้นทางที่คุ้มค่าที่สุดในโครงข่ายโอเอสพีเอฟด้วยวิธีโปรแกรมเชิงเส้น." *การประชุมวิชาการแห่งชาติมหาวิทยาลัยเกษตรศาสตร์ วิทยาเขตกำแพงแสน,* ครั้งที่ 9, 6-7 ธันวาคม 2555.

[17] วุฒิชัย วงษ์ทัศนีย์กร. "การวิจารณ์และเปรียบเทียบ 5 อัลกอริธึมทางด้านวิวัฒนาการ." *วารสารวิชาการพระจอมเกล้าพระนครเหนือ,* ปีที่ 19, ฉบับที่ 2, หน้า 285-290, พฤษภาคม-สิงหาคม 2552.