

การพัฒนาอัลกอริทึมสำหรับวาดแผนผังการศึกษาของหลักสูตรโดยอัตโนมัติ A Development of Algorithm for Automated Drawing of Course Plan Diagram

จิราพร ธัญญประเสริฐกุล*
Jiraporn Thanyaprasertkul*

บทคัดย่อ

บทความนี้นำเสนองานวิจัยเพื่อพัฒนาอัลกอริทึมสำหรับการวาดแผนผังการศึกษาของหลักสูตรของมหาวิทยาลัยเทคโนโลยีพระจอมเกล้าพระนครเหนือ เพื่อที่จะสามารถใช้โปรแกรมคอมพิวเตอร์ช่วยในการวาดแผนผังการศึกษาได้โดยอัตโนมัติและลดภาระงานของบุคลากรในการจัดทำเอกสารหลักสูตร ในงานวิจัยนี้ได้ประยุกต์ใช้อัลกอริทึมสำหรับการจัดวางกราฟประเภทกราฟระดับชั้นซึ่งแผนผังการศึกษามีลักษณะใกล้เคียงกับกราฟประเภทนี้มากที่สุด และใช้วิธีการแก้ปัญหาแบบฮิวริสติกสำหรับการแก้โจทย์ปัญหาที่อาจจะไม่มีคำตอบที่ถูกต้องสมบูรณ์ ผลการทดลองทำให้เห็นว่าอัลกอริทึมที่พัฒนาขึ้นสามารถวาดแผนผังการศึกษาได้ซึ่งจะนำไปใช้กับโปรแกรมคอมพิวเตอร์สำหรับบันทึกและจัดทำข้อมูลหลักสูตรของมหาวิทยาลัยได้ต่อไป

คำสำคัญ : อัลกอริทึมการจัดวางกราฟ กราฟระดับชั้น
กราฟเลเยอร์

Abstract

This paper presents a development of algorithm to automatically draw curriculum plan diagram of King

Mongkut's University of Technology North Bangkok by using a computer program to reduce human tasks in preparing curriculum documents. The algorithm has been applied for Hierarchical Graph layout, the closest type of graph to course plan diagram. The heuristic approach has been applied to solve common sense problem where there might be no perfect solution for such circumstances. The experimental results showed that the developed basic algorithm can automatically draw course plan diagram. The developed algorithm will be further implemented to the computer program to provide curriculum university database.

Keywords : Graph Layout Algorithm, Hierarchical Graph, Layered Graph

1. บทนำ

มหาวิทยาลัยเทคโนโลยีพระจอมเกล้าพระนครเหนือ ได้มีการจัดทำฐานข้อมูลคอมพิวเตอร์สำหรับหลักสูตรที่เปิดสอนมาตั้งแต่ปี พ.ศ. 2545 และได้มีการนำข้อมูลหลักสูตรนี้มาใช้ในระบบงานคอมพิวเตอร์ต่างๆ เช่น ระบบลงทะเบียนเรียนของนักศึกษา ระบบงานจัด

* นักวิชาการคอมพิวเตอร์ สำนักคอมพิวเตอร์และเทคโนโลยีสารสนเทศ มหาวิทยาลัยเทคโนโลยีพระจอมเกล้าพระนครเหนือ Tel. 0-2555-2000, Ext. 2213, E-mail: jiraporn@kmutnb.ac.th

ตารางสอน-ตารางสอบ เป็นต้น ซอฟต์แวร์โปรแกรมที่ใช้สำหรับจัดทำข้อมูลหลักสูตรนี้พัฒนาขึ้นโดยสำนักคอมพิวเตอร์และเทคโนโลยีสารสนเทศ เป็นโปรแกรมเพื่อให้ผู้ใช้งานสามารถกำหนดโครงสร้างของหลักสูตร ป้อนข้อมูลรายวิชา กำหนดเงื่อนไขบูรพวิชา ตลอดจนจัดแผนการศึกษา และจัดเก็บลงฐานข้อมูลได้ทั้งหมด ข้อมูลเหล่านั้นนอกจากจะเป็นข้อมูลพื้นฐานสำหรับระบบงานต่างๆ ดังกล่าวข้างต้นแล้ว โปรแกรมยังสามารถนำมาสร้างเป็นไฟล์เอกสาร (Microsoft Word Document) เพื่อแสดงรายละเอียดของหลักสูตรในรูปแบบที่เป็นมาตรฐานเดียวกันทุกหลักสูตรได้ อย่างไรก็ตามข้อจำกัดของโปรแกรมในขณะนี้คือสามารถแสดงได้เฉพาะข้อมูลที่เป็นตัวหนังสือเท่านั้น แต่ในส่วนของแผนการศึกษาซึ่งตามปกติในเอกสารหลักสูตรของแต่ละภาควิชาจัดทำขึ้นนั้น จะมีการวางแผนผังการศึกษาซึ่งนอกจากจะแสดงรายละเอียดเกี่ยวกับรายวิชาที่ต้องเรียนในแต่ละภาควิชาแล้วยังสามารถแสดงเส้นความต่อเนื่องระหว่างวิชาบังคับแต่ละวิชาด้วย ซึ่งลำดับในการวาดหรือการที่จะเรียงวิชาลงในแผนผังเพื่อให้สามารถดูได้ง่ายและให้มีการลากเส้นหรือมีจุดตัดของเส้นให้น้อยที่สุดนั้นยังคงจำเป็นต้องใช้มนุษย์ช่วยในการตัดสินใจและวาดแผนผังเหล่านั้นเอง ทำให้ไฟล์เอกสารหลักสูตรที่สร้างจากโปรแกรมนั้นไม่สมบูรณ์เท่าที่ควร งานวิจัยนี้จึงมุ่งที่จะศึกษาและนำหลักการของ Graph Layout มาประยุกต์ใช้ เพื่อเพิ่มศักยภาพให้โปรแกรมสามารถวาดแผนผังการศึกษาจากฐานข้อมูลที่มีอยู่ได้อย่างอัตโนมัติ

วัตถุประสงค์การวิจัยครั้งนี้คือพัฒนาอัลกอริทึมการวาดแผนผังการศึกษาของหลักสูตรโดยอัตโนมัติและขอบเขตการทำวิจัยจะพัฒนาเฉพาะส่วนอัลกอริทึมในการวาดกราฟ โดยยังไม่ได้นำอัลกอริทึมที่พัฒนาได้นี้เข้าไปรวมในโปรแกรมหลักสูตร

2. ทฤษฎีที่เกี่ยวข้อง

2.1 การทบทวนวรรณกรรม

งานวิจัยนี้ต้องการนำเสนอข้อมูลแผนการศึกษาในหลักสูตรออกมาเป็นภาพได้โดยอัตโนมัติ ซึ่งปัญหานี้

จัดอยู่ในกลุ่มงานวิจัยประเภทการจัดวางกราฟ (Graph Layout) ที่ต้องใช้คอมพิวเตอร์ช่วยในการจัดเรียงวัตถุเพื่อแสดงภาพให้สวยงามหรือง่ายต่อการทำความเข้าใจของมนุษย์ โดยในที่นี้รายวิชาที่จะต้องเรียนในแต่ละภาคการศึกษาจะถูกเปลี่ยนให้เป็นจุดยอดของกราฟ และความสัมพันธ์ระหว่างบูรพวิชาจะถูกเปลี่ยนให้เป็นเส้นเชื่อมของกราฟ จากนั้นจึงใช้อัลกอริทึมคำนวณหาภาพการจัดวางกราฟที่เหมาะสม

อัลกอริทึมเกี่ยวกับการจัดวางกราฟซึ่งเป็นที่อ้างอิงมากในปัจจุบันมีอยู่สองแนวทางใหญ่ๆ คือแนวทางของ Sugiyama [4] ซึ่งเป็นอัลกอริทึมที่พัฒนาขึ้นมาสำหรับการวาดกราฟระดับชั้นโดยเฉพาะ (Hierarchical หรือ Layered Graph) โดยมุ่งแก้ปัญหาการลดจุดตัดภายในกราฟและหาอัลกอริทึมสำหรับการวางตำแหน่งของจุดยอดของกราฟให้ได้ผลออกมาดีที่สุด Garsner [3] ได้พัฒนาอัลกอริทึมตามแนวทางของ Sugiyama แต่ได้ปรับปรุงวิธีการในขั้นตอนย่อยๆ ส่วนอีกแนวทางหนึ่งของการจัดวางกราฟเป็นแนวทางของ Eades [5] ได้นำเสนอวิธีวาดกราฟแบบที่เรียกว่าการใช้แรงก้ำกั (Force-Directed Graph Drawing) โดยให้มองจุดยอดของกราฟเป็นวัตถุทางฟิสิกส์และมองเส้นเชื่อมเป็นสปริงแล้วทำการจำลองแรงดึงและแรงผลักระหว่างวัตถุเพื่อให้วัตถุเคลื่อนที่เข้าหาหรือออกจากกัน จนกว่ากราฟจะถึงจุด “สมดุล” ทางฟิสิกส์ แนวทางของ Eades นี้สามารถนำไปประยุกต์ใช้งานกับกราฟทั่วไปได้กว้างกว่าไม่จำกัดแค่เพียง Layered Graph เท่านั้น มีงานวิจัยมากมายที่พัฒนาต่อยอดจากวิธีการของ Eades เช่น Kamada [7] นำเสนอการคิดแรงกระทำระหว่างวัตถุด้วยโดยไม่จำกัดเพียงแรงสปริง เพื่อให้ระยะห่างของเส้นเชื่อมของแต่ละจุดยอดมีผลต่อการจัดวางตำแหน่งของแต่ละจุดยอดด้วย และยังมีงานวิจัยที่เสนอวิธีการแบ่งกลุ่มกราฟ [1],[2] เพื่อให้หน้าอัลกอริทึมไปใช้กับกราฟขนาดใหญ่ได้ประสิทธิภาพดีขึ้น ส่วน Davidson [6] เสนอการนำอัลกอริทึมจำลองการอบเหนียว (Simulated Annealing) ซึ่งเป็นกลวิธีฮิวริสติกสำหรับการค้นหาเฉพาะที่ชนิดหนึ่งมาใช้ในการแก้โจทย์ปัญหาการจัดวางกราฟ

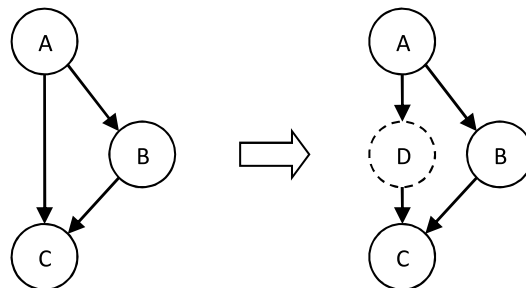
ในงานวิจัยนี้ได้ใช้แนวทางแก้ปัญหาของทั้ง Sugiyama และ Eades ร่วมกัน โดยใช้โครงการทำงาน ของ Sugiyama เป็นหลัก นั่นคือการแบ่งขั้นตอนการ วาดกราฟออกเป็นสี่ขั้นตอน ประกอบด้วย การ Ranking, Cross Reduction, Positioning และ Lining และเนื่องจาก แผนผังการศึกษาเป็นข้อมูลที่มีลักษณะเป็นกราฟระดับ ชั้นในตัวอยู่แล้ว โดยในแต่ละภาคการศึกษาจะแทนด้วย ระดับชั้นของกราฟ ดังนั้นจึงสามารถปรับเข้าใช้งานกับ อัลกอริทึมของ Sugiyama ได้โดยง่าย ในขณะที่ขั้นตอน ของการลดจุดตัดจะเปลี่ยนไปใช้แนวทางของ Eades นั่น คือการใช้วิธีแบบ Force-Directed กับทั้งกราฟแทน เพื่อลดจุดตัดที่ละระดับชั้น เนื่องจากมีความซับซ้อน น้อยกว่าและให้ผลลัพธ์ที่ดีกว่า สำหรับรายละเอียดของ อัลกอริทึมสามารถอธิบายได้ดังหัวข้อต่อไป

2.2 การวาดกราฟระดับชั้น

โจทย์ปัญหาการวาดกราฟระดับชั้นนั้น ตาม แนวทางของ Sugiyama ได้แบ่งขั้นตอนไว้สี่ขั้นดังนี้

1. Ranking คือขั้นตอนของการเตรียมข้อมูลกราฟ โดยทำการกำหนดระดับให้กับแต่ละจุดยอด และโดย เฉพาะสำหรับ Directed Graph ถ้าเป็นกราฟระดับชั้น ในแนวนอน ควรจะจัดให้จุดยอดซึ่งอยู่ด้านต้นของเส้น เชื่อมอยู่ระดับบน และจัดให้จุดยอดซึ่งอยู่ปลายของ เส้นเชื่อมอยู่ระดับล่าง ในบางกรณีอาจต้องมีการ “ตัด แต่ง” กราฟเพื่อให้เป็นกราฟอวัฏจักร (Acyclic) ก่อน ด้วย เพื่อที่จะพิจารณาได้ว่าควรจะนำจุดยอดใดไว้ด้าน บนหรือด้านล่าง นอกจากนี้กราฟที่ “เหมาะสม” จะ นำมาจัดวาง โดยมีเส้นเชื่อมที่ไม่ยาวข้ามเกินหนึ่ง ระดับ จึงจำเป็นต้องมีขั้นตอนการตัดแต่งกราฟเพื่อให้ แต่ละเส้นเชื่อมยาวข้ามไม่เกินหนึ่งระดับด้วย โดยการ เพิ่ม Dummy Vertex เข้าแทรกในเส้นเชื่อมที่มีความ ยาวเกิน ก็จะทำให้เส้นเชื่อมที่ยาวนั้นถูกแบ่งออก เป็นเส้นเชื่อมหลายเส้นที่มีความยาวไม่เกินหนึ่งระดับ ดังตัวอย่างรูปที่ 1

2. Cross Reduction คือขั้นตอนการลดจุดตัดของ กราฟ กล่าวคือเป็นขั้นตอนของการจัดลำดับจุดยอดใน



รูปที่ 1 ตัวอย่างการแทรก Dummy Vertex เข้าในเส้นเชื่อม ที่ยาวเกินหนึ่งระดับ

แต่ละระดับใหม่ เพื่อให้เกิดจุดตัดระหว่างเส้นเชื่อมโดย รวมของทั้งกราฟให้น้อยที่สุด ตามแนวทางของ Sugiyama จะทำการลดรูปโจทย์ปัญหาโดยทำการลดจุดตัดที่ละ ระดับชั้น โดยที่โจทย์ปัญหาการลดจุดตัดนี้จัดว่าอยู่ใน กลุ่ม NP-Hard จึงต้องใช้วิธีฮิวริสติก (Heuristic) ใน การแก้ปัญหา ซึ่งรายละเอียดจะได้กล่าวถึงต่อไป

3. Positioning คือขั้นตอนการวางตำแหน่งจริงหรือ กำหนดพิกัดแกน X ให้แก่จุดยอด ผลจากการเรียงลำดับ จะทำให้เกิดจุดตัดน้อยที่สุดแล้ว อย่างไรก็ตามจำเป็นจะ ต้องหาพิกัด X ของแต่ละจุดยอดที่เหมาะสม เพื่อที่จะ ให้เส้นเชื่อมมีความตรงมากที่สุด จุดยอดจัดเรียงอยู่ใกล้ ชิดกันและมีความสมดุลมากที่สุด ซึ่งก็แก้ปัญหาโดยใช้วิธี ฮิวริสติกเช่นกัน

4. Lining คือขั้นตอนของการวาดภาพกราฟ โดย วาดจุดยอดตามพิกัดที่หาได้แล้ววาดเส้นเชื่อม ทั้งนี้จะ ต้องตัด Dummy Vertex ออกไปและเปลี่ยนเส้นเชื่อมที่ ถูกตัดคั่นให้เป็นเส้นเดียว โดยในขั้นตอนนี้มีเทคนิคใน การวาดเส้นหลายแบบ เช่น วาดเป็นเส้นตรง เส้นโค้ง (Spline) หรือเส้นหักมุม (Orthogonal) นอกจากนี้ยังมี งานวิจัยซึ่งคำนึงถึงความสวยงามอื่นๆ เช่น ตำแหน่งที่ ลากเส้นออกจากจุดยอด (Node Port) หรือการกระโดด ข้ามเส้น เป็นต้น

2.3 การทำ Ranking

โดยสรุปแล้วขั้นตอนการ Ranking จะทำให้เกิด

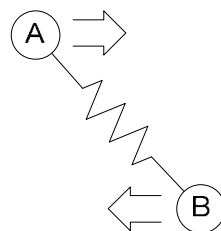
ผลสองประการคือ การแบ่งกลุ่มจุดยอดของกราฟเข้าในแต่ละระดับ และการตัดเส้นเชื่อมที่ยาวเกินหนึ่งระดับโดยแทรก Dummy Vertex เข้าไปคั่นระหว่างกลาง ซึ่งสำหรับข้อมูลหลักสูตรนั้น เราทราบอยู่แล้วว่าวิชาใดต้องจัดวางไว้ที่ภาคการศึกษาใด นั่นหมายความว่าเราทราบระดับของแต่ละจุดยอดอยู่แล้ว และแผนผังการศึกษายังเป็นกราฟ Acyclic ในตัวอยู่แล้วเช่นกัน ดังนั้นในขั้นตอนนี้จึงเหลือเพียงการตัดแต่งกราฟเพื่อให้แต่ละเส้นเชื่อมยาวข้ามไม่เกินหนึ่งระดับเท่านั้น

2.4 การทำ Cross Reduction

การทำ Cross Reduction คือการจัดลำดับของจุดยอดในแต่ละระดับชั้น เพื่อให้เส้นเชื่อมในกราฟนั้นตัดกันเป็นจำนวนที่น้อยที่สุด ก่อนการทำ Cross Reduction จำเป็นต้องใส่ลำดับเริ่มต้นให้แก่ทุกจุดยอดก่อน โดย Garsner [3] ได้แนะนำให้ใส่ลำดับเริ่มต้น (Initial Order) โดยใช้อัลกอริทึมการค้นหาในแนวลึก (Depth-First Search, DFS) ซึ่งจะเป็นการ “คลายปม” ในเบื้องต้นให้กับกราฟ ก่อนที่จะทำการลดจุดตัด

สำหรับขั้นตอนการลดจุดตัดนั้น Sugiyama ได้เสนอแนวทางการลดจุดตัดภายในกราฟด้วยวิธีฮิวริสติก โดยให้มีเครื่องจักรทัวริงทำการลดจุดตัดทีละชั้น กวาดตั้งแต่ชั้นบนสุดลงมาชั้นล่างสุดและกวาดย้อนกลับขึ้นไปทำซ้ำอย่างนี้หลายครั้ง ในแต่ละชั้นที่กวาดผ่านก็จะทำการคำนวณลำดับของจุดยอดใหม่ โดยคำนวณจากค่าศูนย์กลาง (Barycenter) ระหว่างจุดยอดหนึ่งๆ กับจุดยอดอื่นๆ ซึ่งเชื่อมถึงกันอยู่บางงานวิจัยเลือกใช้ค่ากลาง (Median) แทนค่าศูนย์กลาง ซึ่งวิธีนี้จะเป็นการดึงให้จุดยอดที่มีเส้นเชื่อมถึงกันเคลื่อนเข้ามาอยู่ใกล้กัน อย่างไรก็ตามข้อเสียที่พบในอัลกอริทึมนี้คือ การหาค่าศูนย์กลางหรือค่ากลางที่ดี เป็นการกระทำภายในแต่ละระดับชั้นโดยไม่ได้สะท้อนถึงภาพรวมของทั้งกราฟ

ในงานวิจัยนี้ได้ทำการประยุกต์โดยนำแนวทางของ Eades มาใช้ในขั้นตอนการลดจุดตัด ซึ่ง Eades ก็แก้โจทย์ปัญหาการลดจำนวนจุดตัดโดยใช้ฮิวริสติกเช่นกัน แนวทางของ Eades ให้มองจุดยอดของกราฟเป็นวัตถุ



รูปที่ 2 แนวคิดในการมองจุดยอดเป็นวัตถุและเส้นเชื่อมเป็นสปริง จุดยอดจะเคลื่อนที่ตามแรงดึงดูดของสปริง

```
procedure CrossReduction(G: Graph)
begin
  Initial vertices' order using DFS
  repeat
    for each vertex in G
      Calculate the force on vertex
      Move the vertex by  $C4 * \text{force}$ 
    next
  until M times
end
```

รูปที่ 3 Pseudo Code อัลกอริทึมในการลดจุดตัด

ทางฟิสิกส์ และมองเส้นเชื่อมเป็นสปริง ดังรูปที่ 2 จากนั้นจึงทำ Iterate เพื่อคำนวณแรงที่กระทำกับจุดยอดแต่ละจุดโดยรวมแล้วให้จุดยอดนั้นเคลื่อนที่ตามแรงที่ถูกกระทำ ในงานวิจัยนี้ได้ปรับแก้วิธีการเล็กน้อยกล่าวคือ เนื่องจากเราต้องการผลลัพธ์เป็นกราฟระดับชั้น เวคเตอร์ของแรงจากสปริงที่คำนวณได้นั้นจึงแตกมาเฉพาะแรงในแกน X เพื่อให้จุดยอดเกิดการเคลื่อนที่เฉพาะในแกน X เท่านั้น จะเห็นได้ว่าเป็นหลักการเดียวกันกับที่ Sugiyama ใช้ นั่นคือใช้เส้นเชื่อมเป็นสิ่งที่ดึงดูดจุดยอดเข้าหากัน แต่วิธีของ Eades นี้จะเกิดผลโดยรวมกับทั้งกราฟ ไม่ใช่แค่เพียงระหว่างชั้น อัลกอริทึมของ Eades แสดงได้ดังรูปที่ 3

2.5 การ Positioning

ขั้นตอนการ Positioning คือการหาพิกัดหรือตำแหน่งของจุดยอดที่ดีที่สุด ที่จะทำให้กราฟที่ได้มี

ความสวยงามกล่าวคือเส้นเชื่อมในกราฟมีความตรง และสั้นที่สุดโดยการหาตำแหน่งนี้ยังคงอิงการจัดลำดับของจุดยอดในแต่ละชั้นจากขั้นตอนการทำ Cross Reduction ในที่นี้เป็นการหาพิกัดในแกน X หรือแนวนอนของแต่ละจุดยอดเท่านั้น เนื่องจากเราได้กำหนดพิกัดในแกน Y ของแต่ละจุดยอดมาแล้วตั้งแต่ขั้นตอนการ Ranking

วิธีการหาพิกัดของจุดยอดตามแนวทางของ Sugiyama เรียกว่า Priority Layout Method ซึ่งเป็นการแก้ปัญหาแบบฮิวริสติก คือให้ทดลองเคลื่อนย้ายจุดยอดในแต่ละระดับชั้น โดยกวาดตั้งแต่ชั้นบนสุดลงมาชั้นล่างสุดและกวาดย้อนกลับขึ้นไป ในแต่ละครั้งที่กวาด จะทำการคำนวณหาค่าศูนย์ถ่วงของแต่ละจุดยอดในชั้นที่กวาดนั้นกับจุดยอดอื่นๆ ที่เชื่อมถึงกันอยู่ เพื่อที่จะหาตำแหน่งซึ่งจะเป็น “สมดุล” ของจุดยอดนั้นๆ และสาเหตุที่วิธีการนี้เรียกว่า Priority Layout Method นั้นเนื่องจาก ลำดับในการคำนวณของแต่ละจุดยอดนั้น จะขึ้นอยู่กับค่าคะแนนความสำคัญของจุดยอด โดยที่จุดยอดซึ่งมีเส้นเชื่อมจำนวนมากกว่า จะมีค่าคะแนนความสำคัญมากกว่า และจะได้รับการจัดวางตำแหน่งก่อนจุดยอดในชั้นเดียวกันซึ่งมีค่าคะแนนความสำคัญน้อยกว่า ในงานวิจัยนี้นอกจากการเคลื่อนย้ายตำแหน่งของจุดยอดนั้นยังได้นำผลลัพธ์จากการเคลื่อนย้ายมาเข้าฟังก์ชันวัตถุประสงค์ (Objective Function) เพื่อหาผลลัพธ์ที่ให้ความยาวของเส้นเชื่อมที่มีค่าน้อยที่สุดตามแนวทางของ Gansner ด้วย โดยแสดงโจทย์ปัญหา Optimization Problem ได้ดังนี้

$$\min \sum_{e=(v,w)} \Omega(e) |x_w - x_v|$$
$$\text{subject to: } x_b - x_a \geq \rho(a, b)$$

โดยที่ $|x_w - x_v|$ คือผลต่างของพิกัดแกน X ของจุดยอดที่ด้านต้นและด้านปลายของเส้นเชื่อม และ $\Omega(e)$ คือค่าถ่วงน้ำหนักพิเศษ โดยให้มีค่าเท่ากับ 1 สำหรับเส้นที่เชื่อม Vertex จริงทั้งสองด้าน มีค่าเท่ากับ 2 สำหรับเส้นที่เชื่อม Vertex จริงเพียงด้านใดด้านหนึ่ง

และให้มีค่าเท่ากับ 8 สำหรับเส้นที่เชื่อมกับ Dummy Vertex ทั้งสองด้าน ฟังก์ชันสำหรับการ Optimize เช่นนี้เพื่อที่จะให้คะแนนกับ Layout ที่มีเส้นเชื่อม Dummy Vertex เป็นเส้นตรงในแนวตั้งมากที่สุด ทั้งหมดนี้มีข้อบังคับว่าให้ a คือจุดยอดที่อยู่ทางซ้ายของ b บนระดับชั้นเดียวกันและ

$$\rho(a, b) = \frac{xsize(a) + xsize(b)}{2} + nodesep(G)$$

หมายถึงระยะห่างระหว่าง a และ b จะต้องเท่ากับ ความกว้างครึ่งหนึ่งของจุดยอด a บวกกับครึ่งหนึ่งของจุดยอด b บวกกับค่าระยะห่างระหว่าง Node ปกติของกราฟ

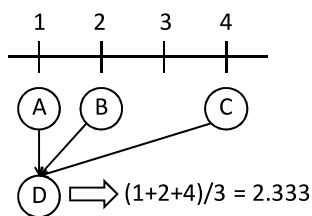
สำหรับขั้นตอนในการหาตำแหน่งของจุดยอดด้วยวิธี Priority Layout Method เป็นดังนี้

1. กำหนดค่าพิกัดแกน X เริ่มต้นตามเลขลำดับของจุดยอดของแต่ละระดับชั้น

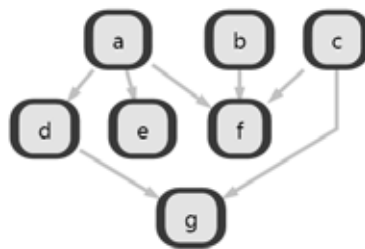
2. ถ้าให้ระดับชั้นแรกคือระดับที่ 1 การหาตำแหน่งของจุดยอดจะเริ่มทำตั้งแต่ระดับชั้นที่ 2, 3 ไปจนถึง n ซึ่งเป็นการทำจากบนลงล่าง จากนั้นจึงค้นหาตำแหน่งของจุดยอดในระดับที่ $n-1$ ย้อนกลับไปถึงระดับที่ 1 ซึ่งเป็นการทำจากล่างขึ้นบน

3. ในการหาตำแหน่งของระดับชั้นหนึ่งๆ จะวางตำแหน่งให้กับจุดยอดที่มีค่าความสำคัญมากที่สุดก่อน แล้วจึงมองหาไปยังจุดยอดที่มีความสำคัญรองลงไป โดยค่าความสำคัญนี้พิจารณาจากจำนวนเส้นเชื่อมที่ต่ออยู่กับจุดยอดนั้น โดยในการหาจากบนลงล่างจะดูค่าความสำคัญจากจำนวนเส้นเชื่อมที่อยู่ด้านบนของจุดยอด ในขณะที่การหาจากล่างขึ้นบนจะดูค่าความสำคัญจากจำนวนเส้นเชื่อมที่อยู่ด้านล่างของจุดยอด

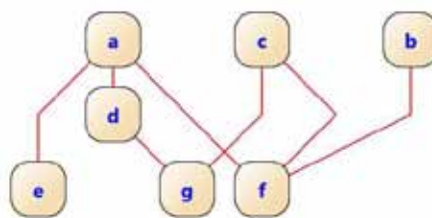
4. จากนั้นจึงคำนวณหาตำแหน่งใหม่ของจุดยอดได้จากการนำค่าพิกัดแกน X ของจุดยอดต่างๆ ที่เชื่อมอยู่ในระดับบนหรือล่างตามแต่กรณีมาหาค่าศูนย์ถ่วง (Barycenter) โดยหากเป็นการหาจากบนลงล่างจะคำนวณค่าศูนย์ถ่วงกับแถวด้านบน แต่หากเป็นการหาแบบล่างขึ้นบนจะคำนวณค่าศูนย์ถ่วงกับแถวด้านล่าง



รูปที่ 4 การคำนวณหาตำแหน่งของจุดยอดโดยหาจากค่าศูนย์ถ่วงเมื่อเทียบกับจุดยอดด้านบน



(ก) ผลการจัดวางกราฟด้วยอัลกอริทึม Sugiyama



(ข) ผลการจัดวางกราฟด้วยอัลกอริทึมที่พัฒนาขึ้นในงานวิจัย

รูปที่ 6 ผลการจัดวางกราฟด้วยอัลกอริทึม Sugiyama และอัลกอริทึมที่พัฒนาขึ้นในงานวิจัย

```

procedure Positioning(G: Graph)
begin
  Initial vertices' X position by current
  ordering number
  GBest = G
  repeat
    Direction = if(odd time, "DOWN", "UP")
    for each rank swept by Direction
      for each vertex order by Priority
        Calculate Barycenter position
        Move vertex to that position
      next
    next
    if G is better than GBest then
      begin
        GBest = G
      end
    until M times
  end
end

```

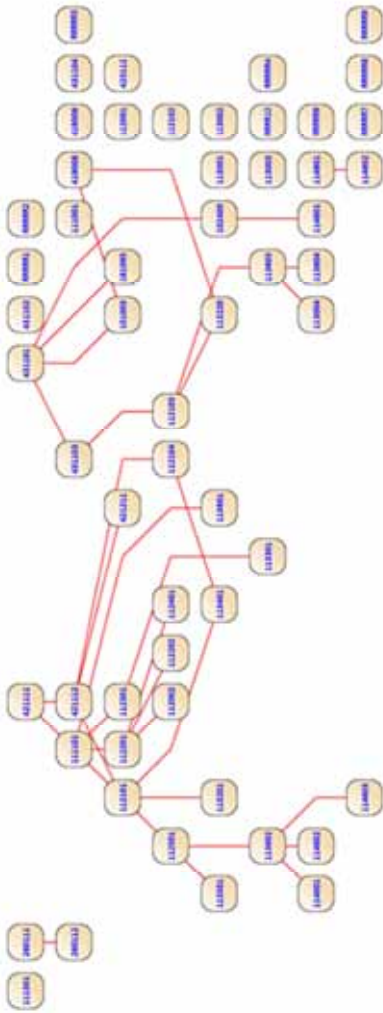
รูปที่ 5 Pseudo Code อัลกอริทึมในการ Positioning

ดังตัวอย่างรูปที่ 4 และอัลกอริทึมของขั้นตอนทั้งหมดแสดงได้ดังรูปที่ 5

3. ผลการวิจัย

ในการวิจัยนี้ได้ใช้โปรแกรม Graph Sharp สำหรับสร้างกราฟด้วยอัลกอริทึม Sugiyama แบบดั้งเดิมแล้วนำมาเปรียบเทียบกับกราฟผลลัพธ์ที่ได้จากโปรแกรมที่เขียนขึ้นใหม่ ตัวกราฟที่นำมาทดสอบได้ทำการสร้างเป็นไฟล์ประเภท GraphML ซึ่งเป็นรูปแบบไฟล์ที่เปิดและเป็นมาตรฐานสำหรับการกำหนดคุณสมบัติของกราฟ จากนั้นนำไปป้อนเข้าโปรแกรม Graph Sharp แล้วนำกราฟเดียวกันนั้นมาป้อนเข้าโปรแกรมที่พัฒนาขึ้นเพื่อเปรียบเทียบผลที่ได้ เนื่องจากงานวิจัยนี้ต้องการที่จะนำ

กราฟมาประยุกต์ใช้ในการแสดงผลแผนการศึกษาของหลักสูตร ซึ่งกราฟแผนการศึกษาจะมีคุณสมบัติพิเศษคือตัวรายวิชาหรือจุดยอดของกราฟจะต้องถูกบังคับตำแหน่งของปีและภาคการศึกษาหรือค่า Rank หรือพิกัดในแกน Y ดังนั้นใน GraphML ที่ได้สร้างขึ้นมาเพื่อทดสอบได้ทำการเพิ่ม Attribute ชื่อว่า "Rank" เข้าไปเป็นกรณีพิเศษ (ไม่ใช่มาตรฐาน GraphML) ซึ่งจะถูกใช้งานโดยโปรแกรมที่พัฒนาขึ้นในงานวิจัยนี้โดยเฉพาะกราฟในรูปที่ 6 (ก) คือผลลัพธ์ในการจัดวางกราฟทดสอบด้วยอัลกอริทึม Sugiyama ที่สร้างขึ้นด้วยโปรแกรม Graph Sharp จากภาพแสดงให้เห็นว่าอัลกอริทึมตามปกติของ Sugiyama นั้นจะพยายามลดความยาวของเส้นเชื่อมให้ได้มากที่สุด โดยที่ไม่มีการบังคับระดับชั้น ซึ่งในกรณีนี้คือจุดยอด e และ f จะถูกดึงเข้าไปใกล้จุดยอด a, b และ c ในขณะที่รูปที่ 6 (ข) คือผลลัพธ์



รูปที่ 7 แผนผังการศึกษาของหลักสูตร EE 411014
สร้างด้วยอัลกอริทึมที่พัฒนาขึ้นในงานวิจัย

ในการจัดวางกราฟทดสอบด้วยอัลกอริทึมที่พัฒนาขึ้นในงานวิจัยนี้ ซึ่งมีการกำหนด Constraint ของจุดยอดไว้ให้ จุดยอด e และ f ต้องอยู่ในระดับที่ 3 เท่านั้น ผลการจัดวางจึงทำให้มีจำนวนจุดตัดที่มากกว่าการจัดวางด้วยอัลกอริทึมแบบปกติ แต่จุดยอด e และ f ยังคงอยู่ในระดับชั้นที่กำหนดไว้ ส่วนในรูปที่ 7 คือผลลัพธ์จากอัลกอริทึมจากงานวิจัยนี้เมื่อนำมาใช้กับข้อมูลแผนการศึกษาจริงของหลักสูตร EE 411014 วิศวกรรมศาสตรบัณฑิต สาขาวิชาวิศวกรรมไฟฟ้า

4. อภิปรายผลและสรุป

แม้ว่าอัลกอริทึมการจัดวางกราฟดั้งเดิมของ Sugiyama จะถูกออกแบบไว้สำหรับกราฟแบบลำดับชั้น แต่ยังมีปัญหาในขั้นตอนของการทำ Cross Reduction ซึ่งหากกราฟไม่ได้รับการ Initial Order อย่างเหมาะสม อัลกอริทึมนี้ก็จะไม่สามารถกำจัดจุดตัดได้ทั้งหมด ในขณะที่อัลกอริทึมที่พัฒนาขึ้นในงานวิจัยนี้ได้นำเอาข้อดีจากอัลกอริทึมของ Eades มาใช้ในระหว่างการทำ Cross Reduction ประเด็นต่อมาคืออัลกอริทึม Sugiyama ในโปรแกรม Graph Sharp ไม่สามารถที่จะใส่ข้อบังคับของระดับชั้นหรือ Rank Constraint ได้ ทำให้กราฟผลลัพธ์ถูกจัดวางโดยที่พยายามทำให้เส้นเชื่อมมีความสั้นที่สุด ซึ่งเป็นการจัดวางที่ดีแต่ไม่เหมาะสมที่จะนำมาใช้กับกราฟแผนการศึกษาที่ระดับชั้นของแต่ละจุดยอดต้องถูกบังคับ ดังจะสังเกตได้จากจุดยอด e และ f ในรูปที่ 6 ซึ่งวางอยู่ในตำแหน่งที่ดี แต่จะผิดพลาดในแง่ของข้อมูลแผนการศึกษา

อย่างไรก็ตามอัลกอริทึมที่พัฒนาขึ้นใหม่นี้ เมื่อนำมาใช้กับข้อมูลหลักสูตรซึ่งมีลักษณะโครงสร้างข้อมูลไม่ได้เป็นรูปต้นไม้เสมอไป กล่าวคือจุดยอดในระดับบนไม่ได้เกิดจากการแตกกิ่งก้านสาขาออกมาจากจุดยอดระดับล่าง แต่จุดยอดระดับบนอาจเชื่อมต่อกับหลายจุดยอดในระดับล่างก็ได้ นอกจากนี้ในอัลกอริทึมยังไม่ได้ทำให้ความกว้างของ Dummy Vertex มีค่าต่ำกว่า Vertex อื่น จึงยังเป็นข้อบกพร่องของอัลกอริทึมในส่วนของการทำ Positioning ซึ่งทำให้ไม่สามารถสร้างกราฟที่มีขนาดความกว้างกระชับได้เท่าที่ควร และในส่วนของการลากเส้นหรือขั้นตอนการ Lining ซึ่งยังไม่ได้ดำเนินการ นอกจากนี้แล้วในการแสดงผลยังควรที่จะปรับปรุงให้นำรหัสวิชา ชื่อย่อ หน่วยกิต มาแสดงผล เพื่อให้เกิดเป็นภาพแผนการศึกษาที่สมบูรณ์ ดังในเอกสารหลักสูตรที่จัดทำด้วยมือมนุษย์ต่อไป

เอกสารอ้างอิง

- [1] A. Quigley, "Large Scale Relational Information Visualization, Clustering, and Abstraction," PhD.

- Thesis, Department of Computer Science and Software Engineering, University of Newcastle, Australia, 2001.
- [2] Daniel Tunkelang, "A Practical Approach to Drawing Undirected Graph, Technical Report," School of Computer Science, Carnegie Mellon University, Jun 1994.
- [3] E.R. Gansner, E. Koutsofios, S. C. North, and K. P. Vo., "A Technique for Drawing Directed Graphs," *IEEE Trans. Software Engineering*, pp. 214-230, 1993.
- [4] K. Sugiyama, S. Tagawa, and M. Toda, "Methods for Visual Understanding of Hierarchical System Structures," *IEEE Transactions on Systems, Man and Cybernetics*, pp. 109-125, 1981.
- [5] P.Eades, *A Heuristic for Graph Drawing*, Congressus Numeratum, 1984
- [6] R. Davidson and D. Harel, "Drawing Graphs Nicely Using Simulated Annealing," *ACM Transactions on Graphics*, pp.301-331, October, 1996.
- [7] T. Kamada and S. Kawai, "An Algorithm for Drawing General Undirected Graphs," *Information Proceeding Letter*, 1989.