การแก้ปัญหาการปรับสมดุลจักรยานแบบสถิตด้วยโครงข่ายที่เรียนรู้แบบเสริมกำลังและการค้นหาแบบทาบู

นายธีร์ธัฐ พรหมประดิษฐ์

วิทยานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรมหาบัณฑิต
สาขาวิชาวิศวกรรมโยธา ภาควิชาวิศวกรรมโยธา
คณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย
ปีการศึกษา 2564

Solving Static Bike Rebalancing Problem with Reinforcement Learning Network and Tabu Search

Mr. Theethad Prohmpradith

A Thesis Submitted in Partial Fulfillment of the Requirements
for the Degree of Master of Engineering in Civil Engineering
Department of Civil Engineering
FACULTY OF ENGINEERING
Chulalongkorn University
Academic Year 2021

หัวข้อวิทยานิพนธ์	การแก้ปัญหาการปรับสมดุลจักรยานแบบสถิตด้วยโครงข่ายที่เรียนรู้แบบเสริมกำลังและการค้นหาแบบทาบู
โดย	นายธีร์ธัฐ พรหมประดิษฐ์
สาขาวิชา	วิศวกรรมโยธา
อาจารย์ที่ปรึกษาวิทยานิพนธ์หลัก	รองศาสตราจารย์ ดร.มาโนช โลหเตปานนท์

คณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย อนุมัติให้นับวิทยานิพนธ์ฉบับนี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรมหาบัณฑิต

---------- คณบดีคณะวิศวกรรมศาสตร์
(ศาสตราจารย์ ดร.สุพจน์ เตชวรสินสกุล)

คณะกรรมการสอบวิทยานิพนธ์

---------- ประธานกรรมการ
(รองศาสตราจารย์ ดร.ศักดิ์สิทธิ์ เฉลิมพงศ์)

---------- อาจารย์ที่ปรึกษาวิทยานิพนธ์หลัก
(รองศาสตราจารย์ ดร.มาโนช โลหเตปานนท์)

---------- กรรมการภายนอกมหาวิทยาลัย
(รองศาสตราจารย์ ดร.สมชาย ปฐมศิริ)

ธีร์ธัฐ พรหมประดิษฐ์ : การแก้ปัญหาการปรับสมดุลจักรยานแบบสถิตด้วยโครงข่ายที่เรียนรู้แบบเสริมกำลังและการค้นหาแบบทาบู. (Solving Static Bike Rebalancing Problem with Reinforcement Learning Network and Tabu Search) อ.ที่ปรึกษาหลัก : รศ. ดร.มาโนช โลหเตปานนท์

ระบบบริการจักรยานให้เช่าเป็นโหมดการเดินทางที่ไม่มีการปล่อยก๊าซหรือของเสีย ปัจจุบันบริการให้เช่าจักรยานเปิดให้ใช้บริการอย่างแพร่หลายเนื่องจากความตื่นตัวต่อภาวะโลกรวน บริการจักรยานให้เช่าเป็นบริการที่ให้ผู้ที่ต้องการใช้ปั่นจักรยานเข้ามาเช่าจักรยานและคืนจักรยานด้วยตนเอง ณ จุดให้บริการ การดำเนินการของบริการจักรยานให้เช่าเป็นปัจจัยที่สำคัญที่ส่งผลต่อประสบการณ์ผู้เช่าอย่างมาก สำหรับปัญหาการดำเนินการของบริการจักรยานให้เช่า ยกตัวอย่างเช่น กรณี ณ สถานีให้เช่าจักรยานไม่มีพื้นที่ว่างสำหรับคืนจักรยานหรือมีจำนวนจักรยานไม่เพียงพอให้เช่า ซึ่งในตัวอย่างนี้จะส่งผลให้ผู้เช่ารู้สึกไม่พึงพอใจต่อระบบให้เช่าจักรยาน การปรับสมดุลจักรยาน คือ การวางแผนการขนส่งเพื่อที่จะหาเส้นทางที่ดีที่สุด มีจุดประสงค์เพื่อทำให้ผู้ดูแลบริการจักรยานให้เช่าสามารถปรับจักรยานตามลำดับของแผนการขนส่งเพื่อเคลื่อนย้ายจำนวนจักรยานจากจำนวน ณ ขณะปัจจุบัน เป็นจำนวนจักรยาน ณ ที่กำหนดไว้ตามความเหมาะสม โดยปัญหาการปรับสมดุลจักรยานสามารถถูกกำหนดให้อยู่ในรูปของปัญหาเชิงการจัด เพื่อที่สามารถแก้ปัญหาเพื่อหาคำตอบจากการกำหนดโจทย์ทางคณิตศาสตร์ จากอดีตถึงปัจจุบันงานวิจัยที่ศึกษาการใช้การเรียนรู้ด้วยเครื่องเพื่อแก้ปัญหาเชิงการจัด สำหรับวิทยานิพนธ์เล่มนี้ผู้เขียนต้องการที่จะเสนอวิธีภาพรวมการเรียนรู้แบบเสริมกำลังและการค้นหาแบบทาบูเพื่อแก้ปัญหาการปรับสมดุลจักรยานแบบสถิตและทดสอบคุณภาพของคำตอบและเวลาที่ใช้ในการแก้ไขปัญหาการปรับสมดุลจักรยานให้เช่าแบบสถิต

สาขาวิชา วิศวกรรมโยธา ลายมือชื่อนิสิต .....

ปีการศึกษา 2564 ลายมือชื่อ อ.ที่ปรึกษาหลัก .....

6370422121 : MAJOR CIVIL ENGINEERING

KEYWORD: Static Bike Rebalancing Problem, Bike Sharing, Reinforcement Learning, Metaheuristic

Theethad Prohmpradith : Solving Static Bike Rebalancing Problem with Reinforcement Learning Network and Tabu Search. Advisor: Assoc. Prof. MANOJ LOHATEPANONT, Ph.D.

Bike sharing system is a zero-emission transport mode which is widely adopted at the time since people are aware of climate change. Bike sharing is a bike rental business that allows users to rent and return by themself, and its operations are the crucial feature that affects user experience. There are many bike sharing problems. For example, if bike sharing station is unable to provide docking space for returning bikes or inadequate bikes for renting, the user may feel dissatisfied. Bike rebalancing solution is a transport planning method that aims to find the best path for bike sharing service providers to relocate bikes from the initial to the final number of bikes at each station. Bike rebalancing problem is often formulated as a combinatorial optimization problem by many researchers to find the solution from its mathematical formulations. In the past, there were many research papers using machine learning for solving combinatorial optimization. In my thesis, I would like to propose reinforcement learning and tabu search overview to solve static bike rebalancing problem and test quality of solution and computational time from solving static bike rebalancing problem.

Field of Study: Civil Engineering Student's Signature .....

Academic Year: 2021 Advisor's Signature .....

กิตติกรรมประกาศ

วิทยานิพนธ์เล่มนี้จะไม่สามารถสำเร็จลุล่วงได้และสำเร็จภายในระยะเวลานี้ได้ หากปราศจากผู้ที่มีพระคุณยิ่งที่ให้กำลังใจ กำลังทุน และความคิด เพื่อประกอบสร้างมาเป็นวิทยานิพนธ์เล่มนี้

ขอขอบคุณบิดาและมารดาที่ให้กำเนิด ดูแล เลี้ยงดู ให้การศึกษาจนผู้เขียนรักในคณิตศาสตร์และวิศวกรรมศาสตร์ ขอบคุณพี่สาวที่เป็นแบบอย่างให้น้องชายในการเรียนและเป็นแรงบันดาลใจในการทำงานหนักแก่ผู้เขียน

ขอขอบคุณท่านอาจารย์ที่ปรึกษา รศ.ดร.มาโนช โลหเตปานนท์ ช่วยให้คำแนะนำการทำวิทยานิพนธ์และสั่งสอนความรู้ด้านการหาค่าดีที่สุดให้แก่ผู้เขียน

ขอขอบคุณ รศ.ดร.ศักดิ์สิทธิ์ เฉลิมพงศ์ และ รศ.ดร.สมชาย ปฐมศิริ ที่คอยให้คำปรึกษาด้านการทำวิจัย และสละเวลามาประเมิณ ให้คำชี้แนะ งานวิทยานิพนธ์

ขอขอบคุณสมาชิกผู้ทำงานในห้องปฏิบัติการขนส่งที่คอยให้กำลังใจ คำแนะนำ ช่วยตรวจเช็คร่ายงานวิทยานิพนธ์ ขอขอบคุณวิลิน ที่คอยให้กำลังใจ ให้การสนับสนุนด้านจิตใจแก่ข้าพเจ้าในช่วงเวลาที่ผ่านมา

ขอขอบคุณทุนอุดหนุนการศึกษาระดับบัณฑิตศึกษาจาก ภาควิชาวิศวกรรมโยธา ทำให้ผู้เขียนสามารถรับผิดชอบค่าใช้จ่ายในการเรียนได้ด้วยตนเอง

ขอขอบคุณพนักงานห้องทะเบียนวิศวกรรมโยธาที่ดูแลเรื่องเอกสารต่างๆให้บรรลุผลแก่ข้าพเจ้า

ผู้เขียนน้อมรับทุกความเห็นในรายงานเพื่อนำไปปรับปรุง โดยข้าพเจ้าหวังว่าจะช่วยผลักดันบริการจักรยานให้เช่าให้มีประสิทธิภาพและส่งผลให้เกิดประสบการณ์ที่ดีแก่ผู้ใช้บริการไม่มากก็น้อย

ธีร์ธัฐ พรหมประดิษฐ์

สารบัญ

หน้า

สารบัญตาราง

หน้า

สารบัญรูปภาพ

หน้า

บทที่ 1
บทนำ

1.1.ที่มา

บริการให้เช่าจักรยาน (Bike sharing) เป็นบริการที่กำลังเติบโตอย่างมากในหลายประเทศ เช่น ประเทศโซนยุโรป เอเชีย และสหรัฐอเมริกา โดยเฉพาะในเมืองที่มีการพัฒนาสภาพแวดล้อมให้เอื้ออำนวยต่อการขับขี่จักรยานอย่างต่อเนื่อง โดยจุดเริ่มต้นของบริการให้เช่าจักรยานเริ่มเติบโตมาจากการบริการให้เช่าจักรยานแบบไม่มีค่าใช้จ่าย จนกระทั่งในปัจจุบันพัฒนาเก็บค่าบริการในหลายพื้นที่ ในปัจจุบันนี้บริการให้เช่าจักรยานในหลายเมืองแข่งขันกันอย่างมาก มีผู้ให้บริการจำนวนที่สูงขึ้น จำนวนผู้เช่าที่สูงขึ้น นอกจากนี้ยังมีการปรับรูปแบบให้บริการเพื่อเพิ่มความสามารถการเข้าถึงจักรยานในหลายรูปแบบ เช่น การร่วมมือกับเจ้าของที่ดินในการติดตั้งสถานี ใช้จักรยานที่เช่าและจอด ณ ตำแหน่งใดก็ได้ ปัจจุบันบริการให้เช่าจักรยานเข้ามาเป็นอีกรูปแบบของการเดินทางสาธารณะ เช่น รถเมล์ รถไฟฟ้าใต้ดิน และ รถไฟฟ้าบนดิน ในอนาคต

Shaheen, Guzman, and Zhang (2010) อธิบายถึงระบบให้เช่าจักรยาน ถูกแบ่งออกเป็น 4 รุ่น ได้แก่โดยรุ่น 1 มีคือ ระบบจักรยานขาว (White Bikes System) รุ่นที่ 2 คือ ระบบการเช่าด้วยการฝากเหรียญ (Coin Deposit System) รุ่นที่ 3 คือ ระบบบนฐานข้อมูลสารสนเทศ (IT-based system) และรุ่นที่ 4 คือ ระบบตอบสนองต่ออุปสงค์ (Demand responsive system) โดยระบบที่พัฒนาขึ้นเพื่อทำให้การสามารถจัดการระบบให้เช่าจักรยานได้มีประสิทธิภาพมากขึ้นในด้านมีจุดประสงค์เพื่อ ป้องกันการขโมยจักรยานและความเสียหาย การกระจายจักรยาน ระบบข้อมูลแบบเรียลไทม์ (Real-time data system) ปัญหาด้านประกันและความน่าเชื่อถือ วิธีการประเมิณความคุ้มทุนก่อนเริ่มต้นโครงการให้ใช้งานจริง

Vogel and Mattfeld (2011) กล่าวว่าปัญหาที่จะพบเจอโดยทั่วไปในบริการให้เช่าจักรยานที่คือความไม่สมดุลกันของการกระจายจำนวนจักรยานปัญหานี้ส่วนมากจะถูกบรรเทาหรือแก้ไขได้ด้วยการออกแบบโครงข่ายจักรยานให้เอื้อต่อการกระจายจักรยานหรือมีจำนวนผู้เช่ากับผู้คืนใกล้เคียงกัน อีกวิธีการคือการเคลื่อนย้ายจักรยานให้ได้จำนวนที่ต้องการหรือการปรับสมดุลจักรยาน

Patrice and Florent (2018) กล่าวว่าการวิจัยเกี่ยวข้องกับระบบให้เช่าจักรยานจะเน้นไปที่งานวิจัย 2 ขอบเขตการวิจัย ขอบเขตแรกจะเกี่ยวข้องกับการออกแบบระบบในรูปแบบของจำนวน

จักรยาน จำนวนสถานี ความจุสถานี และสถานที่ตั้งของสถานี โดยใช้ผลลัพธ์จากการวิเคราะห์ข้อมูลทางประชากรศาตร์ และเศรษฐสถานะ ขอบเขตที่สองจะมุ่งเน้นไปที่การกำหนดนโยบายปรับสมดุลเพื่อป้องกันสถานการณ์จักรยานและหัวล็อคว่างสำหรับจอดจักรยานไม่เพียงพอ วิธีการแก้ปัญหาการปรับสมดุลของระบบบริการให้เช่าจักรยานแบบสถิต (Static Bike Rebalancing Problem) ด้วยการสร้างโมเดลแก้ปัญหาการปรับสมดุลที่จำแนกประเภทคือ ปัญหาการรับของและส่งของแบบชนิดเดียวและยานพาหนะชนิดเดียว (Single Commodity Single Vehicle Pickup and Delivery Problem) โดยที่ใช้ในการจัดการเคลื่อนย้ายจักรยานในตอนกลางคืนเนื่องจากไม่มีการเช่าจักรยาน

การปรับจำนวนจักรยานในแต่ละสถานี คือ นำจักรยานออก (Pickup) หรือ นำจักรยานเข้าสถานี (Delivery) เพื่อให้ตรงกับจำนวนจักรยานที่ตั้งเป้าไว้ โดยใช้ยานพาหนะหรือคนในการขนย้าย หมายความว่าจะมีขนาดความจุสำหรับขนย้ายที่หลายความจุ ซึ่งผู้เขียนคิดว่าปัญหามีลักษณะคล้ายปัญหาการรับของส่งของผู้เขียนต้องการปรับการตั้งปัญหานี้เข้าไปเพื่อวางแผนการเคลื่อนย้ายจักรยานและใช้อัลกอริทึมการแตกกิ่งและการจำกัดขอบเขตด้วยระนาบ (Branch and Cut Algorithm) เพื่อแก้ปัญหาการรับของและส่งของเพื่อลดปัญหาเวลาที่ใช้ในการคำนวณ

1.2.ลักษณะของปัญหา

ในวิทยานิพนธ์เล่มนี้ ได้กำหนดให้เป็นปัญหาการรับของและส่งของ ซึ่งหมายถึงจักรยานที่ถูกเคลื่อนย้ายให้มีจำนวนตามที่กำหนดไว้ก่อนจัดเส้นทางขนย้ายจักรยาน โดยเน้นไปที่พิจารณาจักรยานเพียงขนาดเดียว สามารถเรียกได้ว่าปัญหาการรับของและส่งของแบบชนิดเดียวด้วยยานพาหนะชนิดเดียว ยานพาหนะเริ่มต้นเดินทางออกจากโรงจัดเก็บ ส่งและรับวัตถุหรือจักรยานตามคำสั่งโดนไม่ละเมิดเงื่อนไขความจุยานพาหนะและเป็นเส้นทางการเดินทางรอบเดียว โดยฟังก์ชันวัตถุประสงค์คือต้นทุนการขนย้ายรับและส่งจักรยานที่ต่ำที่สุด สามารถที่จะสร้างแบบจำลองให้อยู่ในรูปของกำหนดการเชิงเส้นจำนวนเต็ม (Mixed-Integer Linear Programming)

1.3.ความสำคัญของปัญหา

การจัดการจักรยานสำหรับบริการให้เช่าจักรยาน เป็นหนึ่งในปัจจัยที่สำคัญเป็นอย่างมากเพื่อให้เกิดความพึงพอใจของลูกค้าในมุมของการจัดการจำนวนจักรยานและจำนวนหัวล็อคว่างให้ตรงกับความต้องการ ตัวอย่างของการจัดการที่ไม่ดีจะส่งผลเป็นปัญหาให้ผู้ใช้ดังนี้

1. จำนวนจักรยานที่ไม่เพียงพอต่อความต้องการผู้ใช้บริการจะไม่มีจักรยานใช้และส่งผลลบต่อการอยากกลับมาใช้งานในครั้งถัดไป

2. จำนวนหัวล็อคที่ไม่เพียงพอจะทำให้ผู้เช่าจักรยานไม่สามารถคืนจักรยานได้และส่งผลให้ผู้เช่าอาจจะต้องหาสถานีจอดที่อื่น ไม่สามารถจอดในสถานีที่ต้องการและเสียเวลาจากแผนกาเดินทางที่วางไว้ และ อาจจะส่งผลทำให้ต้องชำระค่าบริการในช่วงเวลาที่หาสถานีอื่นจอด

ผู้ให้บริการจึงจะต้องจัดสรรจักรยานให้ตรงกับความต้องการ ในขณะที่โครงข่ายจักรยานที่ใหญ่จะเสียค่าใช้จ่ายเป็นจำนวนมาก และการเคลื่อนย้ายจักรยานมีจำนวนรูปแบบมากขึ้นสัมพันธ์กับจำนวนสถานีให้บริการและจักรยาน การจัดทำโมเดลแก้ปัญหาการปรับสมดุลจักรยานเพื่อหาวิธีเคลื่อนย้ายจักรยานที่เหมาะสมจึงจำเป็นอย่างยิ่ง และได้ต้นทุนการดำเนินการที่ต่ำที่สุด

1.4.วัตถุประสงค์

จุดประสงค์ของการทำวิจัยคือทดลองหาวิธีการแก้ปัญหาการปรับสมดุลให้เช่าจักรยานให้ได้ประสิทธิภาพดี จากที่กล่าวไว้ข้างต้นเป้าหมายของการสร้างการแก้ปัญหานี้ เพื่อทำให้ผู้จัดการระบบให้เช่าจักรยานหาเส้นทางที่มีประสิทธิภาพสามารถลดค่าใช้จ่ายหรือเวลาของการขนย้ายจักรยาน เป็นการเพิ่มประสิทธิภาพในการดำเนินงานบริการเช่าจักรยาน ซึ่งวิธีการแก้ปัญหาดังกล่าวควรจะมีความสามารถในการประมวลที่รวดเร็วต่อปัญหาที่มีขนาดใหญ่

1.5.ขอบเขตการศึกษา

การศึกษานี้จะศึกษาการแก้ไขปัญหาการปรับสมดุลจักรยานแบบสถิตซึ่งเป็นปัญหาที่ใช้สำหรับการจัดเส้นทางในการหาเส้นทางการเดินรถเพื่อปรับสมดุลจักรยานให้เช่าเพื่อให้ได้เส้นทางที่สั้นที่สุด และไม่คำนึงถึงการใช้จักรยานในระหว่างปฏิบัติการปรับสมดุลจักรยาน

1.6.ประโยชน์ที่คาดว่าจะได้รับ

จากการพยายามแก้ไขปัญหาการปรับสมดุลจักรยานสำหรับบริการให้เช่าจักรยานคาดหวังว่าจะได้ประโยชน์ดังนี้

1. ศึกษาปัญหาการปรับสมดุลจักรยานให้เช่าของบริการให้เช่าจักรยาน และอัลกอริทึมที่ใช้แก้ปัญหา
2. นำเสนอวิธีการอื่นที่ใช้ในการแก้ไขปัญหาปรับสมดุลจักรยานให้เช่าของบริการให้เช่าจักรยาน
3. วัดประสิทธิภาพของอัลกอริทึมที่ถูกเสนอมาแก้ไขปัญหาการปรับสมดุลจักรยานให้เช่า

บทที่ 2
แนวคิดทฤษฎีและงานวิจัยที่เกี่ยวข้อง

2.1.ระบบบริการให้เช่าจักรยาน

การใช้ยานพาหนะที่ปล่อยมลพิษทางอากาศและภาวะภูมิอากาศเปลี่ยนแปลงจากอุตสาหกรรม (Climate change) ส่งผลให้เกิดความสนใจทางเลือกการขนส่งอื่นที่ยั่งยืนซึ่งคือ บริการให้เช่าจักรยาน มีการให้ให้บริการครั้งแรกตั้งแต่ปี ค.ศ. 1965 Shaheen, Guzman, and Zhang (2010) ศึกษาการเติบโตของระบบบริการใหเช่าจักรยาน (Bike-sharing system) พบว่ามีบริการให้เช่าจักรยานเติบโตทั้งหมด 4 ทวีป ได้แก่ ยุโรป อเมริกาเหนือ อเมริกาใต้ เอเชียที่รวมถึงประเทศออสเตรเลีย จากการศึกษาบริการให้เช่าจักรยานการเติบโตระบบบริการให้เช่าจักรยานแบ่งออกได้เป็น 4 ขั้น ได้แก่

1. ระบบจักรยานสีขาว (White Bikes System) เป็นระบบจักรยานให้เช่าที่ไม่คิดค่าใช้จ่าย ยกตัวอย่าง โครงการ La Rochelle ของฝรั่งเศส ในปี 1974 และ Green Bike Scheme ของสหราชอาณาจักร ปี 1993 และโครงการ vélos jaunes หรือจักรยานเหลือง ของเมืองอัมสเตอดัมเป็นโครงการที่สร้างขึ้นมาเพื่อส่งเสริมตัวชี้วัดความสำเร็จทางด้านสิ่งแวดล้อม
2. ระบบฝากเหรียญ (Coin Deposit System) จากระบบให้เช่าจักรยานฟรีทำให้เกิดปัญหาส่งผลให้รัฐบาลของเมืองปล่อยโครงการให้เช่าจักรยานที่มีขนาดใหญ่มากขึ้น ตัวอย่างเช่น ในทวีปยุโรป มกราคม 1995 โครงการไบไซเคน (Bycyken) เปิดตัวโครงการให้เช่าจักรยาน 1,100 คัน เปิดให้บริการอยู่ในเมืองโคเปนเฮเกน โดยมีค่าเช่า 20 เดนิชโครนต่อครั้ง ประมาณ 90 บาทต่อครั้ง และระบบจะคืนเงินเมื่อคืนจักรยาน อย่างไรก็ตามระบบฝากเหรียญถูกใช้แพร่หลายมากขึ้นแต่ยังพบว่าเกิดปัญหาว่ามีจักรยานที่ไม่ได้คืนเหมือนเดิม
3. ระบบบนฐานของข้อมูลสารสนเทศ (IT-based system) คือระบบจักรยานสามารถติดตามจักรยานและข้อมูลของที่ผู้ใช้บริการเพื่อช่วยป้องกันการโจรกรรมจักรยาน ผู้เช่าจักรยานต้องใช้บัตรเพื่อระบุประวัติการเช่าจักรยานและมีการจ่ายค่าปรับเมื่อไม่ทำตามกฎให้เช่าจักรยาน ทั้งนี้ค่าใช้บริการสามารถใช้กลไกดึงดูดผู้ใช้เริ่มต้นให้บริการเช่าฟรีและเก็บเงินเมื่อผู้เช่านั้นใช้ประจำได้
4. ระบบตอบสนองต่ออุปสงค์ (Demand responsive) เป็นรุ่นของการให้เช่าบริการที่ Shaheen, Guzman, and Zhang (2010) นั้นมองภาพไปยังการพัฒนาและนวัตกรรมในอนาคต ซึ่งระบบนี้

หมายถึงระบบให้เช่าจักรยานที่เชื่อมต่อเข้ากับโหมดการเดินทางชนิดอื่น (Multimodal system) มีการพัฒนาระบบกันขโมยที่ดีขึ้น มีระบบกระจายตัวของจักรยาน (Bicycle redistribution system) เชื่อมกับบัตรเงินสดขนส่งสาธารณะ(Transit smart cart) สถานีจักรยานบริการที่มีประสิทธิภาพใช้พลังงานจากแสงอาทิตย์ และจักรยานเป็นจักรยานประเภทใช้ไฟฟ้าร่วมด้วย

Abolhassani, Afghari, and Mohtashami Borzadaran (2018) ศึกษาข้อมูลเชิงลึกเกี่ยวกับองค์ประกอบที่จะทำให้เลือกระบบให้เช่าจักรยานเป็นการขนส่งหลักในประเทศกำลังพัฒนาด้วยการใช้ Mixed Multinomial Logistics model พบว่าค่าเช่าจักรยาน โครงการสอนปั่นจักรยาน และสถานการณ์จ้างงานส่งผลอย่างมีนัยสำคัญต่อระบบให้เช่าจักรยานในกรุง Mashhad ในประเทศอิหร่าน นอกจากนี้ผู้ให้สัมภาษณ์เต็มใจที่จะจ่ายค่าบริการมากกว่าเดิมเมื่อมีความปลอดภัยที่มากขึ้น เข้าถึงได้ของบริการ และความสะดวกสบาย การสร้างสิ่งอำนวยความสะดวกและการพัฒนาโครงสร้างพื้นฐานในการปั่นจักรยาน เช่น เส้นทางจักรยานแยกจากเส้นทางสำหรับรถยนต์ และความใกล้ของสถานีกับที่อยู่อาศัยหรือสถานที่สำคัญหรือป้ายรถประจำทาง ส่งเสริมให้ผู้คนอยากหันมาใช้ระบบให้เช่าจักรยานมากขึ้น งานวิจัยยังพบว่าประชากรส่วนใหญ่ที่อยู่ในภาวะว่างงานจะมีแนวโน้มไม่ใช้ระบบให้เช่าจักรยาน

Maioli, Corrêa de Carvalho, and Medeiros (2019) ศึกษาเกี่ยวกับบริบทของระบบของบริการจักรยานให้เช่าที่ส่งเสริมการใช้งานบริการให้เช่าจักรยานให้แก่ผู้จัดการระบบให้เช่าจักรยาน โดยใช้เมตริก SERVPERF ประยุกต์ในการประเมินคุณภาพของบริการ และยังบ่งชี้ให้เห็นสาเหตุที่ส่งผลต่อความพึงพอใจของลูกค้าพบว่า “ความพร้อมใช้งานระบบ (System Availability)” บ่งบอกการรับรู้ทั้งบวกและลบอย่างมาก บริษัทที่รับผิดชอบต้องรักษาระดับความพร้อมให้เช่า โดยอย่างยิ่งไม่ให้เกิดปัญหาขณะเช่าจักรยาน นอกจากนั้นใช้ผู้วิจัยใช้สร้างแบบจำลองชนิด Multiple Linear Regression เพื่ออธิบายตัวแปรที่ส่งผลกระทบสูงต่อความพึงพอใจ พบว่ามีตัวแปรดังนี้ที่มีนัยสำคัญทางสถิติที่จะส่งผลต่อความพึงพอใจ ได้แก่ ความสะดวกสบายของจักรยาน ความพร้อมใช้ของบริการให้ระบบให้เช่าจักรยานความคล่องตัวของระบบจักรยาน

2.2.การวิเคราะห์โครงข่ายจักรยาน

Yao et al. (2019) ทำการวิเคราะห์ลักษณะของระบบให้เช่าจักรยานจากระบบข้อมูลแบบเรียลไทม์ (Real-time Information system) ถูกเก็บมากจากระบบให้เช่าจักรยานสาธารณะ โดยประยุกต์การวิเคราะห์โครงข่ายที่ซับซ้อน (Complex Network Analysis Method) เพื่อวิเคราะห์หา

ความสัมพันธ์ระหว่างสถานีในระบบให้เช่าจักรยาน ด้วยการสร้างโครงข่ายเส้นทางจักรยานสาธารณะของแต่ละพื้นที่เขตเมืองในมลทลนานจิง (Nanjing) และทำการวิเคราะห์และเปรียบเทียบตัวชี้วัดต่างๆ ที่มีต่ออัตราการใช้งานบริการให้เช่าจักรยาน ผลลัพธ์ที่ได้คือ ในหลายๆสถานีมีการใช้จักรยานต่ำ มีการเช่าบริการจักรยานที่ต่างกันมากในแต่ละสถานี โดยพื้นที่ที่มีกิจกรรมทางสังคมและเศรษฐกิจมากจะมีคนใช้งานจักรยานสาธารณะเยอะมากกว่าพื้นที่ที่ไม่ใช่พื้นที่เศรษฐกิจ การใช้งานจักรยานไม่เพียงแค่สัมพันธ์กับลักษณะการใช้ที่ดินไกล้เคียง นอกจากนี้ยังสัมพันธ์กับการใช้จักรยานของสถานที่ใกล้ชิดกันด้วย มีการปั่นจักรยานทุกระยะทั้งไกลและใกล้ การปั่นจักรยานจากที่อยู่อาศัย ไปยัง สถานีรถไฟฟ้า หรือ ซุปเปอร์มาเก็ต พบได้ทั่วไป ทั้งนี้สถานีจักรยานที่อยู่บริการรถไฟฟ้าใต้ดินมีความสำคัญมาก จากการวิเคราะห์พบว่าสถานีจอดจักรยานสามารถเป็นทั้ง จุดเริ่มต้น-จุดหมายปลายทาง

2.3.การปรับสมดุลจักรยาน

การปรับสมดุลจักรยานในโครงข่ายจักรยานให้เช่าถูกนำมาเขียนให้อยู่ในรูปแบบของโมเดลทางคณิตศาสตร์ โดยมีจุดประสงค์หลักก็คือเพื่อทำให้ผู้ใช้บริการมีความสะดวกสบายมากขึ้นสามารถใช้จักรยานและคืนจักรยานได้อย่างไรปัญหา ซึ่งจำเป็นต้องใช้การจัดการที่ดีเนื่องจาก โครงข่ายจักรยานมีจำนวนจุดจอดและจักรยานที่จำนวนจำกัด ยกตัวอย่างเช่น สถานการณ์ที่จำนวนจักรยานไม่เพียงพอหรือไม่มีที่จอดมีความเป็นไปได้ที่จะเกิดขึ้นเสมอ โดยทั่วไปโมเดลจะถูกสร้างอาจจะทำให้ข้อมูลที่ป้อนมีทั้งแบบสถิต (Static) และพลวัต (Dynamic) โมเดลที่สร้างให้ป้อนข้อมูลแบบสถิตโดยใช้ข้อมูลที่คำนวณโดยใช้ข้อมูลจากอดีตและเลือกใช้ค่ากลางป้อนเข้าไปในโมเดลเพื่อให้ได้ผลลัพธ์ที่ดีที่สุดออกมา ยกตัวอย่างเช่น โมเดลที่ถูกเสนอโดย Daniel, Frédéric, and Roberto (2013); Sin and Szeto (2014); Patrice and Florent (2018) เลือกที่จะสร้างโมเดลแบบกำหนดการเชิงเส้น (Linear Programming) หรือ กำหนดการเชิงเส้นจำนวนเต็มแบบผสม (Mixed-Integer Programming) แตกต่างกันไป ขณะที่ โมเดลการปรับสมดุลจักรยานแบบพลวัต ยกตัวอย่างเช่น Legros (2019) ซึ่งใช้ กระบวนการมาคอฟ (Markov Processes) คำนวณความน่าจะเป็นและให้ผลเฉลยเป็นคำตอบว่า สถานีจักรยานไหนที่ควรให้ความสำคัญ จำนวนจักรยานที่ควรย้ายเข้าหรือออก Regue and Recker (2014) ได้สร้างโมเดลที่มีการทำนายอุปสงค์การเช่าจักรยาน เพื่อนำไปใช้ต่อในการทำนายจำนวนจักรยานที่อยู่ในคลัง (Bike Inventory) และการย้ายจักรยาน จากนั้นใช้โมเดลหาผลเฉลยเพื่อใช้การวางแผนการเดินทางเคลื่อนย้ายจักรยานที่สั้นที่สุด ในปัญหาการปรับสมดุล การสร้างโมเดลทางคณิตศาสตร์มีการตั้งฟังก์ชันเป้าหมาย (Objective function) ที่แตกต่างกันไป

ยกตัวอย่างเช่น Daniel, Frédéric, and Roberto (2013); Regue and Recker (2014); Cruz et al. (2016) เสนอโมเดลกำหนดให้ผลเฉลยหาค่าที่ดีที่สุดเพื่อย้ายจักรยานต้นทุนสำหรับดำเนินการน้อยที่สุด Patrice and Florent (2018) กำหนดผลเฉลยเกิดความต่างระหว่างจำนวนอุปสงค์การเช่ากับจำนวนจักรยานแตกต่างกันต่ำที่สุด Legros (2019) กำหนดให้ผลเฉลยเกิดอัตราที่เกิดผู้ใช้บริการที่จะรู้สึกไม่พอใจ (Rate of Arrival of Unsatisfied User) ที่เจอสถานีว่างหรือสถานีเต็มความจุ โดยสุดท้ายแล้วคำตอบที่ได้ก็จะอยู่ในรูปของการวางแผนการเดินทางที่ดีที่สุด

Claudio, Catherine, and Louis-Martin (2012) เสนอการเคลื่อนย้ายจักรยานในช่วงมีผู้ใช้บริการด้วยปัญหาแบบ Dynamic Public Bike-Sharing Balancing Problem โดยเฉพาะในช่วงชั่วโมงเร่งด่วน (Peak hour) เพื่อป้องกันไม่ให้สถานีขาดจักรยานหรือปริมาณจักรยานเต็มสถานี โดยใช้ประโยชน์ของการย่อยปัญหาแบบแดนท์ซิก-โวล์ฟ (Dantzig-Wolfe Decomposition) และอีกวิธีซึ่งคือ การย่อยปัญหาของเบนเดอร์ (Bender's Decomposition) เพื่อที่หาคำตอบของขอบเขตล่างและคำตอบที่เป็นไปได้ (Feasible Solution) ในช่วงเวลาที่เร็วกว่าแก้ไขปัญหาแบบอัลกอริทึมพื้นฐานซึ่งคือ อัลกอริทึมการแตกกิ่งและการจำกัดขอบเขต (Branch and bound)

Daniel, Frédéric, and Roberto (2013) กล่าวถึงการเคลื่อนย้ายจักรยานในรูปแบบปัญหาการรับของส่งของแบบของชนิดเดียว ยานพาหนะที่ใช้เคลื่อนย้ายเพื่อให้จักรยานมีจำนวนที่ตั้งเป้าหมายไว้ สามารถเคลื่อนย้ายจักรยานในแต่ละสถานีได้มากกว่า 1 ครั้งเพื่อเป็นบัฟเฟอร์สำหรับการมาเยือนครั้งต่อไปซึ่งเป็นจุดเด่นของอัลกอริทึมนี้ จัดอยู่ในปัญหาแบบเอ็นพี-ฮาร์ด NP-Hard สำหรับปัญหาแบบเอ็นพี-ฮาร์ดย่อมาจาก non-deterministic polynomial-time หมายความว่าเป็นปัญหาที่ไม่สามารถแก้ได้ภายในโพลีนอเมียลไทม์ (Polynomial time) หรือใช้เวลา $O(n^k)$ อัลกอริทึมการแก้ปัญหาคือ อัลกอริทึมการแตกกิ่งและการจำกัดขอบเขตแบบระนาบ(Branch and Cut) สำหรับการผ่อนปรนเงื่อนไขและหาคำตอบสำหรับขอบเขตบนด้วยทาบูเสิร์จ (Tabu search) เพื่อหาการเคลื่อนย้ายจักรยานที่เหมาะสมที่สุด

Sin and Szeto (2014) เสนอการใช้อิทเทอเรททาบูเสิร์จริสติก (Iterated Tabu Search Heuristics) เพื่อแก้ไขปัญหาการปรับสมดุลแบบสถิต เลือกสถานีที่จะเข้าไปจัดการ จัดลำดับการไปรับ/ส่งจักรยาน ระบุจำนวนจักรยานที่ต้องเก็บ/ส่งที่แต่ละสถานี โดยกำหนดเป้าหมายคือการหาเส้นทางการปรับสมดุลจักรยานและ penalties ที่เกิดขึ้นแต่ละสถานี

Regue and Recker (2014) เสนอการปรับสมดุลจักรยานขณะที่ดำเนินการอยู่ระหว่างเปิดใช้งาน ให้ชื่อเรียกว่า ปัญหาการปรับสมดุลจักรยานแบบพลวัต (Dynamic Bike Sharing

Rebalancing Problem) หาเส้นทางเดินรถที่ดีที่สุดและระดับการเก็บจักรยานเพื่อให้ระบบเช่าจักรยานสมดุลขณะเปิดใช้งาน วิธีแก้ที่นำเสนอมานั้นประกอบไปด้วย 4 แบบจำลองเพื่อหาผลเฉลยต่างๆได้แก่

1. แบบจำลองคาดการณ์จักรยานที่ระดับสถานีโดยเกรเดียนบูสต์ติง (Gradient Boosting) ซึ่งคือการเรียนรู้ของเครื่องแบบมีการสอน (Supervised Machine Learning)
2. แบบจำลองจำนวนจักรยานแต่ละสถานี ด้วยคิวแบบ M/M/1/K Process โดยที่ K คือความจุของสถานี ระยะเวลาที่ระหว่างการมาถึง และ ระยะเวลาที่ระหว่างการออกจากสถานีมีกระจายตัวแบบปัวซอง (Poisson distribution) และพารามิเตอร์ในแบบจำลองจะมาจากข้อมูลในอดีต
3. ความต้องการการกระจายจักรยาน จำนวนจักรยานที่ต้องรับและส่งในแต่ละสถานีจะได้จากกำหนดการเชิงเส้นจำนวนเต็มแบบสโตแคสติค (Stochastic Linear Integer Programming) ในกรอบเวลา เพื่อคงไว้ที่ระดับจักรยานในแต่ละสถานีที่ดีที่สุด
4. แบบจำลองการหาเส้นทางเดินรถ ด้วยฟังก์ชันอัตถประโยชน์ (Utility function) และเงื่อนไขการเดินทางในกรอบเวลาและความจุยานพาหนะ

โดยปัญหาการปรับสมดุลจักรยานแบบพลวัตนี้มีลักษณะเหมือนปัญหาการรับของและส่งของแบบชนิดเดียว (One-Commodity Pickup-and-Delivery Vehicle Routing Problem) โดยเพิ่มความซับซ้อนเกี่ยวกับการจัดเก็บจักรยานแต่ละสถานีที่ยืดหยุ่น วิธีแก้ไขคือต้องคาดการณ์ล่วงหน้าได้เพื่อเป็นแก้ไขจัดการจำนวนจักรยานก่อนที่จะเกิดเหตุการณ์ที่ทำให้ไม่พึงพอใจในการใช้จักรยาน ล่วงหน้าก่อนเกิดเหตุการณ์ที่เป็นปัญหาในอนาคต (Proactive)

Cruz et al. (2016) ปัญหาการปรับสมดุลจักรยานในแบบปัญหา One-Commodity Pickup and Delivery Vehicle Routing ที่สามารถเข้าสถานีเดิมได้ซ้ำๆ เพื่อปรับสมดุลจักรยาน เป้าหมายของปัญหาคือเพื่อหาเส้นทางที่ราคาต่ำที่สุดสามารถตอบสนองอุปสงค์ได้ทั้งหมดและไม่เกินความจุยานพาหนะ โดยใช้ Iterated Local Search เป็นอัลกอริทึมให้คำตอบแบบฮิวริสติกซึ่งผลเฉลยจะเป็นค่าที่ใกล้เคียงกับคำตอบที่ดีที่สุด

Patrice and Florent (2018) เสนอการหาจำนวนของจักรยานที่ผู้จัดการระบบจักรยานต้องนำจักรยานเข้า/ออก จากสถานีเป็นจำนวนเท่าไหร่ เพื่อให้แต่ละสถานีมีทั้งจักรยานและที่ว่างสำหรับจอดจักรยาน (Dock) โดยสร้างปัญหาให้อยู่ในรูปของกำหนดการเชิงเส้นซึ่งผลเฉลยคือจำนวนการกระจายตัวของจักรยานที่แต่ละสถานี โดยที่ตอบสนองต่ออุปสงค์ให้มากที่สุด

Legros (2019) ได้ใช้ผลลัพธ์วิธีการกระบวนการตัดสินใจมาคอฟ (Markov Decision Process) เพื่อหาความน่าจะเป็นเพื่อหานโยบายเหมาะสมต่อการจัดการบริการให้เช่าจักรยาน โดยที่ตอบโจทย์ดังต่อไปนี้

1. สถานีที่ควรให้ความสำคัญเป็นลำดับแรก
2. จำนวนจักรยานที่ต้องย้ายเข้าหรือออกจากสถานี

เป้าหมายคือลดอัตราของผู้ใช้งานที่จะเจอสถาณการณ์ที่ไม่น่าพึงพอใจ ซึ่งคือ สถานีเต็มหรือสถานีว่างเปล่า (ไม่มีจักรยานให้บริการ) นอกจากนี้หาความสัมพันธ์ของการทำงานในระบบเปรียบเทียบกับค่าใช้จ่ายเฉลี่ยและสถานะจักรยานที่สถานีต่างๆ โดยใช้ One-step Policy Improvement Method เปรียบเทียบวิธีการจัดการเพื่อเพิ่มคุณภาพการให้บริการ และนโยบายที่จะให้ความสำคัญต่อสถานีที่มีการใช้งานสูง

Francesca et al. (2019) ความต้องการของการเช่าจักรยานจากการสุ่มตามเวลาที่แตกต่างกันไป เพื่อที่จะทำให้ผู้ใช้บริการสามารถรับและส่งจักรยานได้ที่สถานี และดำเนินการย้ายจักรยานตอนช่วงกลางคืน และมีจำนวนจักรยานที่เหมาะสมที่สุดที่วันถัดไป ใช้การแก้ปัญหาชนิด Two Stage and Multistage Stochastic Optimization Model เพื่อที่จะหาจำนวนจักรที่ดีที่สุดในแต่ละสถานีที่เริ่มต้นวันถัดไป

2.4.การหาค่าที่ดีที่สุดเชิงการจัด

การหาค่าที่ดีที่สุดเชิงการจัด (Combinatorial Optimization) คือ ขั้นตอนในการค้นหาค่าที่สูงที่สุดหรือค่าที่ต่ำที่สุดของฟังก์ชันวัตถุประสงค์ (Objective Function) โดยมีโดเมนเป็นคำตอบที่ไม่ต่อเนื่อง (Discrete) ในขณะที่ขนาดพื้นที่ที่ใช้ค้นหาคำตอบที่ดีที่สุดมีขนาดใหญ่ ตัวอย่างของปัญหาการหาค่าที่ดีที่สุดเชิงการจัดสามารถยกตัวอย่างดังเช่น

1. ปัญหาการจัดเส้นทางแก่ผู้ขาย (Traveling Salesman Problem) ปัญหากำหนดโดยตำแหน่ง x และ y ของจำนวน n แต่ละเมืองหรือสถานที่ ปัญหานี้เป็นปัญหาที่ต้องการทราบเส้นทางสำหรับเดินทางไปยังทุกเมืองที่สั้นที่สุด
2. ปัญหาการบรรจุกล่อง (Bin Packing) ปัญหากำหนดจากสิ่งของจำนวน N ชิ้น โดยแต่ละชิ้นมีการกำนดขนาดโดยเฉพาะด้วย s_i จัดใส่กล่องโดยกล่องแต่ละใบมีขนาด B จำนวนกล่องที่น้อยที่สุด

3. กำหนดการเชิงเส้นจำนวนเต็มแบบผสม (Mixed Integer Programming หรือ MIP) ต้องการระบุชุดของของจำนวนเต็ม กำหนดให้ $\chi_1 \ldots \chi_N$ แทนตัวแปรในกำหนดการเชิงเส้นจำนวนเต็มภายใต้เงื่อนไขดังเช่นอสมการ (2.1)

$$a_1\chi_1 + \ldots + a_N\chi_N \leq C \tag{2.1}$$

พื้นสำหรับคำตอบที่เป็นไปได้นั้นโดยปกติแล้วจะมีขนาดใหญ่เกินไปสำหรับการค้นหาโดยใช้บรูตฟอร์ซ (Brute force) เพียงอย่างเดียว ในบางกรณี ปัญหาสามารถแก้ด้วยอัลกอริทึมการแตกกิ่งและการจำกัดขอบเขต อย่างไรก็ตามสำหรับกรณีอื่นสามารถใช้อัลกอริทึมแท้จริง (Exact Algorithm) ใช้แก้ปัญหาได้และอัลกอริทึมค้นหาแบบสุ่ม (Randomized search algorithm) จะถูกนำมาใช้งาน ยกตัวอย่างเช่น การไต่เขาแบบเริ่มต้นใหม่แบบสุ่ม (Random-Restart Hill-Climbing) ซิมูเลทแอนีลิง (Simulated annealing) อัลกอริทึมเชิงพันธุกรรม (Genetic Algorithms) และ ทาบูเสิร์จ (Tabu Search)

2.5.ปัญหาการรับของและส่งของ

Savelsbergh and Sol (1995) ให้คำอธิบายปัญหาการรับของและส่งของ (Pickup and Delivery Problem) เป็นปัญหาที่มีเงื่อนไขว่าเป็นเซ็ตของเส้นทางถูกสร้างขึ้นเพื่อตอบสนองต่อความต้องการการขนส่ง (Transportation request) โดยที่กลุ่มยานพาหนะ (fleet) สามารถวิ่งได้ในเส้นทางหลากหลายเส้นทาง แต่ละยานพาหนะมีความจุที่จำกัดต้องเคลื่อนที่จากจุดเริ่มต้นการเดินทางและจุดสุดท้ายการเดินทาง และที่แต่ละการขนสิ่งของต้องถูกขนย้ายจากจุดรับของไปยังจุดส่งของโดยที่แต่ละการขนย้ายสินค้าต้องขนจากจุดรับของไปยังจุดส่งของโดยไม่ผ่าน Transshipment สถานที่อื่นระหว่างทาง ในกรณีพิเศษของปัญหาการรับของส่งของ เช่น ปัญหาการรับของและส่งของแต่ละความต้องการการขนส่งจะระบุจุดรับสินค้าและจุดส่งสินค้าเพียงอย่างละหนึ่งจุดและ ทุกๆ ยานพาหนะเดินทางออกและกลับมายังคลังพัสดุกลาง ปัญหาการโทรและขนส่ง (The Dial-A-Ride Problem) คือปัญหาที่แทนที่สินค้าด้วยผู้เดินทาง ดังนั้น ปัญหานี้จะกล่าวถึงผู้เดินทางสำหรับทุกความต้องการการเดินทางด้วยปริมาณเท่ากับ 1 หน่วย ปัญหาการกำนหดเส้นทางยานพาหนะ (The Vehicle Routing Problem) อีกนัยหนึ่งคือ ปัญหาการรับของและส่งของ ณ จุดเริ่มต้นเดินทางและจุดสุดท้ายของทุกๆ เส้นทางตั้งอยู่ที่คลังพัสดุ

ลักษณะของปัญหาการจัดเส้นทาง (Routing Problem) นั้นแบ่งจากวิธีที่ได้รับความต้องการการขนส่ง (Transportation request) โดยมี 2 ลักษณะ ดังนี้

1. ปัญหาแบบสถิต (Static problem) คือ ปัญหาที่พิจารณาพารามิเตอร์แบบเป็นค่าคงที่ โดยคำตอบที่เป็นเส้นทางเดินทาจะไม่มีการเปลี่ยนแปลงระหว่างเดินทาง เนื่องจากไม่พิจารณาการเปลี่ยนแปลงของลักษณะหรือความต้องการในโครงข่าย เหมาะกับปัญหาที่โครงข่ายไม่มีการเปลี่ยนแปลงขณะที่ดำเนินอยู่บนเส้นทาง
2. ปัญหาแบบมีพลวัต (Dynamic problem) คือ ปัญหาที่พิจารณาความต้องการขนส่งถูกส่งเข้ามาในช่วงเวลาที่ดำเนินการขนส่งแบบเรียลไทม์ การแก้ปัญหาประเภท Dynamic problem มีลักษณะเป็นการหาคำตอบเส้นทางเดินทางใหม่จากการแก้ปัญหาย่อย (Subproblem) เมื่อมีความต้องการขนส่งใหม่เข้ามา จากนั้นจะใช้เส้นทางใหม่ในการดำเนินการขนส่ง โดยปัญหา Dynamic problem จะถูกแก้ด้วยอัลกอริทึมประเภท On-Line Algorithm

Battarra, Cordeau, and Iori ได้จำแนกปัญหาการรับของส่งที่แตกต่างออกไปโดยใช้หลักการแบ่งจะยึดตามความสัมพันธ์ของจุดรับของและจุดส่งของซึ่งแบ่งปัญหาการรับของและส่งของออกเป็น 3 ประเภท โดยที่จะสอดคล้องกับความต้องการและโครงสร้างดังนี้

1. Many-to-many (M-M) problems สินค้าแต่ละประเภทมีหลาย ๆ จุดเริ่มต้นและจุดหมายปลายทาง โดยปัญหาลักษณะนี้ ยกตัวอย่างเช่น การปรับเปลี่ยนตำแหน่งในคลังพัสดุในแต่ละร้านค้าปลีก และ บริหารจัดการระบบจักรยานให้เช่า (Bike sharing) และรถให้เช่า (Car sharing)
2. One-to-many-to-one problem สินค้าบางประเภทขนย้ายจากคลังไปยังลูกค้า และสินค้าประเภทเก็บจากลูกค้าและเก็บกลับไปที่คลังพัสดุ ตัวอย่างการใช้งานเช่น การกระจายเครื่องดื่มและเก็บกระป๋อง และขวดเปล่ากลับมา
3. One-to-one problem สินค้าแต่ละประเภทมีจุดเริ่มต้นและจุดสุดท้ายอย่างละหนึ่งจุด ยกตัวอย่างการใช้งานเช่น การขนส่งทางรถบรรทุก และขนส่งสินค้าในเมือง

โดยที่แบ่งประเภทของกรอบตัดสินใจของปัญหาออกเป็น 3 ประเภท ดังนี้

1. ปัญหาแบบสถิต (Static problem) คือ ปัญหาที่มีพารามิเตอร์ระบุมาก่อนหน้าครบถ้วนและพารามิเตอร์ไม่มีการเปลี่ยนแปลง ผลเฉลยของเส้นทางการขนส่งที่เป็นคำตอบจะไม่มีการเปลี่ยนแปลง

2. ปัญหาแบบพลวัต (Dynamic problem) คือ ปัญหาที่มีลักษณะข้อมูลได้เข้ามาในการแก่ปัญหาตามเวลาที่ผ่านไป และผลเฉลยของเส้นทางการขนส่งออกมาจะมีการเปลี่ยนแปลงเมื่อพบว่ามีข้อมูลเพิ่มเติมเข้ามาหรือพารามิเตอร์เปลี่ยนแปลง
3. ปัญหาแบบสโตแคสติก (Stochastic problem) เกี่ยวข้องกับพารามิเตอร์ที่ไม่มีความแน่นอนของพารามิเตอร์จัดอยู่ในรูปของการกระจายตัวของความน่าจะเป็น

2.6.อัลกอริทึมแบบให้คำตอบที่แน่นอนสำหรับปัญหาขนาดใหญ่

2.6.1.อัลกอริทึมการแตกกิ่งและจำกัดขอบเขต

อัลกอริทึมการแตกกิ่งและจำกัดขอบเขต (Branch and Bound) เป็นการหาคำตอบเชิงเทคนิคแบบไล่หาคำตอบ (Enumeration) ที่มีประสิทธิภาพและใช้กลยุทธ์การแบ่งแยกและเอาชนะ (Divide and Conquer Strategy) โดยกลยุทธ์ดังกล่าวหมายถึงการแก้ปัญหาโดยอาศัยการตัดเอาเซ็ตของแนวทางที่มีแนวโน้มว่าจะให้คำตอบที่ไม่ดีหรือไม่นำไปสู้ค่าที่ดีที่สุดออกไปเพื่อที่จะทำให้พิจารณาเฉพาะเซ็ตคำตอบที่ดีทำให้การค้นหาคำตอบรวดเร็วมากขึ้น อัลกอริทึมการแตกกิ่งและจำกัดขอบเขตเพื่อใช้หาคำตอบที่ดีที่สุดโดยหาคำตอบจากกำหนดการเชิงเส้น (Linear Programming; LP) ที่ทำการผ่อนปรนปัญหากำหนดการเชิงเส้นจำนวนเต็ม (Integer Programming; IP) กลายเป็นปัญหาแบบปัญหา Relaxed LP ซึ่งหมายถึงปัญหา IP ที่ตัวแปรนำเงื่อนไขการบังคับให้คำตอบเป็นจำนวนเต็มมาถูกผ่อนปรนหรือตัดเงื่อนไขดังกล่าวออกไป เพื่อให้คำตอบของตัวแปรสามารถมีค่าไม่เป็นจำนวนเต็มได้ โครงสร้างต้นไม้การแตกกิ่งและจำกัดขอบเขตจะสร้างด้วยโมเดล Relaxed LP เก็บไว้ในโหนดและไล่หาคำตอบในแต่ละโหนดในต้นไม้ จากคำตอบจาก Relaxed LP จะนำไปสู่การแตกกิ่ง (Branching) ซึ่งเป็นการกำหนดเงื่อนไขให้ตัวแปรที่ได้คำตอบเป็นเศษส่วนนั้นมีค่าเท่ากับเพียงแค่ 0 หรือ 1 หรือเป็นจำนวนเต็ม และจากการไล่แก้ปัญหา Relaxed LP จะทำให้ได้มาซึ่งเงื่อนไขที่จัดให้ตัวแปรมีค่าเป็นจำนวนเต็มจะถูกเพิ่มเข้ามาในโหนดของ Relaxed LP ที่คำนวณอยู่เรื่อย ๆ เพื่อบังคับให้คำตอบเป็นจำนวนเต็มสำหรับทุกตัวแปร กระบวนการนี้เกิดขึ้นซ้ำจนกระทั่งได้คำตอบของตัวแปรที่เป็นจำนวนเต็มและได้คำตอบที่ดีที่สุด อัลกอริทึมการแตกกิ่งและจำกัดขอบเขตจะหยุดการค้นหาคำตอบที่ดีที่สุดเมื่อที่โหนดที่ไม่นำไปสู่การได้คำตอบที่ดีขึ้นอีกแล้ว หรือไม่พบโหนดให้ค้นหาอีกต่อไป

2.6.2.อัลกอริทึมการแตกกิ่งและจำกัดขอบเขตแบบระนาบ

Hernández-Pérez and Salazar-González (2004) ได้ประยุกต์วิธีการอัลกอริทึมการแตกกิ่งและจำกัดขอบเขตแบบระนาบ (Branch and Cut Algorithm) กับ ปัญหาการรับของส่งของ อธิบายว่าอัลกอริทึมเริ่มจากการทำ Branch and Bound Algorithm ดังในหัวข้อ 2.6.1 โดยที่

คำตอบได้มาจาก การหาคำตอบปัญหา Relaxed Linear Program หลังจากนั้นเพิ่มอสมการเข้าไปยังกำหนดการเชิงเส้นหรือเงื่อนไข โดยความแตกต่างระหว่าง Branch and Bound กับ Branch and Cut คือ Branch and Cut จะเพิ่มเงื่อนไขที่ให้คำตอบนั้นเป็นจำนวนเต็มและเพิ่มเงื่อนไขที่บังคับให้คำตอบนั้นเป็นไปตามที่โจทย์ต้องการ ยกตัวอย่างเช่น ปัญหา TSP (Traveling Salesman Problem) ต้องการให้เส้นทางทัวร์นั้นมีจำนวน 1 รอบและกลับมายังจุดเริ่มต้น ปัญหา TSP ต้องการเงื่อนไขที่บังคับไม่ให้เส้นทางมีมากกว่า 1 รอบการเดินทางเข้ามาเพิ่มเติมนอกเหนือจากเงื่อนไขที่บังคับให้ตัวแปรนั้นเป็นจำนวนเต็ม เป็นต้น

2.7.การเรียนรู้แบบเสริมกำลัง

การเรียนรู้แบบเสริมกำลัง (Reinforcement Learning) มีความหมายตามนิยามของ Sutton and Barto (2018) ว่าเป็นการเรียนรู้เพื่อที่จะทราบได้ว่าควรทำอะไร โดยเป็นการเชื่อมโยงระหว่างเหตุการณ์ที่เคยพบเจอกับการกระทำเพื่อที่จะได้มาซึ่งผลลัพธ์ที่ดีที่สุด สำหรับผู้เรียนรู้จะไม่ได้ถูกสั่งให้กระทำการใดๆ แต่จะให้ลองค้นหาและเรียนรู้เองว่าจะต้องกระทำการใดเพื่อให้ได้ผลลัพธ์ที่ดีที่สุดจากการลองทำซ้ำๆ การเรียนรู้แบบเสริมกำลังแตกต่างจากการเรียนรู้แบบมีผู้สอน (Supervised Learning) โดยที่การเรียนรู้แบบมีผู้สอนคือการเรียนรู้จากชุดข้อมูลที่ใช้การฝึกฝนโดยมีการจัดหาชุดความรู้ที่เป็นคำตอบแบบกำหนดแล้ว (Label data) และการเรียนรู้แบบเสริมกำลังก็แตกต่างกับการเรียนรู้แบบไม่มีผู้สอน (Unsupervised Learning) เช่นกัน โดยที่การเรียนรู้แบบไม่มีผู้สอนจะเป็นการหาโครงสร้างที่ซ่อนอยู่ในชุดข้อมูลที่ไร้ข้อมูลแบบกำหนดคำตอบ (Unlabeled data) เพียงอย่างเดียว แต่ไม่ได้ต้องการที่จะคำตอบที่ได้ผลลัพธ์ที่ดีที่สุดแต่อย่างให้ได้มาแต่อย่างใดเลย

รูปที่ 1 ภาพรวมการเรียนรู้แบบเสริมกำลัง

แหล่งที่มา Megajuice (2017)

จากรูปที่ 1 เอเจนท์ (Agent) ในภาพรวมการเรียนรู้แบบเสริมกำลังต้องการค้นหา (Exploration) แนวทางการแก้ปัญหาใหม่ๆและการใช้ประโยชน์ (Exploitation) เพื่อบรรลุผลลัพธ์ (Reward) ที่ดีขึ้น จากการลองผิดลองถูกหรือฝึกฝนในอดีต จากประสบการณ์ที่ได้จากการค้นหา ภายในสิ่งแวดล้อม (Environment) เพื่อเลือกที่กระทำ (Action) บางอย่าง ในทางเลือกที่เป็นไปได้เพื่อที่จะให้ได้มาซึ่งผลลัพธ์ที่ดีขึ้น กลับมา เอเจนท์ที่เรียนรู้แบบเสริมกำลังต้องเรียนรู้เน้นไปที่การกระทำที่เคยลองในอดีต และพบว่าการกระทำแบบดังกล่าวได้ผลลัพธ์มากๆ แต่การค้นพบการกระทำดังกล่าว จะต้องสามารถเลือกการกระทำที่ไม่ได้เคยถูกเลือกมาก่อนด้วยเพื่อค้นหาพื้นที่ของวิธีการอื่น เอเจนท์ต้องใช้ประโยชน์ (Exploit) จากประสบการณ์เพื่อที่จะได้ผลลัพธ์กลับมาที่ดี แต่ก็ต้องค้นหาหรือลองวิธีการอื่น (Explore) เพื่อที่จะสามารถเลือกการกระทำที่ดีกว่าจากประสบการณ์ได้

นโยบาย (Policy) ภายในการเรียนรู้แบบเสริมกำลังใช้ในเอเจนท์ใช้สำหรับหรับกำหนดให้เอเจนท์กระทำจากสถานะ (State) ของเอเจนท์ ณ ขณะนั้น โดยวิธีการโพลิซีเกรเดียนท์ (Policy Gradient) คือวิธีการหนึ่งที่จะเรียนรู้นโยบายจากการปรับตัวแปร (Parametrized Policy) ใช้เลือกการกระทำโดยการเรียนรู้จะเกิดการปรับตัวแปรในนโยบาย กำหนดให้พารามิเตอร์ $\boldsymbol{\theta}$ โดยที่ $\boldsymbol{\theta} \in \mathbb{R}^{d'}$ เป็นพารามิเตอร์ในฟังก์ชันนโยบาย (Policy Function) กำหนดให้ตัวแปร A_t แทนการกระทำ (action) ณ เวลา t และตัวแปร S_t แทนสถานะ (state) ณ เวลา t และ $\boldsymbol{\pi}$ แทน Policy function ที่อยู่ในรูปของความน่าจะเป็น สามารถเขียน Policy function ดังสมการ (2.2)

$$\pi(a|s,\theta) = Pr(A_t = a|S_t = s, \theta_t = \theta) \tag{2.2}$$

จากสมการด้านบน Policy function คือค่าความน่าจะเป็นที่จะทำการกระทำ a ณ เวลา t ในสภาะแวดล้อมมีสภาพ s ที่เวลา t ด้วยพารามิเตอร์ θ และฟังก์ชันประมาณค่า (Value function) จะมีพารามิเตอร์ $\boldsymbol{w}$ ซึ่ง $\boldsymbol{w} \in \mathbb{R}^{d'}$ และตัวแปร S_t แทนสถานะ (state) ณ เวลา t ตัวแปร G_t แทนค่าคาดการณ์ของ Reward signal จากสถานะที่ S_t มีค่าเป็น s โดย value function สามารถหาค่าได้ดังสมการ (2.3)

$$\hat{v}_\pi(s,w) = \mathbb{E}_\pi[G_t|S_t = s] \qquad ,for\ all\ s \in S \tag{2.3}$$

การเรียนรู้ด้วยวิธีการโพลิซีเกรเดียนท์ (Policy Gradient Method) จะใช้ค่าเกรเดียนท์ของเวกเตอร์จากฟังก์ชัน $\mathcal{J}(\boldsymbol{\theta})$ เพื่อปรับพารามิเตอร์ภายในนโยบาย เพื่อให้ได้ค่าผลลัพธ์ของฟังก์ชัน $\mathcal{J}(\boldsymbol{\theta})$ ที่ดี

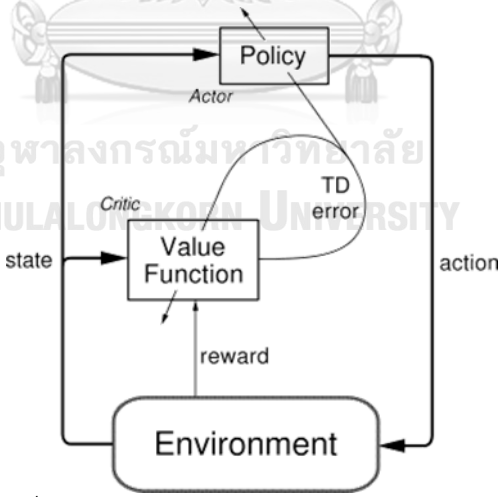
ที่สุดหรือต่ำที่สุดโดยจะอัพเดทโดยกำหนดค่าให้พารามิเตอร์ดังนี้โดยที่ $\widehat{\nabla J(\boldsymbol{\theta})} \in \mathbb{R}^{d'}$ คือเกรเดียนท์ขององสเกลาร์ที่ใช้วัดประสิทธิภาพของ Policy function เพื่อนำมาปรับพารามิเตอร์ในการฝึกฝนครั้งต่อไป $t+1$ โดยวิธีการปรับพารามิเตอร์ θ_{t+1} โดยสมการ (2.4)

$$\theta_{t+1} = \theta_t + \alpha\widehat{\nabla J(\theta)} \tag{2.4}$$

โดยพารามิเตอร์ θ จะปรับดังสมการจนกระทั้งครบจำนวนรอบในการฝึกฝนจนกระทั่งครบตามจำนวนที่กำหนดไว้

2.7.1.Actor Critic Method

วิธีการแอคเตอร์-คริติค (Actor Critic Methods) คือหนึ่งในรูปแบบวิธีการเรียนรู้แบบเสริมกำลัง เป็นวิธีที่มีโครงสร้างหน่วยความจำแยกต่างหากเพื่อแสดงให้เห็นว่าฟังก์ชันนโยบาย (Policy function) แยกกับฟังก์ชันประมาณค่า (Value function) โดยโครงสร้างของฟังก์ชันนโยบายจะถูกเรียกในชื่อ แอคเตอร์ (Actor) เพราะถูกใช้เพื่อเลือกการกระทำ (Action) ต่อสภาพแวดล้อม ส่วนฟังก์ชันประมาณค่าจะถูกเรียกในชื่อของ คริติค (Critic) เพราะจะทำหน้าที่วิจารณ์การกระทำที่ถูกเลือกโดยแอคเตอร์ ทิศทางการเรียนรู้ของวิธีการแอคเตอร์คริติคจะขึ้นอยู่กับ



รูปที่ 2 สถาปัตยกรรมของวิธีการแอคเตอร์-คริติค
แหล่งที่มา Sutton and Barto (2018)

ตัวแปร δ_t แทนค่าฟังก์ชันของการสูญเสีย (Loss function) จากการ Reward signal ของ Actor network และค่าจาก Value function ของ Critic network ตัวแปร G_{t+1} แทนค่าของ Reward

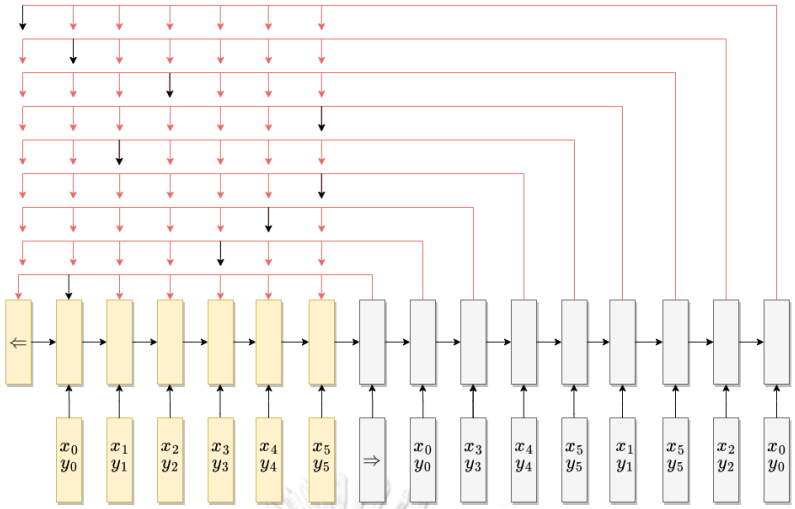
signal หรือรางวัลจากการกระทำ ณ ครั้งที่ $t+1$ และ $V(s_t)$ แทนค่าประมาณ Reward จาก Value function ณ ครั้งที่ t โดยหาค่า Loss function ดังสมการ (2.5)

$$\delta_t = G_{t+1} - V(s_t) \tag{2.5}$$

จากสมการ Loss function ข้างต้นจะใช้สำหรับการของแอคเตอร์คริติคในลักษณะของเป็น Episodic algorithm ซึ่งหมายความว่าจะมีการปรับพารามิเตอร์จากการเรียนรู้เมื่อจบเอพิโสดหรือผ่านการลองผิดลองถูกจนไม่สามารถเล่นต่อได้เป็นที่เรียบร้อย ฟังก์ชันประมาณค่าจะทำหน้าที่ประเมิณว่าการกระทำนั้นเป็นการกระทำที่ดีหรือไม่โดยหากค่าความสูญเสีย δ_t มากกว่า 0 หมายความว่าการกระทำดังกล่าวจะเป็นการกระทำที่ดี ในทางตรงกันข้ามหาก δ_t น้อยกว่าหรือเท่ากับ 0 จะหมายความว่าการกระทำดังกล่าววเป็นการกระทำที่ไม่ดี ซึ่งพารามิเตอร์จะปรับเพื่อให้เกิดการกระทำที่ส่งผลดีที่ทำให้ Reward ไปในทิศทางที่ดีขึ้นขณะฝึกฝน

2.8.โครงข่ายตัวบ่งชี้

โครงข่ายตัวบ่งชี้ (Pointer Network) เป็นสถาปัตยกรรมประสาทที่นำเสนอโดย Vinyals, Fortunato, and Jaitly (2015)เพื่อจัดการกับปัญหาที่ความยาวของของเอาต์พุตไม่แน่นอนโดยใช้ SoftMax Probability Distribution แทนตัวบ่งชี้ (Pointer) โดยโครงข่ายตัวการบ่งชี้เรียนรู้เพื่อแก้ปัญหาเรขาคณิตกับแก้ปัญหาแบบประมาณการจัดเส้นทางผู้ขาย (Traveling Salesman Problem) ที่มีขนาดเล็ก (n<50) โดย Pointer network ประกอบด้วย Decoder layer และ Encoder layer



รูปที่ 3 ตัวอย่างโครงข่ายตัวบ่งชี้ (Pointer Network)

โดยสถาปัตยกรรมของ Pointer Network จะเป็นการปรับค่าของพารามิเตอร์ใน Attention model เพื่อที่สามารถใช้ในการแก้ไขปัญหาการหาค่าที่ดีที่สุดเชิงการจัด (Combinatorial Optimization) โดยของคำตอบจะขึ้นอยู่กับขนาดของลำดับของข้อมูลอินพุต (Input sequence)

สถาปัตยกรรมของ Pointer Network พัฒนามาจากโมเดล Sequence-to-sequence เป็นโมเดลที่คำนวณจากจากอินพุต $(\mathcal{P}, \mathcal{C}^{\mathcal{P}})$ จนกระทั้ง Decoder layer สามารถเสร็จสิ้นการหาคำตอบ โดยอาศัยการตั้งค่าสัญญาณเพื่อใช้การหยุด Decoder ครั้งถัดไปโดยอธิบายความหมายของอินพุตได้ดังนี้

- ตัวแปร $\mathcal{P}$ แทนข้อมูลของลำดับ เช่น ข้อมูลพิกัดของแต่ละสถานีทั้งหมดในโครงข่ายที่ถูก มาและเก็บไว้ใน Encoder layer
- ตัวแปร $\mathcal{C}^{\mathcal{P}}$ แทนข้อมูลของลำดับข้อมูลที่ถูกเลือกมาจากข้อมูลจาก Encoder layer และเป็นชุดคำตอบล่าสุดที่ยังไม่เสร็จสิ้นบน Decoder layer

โดยที่ $\mathcal{P} = \{P_1, \ldots, P_n\}$ เป็นลำดับของ n เวกเตอร์ และ $C_P = \{C_1, \ldots., C_{m(P)}\}$ เป็นลำดับของ $m(P)$ โดย $m(P)$ มีค่าระหว่าง 1 ถึง n เป็นการเลือกลำดับที่ของอินพุตที่เข้ามาในชั้นของ Encoder layer $p(\mathcal{P}, \mathcal{C}^{\mathcal{P}}; \theta)$ คือฟังก์ชันที่คำนวณความน่าจะเป็นในการเลือกจุดยอดจาก Encoder layer เพื่อประมาณค่าของความน่าจะเป็นแบบกฎลูกโซ่เพื่อใช้หาจุดยอดที่จะเยือนเป็นครั้งต่อไปดังที่แสดงในสมการ (2.6)

$$p(\mathcal{P}, \mathcal{C}^{\mathcal{P}}; \theta) = \prod_{i=1}^{m(\mathcal{P})} p_\theta(C_1, \ldots C_{i-1}, \mathcal{P}; \theta) \tag{2.6}$$

พารามิเตอร์ในโมเดลนี้จะถูกปรับเพื่อให้ค่าความน่าจะเป็นแบบมีเงื่อนไขสูงที่สุดจากข้อมูลที่ใช้ในการเรียนรู้ ในกรณีของ Pointer network จะปรับ $p(C_1, \dots C_{i-1}, \mathcal{P})$ โดยใช้ attention mechanism ดังที่แสดงในสมการ (2.7) และ (2.8)

$$u_j^i = v^T \tanh\left(w_1 e_j + w_2 d_i\right) \qquad j \in (1, \dots, n) \tag{2.7}$$

$$p(C_1, \dots C_{i-1}, \mathcal{P}) = softmax(u^i) \tag{2.8}$$

โดยที่ $softmax$ จะปรับเวกเตอร์ u_i ที่มีความยาว n เป็นการกระจายตัวของความน่าจะเป็นของค่าในเวคเตอร์ u_i ที่จะใช้ $softmax$ เลือกเวคเตอร์ที่จะเป็นเอาต์พุต สำหรับพารามิเตอร์ v W_1 W_2 จะเป็นพารามิเตอร์ที่ใช้ปรับเพื่อเรียนรู้เพื่อให้ได้ผลลัพธ์ที่ถูกต้องหรือได้ประสิทธิภาพมากขึ้น

2.9.การใช้การเรียนรู้แบบเสริมกำลังในการแก้ไขปัญหาการหาค่าที่ดีที่สุดเชิงการจัด

Bello et al. (2016) นำเสนอวิธีการที่ใช้ในการแก้ปัญหาการหาค่าที่ดีที่สุดเชิงการจัดโดยใช้ Pointer network ที่อธิบายหัวข้อ 2.8 และการเรียนรู้แบบเสริมกำลังผสมเข้าด้วยกันโดยใช้แก้ปัญหาที่มีลักษณะเป็นการค้นหาพื้นที่ของการเรียงสับเปลี่ยนเพื่อที่จะหาค่าที่ดีที่สุดเช่น ลำดับของจุดยอดที่ดีที่สุดที่ได้ผลรวมของน้ำหนักเส้นเชื่อมต่ำที่สุดหรือเส้นทางสั้นที่สุดดังที่แสดงในสมการ (2.9)

$$L(\pi \mid s) = \left\|x_{\pi(n)} - x_{\pi(1)}\right\|_2 + \sum_{i=1}^{n-1} \left\|x_{\pi(i)} - x_{\pi(i+1)}\right\|_2 \tag{2.9}$$

ขณะที่ $\|\cdot\|$ หมายถึง ℓ_2 นอร์ม โดยความน่าจะเป็นของแต่ละการไปเยือนที่จุดยอดลำดับที่ i ของขั้นในลำดับถัดไปของเส้นทาง แทน $\mathrm{p}(\pi \mid \mathrm{s})$ ความน่าจะเป็นที่จะสร้างลำดับเส้นทาง π โดยมีเงื่อนไขข้อมูลนำเข้าเป็นลำดับจุดยอด s คำนวณจาก Product ของ $\mathrm{p}\left(\pi(\mathrm{i}) \mid \pi(<\mathrm{i}), \mathrm{s}\right)$ โดย i เท่ากับ 1 ถึง n ดังที่แสดงในสมการ (2.10)

$$p(\pi \mid s) = \prod_{i=1}^{n} p\left(\pi(i) \mid \pi(< i), s\right) \tag{2.10}$$

และมีกลไกที่ใช้ในการเลือกจุดที่จะไปเยือนถัดไปด้วยกลไกการชี้ (Pointing mechanism) กลไกการเข้าร่วม (Attending mechanism) และมีการทำงานดังนี้

1. Pointing mechanism ที่ Decoder layer ณ จุดยอดลำดับที่ j จะกำหนดค่าของความน่าจะเป็นที่จะเลือกไปเป็นจุดถัดไป $\pi(j)$ ของทัวร์ดังกล่าวดังที่แสดงในสมการ (2.11)

$$p(\pi(j)|\pi(<j), s) \stackrel{\text{def}}{=} A(enc_{1:n}, dec_j) \tag{2.11}$$

2. Attending mechanism เป็นการใช้ กริมปส์ฟังชัน (Glimpse function) $G(ref, q)$ ใช้ข้อมูลนำที่ซ้ำเดิมเป็นฟังก์ชันเพื่อใช้ในการ Glimpse function G เป็นฟังก์ชันที่คำนวณเวกเตอร์เชิงเส้นรวมกันของเวกเตอร์ ref กับความน่าจะเป็นของการเข้าร่วม

สำหรับการเรียนรู้ที่อธิบายในข้างต้นจะเป็นการเรียนรู้เสริมกำลังฐานบนนโยบายและปราศจากโมเดล (Model-free Policy-based Reinforcement Learning) ซึ่งหมายความว่าการเรียนรู้เสริมกำลังนี้จะเป็นการเรียนรู้ที่เอเจนต์จะไม่ทราบภาพรวมหรือโครงสร้างของสภาพแวดล้อมที่กำลังเผชิญหรือกำลังเรียนรู้อยู่และใช้ฟังชันนโยบายในการกำหนดการกระทำของเอเจนต์ โดยฟังก์ชันวัตุประสงค์ของการฝึกฝน (Training Objective) จะคือค่าคาดหมายของความยาวเส้นทางทัวร์ $L(\pi \mid s)$ จากอินพุตเป็นกราฟ s ถูกกำหนดดังที่แสดงในสมการ (2.12)

$$\mathcal{J}(\theta|s) = \mathbb{E}_{\pi \sim p_\theta(.|s)} L(\pi|s) \tag{2.12}$$

โดยโพลิซีเกรเดียนท์และสโตคาสติคเกรเดียนท์เดสเซนท์เพื่อที่จะหาค่าพารามิเตอร์ที่ดีที่สุดแก่ฟังก์ชันวัตถุประสงค์สำหรับโครงข่ายแอคเตอร์ของการฝึกฝนโดยอัลกอริทึม REINFORCE เสนอโดย Williams (1992) ฟังก์ชันนโยบายจะปรับค่าพารามิเตอร์จาก $\nabla_\theta \mathcal{J}(\theta|s)$ ดังที่แสดงในสมการ (2.13)

$$\nabla_\theta \mathcal{J}(\theta|s) = \mathbb{E}_{\pi \sim p_\theta(.|s)}[(L(\pi|s) - b(s)) \nabla_\theta \, log \, p_\theta(\pi|s)] \tag{2.13}$$

โดย $\mathrm{b(s)}$ คือ ฟังก์ชันเบสไลน์ (Baseline function) ที่ไม่เกี่ยวกับฟังก์ชันของโพลิซี π เพื่อประมาณค่าความยาวของเส้นทางทัวร์ที่คาดหวังเพื่อลดค่าความแปรปรวน (variance) ฟังก์ชันประมาณการณ์ค่าจะปรับพารามิเตอร์ด้วยอง $\nabla_{\theta_v} \mathcal{L}(\theta_v)$ ดังที่แสดงในสมการ (2.14)

$$\mathcal{L}(\theta_v) = \mathbb{E}_{\pi \sim p_\theta(.|s)} \left\| b_{\theta_v}(s_i) - L(\pi|s) \right\|_2^2 \tag{2.14}$$

โครงข่ายออกซิลาลี (Auxiliary network) หรือ คริติค (Critic) คือโครงข่ายประสาทที่ประกอบด้วยพารามิเตอร์ θ_v เพื่อเรียนรู้และประมาณการณ์ระยะเส้นทางรวมที่คาดหวังจากนโยบาย p_θ จากลำดับอินพุต s

2.10.ปัญหาการปรับสมดลุจักรยานแบบสถิต

ปัญหาปรับสมดุลจักรยานแบบสถิต (Static Rebalancing Problem) เป็นปัญหาที่สร้างขึ้นมาเพื่อที่จะหาคำตอบว่าต้องเข้าไปรับจักรยานส่งจักรยานที่สถานีเป็นลำดับอย่างไรให้ได้ต้นทุนการขนส่งต่ำที่สุดและต้องขนจักรยานข้ามแต่ละสถานีในลำดับเป็นจำนวนเท่าใด โดย Daniel, Frédéric, and Roberto (2013) ได้เสนอการกำหนดปัญหาเพื่อหาเส้นทางที่ดีที่สุดเพื่อย้ายรถจักรยานจากจำนวนจักรยานของแต่ละสถานีในปัจจุบันให้มีจำนวนไปยังเป็นจำนวนเท่าใดดังตัวอย่างโครงข่ายด้านล่างประกอบด้วยจุดยอดของสถานีจักรยานโดยแต่ละจุดยอดจะมีคู่อันดับ (p_i, q_i) โดย p_i คือ จำนวนจักรยาน ณ เวลาปัจจุบันหรือเรียกว่าสถานะเริ่มต้น (initial state) และ q_i คือ จำนวนจักรยานที่ต้องการให้สถานี i มีหรือเรียกว่าสถานะสิ้นสุด (final state) นอกจากนี้จะมีจุดยอดสี่เหลี่ยมหรือจุดยอดคลังเก็บจักรยาน (Depot)

รูปที่ 4 ตัวอย่างของการกำหนดปัญหาประสมดุลจักรยาน

กำหนดให้ทุกซับเซท $S \subseteq V$:

1. $\bar{S} = V \setminus S$
2. $\delta^+(s) \coloneqq \{(i,j) \in A : i \in S ; j \in \bar{S}\}$
3. $\delta^-(s) \coloneqq \{(i,j) \in A : i \in \bar{S}; j \in S\}$

4. $\delta(0) \coloneqq \delta^{+}(0) \cup \delta^{-}(0)$
5. $d(S) = \sum_{j \in S} d_j$
6. $\mu(S)$ เท่ากับ 1 เมื่อมีอย่างน้อยจุดยอดไม่สมดุลในเซ็ตจุดยอด S แต่ในทางตรงกันข้ามในเซ็ตจุดยอด S โดย $\mu(S)$ มีค่าเท่ากับ 0

กำหนดการเชิงเส้นจำนวนเต็ม ประกอบไปด้วย $z_{(i,j)}$ เพียงตัวแปรชนิดเดียวโดยมีฟังก์ชันวัตถุประสงค์และข้อกัดต่างๆดังดังที่แสดงในสมการและอสมการ (2.15)-(2.20)

$$z = min \sum_{(i,j) \in A} c_{(i,j)} z_{(i,j)} \tag{2.15}$$

s.t.

$$\sum_{j \in V} z_{(i,j)} = \sum_{j \in V} z_{(j,i)} \qquad \forall i \in V \tag{2.16}$$

$$\sum_{i \in V\{0\}} z_{(0,i)} = 1 \tag{2.17}$$

$$\sum_{(i,j) \in \delta^{+}(S)} z_{(i,j)} \geq \mu(S) \quad \forall S \subseteq V\{0\} \tag{2.18}$$

$$\sum_{(i,j) \in \delta^{+}(S) \setminus \delta(0)} z_{(i,j)} \geq \left\lceil \frac{d(S)}{Q} \right\rceil \quad \forall S \subseteq V \tag{2.19}$$

$$z_{(i,j)} \in \mathbb{Z}_{+} \qquad \forall (i,j) \in A \tag{2.20}$$

จากกำหนดการเชิงเส้นข้างต้น V แทนเซ็ตของจุดยอดทั้งหมดที่พิจารณาในปัญหาปรับสมดุลจักรยาน S แทนเซ็ตของกลุ่มจุดยอดใดๆ $\delta(S)$ แทนเซ็ตของเส้นเชื่อมระหว่างเซ็ตของจุดยอดของ S กับ S’ สำหรับฟังก์ชันวัตถุประสงค์คือต้องการหาคำตอบของ $z_{(i,j)}$ ที่ทำให้ต้นทุนการเดินทางรวมต่ำที่สุด เงื่อนไขที่ (2) คือต้องการให้เส้นทางเดินทางนั้นสมดุลในแต่ละจุดยอด เงื่อนไข (3) คือต้องการให้เดินทางออกจากจุดยอด 0 เพียงครั้งเดียว เงื่อนไข (4) คือต้องการให้เส้นเชื่อมในระหว่างเซ็นของจุดยอด S กับ S' เชื่อมต่อกัน เงื่อนไข (5) คือต้องการให้ยานพาหนะเข้าไปยังเซ็ตของจุดยอด S อย่างน้อย $\left\lceil \frac{d(S)}{Q} \right\rceil$ ครั้งเพื่อให้เพียงพอต่อจำนวนจักรยานที่ต้องขนย้าย

บทที่ 3
แนวคิดในการแก้ไขปัญหา

ในบทนี้จะเสนอแนวทางวิธีการแก้ไขปัญหาปรับสมดุลจักรยานด้วยวิธีที่ถูกนำเสนอโดย Daniel, Frédéric, and Roberto (2013) ด้วย Pointer network ที่นำเสนอใน Vinyals, Fortunato, and Jaitly (2015) และการเรียนรู้ Reinforcement learning เพื่อแก้ไขปัญหาเชิงการจัดจาก Bello et al. (2016)

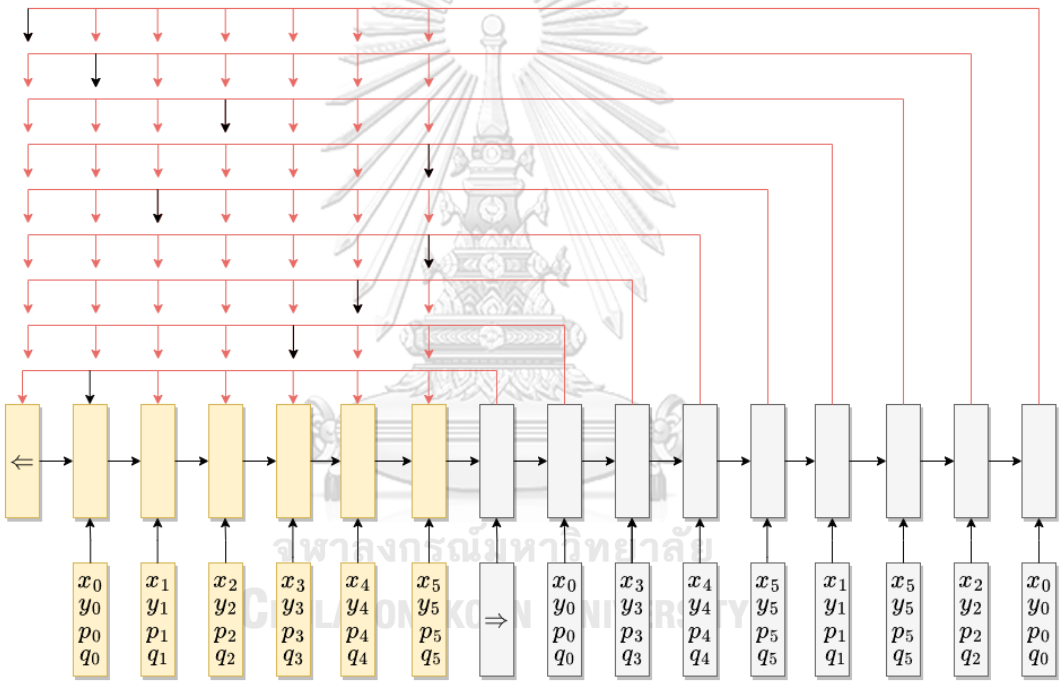
3.1.Pointer Network สำหรับแก้ปัญหาปรับสมดุลจักรยาน

ปัญหาการปรับสมดุลจักรยานแบบสถิตถือว่าเป็นโจทย์ที่เป็นการหาค่าที่ดีที่สุดเชิงการจัดและใช้เวลาในการแก้ปัญหาค่อนข้างสูง ตัวอย่างของปัญหาปรับสมดุลดุลจักรยานจะระบุตัวอย่างตามรูปที่ 5 โดยในโครงข่ายประกอบด้วยจุดยอดหรือสถานีให้บริการเช่าจักรยานและ (p_i, q_i) สำหรับ $i \in \mathcal{V}$ โดย $\mathcal{V}$ คือ เซ็ตของจุดยอด p_i คือ จำนวนจักรยานที่มีอยู่ ณ ปัจจุบัน และ q_i คือจำนวนจักนยานที่ต้องการให้มีหลังขนย้ายจักรยานดำเนินการเสร็จสิ้น และจุดยอดสี่เหลี่ยมซึ่งคือสถานีจุดเริ่มต้นของยานพาหนะในการไปรับจักรยานและขนส่งจักรยาน

รูปที่ 5 ตัวอย่างโครงข่ายของปัญหาปรับสมดุลจักรยานที่ประกอบด้วยจุดยอดของจุดเริ่มต้นและสิ้นสุดรวมถึงสถานีให้บริการในโครงข่ายบริการจักรยานให้เช่า

จากรูปแบบปัญหาการปรับสมดุลจักรยานข้างต้นที่กล่าวมาพบว่าแต่ละสถานีจะมีคุณสมบัติทางด้านตำแหน่งที่ตั้งอยู่และจำนวนของจักรยาน ณ ปัจจุบัน และจำนวนของจักรยานที่ต้องการให้มีโดยคุณสมบัติทางด้านที่ตั้ง ในรายงานนี้จะกำหนดให้เป็นในลักษณะของ 2D Euclidean space หรือเป็นพิกัดคู่อันดับ (x_i, y_i) และจำนวนจักรยานเป็นคู่อันดับ (p_i, q_i) โดย $i \in \mathcal{V}$

ในรายงานเล่มนี้เราประยุกต์คุณสมบัติของโครงข่ายจักรยานเข้ากับ Pointer Network อินพุตเป็นเวกเตอร์ของลำดับแต่ละสถานีใน $\mathcal{V}$ โดยที่แต่ละ i จะมีเวกเตอร์เท่ากับ $[x_i, y_i, p_i, q_i]$ เวกเตอร์ถูกส่งเข้าไป Encoder layer เรียงตามจำนวนสถานีที่มีในโครงข่ายจักรยานดังแสดงในรูปที่ 6



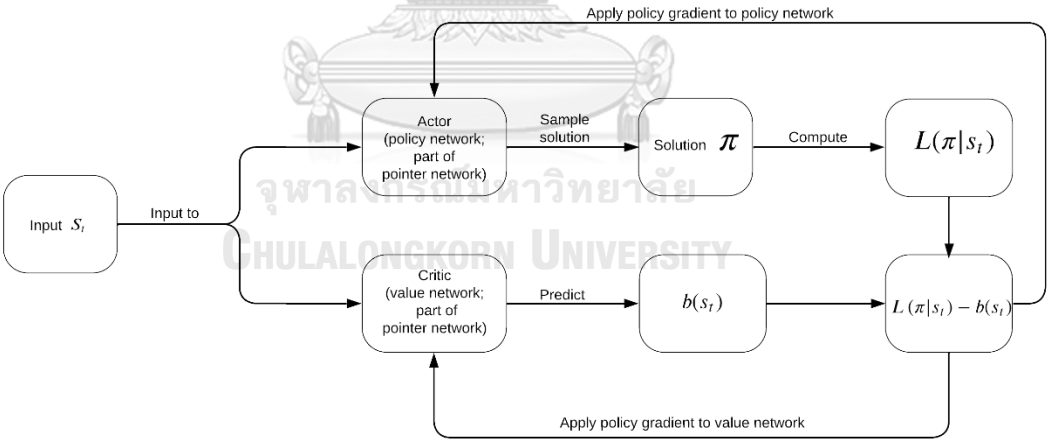
รูปที่ 6 Pointer network สำหรับปัญหาการปรับสมดุลจักรยาน

จากนั้น หลังจากที่ Encoder layer รับข้อมูลนำเข้าจากข้อมูลแต่ละสถานีทั้งหมดจะส่ง Encoded array ผ่าน cell state ของ LSTM ในใน Encoder layer และ Decoder layer ต่อไปที่ Decoder layer และจากนั้น Encoder layer จะทำหน้าที่เข้ารหัสเริ่มจากเวกเตอร์เริ่มต้น $\Rightarrow$ จากนั้นจะเริ่มต้นที่สถานีที่ Depot หรือสถานีที่ $i = 0$ จากนั้นจะส่งผลลัพธ์จาก Decoder layer ไปยัง Pointing mechanism เพื่อเลือกสถานีที่จะเข้าไปเยือนเป็นสถานีถัดไป จากนั้น Attending mechanism เพื่อส่งค่าไปยัง decoder layer ในครั้งถัดไปจนกระทั่งสิ้นสุดการ Decoding เมื่อเลือก

สถานี depot เป็นสถานีถัดไปจากนั้น Pointing mechanism จะเลือก Encoder ลำดับของ ⇐ เพื่อหยุดการทำงาน

3.2.การเรียนรู้แบบเสริมกำลังเพื่อแก้ปัญหาปรับสมดุลจักรยาน

จากงานวิจัย Bello et al. (2016) พบว่าการเรียนรู้แบบเสริมกำลังในการแก้ปัญหา Combinatorial optimization ด้วย Actor Critic Method ได้อย่างมีประสิทธิภาพ โดยในวิทยานิพนธ์นี้จะใช้วิธีการดังกล่าวเพื่อเป็นวิธีการในงานนี้ โดย โดยวิธีการ Actor Critic Method มีภาพรวมในการเรียนรู้ดังรูปที่ 7 โดยที่ Pointer network จะเป็นโครงข่ายที่ใช้ในคำนวณในฟังก์ชันสำหรับ Actor และ Critic หรือคำนวณค่าของ Policy function และ Value function ตามลำดับ โดยหน้าที่ของ Actor จะทำหน้าที่ทดลองสร้างคำตอบ จากนั้นนำคำตอบที่ได้มาคำนวณเป็นเส้นทางรวมและค่าของ Degree of infeasibility ของคำตอบซึ่งจะอธิบายในหัวข้อ 3.3.1 และหน้าที่ของ Critic จะประมาณค่าระยะเส้นทางจากข้อมูลนำเข้าเพื่อนำมาเปรียบเทียบกับค่าของคำตอบจาก Actor เพื่อบอกว่าเส้นทางที่ได้จาก Actor เป็นการสร้างคำตอบเส้นทางเดินทางที่ดีหรือแย่เพื่อนำมาปรับพารามิเตอร์ใน Pointer network ต่อไป



รูปที่ 7 ภาพรวมของ Actor Critic Method สำหรับแก้ไข Combinatorial Optimization

Policy gradient จะเป็นอัลกอริทึมที่ใช้ในการปรับพารามิเตอร์ใน Policy network และ Value network เพื่อเสริมให้เกิดการสร้างเส้นทางที่ได้ผลลัพธ์ดีขึ้น ซึ่ง Policy gradient จะอธิบายในหัวข้อ 3.3.2 ต่อไป

3.3.การหาค่าที่ดีที่สุดของพารามิเตอร์ด้วย Policy Gradient

โครงข่าย Pointer network จะทำหน้าที่ในการคำนวณค่าข้อมูลนำเข้าประกอบกับพารามิเตอร์ในโครงข่ายสำหรับในวิทยานิพนธ์เล่มนี้ปรับเปลี่ยน Reward signal หรือ Objective function คือค่าของระยะเส้นทางรวมกับจำนวนจักรยานที่ไม่สามารถจัดให้เป็นไปตามที่ต้องการให้เป็นได้ เพื่อให้โครงข่ายเรียนรู้ที่จะสร้างลำดับของจุดยอด ซึ่ง Reinforcement learning จะเป็นวิธีในการปรับพารามิเตอร์เพื่อทำให้ Objective function ต่ำลงหรือเป็นผลลัพธ์ดีขึ้นกว่าการฝึกฝนครั้งก่อนหน้าและจำนวนจักรยานที่จัดได้ตามไว้ที่กำหนดไว้หลังจากเดินทางครบเส้นทาง

3.3.1.การหาจำนวนจักรยานที่ขนย้าย

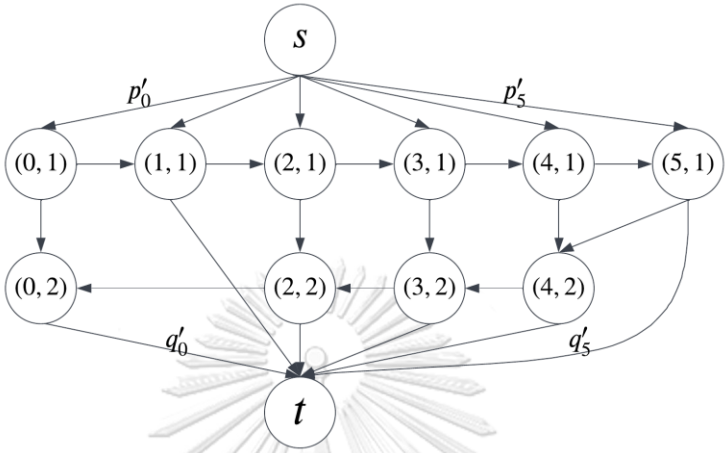
จากกำหนดการเชิงเส้นจำนวนเต็มที่ได้กล่าวไป การจะให้ผลเฉลยจะมีลักษณะเป็นลำดับของจุดยอดที่จะประกอบขึ้นเป็นเส้นทางลำดับของสถานีในการรับของและส่งของ หรือเป็นลำดับการเดินทางของยานพาหนะที่จะไปเยือนแต่ละสถานี ลำดับมีลักษณะ $i_1, i_2, \dots, i_k$ โดยเริ่มต้นที่สถานีเก็บของ $0 = i_1 = i_k$ ด้วย สามารถหาจำนวนจักรยานที่จะขนย้ายในลำดับดังกล่าวด้วยการจัดเป็นกราฟที่แสดงการขนย้ายจักรยานและหาการไหลสูงสุด maximum s-t flow หรือ การไหลระหว่างจุดจุดยอด s (source) และ t (sink) ที่มากที่สุดด้วยอัลกอริทึม เช่น เอดมอนส์-คาร์ปอัลกอริทึม ฟอร์ด-ฟูลเคอสันอัลกอริทึม ไดนิคสอัลกอริทึม เป็นต้น โดยกราฟเป็นไดเรคกราฟ (Directed graph) $D = (U, A')$ โดย U คือ เซ็ตของจุดยอดแต่ละจุดยอดแสดงเลขที่ประจำสถานีประกอบกับครั้งที่มาเยือนสถานีและมีจุดยอด s กับ t ยกตัวอย่างเช่น จุดยอด $(1,2)$ คือ จุดยอดของสถานีที่ 1 โดยเข้ามาที่เป็นครั้งที่ 2 โดยเซ็ตของเส้นเชื่อม A' มีทั้งหมด 4 ประเภทประกอบด้วย

1. เส้นเชื่อมระหว่างจุดยอด s และจุดยอดที่เป็นการไปเยือนครั้งแรกของแต่ละสถานีโดยมีความจุเส้นเชื่อมเท่ากับ p_i
2. เส้นเชื่อม (i_j, i_{j+1}) สำหรับแต่ละ $j = 1, \dots, k-1$ โดยมีความจุบนเส้นเชื่อมเท่ากับ Q ซึ่งคือความจุยานพาหนะ
3. เส้นเชื่อมระหว่าง จุดยอดที่เป็นสถานีเดียวกันครั้งที่ t กับครั้งที่ $t+1$ โดยมีความจุบนเส้นเชื่อมเท่ากับ C ซึ่งคือความจุสถานีให้บริการเช่าจักรยาน
4. เส้นเชื่อมระหว่างจุดยอดครั้งสุดท้ายของแต่ละสถานีในลำดับกับ t โดยมีความจุเส้นเชื่อม q_i

ยกตัวอย่างหากคำตอบที่ได้จาก Pointer network เป็นลำดับจุดยอดโดยแต่ละจุดยอดแทนเลขที่ของสถานี ยกตัวอย่างเช่น

$$0 \to 1 \to 2 \to 3 \to 4 \to 5 \to 4 \to 3 \to 2 \to 0$$

โดยจากลำดับดังกล่าวจะมีเซ็ตจุดยอด U ประกอบด้วย $(0,1)$ $(0,2)$ $(1,1)$ $(2,1)$ $(2,2)$ $(3,1)$ $(3,2)$ $(4,1)$ $(4,2)$ $(5,1)$ จุดยอด s และ t มีลักษณะกราฟดังรูปที่ดังด้านล่าง



รูปที่ 8 ตัวอย่างกราฟเพื่อใช้หาปริมาณการไหลหรือจำนวนจักรยานที่ขนย้าย

คำตอบการไหลสูงสุดจะได้เป็นค่าของ p_i' และ q_i' ของแต่ละจุดยอด i โดยสามารถอธิบายได้ดังนี้

1. p_i' คือจำนวนจักรยาน ณ จุดเริ่มต้นจากโครงสร้างของลำดับจุดยอดในการรับสินค้าและส่งสินค้าที่หาได้จากปัญหา *Maximum flow problem* ณ สถานีที่ i
2. q_i' คือจำนวนจักรยาน ณ เวลาสุดท้ายจากโครงสร้างของลำดับจุดยอดในการรับสินค้าและส่งสินค้าที่หาได้จากปัญหา *Maximum flow problem* ณ สถานีที่ i

ในกรณีจุดยอดหรือจุดยอดไม่ได้ปรากฏอยู่ในลำดับของคำตอบจะมีค่า p' และ q' ดังแสดงในสมการ (3.1)

$$p_i' = q_i' = min(p_i, q_i) \tag{3.1}$$

ซึ่งถ้าหาก $p_i' = p_i$ สำหรับทุกจุดยอด i คือ สถานีที่อยู่ในโครงข่ายจะแปลว่าคำตอบนั้นเป็นไปได้ (Feasible Solution) หรือสามารถย้ายให้จักรยานเป็นไปตามจำนวนที่กำหนดไว้ได้ ค่าของ $\sum_{i \in V}(|p_i - p_i'|)$ แสดงถึงจำนวนของจักรยานที่ไม่เป็นไปตามที่ต้องการให้เป็น ณ สิ้นสุดการขนย้ายจักรยาน

3.3.2 กลไกการทำงานของ Pointer Network

Bello, et al. (2017) นำเสนอแนวคิดที่ใช้ในการแก้ปัญหาการหาค่าที่ดีที่สุดเชิงการจัดโดยใช้โครงข่ายการบ่งชี้ที่อธิบายหัวข้อ 2.9 และการเรียนรู้แบบเสริมกำลังผสมเข้าด้วยกันโดยใช้แก้ปัญหาที่มีลักษณะเป็นการค้นหาพื้นที่ของการเรียงสับเปลี่ยนเพื่อที่จะหาค่าที่ดีที่สุด สำหรับในงานวิจัยนี้จะกำหนดให้ฟังก์ชันวัตถุประสงค์ (Objective function) คือ ผลรวมของน้ำหนักเส้นเชื่อมต่ำที่สุดหรือผลรวมเส้นทางสั้นที่สุดจากคำตอบของลำดับของจุดยอดหรือจุดยอดของเส้นทางกับ Degree of Infeasibility ซึ่งมีค่าเท่ากับจำนวนจักรยานที่ไม่สามารถจัดให้มีจำนวนตามที่ต้องการสามารถหาจำนวนจักรยาน หลังจากปรับสมดุลจักรยานได้จากปัญหา Maximum s-t flow ที่ได้กล่าวไว้ข้างต้นได้ดังที่แสดงในสมการ (3.2)

$$L(\pi \mid s) = \left\|x_{\pi(n)} - x_{\pi(1)}\right\|_2 + \sum_{i=1}^{n-1} \left\|x_{\pi(i)} - x_{\pi(i+1)}\right\|_2 + \sum_{i \in \mathcal{V}} (|p_i - p_i'|) \tag{3.2}$$

ขณะที่ $\|\cdot\|$ หมายถึง ℓ_2 norm โดยความน่าจะเป็นของแต่ละการไปเยือนที่แต่ละขั้นในลำดับของเส้นทางมีดังที่แสดงในสมการ (3.3)

$$p(\pi \mid s) = \prod_{i=1}^{n} p\left(\pi(i) \mid \pi(< i), s\right) \tag{3.3}$$

ซึ่งการคำนวณความน่าจะเป็นที่กล่าวข้างต้นจะอาศัยกลไกที่ใช้ในการเลือกจุดที่จะไปเยือนถัดไปด้วยกลไกการชี้ (Pointing mechanism) กลไกการเข้าร่วม (Attending mechanism) และมีการทำงานดังนี้ Pointing mechanism เป็นการคำนวณที่เกิดจากพารามิเตอร์ 3 ชนิด คือ 2 พารามิเตอร์ชนิดเมทริกซ์ $W_{ref} \in \mathbb{R}^{d \times d}$ $W_q \in \mathbb{R}^{d \times d}$ และ 1 พารามิเตอร์ชนิดเวกเตอร์ $v \in \mathbb{R}^d$ ซึ่งมีดังที่แสดงในสมการ (3.4) และ (3.5)

$$u_i = v^T \cdot tanh(W_{ref} \cdot r_i + W_q \cdot q) \qquad for\ i = 1,2,\ldots,k \tag{3.4}$$

$$A(ref, q;\ W_{ref}, W_q, v) \stackrel{\text{def}}{=} softmax(u) \tag{3.5}$$

โดยที่ Pointer Network ที่ดีโคดเดอร์ ณ ขั้นที่ j จะกำหนดค่าของความน่าจะเป็นที่จะเลือกไปเป็นจุดถัดไป $\pi(j)$ ของทัวร์ดังกล่าวดังที่แสดงในสมการ (3.6)

$$p(\pi(j)|\pi(<j), s) \stackrel{\text{def}}{=} A(enc_{1:n}, dec_j) \tag{3.6}$$

Attending mechanism เป็นการใช้ค่าของกริมปส์ฟังก์ชัน (Glimpse function) $G(ref, q)$ ใช้ข้อมูลนำที่ซ้ำเดิมนำเข้าสู่ฟังก์ชันการเข้าร่วม A ที่อธิบายไว้จากสมการข้างต้น ซึ่งในสมการดังกล่าว ตัวแปร ref จะแทนอินพุตของสถานีล่าสุดที่ถูกเลือกอยู่ในรูปของอาเรย์ $[x_i, y_i, p_i, q_i]$ และตัวแปร q คือค่าที่ถูกส่งผ่านมาจากกริมปส์ฟังก์ชัน ถูกคำนวณร่วมกับพารามิเตอร์ $W_{ref}^g \in \mathbb{R}^{d\times d}$ $W_q^g \in \mathbb{R}^{d\times d}$ และ $v^g \in \mathbb{R}^d$ แทนการคำนวณหา p ดังที่แสดงในสมการ (3.7) และ (3.8)

$$p = A(ref, q; W_{ref}^g, W_q^g, v^g) \tag{3.7}$$

$$G(ref, q; W_{ref}^g, W_q^g, v^g) \stackrel{\text{def}}{=} \sum_{i=1}^{k} r_i p_i \tag{3.8}$$

Glimpse function G เป็นฟังก์ชันที่คำนวณเวกเตอร์เชิงเส้นรวมกันของเวกเตอร์ ref กับความน่าจะเป็นของการเข้าร่วม โดยถูกประยุกต์ใช้ดังที่แสดงในสมการ (3.9) และ (3.10)

$$g_0 \stackrel{\text{def}}{=} q \tag{3.9}$$

$$g_l \stackrel{\text{def}}{=} G(ref, q_{l-1}; W_{ref}^g, W_q^g, v^g) \tag{3.10}$$

สุดท้ายแล้ว เวกเตอร์ g_l จะถูกนำไปใช้ในแอนเทนชันฟังก์ชัน $A(ref, q; W_{ref}, W_q, v)$ เพื่อคำนวนความน่าจะเป็นของกลไกการชี้ออกมาอีกใน ณ ขั้นถัดไป

3.3.3.วิธีการเรียนรู้ด้วย Policy Gradient

สำหรับการเรียนรู้ของที่เสนอโดย Bello I., et al (2013) จะเป็นการเรียนรู้เสริมกำลังฐานบนนโยบายและปราศจากโมเดล (Model-free Policy-based Reinforcement Learning) ซึ่งหมายความว่าการเรียนรู้เสริมกำลังนี้จะเป็นการเรียนรู้ที่เอเจนต์จะไม่ทราบภาพรวมหรือโครงสร้างของสภาพแวดล้อมที่กำลังเผชิญหรือกำลังเรียนรู้อยู่และใช้ฟังชันนโยบายในการกำหนดการกระทำของเอเจนต์ โดยฟังก์ชันวัตุประสงค์ของการฝึกฝน (Training Objective) จะคือค่าคาดหมายของความยาวเส้นทางทัวร์ $L(\pi \mid s)$ จากอินพุตเป็นกราฟ s ถูกกำหนดดังที่แสดงในสมการ (3.11)

$$\mathcal{J}(\theta|s) = \mathbb{E}_{\pi\sim p_\theta(.|s)} L(\pi|s) \tag{3.11}$$

โดยโพลิซีเกรเดียนท์และสโตคาสติคเกรเดียนท์เดสเซนท์เพื่อที่จะหาค่าพารามิเตอร์ที่ดีที่สุดแก่ฟังก์ชันวัตถุประสงค์สำหรับโครงข่ายแอคเตอร์ของการฝึกฝนโดยอัลกอริทึม REINFORCE เสนอโดย Williams (1992) ดังแสดงในสมการ (3.12)

$$\nabla_\theta \mathcal{J}(\theta|s) = \mathbb{E}_{\pi\sim p_\theta(.|s)}[(L(\pi|s) - b(s))\nabla_\theta \, log \, p_\theta(\pi|s)] \tag{3.12}$$

โดย $\mathrm{b}(s)$ คือ ฟังก์ชันเบสไลน์ (Baseline function) ที่ไม่เกี่ยวกับฟังก์ชันของโพลิซี π เพื่อประมาณค่าความยาวของเส้นทางทัวร์ที่คาดหวังเพื่อลดค่าความแปรปรวน (variance) โดยฟังก์ชันนโยบายจะปรับค่าพารามิเตอร์จาก $\nabla_\theta \mathcal{J}(\theta|s)$ ดังที่แสดงในสมการ (3.13)

$$\mathcal{L}(\theta_v) = \mathbb{E}_{\pi\sim p_\theta(.|s)} \left\| b_{\theta_v}(s_i) - L(\pi|s) \right\|_2^2 \tag{3.13}$$

โครงข่ายออกซิลาลี (Auxiliary network) หรือ คริติค (Critic) ที่ประกอบด้วยพารามิเตอร์ด้วย θ_v เพื่อเรียนรู้และประมาณการณ์ระยะเส้นทางรวมที่คาดหวังจากนโยบาย p_θ จากลำดับอินพุต s โดยฟังก์ชันประมาณการณ์ค่าจะปรับพารามิเตอร์ด้วยอง $\nabla_{\theta_v}\mathcal{L}(\theta_v)$ โดยการเรียนรู้แบบ Actor critic method จะทำตามรหัสขั้นตอนดังรูปที่ 9

Algorithm 1 Actor-critic training

procedure TRAIN (training set S, number of training steps T)
    Initialize pointer network params θ
    Initialize critic network params θ_v
    for t=1 to T **do**
        $s \sim \text{SampleInput}(S)$
        $\pi \sim \text{SampleSolution}\big(p_\theta(.|s_i)\big)$
        $b \leftarrow b_{\theta_v}(s_i)$
        $g_\theta \leftarrow (L(\pi_i|s_i) - b_i)\nabla_\theta log \, p_\theta(\pi_i|s_i)$
        $\mathcal{L}_v \leftarrow \|b_i - L(\pi_i)\|_2^2$
        $\theta \leftarrow ADAM(\theta, g_\theta)$
        $\theta_v \leftarrow ADAM(\theta_v, \nabla_{\theta_v}\mathcal{L}_v)$
    end for
    return θ
end procedure

รูปที่ 9 รหัสขั้นตอนการเรียนรู้แบบแอคเตอร์-คริติค

แหล่งที่มา Bello et al. (2016)

3.4.การค้นหาแบบทาบู

อัลกอริทึมเพื่อใช้คำนวณขอบเขตบนของค่าที่ดีที่สุด คือ การค้นหาแบบทาบู (Tabu search) ในเบื้องต้นการค้นหาแบบทาบูมีลักษณะตรงข้ามกับ Greedy Local Search ซึ่งวิธีแบบ Greedy Local Search จะหยุดเมื่อไม่พบก้าวที่ดีกว่าเพื่อนบ้านของของคำตอบ $\boldsymbol{s}$ หลักการ Tabu search คือ การค้นหาโดยอนุญาติให้ไปยังก้าวที่ไม่ใช้ก้าวที่ทำให้คำตอบที่ดีขึ้น เพื่อป้องกันวงจรของการพบเจอแต่คำตอบเดิม ๆ ณ ก้าวสุดท้าย $\boldsymbol{\ell}$ ถูกเก็บไว้ใน tabu list ที่จะถูกอัพเดทในแต่ละการทำซ้ำ เกณฑ์ที่ใช้สำหรับหยุดการทำงานอัลกอริทึม ยกตัวอย่างเช่น ระบุจำนวนการทำซ้ำไว้ก่อนหน้าไม่มีคำตอบที่ดีขึ้นเป็นต้น โดยองค์ประกอบหลักของ Tabu Search มีดังนี้

1. ฟังก์ชันต้นทุนของคำตอบล่าสุด
2. คำตอบเริ่มต้น
3. วิธีการค้นหาเพื่อนบ้าน
4. ทาบูลิสต์
5. วิธีการหาคำตอบของการค้นหาแบบทาบู

โดยองค์ประกอบของ Tabu Search จะมีรายละเอียดดังต่อไปนี้

3.4.1.ฟังก์ชันต้นทุนของการเคลื่อนย้ายจักรยาน

ต้นทุนของการเคลื่อนย้ายจักรยานนี้จะเป็นระยะทางรวม $\sum_{j=1}^{k-1} c_{(i_j,i_{j+1})}$ ที่จะรวมค่าปรับเมื่อลำดับในคำตอบ infeasible ยกตัวอย่าง กรณีที่ไม่มีคำตอบที่เป็นไปได้เลยสำหรับ SVOCPDP โดนจะเกิดจากการผ่อนปรนเงื่อนไขและผลลัพธ์ของฟังก์ชันต้นทุนจะให้นิยามดังที่แสดงในสมการ (3.14)

$$f(s) = \sum_{j=1}^{k-1} c_{(i_j,i_{j+1})} + \gamma \sum_{i \in V} (p_i - p'_i) \tag{3.14}$$

โดยที่ γ เป็นค่าคงที่บวก และ p'_i คำนวณมากจากอัลกอริทึม max-flow โดย $p'_i \leq p_i$

3.4.2.คำตอบเริ่มต้น

คำตอบเริ่มต้นควรจะมีลักษณะเป็น feasible solution เพื่อเป็นแนวทางแก่ Tabu search ในการพัฒนาคำตอบที่รวดเร็วขึ้นและได้ระยะเส้นทางรวมที่สั้น ในงานการทดลองงานวิจัยนี้จะใช้เส้นทางเริ่มต้นทางเป็นเส้นทางที่ดีที่สุดที่ได้จากโครงข่ายที่เรียนรู้แบบ Reinforcement Learning มาพัฒนาคำตอบให้ดียิ่งขึ้น

3.4.3.การค้นหาเพื่อนบ้าน

แต่ละขั้นตอนการทำซ้ำเพื่อนบ้านจะถูกสำรวจทั้งหมด สมมติ $\mathbf{k}$ แทนความยาวของลำดับ และ $\mathbf{k}$ โดยปกติจะมากกว่า $\mathbf{n}$ ต่อไปนี้จะอธิบายถึงวิธีการขยับของการหาเพื่อนบ้านที่จะกลายมาเป็นพื้นที่ว่างของการหาคำตอบนี้

1. *2-OPT* เลือกคู่ของเส้นเชื่อม *(Arc)* ที่ไม่ได้ต่อเนื่องกัน ลบเส้นเชื่อมนั้นออกไปจากลำดับและแทรกเส้นเชื่อมใหม่เข้าไปเข้าไป ดังนั้นลำดับการเดินทางจะสลับและได้ลำดับของเส้นทางใหม่หากได้เส้นทางที่ดีขึ้น ใช้เวลา $O(k^2)$ ในการทดสอบทุกรูปแบบการเคลื่อนที่
2. การลบจุดยอด *(Suppression)* ในลำดับการเคลื่อนที่ ใช้เวลา $O(k)$ ในการทดสอบทุกรูปแบบการเคลื่อนที่
3. การเพิ่มจุดยอดที่ไม่สมดุล การขยับวิธีนี้จะทำงานก็ต่อเมื่อในลำดับล่าสุดนั้น *infeasible* ทุกจุดยอดจะถูกตรวจสอบ และเลือกจุดยอดที่ไม่สมดุลมากที่สุด โดยให้จุดยอด i แทนจุดยอดที่มีจักรยานเกินมากที่สุด และ จุดยอด j แทนจุดยอดที่จักรยานขาดมากที่สุด *(*จุดยอดที่ $|(q_i - q'_i) - (p_i - p'_i)|$ มากที่สุด*)* จะเติมจุดยอดเข้าไปโดยเพิ่มขยับ $\rightarrow j \rightarrow i$ หลัง i ในลำดับ ในทางตรงกันข้ามเพิ่มขยับ $\rightarrow i \rightarrow j$ หลัง j ในลำดับ ถ้าไม่มี i และ j ในลำดับการส่งของจะเพิ่ม $\rightarrow i \rightarrow j$ ในลำดับสุดท้าย
4. การเพิ่มจุดยอดบัฟเฟอร์ การขยับนี้พยายามที่จะเพิ่มจุดยอดซ้ำเข้าไปในลำดับ จุดยอดซ้ำนี้ใช้เป็นจุดยอดบัฟเฟอร์เนื่องจากสามารถใช้เป็นที่รับหรือส่งจักรยานได้ ดังนั้นถ้า จุดยอดใช้เป็นจุดยอดบัฟเฟอร์จะต้องมีรถมาเยือนอย่างน้อย *2* ครั้ง ทุกจุดยอดถูกใช้เป็นจุดมมุมบัฟเฟอร์ได้ และจุดยอดซ้ำนี้จะถูกลองเข้าไปแทรกในลำดับการเคลื่อนที่ ถ้าเป็นจุดยอดที่สมดุลอยู่แล้วหรือไม่ปรากฏในลำดับแต่แรกจะถูกใช้เป็นบัฟเฟอร์ได้ *2* ครั้ง ใช้เวลา $O(k^3)$ ในการทดสอบทุกรูปแบบการเคลื่อนที่

3.4.4.ทาบูลิสต์

ในทุกๆ การเปลี่ยนคำตอบล่าสุดที่กล่าวมาข้างต้นจะมีเส้นเชื่อมที่หายจากคำตอบเดิมและเส้นเชื่อมใหม่เข้าใหม่มาในคำตอบ ทาบูลิสต์ใช้สำหรับเก็บข้อมูลเส้นเชื่อมที่ยังคงอยู่หรือกล่าวได้ว่าไม่ได้เข้ามาในคำตอบใหม่แบะออกไปจากคำตอบตอบเก่าและตำแหน่งของมันที่ปรากฏในคำตอบที่ได้ล่าสุด โดยเส้นเชื่อมจะถูกต้องห้ามเป็นระยะการทำซ้ำ ℓ ครั้ง แต่สามารถนำมาแทรกใหม่อีกครั้งได้ที่ตำแหน่งอื่น

3.4.5.วิธีการหาคำตอบของการค้นหาแบบทาบู

องค์ประกอบของ Tabu search ที่กล่าวมาข้างต้นจะถูกนำมาเรียงเป็นอัลกอริทึมการค้นหาแบบทาบูดังนี้ โดนมีการใช้สัญลักษณ์แทน s คือคำตอบล่าสุด s* คือคำตอบที่ดีที่สุดที่เป็นไปได้จากการเริ่มต้นการค้นหาแบบทาบู f(s) คือ ค่าของฟังก์ชันเป้าหมาย (objective function) $NbIterMax$ คือ จำนวนการวนซ้ำที่มากที่สุดก่อนที่การค้นหาแบบทาบูจะสิ้นสุด i คือ รอบการวนซ้ำล่าสุด $\bar{s}_X$ คือ คำตอบที่ดีที่สุดมาจากการขยับรูปแบบ X สำหรับ $s^{\#}$ แทนคำตอบที่ดีที่สุดจากจากการค้นหาด้วยอัลกอริทึมทั้งหมด และ a คือ ตัวแปรสุ่มที่กระจายตัวแบบสม่ำเสมอระหว่าง 0 และ 1

Algorithm 2 Tabu Search algorithm

0: **procedure** Tabu Search (Initial solution s, Number of maximum iterations $NbIterMax$)
1: $s \leftarrow ComputeInitialSolution()$
2: $s^* \leftarrow s$
3: $i \leftarrow 0$
4: **while** $i \leq NbIterMax$ **do**
5: $\bar{s}_{2OPT} \leftarrow Explore2OPT(s)$
6: $\bar{s}_{Sup} \leftarrow ExploreSuppression(s)$
7: $s^{\#} \leftarrow argmin(f(\bar{S}_{2OPT}), f(\bar{S}_{Sup}))$
8: **if** s is not a feasible route **then**
9: $\bar{S}_{AddUnc} \leftarrow ExploreAddUnbalance(s)$
10: **if** $f(s^{\#}) > f(\bar{S}_{AddUnb})$ **then**
11: $\boldsymbol{s^{\#} \leftarrow \bar{S}_{AddUnb}}$
12: **end if**
13: **end if**
14: a $\leftarrow Random()$
15: **if** $\boldsymbol{a \leq}$ 0.2 **then**
16: $\bar{S}_{AddBuf} \leftarrow ExploreAddBuffer(s)$
17: **if** $f(s^{\#}) > f(\bar{S}_{AddBuf})$ **then**
18: $s^{\#} \leftarrow \bar{S}_{AddBuf}$
19: **end if**
20: **end if**
21: $\boldsymbol{s \leftarrow s^{\#}}$ and update the tabu list
22: **if** s in a route and $\boldsymbol{f(s) > f(s^{\#})}$ **then**
23: $\boldsymbol{s^* \leftarrow s}$
24: **end if**
25: $i \leftarrow i + 1$
26: **end while**
27: **end procedure**

รูปที่ 10 รหัสขั้นตอนการค้นหาแบบทาบู

แหล่งที่มา Daniel, Frédéric, and Roberto (2013)

บทที่ 4
แนวทางการทดลองและผลการทดลอง

4.1.ตัวอย่างที่ใช้ในการทดลอง

สำหรับตัวอย่างที่ใช้ในการทดสอบปัญหานี้ยังไม่มีเกณฑ์มาตรฐานสำหรับการวัดประสิทธิภาพของอัลกอริทึม ในงานวิจัยนี้จะใช้แนวทางการสุ่มตัวอย่างจาก Hernandez-Perez, H., Salazar-Gonzalez, J.S., (2004). ซึ่งจะสุ่มตัวอย่างโดย ตำแหน่งของพิกัดสถานีเริ่มต้นของยานพาหนะจะคือ [0,0] ตำแหน่งพิกัดของสถานีต่างๆจะอยู่ในรูปแบบ 2D Euclidean Space หรือคู่อันดับ x และ y โดย สุ่มออกมาเป็นคู่อันดับ [-500,500]x[-500,500] โดยกระจายตัวแบบสม่ำเสมอ (Uniform Distribution) สำหรับจำนวนจักรยานเริ่มต้น (Initial state) ที่แต่ละสถานีมีค่าเท่ากับ 10 โดยที่แต่ละสถานีมีความจุ (c) ทั้งสิ้น 20 คัน และจำนวนจักรยาน ณ เวลาสุดท้าย (Final state) ในแต่ะละสถานีจะมีค่าเท่ากับ 10 + d’ โดย d’ แทนการกระจายตัวแบบสม่ำเสมอระหว่าง [-10,10] สำหรับยานพาหนะที่ใช้ขนย้ายจะมีความจุ (q) เท่ากับ 10 คัน โดยขนาดของโครงข่ายจักรยานที่ทดลองจะมีขนาด 20 สถานี 40 สถานี และ 60 สถานี โดยทดลองแต่ละขนาดสถานีละ 3 ตัวอย่างที่ที่ถูกสุ่มออกมา

4.2.รายละเอียดการฝึกฝนโครงข่ายด้วยการเรียนรู้แบบเสริมกำลัง

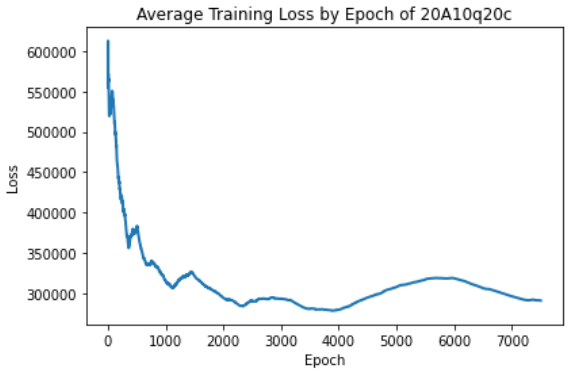
สำหรับ Pointer network ในชั้นของ Encoder layer และ Decoder layer จะประกอบด้วยโครงข่าย LSTM เนื่องจากเหมาะสมในการจดจำข้อมูลที่รับเข้าแบบลำดับเป็นโครงข่ายลักษณะ Sequence-to-sequence network และในงานวิจัยก่อนหน้าใช้ในการทดลอง การฝึกฝนและทดสอบ Pointer network ฝึกฝนเพื่อแก้ปัญหาให้กับเฉพาะโครงข่ายจักรยานให้เช่าต่อ 1 โครงข่ายเท่านั้น กล่าวคือ จะไม่ฝึกฝนและทดสอบบนโครงข่ายที่ลักษณะหรือพิกัดของแต่ละสถานีต่างกัน การฝึกฝนจะให้ Pointer network ในการเรียนรู้แบบเสริมกำลังทั้ง 7,500 รอบ หรือลองให้โมเดลคำนวณคำตอบและเรียนรู้หรือปรับพารามิเตอร์เพื่อพัฒนาคำตอบ โดยฝึกฝนด้วยการกระจายตัวของจำนวนจักรยาน ณ เวลาสุดท้าย (Final state) ที่แตกต่างกัน 10 ชุดหรือจะเปลี่ยนชุดโจทย์การฝึกฝนเมื่อฝึกฝนและปรับพารามิเตอร์ครบ 750 รอบ สำหรับชุดข้อมูลทดสอบจะเป็นชุดเดียวกับกับที่ทดสอบอัลกอริทึม Branch and Cut

4.3.รายละเอียดของการทดลอง

การทดลองในรายงานเปรียบเทียบประสิทธิภาพระหว่างอัลกอริทึม Branch and Cut และ Reinforcement Learning Network (RL Network) ร่วมกับ Tabu Search เพื่อใช้แก้ปัญหาปรับสมดุลจักรยานในบริการจักรยานให้เช่าโดยจะเปรียบเทียบในแง่มุมของ เวลาที่ใช้ในการคำนวณ คุณภาพของคำตอบที่ได้ รวมถึงจำนวน Feasible solution จาก RL Network โดยอัลกอริทึมและสคริปที่ใช้ในการรันโมเดลถูกเขียนด้วยภาษา Python 3.9 โดยใช้ไลบรารี่ gurobipy 9.5.2 ของ Gurobi Optimization เพื่อหาคำตอบของ Linear Programming สำหรับอัลกอริทึม Branch and Cut สร้างโมเดลโครงข่ายการเรียนรู้แบบเสริมกำลังด้วยไลบราลี่ TensorFlow 2.8 และ NumPy 1.23.1 สำหรับการคำนวณทางพีชคณิตเชิงเส้น และไลบราลี่ NetworkX 2.8 สำหรับการแก้ปัญหา Maximum flow ในการหาจำนวนจักรยานที่จะถูกขนย้ายระหว่างสถานี โดยการทดลอง Branch and Cut จะให้เวลาอัลกอริทึมค้นหาคำตอบ 60 นาที สำหรับโครงข่ายที่ถูกฝึกฝนด้วยการเรียนรู้แบบเสริมกำลังมานั้น จะให้เวลาในการประมวลผลคำตอบ 10 นาทีและเลือกคำตอบที่ดีที่สุด จากนั้นนำคำตอบที่ดีที่สุดดังกล่าวเป็นคำตอบเริ่มต้นเพื่อพัฒนาคุณภาพคำตอบด้วย Tabu Search โดยการวนซ้ำหาคำตอบหรือ NbIterMax เท่ากับ 100 ครั้ง หากไม่พบคำตอบที่พัฒนาขึ้นในการวนลูปภายใน 20 ครั้งจะหยุดการทำงานของอัลกอริทึม Tabu search โดยการทดลองจะทดลองบนบริการ Google Colaboratory PRO+ ปี 2022 โดยให้บริการ GPU หลากหลายชนิดสลับกัน อาธิเช่น K80 T4 และ P100 RAM 52GB และไม่ได้ระบุ CPU ที่ให้บริการ

4.4.ผลการฝึกฝนหาคำตอบการปรับสมดุลจักรยานด้วยการเรียนรู้แบบเสริมกำลัง

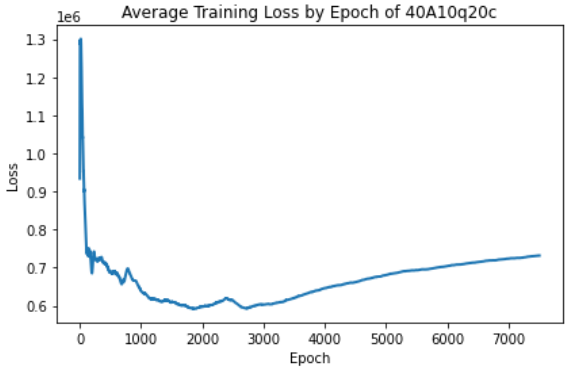
จากกราฟในรูปที่ แสดง Loss function หรือ $L(\theta|s)$ มีค่าเท่ากับระยะเส้นทางรวมและความ imbalance ของจำนวนจักรยาน เมื่อผ่านการฝึกฝนหลายๆรอบ (epoch) พบว่า Loss function มีค่าที่ลดลงเมื่อเวลาผ่านไป และจะมีในบางกรณีที่ Loss function มีค่าที่กลับมาเพิ่มขึ้น อย่างเช่นตัวอย่าง 40A10q20c เป็นต้น แนวโน้มที่ลดลงของ Loss function หมายความว่าโครงข่ายที่ฝึกฝนสามารถเรียนรู้และสามารถหาเส้นทางที่สั้นลงได้โดยเฉลี่ย



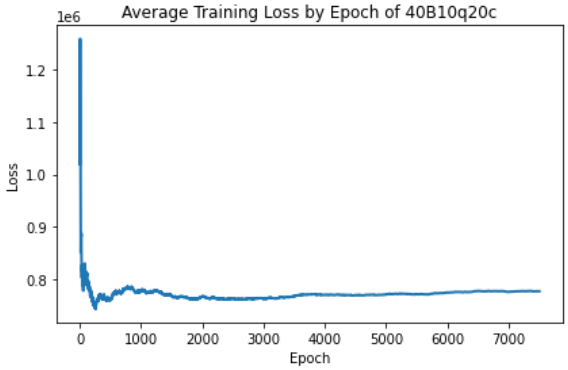
รูปที่ 11 กราฟอัตราการสูญเสียที่ลดลงขณะที่เรียนรู้ของตัวอย่างการทดลอง 20A10q20c

รูปที่ 12 กราฟอัตราการสูญเสียที่ลดลงขณะที่เรียนรู้ของตัวอย่างการทดลอง 20B10q20c

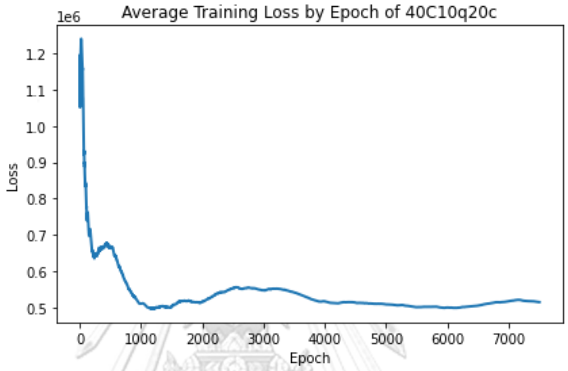
รูปที่ 13 กราฟอัตราการสูญเสียที่ลดลงขณะที่เรียนรู้ของตัวอย่างการทดลอง 20C10q20c



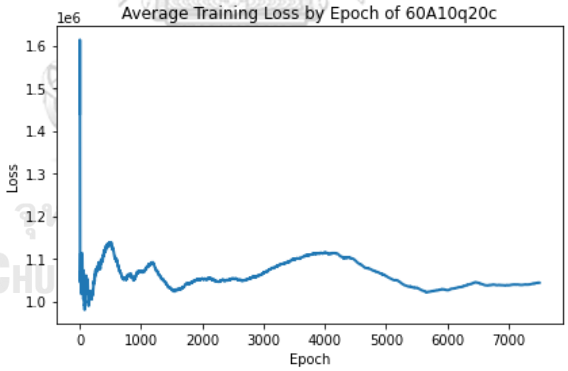
รูปที่ 14 กราฟอัตราการสูญเสียที่ลดลงขณะที่เรียนรู้ของตัวอย่างการทดลอง 40A10q20c



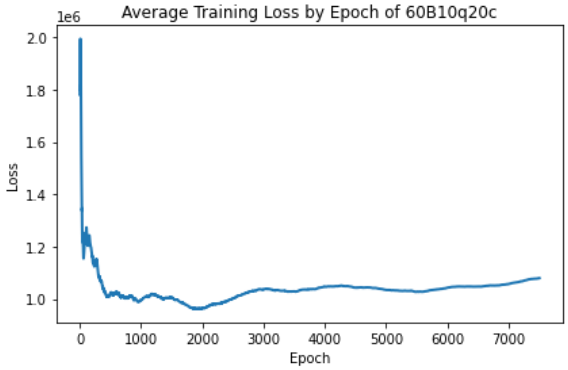
รูปที่ 15 กราฟอัตราการสูญเสียที่ลดลงขณะที่เรียนรู้ของตัวอย่างการทดลอง 40B10q20c



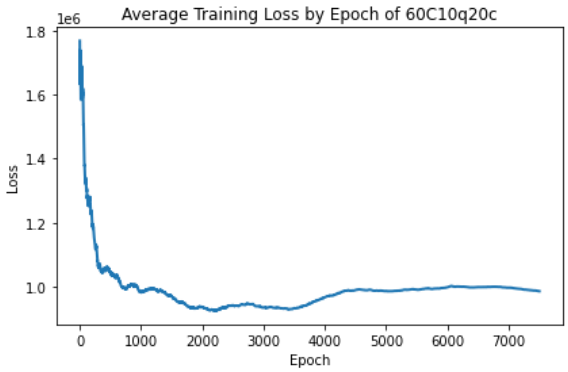
รูปที่ 16 กราฟอัตราการสูญเสียที่ลดลงขณะที่เรียนรู้ของตัวอย่างการทดลอง 40C10q20c



รูปที่ 17 กราฟอัตราการสูญเสียที่ลดลงขณะที่เรียนรู้ของตัวอย่างการทดลอง 60A10q20c



รูปที่ 18 กราฟอัตราการสูญเสียที่ลดลงขณะที่เรียนรู้ของตัวอย่างการทดลอง 60B10q20c



รูปที่ 19 กราฟอัตราการสูญเสียที่ลดลงขณะที่เรียนรู้ของตัวอย่างการทดลอง 60C10q20c

4.5.ผลการทดลอง

ผลการทดลองนี้แบ่งออกเป็น 3 ส่วนโดยเพื่อง่ายต่อการวิธีประสิทธิภาพของโครงข่ายที่เรียนรู้แบบเสริมกำลัง

4.5.1.จำนวนคำตอบทั้งหมดและจำนวน Feasible solution จากโครงข่ายที่เรียนรู้แบบเสริมกำลัง

จากผลการทดลองพบว่าภายใน 10 นาที โครงข่ายแต่ละขนาดมีความเร็วในการคำนวณคำตอบออกมาที่แตกต่างกัน โดยขนาดของโครงข่ายที่ใหญ่ขึ้น จะส่งผลให้คำนวณคำตอบได้น้อยลงหรือช้าลง นอกจากนี้จำนวนของคำตอบที่เป็น Feasible solution มีจำนวนน้อยลงเมื่อขนาดของโครงข่ายหรือจำนวนสถานีมากขึ้นเช่นกัน จำนวนจำนวนคำตอบที่คำนวณออกมาได้มีความผันผวนสูงสำหรับโครงข่ายจักรยานแต่ละขนาดและยังไม่สามารถหาความสัมพันธ์ได้ในเชิงสถิติ

ชื่อตัวอย่าง	จำนวนคำตอบทั้งหมด	จำนวนคำตอบที่เป็น Feasible solution
20A10q20c	267	9
20B10q20c	214	47
20C10q20c	234	39
40A10q20c	207	1
40B10q20c	362	1
40C10q20c	108	2
60A10q20c	127	0
60B10q20c	167	0
60C10q20c	111	0

ตารางที่ 1 จำนวนคำตอบที่โครงข่ายที่เรียนรู้แบบเสริมกำลังคำนวณได้ใน 10 นาทีและจำนวน Feasible solution

4.5.2.คุณภาพของคำตอบที่ดีที่สุด

จากตารางด้านล่างพบว่าเส้นทางที่ดีที่สุดที่ได้จากโครงข่ายที่เรียนรู้แบบ Reinforcement learning ทั้งหมดยังไม่ใกล้เคียงกับคำตอบที่ได้จากอัลกอริทึม Branch and Cut แต่เมื่อนำคำตอบที่ได้จากโครงข่ายดังกล่าวมาพัฒนาคำตอบด้วยเมตาฮิวริสติก Tabu Search จะพบว่าสามารถให้คุณภาพของคำตอบได้ดีใกล้เคียงกับคำตอบจากอัลกอริทึม Branch and Cut สำหรับโครงข่ายให้บริการเช่าจักรยานขนาด 20 และ 40 สถานี อย่างไรก็ตามสำหรับโครงข่ายให้บริการเช่าจักรยานขนาด 60 สถานี โครงข่ายที่เรียนรู้แบบ Reinforcement learning ให้คำตอบที่แตกต่างจากคำตอบจากอัลกอริทึม Branch and Cut อย่างมาก แม้พัฒนาคำตอบด้วย Tabu search จะพบว่าคำตอบก็จะยังไม่ได้คำตอบที่ใกล้เคียงกับอัลกอริทึม Branch and Cut

ชื่อตัวอย่าง	**Reinforcement Learning network**	**Reinforcement Learning network และ Tabu search**	**Branch and Cut**
20A10q20c	21,474.81	4,765.83	**3,644.02**
20B10q20c	25,148.06	5,344.95	**4,273.25**
20C10q20c	21,911.62	**6,297.79**	7,008.65
40A10q20c	61,253.99	**7,965.64**	9,396.55
40B10q20c	58,550.61	8,629.20	**7,812.11**
40C10q20c	58,266.26	**9,519.71**	9,847.77
60A10q20c	135,146.06	21,546.91	**10,726.67**
60B10q20c	101,708.88	28,443.85	**19,046.58**
60C10q20c	197,420.56	29,889.00	**13,389.03**

ตารางที่ 2 ตารางแสดงคุณภาพคำตอบที่ดีที่สุดหรือความยาวเส้นทางที่สั้นที่สุดของคำตอบที่ได้แต่ละวิธีการหาคำตอบ (ยิ่งต่ำยิ่งดี)

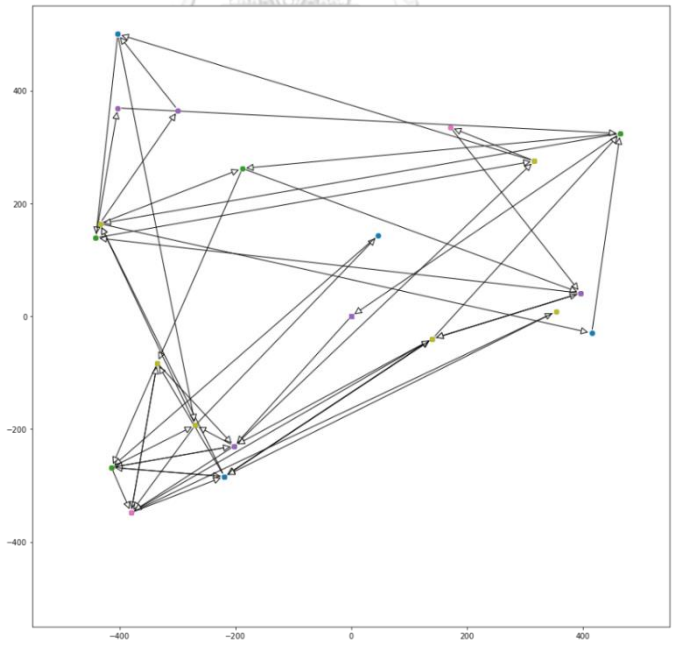
4.5.3.เวลาที่ใช้คำณวนหาคำตอบ

สำหรับการเรียนรู้แบบ Reinforcement learning จะถูกทดลองให้คำนวณเพื่อให้ได้คำตอบในเวลา 10 นาที หรือ 600 วินาทีและจากนั้นใช้ Tabu search ในการพัฒนาคำตอบโดยให้วนลูปทั้งหมด 100 รอบและหากไม่พบคำตอบที่ดีขึ้นภายใน 20 รอบจะทำการหยุดการค้นหาและ Tabu list จะเก็บเส้นเชื่อมไว้นาน 10 รอบเพื่อไม่ให้ผลเฉลยประกอบด้วยเส้นเชื่อมใน Tabu list พบว่าเวลาที่ใช้ในการคำนวณของโครงข่ายเรียนรู้แบบ Reinforcement learning และใช้เมตาฮิวริสติค Tabu search จะใช้เวลาสูงกว่าอัลกอริทึม Branch and cut

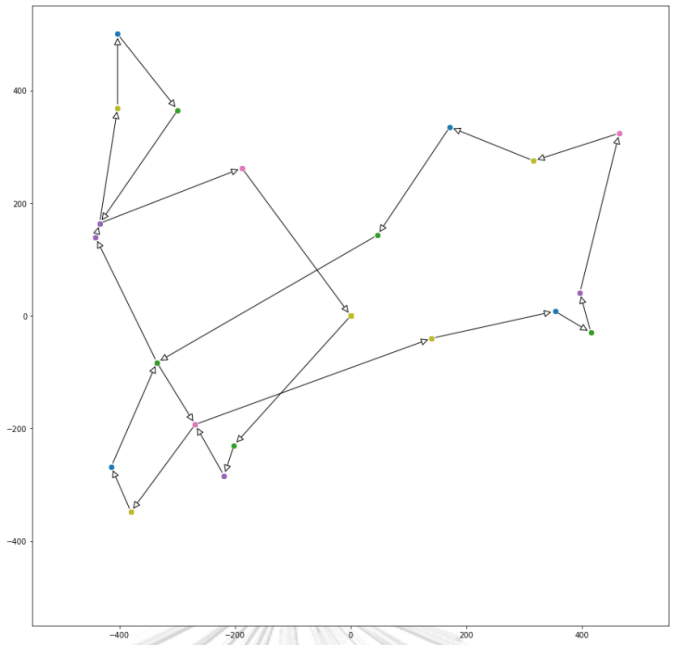
ชื่อตัวอย่าง	Reinforcement Learning network with Tabu search (วินาที)	Branch and Cut (วินาที)
20A10q20c	1,300.54	1,048.80
20B10q20c	899.45	264.00
20C10q20c	998.26	3,600.00
40A10q20c	3,836.10	3,600.00
40B10q20c	3,614.98	3,600.03
40C10q20c	3,752.65	3,600.00
60A10q20c	4,119.93	3,600.09
60B10q20c	4,033.17	3,600.70
60C10q20c	4,474.57	3,600.35

ตารางที่ 3 ตารางแสดงเวลาที่ใช้หาคำตอบในการทดลอง(วินาที)

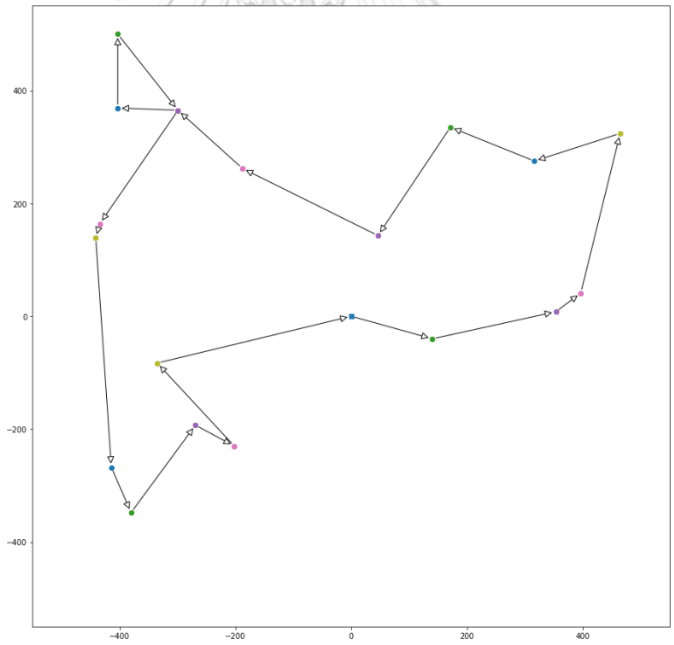
4.6.ตัวอย่างเส้นทางที่ได้จากแต่ละวิธีการ



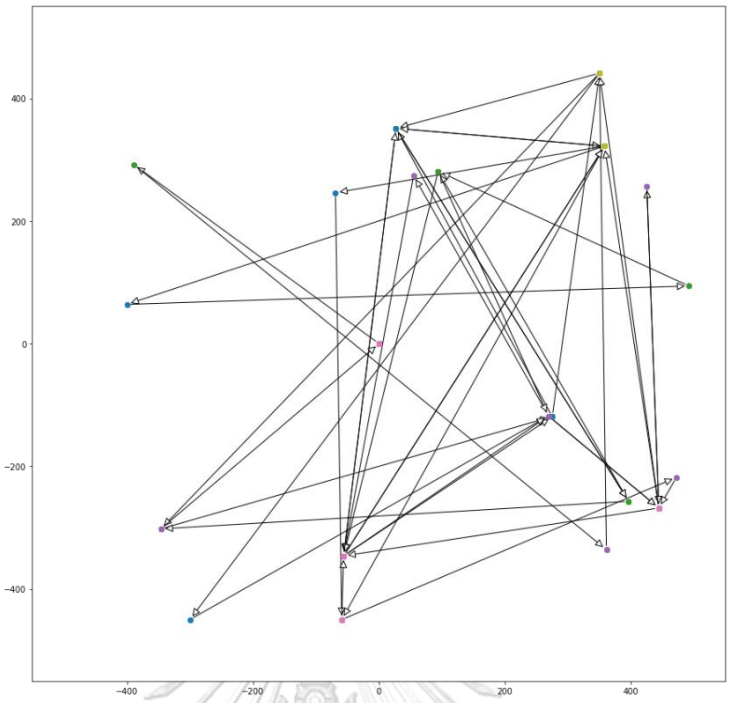
รูปที่ 20 เส้นทางของคำตอบจากตัวอย่าง 20A10q20c คำตอบจาก Reinforcement Learning Network (21,474.81)



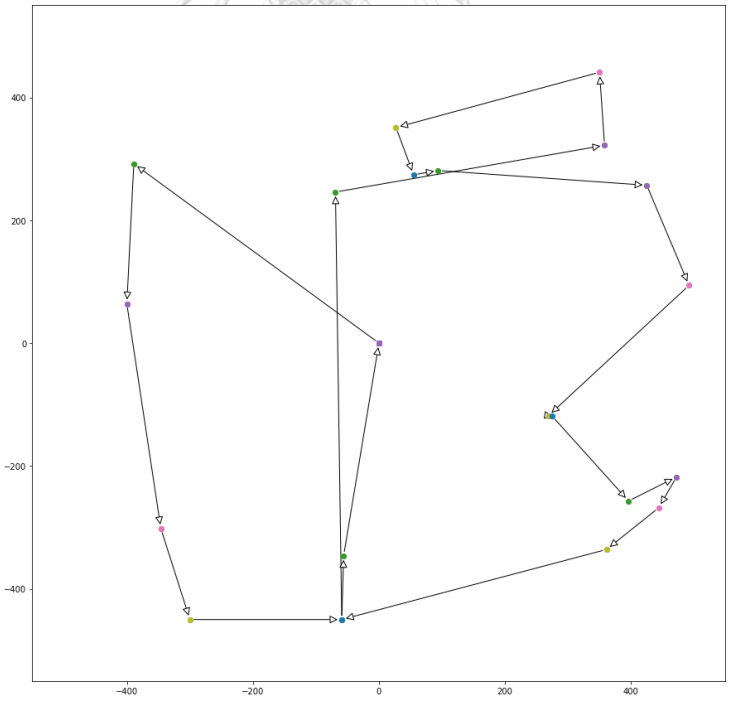
รูปที่ 21 เส้นทางของคำตอบจากตัวอย่าง 20A10q20c จาก Reinforcement Learning Network and Tabu Search (4,765.83)



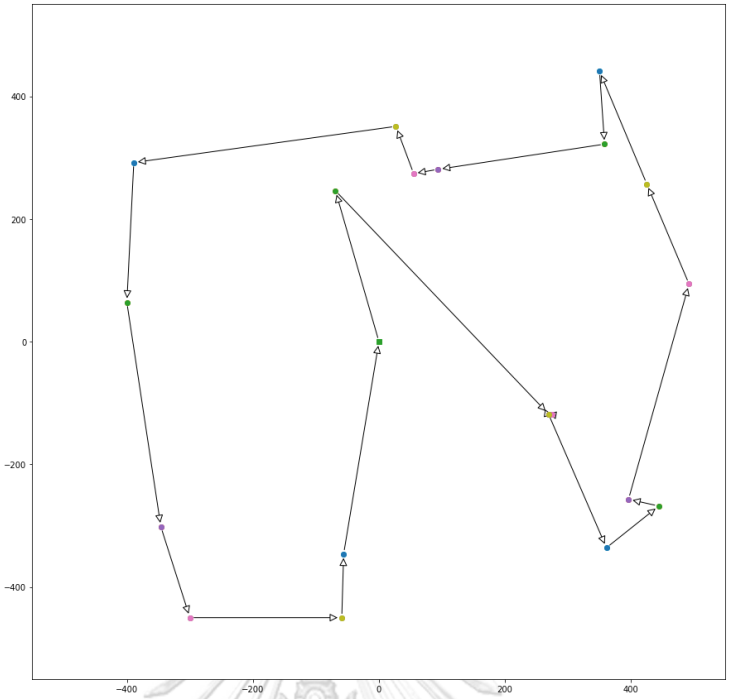
รูปที่ 22 เส้นทางของคำตอบจากตัวอย่าง 20A10q20c จาก Branch and Cut (3,644.02)



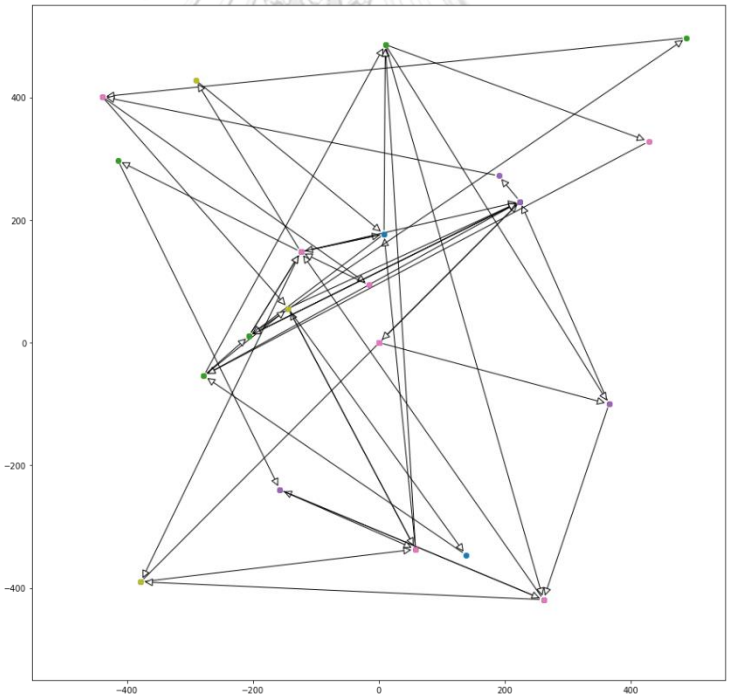
รูปที่ 23 เส้นทางของคำตอบจากตัวอย่าง 20B10q20c จาก Reinforcement Learning Network (25,148.06)



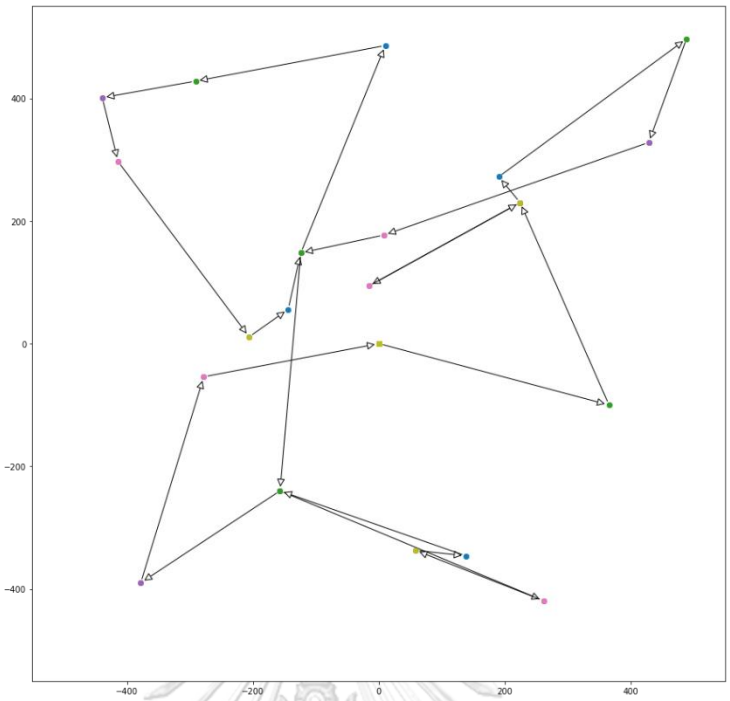
รูปที่ 24 เส้นทางของคำตอบจากตัวอย่าง 20B10q20c จาก Reinforcement Learning Network and Tabu search (5,344.95)



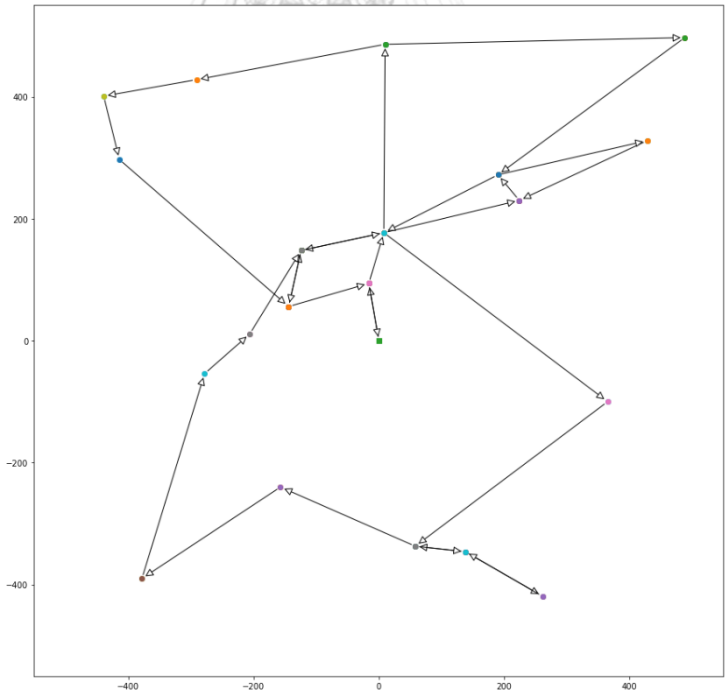
รูปที่ 25 เส้นทางของคำตอบจากตัวอย่าง 20B10q20c จาก Branch and Cut (4,273.25)



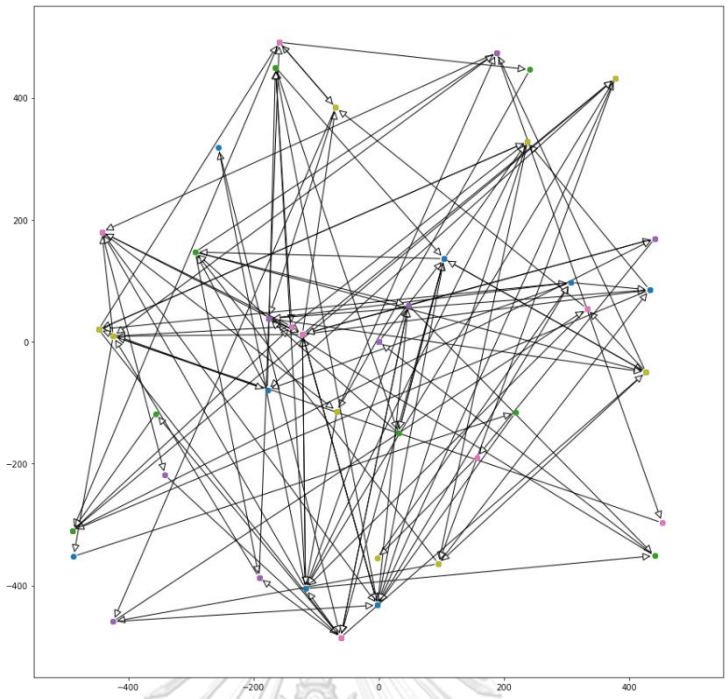
รูปที่ 26 เส้นทางของคำตอบจากตัวอย่าง20C10q20c จาก Reinforcement Learning Network (21,911.62)



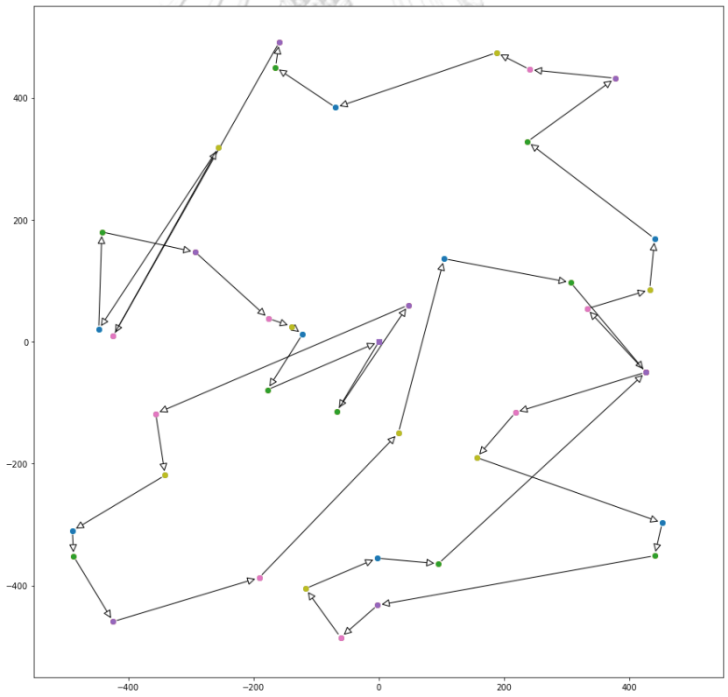
รูปที่ 27 เส้นทางของคำตอบจากตัวอย่าง 20C10q20c จาก Reinforcement Learning Network and Tabu search (6,297.79)



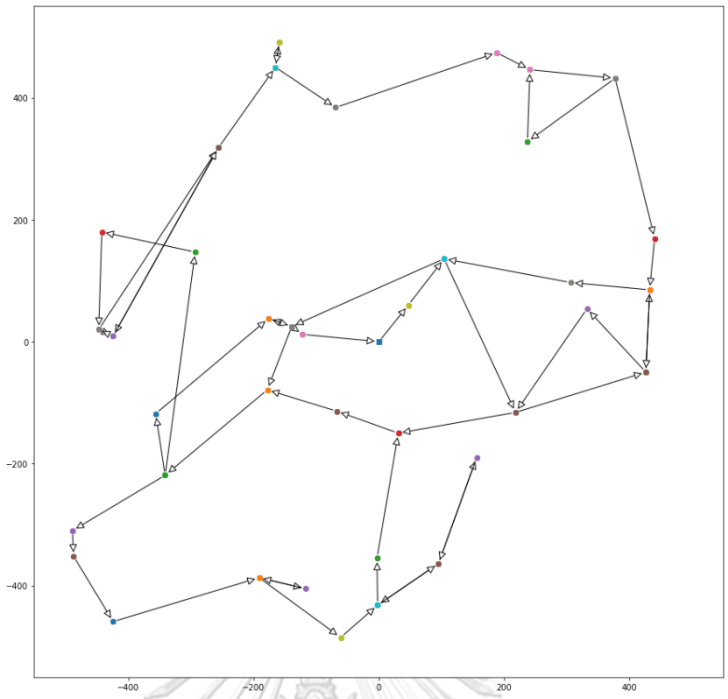
รูปที่ 28 เส้นทางของคำตอบจากตัวอย่าง 20C10q20c จาก Branch and Cut (7,008.65)



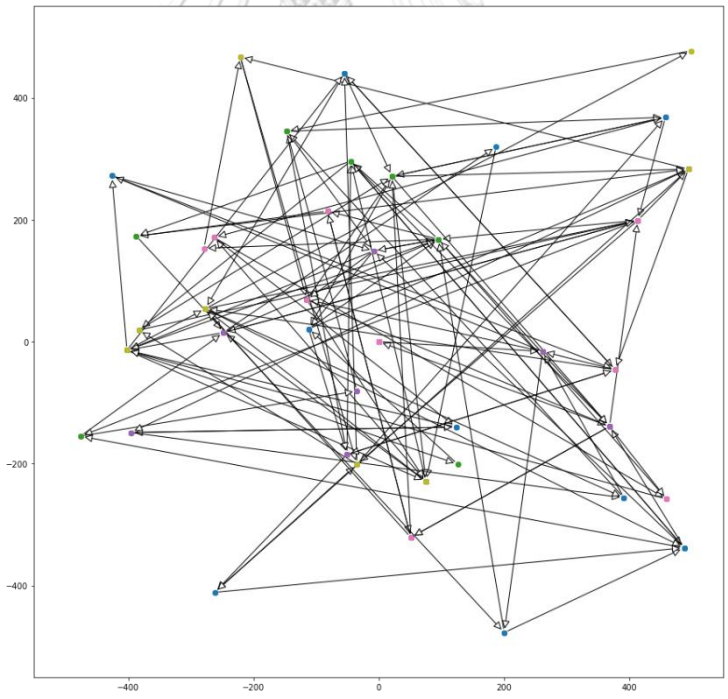
รูปที่ 29 เส้นทางของคำตอบจากตัวอย่าง 40A10q20c จาก RL network (61,253.99)



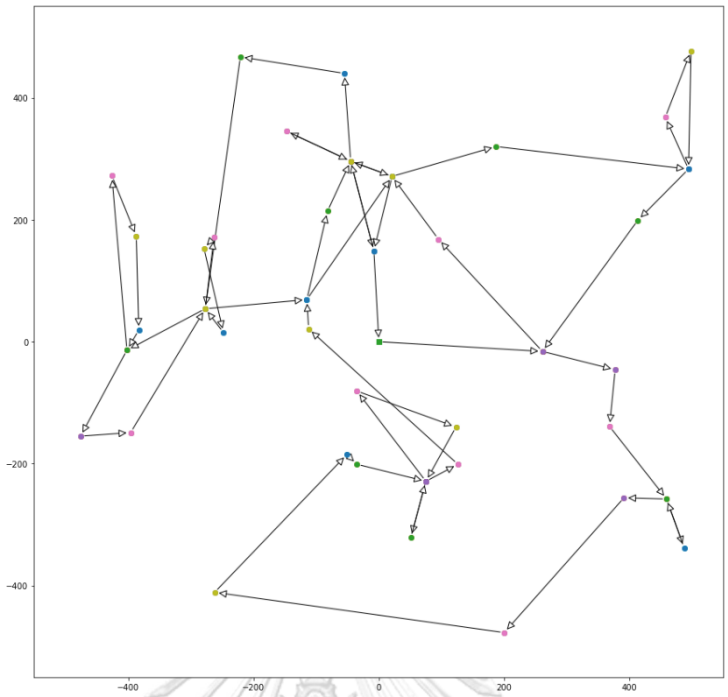
รูปที่ 30 เส้นทางของคำตอบจากตัวอย่าง 40A10q20c จาก Reinforcement Learning network และ Tabu Search (7,965.64)



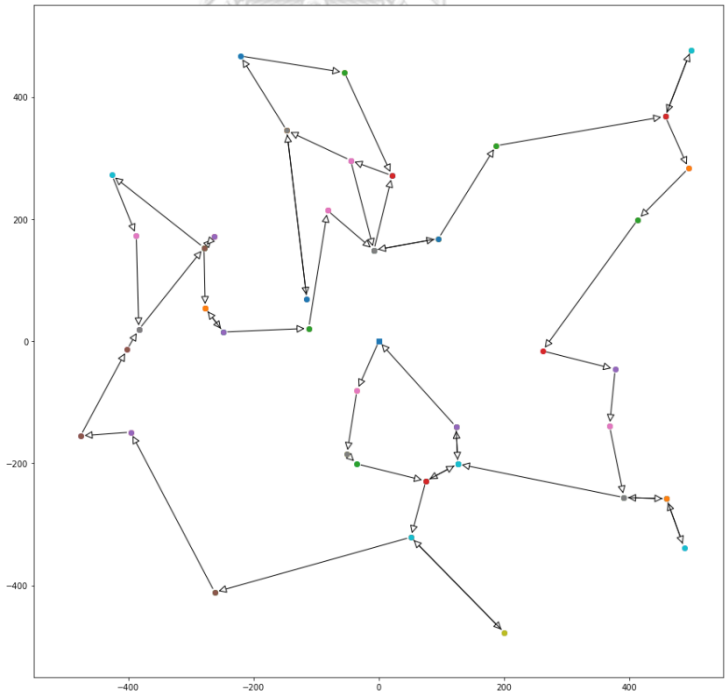
รูปที่ 31 เส้นทางของคำตอบจากตัวอย่าง 40A10q20c จาก Branch and Cut (9,396.55)



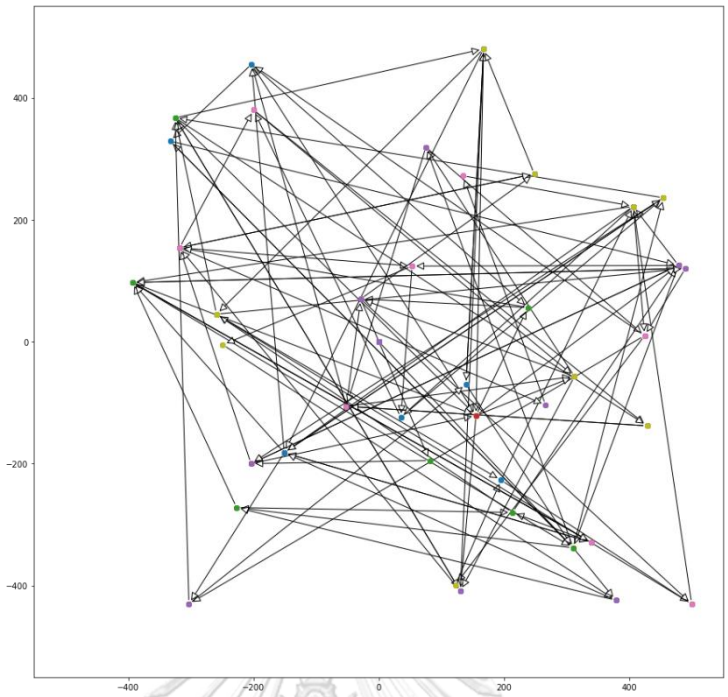
รูปที่ 32 เส้นทางของคำตอบจากตัวอย่าง 40B10q20c จาก Reinforcement Learning network (58,550.61)



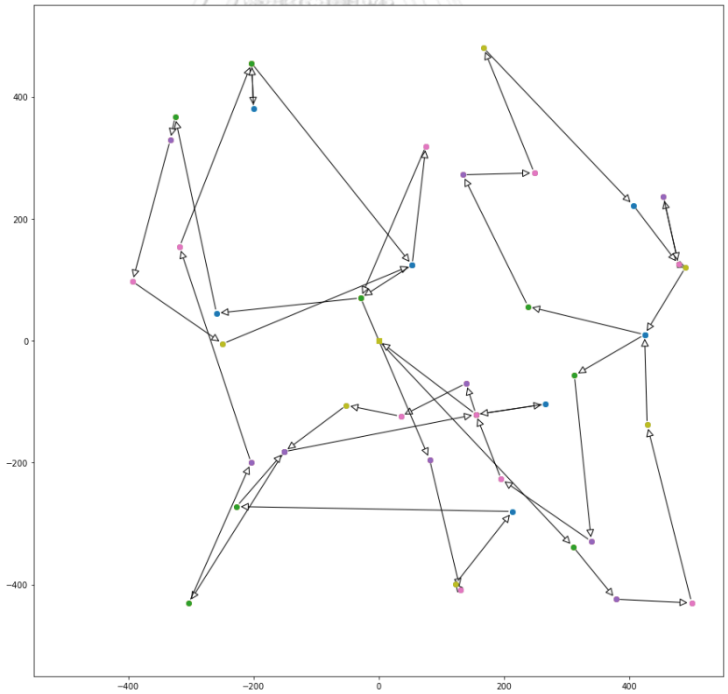
รูปที่ 33 เส้นทางของคำตอบจากตัวอย่าง 40B10q20c จาก Reinforcement Learning network และ Tabu Search (8,629.20)



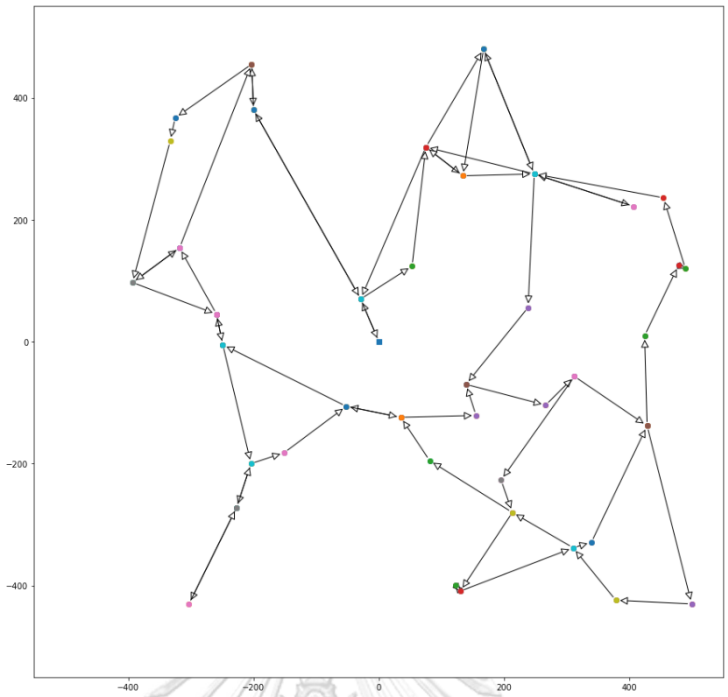
รูปที่ 34 เส้นทางของคำตอบจากตัวอย่าง 40B10q20c จาก Branch and Cut (7,812.11)



รูปที่ 35 เส้นทางของคำตอบจากตัวอย่าง 40C10q20c จาก Reinforcement Learning (58,266.26)



รูปที่ 36 เส้นทางของคำตอบจากตัวอย่าง 40C10q20c จาก Reinforcement Learning and Tabu Search (9,519.71)



รูปที่ 37 เส้นทางของคำตอบจากตัวอย่าง 40C10q20c จาก Branch and Cut (9,847.77)

จากตัวอย่างเส้นทางของคำตอบพบว่าเส้นทางจากคำตอบของอัลกอริทึม Branch and Cut มีรูปร่างเส้นทางที่ใกล้เคียงกับเส้นทางของคำตอบจาก Reinforcement Learning และ Tabu Search ในโครงข่ายจักรยานให้เช่า 20 และ 40 สถานี อย่างไรก็ตาม Branch and Cut จะให้เส้นทางที่เรียบกว่าและเหมาะกับการใช้ในการขนย้ายจักรยานกว่า หรือสามารถพิจารณาเป็นตัวเลือกในการใช้งานจริงได้

บทที่ 5
สรุปผลการวิจัยและแนวทางการวิจัยในขั้นต่อไป

5.1.สรุปผลและอภิปรายงานวิจัย

งานวิจัยนี้ประยุกต์โครงข่าย Pointer network ที่ถูกฝึกฝนด้วยวิธี Reinforcement learning เพื่อแก้ไขปัญหา Static rebalancing problem จากงานวิจัยของ Bello et al. (2016) ที่ประยุกต์กับปัญหา Traveling salesman problem และพบว่าสามารถแก้ไขปัญหาดังกล่าวได้อย่างมีประสิทธิภาพ โดยในงานวิจัยนี้ต้องการนำวิธีดังกล่าวมาประยุกต์ใช้กับปัญหาการปรับสมดุลจักรยานหรือ Static rebalancing problem โดยเป็นปัญหาที่ต้องการคำตอบเป็นเส้นทางในการเข้าไปรับและส่งจักรยานที่แต่ละสถานี โดยกำหนดให้จักรยานมีขนาดเดียวและสำหรับเส้นทางแก่ยานพาหนะที่มีความจุที่จำกัด 1 คัน ซึ่งยังไม่เคยมีงานวิจัยที่วัดประสิทธิภาพของการประยุกต์ Pointer network ที่ฝึกฝนด้วย Reinforcement learning ในการแก้ปัญหา Static rebalancing problem มาก่อนในงานวิจัยนี้ต้องการวัดประสิทธิภาพของวิธีการนี้ในการแก้ไขปัญหาดังกล่าว

Pointer network ถูกใช้ในการแก้ปัญหาการประมวลภาษาธรรมชาติอย่างแพร่หลาย คำนวณพารามิเตอร์ในโครงข่ายและส่งคำตอบที่เป็นลักษณะลำดับของจุดยอดเรียงกัน และคำตอบตอบจะสิ้นสุดด้วยเงื่อนไขที่ผู้ใช้งานได้กำหนดไว้ สามารถนำมาประยุกต์กับปัญหาเชิงการจัดเส้นทาง 1 เส้นทาง เนื่องจากโครงข่ายดังกล่าวให้คำตอบที่เป็น 1 ลำดับเรียงกันอยู่เดิมแล้ว อีกทั้งสามารถวัดความจุของยานพาหนะและความจุช่องจอดจักรยานแต่ละสถานีจากการแก้ไขปัญหาการไหลสูงสุดบนกราฟของคำตอบเพื่อวัดความ Infeasible ได้และเมื่อนำ Pointer network มาเรียนรู้ด้วยวิธีการ Reinforcement Learning จะพบว่า Pointer network มีการพัฒนาในการจัดเส้นทาง อย่างไรก็ตามเมื่อผ่านการเรียนรู้ระดับหนึ่ง จะพบว่า Pointer network ไม่สามารถเรียนรู้ได้ดีขึ้น

เมื่อเปรียบเทียบลักษณะของปัญหา Static rebalancing problem กับปัญหา Traveling salesman problem แล้วพบว่า Static rebalancing problem จะมีความซับซ้อนกว่าเนื่องจากปัญหาต้องการคำตอบให้มีจำนวนจักรยานสุดท้ายเท่ากับจำนวนที่ต้องการหรือกำหนดไว้ตามแผน ในขณะที่ Traveling salesman problem เป็นปัญหาที่เพียงต้องการหาเส้นทางเดินทางไปให้ครบทุกสถานีหรือซับซ้อนน้อยกว่าในแง่ของคุณสมบัติของคำตอบที่ต้องการ

จากผลการทดลอง คำตอบที่ได้จากโมเดลโครงข่ายประสาท Pointer network ที่เรียนรู้แบบ Reinforcement learning นั้นพบว่าคำตอบยังไม่ใกล้เคียงกับคำตอบจากวิธีการแบบเดิมหรือ

Branch and Cut แต่คำตอบจาก Pointer network ที่เรียนรู้แบบ Reinforcement learning นั้น สามารถนำไปพัฒนาคำตอบด้วยเมตาฮิวริสติค Tabu search ได้อีกและให้คำตอบที่ใกล้เคียงกับอัลกอริทึม Branch and cut อย่างไรก็ตามสำหรับคำตอบจาก Pointer network ที่เรียนรู้แบบ Reinforcement learning และพัฒนาคำตอบด้วย Tabu search จะเหมาะกับเพียงกับการแก้ปัญหาปรับสมดุลจักรยานที่มีขนาดไม่ใหญ่มาก จากผลการทดลองวิธีการที่ใช้ Reinforcement learning ร่วมกับ Tabu search จะเหมาะกับโครงข่ายจักรยานให้เช่าที่มีจำนวนสถานีไม่เกิน 40 สถานี

สำหรับการดำเนินการเพื่อดูแลระบบของระบบให้เช่าจักรยานการใช้วิธี Branch and cut จะให้ประสิทธิภาพด้านคุณภาพคำตอบโดยรวมที่ดีกว่าหรือสามารถทดลองหาเส้นทางจากทั้งวิธีการ Branch and cut และ Reinforcement Learning Network และ Tabu search เพื่อหาคำตอบที่ดีกว่าในการใช้งานจริง

ในแง่ของเวลาที่ใช้การหาเส้นทางพบว่า Reinforcement Learning Network และ Tabu search จะใช้เวลาที่มากกว่าแต่ใกล้เคียงกับ Branch and Cut ในโครงข่ายจักรยานให้เช่าขนาด 20 และ 40 สถานี แต่ Reinforcement Learning Network และ Tabu search จะใช้เวลามากกว่า Branch and Cut ในโครงข่ายจักรยานให้เช่าขนาด 60 สถานี

5.2.แนวทางการวิจัยขั้นถัดไป

ตัวอย่างโครงข่ายและสถานการณ์การปรับสมดุลจักรยานที่วิทยานิพนธ์นี้ใช้ทดลองนั้นอาจจะไม่ตรงกับความเป็นจริงและไม่สามารถเห็นภาพได้ถึงเส้นทางในโครงข่ายที่เกิดขึ้นจริง ดังนั้นงานวิจัยถัดไปควรทดลองใช้วิธีการแก้ปัญหาที่กล่าวมาในวิทยานิพนธ์กับโครงข่ายและสถานการณ์จริง เพื่อยืนยันว่าสามารถใช้ผลเฉลยแก้ไขปัญหาแก้ได้จริงและความเหมาะสมของเวลาที่ใช้ในการคำนวณคำตอบ นอกจากนั้นสามารถประยุกต์วิธีการแก้ปัญหากับการปรับสมดุลจักรยานทั้งสถานการณ์ทั้งตอนช่วงเวลากลางวันและในช่วงเวลากลางคืน

ในงานวิจัยนี้ได้ประยุกต์ใช้โครงข่าย Pointer network ซึ่งเดิมทีใช้ในการแก้ปัญหาประเภท Traveling salesman problem ซึ่งมีความซับซ้อนน้อยกว่าปัญหา Static Traveling salesman problem ผลการทดลองแสดงให้เห็นเส้นทางจากโครงข่ายที่ทำการทดลองไม่สามารถเรียนรู้การสร้างคำตอบที่เป็นเส้นทาง Feasible solution ได้ดีนัก เพราะ จากผลการทดลองมีจำนวนคำตอบที่เป็น Feasible solution ค่อนข้างน้อย แสดงให้เห็นว่าการเพิ่ม Penalty ไปยัง objective function ร่วมกับระยะทางเส้นทางรวมเพียงอย่างเดียวไม่สามารถทำให้โครงข่ายนั้นสามารถหาคำตอบได้เป็น Feasible solution ที่เพียงพอ ดังนั้นผู้เขียนวิทยานิพนธ์เล่มนี้เสนอว่าควรจะปรับสถาปัตยกรรมของ

โครงข่ายประสาทให้เข้าใจการจัดเส้นทางลำดับของสถานีจักรยาน นอกนั้นต้องมีกลไกที่พิจารณาความ Infeasible ภายในโครงข่ายประสาทเพื่อเพิ่มประสิทธิภาพของการเรียนรู้และคำตอบ

นอกจากนี้งานวิจัยในขั้นถัดไปควรพิจารณาการประยุกต์ Reinforcement Learning แก้ปัญหา Combinatorial optimization ที่เป็นปัญหาปรับสมดุลจักรยานให้หลากหลายมากขึ้น เช่น สำหรับแก้โจทย์ที่มีจักรยานมากกว่า 1 ชนิด และยานพาหนะมากกว่า 1 คัน รวมถึงส่งให้ได้ภายใต้กรอบเวลา (Time window) และสำหรับปัญหาแบบ Dynamic เพื่อให้โจทย์นั้นมีความทั่วไปมากขึ้นและใช้ได้กับหลากหลายชนิดด้วยโมเดลชนิดเดียว ผู้เขียนคาดว่าจะต้องหาโครงข่ายที่มีสถาปัตยากรรมที่พัฒนาขึ้นไปอีกหากโครงข่ายประสาทสามารถให้คำตอบได้อย่างรวดเร็วและใกล้เคียงกับอัลกอริทึม Branch and Cut จะมีประโยชน์อย่างมากเนื่องจาก เพื่อที่จะส่งผลให้ไม่ต้องใช้เวลากับการพัฒนาคำตอบด้วย Tabu search มากนัก

โดยแนวทางของการแก้ปัญหาที่มีลักษณะ Dynamic ยกตัวอย่างเช่น มีการเข้ามาใช้บริการและการคืนจักรยานในขณะทำการขนย้ายจักรยาน โดยข้อมูลนำเข้าเข้าที่ใส่เข้าไปในโครงข่ายที่เรียนรู้แบบเสริมกำลังมีการเปลี่ยนแปลงและผลเฉลยเป็นเส้นทางที่ใหม่เข้ามาต่อข้อมูลนำเข้าที่เปลี่ยนแปลง ซึ่งจะเหมาะสมกับการหาเส้นทางขนส่งในช่วงเวลาที่บริการจักรยานให้เช่านั้นเปิดบริการ สำหรับปัญหาภายใต้กรอบเวลาสามารถใช้แนวทางการสร้างโครงข่ายเสริมเพื่อจัดเส้นทางให้ภายใต้กรอบเวลาที่สุดและใช้การตัดสินโทษของเส้นทางว่าที่ขัดแย้งกับเงื่อนไขของโจทย์เพื่อให้การฝึกฝนสามารถจัดเส้นทางได้ภายใต้กรอบเวลา งานวิจัยควรทดลองจำลองสภาพการณ์ใช้งานจริงเพื่อวัดประสิทธิภาพของแนวทางแก้ปัญหาที่กล่าวมาข้างต้น

บรรณานุกรม

Abolhassani, Leili, Amir Afghari, and Hamideh Mohtashami Borzadaran. 2018. "Public Preferences towards Bicycle Sharing System in Developing Countries: The Case of Mashhad, Iran." *Sustainable Cities and Society* 44. doi: 10.1016/j.scs.2018.10.032.

Battarra, Maria, Jean-François Cordeau, and Manuel Iori. "Chapter 6: Pickup-and-Delivery Problems for Goods Transportation." In *Vehicle Routing*, 161-91.

Bello, Irwan, Hieu Pham, Quoc V Le, Mohammad Norouzi, and Samy Bengio. 2016. "Neural combinatorial optimization with reinforcement learning." *arXiv preprint arXiv:*1611.09940.

Claudio, Contardo, Morency Catherine, and Rousseau Louis-Martin. 2012. "Balancing a Dynamic Public Bike-Sharing System." In.

Cruz, Fábio, Anand Subramanian, Bruno Bruck, and Manuel Iori. 2016. "A heuristic algorithm for a single vehicle static bike sharing rebalancing problem." *Computers & Operations Research* 79. doi: 10.1016/j.cor.2016.09.025.

Daniel, Chemla, Meunier Frédéric, and Roberto. 2013. "Bike sharing systems: Solving the static rebalancing problem." *Discrete Optimization* 10 (2):120-46. doi: https://doi.org/10.1016/j.disopt.2012.11.005.

Francesca, Maggioni, Cagnolari Matteo, Bertazzi Luca, and W. Wallace Stein. 2019. "Stochastic optimization models for a bike-sharing problem with transshipment." *European Journal of Operational Research* 276 (1):272-83. doi: https://doi.org/10.1016/j.ejor.2018.12.031.

Hernández-Pérez, Hipólito, and Juan-José Salazar-González. 2004. "Heuristics for the One-Commodity Pickup-and-Delivery Traveling Salesman Problem." *Transportation Science* 38 (2):245-55. doi: 10.1287/trsc.1030.0086.

Legros, Benjamin. 2019. "Dynamic repositioning strategy in a bike-sharing system; how to prioritize and how to rebalance a bike station." *European Journal of Operational Research, Elsevier* 272(2):740-53.

Maioli, Heictor, Raíssa Corrêa de Carvalho, and Denise Medeiros. 2019. "SERVBIKE: Riding

customer satisfaction of bicycle sharing service." *Sustainable Cities and Society* 50:101680. doi: 10.1016/j.scs.2019.101680.

Megajuice. 2017. "Reinforcement learning diagram." In *Wikimedia Commons*, edited by Reinforcement learning diagram, Diagram showing the components in a typical Reinforcement Learning (RL) system. An agent takes actions in an environment which is interpreted into a reward and a representation of the state which is fed back into the agent. Incorporates other CC0 work: https://openclipart.org/detail/202735/eye-side-view. Wikimedia Commons.

Patrice, Leclaire, and Couffin Florent. 2018. "Method for Static Rebalancing of a Bike Sharing System." *IFAC-PapersOnLine* 51 (11):1561-6. doi: https://doi.org/10.1016/j.ifacol.2018.08.274.

Regue, Robert, and Will Recker. 2014. "Proactive vehicle routing with inferred demand to solve the bikesharing rebalancing problem." *Transportation Research Part E: Logistics and Transportation Review* 72 (C):192-209.

Savelsbergh, M. W. P., and M. Sol. 1995. "The General Pickup and Delivery Problem." *Transportation Science* 29 (1):17-29.

Shaheen, Susan, Stacey Guzman, and Hua Zhang. 2010. "Bikesharing in Europe, the Americas, and Asia: Past, Present, and Future." *Institute of Transportation Studies, UC Davis, Institute of Transportation Studies, Working Paper Series* 2143. doi: 10.3141/2143-20.

Sin, C. Ho, and W. Y. Szeto. 2014. "Solving a static repositioning problem in bike-sharing systems using iterated tabu search." *Transportation Research Part E: Logistics and Transportation Review* 69:180-98. doi: https://doi.org/10.1016/j.tre.2014.05.017.

Sutton, Richard S., and Andrew G. Barto. 2018. *Reinforcement Learning: An Introduction*: A Bradford Book.

Vinyals, Oriol, Meire Fortunato, and Navdeep Jaitly. 2015. "Pointer networks." *Advances in neural information processing systems* 28.

Vogel, Patrick, and Dirk C. Mattfeld. 2011. Strategic and Operational Planning of Bike-Sharing Systems by Data Mining – A Case Study. Paper presented at the Computational Logistics, Berlin, Heidelberg, 2011//.

Yao, Yao, Linwei Liu, Zibin Guo, Ziheng Liu, and Huiyu Zhou. 2019. "Experimental Study on Shared Bike Use Behavior under Bounded Rational Theory and Credit Supervision Mechanism %M doi:10.3390/su11010127 %U https://www.mdpi.com/2071-1050/11/1/127." *Sustainability %@* 2071-1050 11 (1):127.

ประวัติผู้เขียน

ชื่อ-สกุล	ธีร์ธัฐ พรหมประดิษฐ์
วัน เดือน ปี เกิด	3 ธันวาคม พ.ศ.2540
สถานที่เกิด	โรงพยาบาลรามาธิบดี
วุฒิการศึกษา	เรียนจบชั้นประถมศึกษาจากโรงเรียนอนุราชประสิทธิ์ ปี 2553 เรียนจบชั้นมัธยมศึกษาจากโรงเรียนสามเสนวิทยาลัย ปี 2559 และเรียนจบปริญญาตรี วิศวกรรมศาสตร์บัณฑิต จุฬาลงกรณ์มหาวิทยาลัย ปี 2563
ที่อยู่ปัจจุบัน	11 ซ.เลี่ยงเมืองนนทบุรี 15 ถ.เลี่ยงเมืองนนทบุรี ต.บางกระสอ อ.เมือง จ.นนทบุรี 11000