



การค้นหาลงประดิษฐกรรมโครงข่ายประสาทเทียมโดยใช้อัลกอริทึมเชิงวิวัฒนาการแบบหลาย
วัตถุประสงค



โดย
นางสาวสุวิภา สชีวะผาสุก

วิทยานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรวิศวกรรมศาสตรมหาบัณฑิต

สาขาวิชาวิศวกรรมไฟฟ้าและคอมพิวเตอร์ แผน ก แบบ ก 2

ภาควิชาวิศวกรรมไฟฟ้า

มหาวิทยาลัยศิลปากร

ปีการศึกษา 2567

ลิขสิทธิ์ของมหาวิทยาลัยศิลปากร

การค้นหายาแก้ปวดบรรเทาอาการไข้หวัดใหญ่ด้วยอัลกอริทึมเชิงวิวัฒนาการแบบหลาย
วัตถุประสงค์



วิทยานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรวิศวกรรมศาสตรมหาบัณฑิต
สาขาวิชาวิศวกรรมไฟฟ้าและคอมพิวเตอร์ แผน ก แบบ ก 2
ภาควิชาวิศวกรรมไฟฟ้า
มหาวิทยาลัยศิลปากร
ปีการศึกษา 2567
ลิขสิทธิ์ของมหาวิทยาลัยศิลปากร

NEURAL ARCHITECTURE SEARCH USING MULTI-OBJECTIVE EVOLUTIONARY
ALGORITHMS



A Thesis Submitted in Partial Fulfillment of the Requirements
for Master of Engineering (ELECTRICAL AND COMPUTER ENGINEERING)
Department of ELECTRICAL ENGINEERING
Academic Year 2024
Copyright of Silpakorn University

640920028 : วิศวกรรมไฟฟ้าและคอมพิวเตอร์ แผน ก แบบ ก 2

คำสำคัญ : การค้นหาสถาปัตยกรรมโครงข่ายประสาทเทียม, การหาค่าเหมาะที่สุดแบบหลาย

วัตถุประสงค์, อัลกอริทึมเชิงวิวัฒนาการ, อัลกอริทึมเชิงพันธุกรรม

นางสาว สุวิภา สชีวะผาสุก: การค้นหาสถาปัตยกรรมโครงข่ายประสาทเทียมโดยใช้
อัลกอริทึมเชิงวิวัฒนาการแบบหลายวัตถุประสงค์ อาจารย์ที่ปรึกษาวิทยานิพนธ์หลัก : ผู้ช่วย
ศาสตราจารย์ ดร. ยุทธนา เจวจินดา

วิทยานิพนธ์ฉบับนี้นำเสนอการค้นหาโครงข่ายประสาทเทียมของ Vision transformer โดยใช้อัลกอริทึมการหาค่าเหมาะที่สุดเชิงวิวัฒนาการแบบหลายวัตถุประสงค์ (Evolutionary multi-objective optimization) และศึกษาผลของตัวดำเนินการครอสโอเวอร์สองรูปแบบ ได้แก่ การครอสโอเวอร์แบบสองจุด (Two-point crossover) และการครอสโอเวอร์แบบสม่ำเสมอ (Uniform crossover) เพื่อประเมินผลต่อความหลากหลายของกลุ่มคำตอบในประชากรที่ได้ โดยประยุกต์ใช้อัลกอริทึมเชิงวิวัฒนาการแบบหลายวัตถุประสงค์ ได้แก่ อัลกอริทึม NSGA-II ร่วมกับการเข้ารหัสโครงสร้างแบบบล็อก (Block-based encoding) สำหรับแบบจำลอง Vision transformer ซึ่งช่วยให้สามารถสำรวจโครงสร้างของแบบจำลองได้อย่างยืดหยุ่นและมีประสิทธิภาพ นอกจากนี้ยังได้วิเคราะห์ผลของตัวดำเนินการทางพันธุกรรมต่อคุณภาพของแบบจำลองอย่างครอบคลุม และผลการทดลองบนชุดข้อมูล CIFAR-10, CIFAR-100 และ ImageNet (10 คลาส) แสดงให้เห็นถึงความสามารถของวิธีที่นำเสนอในการสร้างแบบจำลองที่มีความสมดุลระหว่างขนาดของแบบจำลองและความแม่นยำในการจำแนกรูปภาพ พร้อมทั้งมีการเปรียบเทียบกับแบบจำลองที่มีอยู่ในปัจจุบัน

640920028 : Major (ELECTRICAL AND COMPUTER ENGINEERING)

Keyword : Vision Transformer, Non-Dominated Sorting Genetic Algorithm II, Multi-Objective Optimization, Crossover, Genetic algorithm

MISS Suvipa SACHEEWAPASUK : Neural Architecture Search using Multi-Objective Evolutionary Algorithms Thesis advisor : Assistant Professor Yutana Jewajinda, Ph.D.

This thesis presents an evolutionary multi-objective optimization approach for designing Vision transformer architectures, focusing on the impact of two genetic crossover operators—two-point crossover and uniform crossover—on the diversity and quality of solutions in the population. The proposed method applies the NSGA-II algorithm in conjunction with a block-based encoding strategy that enables flexible and effective exploration of various ViT configurations. By adopting configurable building blocks, the method allows the search process to construct diverse architectures tailored to multiple objectives, particularly model size and classification accuracy. The study includes a comprehensive analysis of the influence of crossover operators on the performance and diversity of evolved models, offering practical insights for future research in neural architecture search. Experiments were conducted on three image classification datasets: CIFAR-10, CIFAR-100, and a 10-class subset of ImageNet, and the results demonstrate that the proposed approach effectively discovers models that balance compactness and accuracy, which are also compared with existing models to assess performance.

กิตติกรรมประกาศ

ขอขอบคุณผู้ช่วยศาสตราจารย์ ดร.ยุพธนา เจวจินดา อาจารย์ที่ปรึกษา ที่กรุณาให้คำแนะนำ และชี้แนะแนวทางในการทำวิทยานิพนธ์ตลอดระยะเวลาอย่างสม่ำเสมอ และให้กำลังใจตลอดระยะเวลาการทำวิทยานิพนธ์ฉบับนี้ ทำให้สามารถดำเนินงานทั้งหมดจนสำเร็จลุล่วงไปได้ด้วยดี

ขอขอบคุณรองศาสตราจารย์ ดร.ชูเกียรติ สอดศรี ประธานกรรมการสอบ และผู้ช่วยศาสตราจารย์ ดร.จิรัฏฐ์ เหมือนชู ผู้ทรงคุณวุฒิภายใน ที่กรุณาให้ข้อเสนอแนะในการตรวจทานและแก้ไขข้อบกพร่อง ทำให้สามารถพัฒนาวิทยานิพนธ์ให้สมบูรณ์และสำเร็จลุล่วงไปได้ด้วยดี

ขอขอบคุณรองศาสตราจารย์ ดร.ยรรยงค์ พันธุ์สวัสดิ์ อาจารย์ประจำหลักสูตรเทคโนโลยีดิจิทัลทางการแพทย์ มหาวิทยาลัยวลัยลักษณ์ ที่ให้เกียรติเป็นผู้ทรงคุณวุฒิภายนอกในการสอบวิทยานิพนธ์ และให้คำแนะนำที่เป็นประโยชน์อย่างยิ่งต่อการพัฒนาวิทยานิพนธ์ฉบับนี้

ขอบคุณคณะวิศวกรรมศาสตร์และเทคโนโลยีอุตสาหกรรม ภาควิชาวิศวกรรมไฟฟ้า มหาวิทยาลัยศิลปากร ที่สนับสนุนด้านสถานที่ อุปกรณ์ และโอกาสในการศึกษาค้นคว้า

สุดท้ายนี้ขอขอบคุณครอบครัวและเพื่อนๆทุกคนที่คอยให้กำลังใจเสมอมา ความสำเร็จของวิทยานิพนธ์ฉบับนี้เกิดขึ้นได้จากแรงสนับสนุนของทุกคนที่กล่าวมาทั้งหมด และผู้วิจัยหวังเป็นอย่างยิ่งว่า วิทยานิพนธ์ฉบับนี้จะมีประโยชน์แก่ผู้ที่สนใจไม่มากนักน้อย หากมีข้อผิดพลาดประการใด ผู้วิจัยขออภัยมา ณ ที่นี้ด้วย และยินดีที่จะน้อมรับคำแนะนำจากทุกท่านที่ได้เข้ามาศึกษา เพื่อเป็นประโยชน์ในการพัฒนาวิจัยต่อไป

สุวิภา สชีวมาสุก

สารบัญ

	หน้า
บทคัดย่อภาษาไทย	ง
บทคัดย่อภาษาอังกฤษ	จ
กิตติกรรมประกาศ.....	ฉ
สารบัญ.....	ช
สารบัญตาราง.....	ญ
สารบัญภาพ	ฎ
บทที่ 1 บทนำ	13
1.1 ความเป็นมาและความสำคัญของปัญหา	13
1.2 วัตถุประสงค์ของงานวิจัย	15
1.3 สมมติฐานของงานวิจัย	15
1.4 ขอบเขตของการวิจัย	15
1.5 ความจำกัดของการวิจัย	16
1.6 ประโยชน์ที่คาดว่าจะได้รับ.....	16
บทที่ 2 ทฤษฎีและงานวิจัยที่เกี่ยวข้อง	17
2.1 การค้นหาสถาปัตยกรรมโครงข่ายประสาทเทียม (Neural architecture search)	17
2.1.1 โครงข่ายประสาทเทียมเชิงลึก (Deep neural network)	17
2.1.1.1 Convolutional neural network (CNN)	17
2.1.1.2 Deep belief network (DBN).....	18
2.1.1.3 Recurrent neural network (RNN).....	18
2.1.1.4 Generative adversarial network (GAN)	19
2.1.2 NAS-DNN.....	20

2.2 อัลกอริทึมเชิงวิวัฒนาการ (Evolutionary algorithms, EA)	21
2.2.1 อัลกอริทึมเชิงพันธุกรรม (Genetic algorithm).....	22
2.3 การหาค่าเหมาะที่สุดแบบหลายวัตถุประสงค์ (Multi-objective optimization, MOO)	24
2.4 การหาค่าเหมาะที่สุดแบบหลายวัตถุประสงค์เชิงวิวัฒนาการ (Evolutionary multi-objective optimization, EMO)	26
2.5 Non-dominated sorting genetic algorithm-ii (NSGAI).....	27
2.6 Vision transformer (ViT)	31
2.7 การเข้ารหัสโครโมโซมสำหรับ NAS	33
2.8 ตัวดำเนินการทางพันธุกรรมที่ใช้ใน NAS	34
วรรณกรรมที่เกี่ยวข้องกับการค้นหาสถาปัตยกรรมโครงข่ายประสาทเทียมโดยใช้อัลกอริทึมเชิงวิวัฒนาการแบบหลายวัตถุประสงค์	36
บทที่ 3 วิธีการดำเนินงานวิจัย	43
3.1 รวบรวมข้อมูลและกำหนดขอบเขตของงาน	43
3.1.1 การกำหนดเครื่องมือและโปรแกรมที่ใช้ในการพัฒนาอัลกอริทึม	43
3.1.2 กำหนดชุดข้อมูลที่ใช้ในการทดสอบและประเมินประสิทธิภาพ	43
3.1.3 การกำหนดขอบเขตของงานวิจัย	43
3.2 ศึกษาทฤษฎีที่เกี่ยวข้องกับการประยุกต์ใช้อัลกอริทึมเชิงวิวัฒนาการในการหาค่าที่เหมาะสมแบบหลายวัตถุประสงค์	44
3.2.1 การศึกษาทฤษฎีเกี่ยวกับการหาค่าเหมาะที่สุดแบบหลายวัตถุประสงค์	44
3.2.2 การศึกษาการประยุกต์ใช้อัลกอริทึมเชิงวิวัฒนาการในการแก้ปัญหา NAS	44
3.3 ออกแบบอัลกอริทึมการหาค่าเหมาะที่สุดแบบหลายวัตถุประสงค์สำหรับปัญหา NAS และทำการประเมินประสิทธิภาพอัลกอริทึม รวมถึงการประเมินประสิทธิภาพของกระบวนการผ่านการทดลองและการวิเคราะห์ผลลัพธ์	44
3.3.1 ภาพรวมของกระบวนการที่เสนอ	45
3.3.2 การเข้ารหัสโครโมโซมเพื่อแทนสถาปัตยกรรมของแบบจำลอง	45

3.3.4 การประยุกต์ใช้ NSGA-II กับกระบวนการค้นหา	48
3.3.4.1 การประเมินค่าความเหมาะสมและการคัดเลือกด้วย NSGA-II	48
3.3.4.2 การสร้างลูกหลาน	49
3.3.4.2.1 การครอสโอเวอร์ (Crossover)	49
3.3.4.2.2 การกลายพันธุ์ (Mutation)	52
3.3.4.3 การประเมินผลและเกณฑ์การหยุดการทำงาน	52
3.3.4.4 การเปรียบเทียบกับอัลกอริทึมค้นหาแบบ NAS ที่มีอยู่แล้ว.....	53
3.4 สรุปผลการดำเนินงานวิจัยและจัดทำรายงานวิทยานิพนธ์	53
บทที่ 4 ผลการทดลอง.....	54
4.1 การตั้งค่าการทดลอง.....	54
4.2 ผลการทดลอง	54
4.3 การฝึกสอนขั้นสุดท้ายและการเปรียบเทียบกับแบบจำลองต้นฉบับ	63
4.4 อภิปรายผลและสรุปผลการทดลอง	67
บทที่ 5 สรุปและข้อเสนอแนะ	68
5.1 สรุปผลการวิจัย.....	68
5.2 ข้อเสนอแนะ	69
รายการอ้างอิง	70
ประวัติผู้เขียน	72

สารบัญตาราง

	หน้า
ตารางที่ 1 ไฮเปอร์พารามิเตอร์ทั่วไปของแบบจำลองการเรียนรู้เชิงลึก	20
ตารางที่ 2 พารามิเตอร์ที่สามารถกำหนดได้ในการเข้ารหัสสถาปัตยกรรม ViT	48
ตารางที่ 3 แสดงโครโมโซมในประชากรรุ่นที่ 64 ขณะใช้การครอสโอเวอร์แบบสองจุด	61
ตารางที่ 4 แสดงโครโมโซมในประชากรรุ่นที่ 64 ขณะใช้การครอสโอเวอร์แบบยูนิฟอร์ม	62
ตารางที่ 5 แสดงโครโมโซมที่ถูกเลือกจากการครอสโอเวอร์แบบสองจุด ที่ผ่านการฝึกแบบละเอียด จำนวน 500 epochs บนชุดข้อมูล CIFAR-10	63
ตารางที่ 6 แสดงโครโมโซมที่ถูกเลือกจากการครอสโอเวอร์แบบยูนิฟอร์ม ที่ผ่านการฝึกแบบ ละเอียดจำนวน 500 epochs บนชุดข้อมูล CIFAR-10	63
ตารางที่ 7 แสดงโครโมโซมที่ถูกเลือกจากการครอสโอเวอร์แบบสองจุด ที่ผ่านการฝึกแบบละเอียด จำนวน 500 epochs บนชุดข้อมูล CIFAR-100	64
ตารางที่ 8 แสดงโครโมโซมที่ถูกเลือกจากการครอสโอเวอร์แบบยูนิฟอร์ม ที่ผ่านการฝึกแบบ ละเอียดจำนวน 500 epochs บนชุดข้อมูล CIFAR-100	64
ตารางที่ 9 แสดงโครโมโซมที่ถูกเลือกจากการครอสโอเวอร์แบบสองจุด ที่ผ่านการฝึกแบบละเอียด จำนวน 500 epochs บนชุดข้อมูล IMAGENET (10 คลาส)	65
ตารางที่ 10 แสดงโครโมโซมที่ถูกเลือกจากการครอสโอเวอร์แบบยูนิฟอร์ม ที่ผ่านการฝึกแบบ ละเอียดจำนวน 500 epochs บนชุดข้อมูล IMAGENET (10 คลาส)	65
ตารางที่ 11 การเปรียบเทียบผลการฝึกแบบจำลองในงานวิจัยกับแบบจำลอง ViT ต้นแบบ บนชุด ข้อมูล (CIFAR-10)	66
ตารางที่ 12 เปรียบเทียบผลลัพธ์การครอสโอเวอร์แบบ Two-point และ Uniform	67

สารบัญภาพ

	หน้า
ภาพที่ 1 โครงสร้างโครงข่ายประสาทเทียมชนิด Convolutional neural network (CNN)	18
ภาพที่ 2 โครงสร้างโครงข่ายประสาทเทียมชนิด Deep belief network (DBN).....	18
ภาพที่ 3 โครงสร้างโครงข่ายประสาทเทียมชนิด Recurrent neural network (RNN)	19
ภาพที่ 4 โครงสร้างโครงข่ายประสาทเทียมชนิด Generative adversarial network.....	19
ภาพที่ 5 กระบวนการพื้นฐานของอัลกอริทึมเชิงวิวัฒนาการ	21
ภาพที่ 6 ตัวอย่างการเข้ารหัสประชากรเริ่มต้นในรูปแบบไบนารี	22
ภาพที่ 7 การแลกเปลี่ยนยีนระหว่างพ่อแม่	23
ภาพที่ 8 การกลายพันธุ์ (Mutation) ก่อนและหลัง	23
ภาพที่ 9 ความสัมพันธ์ระหว่าง Decision space และ Objective space	24
ภาพที่ 10 แนวคิด Pareto-optimal	25
ภาพที่ 11 ตัวอย่างของ Pareto optimal front (POF) สำหรับวัตถุประสงค์ $f_1(x)$ และ $f_2(x)$	26
ภาพที่ 12 กระบวนการของอัลกอริทึมการค้นหา	27
ภาพที่ 13 Pareto-optimal solution (Non-dominated).....	28
ภาพที่ 14 ขั้นตอนของอัลกอริทึม NSGA-II.....	29
ภาพที่ 15 Pseudo-code ของอัลกอริทึม NSGA-II	30
ภาพที่ 16 สถาปัตยกรรมของ Vision transformer (ViT) สำหรับการจำแนกภาพ	31
ภาพที่ 17 ภาพรวมของกระบวนการ NSGA-Net	37
ภาพที่ 18 โครงสร้างของบล็อกพื้นฐานที่ใช้สำหรับ Encoder	38
ภาพที่ 19 ตัวอย่างโครโมโซมที่ได้จากการสุ่มลำดับบล็อกและพารามิเตอร์	39
ภาพที่ 20 ภาพแสดงแนวคิดของกระบวนการครอสโอเวอร์แบบขยายความยาวเสมือน (Virtually extending crossover).....	41

ภาพที่ 21 ตัวอย่างของกระบวนการกลายพันธุ์ (Mutation)	42
ภาพที่ 22 แผนภาพแสดงกระบวนการค้นหาสถาปัตยกรรมแบบหลายวัตถุประสงค์	45
ภาพที่ 23 องค์ประกอบของแบบจำลอง Vision transformer ที่ใช้ในการเข้ารหัสโครโมโซม	45
ภาพที่ 24 ตัวอย่างโครโมโซมที่แทนสถาปัตยกรรมของแบบจำลอง Vision transformer	46
ภาพที่ 25 โครงสร้างและการกำหนดโครงสร้างในการเข้ารหัสโครโมโซม	47
ภาพที่ 26 ตัวอย่างโครโมโซมที่แทนสถาปัตยกรรม	47
ภาพที่ 27 การครอสโอเวอร์แบบสองจุด (Two-point crossover)	50
ภาพที่ 28 การครอสโอเวอร์แบบยูนิฟอร์ม (Uniform crossover)	51
ภาพที่ 29 การกลายพันธุ์ (Mutation)	52
ภาพที่ 30 แสดง Pareto front ของประชากรในแต่ละรุ่นที่ใช้การครอสโอเวอร์แบบสองจุด	55
ภาพที่ 31 แสดง Pareto front ของประชากรในแต่ละรุ่นที่ใช้การครอสโอเวอร์ยูนิฟอร์ม	55
ภาพที่ 32 แสดง Pareto front ของอันดับที่ 1 และ 2 ของประชากรในรุ่นที่ 1, 32 และ 64 ขณะใช้การครอสโอเวอร์แบบจุดสองจุด	57
ภาพที่ 33 แสดง Pareto front ของอันดับที่ 1 และ 2 ของประชากรในรุ่นที่ 1, 32 และ 64 ขณะใช้การครอสโอเวอร์แบบยูนิฟอร์ม	58
ภาพที่ 34 การเปรียบเทียบ Pareto front ของประชากรในรุ่นที่ 32	59
ภาพที่ 35 การเปรียบเทียบ Pareto front ของประชากรในรุ่นที่ 64	59

บทที่ 1

บทนำ

1.1 ความเป็นมาและความสำคัญของปัญหา

โครงข่ายประสาทเทียมเชิงลึก (Deep neural network, DNN) เป็นหนึ่งในเทคนิคการเรียนรู้ของเครื่องที่ทรงพลังที่สุดในด้านการเรียนรู้จากข้อมูลที่มีความซับซ้อนสูงและมีปริมาณมาก เนื่องจากมีสถาปัตยกรรมหลากหลายประเภทที่เหมาะสมกับข้อมูลแต่ละรูปแบบ ตัวอย่างเช่น Convolutional neural network (CNN), Recurrent neural network (RNN) เป็นต้น แต่เป็นที่ทราบกันดีว่าประสิทธิภาพของ DNN นั้นจะขึ้นอยู่กับทั้งสถาปัตยกรรมของแบบจำลองและพารามิเตอร์ของแบบจำลอง [1] ในทางปฏิบัติการค้นหาแบบจำลอง DNN ที่เหมาะสมที่สุดเป็นเรื่องที่ท้าทายอย่างมากในการแก้ปัญหาที่กำหนด เนื่องจากสอดคล้องกับการแก้ปัญหาที่มีขนาดใหญ่และมีความซับซ้อนในลักษณะ Non-convex และ Black-box โดยวิธีดั้งเดิมที่ได้รับการออกแบบเองมักจะนำไปสู่ประสิทธิภาพที่ต่ำ เนื่องจากขาดความรู้จากผู้เชี่ยวชาญและขาดแรงงานมนุษย์ที่เพียงพอในการเลือกค่าที่ดีที่สุดในการสร้าง DNN แบบอัตโนมัติ สถาปัตยกรรมที่ใช้ในปัจจุบันส่วนใหญ่ได้รับการพัฒนาโดยผู้เชี่ยวชาญที่เป็นมนุษย์ ซึ่งเป็นกระบวนการที่ใช้เวลานานและเกิดข้อผิดพลาดได้ง่าย [2] ดังนั้นจึงมีการเสนอวิธีการค้นหาสถาปัตยกรรมโครงข่ายประสาทเทียม (Neural architecture search, NAS) เพื่อออกแบบสถาปัตยกรรมแบบจำลองโดยอัตโนมัติ แต่การค้นหาเหล่านี้ต้องใช้ทรัพยากรในการคำนวณมหาศาล ซึ่งงานก่อนหน้าที่เกี่ยวข้องกับ NAS มักจะกำหนดเป็นปัญหาแบบวัตถุประสงค์เดียว (Single-objective optimization problem) อย่างไรก็ตาม ในการประยุกต์ใช้การเรียนรู้เชิงลึกส่วนใหญ่ ไม่เพียงแต่ต้องการประสิทธิภาพที่สูงอย่างเดียว แต่ยังต้องการลดจำนวนทรัพยากรลง ไม่ว่าจะเป็นขนาดของแบบจำลอง, เวลาในการอนุมานและปริมาณพลังงานที่ใช้ ทำให้สถาปัตยกรรมหลายตัวล่าสุดได้รับการออกแบบโดยมีจุดมุ่งหมายเพื่อลดการใช้ทรัพยากรในขณะที่ยังคงประสิทธิภาพการทำนายที่สูง [2]

ในช่วงไม่กี่ปีที่ผ่านมา การค้นหาสถาปัตยกรรมโครงข่ายประสาทเทียม (Neural architecture search, NAS) ได้รับความสนใจจากงานวิจัยอย่างมาก และมีวิธีการเรียนรู้เชิงลึกจำนวนมากได้รับการออกแบบโดยใช้วิธีการ NAS เนื่องจากประสบความสำเร็จในการแสดงให้เห็นถึงความเหนือกว่าเมื่อเทียบกับการออกแบบ DNN ด้วยตัวเอง ซึ่งเทคนิคที่ได้รับความนิยมมากที่สุดสำหรับ NAS ได้แก่ อัลกอริทึมเชิงวิวัฒนาการ (Evolutionary algorithms), Reinforcement learning (RL), Bayesian optimization, Gradient descent, Neural predictors, และ Local

search [3] แม้ว่าวิธีการ NAS เหล่านี้จะสามารถนำไปใช้ในการออกแบบ DNN ได้สำเร็จ แต่ประสิทธิภาพของวิธีการ NAS ยังขึ้นอยู่กับโครงสร้างสถาปัตยกรรมที่แตกต่างกันไปตามงานที่แตกต่างกัน รวมถึงการเข้ารหัสสถาปัตยกรรมเพื่อหาค่าที่เหมาะสม ซึ่งพารามิเตอร์สถาปัตยกรรมประกอบด้วย จำนวนเลเยอร์, ความสัมพันธ์ระหว่างเลเยอร์ต่างๆ เช่น Fully connected layer และพารามิเตอร์ที่เกี่ยวข้องกับโครงสร้าง เช่น Kernel size, Channels ดังนั้นเป็นเรื่องยากที่จะออกแบบวิธีการ NAS ในการเข้ารหัสสถาปัตยกรรมที่ตายตัวและสามารถนำไปใช้ในการเรียนรู้สถาปัตยกรรมที่แตกต่างกันสำหรับแก้ปัญหาที่แตกต่างกันได้ [4] ดังนั้นเราจึงกำหนดปัญหาเหล่านี้ให้เป็นปัญหาการหาค่าเหมาะที่สุดแบบหลายวัตถุประสงค์ (Multi-objective optimization)

ปัจจุบันปัญหาส่วนใหญ่ต้องการจัดการกับวัตถุประสงค์ที่ขัดแย้งกันหลายประการแทนที่จะเป็นวัตถุประสงค์เดียว สำหรับปัญหาดังกล่าวนั้น การหาค่าเหมาะที่สุดแบบหลายวัตถุประสงค์ (Multi-objective optimization, MOO) เป็นเทคนิคที่มีประสิทธิภาพในการค้นหาชุดคำตอบที่กำหนดการแลกเปลี่ยนที่ดีที่สุดระหว่างวัตถุประสงค์ที่ขัดแย้งกัน [5] มีการนำอัลกอริทึมเชิงวิวัฒนาการแบบหลายวัตถุประสงค์ (Multi-objective evolutionary algorithms) มาใช้ ด้วยเหตุนี้ทำให้อัลกอริทึมเชิงวิวัฒนาการ (EA) และการคำนวณเชิงวิวัฒนาการ (EC) ได้รับความสนใจและค่านิยมเพิ่มขึ้นในการแก้ปัญหาการหาค่าเหมาะที่สุด เนื่องจากความสามารถที่โดดเด่นในการแก้ปัญหาการหาค่าที่เหมาะสมที่มีลักษณะ Non-convex, Nondifferentiable และ Black-box ตลอดจนความก้าวหน้าอย่างรวดเร็วในการประมวลผลที่มีประสิทธิภาพสูงในการ คำนวณ และแสดงให้เห็นถึงประสิทธิภาพที่มีแนวโน้มที่ดีและประสบความสำเร็จในการทดสอบชุดข้อมูลมาตรฐานมากมาย เช่น CIFAR-10, CIFAR-100 และ ImageNet [1] ดังนั้นงานวิจัยจำนวนมากจึงใช้อัลกอริทึมเชิงวิวัฒนาการแบบหลายวัตถุประสงค์มาใช้เพื่อค้นหาแบบจำลองที่ดีขึ้น [6]

งานวิจัยนี้มีจุดมุ่งหมายเพื่อนำเสนออัลกอริทึมเชิงวิวัฒนาการในการหาค่าเหมาะที่สุดแบบหลายวัตถุประสงค์ (Evolutionary Multi-objective optimization, EMO) ซึ่งจะสามารถช่วยให้ประมาณระนาบ Pareto front ของสถาปัตยกรรมภายใต้วัตถุประสงค์หลายวัตถุประสงค์ได้อย่างมีประสิทธิภาพ และสามารถลดจำนวนการประเมินและเพิ่มความเร็วในการค้นหา โดยจะทดสอบกับชุดข้อมูลมาตรฐาน ได้แก่ CIFAR-10, CIFAR-100 และ ImageNet และนำผลลัพธ์ที่ได้มาเปรียบเทียบกับงานวิจัยอื่นๆ เช่น งานวิจัยที่อยู่ใน [11]

1.2 วัตถุประสงค์ของงานวิจัย

งานวิจัยนี้มีวัตถุประสงค์เพื่อแก้ปัญหาค้นหาสถาปัตยกรรมโครงข่ายประสาทเทียม (NAS) โดยใช้อัลกอริทึมเชิงวิวัฒนาการแบบหลายวัตถุประสงค์ โดยมีจุดมุ่งหมายหลักของงานวิจัยคือ

- 1.2.1 เพื่อศึกษาทฤษฎีและความรู้พื้นฐานเกี่ยวกับปัญหาค้นหาสถาปัตยกรรมโครงข่ายประสาทเทียม
- 1.2.2 เพื่อศึกษาทฤษฎีและความรู้พื้นฐานเกี่ยวกับการคำนวณเชิงวิวัฒนาการ โดยเฉพาะอัลกอริทึมเชิงวิวัฒนาการแบบหลายวัตถุประสงค์ เพื่อนำมาใช้กับปัญหา NAS
- 1.2.3 เพื่อพัฒนาและนำเสนอวิธีการใหม่ในการเพิ่มประสิทธิภาพในการค้นหาสถาปัตยกรรม โดยใช้การคำนวณเชิงวิวัฒนาการแบบหลายวัตถุประสงค์

1.3 สมมติฐานของงานวิจัย

จากการศึกษาพบว่าปัญหาค้นหาสถาปัตยกรรมโครงข่ายประสาทเทียมอัตโนมัติเกี่ยวข้องกับการหาโครงสร้างโครงข่ายประสาทเทียมและพารามิเตอร์ที่เหมาะสม ทำให้นักวิจัยหันมาสนใจความสามารถอัลกอริทึมเชิงวิวัฒนาการ (EA) ในการนำมาแก้ไขปัญหานี้ จากการศึกษางานวิจัยที่ผ่านมาพบว่าปัญหาปัจจุบันนั้นมีวัตถุประสงค์หลายวัตถุประสงค์ที่ขัดแย้งกันแทนที่จะเป็นวัตถุประสงค์เดียวแบบในอดีต ทำให้การหาค่าเหมาะที่สุดแบบหลายวัตถุประสงค์เชิงวิวัฒนาการ ได้รับความนิยมในการแก้ไขปัญห NAS ดังนั้นในงานวิจัยนี้จึงนำอัลกอริทึมการหาค่าเหมาะที่สุดแบบหลายวัตถุประสงค์เชิงวิวัฒนาการ (EMO) มาใช้ในการแก้ไขปัญห NAS เนื่องจากมีประสิทธิภาพในการค้นหาค่าตอบที่กำหนดการแลกเปลี่ยนที่ดีที่สุดระหว่างวัตถุประสงค์ที่ขัดแย้งกัน และความสามารถในการคำนวณที่มีประสิทธิภาพสูง จึงนำวิธีการนี้มาเพื่อนำเสนอวิธีการหาค่าที่เหมาะสมแบบหลายวัตถุประสงค์ในการค้นหาสถาปัตยกรรมอัตโนมัติ

1.4 ขอบเขตของการวิจัย

ขอบเขตของงานวิจัยการหาค่าเหมาะที่สุดแบบหลายวัตถุประสงค์ โดยใช้อัลกอริทึมเชิงวิวัฒนาการแบบหลายวัตถุประสงค์ มีดังนี้

1. ทำการศึกษาทฤษฎีพื้นฐานของอัลกอริทึมเชิงวิวัฒนาการและการเข้ารหัสเพื่อใช้ในการหาค่าที่เหมาะสมของสถาปัตยกรรม โดยทำการสำรวจวรรณกรรมที่เกี่ยวข้อง
2. ทำการศึกษาและหาวิธีการใหม่ในการประยุกต์ใช้อัลกอริทึมเชิงวิวัฒนาการแบบหลายวัตถุประสงค์กับการค้นหาสถาปัตยกรรมการเรียนรู้เชิงลึกจากวรรณกรรมที่เกี่ยวข้อง

3. ทำการทดสอบประสิทธิภาพกับชุดข้อมูลมาตรฐานที่มีลักษณะแตกต่างกัน เช่น CIFAR-10, CIFAR-100 และ ImageNet โดยเปรียบเทียบประสิทธิภาพกับงานวิจัยอื่นๆที่เกี่ยวข้อง เช่น [11]

1.5 ความจำกัดของการวิจัย

1. ใช้ทรัพยากรในการคำนวณสูง เนื่องจากในการออกแบบระบบนั้นต้องใช้การคำนวณเพื่อให้ได้ผลลัพธ์ที่ดีที่สุด
2. งานวิจัยนี้จะทำการจำลองการทำงานโดยใช้บนเครื่องคอมพิวเตอร์เท่านั้น
3. วิธีการใหม่ที่ได้ไม่สามารถแก้ไขปัญหาที่เหมาะสมได้สำหรับทุกปัญหา

1.6 ประโยชน์ที่คาดว่าจะได้รับ

ได้วิธีการใหม่ในการหาค่าที่เหมาะสมในการค้นหาสถาปัตยกรรมโดยใช้อัลกอริทึมเชิงวิวัฒนาการแบบหลายวัตถุประสงค์ ซึ่งมีประสิทธิภาพมากกว่าวิธีดั้งเดิม



บทที่ 2

ทฤษฎีและงานวิจัยที่เกี่ยวข้อง

บทนี้กล่าวถึงความรู้พื้นฐานและงานวิจัยที่เกี่ยวข้องกับการหาค่าที่เหมาะสมของการค้นหาสถาปัตยกรรมโครงข่ายประสาทเทียมอัตโนมัติ โดยเนื้อหาในส่วนนี้ประกอบด้วยความรู้พื้นฐานและวรรณกรรมที่เกี่ยวข้องกับอัลกอริทึมเชิงวิวัฒนาการ, การหาค่าเหมาะสมที่สุดแบบหลายวัตถุประสงค์ และการประยุกต์ใช้อัลกอริทึมเชิงวิวัฒนาการแบบหลายวัตถุประสงค์

2.1 การค้นหาสถาปัตยกรรมโครงข่ายประสาทเทียม (Neural architecture search)

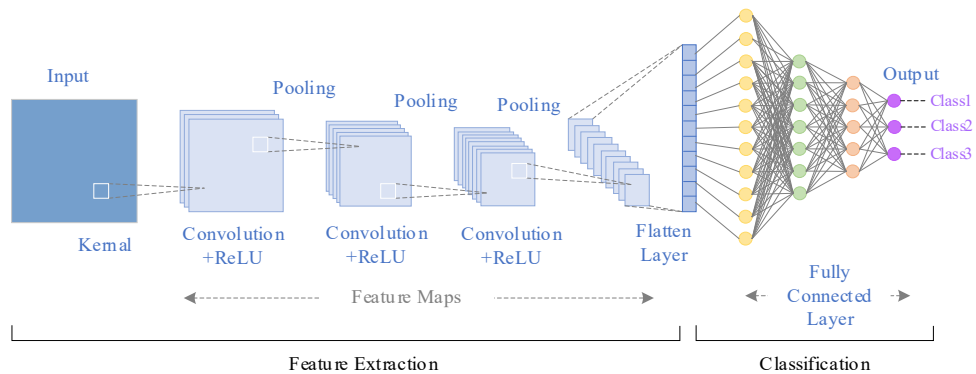
NAS เป็นกระบวนการค้นหาสถาปัตยกรรมโครงข่ายประสาทเทียมแบบอัตโนมัติ โดยวิธีการของ NAS นั้นมีประสิทธิภาพเหนือกว่าสถาปัตยกรรมที่ออกแบบด้วยตนเองในบางงาน เช่น การจัดประเภทรูปภาพ (Classification), การตรวจจับวัตถุ (Object detection), หรือการตีความบางส่วน of ภาพ (Semantic segmentation) ซึ่งสถาปัตยกรรมโครงข่ายประสาทเทียมแบบเฉพาะจะแตกต่างกันไปตามงานที่แตกต่าง และพารามิเตอร์ภายในโครงสร้างสถาปัตยกรรมจะประกอบด้วยเลเยอร์ที่แตกต่าง จึงเป็นการยากที่จะออกแบบโดยใช้แบบ manual ทำให้ต้องมีวิธีการค้นหาแบบอัตโนมัติ เนื่องจากประสิทธิภาพของ NAS นั้นจะขึ้นอยู่กับทั้งสถาปัตยกรรมของแบบจำลองและพารามิเตอร์ของแบบจำลองของ DNN

2.1.1 โครงข่ายประสาทเทียมเชิงลึก (Deep neural network)

แบบจำลองการเรียนรู้เชิงลึก มักจะใช้แบบจำลอง DNN เพื่อจัดการกับปัญหาการเรียนรู้ รวมถึงการจำแนกประเภทและการทำนาย โดยแบบจำลองที่ใช้กันอย่างแพร่หลาย สามารถแบ่งคร่าวๆ ได้ 5 หมวดหมู่ตามแบบจำลองเบื้องต้น เช่น Convolutional neural network (CNN), Deep belief network (DBN), Stacked auto encoder (SAE), Recurrent neural network (RNN), และ Generative adversarial network (GAN) เป็นต้น

2.1.1.1 Convolutional neural network (CNN)

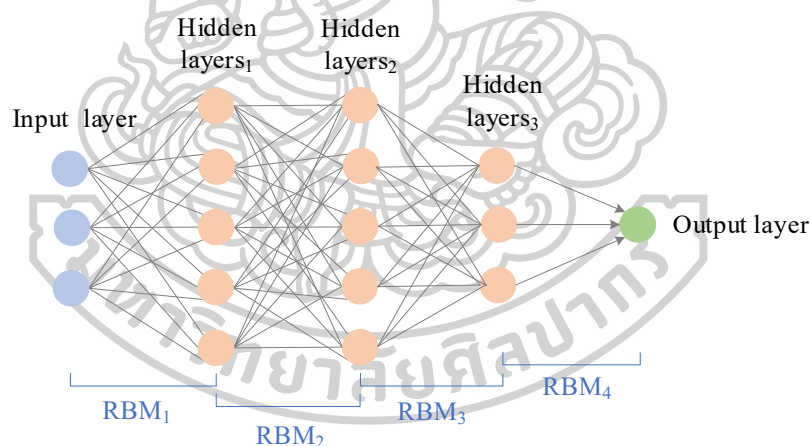
เป็นโครงสร้าง DNN ที่ใช้กันอย่างแพร่หลาย โครงสร้างภายในทำงานได้ดีกับข้อมูลประเภทรูปภาพและวิดีโอ เนื่องจากมีโครงสร้างที่เหมาะสมสำหรับการประมวลผลภาพ 2 มิติ และ 3 มิติ และมีประสิทธิภาพในการสกัดคุณสมบัติ (Features) หรือลักษณะเด่นออกมาจากรูปภาพ ซึ่งโครงสร้างของ CNN ประกอบด้วย Convolutional layers, Pooling layers และ Fully connected layers ภาพที่ 1 จะแสดงตัวอย่างโครงสร้างทั่วไปของ CNN



ภาพที่ 1 โครงสร้างโครงข่ายประสาทเทียมชนิด Convolutional neural network (CNN)

2.1.1.2 Deep belief network (DBN)

เป็น Graphical model หรือเรียกอีกอย่างว่าประเภทของ DNN ที่ประกอบด้วย Hidden units หลายๆชั้น โดยมีการเชื่อมต่อระหว่างชั้นต่างๆ แต่ไม่ใช่ระหว่างหน่วยภายในแต่ละชั้น ซึ่ง DBN จะทำงานได้ดีกับข้อมูลที่เป็น Independent features โดยโครงสร้างของ DBN จะแสดงดังภาพที่ 2

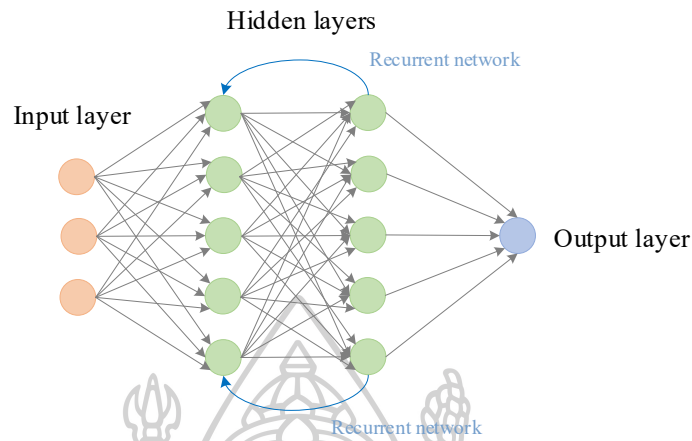


ภาพที่ 2 โครงสร้างโครงข่ายประสาทเทียมชนิด Deep belief network (DBN)

2.1.1.3 Recurrent neural network (RNN)

เป็นโครงข่ายประสาทเทียมที่มีการเรียนรู้แบบมีผู้สอน โดยชั้น Input layers, Recurrent hidden layers และ Output layers นั้นเป็นพื้นฐาน 3 ชั้นใน RNN อย่างง่าย คุณลักษณะหลักๆของ RNN คือโครงข่ายมีการเชื่อมต่อ Feedback ซึ่งแตกต่างจาก Feedforward แบบเดิม โดย Feedback loop นี้จะให้ RNN จำลองผลกระทบของส่วนก่อนหน้า ซึ่งเป็นคุณลักษณะ

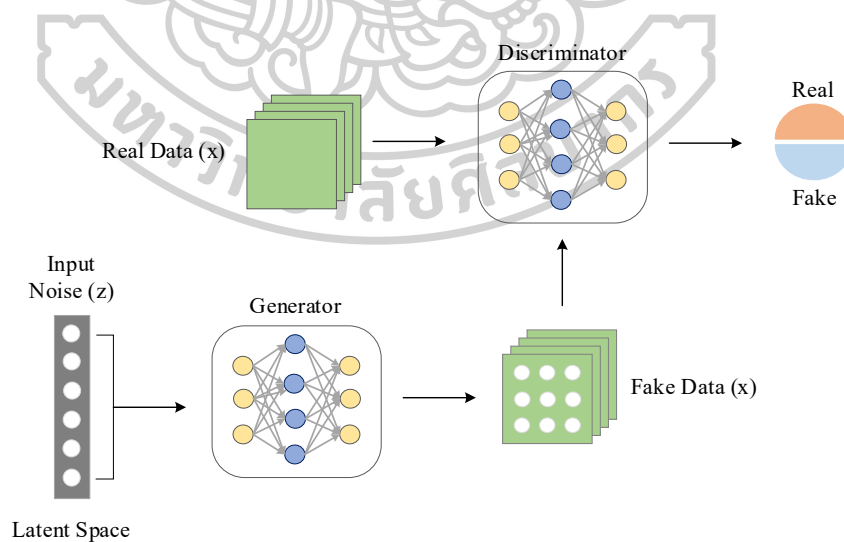
ที่สำคัญมากเมื่อกล่าวถึงแบบจำลอง Sequences ซึ่งจะทำงานได้ดีในข้อมูล Sequences เช่น Texts และ Time series โครงสร้างแบบจำลองจะแสดงดังภาพที่ 3



ภาพที่ 3 โครงสร้างโครงข่ายประสาทเทียมชนิด Recurrent neural network (RNN)

2.1.1.4 Generative adversarial network (GAN)

เป็นวิธีการเรียนรู้เชิงลึกแบบไม่มีผู้สอน เป็นสถาปัตยกรรมที่ใช้โครงข่ายประสาทเทียม 2 โครงข่าย โดยแยกโครงข่ายหนึ่งกับอีกโครงข่ายหนึ่ง เพื่อสร้างตัวแทนข้อมูลใหม่ที่สามารถส่งผ่านข้อมูลจริงได้ มีการใช้กันอย่างแพร่หลาย เช่น การสร้างภาพ การสร้างวิดีโอ และการสร้างเสียง โดยโครงสร้างของ GAN สามารถแสดงได้ดังภาพที่ 4



ภาพที่ 4 โครงสร้างโครงข่ายประสาทเทียมชนิด Generative adversarial network

นอกจากโครงสร้างของสถาปัตยกรรม DNN แต่ละชนิดแล้ว การกำหนดค่าไฮเปอร์พารามิเตอร์ยังเป็นปัจจัยสำคัญที่มีอิทธิพลต่อประสิทธิภาพของแบบจำลอง โดยที่ไฮเปอร์พารามิเตอร์ทั่วไปของแบบจำลองการเรียนรู้เชิงลึกนั้นแสดงไว้ในตารางที่ 1

ตารางที่ 1 ไฮเปอร์พารามิเตอร์ทั่วไปของแบบจำลองการเรียนรู้เชิงลึก

Model	Hyperparameters	Containing hyperparameters	Included variable types
CNN [82,83,92,93]	Convolutional layer	Number of convolutional layers, Kernel number, Filter size (width and height), Stride size (width and height), Feature map size, Convolution type, Batch normalization type, Activation function type	Integer and category
	Pooling layer	Filter size (width and height), Stride size (width and height), Pooling type	Integer and category
	Fully-connected layer	Number of neurons, Activation function type, Dropout rate	Integer, category, and continuous
DBN [9,86,87]	Hidden layer	Number of hidden layers, Neurons per layer, Activation function type, Parameters for pre-training and fine-tuning, Learning rate	Integer and category
RNN [88,89]	Hidden layer	Number of hidden layers, Neurons per layer, Number of time slot	Integer
AE [12,90]	Hidden layer	Number of hidden layers, Neurons per layer	Integer
GAN [91]	Layers in generator and discriminator	Number of iterations of each training epoch for generator and discriminator, the parameter in the loss function of generator and discriminator	Integer and real number

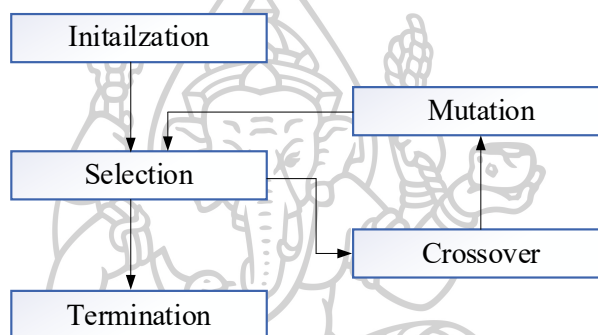
ที่มา : Zhan, Z.-H., et al. (2022) [6]

2.1.2 NAS-DNN

การได้มาซึ่งสถาปัตยกรรมที่เหมาะสมที่สุดนั้นไม่สามารถกำหนดได้โดยตรงด้วยฟังก์ชันต่อเนื่อง และไม่มีฟังก์ชันที่ชัดเจนในกระบวนการค้นหาสถาปัตยกรรมที่เหมาะสมที่สุด นอกจากนี้สถาปัตยกรรม DNN มักขึ้นอยู่กับปัญหา หากการกระจายของข้อมูลเปลี่ยนไป สถาปัตยกรรมจะต้องได้รับการออกแบบใหม่ตาม การค้นหาสถาปัตยกรรมโครงข่ายประสาทเทียม (NAS) จึงมีจุดมุ่งหมายเพื่อทำให้การออกแบบสถาปัตยกรรมของ DNN เป็นไปโดยอัตโนมัติ ได้รับการระบุว่า เป็นวิธีที่มีแนวโน้มที่ดีในการจัดการกับปัญหานี้ โดยทั่วไปประสิทธิภาพของ DNN ขึ้นอยู่กับ 2 ด้าน ได้แก่ สถาปัตยกรรมของ DNN และ ค่าน้ำหนัก (Weights) ที่เกี่ยวข้อง เมื่อทั้ง 2 ด้านนั้นเหมาะสมพร้อมๆ กัน ประสิทธิภาพของ DNN ก็คาดว่าจะมีแนวโน้มที่ดีขึ้น

2.2 อัลกอริทึมเชิงวิวัฒนาการ (Evolutionary algorithms, EA)

อัลกอริทึมเชิงวิวัฒนาการ (Evolutionary algorithms, EA) เป็นกลุ่มอัลกอริทึมที่ถูกพัฒนาขึ้นมาเพื่อแก้ปัญหาเกี่ยวกับการหาค่าที่เหมาะสมที่สุด (Optimization) หรือปัญหาการค้นหา (Search problem) ที่ได้รับแรงบันดาลใจมาจากธรรมชาติและชีววิทยา แก้ปัญหาผ่านกระบวนการที่เลียนแบบพฤติกรรมของสิ่งมีชีวิต ได้แก่ การคัดเลือก (Selection), การสืบพันธุ์ (Crossover), การกลายพันธุ์ (Mutation) โดยหลักการการทำงานพื้นฐานของอัลกอริทึมเชิงวิวัฒนาการนั้นผู้ที่อ่อนแอที่สุดจะถูกกำจัดออกไป ในขณะที่ตัวเลือกที่แข็งแกร่งกว่าและมีความเป็นไปได้ที่จะอยู่รอดนั้นถูกรักษาไว้และทำการประเมินผลอีกครั้งในการวิวัฒนาการครั้งต่อไป สามารถอธิบายได้ดังภาพที่ 5



ภาพที่ 5 กระบวนการพื้นฐานของอัลกอริทึมเชิงวิวัฒนาการ

- Initialization

เป็นกระบวนการเริ่มต้นของอัลกอริทึมเชิงวิวัฒนาการ โดยกระบวนการนี้เป็นการสร้างประชากรกลุ่มแรกโดยการสุ่มจากพื้นที่ค้นหา (Search space) ซึ่งในรหัสพันธุกรรมของประชากรจะมีตัวแปรหรือค่าคำตอบของปัญหา เช่น ค่าไฮเปอร์พารามิเตอร์ในสถาปัตยกรรมที่ต้องการหาค่าที่เหมาะสม

- Selection

เมื่อสร้างประชากรแล้ว สมาชิกของประชากรต้องได้รับการประเมินตาม Fitness function โดยคำนวณหาความเหมาะสมของสมาชิกทั้งหมด และเลือกสมาชิกที่มีค่า Fitness สูง โดยประชากรส่วนหนึ่งถูกเลือกเพื่อนำไปผสมพันธุ์ (Crossover) และกลายพันธุ์ (Mutation) ทำให้มีการแลกเปลี่ยนรหัสพันธุกรรม เกิดเป็นประชากรใหม่ที่มีคุณสมบัติแตกต่างไปจากประชากรเดิม เนื่องจากมีรหัสพันธุกรรมที่แตกต่างไปจากเดิม

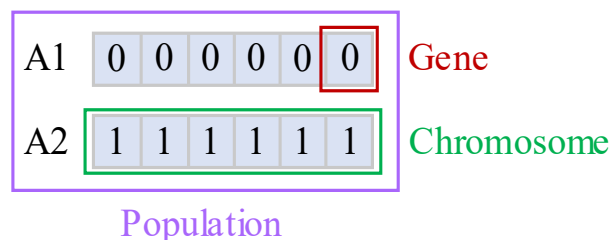
- Termination

กระบวนการนี้เป็นกระบวนการทำซ้ำขั้นตอนที่ 1 และ 2 เพื่อกำจัดประชากรที่อ่อนแอที่สุดออกไปเรื่อยๆ แล้วแทนที่ด้วยประชากรที่แข็งแกร่งกว่าที่ได้จากกระบวนการคัดเลือก ทำให้ประชากรวิวัฒนาการรุ่นถัดมามีคุณสมบัติและความสามารถในการอยู่รอดที่สูงขึ้น โดยกระบวนการนี้จะทำซ้ำจนกว่าจะถึงเงื่อนไขที่กำหนด

2.2.1 อัลกอริทึมเชิงพันธุกรรม (Genetic algorithm)

อัลกอริทึม GA เป็นอัลกอริทึมที่มีการค้นหาตามแนวคิดของการคัดเลือกโดยธรรมชาติและพันธุกรรม ใน GA จะมีกลุ่มหรือประชากรของคำตอบ (Solution) ที่เป็นไปได้สำหรับปัญหาที่กำหนด โดยคำตอบเหล่านี้จะถูกรวมตัวกัน (Recombination) และกลายพันธุ์ ทำให้เกิดลูกใหม่ขึ้นมา และกระบวนการนี้ทำซ้ำไปมาในหลายๆรุ่น ซึ่ง GA จะทำการค้นหา Optimal solution ของปัญหาการหาค่าเหมาะที่สุด (Optimization problem) โดยจัดการกับคำตอบต่างๆพร้อมกัน ซึ่งแต่ละคำตอบจะได้รับการเข้ารหัสเป็น String ที่จะแสดงถึง Candidate solution (เรียกว่า “โครโมโซม”) ที่มีค่าฟังก์ชันวัตถุประสงค์ของโครโมโซมแต่ละตัว เรียกว่า ค่า Fitness โดยแต่ละ String หรือแต่ละโครโมโซมนั้นจะถือว่าเป็น Individual หรือคำตอบหนึ่งคำตอบ และกลุ่มของ Individual จะเรียกว่า ประชากร (Population) โดยอัลกอริทึม GA นั้นมีด้วยกัน 5 ขั้นตอนหลักๆ ได้แก่ Initial population, Fitness function, Selection, Crossover, Mutation

Initial population : กระบวนการเริ่มต้นของชุดคำตอบหรือชุด Individual ที่เรียกว่า Population และแต่ละ Individual เป็นคำตอบของปัญหาที่ต้องการแก้ไข ซึ่ง Individual มีลักษณะเฉพาะด้วยชุดของพารามิเตอร์ เรียกว่า ยีนส์ โดยยีนส์จะถูกรวมเข้ากับ String เพื่อสร้างโครโมโซม และใน GA นั้นชุดของยีนส์แต่ละ Individual จะถูกแสดงโดยใช้ String ในรูปแบบไบนารี ตัวอย่างดังภาพที่ 6

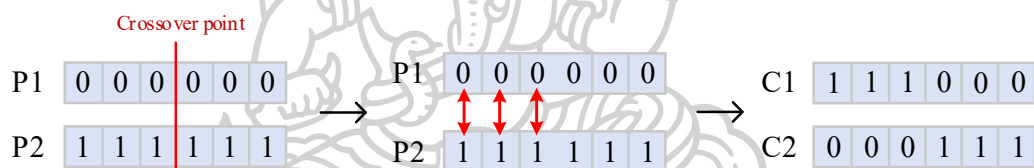


ภาพที่ 6 ตัวอย่างการเข้ารหัสประชากรเริ่มต้นในรูปแบบไบนารี

Fitness function : เป็นตัวกำหนดว่าแต่ละ Individual มีความเหมาะสมแค่ไหน (ability ของ Individual ในการแข่งขันกับ Individual อื่น) โดยจะให้ Fitness score แต่ละ Individual ซึ่งความน่าจะเป็นที่ Individual จะถูกเลือกเพื่อสืบพันธุ์ (Reproduction) จะขึ้นอยู่กับ Fitness score

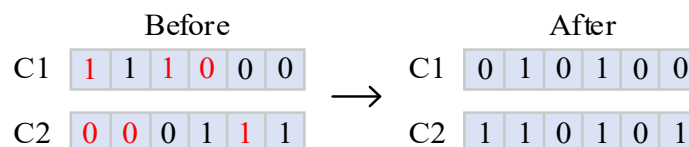
Selection : แนวคิดของขั้นตอนการคัดเลือกคือ การคัดเลือก Individual ที่เหมาะสมที่สุด โดย Individual 2 คู่ (พ่อและแม่) จะถูกเลือกโดยพิจารณาจาก Fitness scores ของแต่ละ Individual และเลือก Individual ที่มีความเหมาะสมที่สุด ซึ่ง Individual ที่มีค่า Fitness สูง จะมีโอกาสได้รับการคัดเลือกในการนำไปสืบพันธุ์มากกว่า

Crossover : เป็นขั้นตอนสำคัญที่สุดใน GA โดยจุด Crossover ที่ถูกเลือกจากการสุ่มภายในยีนส์จะแสดงดังรูปที่ 7 (ซ้าย) และลูก (Offspring) ที่เกิดขึ้น ถูกสร้างโดยการแลกเปลี่ยนยีนส์จากพ่อแม่กันเองจนกว่าจะถึง Crossover point จากนั้นลูกที่ได้ใหม่ (New offspring) จะถูกเพิ่มเข้าไปในประชากรดังภาพที่ 7 (ขวา)



ภาพที่ 7 การแลกเปลี่ยนยีนส์ระหว่างพ่อแม่

Mutation : ยีนส์บางตัวที่เกิดขึ้นใน New offspring อาจจะถูกกลายพันธุ์โดยความน่าจะเป็นในการสุ่มที่ต่ำ หมายความว่า Bit บางตัวใน Bit string สามารถ Flip ได้ ในการกลายพันธุ์จะเกิดขึ้นเพื่อรักษาความหลากหลายในประชากร แสดงดังภาพที่ 8



ภาพที่ 8 การกลายพันธุ์ (Mutation) ก่อนและหลัง

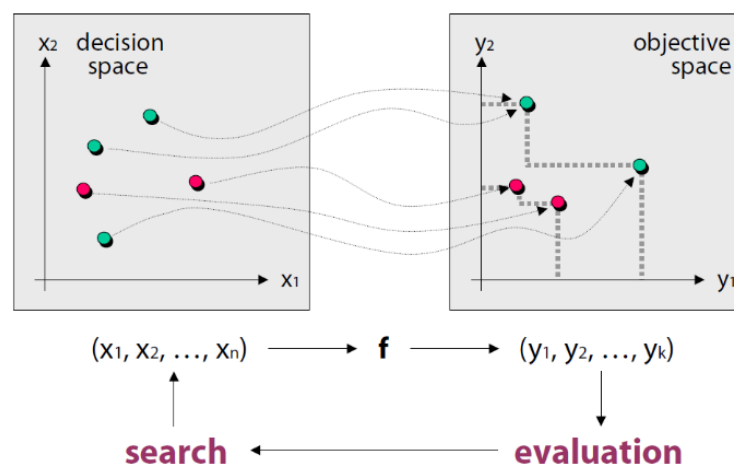
Termination : ขั้นตอนสุดท้ายในกระบวนการของอัลกอริทึมเชิงพันธุกรรม โดยขั้นตอนนี้จะสิ้นสุดลงหากจำนวนประชากรทั้งหมดไม่ได้ผลิตลูกหลานที่แตกต่างจากรุ่นก่อนอย่างมีนัยสำคัญ

โดยการเข้ารหัสของ GA จะมี 2 แบบหลักๆ ได้แก่ การเข้ารหัสแบบไบนารี โดยจะแทนโครโมโซมด้วยเลขไบนารีที่เป็นจำนวนเต็มเท่านั้น จะใช้งานได้ดีกับปัญหาที่เกี่ยวกับโครงสร้างของโครงข่ายประสาทเทียม เช่น หาจำนวนชั้นหรือฟิลเตอร์ จากนั้นนำโครโมโซมมาผ่านการ Crossover และ Mutation ส่วนวิธีที่สองเป็นวิธีการเข้ารหัสด้วยเลขจำนวนจริง ซึ่งจะแทนโครโมโซมด้วยเลขจำนวนจริง เช่นปัญหาในการหาค่า Weight และค่า Learning rate ที่เป็นเลขทศนิยม

2.3 การหาค่าเหมาะที่สุดแบบหลายวัตถุประสงค์ (Multi-objective optimization, MOO)

Multi-objective optimization (MOO) คือกระบวนการค้นหาคำตอบที่ดีที่สุดภายใต้เงื่อนไขที่มีมากกว่าหนึ่งวัตถุประสงค์ ซึ่งอาจสอดคล้องกันหรือขัดแย้งกัน เช่น ความแม่นยำของโมเดลกับขนาดของโมเดล โดยเมื่อวัตถุประสงค์ขัดแย้งกัน จะไม่สามารถปรับให้ดีขึ้นทุกด้านพร้อมกันได้ จึงต้องหา “ทางเลือกที่สมดุล” ที่เรียกว่า Pareto optimal

การที่วัตถุประสงค์ขัดแย้งกัน ทำให้เกิดการเชื่อมโยงกันระหว่าง Decision space และ Objective space โดย Decision space ซึ่งเป็นชุดของตัวแปรที่เราควบคุมได้ เช่น พารามิเตอร์ของโมเดล แต่ละจุดจะถูกส่งผ่านฟังก์ชันวัตถุประสงค์ ไปยัง Objective space ซึ่งแสดงผลลัพธ์ เช่น ความแม่นยำของแบบจำลอง และขนาดของแบบจำลอง โดยอัลกอริทึมพยายามจะค้นหาค่า x (ตัวแปรที่ควบคุมได้) แล้วนำ x ที่ได้นั้นไปแทนในฟังก์ชันวัตถุประสงค์ จากนั้นไปประเมินแล้วทำการค้นหาใหม่ แสดงดังภาพที่ 9 ซึ่งเป้าหมายของ MOO คือการค้นหาค่าที่ดีที่สุด ใน Decision space และ ประเมินผล (Evaluation) ใน Objective space เพื่อเข้าใกล้ Pareto front ให้มากที่สุด

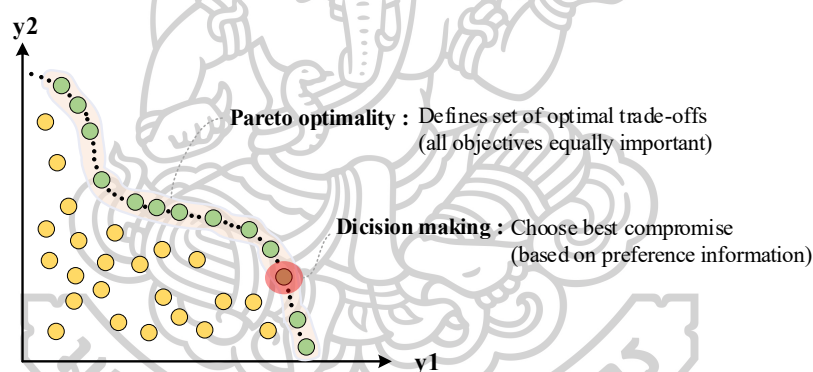


ภาพที่ 9 ความสัมพันธ์ระหว่าง Decision space และ Objective space

ที่มา : Zitzler, E., et al. (2004) [7]

ในปัญหาเชิงปฏิบัติส่วนใหญ่มักมีวัตถุประสงค์มากกว่าหนึ่งอย่าง และวัตถุประสงค์เหล่านั้นมักขัดแย้งกัน เช่น ความแม่นยำกับขนาดโมเดล ดังนั้นแนวคิดของชุด Pareto-optimal (ภาพที่ 10) ซึ่งเป็นจุดที่เหมาะสมที่สุดในแง่ที่ว่าไม่มีคำตอบใดสามารถปรับให้ดีขึ้นในวัตถุประสงค์หนึ่งโดยไม่ทำให้วัตถุประสงค์อื่นแย่ลง ทำให้ Pareto-optimal ถูกนำมาใช้เป็นวิธีการแก้ปัญหาสำหรับการหาค่าที่เหมาะสมที่สุดแบบหลายวัตถุประสงค์ โดยจุดที่สนใจจะเป็นจุดที่อยู่บนระนาบ Pareto optimality หรือ Pareto front ซึ่งเป็นระนาบที่ให้ความสำคัญของวัตถุประสงค์เท่าๆกันที่แสดงถึงชุดของคำตอบที่ไม่ถูกรอบงำ (Non-dominated) หมายความว่า ไม่มีจุดใดในชุดนี้ที่แยกว่าอีกจุดหนึ่งในทุกวัตถุประสงค์ จากนั้นต้องตัดสินใจว่าจะเลือกจุดไหนบนระนาบที่เป็นคำตอบที่ดีที่สุด ซึ่งเป็นการเลือกหนึ่งคำตอบจาก Pareto front ตามเงื่อนไขหรือความต้องการเฉพาะ

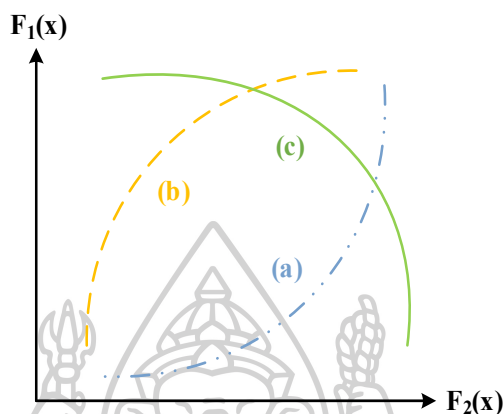
ดังนั้นกระบวนการ MOO จะเริ่มจากการสร้าง Pareto front และสิ้นสุดที่การตัดสินใจเลือกคำตอบที่ดีที่สุดสำหรับบริบทของปัญหานั้น ๆ



ภาพที่ 10 แนวคิด Pareto-optimal

ในปัญหาการหาค่าที่เหมาะสมที่สุดแบบวัตถุประสงค์เดียว ความเหนือกว่าของคำตอบนั้นสามารถกำหนดได้โดยการเปรียบเทียบค่าฟังก์ชันวัตถุประสงค์ แต่ในปัญหาการหาค่าที่เหมาะสมที่สุดแบบหลายวัตถุประสงค์นั้น คำตอบที่ดีจะถูกกำหนดโดยแนวคิด Dominance มีหลักการคือ คำตอบ x_1 Dominate คำตอบ x_2 ต่อเมื่อ คำตอบ x_1 ไม่แยกว่า x_2 ในทุกวัตถุประสงค์ และ คำตอบ x_1 ดีกว่า x_2 อย่างเห็นได้ชัดอย่างน้อยหนึ่งวัตถุประสงค์ และในการหา Pareto optimal solution จะพิจารณาชุดคำตอบที่เป็น Non-dominated solution ซึ่งคือชุดของคำตอบทั้งหมดที่ไม่ได้ถูก Dominate โดยคำตอบใดๆ โดยชุด Non-dominated ของพื้นที่การตัดสินใจ (Decision space) เรียกว่า Pareto-optimal set โดยขอบเขตที่กำหนดโดยชุดของจุดที่เชื่อมต่อกันจาก Pareto optimal set เรียกว่า Pareto optimal front (POF)

ตัวอย่างของ POF สำหรับวัตถุประสงค์ $f_1(x)$ และ $f_2(x)$ จะแสดงดังภาพที่ 11 โดย 11(a) จะเป็น POF ที่กำหนดให้ $f_1(x)$ มีค่าน้อยที่สุดและ $f_2(x)$ มีค่ามากที่สุด ส่วน 11(b) กำหนดให้ $f_1(x)$ มีค่ามากที่สุดและ $f_2(x)$ มีค่าน้อยที่สุด และ 11(c) กำหนดให้ $f_1(x)$ มีค่ามากที่สุดและ $f_2(x)$ มีค่ามากที่สุด



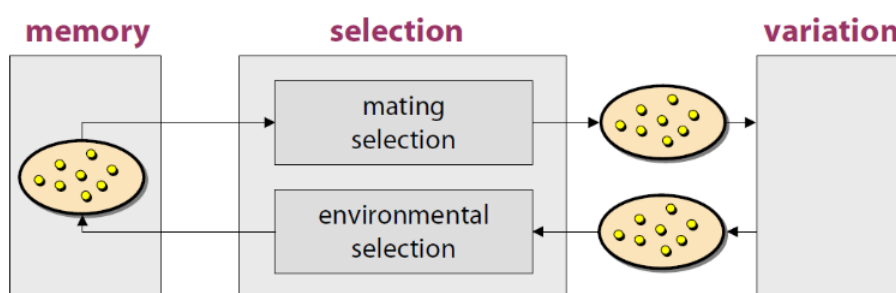
ภาพที่ 11 ตัวอย่างของ Pareto optimal front (POF) สำหรับวัตถุประสงค์ $f_1(x)$ และ $f_2(x)$

2.4 การหาค่าเหมาะที่สุดแบบหลายวัตถุประสงค์เชิงวิวัฒนาการ (Evolutionary multi-objective optimization, EMO)

อัลกอริทึมเชิงวิวัฒนาการสำหรับการหาค่าเหมาะที่สุดแบบหลายวัตถุประสงค์ หรือ EMO เป็นแนวทางที่ได้รับความนิยมในการแก้ปัญหา NAS เนื่องจากสามารถจัดการกับปัญหาการหาค่าที่เหมาะสมแบบหลายวัตถุประสงค์โดยการใช้การคำนวณเชิงวิวัฒนาการกับวัตถุประสงค์ที่มีความขัดแย้งกัน โดยอาศัยหลักการของอัลกอริทึมเชิงวิวัฒนาการ เช่น การคัดเลือก การผสมพันธุ์ และการกลายพันธุ์ เพื่อค้นหาหากลุ่มคำตอบที่มีความหลากหลาย และตอบสนองวัตถุประสงค์ที่ขัดแย้งกันได้ดี การใช้วิธีเหล่านี้ช่วยให้สามารถค้นหาชุดคำตอบที่อยู่ใกล้กับ Pareto front ได้อย่างมีประสิทธิภาพ และสามารถนำไปใช้ในการตัดสินใจเลือกแบบจำลองที่เหมาะสมที่สุด ทำให้ EMO ได้กลายเป็นสาขาการวิจัยและการประยุกต์ใช้ที่ได้รับความนิยมและมีประโยชน์ในการออกแบบโครงข่ายประสาทเทียม (Neural networks)

ในกระบวนการของอัลกอริทึมการค้นหา ดังแสดงภาพที่ 12 จะประกอบไปด้วย 3 ขั้นตอนหลัก ได้แก่ หน่วยความจำ (Memory) ขั้นตอนการคัดเลือก (Selection) และขั้นตอนการ Variation โดยในขั้นแรกเป็นขั้นตอน Memory จะเก็บประชากรในรุ่นปัจจุบัน เช่น โครงสร้างของแบบจำลองแต่ละตัว จากนั้นเข้าสู่ขั้นตอนการคัดเลือก (Selection) ซึ่งแบ่งเป็น 2 ส่วนคือ Mating selection

เป็นการเลือกพ่อแม่เพื่อนำไปสร้างประชากรลูก ส่วน Environmental selection จะเป็นการคัดเลือกประชากรรวมจากพ่อแม่และลูก เพื่อเก็บเฉพาะโซลูชันที่ดีที่สุดไว้เป็นรุ่นถัดไป จากนั้นโซลูชันที่ถูกเลือกจะเข้าสู่กระบวนการ Variation เช่น การ ผสมพันธุ (Crossover) หรือ กลายพันธุ์ (Mutation) เพื่อสร้างประชากรรุ่นใหม่ กระบวนการนี้จะทำวนซ้ำไปเรื่อย ๆ จนกว่าจะครบเงื่อนไขที่กำหนดไว้ เช่น จำนวนรุ่น (Generation)



ภาพที่ 12 กระบวนการของอัลกอริทึมการค้นหา

ที่มา : Zitzler, E., et al. (2004) [7]

แนวคิดสำคัญของอัลกอริทึมเชิงวิวัฒนาการแบบหลายวัตถุประสงค์ คือ การผลักดันให้โซลูชันเข้าใกล้ Pareto front ให้มากที่สุด โดยใช้กลยุทธ์ เช่น การจัดอันดับและการวัดความกระจายเพื่อคัดเลือกโซลูชันที่ "แข็งแรง" และมีศักยภาพในการพัฒนาไปสู่คำตอบที่ดียิ่งขึ้นในรุ่นถัดไป โดยจะสนใจคำตอบที่อยู่บริเวณ Pareto optimality หรือ Pareto front ที่เป็นระนาบที่ให้ความสำคัญของ 2 วัตถุประสงค์เท่าๆกัน แล้วเลือกว่าคำตอบไหนเป็นจุดที่ดีที่สุดใ้ในระนาบนั้นๆ เพื่อเลือกมาเป็นพ่อแม่ก่อนที่จะนำไปผสมพันธุ์ ดังนั้นอัลกอริทึมจะพยายามเลือกประชากรที่มุ่งเข้าหาแนวเส้นระนาบ Pareto เพราะมีความเชื่อว่าประชากรที่อยู่บนเส้นระนาบจะทำให้อัลกอริทึมลูู่เข้าสู่คำตอบที่ดีได้ จากนั้นอัลกอริทึมจะเลือกคำตอบที่มีการกระจายตัวพอกันบนระนาบเพื่อรักษาความหลากหลายในการค้นหา และที่สำคัญคือต้องเป็นคำตอบที่ Non-dominatated เพราะเชื่อว่าคำตอบจะมีประสิทธิภาพ

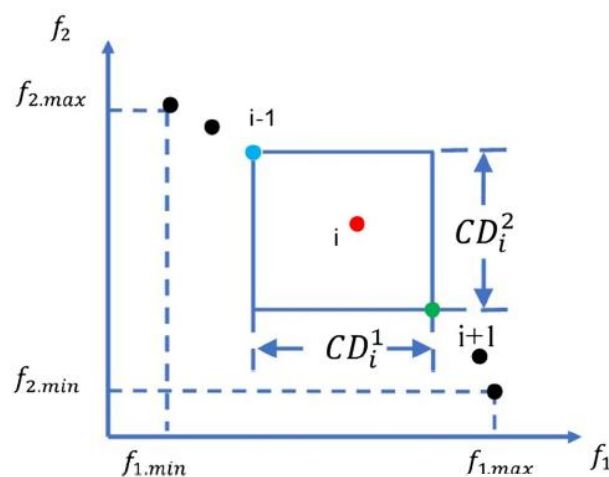
2.5 Non-dominated sorting genetic algorithm-ii (NSGAI)

อัลกอริทึม NSGA-II เป็นหนึ่งในขั้นตอนของ EMO ที่ได้รับความนิยม ถูกพัฒนาขึ้นมาสำหรับการค่าเหมาะที่สุดแบบหลายวัตถุประสงค์โดยมีแนวคิดมาจากอัลกอริทึมเชิงพันธุกรรม (Genetic algorithm), การจัดเรียง (Sorting), แนวคิด Dominance และการกระจายตัวของกลุ่มคำตอบ ซึ่งอัลกอริทึมพยายามค้นหา Pareto-optimal โดย NSGA-II นั้นใช้หลักการ Elitist คือมีการเก็บตัวที่ดี

ที่สุดไว้เพื่อเอาไปใช้ในการผสมพันธุ้ในรุ่นต่อไป และเน้นที่การจัดเรียงแบบไม่ครอบคลุม (Non-dominated sorting) และการรักษาความหลากหลายของคำตอบเอาไว้โดย Crowding distance [8]

ขั้นตอนของ NSGA-II สำหรับการหาค่าเหมาะที่สุดกล่าวคือ ในขั้นแรกประชากรเริ่มต้นจะถูกสร้างจากการสุ่ม จากนั้นสมาชิกหรือกลุ่มคำตอบทั้งหมดในประชากรจะถูกประเมิน ต่อมาชุดคำตอบจะถูกจัดอันดับโหลชั้นโดยใช้แนวคิด Non-dominated sorting เพื่อจำแนกโหลชั้นตามระดับความเหนือกว่า (Dominance rank) ซึ่งจะได้ชุดระนาบ $F_1, F_2, F_3, \dots, F_n$ ตามลำดับ โดยเริ่มจากการเลือกคำตอบเพื่อเข้าสู่รุ่นถัดไปจากระนาบที่มีลำดับดีที่สุดก่อน เช่นชุดคำตอบที่อยู่ใน F_1 จะเป็นคำตอบที่มีประสิทธิภาพมากกว่าคำตอบที่อยู่ใน F_2 จึงทำให้มีโอกาสถูกเลือกมากกว่า เลือกจนกระทั่งได้จำนวนเท่ากับขนาดประชากร และถ้าหากชุดคำตอบที่เข้าไปเกินจำนวนที่กำหนด จะมีการใช้ Crowding distance เพื่อวัดระยะห่างระหว่างโหลชั้น และเลือกเฉพาะโหลชั้นที่กระจายตัวดีที่สุดเพื่อคงไว้ในรุ่นถัดไป ผลลัพธ์ที่ได้คือประชากรรุ่นใหม่ซึ่งจะถูกนำไปใช้ในรอบการวิวัฒนาการถัดไป กระบวนการนี้จะทำซ้ำไปเรื่อยๆจนกว่าจะครบจำนวนรุ่น (Generation) ที่กำหนดไว้ล่วงหน้า

สำหรับ Crowding distance แต่ละคำตอบ ใน Pareto set นั้นจะถูกคำนวณอัตราส่วนของระยะห่างระหว่างคำตอบทางด้านขวาและด้านซ้ายของคำตอบนั้นๆ ดังแสดงในภาพที่ 13 และกำหนดสมการทางคณิตศาสตร์ ดังสมการที่ 1,2,3



ภาพที่ 13 Pareto-optimal solution (Non-dominated)

ที่มา : Rostamian, A., et al. (2019) [9]

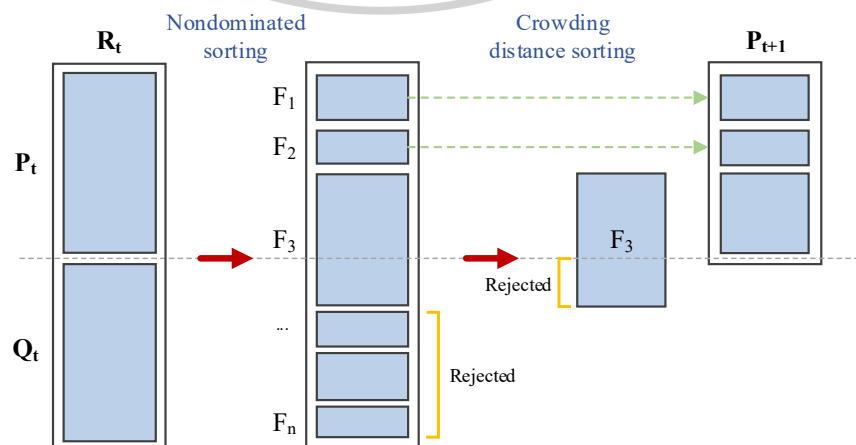
$$CD_i^1 = \frac{f_1^{i+1} - f_1^{i-1}}{f_1^{max} - f_1^{min}} \quad (1)$$

$$CD_i^2 = \frac{f_2^{i+1} - f_2^{i-1}}{f_2^{max} - f_2^{min}} \quad (2)$$

$$Total\ Crowding\ Distance = CD_i^1 + CD_i^2 \quad (3)$$

ในขั้นตอนนี้ ค่าตอบที่อยู่ตำแหน่งขอบเขตของ Objective space จะถูกกำหนดค่า Crowding distance เป็นอนันต์ (Infinity) และแต่ละคำตอบจะถูกกำหนด 2 อย่างคือ Rank และ ค่า Crowding distance ซึ่งการรวมกันของ 2 องค์ประกอบนี้ใช้กับกระบวนการคัดเลือก (Selection) ที่ช่วยให้ NSGA-II นั้นรักษาคุณภาพและความหลากหลายของคำตอบ โดยการเปรียบเทียบความหนาแน่นกับจำนวนประชากรของคำตอบนั้น คำตอบที่มี Rank ต่ำกว่าและมี Crowding distance มากกว่า จะถูกเลือกสำหรับรุ่นถัดไป ดังนั้นการเปรียบเทียบ Crowding distance ถูกนำมาใช้เพื่อจัดเรียงคำตอบใน Pareto set โดยกระบวนการคัดเลือกของ NSGA-II จะเป็น Binary tournament

ในขั้นตอนการผสมพันธุ์ พ่อแม่ (Parents) ที่ถูกเลือกจะถูกผสมพันธุ์เพื่อสร้างลูกหลานใหม่ (Offspring) โดยการกลายพันธุ์ (Mutation) และการผสมพันธุ์ (Crossover) และประชากรใหม่ที่ได้ (Offspring) จะถูกนำไปรวมกับประชากรพ่อแม่ ทำให้ประชากรใหม่ถูกสร้างขึ้น จากนั้นประชากรใหม่ที่เกิดจากการรวมกันจะถูกจัดเรียงตามแนวคิด Dominance เพื่อกำหนดระนาบของชุด Non-dominated fronts ให้ได้ Rank และ Crowding distance โดยจะทำซ้ำจนกว่าจะครบเงื่อนไขที่กำหนด ดังแสดงในภาพที่ 14



ภาพที่ 14 ขั้นตอนของอัลกอริทึม NSGA-II

จุดประสงค์ในการคัดเลือกพ่อแม่ คือสมาชิกหรือคำตอบที่เหมาะสมที่สุดควรมีโอกาสรอดสูงกว่า ดังนั้นจึงเป็นไปได้สูงที่สมาชิกที่เหมาะสมที่สุดจะอยู่รอดและนำไปผสมพันธุ์ในรุ่นต่อไปได้ดีกว่า แม้ว่าคำตอบที่อ่อนแอกว่าจะมีโอกาสรอดน้อยกว่า แต่ก็มีโอกาสที่จะปรับปรุงโดยอยู่ภายใต้การกลายพันธุ์ในรุ่นต่อไปในอนาคต ใน NSGA-II มีกระบวนการการเลือกทัวร์นาเมนต์ (Tournament selection) ที่เป็นกระบวนการหลักๆ ในการผสมพันธุ์เพื่อเร่งกระบวนการค้นหา Pareto front สุดท้าย

Tournament selection ใน NSGA-II ในขั้นแรกนั้นจะมีการสุ่มคำตอบจากประชากร แต่ละคำตอบจะถูกเปรียบเทียบกับกันตาม Rank ก่อนแล้วจึงเปรียบเทียบกับกันตาม Crowding distance โดยถ้าหากคำตอบ มี Front ที่ต่างกัน คำตอบที่มี Rank ต่ำกว่าจะถูกเลือก แต่ถ้าหากคำตอบถูกเลือกจาก Front เดียวกัน จะเลือกจากคำตอบที่มี Crowding distance มากกว่า แล้วนำคำตอบที่ได้ไปสร้างประชากรลูกหลานต่อไป

Algorithm 1 : NSGA-II algorithm

Q_t and P_t - Offspring and the population at generation t
 F_t - Ranked sets at generation t
 $|X|$ - size of set X
 N - size of the population

- 1 Initialize population P_1
- 2 FitnessEvaluation(P_1)
- 3 $Q_t = \emptyset$
- 4 **For** $t = 1$ to MaxGen **Do**
- 5 $R_{temp} = P_t \cup Q_t$
- 6 $Q_{t+1} = \text{offspring-generating}(R_{temp})$
- 7 FitnessEvaluation(Q_{t+1})
- 8 $R_t = P_t \cup Q_{t+1}$
- 9 $F_t = \text{fast-Non-domination-sort}(R_t)$
- 10 $P_{t+1} = \emptyset$ and $i = 1$
- 11 **while** $|P_{t+1}| + |F_i| \leq N$ **do**
- 12 crowding-distance-assignment(F_i)
- 13 $P_{t+1} = P_{t+1} \cup F_i$
- 14 $i = i+1$
- 15 **end**
- 16 Sort(F_i, n)
- 17 $P_{t+1} = P_{t+1} \cup F_i[1:N - |P_{t+1}|]$
- 18 Fitness evaluation of P_{t+1}
- 19 $t = t+1$

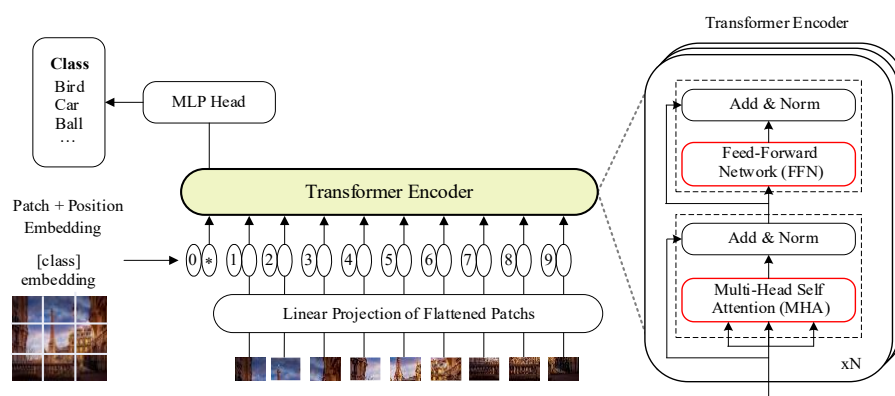
ภาพที่ 15 Pseudo-code ของอัลกอริทึม NSGA-II

ภาพที่ 15 Pseudo-code ของอัลกอริทึม NSGA-II แสดงขั้นตอนหลักของการพัฒนาแต่ละรุ่นของประชากรโดยอาศัยแนวคิด Non-dominated sorting และ Crowding distance โดยในบรรทัดที่ 6 มีการสร้างประชากรลูกหลานผ่านกระบวนการ Crossover และ Mutation ซึ่งมีบทบาทสำคัญทั้งด้านการสำรวจ (Exploration) และการใช้ประโยชน์ (Exploitation) จากพื้นที่ค้นหา จากนั้นคำตอบจะถูกจัดอันดับด้วย Non-dominated sorting (บรรทัดที่ 9) และเลือกคำตอบที่ดีที่สุดผ่าน Elitist selection (บรรทัดที่ 10–17) โดยใช้ Rank และ Crowding distance ในการพิจารณาร่วมกัน [8]

2.6 Vision transformer (ViT)

ในช่วงไม่กี่ปีที่ผ่านมา แบบจำลอง Transformer ซึ่งมีจุดเริ่มต้นจากงานในด้านการประมวลผลภาษาธรรมชาติ (Natural language processing, NLP) ได้รับความสนใจและถูกนำมาประยุกต์ใช้กับปัญหาด้านการประมวลผลภาพอย่างแพร่หลาย โดยเฉพาะอย่างยิ่ง Vision transformer (ViT) ซึ่งเป็นงานแรกที่เสนอการนำสถาปัตยกรรมของ Transformer มาใช้กับภาพโดยตรง โดยไม่พึ่งพาโครงสร้าง Convolutional แบบเดิม [10]

ViT ได้รับการเสนอครั้งแรกในงานวิจัย [11] ซึ่งเสนอแนวคิดใหม่ในการประมวลผลภาพโดยแปลงภาพให้เป็นลำดับของแพตช์ย่อย และใช้ Self-attention เพื่อเรียนรู้บริบทภายในภาพ เช่นเดียวกับที่ Transformer ใช้กับลำดับคำใน NLP โดยแบ่งภาพอินพุตขนาด $H \times W \times C$ ออกเป็น Patch ขนาดเล็ก $P \times P$ และแปลง Patch แต่ละก้อนเป็นเวกเตอร์ผ่าน เลเยอร์ฉายเชิงเส้น (Linear projection) ก่อนที่จะป้อนเข้าสู่ Encoder ของ Transformer ซึ่งประกอบด้วยหลายชั้นและใช้กลไก Self-attention เพื่อเรียนรู้ความสัมพันธ์ระหว่าง Patch ต่างๆ และผลลัพธ์สุดท้ายของแบบจำลองจะถูกนำไปใช้ในการจำแนกประเภทภาพ



ภาพที่ 16 สถาปัตยกรรมของ Vision transformer (ViT) สำหรับการจำแนกภาพ

ภาพที่ 16 แสดงสถาปัตยกรรมของ Vision transformer แสดงขั้นตอนการแปลงภาพเป็นลำดับของ patch และการประมวลผลผ่าน Transformer encoder จนได้ผลลัพธ์สำหรับการจำแนกประเภท

1. Image splitting

ภาพขนาด $H \times W \times CH$ (เช่น $224 \times 224 \times 3$) จะถูกแบ่งออกเป็น patch ย่อย ขนาด $P \times P$ (เช่น 16×16) โดยไม่ซ้อนทับกัน ทำให้ได้จำนวน patch ทั้งหมดเท่ากับ $N = \frac{HW}{P^2}$

2. การแบ่งภาพเป็นแพตช์ (Patch embedding)

Patch แต่ละก้อนจะถูกแปลงเป็นเวกเตอร์ Flatten แล้วผ่านเลเยอร์ฉายเชิงเส้น (Linear projection) เพื่อให้ได้เวกเตอร์ความยาวคงที่ D (เช่น 768) ซึ่งเป็นขนาดของ Embedding space

3. การเพิ่มตำแหน่ง (Position embedding)

เนื่องจาก Transformer ไม่สามารถรับรู้ลำดับตำแหน่งข้อมูลได้โดยตรง จึงต้องมีการเพิ่มเวกเตอร์แสดงตำแหน่ง (Positional embedding) เข้าไปในแต่ละแพตช์เวกเตอร์ เพื่อเก็บข้อมูลเชิงพื้นที่ (Spatial location)

4. Class token

มีการเติม [CLS] token ไปที่ต้นของลำดับเวกเตอร์ ซึ่งจะเป็นตัวแทนสำหรับการเรียนรู้ และใช้เป็นตัวแทนของภาพทั้งหมดสำหรับการกึ่งในการจำแนกประเภท

5. Transformer encoder

ลำดับเวกเตอร์ทั้งหมด (รวม [CLS] token และ Patch embeddings) จะถูกป้อนเข้าสู่ Transformer encoder ซึ่งประกอบด้วย L ชั้น โดยแต่ละชั้นมีโครงสร้างดังนี้

- Multi-head self-attention (MHSA)
- Feedforward neural network (FFN)
- LayerNorm และ Residual connection

6. MLP Head สำหรับการจำแนกประเภท

หลังจากผ่าน Encoder แล้ว เวกเตอร์ของ [CLS] token จะถูกนำไปผ่าน MLP head ซึ่งมักจะเป็น Fully connected layer หรือ Linear classifier เพื่อทำนาย Class label

จุดเด่นของ ViT คือการเรียนรู้ความสัมพันธ์ระยะไกลระหว่างส่วนต่าง ๆ ของภาพผ่านกลไก Self-attention โดยไม่จำเป็นต้องใช้ Convolutional filter ซึ่งช่วยให้แบบจำลองสามารถเข้าใจโครงสร้างทั่วทั้งภาพได้ดียิ่งขึ้น อย่างไรก็ตาม การฝึก ViT ให้ได้ประสิทธิภาพเทียบเท่าหรือดีกว่า CNN จำเป็นต้องใช้ข้อมูลจำนวนมากและการประมวลผลสูง

2.7 การเข้ารหัสโครโมโซมสำหรับ NAS

ในการประยุกต์ใช้อัลกอริทึมเชิงวิวัฒนาการกับกระบวนการค้นหาโครงข่ายประสาทเทียม จำเป็นต้องมีการแปลงสถาปัตยกรรมของแบบจำลองให้เป็นรูปแบบที่อัลกอริทึมสามารถจัดการได้ ซึ่งเรียกว่า “การเข้ารหัสโครโมโซม (Chromosome encoding)” โดยที่โครโมโซมนี้ทำหน้าที่เป็นตัวแทนของสถาปัตยกรรมแบบจำลองหนึ่งตัวในประชากร (Population)

รูปแบบของการเข้ารหัสโครโมโซมใน NAS มีความหลากหลาย ขึ้นอยู่กับลักษณะของแบบจำลองที่ใช้ในการค้นหา โดยรูปแบบของการเข้ารหัสจะแตกต่างกันไปตามประเภทของแบบจำลองและขอบเขตการค้นหา (Search space) ที่กำหนดไว้ โดยทั่วไปสามารถแบ่งออกเป็นประเภทหลักได้ดังนี้

1. Binary encoding

เป็นการแทนแต่ละลักษณะของสถาปัตยกรรมด้วยบิต 0 หรือ 1 เช่น การมีหรือไม่มีเลเยอร์บางประเภท หรือการเปิด/ปิดการเชื่อมต่อระหว่างเลเยอร์ ซึ่งนิยมในกรณีที่ Search space มีโครงสร้างคงที่

2. Integer encoding

ใช้จำนวนเต็มเพื่อแทนค่าพารามิเตอร์ที่เกี่ยวข้องกับโครงสร้างของแบบจำลอง เช่น จำนวนฟิลเตอร์ใน Convolutional layer, ขนาด Kernel, ประเภทของ Block, หรือรูปแบบการเชื่อมต่อระหว่างเลเยอร์

3. Mixed encoding

ในหลายกรณีมีการใช้รูปแบบเข้ารหัสแบบผสม โดยโครโมโซมอาจประกอบด้วยส่วนที่เป็นจำนวนเต็มและไบนารี (เช่น การเปิด/ปิดพีเจอร์บางประเภท) เพื่อเพิ่มความยืดหยุ่นและรองรับความซับซ้อนของสถาปัตยกรรม

การเลือกใช้รูปแบบการเข้ารหัสที่เหมาะสมกับลักษณะของปัญหาและโครงสร้างของแบบจำลองที่ต้องการค้นหาจะช่วยเพิ่มประสิทธิภาพของกระบวนการสืบค้น และสามารถนำไป

ประยุกต์ใช้ร่วมกับกลยุทธ์ทางพันธุกรรม เช่น Crossover และ Mutation ได้อย่างมีประสิทธิภาพ โดยแนวคิดดังกล่าวได้รับการศึกษาไว้อย่างครอบคลุมในงานของ [12] ซึ่งจำแนกรูปแบบการแทนโครงสร้างของแบบจำลองไว้หลากหลายประเภท เช่น การเข้ารหัสแบบไบนารี [13] และ การเข้ารหัสแบบจำนวนเต็มที่สามารถนำมาใช้ร่วมกับอัลกอริทึมเชิงวิวัฒนาการได้อย่างมีประสิทธิภาพ

2.8 ตัวดำเนินการทางพันธุกรรมที่ใช้ใน NAS

การค้นหาโครงสร้างของโครงข่ายประสาทเทียมด้วยวิธีการค้นหาเชิงวิวัฒนาการ จำเป็นต้องอาศัย ตัวดำเนินการทางพันธุกรรม (Genetic operators) ซึ่งมีบทบาทสำคัญในการสร้างประชากรใหม่ในแต่ละรอบของกระบวนการวิวัฒนาการ โดยเป้าหมายคือการปรับปรุงคุณลักษณะของโครงสร้างแบบจำลองอย่างต่อเนื่องเพื่อให้ได้สถาปัตยกรรมที่มีประสิทธิภาพสูงขึ้น

ตัวดำเนินการหลักที่นิยมใช้ใน NAS ได้แก่ Crossover, Mutation และ Selection ซึ่งแต่ละตัวเป็นหัวใจสำคัญของกลยุทธ์การวิวัฒนาการและมีหน้าที่และกลไกเฉพาะที่ส่งผลต่อความหลากหลายของประชากรและคุณภาพของคำตอบ

1. การครอสโอเวอร์ (Crossover)

เป็นกระบวนการที่สร้างโครโมโซมลูกหลาน (Offspring) โดยการผสมโครโมโซมจากพ่อแม่ โดยมีจุดประสงค์เพื่อสืบทอดคุณสมบัติที่ดีจากแต่ละฝ่าย วิธีการทั่วไปที่ได้รับความนิยมได้แก่

- การครอสโอเวอร์หนึ่งจุด (One-point crossover) : สุ่มตำแหน่งเดียวบนโครโมโซม จากนั้นแลกเปลี่ยนข้อมูลระหว่างพ่อแม่หลังตำแหน่งดังกล่าว วิธีนี้เหมาะกับโครโมโซมที่มีโครงสร้างเรียงลำดับ
- การครอสโอเวอร์สองจุด (Two-point crossover) : ทำการสุ่มสองตำแหน่งบนโครโมโซม แล้วทำการแลกเปลี่ยนเฉพาะช่วงกลางระหว่างตำแหน่งทั้งสอง วิธีนี้ช่วยเพิ่มความหลากหลายมากกว่าแบบหนึ่งจุด
- การครอสโอเวอร์แบบสม่ำเสมอ (Uniform crossover) : พิจารณาแต่ละตำแหน่งของโครโมโซมอย่างเป็นอิสระ โดยสุ่มเลือกว่าจะใช้จากพ่อหรือแม่ในแต่ละตำแหน่งตามความน่าจะเป็น

การเลือกประเภท Crossover ต้องคำนึงถึงลักษณะของโครโมโซม เนื่องจากอาจมีผลต่อโครงสร้างของแบบจำลอง เช่น ความลึก จำนวนเลเยอร์ หรือความเข้ากันได้ระหว่างบล็อกต่างๆ

2. การกลายพันธุ์ (Mutation)

เป็นกระบวนการที่ปรับเปลี่ยนค่าในบางตำแหน่งของโครโมโซมอย่างสุ่ม เพื่อเพิ่มความหลากหลายทางพันธุกรรมในประชากร และหลีกเลี่ยงการติดอยู่ในค่าที่ไม่เหมาะสม (Local optimum) โดยรูปแบบของ Mutation ขึ้นอยู่กับการเข้ารหัสของโครโมโซม เช่น

- การสุ่มเปลี่ยนค่าจำนวนเต็ม (Integer) เป็นค่าอื่นในช่วงที่กำหนด
- การสลับตำแหน่งของยีน
- การเปิดหรือปิดบางพีเจอร์ (ในกรณี Binary encoding)

มักใช้ Mutation rate เป็นพารามิเตอร์กำกับว่าควรเปลี่ยนแปลงกี่ตำแหน่งในโครโมโซม เช่น 0.1 หมายถึง 10% ของยีนจะถูกสุ่มเปลี่ยน

3. การคัดเลือก (Selection)

เป็นกลไกในการเลือกโครโมโซมที่เหมาะสมจากประชากรเดิมมาเป็นพ่อแม่ของประชากรรุ่นถัดไป โดยมีจุดมุ่งหมายเพื่อรักษาความได้เปรียบของโครงสร้างที่มีประสิทธิภาพ ซึ่งวิธี Selection ที่นิยมใน NAS ได้แก่

- การคัดเลือกแบบการแข่งขัน (Tournament selection) : สุ่มโครโมโซมจำนวนหนึ่งแล้วเลือกตัวที่ดีที่สุด
- การคัดเลือกแบบวงล้อรูเล็ต (Roulette wheel selection) : ใช้ความน่าจะเป็นตามค่าความเหมาะสม (Fitness)
- การจัดอันดับแบบไม่ถูกครอบงำ (Non-dominated sorting) : ใช้สำหรับกรณีที่มีหลายวัตถุประสงค์ (Multi-objective) โดยจัดลำดับจาก Front ที่ไม่ถูกครอบงำ

ตัวดำเนินการทางพันธุกรรมถือเป็นหัวใจสำคัญของการประยุกต์ใช้อัลกอริทึมเชิงวิวัฒนาการกับ NAS โดยทำหน้าที่สร้างประชากรที่หลากหลายและพัฒนาไปสู่โครงสร้างแบบจำลองที่มีศักยภาพสูงขึ้นอย่างต่อเนื่อง การออกแบบและกำหนดพารามิเตอร์ของตัวดำเนินการเหล่านี้จึงมีผลต่อประสิทธิภาพของระบบ NAS อย่างมาก

วรรณกรรมที่เกี่ยวข้องกับการค้นหาสถาปัตยกรรมโครงข่ายประสาทเทียมโดยใช้อัลกอริทึมเชิงวิวัฒนาการแบบหลายวัตถุประสงค์

1. Nsga-net: Neural architecture search using multi-objective genetic algorithm, Z. Lu, I. Whalen, V. Boddeti, Y. Dhebar, K. Deb, E. Goodman, et al., Proceedings of the genetic and evolutionary computation conference 2019, Pages: 419-427 [14]

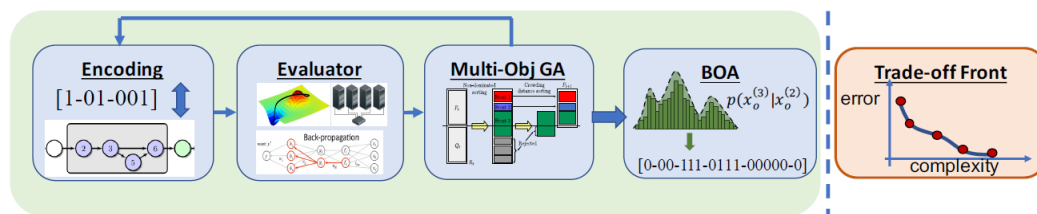
งานวิจัยนี้ได้นำเสนอ NSGA-Net ซึ่งเป็นแนวทางการค้นหาโครงสร้างโครงข่ายประสาทเทียม (Neural architecture search, NAS) โดยใช้อัลกอริทึมพันธุกรรมแบบหลายวัตถุประสงค์ (Multi-objective genetic algorithm) เพื่อค้นหาโครงสร้างของโครงข่ายที่มีประสิทธิภาพสูงทั้งในด้านความแม่นยำและความซับซ้อนของแบบจำลอง

NSGA-Net มีจุดเด่นที่แตกต่างจาก NAS แบบเดิมคือ การไม่จำกัดการค้นหาอยู่แค่โครงสร้างของบล็อกรูปแบบเดียวเพื่อนำไปทำซ้ำ (เช่น NASNet หรือ RL-based methods อื่นๆ) แต่เปิดโอกาสให้มีการปรับเปลี่ยนโครงสร้างในแต่ละช่วงของแบบจำลองได้อย่างยืดหยุ่น โดยใช้การเข้ารหัสโครงสร้างที่ยืดหยุ่นและดำเนินการวิวัฒนาการผ่านกระบวนการคัดเลือก ซึ่งส่งผลให้พื้นที่การค้นหา (Search space) กว้างขึ้นและมีความหลากหลายมากขึ้น

กลยุทธ์สำคัญของ NSGA-Net ประกอบด้วย

1. การเข้ารหัสโครงสร้างโดยใช้ Binary string เพื่อแทนการเชื่อมโยงระหว่าง Node ในแต่ละ Phase ซึ่งมีลักษณะเป็นกราฟไม่วน (Directed acyclic graph, DAG) ช่วยให้สามารถแทนโครงสร้างที่ซับซ้อนด้วยรูปแบบที่เหมาะสมกับกระบวนการทางพันธุกรรม
2. การดำเนินการทางพันธุกรรม เช่น Crossover ที่ออกแบบให้สามารถรักษาโครงสร้างย่อย (Sub-structure) ที่มีประสิทธิภาพ และ Mutation เพื่อเพิ่มความหลากหลายของประชากร ซึ่งมีผลต่อการสำรวจพื้นที่ค้นหาได้อย่างมีประสิทธิภาพมากขึ้น
3. การใช้ Bayesian optimization algorithm (BOA) ในขั้นตอน Exploitation เพื่อเรียนรู้ Distribution ของโครงสร้างที่ประสบความสำเร็จ แล้วใช้ความรู้นี้ในการสุ่มสร้างโครงสร้างใหม่ที่มีแนวโน้มคุณภาพดี ลดความซ้ำซ้อนและเร่งการเข้าสู่ค่าตอบ
4. การประเมินแบบหลายวัตถุประสงค์โดยใช้อัลกอริทึม NSGA-II ในการจัดอันดับโครงสร้างที่อยู่บน Pareto front เพื่อสร้างชุดคำตอบที่หลากหลายระหว่างความแม่นยำและความซับซ้อนของแบบจำลอง ซึ่งเหมาะกับการตัดสินใจในบริบทที่มีข้อจำกัดของทรัพยากร

กลยุทธ์สำคัญของ NSGA-Net



ภาพที่ 17 ภาพรวมของกระบวนการ NSGA-Net

ภาพที่ 17 แสดงภาพรวมของกระบวนการ NSGA-Net แสดงลำดับขั้นตอนตั้งแต่การเข้ารหัสโครงสร้างในรูปแบบ Binary string encoding, การประเมินประสิทธิภาพของแต่ละโครงสร้างผ่านกระบวนการ Backpropagation, การดำเนินการวิวัฒนาการด้วยอัลกอริทึมพันธุกรรม (Multi-objective GA), การแนะนำโครงสร้างใหม่ผ่าน Bayesian optimization algorithm (BOA) และการเลือกชุดคำตอบจาก Pareto front ที่แสดงความสมดุลระหว่าง Error และ Complexity

ภาพนี้แสดงให้เห็นว่า NSGA-Net ผสานทั้งกลยุทธ์การสำรวจ (Exploration) และการใช้ประโยชน์จากความรู้เดิม (Exploitation) เข้าไว้ในกระบวนการเดียวกัน โดยการใช้ทั้ง NSGA-II และ BOA ช่วยให้อาจพัฒนาโครงสร้างแบบจำลองที่ดีขึ้นอย่างต่อเนื่องในแต่ละรอบของวิวัฒนาการ

ผลการทดลองแสดงให้เห็นว่า บนชุดข้อมูล CIFAR-10 แบบจำลองที่ได้จาก NSGA-Net สามารถลดอัตราความผิดพลาด (Error rate) ลงเหลือเพียง 3.72% ที่ระดับการคำนวณ FLOPs เพียง 4.5 ล้าน ซึ่งถือเป็นผลลัพธ์ที่เทียบเคียงได้กับแบบจำลองที่ออกแบบด้วยมือหรือแบบจำลองจาก NAS ที่ใช้ทรัพยากรมากกว่าหลายเท่าอย่างมีนัยสำคัญ ทั้งนี้ NSGA-Net ยังสามารถสร้าง ชุดของโครงสร้างทางเลือกบนเส้น Pareto front ซึ่งครอบคลุมหลายระดับของความแม่นยำและความซับซ้อนของแบบจำลอง ทำให้ผู้ใช้งานสามารถเลือกแบบจำลองให้เหมาะสมกับข้อจำกัดเฉพาะด้านทรัพยากร เช่น เวลาในการประมวลผล หรือหน่วยความจำ

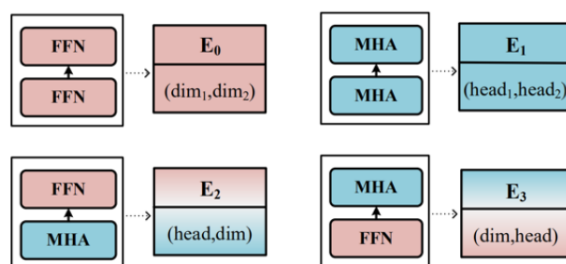
โดยสรุป NSGA-Net แสดงให้เห็นถึงศักยภาพของแนวทางการค้นหาโครงสร้างโครงข่ายที่อาศัยการปรับสมดุลระหว่างความแม่นยำและความซับซ้อน พร้อมทั้งมีการออกแบบกลไกวิวัฒนาการที่สามารถเรียนรู้จากประวัติการค้นหาเดิม ซึ่งแนวคิดเหล่านี้สอดคล้องกับเป้าหมายของงานวิจัยฉบับปัจจุบันที่มุ่งเน้นการพัฒนาแบบจำลอง Deep learning ที่เหมาะสมกับข้อจำกัดเชิงทรัพยากร

2. Genetic Algorithm-based Transformer Architecture Design for Neural Machine Translation, J. Wu, B. Feng and Y. Sun, Proceedings of the 2022 5th International Conference on Machine Learning and Machine Intelligence 2022, Pages: 98-104 [15]

งานวิจัยนี้นำเสนอแนวทางใหม่ในการออกแบบสถาปัตยกรรมของแบบจำลอง Transformer สำหรับงานแปลภาษาด้วยเครื่อง (Neural machine translation, NMT) โดยประยุกต์ใช้กระบวนการค้นหาเชิงวิวัฒนาการผ่านอัลกอริทึมพันธุกรรม (Genetic algorithm, GA) ซึ่งผู้วิจัยได้พัฒนาอัลกอริทึมชื่อว่า Genetic-Trans สำหรับการค้นหาโครงสร้างที่เหมาะสมโดยอัตโนมัติ ทั้งในแง่ของลำดับเลเยอร์และค่าพารามิเตอร์ภายในแต่ละบล็อก

ภายในงานมีการออกแบบกลไกการเข้ารหัสโครงสร้างของ Transformer โดยใช้แนวคิด Block-based encoding เพื่อเพิ่มความยืดหยุ่นและความหลากหลายของสถาปัตยกรรมที่สามารถค้นหาได้ โดยบล็อกเหล่านี้ถูกใช้ประกอบหลักในการแทนโครโมโซมของแต่ละ Individual ในกระบวนการวิวัฒนาการ ซึ่งแสดงในรูปแบบของโครโมโซมที่ประกอบด้วยยีนหลายตัว โดยแต่ละยีนแทนค่าพารามิเตอร์หรือองค์ประกอบสำคัญ เช่น จำนวนบล็อกของ Encoder และ Decoder, จำนวนหัว Attention และขนาดของ Feed-forward network

จุดเด่นของงานวิจัยนี้อยู่ที่การออกแบบ กลยุทธ์การเข้ารหัสโครโมโซม (Gene encoding strategy) ซึ่งสามารถแทนลำดับเลเยอร์และค่าพารามิเตอร์ได้อย่างยืดหยุ่น รองรับโครงสร้างที่มีความซับซ้อนหลากหลาย ซึ่งเป็นประโยชน์อย่างยิ่งในการนำไปใช้ร่วมกับอัลกอริทึมเชิงพันธุกรรมเพื่อการค้นหาเชิงสถาปัตยกรรม ดังภาพที่ 18



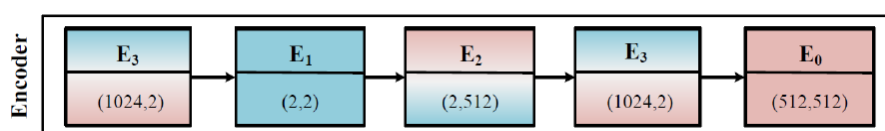
ภาพที่ 18 โครงสร้างของบล็อกพื้นฐานที่ใช้สำหรับ Encoder

บล็อกพื้นฐานในฝั่ง Encoder ที่ใช้ในการสร้างโครโมโซมของแต่ละ Individual โดยประกอบด้วยโมดูลย่อย ได้แก่ Multi-head attention (MHA) และ Feed-forward network (FFN) ซึ่งมีการ

จัดเรียงในรูปแบบต่าง ๆ เพื่อเพิ่มความหลากหลายของโครงสร้าง โดยกำหนดชนิดของบล็อกเป็น E_0 ถึง E_3 ตามลำดับโครงสร้างภายใน

- E_0 : Feed-forward network (FFN) → Feed-forward network (FFN)
- E_1 : Multi-head attention (MHA) → Multi-head attention (MHA)
- E_2 : Multi-head attention (MHA) → Feed-forward network (FFN)
- E_3 : Feed-forward network (FFN) → Multi-head attention (MHA)

ลักษณะของบล็อกเหล่านี้จะถูกสุ่มเรียงลำดับเป็นโครโมโซมที่แทนโครงสร้างของ Encoder ในแต่ละโครโมโซม และจะถูกพัฒนาต่อผ่านกระบวนการวิวัฒนาการ ซึ่งประกอบด้วยขั้นตอนหลัก ได้แก่ การสุ่มประชากรเริ่มต้น การคัดเลือก (Selection) การครอสโอเวอร์ (Crossover) และการกลายพันธุ์ (Mutation)



ภาพที่ 19 ตัวอย่างโครโมโซมที่ได้จากการสุ่มลำดับบล็อกและพารามิเตอร์

ภาพที่ 19 แสดงตัวอย่างโครโมโซมที่ได้จากการสุ่มโครงสร้างจากบล็อกพื้นฐาน E_0 – E_3 ดังภาพที่ 18 ซึ่งแต่ละบล็อกมีค่าพารามิเตอร์เฉพาะ เช่น จำนวนหัว Attention และขนาด Hidden dimension โดยลำดับของบล็อกจะมีผลต่อความสามารถของแบบจำลองในการแปลภาษา รวมถึงขนาดของแบบจำลองที่ได้

การประเมินผลของแต่ละสถาปัตยกรรมที่ได้จากกระบวนการวิวัฒนาการ ใช้ตัวชี้วัดมาตรฐาน ได้แก่ BLEU score และ Perplexity เพื่อสะท้อนความแม่นยำและประสิทธิภาพของแบบจำลอง ผลการทดลองบนชุดข้อมูล IWSLT14 (German → English) แสดงให้เห็นว่าแบบจำลองที่ได้จากการค้นหาด้วย Genetic-Trans ให้ผลลัพธ์ที่เหนือกว่า Transformer พื้นฐานในหลายกรณี ทั้งในด้านความแม่นยำที่เพิ่มขึ้นและจำนวนพารามิเตอร์ที่ลดลงอย่างมีนัยสำคัญ

โดยเฉพาะอย่างยิ่งในกรณีของ Embedding size ขนาด 128 และ 256 แบบจำลองที่ได้จาก Genetic-Trans สามารถให้ค่า BLEU score ที่สูงขึ้นในขณะที่ยังรักษาประสิทธิภาพเชิงขนาดไว้ได้ดี

สะท้อนให้เห็นถึงศักยภาพของ GA ในการออกแบบสถาปัตยกรรมที่มีคุณภาพภายใต้ข้อจำกัดด้านทรัพยากร

งานวิจัยนี้ชี้ให้เห็นว่า การผสมผสานกลไกการออกแบบสถาปัตยกรรมแบบยืดหยุ่นเข้ากับการค้นหาเชิงวิวัฒนาการ สามารถนำไปสู่การพัฒนาแบบจำลองที่มีความสามารถสูงในงานด้านการประมวลผลภาษาธรรมชาติ โดยเฉพาะในบริบทที่ต้องการลดภาระเชิงโครงสร้าง แนวทางดังกล่าวยังสามารถขยายผลไปสู่การประยุกต์ใช้งานอื่น ๆ เช่น การจำแนกข้อความ หรือระบบแนะนำอัตโนมัติที่มีลักษณะข้อมูลเชิงลำดับเช่นเดียวกัน

งานวิจัยนี้จึงถือเป็นพื้นฐานสำคัญ ที่แสดงให้เห็นถึงศักยภาพของการประยุกต์ใช้อัลกอริทึมพันธุกรรมในการปรับแต่งโครงสร้างแบบจำลอง Transformer ซึ่งแนวคิดในการเข้ารหัสบล็อกและการจัดการโครโมโซมที่มีความยืดหยุ่น ได้ถูกนำมาต่อยอดและพัฒนาเพิ่มเติมในงานวิจัยฉบับนี้

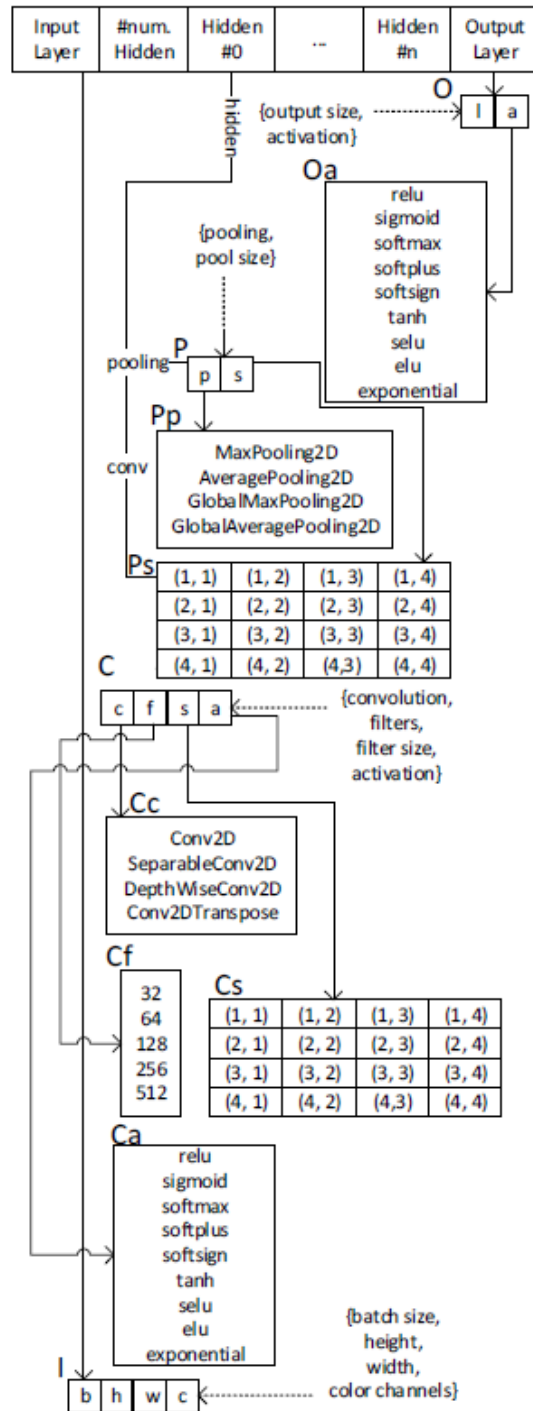
3. Effects of Genetic Operators on Neural architecture search Using Multi-Objective Genetic Algorithm, P. Patcharabumrung and Y. Jewajinda, 2023 20th International Joint Conference on Computer Science and Software Engineering (JCSSE) 2023, Publisher: IEEE Pages: 61-66 [16]

ได้นำเสนอการศึกษาผลของตัวดำเนินการทางพันธุกรรม ต่อประสิทธิภาพของกระบวนการค้นหาโครงสร้างโครงข่ายประสาทเทียม โดยใช้อัลกอริทึมพันธุกรรมแบบหลายวัตถุประสงค์ ซึ่งพัฒนาต่อยอดจากอัลกอริทึม NSGA-II และเสริมด้วยแนวคิด Dual population เพื่อเพิ่มความหลากหลายเชิงโครงสร้างในประชากร

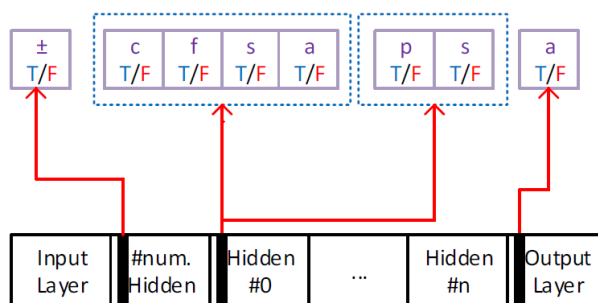
วัตถุประสงค์หลักของงานวิจัยนี้คือการประเมินผลกระทบของกลยุทธ์ Crossover และ Mutation ต่อประสิทธิภาพในการค้นหาสถาปัตยกรรมของโครงข่าย ซึ่งในการทดลองได้ใช้โครงสร้างแบบจำลองที่มีลักษณะใกล้เคียงกับ VGG โดยทำการเข้ารหัสโครงสร้างในรูปแบบ Block-chained encoding ซึ่งรองรับความยาวโครโมโซมที่แปรผันได้อย่างยืดหยุ่น อันทำให้สามารถแทนแบบจำลองที่มีจำนวนเลเยอร์แตกต่างกันได้

ลักษณะการเข้ารหัสดังกล่าวประกอบด้วยการแทนแต่ละเลเยอร์เป็นยีนภายในโครโมโซม และเสริมด้วย Control chromosome สำหรับควบคุมตำแหน่งของการกลายพันธุ์ โดยมี

วัตถุประสงค์เพื่อรักษาโครงสร้างที่ดีไว้และลดการเปลี่ยนแปลงที่ไม่จำเป็นในระหว่างกระบวนการวิวัฒนาการ



ภาพที่ 20 ภาพแสดงแนวคิดของกระบวนการครอสโอเวอร์แบบขยายความยาวเสมือน (Virtually extending crossover)



ภาพที่ 21 ตัวอย่างของกระบวนการกลายพันธุ์ (Mutation)

การทดลองในงานวิจัยนี้มุ่งเปรียบเทียบผลลัพธ์ระหว่างการใช้ Crossover เพียงอย่างเดียว และ Mutation เพียงอย่างเดียว และการใช้ทั้งสองกลยุทธ์ร่วมกัน ภายใต้สภาวะที่แตกต่างกันในด้านความหลากหลายเริ่มต้นของประชากร โดยใช้ชุดข้อมูล CIFAR-10 ในการประเมิน โดยพิจารณาทั้งค่าความแม่นยำในการจำแนกรูปภาพและจำนวนพารามิเตอร์ของแบบจำลอง

ผลลัพธ์แสดงให้เห็นว่า กลยุทธ์ที่ผสมผสานระหว่าง Crossover และ Mutation มีประสิทธิภาพสูงสุดในแง่ของการเข้าสู่กลุ่มคำตอบที่ดี และสามารถรักษาความหลากหลายของประชากรได้ดีกว่ากลยุทธ์ที่ใช้เพียงอย่างเดียวใดอย่างหนึ่ง โดยเฉพาะอย่างยิ่ง Mutation เพียงอย่างเดียว นั้นแม้จะให้ผลลัพธ์ที่ดีในช่วงต้น แต่เมื่อเวลาผ่านไปมีแนวโน้มที่จะติดอยู่กับคำตอบเฉพาะกลุ่ม และมีความหลากหลายน้อยลงอย่างเห็นได้ชัด

อย่างไรก็ตาม งานวิจัยยังได้กล่าวถึงข้อจำกัดของแนวทางที่เสนอ โดยเฉพาะในด้านการเข้ารหัสโครงสร้างที่ยังไม่สามารถรองรับองค์ประกอบสมัยใหม่บางประเภท เช่น Skip connection หรือ Parallel paths และยังไม่มีการรวมเทคนิคอย่าง Batch normalization ซึ่งเป็นองค์ประกอบที่สำคัญต่อประสิทธิภาพของแบบจำลองในปัจจุบัน

บทที่ 3

วิธีการดำเนินงานวิจัย

ในบทนี้จะกล่าวถึงกระบวนการดำเนินงานวิจัยโดยละเอียด ซึ่งครอบคลุมตั้งแต่การรวบรวมข้อมูลและการกำหนดขอบเขตของงาน ไปจนถึงการออกแบบอัลกอริทึมการหาค่าเหมาะที่สุดแบบหลายวัตถุประสงค์สำหรับปัญหาการค้นหาโครงสร้างของแบบจำลอง (Neural architecture search, NAS) ตลอดจนการประเมินประสิทธิภาพของอัลกอริทึมที่เสนอ ผ่านกระบวนการทดลองและวิเคราะห์ผลลัพธ์ ซึ่งมีรายละเอียดของแต่ละขั้นตอนการดำเนินงานดังนี้

3.1 รวบรวมข้อมูลและกำหนดขอบเขตของงาน

ในขั้นตอนนี้ เป็นการศึกษาข้อมูลและงานวิจัยที่เกี่ยวข้องเพื่อใช้เป็นพื้นฐานในการออกแบบกระบวนการวิจัย และนำข้อมูลที่ได้มาวิเคราะห์เพื่อกำหนดขอบเขตของงานวิจัยให้สอดคล้องกับวัตถุประสงค์ โดยมีรายละเอียดดังนี้

3.1.1 การกำหนดเครื่องมือและโปรแกรมที่ใช้ในการพัฒนาอัลกอริทึม

งานวิจัยนี้เลือกใช้ภาษา Python เนื่องจากมีไลบรารีและกรอบงานที่รองรับการทำงานด้านการทำงานของเครื่อง (Machine learning) และอัลกอริทึมเชิงวิวัฒนาการ (Evolutionary algorithm) สำหรับสร้างแบบจำลอง Vision transformer เช่นกรอบงาน PyTorch โดยใช้โปรแกรมในการพัฒนาคือ PyCharm ซึ่งสะดวกต่อการจัดการและมีประสิทธิภาพ

3.1.2 กำหนดชุดข้อมูลที่ใช้ในการทดสอบและประเมินประสิทธิภาพ

เพื่อให้การประเมินประสิทธิภาพของอัลกอริทึมที่ออกแบบมีความน่าเชื่อถือและสามารถเปรียบเทียบกับงานวิจัยอื่นได้ ผู้วิจัยเลือกใช้ชุดข้อมูลมาตรฐานที่นิยมในงานด้านการจำแนกภาพ ได้แก่ ชุดข้อมูลมาตรฐาน CIFAR-10, CIFAR-100 และ ImageNet

3.1.3 การกำหนดขอบเขตของงานวิจัย

ขอบเขตของงานวิจัยนี้มุ่งเน้นที่การประยุกต์ใช้อัลกอริทึม NSGA-II ในการค้นหาโครงสร้างของแบบจำลอง Vision transformer (ViT) โดยกำหนดวัตถุประสงค์ไว้ 2 ด้าน ได้แก่ ประสิทธิภาพของแบบจำลอง (Accuracy) และขนาดของแบบจำลอง (Parameter size)

3.2 ศึกษาทฤษฎีที่เกี่ยวข้องกับการประยุกต์ใช้อัลกอริทึมเชิงวิวัฒนาการในการหาค่าที่เหมาะสมแบบหลายวัตถุประสงค์

ในขั้นตอนนี้ ได้ศึกษาทฤษฎีที่เกี่ยวข้องกับการหาค่าที่เหมาะสมแบบหลายวัตถุประสงค์ (Multi-objective optimization) โดยมุ่งเน้นการประยุกต์ใช้อัลกอริทึมเชิงวิวัฒนาการ เพื่อใช้เป็นแนวทางในการออกแบบอัลกอริทึมสำหรับการค้นหาโครงสร้างของแบบจำลองในงานเรียนรู้เชิงลึก ซึ่งเนื้อหาที่ศึกษาแบ่งออกเป็น 2 ประเด็นหลัก ดังนี้

3.2.1 การศึกษาทฤษฎีเกี่ยวกับการหาค่าเหมาะสมที่สุดแบบหลายวัตถุประสงค์

ศึกษาแนวคิดพื้นฐานของการหาค่าเหมาะสมที่สุดในกรณีที่มีวัตถุประสงค์หลายด้านที่อาจขัดแย้งกัน เช่น ความแม่นยำของแบบจำลองและขนาดของแบบจำลอง รวมถึงความสัมพันธ์ระหว่างพื้นที่ตัดสินใจ (Decision space) และพื้นที่วัตถุประสงค์ (Objective space) โดยมุ่งเน้นการทำความเข้าใจแนวคิด Pareto-optimality ซึ่งเป็นแนวทางที่ไม่ให้ค่าน้ำหนักกับวัตถุประสงค์ใดวัตถุประสงค์หนึ่งโดยเฉพาะ แต่พิจารณาทุกวัตถุประสงค์อย่างเท่าเทียมกัน แนวคิดดังกล่าวถือเป็นพื้นฐานของอัลกอริทึม NSGA-II ที่ใช้ในงานวิจัยนี้

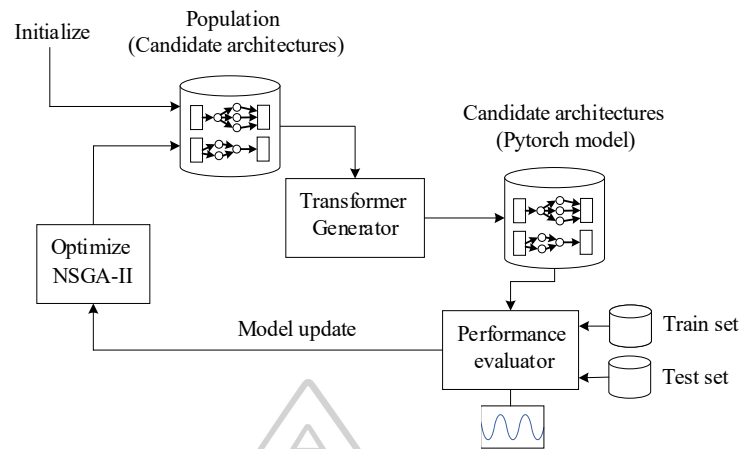
3.2.2 การศึกษาการประยุกต์ใช้อัลกอริทึมเชิงวิวัฒนาการในการแก้ปัญหา NAS

ศึกษาแนวทางการนำอัลกอริทึมเชิงวิวัฒนาการ มาใช้ในการแก้ปัญหาการค้นหาโครงสร้างของแบบจำลอง โดยเน้นศึกษาวิธีการเข้ารหัสโครโมโซมที่ใช้แทนสถาปัตยกรรมของโครงข่ายประสาทเทียม รวมถึงการศึกษากลไกภายในของอัลกอริทึม NSGA-II ซึ่งได้รับการออกแบบมาเพื่อจัดการกับปัญหาการหาค่าที่เหมาะสมแบบหลายวัตถุประสงค์

3.3 ออกแบบอัลกอริทึมการหาค่าเหมาะสมที่สุดแบบหลายวัตถุประสงค์สำหรับปัญหา NAS และทำการประเมินประสิทธิภาพอัลกอริทึม รวมถึงการประเมินประสิทธิภาพของกระบวนการผ่านการทดลองและการวิเคราะห์ผลลัพธ์

ในขั้นตอนนี้จะกล่าวถึงกระบวนการออกแบบการแทนแบบจำลอง Vision transformer (ViT) ให้อยู่ในรูปของโครโมโซม เพื่อใช้ร่วมกับอัลกอริทึมการหาค่าเหมาะสมที่สุดแบบหลายวัตถุประสงค์ โดยอาศัยอัลกอริทึม NSGA-II สำหรับการค้นหาสถาปัตยกรรมที่มีประสิทธิภาพ และประเมินประสิทธิภาพกับชุดข้อมูลมาตรฐาน พร้อมวิเคราะห์ผลลัพธ์ที่ได้และเปรียบเทียบกับงานวิจัยที่มีอยู่ในปัจจุบัน เช่น แบบจำลองที่อยู่ในงาน [11]

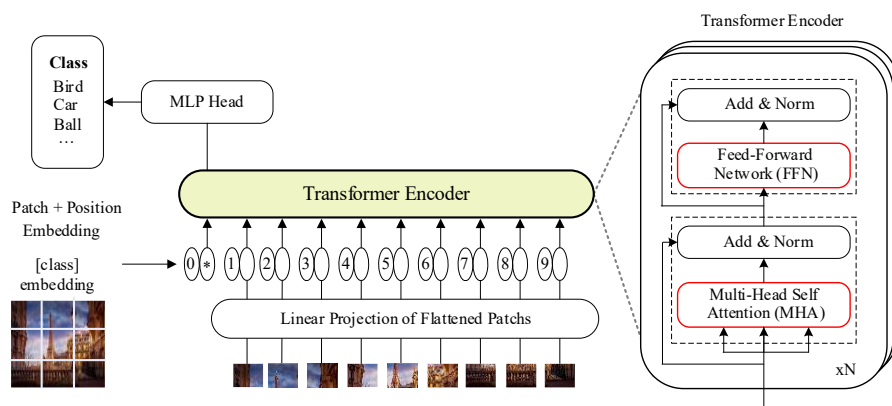
3.3.1 ภาพรวมของกระบวนการที่เสนอ



ภาพที่ 22 แผนภาพแสดงกระบวนการค้นหาสถาปัตยกรรมแบบหลายวัตถุประสงค์

ภาพที่ 22 เริ่มต้นด้วยการสุ่มสร้างประชากรเริ่มต้น จากนั้นจะใช้ตัวสร้างแบบจำลองอัตโนมัติผ่าน PyTorch generator เพื่อสร้างแบบจำลองจากโครโมโซมที่ได้จากประชากรเริ่มต้น โดยแต่ละโครโมโซมจะถูกแปลงเป็นสถาปัตยกรรมของแบบจำลองในรูปแบบของแบบจำลอง PyTorch จากนั้นจะทำการประเมินค่าความเหมาะสมของแต่ละแบบจำลอง โดยการฝึกสอนและทดสอบบนชุดข้อมูลมาตรฐาน โดยตัวชี้วัดประสิทธิภาพที่ได้ คือ ความแม่นยำในการจำแนกรูปภาพ และขนาดของแบบจำลอง จากนั้นอัลกอริทึม NSGA-II จะถูกใช้เพื่อปรับปรุงประชากรและสร้างประชากรใหม่ในรุ่นถัดไป โดยใช้ผลการประเมินเป็นเกณฑ์ในการคัดเลือกสถาปัตยกรรมที่เหมาะสมที่สุด กระบวนการทั้งหมดนี้จะถูกทำซ้ำอย่างต่อเนื่องจนกว่าจะถึงเงื่อนไขการหยุดที่กำหนด

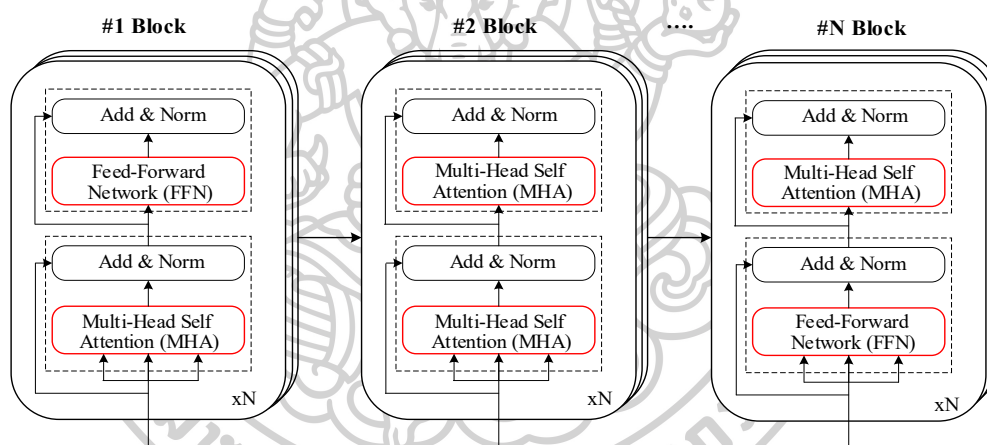
3.3.2 การเข้ารหัสโครโมโซมเพื่อแทนสถาปัตยกรรมของแบบจำลอง



ภาพที่ 23 องค์ประกอบของแบบจำลอง Vision transformer ที่ใช้ในการเข้ารหัสโครโมโซม

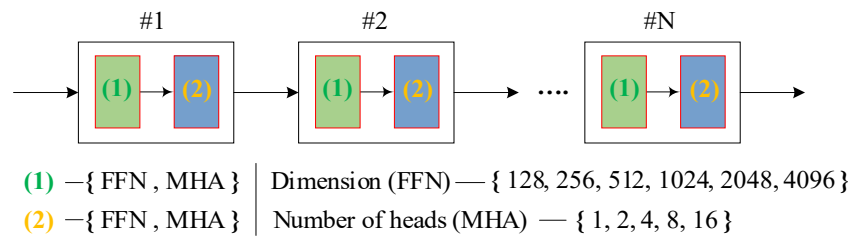
ใน Vision transformer (ViT) ดั้งเดิม โครงสร้างของตัวเข้ารหัส (Encoder) จะประกอบด้วยบล็อกย่อยหลายบล็อกที่มีโครงสร้างและค่าพารามิเตอร์เหมือนกันทุกบล็อก โดยเรียงซ้อนกันเป็นสแต็ก แต่ละบล็อกประกอบด้วยโมดูล Multi-head self-attention (MHA) และ Feed-forward network (FFN) [17, 18]

อย่างไรก็ตาม ในงานวิจัยนี้ ได้มีการประยุกต์โครงสร้างของ ViT ภายใน Transformer encoder เพื่อใช้ในการแทนสถาปัตยกรรมด้วยโครโมโซม โดยพิจารณาเฉพาะองค์ประกอบสำคัญที่สามารถปรับเปลี่ยนได้อย่างยืดหยุ่น เพื่อใช้ในการค้นหาแบบหลายวัตถุประสงค์อย่างมีประสิทธิภาพ โดยองค์ประกอบที่นำมาพิจารณา ได้แก่ จำนวนบล็อกภายใน Encoder รวมถึงพารามิเตอร์ภายในชั้น MHA และ FFN ซึ่งจะถูกนำมาเข้ารหัสเป็นโครโมโซม เพื่อแทนสถาปัตยกรรมที่หลากหลายและสามารถประเมินประสิทธิภาพได้ภายใต้กระบวนการค้นหาที่ใช้ NSGA-II

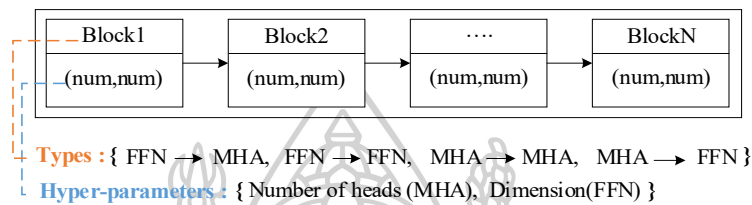


ภาพที่ 24 ตัวอย่างโครโมโซมที่แทนสถาปัตยกรรมของแบบจำลอง Vision transformer

ในกระบวนการค้นหาสถาปัตยกรรมของแบบจำลอง Vision transformer (ViT) จำเป็นต้องมีการแทนโครงสร้างของแบบจำลองในรูปแบบโครโมโซม เพื่อให้สามารถนำไปประมวลผลในอัลกอริทึมเชิงวิวัฒนาการได้อย่างมีประสิทธิภาพ โดยงานวิจัยนี้ได้นำเสนอวิธีการเข้ารหัสโครงสร้างของ ViT ในลักษณะ Block-based encoding ที่รองรับความยืดหยุ่นด้านจำนวนบล็อกและชนิดของบล็อกใน Encoder stack ดังแสดงตัวอย่างโครโมโซมที่ใช้แทนสถาปัตยกรรมของแบบจำลอง ViT ในภาพที่ 24



ภาพที่ 25 โครงสร้างและการกำหนดโครงสร้างในการเข้ารหัสโครโมโซม



ภาพที่ 26 ตัวอย่างโครโมโซมที่แทนสถาปัตยกรรม

ในการแทนสถาปัตยกรรมด้วยโครโมโซมในงานวิจัยนี้ จะแสดงในภาพที่ 24, 25 และภาพที่ 26 โดยแต่ละโครโมโซมจะแทนสถาปัตยกรรม ViT หนึ่งตัว ซึ่งประกอบด้วยจำนวน N บล็อก และในแต่ละบล็อกมีบล็อกย่อยจำนวน 2 บล็อก ที่สามารถเลือกประเภทได้ระหว่าง Multi-head self-attention (MHA) และ Feedforward network (FFN) ซึ่งการเรียงลำดับของบล็อกย่อยนี้ทำให้เกิดรูปแบบได้ทั้งหมด 4 ประเภท ได้แก่ FFN → FFN, FFN → MHA, MHA → FFN และ MHA → MHA

นอกจากนี้ บล็อกแต่ละตัวจะมีการระบุค่าของพารามิเตอร์ภายใน ได้แก่ จำนวน Attention heads ของชั้น MHA และ ขนาดของ Dimension ของชั้น FFN โดยเลือกค่าจากเซตจำกัด ได้แก่ :

- Attention heads : [2, 8, 16]
- FFN dimension : {128, 256, 512, 1024, 2048, 4096}

พารามิเตอร์ที่สามารถกำหนดได้ในการออกแบบสถาปัตยกรรม ViT ถูกสรุปไว้ในตารางที่ 2 ซึ่งแสดงค่าพารามิเตอร์ของแบบจำลอง Vision transformer (ViT) ทั้งหมดที่ใช้ในการเข้ารหัสโครโมโซม ได้แก่ จำนวนบล็อก, ประเภทของบล็อก, จำนวน Attention heads ใน MHA และขนาด Dimension ใน FFN ซึ่งล้วนเป็นองค์ประกอบสำคัญในการออกแบบโครงสร้าง ViT ที่ส่งผลต่อประสิทธิภาพโดยรวมของแบบจำลอง ทั้งนี้การกำหนดค่าของแต่ละบล็อกที่นำเสนอมีบทบาทสำคัญในการสร้างความหลากหลายให้กับสถาปัตยกรรมของแบบจำลองอีกด้วย

ตารางที่ 2 พารามิเตอร์ที่สามารถกำหนดได้ในการเข้ารหัสสถาปัตยกรรม ViT

Set no.	Configurable parameters	Values
1	Number of blocks	1 to 12
2	Configuration of a block	FFN \rightarrow FFN FFN \rightarrow MHA MHA \rightarrow FFN MHA \rightarrow MHA
3	Attention heads of MHA	1, 2, 4, 8, 16
4	Dimension of FFN	128, 256, 512, 1024, 2048, 4096

สำหรับการแทนค่าพารามิเตอร์ดังกล่าวงานวิจัยนี้ได้ใช้รูปแบบ ลิสต์ของคู่จำนวนเต็ม โดยแต่ละคู่แทนค่าพารามิเตอร์ของบล็อกหนึ่งบล็อก เช่น บล็อกที่มีคู (1, 256) แสดงถึงบล็อกที่มีจำนวน Attention heads เท่ากับ 1 และ FFN dimension เท่ากับ 256

แนวทางการเข้ารหัสนี้มีความยืดหยุ่นสูง ทั้งในแง่ของจำนวนบล็อก ลำดับของบล็อกย่อย และค่าพารามิเตอร์ภายในแต่ละบล็อก ซึ่งส่งเสริมความหลากหลายของสถาปัตยกรรมในประชากร และช่วยเพิ่มประสิทธิภาพในการสำรวจพื้นที่ค้นหาผ่านอัลกอริทึม NSGA-II ได้อย่างมีประสิทธิภาพ

3.3.4 การประยุกต์ใช้ NSGA-II กับกระบวนการค้นหา

ในงานวิจัยนี้ได้ประยุกต์ใช้อัลกอริทึม NSGA-II (Non-dominated sorting genetic algorithm-ii) สำหรับการค้นหาสถาปัตยกรรมของแบบจำลอง Vision transformer (ViT) ภายใต้กรอบของปัญหาแบบหลายวัตถุประสงค์ โดยมีเป้าหมายเพื่อเพิ่มความแม่นยำในการจำแนกรูปภาพ และลดขนาดของแบบจำลองให้มีขนาดเล็กที่สุดในเวลาเดียวกัน

อัลกอริทึม NSGA-II จะดำเนินการตามหลักการที่อธิบายไว้ในบทที่ 2 โดยมีการดัดแปลงตัวดำเนินการเชิงพันธุกรรมให้เหมาะสมกับโครงสร้างโครโมโซมที่ใช้แทนสถาปัตยกรรม ViT ซึ่งมีลักษณะความยาวแปรผันตามจำนวนบล็อกในแบบจำลอง

3.3.4.1 การประเมินค่าความเหมาะสมและการคัดเลือกด้วย NSGA-II

ประชากรเริ่มต้นในแต่ละรุ่นของอัลกอริทึมจะประกอบด้วยโครโมโซมที่แทนสถาปัตยกรรมของ ViT แต่ละตัวซึ่งแตกต่างกัน โดยโครโมโซมเหล่านี้จะถูกประเมินค่าความเหมาะสม

(Fitness evaluation) โดยใช้ฟังก์ชันวัตถุประสงค์ 2 วัตถุประสงค์ ได้แก่ ความแม่นยำในการจำแนกรูปภาพ (Accuracy) ซึ่งได้จากการฝึกสอนและทดสอบบนชุดข้อมูลมาตรฐาน เช่น CIFAR-10, CIFAR-100 และ ImageNet และขนาดของแบบจำลอง (Parameters size) ซึ่งคำนวณจากจำนวนพารามิเตอร์ทั้งหมดของแบบจำลอง

หลังจากการประเมินค่าความเหมาะสมแล้ว อัลกอริทึม NSGA-II จะดำเนินการคัดเลือกโครโมโซมเข้าสู่รุ่นถัดไป โดยใช้กลไกการจัดลำดับแบบไม่ครอบงำ (Non-dominated sorting) เพื่อจัดเรียงโครโมโซมเป็นชุดของคำตอบที่ไม่ครอบงำกัน (Pareto fronts) ซึ่งสะท้อนถึงการแลกเปลี่ยนกันระหว่างสองวัตถุประสงค์ที่มีลักษณะขัดแย้งกัน จากนั้นจะทำการประเมินความหนาแน่นของประชากร (Crowding distance) ภายในแต่ละ Front เพื่อรักษาความหลากหลายของคำตอบ และหลีกเลี่ยงการลู่เข้าสู่พื้นที่จำกัดของ Search space เร็วเกินไป

สุดท้ายจะใช้กลยุทธ์การคัดเลือกแบบรักษาคำตอบที่มีคุณภาพสูง (Elitism) เพื่อให้แน่ใจว่าโครโมโซมที่มีคุณภาพสูงในรุ่นปัจจุบันจะได้รับการส่งต่อไปยังรุ่นถัดไปได้อย่างมีประสิทธิภาพ กระบวนการทั้งหมดนี้ช่วยสร้างสมดุลระหว่างการค้นหาคำตอบใหม่และการรักษาคำตอบที่มีประสิทธิภาพสูงภายในประชากร

3.3.4.2 การสร้างลูกหลาน

ในอัลกอริทึมเชิงพันธุกรรม กระบวนการสร้างลูกหลานซึ่งประกอบด้วยการครอสโอเวอร์ (Crossover) และการกลายพันธุ์ (Mutation) ถือเป็นองค์ประกอบสำคัญที่ส่งผลอย่างมีนัยสำคัญต่อการรักษาความหลากหลายทางพันธุกรรมภายในประชากร การดำเนินการเหล่านี้ช่วยให้การสำรวจพื้นที่ค้นหาที่มีความหลากหลายและมีประสิทธิภาพ

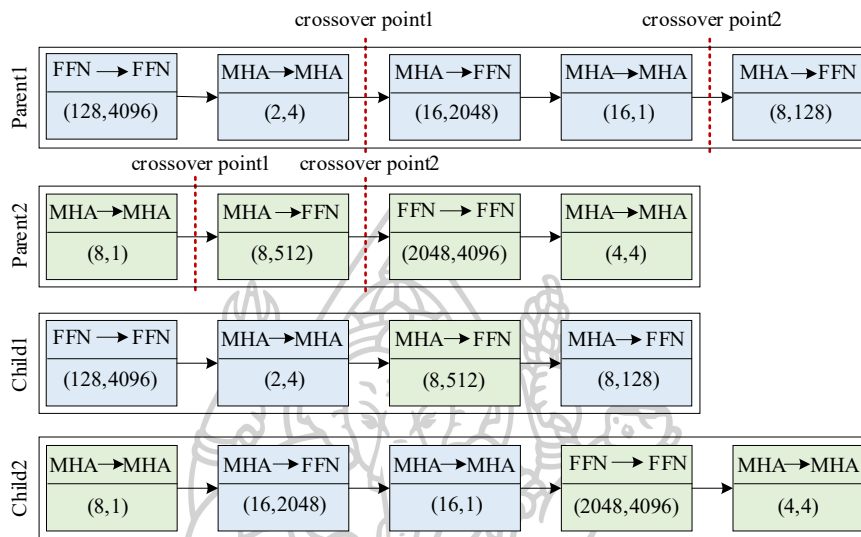
ในการแทนสถาปัตยกรรมของแบบจำลอง Vision transformer (ViT) ด้วยโครโมโซมพบว่าโครงสร้างของโครโมโซมมีจำนวนบล็อกที่สามารถเปลี่ยนแปลงได้จากกระบวนการครอสโอเวอร์ ส่งผลให้ความยาวของโครโมโซมแปรผันได้ระหว่างแต่ละรุ่นของประชากร ดังนั้นการออกแบบกระบวนการสร้างลูกหลานจึงจำเป็นต้องพิจารณาความแตกต่างของขนาดโครโมโซมระหว่างพ่อแม่ เพื่อให้สามารถดำเนินการครอสโอเวอร์และกลายพันธุ์ได้อย่างถูกต้องและมีประสิทธิภาพ

3.3.4.2.1 การครอสโอเวอร์ (Crossover)

ในงานวิจัยนี้ได้พิจารณากลยุทธ์การครอสโอเวอร์ 2 รูปแบบ ได้แก่ Two-point และ Uniform เพื่อสร้างโครโมโซมลูกหลานที่มีความหลากหลายทางสถาปัตยกรรมของ Vision

transformer (ViT) โดยรองรับกรณีที่โครโมโซมของพ่อแม่มีความยาวไม่เท่ากัน ซึ่งเป็นผลจากจำนวนบล็อกที่แปรผันได้ตามกระบวนการวิวัฒนาการ

- การครอสโอเวอร์แบบจุดสองจุด (Two-point crossover)



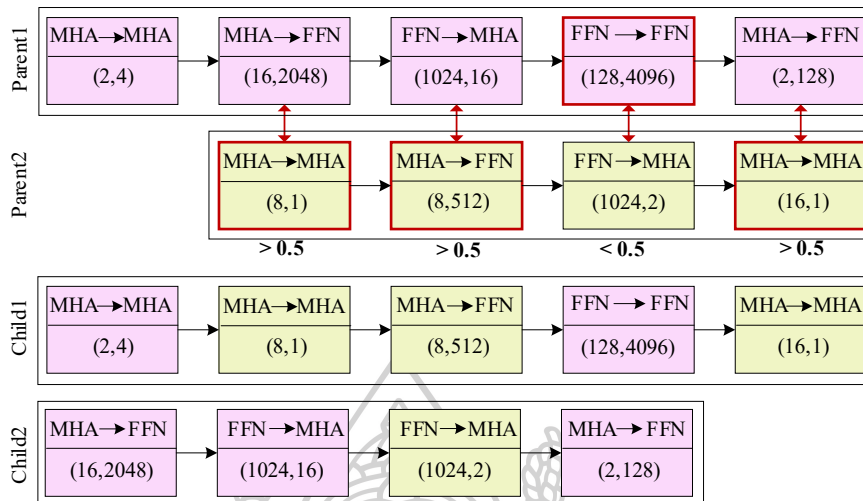
ภาพที่ 27 การครอสโอเวอร์แบบสองจุด (Two-point crossover)

ในการครอสโอเวอร์แบบสองจุด (Two-point crossover) โครโมโซมของพ่อแม่แต่ละตัวจะถูกสุ่มเลือกตำแหน่ง 2 ตำแหน่งภายในโครโมโซม จากนั้นจะทำการสลับส่วนของโครโมโซมที่อยู่ระหว่างตำแหน่งทั้งสอง เพื่อสร้างโครโมโซมลูกหลานใหม่ ดังแสดงในภาพที่ 27

จากภาพที่ 27 แสดงตัวอย่างกรณีที่โครโมโซมของพ่อแม่มีความยาว 5 บล็อก และ 4 บล็อก ตามลำดับ โดยตำแหน่งที่สามารถสุ่มได้จะขึ้นอยู่กับความยาวของแต่ละโครโมโซม เช่น โครโมโซมที่มีความยาว 5 บล็อกสามารถสุ่มตำแหน่งระหว่างตำแหน่ง (1, 2), (2, 3), (3, 4), และ (4, 5) ส่วนโครโมโซมที่มีความยาว 4 บล็อกจะสุ่มได้ระหว่าง (1, 2), (2, 3), และ (3, 4) เมื่อโครโมโซมสลับส่วนกันแล้ว จะส่งผลให้เกิดโครโมโซมลูกหลานจำนวน 2 โครโมโซมที่มีโครงสร้างใหม่ซึ่งได้มาจากโครงสร้างระหว่างพ่อแม่

การสลับโครงสร้างในลักษณะนี้อาจทำให้ความยาวของโครโมโซมลูกหลานแตกต่างไปจากพ่อแม่ ซึ่งช่วยเพิ่มความยืดหยุ่นในการแทนสถาปัตยกรรม และเสริมความหลากหลายของประชากร ส่งผลให้สามารถสำรวจพื้นที่ค้นหาได้อย่างครอบคลุมและมีประสิทธิภาพยิ่งขึ้น

- การครอสโอเวอร์แบบยูนิฟอร์ม (Uniform crossover)



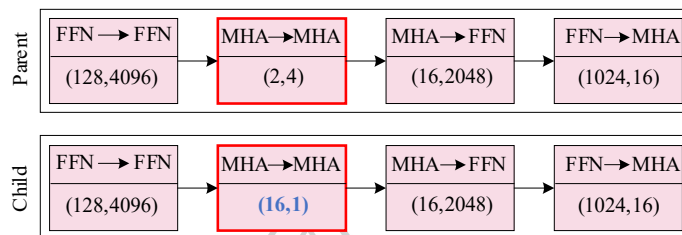
ภาพที่ 28 การครอสโอเวอร์แบบยูนิฟอร์ม (Uniform crossover)

โดยทั่วไปจะใช้กับโครโมโซมที่มีความยาวเท่ากัน เนื่องจากเป็นกระบวนการที่สุ่มเลือกแต่ละบล็อกจากพ่อแม่ทั้งสองตัวด้วยความน่าจะเป็นที่กำหนดไว้ อย่างไรก็ตามในงานวิจัยนี้ได้ประยุกต์ใช้วิธีการเฉพาะเพื่อรองรับกรณีที่โครโมโซมของพ่อแม่มีความยาวไม่เท่ากันที่อาจเกิดขึ้นจากกระบวนการวิวัฒนาการที่มีการเปลี่ยนแปลงจำนวนบล็อกในแต่ละโครโมโซม แนวทางที่นำมาใช้คือการสุ่มเลือกช่วงย่อยจากโครโมโซมที่ยาวกว่า โดยมีช่วงความยาวเท่ากับโครโมโซมที่สั้นกว่า เพื่อใช้ในการจับคู่บล็อกระหว่างโครโมโซมทั้งสองอย่างเหมาะสม โดยตำแหน่งเริ่มต้นของช่วงดังกล่าวจะถูกสุ่มจากตำแหน่งที่เป็นไปได้ทั้งหมดภายในโครโมโซมที่ยาวกว่า

ตัวอย่างในภาพที่ 28 แสดงให้เห็นกรณีที่โครโมโซมของพ่อแม่มีความยาวไม่เท่ากัน โดยโครโมโซมตัวที่สั้นกว่า (Parent2) จะถูกวางซ้อนบนโครโมโซมตัวที่ยาวกว่า (Parent1) ในตำแหน่งต่างๆที่สามารถจับคู่กันได้ เช่นตำแหน่งที่ (1,2,3), (2,3,4) หรือ (4,5,6) และเมื่อได้ช่วงที่ตรงกันแล้วจะทำการสุ่มเลือกบล็อกจากพ่อแม่ทั้งสองด้วยความน่าจะเป็นในการครอสโอเวอร์ (Crossover rate) เท่ากับ 0.5 สำหรับแต่ละตำแหน่ง โดยกำหนดว่า หากสุ่มได้ค่าความน่าจะเป็นน้อยกว่า 0.5 จะเลือกบล็อกจากโครโมโซม Parent1 และหากสุ่มได้ค่ามากกว่า 0.5 จะเลือกบล็อกจากโครโมโซม Parent2 และสำหรับบล็อกที่เหลืออยู่ในโครโมโซมที่ยาวกว่า ซึ่งไม่ได้ถูกจับคู่กับอีกโครโมโซมหนึ่ง บล็อกเหล่านี้จะถูกรวมเข้ากับโครโมโซมลูกหลานโดยตรง เพื่อรักษาโครงสร้างที่อาจมีความสำคัญต่อการค้นหาคำตอบที่มีประสิทธิภาพ

แนวทางการครอสโอเวอร์ในลักษณะนี้ช่วยให้สามารถรองรับโครโมโซมที่มีความยาวไม่เท่ากันได้อย่างยืดหยุ่น พร้อมทั้งยังคงส่งเสริมความหลากหลายทางพันธุกรรมในประชากร และเพิ่มประสิทธิภาพในการสำรวจพื้นที่ค้นหาเพื่อหาคำตอบที่เหมาะสมที่สุด

3.3.4.2.2 การกลายพันธุ์ (Mutation)



ภาพที่ 29 การกลายพันธุ์ (Mutation)

การกลายพันธุ์ (Mutation) เป็นกระบวนการสำคัญที่ช่วยรักษาความหลากหลายทางพันธุกรรมในประชากร และช่วยให้กระบวนการค้นหาของอัลกอริทึมเชิงวิวัฒนาการไม่จำกัอยู่เฉพาะบริเวณใดบริเวณหนึ่งในพื้นที่ค้นหา พร้อมทั้งเพิ่มโอกาสในการค้นพบโครงสร้างทางเลือกที่มีประสิทธิภาพมากยิ่งขึ้น โดยในงานวิจัยนี้กำหนดอัตราการกลายพันธุ์ (Mutation rate) ไว้ที่ 0.1 ซึ่งถือว่าเป็นอัตราที่ค่อนข้างต่ำ เพื่อให้มั่นใจว่ามีเพียงบางส่วนของโครโมโซมเท่านั้นที่ถูกเปลี่ยนแปลง ทั้งนี้เพื่อรักษาโครงสร้างและความยาวของโครโมโซมหลังจากกระบวนการครอสโอเวอร์ไว้โดยไม่เปลี่ยนแปลงโดยรวมมากเกินไป

จากภาพที่ 29 แสดงกระบวนการกลายพันธุ์ (Mutation) ซึ่งดำเนินการโดยการสุ่มเลือกหนึ่งบิตภายในโครโมโซม แล้วทำการสุ่มเปลี่ยนค่าพารามิเตอร์ภายในบิตนั้น ให้แตกต่างไปจากค่าเดิม เช่น จำนวน Attention heads ใน Multi-head self-attention (MHA) หรือขนาด Dimension ใน Feedforward network (FFN) ขณะที่บิตอื่นๆ ภายในโครโมโซมจะยังคงไม่เปลี่ยนแปลง เพื่อคงไว้ซึ่งโครงสร้างหลักของสถาปัตยกรรม อีกทั้งทำให้ไม่รบกวนโครงสร้างหลักของแบบจำลองมากเกินไป และสร้างความหลากหลายในการสำรวจสถาปัตยกรรมแบบจำลอง โดยไม่ทำลายโครงสร้างที่ดีจากโครโมโซมรุ่นก่อนหน้า และยังช่วยส่งเสริมประสิทธิภาพในการค้นหาคำตอบที่เหมาะสมยิ่งขึ้น

3.3.4.3 การประเมินผลและเกณฑ์การหยุดการทำงาน

การประเมินผลในกระบวนการค้นหาสถาปัตยกรรมของแบบจำลอง Vision transformer (ViT) ใช้ฟังก์ชันวัตถุประสงค์แบบหลายวัตถุประสงค์ เพื่อวัดความเหมาะสมของโครโมโซมแต่ละตัว

โดยใช้ตัวชี้วัดหลัก 2 ด้าน ได้แก่ความแม่นยำในการจำแนกรูปภาพ (Accuracy) บนชุดข้อมูลมาตรฐาน เช่น CIFAR10, CIFAR100 และ ImageNet และขนาดของแบบจำลอง (Parameters size) โดยโครโมโซมแต่ละตัวจะถูกฝึกและทดสอบภายใต้กระบวนการเดียวกันเพื่อประเมินค่าฟังก์ชันวัตถุประสงค์เหล่านี้ และค่าที่ได้จะถูกใช้ในขั้นตอนการคัดเลือกของอัลกอริทึม NSGA-II เพื่อจัดอันดับความเหมาะสมของสถาปัตยกรรม

สำหรับเกณฑ์การหยุดการทำงาน (Termination criterion) ได้กำหนดให้หยุดเมื่อจำนวนรอบของการวิวัฒนาการ (Generation) ครบตามที่กำหนดไว้ล่วงหน้า โดยในการทดลองนี้กำหนดค่าจำนวนรุ่นสูงสุด (Max generation) ไว้ที่ 64 รุ่น

3.3.4.4 การเปรียบเทียบกับอัลกอริทึมค้นหาแบบ NAS ที่มีอยู่แล้ว

เพื่อประเมินประสิทธิภาพของกระบวนการค้นหาที่เสนอในภาพรวม (หัวข้อ 3.3.1) งานวิจัยนี้ได้ทำการ เปรียบเทียบผลลัพธ์กับวิธีการค้นหาสถาปัตยกรรมอื่นที่มีอยู่แล้ว ได้แก่ แบบจำลองที่อยู่ใน [11] โดยใช้ตัวชี้วัดมาตรฐานเดียวกัน เช่น ค่าความแม่นยำในการจำแนกประเภท และขนาดของแบบจำลองที่ได้ ซึ่งประเมินบนชุดข้อมูลเดียวกัน เพื่อให้การเปรียบเทียบมีความยุติธรรม และสอดคล้องกับแนวปฏิบัติทั่วไปในการประเมินผลงานด้าน Neural architecture search

3.4 สรุปผลการดำเนินงานวิจัยและจัดทำรายงานวิทยานิพนธ์

การสรุปผลการดำเนินงานวิจัยและจัดทำรายงานวิทยานิพนธ์ถือเป็นขั้นตอนสุดท้ายของงานวิจัยนี้ โดยได้รวบรวมผลลัพธ์ที่ได้จากการทดลองทั้งหมดมาวิเคราะห์และประเมินอย่างเป็นระบบ เพื่อตรวจสอบว่าเป็นไปตามวัตถุประสงค์ที่กำหนดไว้หรือไม่ และเผยแพร่ผลงานวิจัยในการประชุมวิชาการ

บทที่ 4

ผลการทดลอง

ในบทนี้ได้นำเสนอการออกแบบการทดลองและผลลัพธ์ที่ได้จากการประเมินประสิทธิภาพของกระบวนการค้นหาสถาปัตยกรรม VIT แบบหลายวัตถุประสงค์ที่เสนอไว้ โดยการทดลองถูกแบ่งออกเป็น 2 กลุ่มหลักตามประเภทของตัวดำเนินการครอสโอเวอร์ ได้แก่ การครอสโอเวอร์แบบสองจุด (Two-point crossover) และการครอสโอเวอร์แบบยูนิฟอร์ม (Uniform crossover)

4.1 การตั้งค่าการทดลอง

ในการทดลองนี้ได้พัฒนาอัลกอริทึมที่เสนอโดยใช้ภาษา Python ภายใต้กรอบงาน PyTorch และทำการประเมินประสิทธิภาพบนชุดข้อมูลมาตรฐาน ได้แก่ CIFAR-10, CIFAR-100 และ ImageNet ซึ่งเป็นชุดข้อมูลมาตรฐานที่นิยมใช้ในการทดลองด้านการจำแนกรูปภาพ โดยใช้อุปกรณ์ประมวลผลหลักเป็นการ์ดจอ NVIDIA RTX 4090

กระบวนการฝึกสอนแบ่งออกเป็น 2 ขั้นตอนหลัก ได้แก่

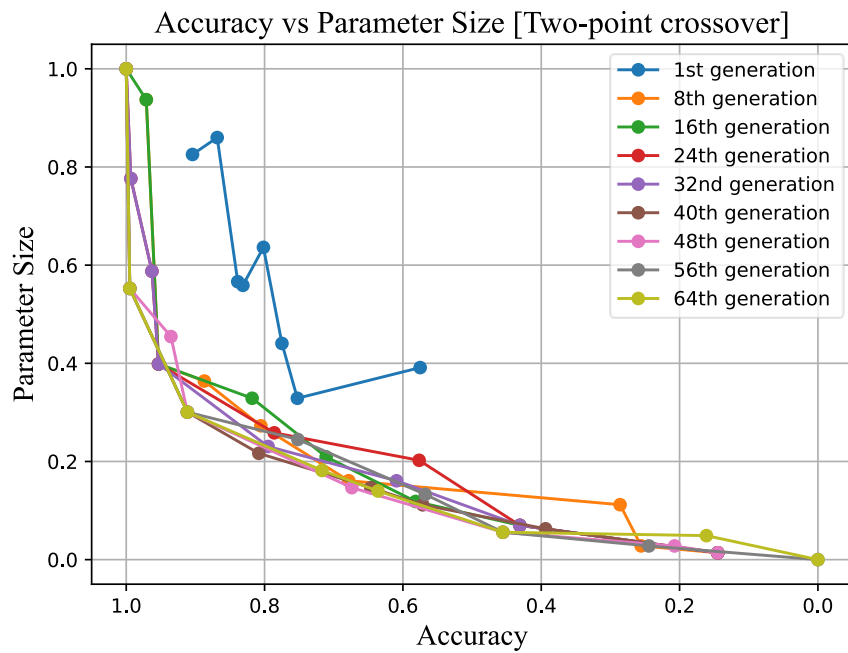
- ขั้นตอนการค้นหา (Search stage) : ทำการฝึกสอนแบบจำลองจำนวน 32 epochs โดยใช้ อัลกอริทึม NSGA-II ในการค้นหาชุดคำตอบของสถาปัตยกรรม
- ขั้นตอนการปรับจูนขั้นสุดท้าย (Fine-tuning stage) : ทำการฝึกสอนเพิ่มเติมจำนวน 500 epochs สำหรับกับประชากรชุดสุดท้าย (รุ่น 64) ที่ได้จากขั้นตอนการค้นหาเพื่อประเมินประสิทธิภาพของสถาปัตยกรรมแต่ละตัวอย่างละเอียด

ในแต่ละรุ่นจะเริ่มต้นด้วยประชากรจำนวน 8 โครโมโซม และจะถูกนำไปผ่านกระบวนการ Crossover และ Mutation เพื่อสร้างลูกหลานเพิ่มอีก 8 โครโมโซม รวมเป็นประชากรจำนวน 16 โครโมโซม จากนั้นจะนำประชากรทั้งหมดเข้าสู่กระบวนการจัดอันดับและคัดเลือกด้วยอัลกอริทึม NSGA-II โดยเลือกโครโมโซมจำนวน 8 ตัวที่มีค่าความเหมาะสมสูงสุดเพื่อส่งต่อเข้าสู่รุ่นถัดไป และตั้งค่าจำนวนรุ่นสูงสุดของกระบวนการวิวัฒนาการ (Maximum generations) ไว้ที่ 64 รุ่น และโดยใช้ อัตราการครอสโอเวอร์ (Crossover rate) เท่ากับ 80% และอัตราการกลายพันธุ์ (Mutation rate) เท่ากับ 10% สำหรับการสร้างลูกหลานในแต่ละรุ่น

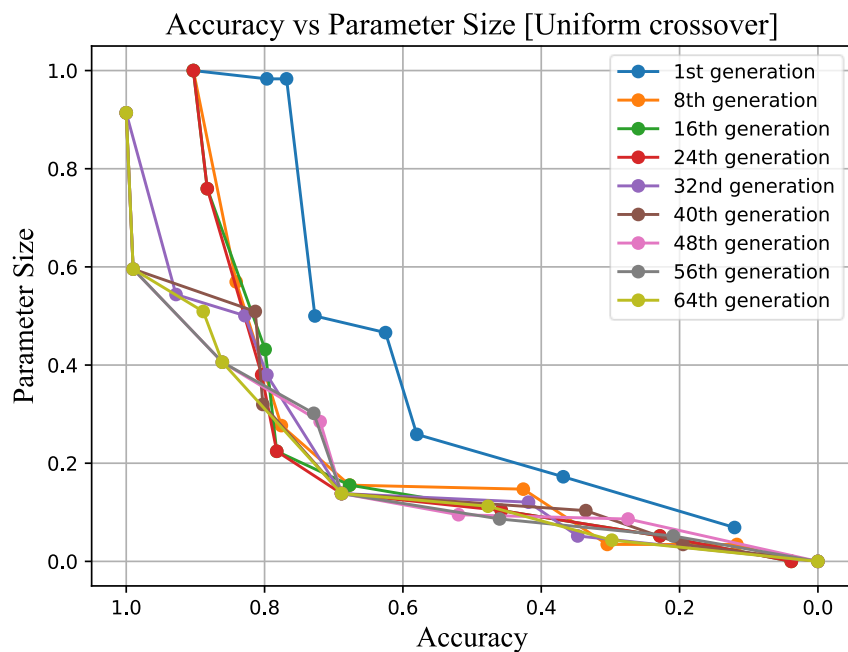
4.2 ผลการทดลอง

ผลการทดลองแสดงให้เห็นถึงผลกระทบของการเลือกใช้ตัวดำเนินการครอสโอเวอร์แบบสองจุดและแบบยูนิฟอร์ม ต่อการพัฒนาประชากรภายใต้กระบวนการเพิ่มประสิทธิภาพแบบหลาย

วัตถุประสงค์ ที่ดำเนินการอย่างต่อเนื่องหลายรุ่น โดยมีวัตถุประสงค์หลักเพื่อเพิ่มความแม่นยำของการจำแนกรูปภาพ (Accuracy) และลดขนาดของแบบจำลอง (Model size) ในเวลาเดียวกัน



ภาพที่ 30 แสดง Pareto front ของประชากรในแต่ละรุ่นที่ใช้การครอสโอเวอร์แบบสองจุด



ภาพที่ 31 แสดง Pareto front ของประชากรในแต่ละรุ่นที่ใช้การครอสโอเวอร์ยูนิฟอร์ม

จากภาพที่ 30 และภาพที่ 31 แสดง Pareto front ของประชากรในแต่ละรุ่นที่ได้จากขั้นตอนการค้นหา (Search stage) ที่ดำเนินการฝึกสอนแบบจำลองจำนวน 32 epochs สำหรับโครโมโซมแต่ละตัวในประชากร ภายใต้การใช้ตัวดำเนินการครอสโอเวอร์แบบสองจุดและแบบยูนิฟอร์ม ตามลำดับ โดยในทั้งสองภาพจะแสดงโครโมโซมจำนวน 8 ตัวจากทุกๆ 8 รุ่น ตั้งแต่รุ่นที่ 1 ถึงรุ่นที่ 64 มีแกน x ที่แสดงค่าความแม่นยำในการจำแนกประเภท และแกน y แสดงขนาดของแบบจำลอง ค่าบนทั้งสองแกนถูกปรับด้วยเทคนิค Min-max normalization ให้อยู่ในช่วง $[0, 1]$ เพื่อให้ง่ายต่อการเปรียบเทียบ

สำหรับกรณีของการใช้ครอสโอเวอร์แบบสองจุด (ภาพที่ 30) ค่าสูงสุดของแกน x และ y คือ 67.99 และ 10,004,970 และค่าต่ำสุดคือ 54.53 และ 610,666 ตามลำดับ ส่วนในกรณีของครอสโอเวอร์แบบยูนิฟอร์ม (ภาพที่ 31) ค่าสูงสุด ของแกน x และ y คือ 68.00 และ 7,634,922 และค่าต่ำสุดคือ 56.05 และ 675,050 ตามลำดับ

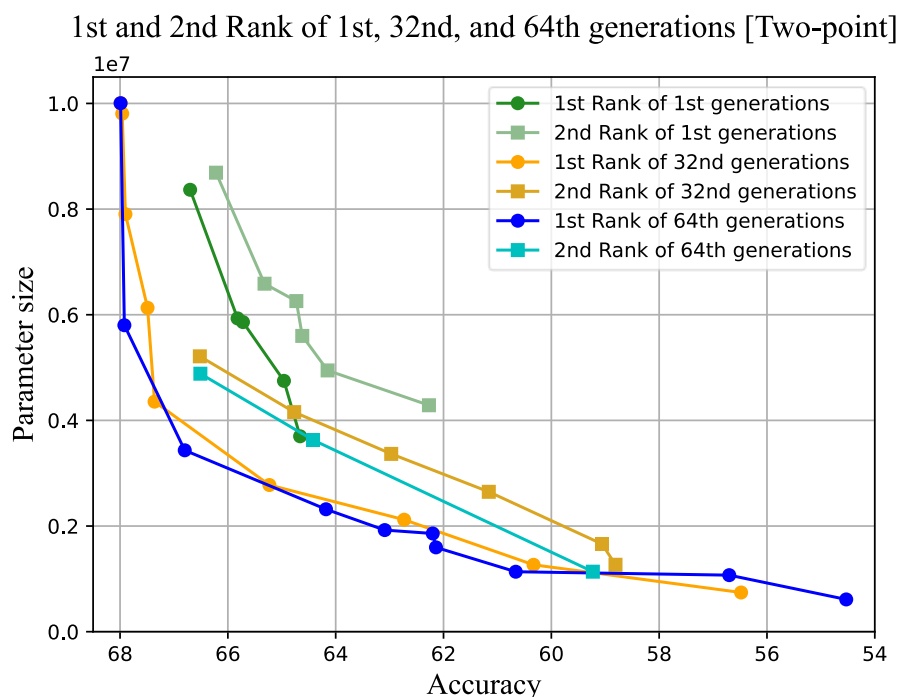
ผลลัพธ์จากภาพที่ 30 และภาพที่ 31 จะเห็นได้ว่า ในระหว่างกระบวนการค้นหาตั้งแต่รุ่นที่ 1 (สีน้ำเงิน) ไปจนถึงรุ่นที่ 64 (สีเหลือง) Pareto front มีแนวโน้มลู่เข้าสู่บริเวณจุดกำเนิด (Origin) ซึ่งแสดงถึงชุดคำตอบที่มีความสมดุลระหว่างสองวัตถุประสงค์ กล่าวคือ แบบจำลองที่มีความแม่นยำสูงและขนาดเล็กที่สุด แนวโน้มการลู่เข้าในลักษณะนี้แสดงถึงประสิทธิภาพของอัลกอริทึม NSGA-II และตัวดำเนินการครอสโอเวอร์ทั้งสองประเภทในการค้นหากลุ่มคำตอบที่มีความเหมาะสมสูงและมีความสมดุลของวัตถุประสงค์

ในกรณีของครอสโอเวอร์แบบสองจุด (Two-point crossover) ตามภาพที่ 30 พบว่า Pareto front มีแนวโน้มลู่เข้าใกล้ค่าที่เหมาะสมที่สุดได้เร็วกว่า โดยเริ่มเห็นความเสถียรของกลุ่มคำตอบตั้งแต่รุ่นที่ 40 เป็นต้นไป กล่าวคือ ผลลัพธ์จากรุ่นที่ 40, 48, 56 และ 64 มีการกระจุกตัวอยู่ในบริเวณที่มีค่าความแม่นยำระหว่าง 0.4 ถึง 1.0 และค่าขนาดของแบบจำลองอยู่ในช่วง 0.4 ถึง 0.0 ซึ่งแสดงให้เห็นถึงประสิทธิภาพของกระบวนการวิวัฒนาการในการพัฒนาคำตอบที่สมดุลระหว่างสองวัตถุประสงค์ นอกจากนี้จุดปลายของ Pareto front ในรุ่นที่ 64 ยังสามารถเข้าถึงค่าความแม่นยำสูงสุดพร้อมกับขนาดแบบจำลองต่ำสุด ซึ่งบ่งชี้ถึงความสามารถของอัลกอริทึมในการค้นหาคำตอบที่เหมาะสมที่สุด

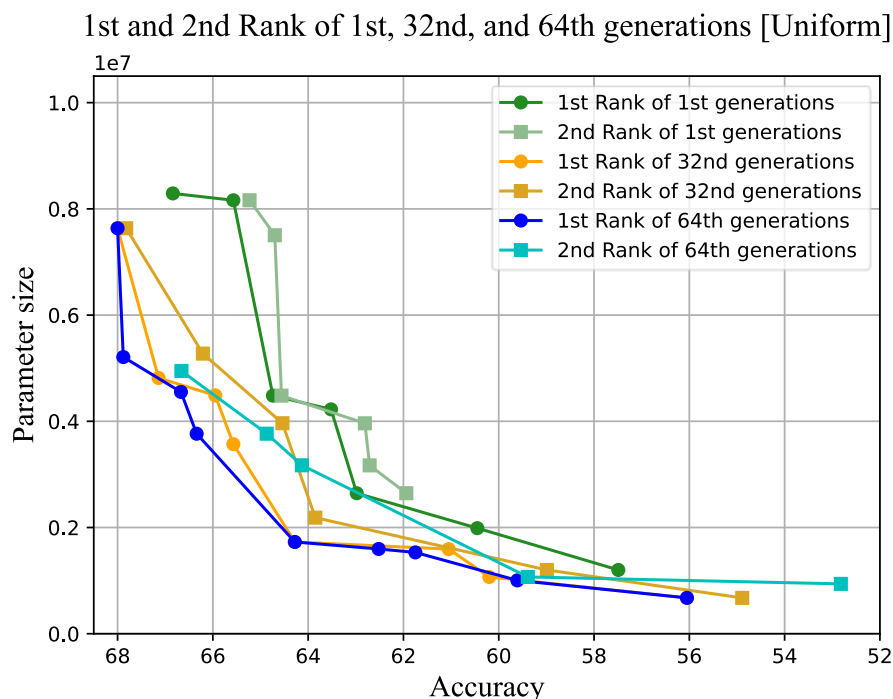
ในขณะที่กรณีของครอสโอเวอร์แบบยูนิฟอร์ม (Uniform crossover) ตามภาพที่ 31 พบว่าการลู่เข้าใกล้ชุดคำตอบที่เหมาะสม เกิดขึ้นช้ากว่า โดยเฉพาะในบริเวณที่มีค่าความแม่นยำสูง (ช่วง

0.8 ถึง 1.0) ซึ่งสามารถเปรียบเทียบได้จากความหนาแน่นของจุดในบริเวณดังกล่าวระหว่างภาพที่ 30 และภาพที่ 31 อย่างไรก็ตาม แม้จะลู่ช้ากว่า แต่การกระจายตัวของคำตอบแบบยูนิฟอร์มก็ยังคงมีแนวโน้มไปในทิศทางที่ดีและสามารถพัฒนาได้อย่างต่อเนื่องเช่นกัน

นอกจากนี้ผลการทดลองยังสะท้อนให้เห็นว่า การใช้ครอสโอเวอร์แบบสองจุด (Two-point crossover) สามารถสร้างความหลากหลายของประชากรได้มากกว่าการใช้ครอสโอเวอร์แบบยูนิฟอร์ม (Uniform crossover) เนื่องจากพื้นที่ของกลุ่มคำตอบจาก Two-point crossover กระจายตัวได้หลากหลายกว่าแบบยูนิฟอร์มได้อย่างชัดเจน และกลไกของ Two-point crossover ช่วยให้โครโมโซมลูกหลานมีความยาวที่หลากหลายมากขึ้นจากการสลับโครงสร้างภายใน ส่งผลให้เกิดความหลากหลายในเชิงโครงสร้างของแบบจำลองในแต่ละรุ่น ขณะที่ Uniform crossover มีข้อจำกัดในด้านความยาวของโครโมโซม ซึ่งจะคงไว้เท่ากับพ่อแม่เสมอ ส่งผลให้โอกาสในการสร้างสถาปัตยกรรมใหม่ที่หลากหลายลดลง และทำให้ความหลากหลายของประชากรในแต่ละรุ่นมีแนวโน้มจำกัดอยู่ในช่วงโครงสร้างเดิม ซึ่งอาจลดศักยภาพในการสำรวจพื้นที่ค้นหาอย่างครอบคลุม



ภาพที่ 32 แสดง Pareto front ของอันดับที่ 1 และ 2 ของประชากรในรุ่นที่ 1, 32 และ 64 ขณะใช้การครอสโอเวอร์แบบจุดสองจุด



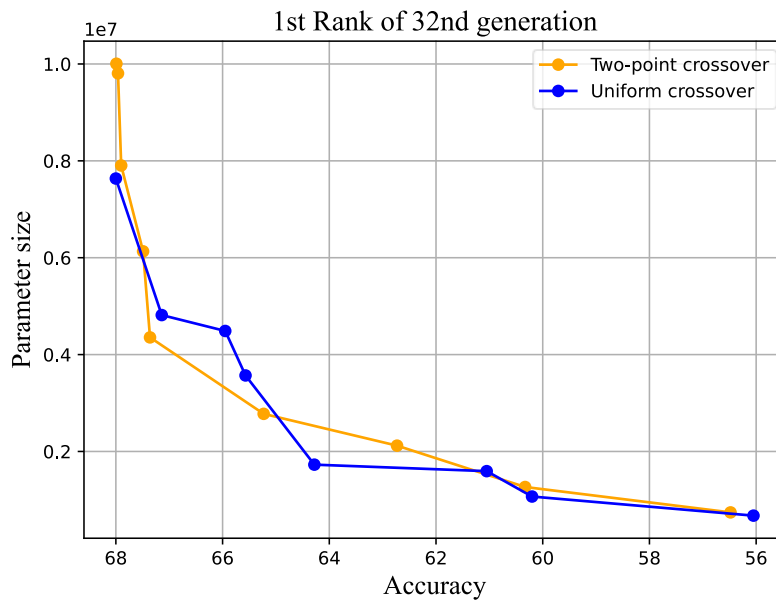
ภาพที่ 33 แสดง Pareto front ของอันดับที่ 1 และ 2 ของประชากรในรุ่นที่ 1, 32 และ 64 ขณะใช้การครอสโอเวอร์แบบยูนิฟอร์ม

นอกจากนี้ เพื่อวิเคราะห์การเปลี่ยนแปลงของกลุ่มคำตอบที่ไม่ถูกครอบงำ (Non-dominated solutions) ในแต่ละรุ่น ภาพที่ 32 และภาพที่ 33 แสดงกลุ่มคำตอบที่ถูกจัดลำดับเป็นอันดับที่หนึ่ง (1st Rank) และอันดับที่สอง (2nd Rank) จากผลลัพธ์ของอัลกอริทึม NSGA-II สำหรับกรณีของ Two-point และ Uniform ตามลำดับ โดยแสดงผลในรุ่นที่ 1, 32 และ 64

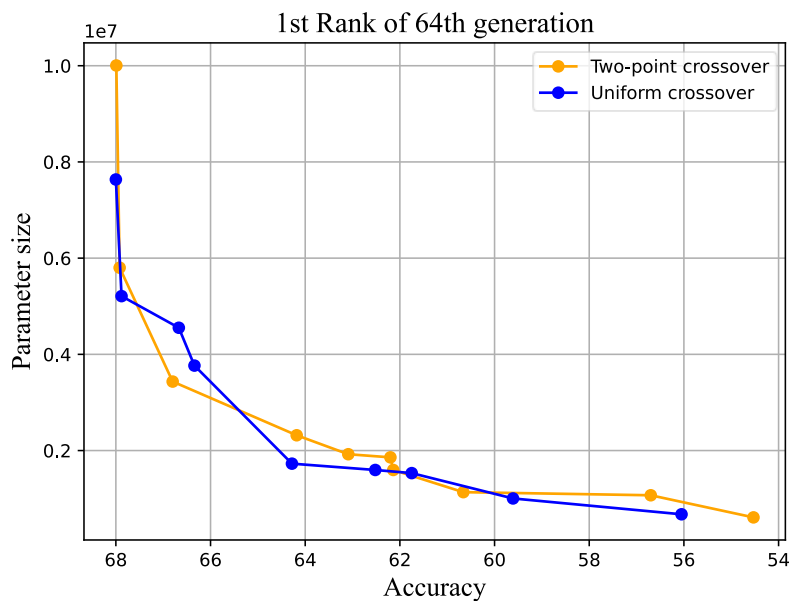
จากกราฟในภาพที่ 32 และภาพที่ 33 กลุ่มคำตอบอันดับหนึ่งหมายถึงกลุ่มของคำตอบที่ไม่ถูกครอบงำ ซึ่งถือเป็นชุดคำตอบแบบที่ดีที่สุด ในขณะที่กลุ่มอันดับสองคือกลุ่มคำตอบที่ไม่ถูกครอบงำในลำดับถัดไป โดยจากรูปจะเห็นว่า Pareto front ทั้งอันดับหนึ่งและอันดับสองทั้งสองวิธีมีแนวโน้มลู่เข้าสู่จุดกำเนิดเมื่อจำนวนรุ่นเพิ่มขึ้น ซึ่งแสดงให้เห็นถึงประสิทธิภาพของกระบวนการเพิ่มประสิทธิภาพแบบหลายวัตถุประสงค์

ในกรณีของ Two-point (ภาพที่ 32) พบว่าจำนวนกลุ่มคำตอบในอันดับหนึ่งเพิ่มขึ้นอย่างต่อเนื่องเมื่อจำนวนรุ่นมากขึ้น โดยเฉพาะในรุ่นที่ 64 ที่มีความหนาแน่นของจุดในกลุ่ม Pareto front แรกมากกว่ากรณีของ Uniform ซึ่งแสดงถึงความสามารถของกลไก Two-point ในการสร้างกลุ่มคำตอบที่มีคุณภาพสูงและมีความหลากหลายต่อเนื่องตลอดกระบวนการวิวัฒนาการ ในขณะที่

Uniform (ภาพที่ 33) แม้จะมีแนวโน้มของกลุ่มคำตอบ Pareto front ที่ลู่เข้าเหมือนกัน แต่การขยายตัวของกลุ่มคำตอบอันดับหนึ่งกลับเพิ่มช้ากว่า และมีความหนาแน่นน้อยกว่าเมื่อเทียบกับแบบ Two-point ทำให้เห็นถึงข้อจำกัดของวิธีนี้ในการสร้างโครงสร้างที่หลากหลายและไม่ซ้ำกับโครโมโซมเดิมในแต่ละรุ่น



ภาพที่ 34 การเปรียบเทียบ Pareto front ของประชากรในรุ่นที่ 32



ภาพที่ 35 การเปรียบเทียบ Pareto front ของประชากรในรุ่นที่ 64

จากกราฟในภาพที่ 34 และภาพที่ 35 แสดงการเปรียบเทียบคำตอบในกลุ่มอันดับหนึ่ง (1st Rank) ระหว่างวิธีการ Two-point และ Uniform ในรุ่นที่ 32 และ 64 ตามลำดับ เพื่อประเมินประสิทธิภาพของตัวดำเนินการครอสโอเวอร์แต่ละประเภทในการสร้างคำตอบที่ไม่ถูกครอบงำ (Non-dominated solutions)

ในรุ่นที่ 32 (ภาพที่ 34) พบว่า Two-point สามารถสร้างคำตอบที่ไม่ถูกครอบงำได้จำนวน 9 คำตอบ ในขณะที่ Uniform สร้างได้ 8 คำตอบ ซึ่งสะท้อนถึงศักยภาพของ Two-point ในการสร้างความหลากหลายของประชากรและค้นพบคำตอบที่ดีได้เร็วกว่าภายใต้จำนวนรุ่นที่เท่ากัน โดยเฉพาะอย่างยิ่งในช่วงค่าความแม่นยำระหว่าง 60–68% ซึ่งเป็นช่วงที่คำตอบจากทั้งสองวิธีมีแนวโน้มเข้าสู่จุดสมดุลระหว่างความแม่นยำและขนาดแบบจำลอง

สำหรับรุ่นที่ 64 (ภาพที่ 35) ผลลัพธ์แสดงให้เห็นว่าทั้งสองวิธีมีแนว Pareto front ที่ใกล้เคียงกันมากขึ้น โดยคุณภาพของคำตอบจากทั้งสองวิธีมีความคล้ายคลึงกันในเชิงความสมดุลของวัตถุประสงค์ทั้งสอง นอกจากนี้ยังพบว่าการกระจายตัวของกลุ่มคำตอบอันดับหนึ่งของ Two-point ยังคงมีลักษณะกว้างกว่าในบางช่วง ซึ่งสะท้อนให้เห็นถึงความสามารถในการรักษาความหลากหลายของคำตอบได้อย่างต่อเนื่อง

สรุปได้ว่า แม้ทั้งสองวิธีจะสามารถเข้าสู่กลุ่มคำตอบที่สมดุลได้ใกล้เคียงกันในระยะยาว แต่ Two-point มีข้อได้เปรียบในด้านการสร้างความหลากหลายและค้นพบคำตอบคุณภาพสูงได้เร็วกว่าในช่วงต้นของการวิวัฒนาการ โดยไม่ลดคุณภาพของคำตอบในภาพรวมของประชากร

จากตารางที่ 3 และตารางที่ 4 แสดงรายละเอียดของโครโมโซมจำนวน 8 ตัวในประชากรที่ผ่านการฝึกสอนจำนวน 64 รุ่น โดยแต่ละรุ่นฝึกสอนเท่ากับ 32 epochs โดยเปรียบเทียบระหว่างการครอสโอเวอร์แบบสองจุด และแบบยูนิฟอร์ม ซึ่งสอดคล้องกับภาพที่ 30 และภาพที่ 31

ในทั้งสองกรณี โครโมโซมถูกจัดลำดับหมายเลขตั้งแต่ 1 ถึง 8 โดยมีจำนวนบล็อกภายในโครงสร้างที่แตกต่างกันตั้งแต่ 3 ถึง 9 บล็อก ซึ่งจำนวนบล็อกที่มากขึ้นมักส่งผลให้ขนาดของแบบจำลองมีค่ามากขึ้นตามไปด้วย และขนาดของแบบจำลองอยู่ในช่วงประมาณ 500,000 พารามิเตอร์ ไปจนถึงเกือบ 10 ล้านพารามิเตอร์

ตารางที่ 3 แสดงโครโมโซมในประชากรรุ่นที่ 64 ขณะใช้การครอสโอเวอร์แบบสองจุด

No.	Num. of blocks	Chromosomes	Parameter size	Accuracy (%)
1	9	[(512, 1), (8, 4096), (256, 1024), (1, 2048), (512, 256), (4, 2048), (16, 4), (1, 128), (8, 4096)]	10,004,970	67.99
2	3	[(512, 1), (1, 128), (1, 128)]	610,666	54.53
3	7	[(1, 4096), (1, 2048), (512, 256), (4, 2048), (16, 4), (1, 128), (2, 256)]	5,801,450	67.92
4	5	[(512, 1), (2048, 256), (1, 2048), (16, 8), (8, 256)]	3,433,834	66.8
5	3	[(512, 1), (1, 1024), (1, 128)]	1,070,314	56.7
6	3	[(512, 1), (256, 8), (1, 128)]	1,135,082	60.66
7	4	[(512, 1), (2048, 256), (256, 8), (8, 1024)]	2,317,674	64.18
8	4	[(512, 1), (2048, 256), (256, 8), (8, 256)]	1,923,690	63.09

ค่าความแม่นยำของโครโมโซมในตารางทั้งสองมีช่วงใกล้เคียงกัน โดยมีค่าความแม่นยำสูงสุดอยู่ที่ประมาณ 68% และค่าต่ำสุดอยู่ที่ประมาณ 54-56% ซึ่งแสดงให้เห็นว่าแม้จะมีความแตกต่างของโครงสร้างภายในโครโมโซมและประเภทของการครอสโอเวอร์ที่ใช้ แต่โครโมโซมที่ได้รับการคัดเลือกเข้าสู่รุ่นที่ 64 ต่างก็มีประสิทธิภาพใกล้เคียงกันในเชิงของผลลัพธ์การจำแนกรูปภาพ

ตารางที่ 4 แสดงโครโมโซมในประชากรรุ่นที่ 64 ขณะใช้การครอสโอเวอร์แบบยูนิฟอร์ม

No.	Num. of block	Chromosomes	Parameter size	Accuracy (%)
1	3	[(1, 128), (16, 2), (8, 2)]	675,050	56.05
2	9	[(4, 256), (2, 16), (4, 2048), (256, 512), (8, 1024), (16, 2), (1, 128), (16, 1024), (8, 2)]	7,634,922	68.0
3	4	[(2048, 512), (256, 512), (1, 128), (8, 4)]	1,727,722	64.28
4	7	[(512, 1), (2048, 256), (1, 2048), (16, 8), (8, 256)]	3,765,866	66.34
5	7	[(8, 2048), (2048, 512), (2048, 512), (8, 2048), (256, 512), (8, 2), (2, 128)]	5,210,346	67.88
6	3	[(256, 512), (8, 2), (2, 128)]	1,004,266	59.61
7	7	[(16, 256), (256, 256), (2048, 512), (8, 2048), (256, 512), (8, 2), (2, 128)]	4,552,682	66.67
8	4	[(2048, 512), (256, 512), (8, 2), (2, 128)]	1,531,114	61.75

4.3 การฝึกสอนขั้นสุดท้ายและการเปรียบเทียบกับแบบจำลองต้นฉบับ

ตารางที่ 5 แสดงโครโมโซมที่ถูกเลือกจากการครอสโอเวอร์แบบสองจุด ที่ผ่านการฝึกแบบละเอียด

จำนวน 500 epochs บนชุดข้อมูล CIFAR-10

Two-point crossover				
No.	Num. of blocks	Chromosomes	Model size	Accuracy (%)
1	9	[(512, 1), (8, 4096), (256, 1024), (1, 2048), (512, 256), (4, 2048), (16, 4), (1, 128), (8, 4096)]	10,004,970	88.47%
4	5	[(512, 1), (2048, 256), (1,2048), (16, 8), (8, 256)]	3,433,834	88.09%
6	3	[(512, 1), (256, 8), (1, 128)]	1,135,082	88.26%

ตารางที่ 6 แสดงโครโมโซมที่ถูกเลือกจากการครอสโอเวอร์แบบยูนิฟอร์ม ที่ผ่านการฝึกแบบละเอียด

จำนวน 500 epochs บนชุดข้อมูล CIFAR-10

Uniform crossover				
No.	Num. of blocks	Chromosomes	Model size	Accuracy (%)
2	9	[(4, 256), (2, 16), (4, 2048), (256, 512), (8, 1024), (16, 2), (1, 128), (16, 1024), (8, 2)]	7,634,922	87.61%
3	4	[(2048, 512), (256, 512), (1, 128), (8, 4)]	1,727,722	85.06%
5	7	[(8, 2048), (2048, 512), (2048, 512), (8, 2048), (256, 512), (8, 2), (2, 128)]	5,210,346	88.05%

ตารางที่ 7 แสดงโครโมโซมที่ถูกเลือกจากการครอสโอเวอร์แบบสองจุด ที่ผ่านการฝึกแบบละเอียด
จำนวน 500 epochs บนชุดข้อมูล CIFAR-100

Two-point crossover				
No.	Num. of blocks	Chromosomes	Model size	Accuracy (%)
1	9	[(512, 1), (8, 4096), (256, 1024), (1, 2048), (512, 256), (4, 2048), (16, 4), (1, 128), (8, 4096)]	10,004,970	88.47%
4	5	[(512, 1), (2048, 256), (1,2048), (16, 8), (8, 256)]	3,433,834	88.09%
6	3	[(512, 1), (256, 8), (1, 128)]	1,135,082	88.26%

ตารางที่ 8 แสดงโครโมโซมที่ถูกเลือกจากการครอสโอเวอร์แบบยูนิฟอร์ม ที่ผ่านการฝึกแบบละเอียด
จำนวน 500 epochs บนชุดข้อมูล CIFAR-100

Uniform crossover				
No.	Num. of blocks	Chromosomes	Model size	Accuracy (%)
2	9	[(4, 256), (2, 16), (4, 2048), (256, 512), (8, 1024), (16, 2), (1, 128), (16, 1024), (8, 2)]	7,634,922	87.61%
3	4	[(2048, 512), (256, 512), (1, 128), (8, 4)]	1,727,722	85.06%
5	7	[(8, 2048), (2048, 512), (2048, 512), (8, 2048), (256, 512), (8, 2), (2, 128)]	5,210,346	88.05%

ตารางที่ 9 แสดงโครโมโซมที่ถูกเลือกจากการครอสโอเวอร์แบบสองจุด ที่ผ่านการฝึกแบบละเอียด จำนวน 500 epochs บนชุดข้อมูล IMAGENET (10 คลาส)

Two-point crossover				
No.	Num. of blocks	Chromosomes	Model size	Accuracy (%)
1	9	[(512, 1), (8, 4096), (256, 1024), (1, 2048), (512, 256), (4, 2048), (16, 4), (1, 128), (8, 4096)]	10,004,970	88.47%
4	5	[(512, 1), (2048, 256), (1,2048), (16, 8), (8, 256)]	3,433,834	88.09%
6	3	[(512, 1), (256, 8), (1, 128)]	1,135,082	88.26%

ตารางที่ 10 แสดงโครโมโซมที่ถูกเลือกจากการครอสโอเวอร์แบบยูนิฟอร์ม ที่ผ่านการฝึกแบบละเอียด จำนวน 500 epochs บนชุดข้อมูล IMAGENET (10 คลาส)

Uniform crossover				
No.	Num. of blocks	Chromosomes	Model size	Accuracy (%)
2	9	[(4, 256), (2, 16), (4, 2048), (256, 512), (8, 1024), (16, 2), (1, 128), (16, 1024), (8, 2)]	7,634,922	87.61%
3	4	[(2048, 512), (256, 512), (1, 128), (8, 4)]	1,727,722	85.06%
5	7	[(8, 2048), (2048, 512), (2048, 512), (8, 2048), (256, 512), (8, 2), (2, 128)]	5,210,346	88.05%

ตารางที่ 5 - 10 แสดงรายละเอียดของโครโมโซมที่ถูกคัดเลือกจากรุ่นที่ 64 ซึ่งผ่านการฝึกสอนในขั้นตอนสุดท้าย (Fine-tuning stage) จำนวน 500 epochs เพื่อประเมินประสิทธิภาพที่แท้จริงหลังจากการค้นหาเสร็จสิ้น โดยแบ่งตามประเภทของตัวดำเนินการครอสโอเวอร์ที่ใช้ ได้แก่ Two-point และ Uniform บนชุดข้อมูลมาตรฐาน CIFAR-10, CIFAR-100, IMAGENET-10 ตามลำดับ

จากการเปรียบเทียบข้ามกลุ่ม พบว่า Two-point สามารถให้โครโมโซมที่มีความแม่นยำสูงกว่า แม้ในกรณีที่มีจำนวนบล็อกเท่ากัน และยังมีแนวโน้มในการรักษาความหลากหลายของโครงสร้างได้ดีกว่า

ตารางที่ 11 การเปรียบเทียบผลการฝึกแบบจำลองในงานวิจัยกับแบบจำลอง ViT ต้นแบบ บนชุดข้อมูล (CIFAR-10)

Description	Num. Layers	Accuracy (%)	Params (1M)	GPU-days
ViT-Base [11]	12	91.4	86	-
ViT-Large [11]	24	99.4	307	-
ViT-Huge [11]	32	100	632	-
ViT-two-point-1	9	88.47	10	4.543
ViT-two-point-4	5	88.09	3.4	4.504
ViT-two-point-6	3	88.26	1.1	4.535
Vit-uniform-2	9	87.61	7.6	4.142
Vit-uniform-3	4	85.06	1.7	4.138
Vit-uniform-5	7	88.05	5.2	4.141

ตารางที่ 11 แสดงการเปรียบเทียบระหว่างแบบจำลอง ViT ที่ได้รับการปรับแต่งผ่านกระบวนการของงานวิจัยนี้กับแบบจำลอง ViT ดั้งเดิมตามที่ระบุใน [11] โดยพบว่าแบบจำลองที่ได้จากการค้นหาโดยใช้อัลกอริทึม NSGA-II มีขนาดแบบจำลองที่เล็กกว่ามาก ขณะที่ค่าความแม่นยำยังต่ำกว่าเล็กน้อย ซึ่งเป็นข้อเท็จจริงที่ทราบกันดีว่า แบบจำลองที่มีขนาดใหญ่กว่า มักให้ประสิทธิภาพที่ดีกว่า อย่างไรก็ตาม ผลการเปรียบเทียบนี้มีสาเหตุส่วนหนึ่งมาจากการที่แบบจำลองในงานวิจัยนี้ถูก

ฝึกเพียง 32 epochs ในขั้นตอนการค้นหา และมีเพียงบางแบบจำลองเท่านั้นที่ได้รับการฝึกเพิ่มเติม ในขั้นตอน Fine-tuning จำนวน 500 epochs เพื่อการประเมินผลอย่างละเอียด ขณะที่แบบจำลอง ต้นฉบับของ ViT นั้นได้รับการฝึกสอนด้วยจำนวน Epochs ที่สูงกว่ามาก การเปรียบเทียบนี้จึงสะท้อนให้เห็นถึงข้อจำกัดด้านทรัพยากรระหว่างการค้นหา และไม่สามารถสะท้อนศักยภาพสูงสุดของ แบบจำลองที่ค้นหาได้ทั้งหมด

4.4 อภิปรายผลและสรุปผลการทดลอง

สรุปรายละเอียดเปรียบเทียบระหว่าง Two-point crossover และ Uniform crossover ไว้ ในตารางที่ 12 ดังนี้

ตารางที่ 12 เปรียบเทียบผลลัพธ์การครอสโอเวอร์แบบ Two-point และ Uniform

หมวดหมู่	Two-point crossover	Uniform crossover
ช่วงค่าความแม่นยำ (ขั้นตอนการค้นหา)	54% – 68%	56% – 68%
ขนาดแบบจำลอง	610,666 – 10,004,970	675,050 – 7,634,922
ความหลากหลายของโครโมโซม	ความยาวบล็อกแปรผัน	ความยาวคงที่ตามพ่อแม่
การลู่เข้า (Convergence)	เร็วกว่า	ช้ากว่า
จำนวนคำตอบอันดับ 1 (Rank1)	มากกว่า	น้อยกว่า
การกระจายตัวของคำตอบ	กว้างกว่า (ครอบคลุมหลากหลายจุดใน Pareto)	แคบกว่า (กระจุกตัวในช่วงจำกัด)

ในภาพรวม อัลกอริทึม NSGA-II ที่ประยุกต์ใช้ในงานวิจัยนี้สามารถสร้างแบบจำลองที่มี ประสิทธิภาพทั้งในด้านขนาดและความแม่นยำได้อย่างสมดุล โดย Two-point แสดงผลลัพธ์ที่โดดเด่นกว่าในหลายด้าน ทั้งด้านความเร็วในการลู่เข้าสู่คำตอบที่ดี และความสามารถในการรักษาความ หลากหลายของประชากร และการค้นพบสถาปัตยกรรมที่มีคุณภาพสูงในช่วงต้นของการวิวัฒนาการ

บทที่ 5

สรุปและข้อเสนอแนะ

5.1 สรุปผลการวิจัย

งานวิจัยนี้มีวัตถุประสงค์เพื่อพัฒนากระบวนการค้นหาสถาปัตยกรรมของแบบจำลอง Vision transformer (ViT) โดยประยุกต์ใช้อัลกอริทึมเชิงวิวัฒนาการแบบหลายวัตถุประสงค์ (Multi-objective evolutionary algorithm) ซึ่งเลือกใช้อัลกอริทึม NSGA-II เป็นกลไกหลักในการเพิ่มประสิทธิภาพของแบบจำลอง โดยอัลกอริทึมนี้มีความสามารถในการจัดการกับปัญหาหลายวัตถุประสงค์ผ่านกลไกการจัดลำดับแบบไม่ครอบงำ (Non-dominated sorting), การคงไว้ซึ่งคำตอบที่ดี (Elitism) และตัวดำเนินการทางพันธุกรรม (Genetic operators) โดยมีเป้าหมายหลักสองประการ คือ การเพิ่มความแม่นยำของแบบจำลองและการลดขนาดของแบบจำลองในเวลาเดียวกัน เพื่อให้ได้แบบจำลองที่มีประสิทธิภาพและเหมาะสมกับการใช้งานจริงภายใต้ข้อจำกัดของทรัพยากร

เพื่อนำเสนอแนวทางที่มีประสิทธิภาพ งานวิจัยได้ออกแบบโครงสร้างโครโมโซมของ ViT โดยใช้การเข้ารหัสแบบ Block-based ซึ่งสามารถแทนชนิดของบล็อกภายในสถาปัตยกรรมได้ถึง 4 รูปแบบ และรองรับความยืดหยุ่นของความยาวโครโมโซมผ่านโครงสร้างแบบ Block-chained โดยกระบวนการเข้ารหัสนี้ช่วยเพิ่มความหลากหลายของโครงสร้างของสถาปัตยกรรม

นอกจากนี้ยังศึกษาผลของตัวดำเนินการครอสโอเวอร์ 2 ประเภท ได้แก่ Two-point crossover และ Uniform crossover เพื่อเปรียบเทียบประสิทธิภาพในการเข้าสู่คำตอบและความหลากหลายของประชากร ซึ่ง Two-point crossover แสดงให้เห็นถึงข้อได้เปรียบด้านการสร้างโครงสร้างที่หลากหลายและสามารถเข้าสู่คำตอบที่ดีได้เร็วกว่า โดยเฉพาะในช่วงต้นของกระบวนการวิวัฒนาการ ขณะที่ Uniform crossover มีข้อจำกัดในการเปลี่ยนแปลงความยาวโครโมโซม ส่งผลให้ความหลากหลายน้อยกว่า

ผลการทดลองบนชุดข้อมูลมาตรฐาน เช่น CIFAR-10, CIFAR-100 และ ImageNet แสดงให้เห็นว่า แนวทางที่นำเสนอสามารถค้นหาแบบจำลองที่มีความสมดุลระหว่างขนาดแบบจำลองกับความแม่นยำได้อย่างมีประสิทธิภาพ โดยบางโครงสร้างสามารถบรรลุความแม่นยำได้สูงถึง 88% หลังจากผ่านกระบวนการฝึกสอนเพิ่มเติม (Fine-tuning) จำนวน 500 epochs ขณะเดียวกันยังคงจำนวนพารามิเตอร์ไว้ต่ำกว่า 10 ล้านพารามิเตอร์ ซึ่งสะท้อนถึงศักยภาพของ NSGA-II ในการค้นหาแบบจำลองที่มีพื้นฐานดีได้ตั้งแต่ขั้นตอนการฝึกสอนเบื้องต้นที่ใช้เพียง 32 epochs

5.2 ข้อเสนอแนะ

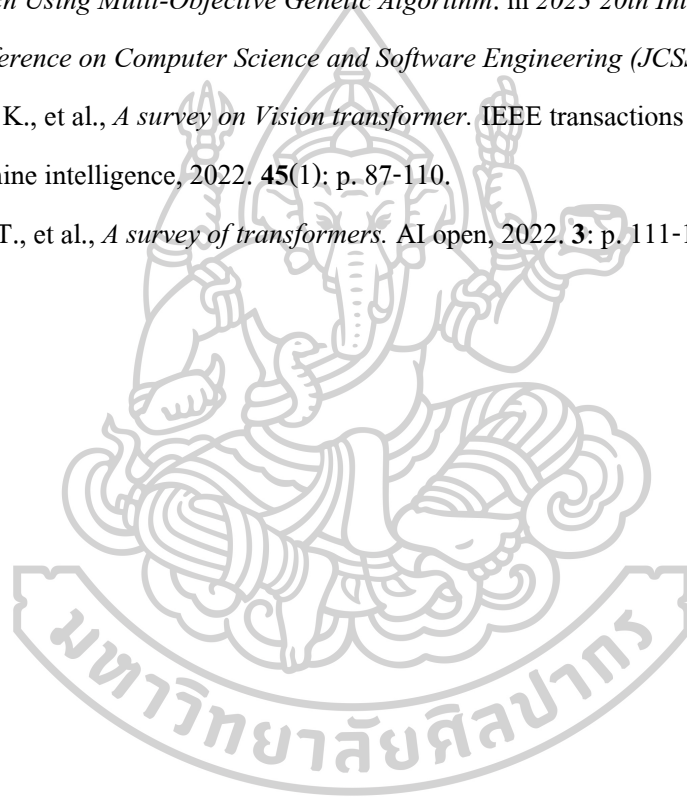
พิจารณาวัตถุประสงค์อื่นเพิ่มเติม เช่น FLOPS (Floating point operations per second) และการประยุกต์ใช้แนวคิดของวิธีการเข้ารหัสในวิทยานิพนธ์นี้กับโมเดลใหม่



รายการอ้างอิง

1. Zhou, X., et al., *A survey on evolutionary construction of deep neural networks*. IEEE Transactions on Evolutionary Computation, 2021. **25**(5): p. 894-912.
2. Elsken, T., J.H. Metzen, and F. Hutter, *Efficient multi-objective Neural architecture search via lamarckian evolution*. arXiv preprint arXiv:1804.09081, 2018.
3. White, C., et al., *A study on encodings for Neural architecture search*. Advances in neural information processing systems, 2020. **33**: p. 20309-20319.
4. Zhang, T., et al., *AS-NAS: Adaptive scalable Neural architecture search with reinforced evolutionary algorithm for deep learning*. IEEE Transactions on Evolutionary Computation, 2021. **25**(5): p. 830-841.
5. Sharifi, M.R., et al., *A new optimization algorithm to solve multi-objective problems*. Scientific Reports, 2021. **11**(1): p. 20326.
6. Zhan, Z.-H., J.-Y. Li, and J. Zhang, *Evolutionary deep learning: A survey*. Neurocomputing, 2022. **483**: p. 42-58.
7. Zitzler, E., M. Laumanns, and S. Bleuler, *A tutorial on evolutionary multiobjective optimization*. Metaheuristics for multiobjective optimisation, 2004: p. 3-37.
8. Deb, K., et al., *A fast and elitist multiobjective genetic algorithm: NSGA-II*. IEEE transactions on evolutionary computation, 2002. **6**(2): p. 182-197.
9. Rostamian, A., S. Jamshidi, and E. Zirbes, *The development of a novel Multi-objective optimization framework for Non-vertical well placement based on a modified Non-dominated sorting genetic algorithm-II*. Computational Geosciences, 2019. **23**: p. 1065-1085.
10. Vaswani, A., *Attention is all you need*. Advances in Neural Information Processing Systems, 2017.
11. Alexey, D., *An image is worth 16x16 words: Transformers for image recognition at scale*. arXiv preprint arXiv: 2010.11929, 2020.
12. Elsken, T., J.H. Metzen, and F. Hutter, *Neural architecture search: A survey*. Journal of Machine Learning Research, 2019. **20**(55): p. 1-21.
13. Xie, L. and A. Yuille. *Genetic cnn*. in *Proceedings of the IEEE international conference on*

- computer vision*. 2017.
14. Lu, Z., et al. *Nsga-net: Neural architecture search using multi-objective genetic algorithm*. in *Proceedings of the genetic and evolutionary computation conference*. 2019.
 15. Wu, J., B. Feng, and Y. Sun. *Genetic Algorithm-based Transformer Architecture Design for Neural Machine Translation*. in *Proceedings of the 2022 5th International Conference on Machine Learning and Machine Intelligence*. 2022.
 16. Patcharabumrung, P. and Y. Jewajinda. *Effects of Genetic Operators on Neural architecture search Using Multi-Objective Genetic Algorithm*. in *2023 20th International Joint Conference on Computer Science and Software Engineering (JCSSE)*. 2023. IEEE.
 17. Han, K., et al., *A survey on Vision transformer*. *IEEE transactions on pattern analysis and machine intelligence*, 2022. **45**(1): p. 87-110.
 18. Lin, T., et al., *A survey of transformers*. *AI open*, 2022. **3**: p. 111-132.



ประวัติผู้เขียน

ชื่อ-สกุล

สุวิภา สชีวผาสุก

วุฒิการศึกษา

ปริญญาตรี

วิศวกรรมศาสตรบัณฑิต (วศ.บ.) วิศวกรรมอิเล็กทรอนิกส์และระบบคอมพิวเตอร์ สำเร็จการศึกษาปี 2563

ผลงานตีพิมพ์

Sacheewapasuk, S. and Y. Jewajinda (2024). An Evolutionary Multi-objective Optimization of Vision Transformer. 2024 8th International Conference on Information Technology (InCIT), IEEE.

