

บทที่ 2

เอกสารงานวิจัยที่เกี่ยวข้อง

ในบทนี้จะกล่าวถึงทฤษฎีของเจเนติกส์อัลกอริทึมอย่างง่าย เจเนติกส์อัลกอริทึมแบบกระจายและปัญหาการเดินทางของเซลส์แมน (Traveling Salesman Problem หรือ TSP) รวมถึงงานวิจัยที่นำเจเนติกส์อัลกอริทึมอย่างง่ายและเจเนติกส์อัลกอริทึมแบบกระจายไปประยุกต์ใช้ในการแก้ไขปัญห และตัวอย่างการกำหนดพารามิเตอร์ของทั้งอัลกอริทึมทั้งสองแบบ เพื่อนำไปเป็นแนวทางในการศึกษาเปรียบเทียบเพื่อวัดประสิทธิภาพและหาข้อดีข้อเสีย

เจเนติกส์อัลกอริทึม (Genetic algorithm)

1. ความเป็นมาของ Genetic algorithm

Genetic เป็นการถ่ายทอดลักษณะทางพันธุกรรมจากบรรพบุรุษไปสู่รุ่นลูกหลาน โดยถ่ายทอดผ่านโครโมโซม ในปี ค.ศ.1895 Charls Darwin ได้มีการพิมพ์หนังสือเผยแพร่เรื่อง จุดกำเนิดของสปีชีส์ ทำให้ทราบถึงวิธีการทางธรรมชาติที่เกิดขึ้น ซึ่ง Genetic มีหลักการถ่ายทอดทางพันธุกรรมคือ สิ่งมีชีวิตตามธรรมชาติจะมีการถ่ายทอดลักษณะเฉพาะของตนเองไปยังรุ่นลูกหลาน เมื่อสิ่งมีชีวิตแต่ละชนิดเกิดการผสมพันธุ์ในแต่ละรุ่นทำให้เกิดความหลากหลายของสิ่งมีชีวิต ซึ่งสิ่งมีชีวิตที่สามารถปรับตัวเข้ากับสภาพแวดล้อมและมีความแข็งแรง จะมีโอกาสอยู่รอดในธรรมชาติสูง ถ้าสิ่งแวดล้อมเกิดการเปลี่ยนแปลง สิ่งมีชีวิตอาจจะเกิดการกลายพันธุ์เพื่อให้สามารถดำรงชีวิตอยู่กับสภาพแวดล้อมที่เกิดการเปลี่ยนแปลงนั้นได้ ในปี ค.ศ. 1975 John Holland ได้นำหลักการทางธรรมชาตินี้มาประยุกต์ใช้ในการแก้ปัญหาการหาคำตอบที่เหมาะสมที่สุดของปัญหาและใช้ชื่อว่า “เจเนติกส์อัลกอริทึม”

2. การทำงานของ Genetic algorithm

การที่จะนำ เจเนติกส์อัลกอริทึม ไปประยุกต์ใช้ในการแก้ปัญหา ต้องทราบถึงหลักการทำงานของเจเนติกส์อัลกอริทึมว่าเหมาะสมกับปัญหาที่จะนำไปใช้หรือไม่ เจเนติกส์อัลกอริทึมใช้ในการค้นหาคำตอบที่เหมาะสมที่สุดในปัญหานั้นๆ ต้องมีการกำหนดฟังก์ชัน ความเหมาะสม และกำหนดรูปแบบโครโมโซม ซึ่งภายในโครโมโซมจะประกอบไปด้วยยีนส์ (Genes) ที่มีเงื่อนไขใน

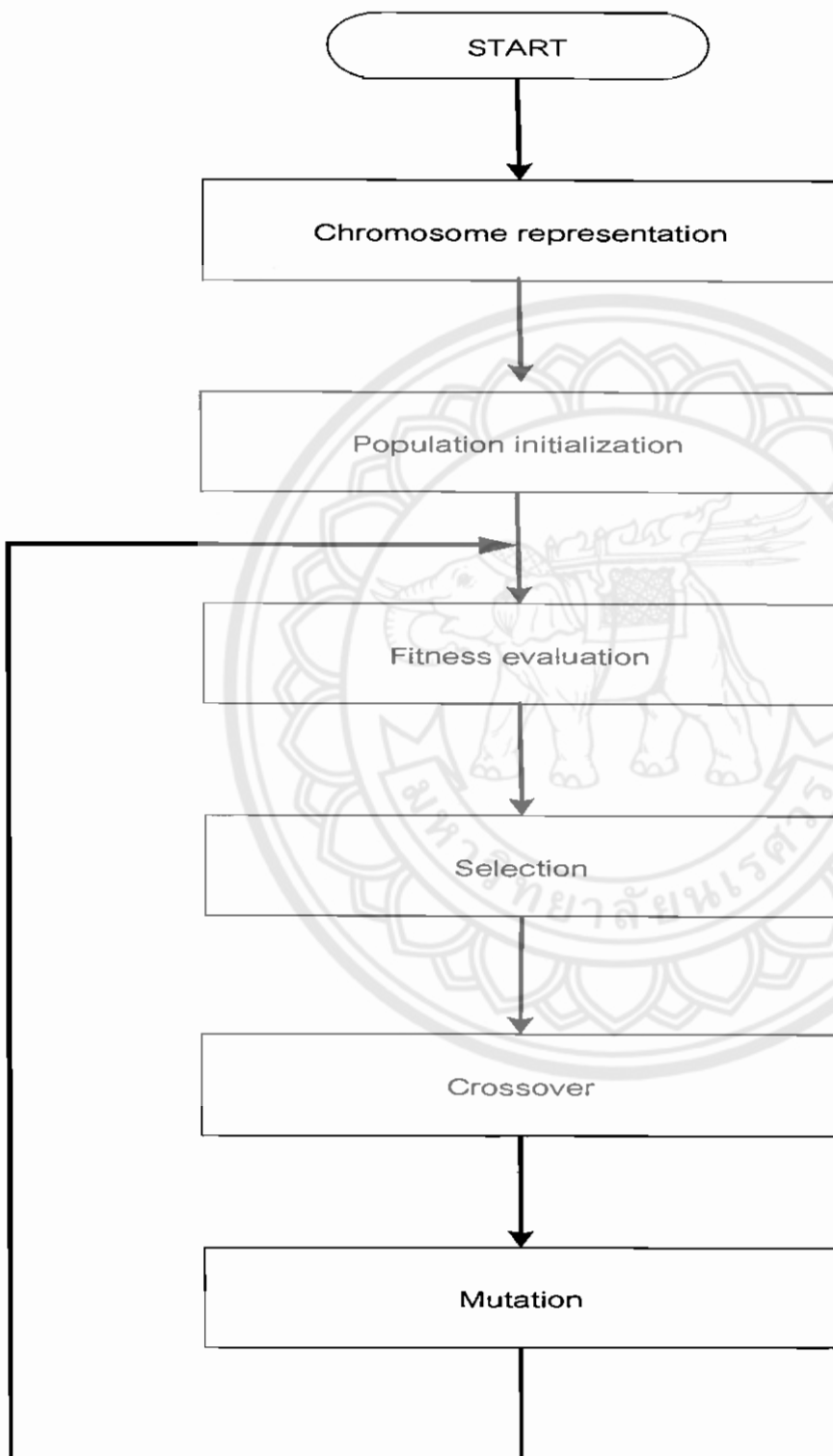
ตามขนาดของประชากรที่กำหนด เมื่อได้ประชากรต้นกำเนิดแล้วนำโครโมโซมของประชากรต้นกำเนิดมาหาค่าความเหมาะสม (Fitness) และทำการคัดเลือกโครโมโซม (Selection) หลังจากทำการคัดเลือกโครโมโซมแล้วทำการแลกเปลี่ยนเงื่อนไขที่อยู่ภายในโครโมโซมที่เรียกว่าการสลับสายพันธุ์ (Crossover) และสุ่มเลือกโครโมโซมบางตัวมาทำการสลับตำแหน่งเงื่อนไขของตัวเองที่เรียกว่าการกลายพันธุ์ (Mutation) โครโมโซมที่เกิดขึ้นใหม่จากการสลับสายพันธุ์และการกลายพันธุ์จะเรียกว่าโครโมโซมลูก (Offspring) โครโมโซมลูกซึ่งเกิดเป็นประชากรรุ่นต่อไป กระบวนการของเจเนติกส์อัลกอริทึมจะทำการวนซ้ำจนกว่าจะครบตามรุ่นที่ผู้ใช้กำหนดการทำงานของเจเนติกส์อัลกอริทึมสามารถแสดงเป็นโครงสร้าง (M.Srinivas,Lalit M. Patnaik,1994) และแผนผังได้ดังนี้

```

Simple genetic algorithm ()
[
    population initialize;
    population evaluate;
    While termination criterion not reached
    [
        select solutions for next population;
        perform crossover and mutation;
        population evaluate;
    ]
]

```

แสดงโครงสร้างการทำงานของ Simple genetic algorithm



ภาพ 1 แสดงโครงสร้างการทำงานของ Simple genetic algorithm

3. การกำหนดรูปแบบโครโมโซม (Chromosome representation)

การกำหนดรูปแบบของโครโมโซมเป็นขั้นตอนแรกของ Genetic algorithm ภายในโครโมโซมจะประกอบไปด้วยยีนส์หลายตัวนำมาเรียงต่อกัน ใน 1 โครโมโซมจะมียีนส์ทั้งหมดที่ตัวนั้นผู้ใช้เป็นผู้กำหนด โครโมโซมแต่ละตัวต้องมีความยาวเท่ากัน ในแต่ละยีนส์คือเงื่อนไขที่ใช้ในการแก้ปัญหา เงื่อนไขที่อยู่ในแต่ละยีนส์จะไม่เหมือนกัน การจัดเรียงของยีนส์ในแต่ละโครโมโซมจะมีตำแหน่งการจัดเรียงที่แตกต่างกัน

4. สร้างประชากรต้นกำเนิด (Population initialization)

การสร้างประชากรต้นกำเนิดเป็นการสุ่มเลือกยีนส์ (Genes) จำนวนหนึ่งมาประกอบกันเป็นโครโมโซมที่แทนผลเฉลยของคำตอบซึ่งจะสุ่มสร้างเป็นประชากรต้นกำเนิดจำนวนเท่าใดนั้นผู้ใช้จะต้องเป็นผู้กำหนดแต่ต้องกำหนดให้เหมาะสมกับปัญหาที่นำมาประยุกต์ใช้

5. การประเมินค่าความเหมาะสม (Fitness evaluation)

การประเมินค่าความเหมาะสมเป็นการนำโครโมโซมมาถอดรหัสเพื่อหาค่าว่าโครโมโซมตัวใดมีค่าความเหมาะสมมากที่สุด เพื่อความอยู่รอดของโครโมโซมและนำค่าความอยู่รอดไปเข้ากระบวนการคัดสรรต่อไป

6. การคัดสรร (Selection)

การคัดสรรเป็นการเลือกนำเอาโครโมโซมที่มีค่าความเหมาะสมมากที่สุดไปเป็นประชากรในรุ่นต่อไป การคัดสรรสามารถทำได้หลายวิธี เช่น Ordinal Selection ได้แก่ การคัดสรรแบบ tournament selecting (Goldberg et al., 1989b) และวิธี truncation selection (Mühlenbein and Schlierkamp-Voosen, 1993) สำหรับวิธี tournament selecting โครโมโซม s โครโมโซมจะถูกเลือกมาแบบสุ่ม (อาจแทนที่หรือไม่แทนที่ก็ได้) แล้วเข้าร่วม tournament กับแต่ละตัว ตัวที่เหมาะสมที่สุดในกลุ่มโครโมโซม k โครโมโซมเป็นผูชนะ tournament และจะได้รับเลือกให้เป็นพ่อแม่ (Parent) ส่วนใหญ่แล้วจะใช้ค่า $s=2$ การคัดสรรวิธีนี้ต้องทำ tournament n ครั้งเพื่อให้ได้สมาชิก n ตัว ส่วนวิธี truncation selection สมาชิกตัวที่อยู่ในกลุ่มบนของ $\frac{1}{s}$ แต่ละตัวจะมี s ลำเนาในประชากรรุ่นต่อไป แต่วิธีที่นิยมมากที่สุดคือวงล้อรูเล็ตต์ (Roulette wheel selection) หลักการทำงานของวงล้อรูเล็ตต์ (Kumara Sastry, David Goldberg, 2004) คือการสุ่มเลือกโครโมโซมที่มีค่าความเหมาะสม

กำหนดลงในในวงล้อรูเล็ตโดยมีขนาดช่องแปรผันตามค่าความเหมาะสมของสมาชิก วงล้อรูเล็ตที่แบ่งช่องแล้วโครโมโซมที่มีความเหมาะสมมากจะมีช่องที่กว้างกว่าโครโมโซมที่มีค่าความเหมาะสมน้อย โดยมีขั้นตอนดังต่อไปนี้

6.1 คำนวณค่าความเหมาะสม f_i ของสมาชิกแต่ละตัวในกลุ่มประชากร

6.2 คำนวณหาค่าความน่าจะเป็น ขนาดของช่อง p_i ของสมาชิกแต่ละตัวที่นำมา

คัดสรรโดย $p_i = \frac{f_i}{\sum_{j=1}^n f_j}$ เมื่อ n เป็นขนาดประชากร

6.3 คำนวณหาค่าความเหมาะสม q_i ของสมาชิกแต่ละตัว โดย $q_i = \sum_{j=1}^i p_j$

6.4 สร้างตัวเลขแบบสุ่มขึ้นมาตัวหนึ่งขึ้นมาในช่วง $r \in (0,1)$

6.5 ถ้า $r < q_1$ แล้วจะเลือกโครโมโซม x_1 แต่ถ้าไม่ใช่คือ $q_{i-1} < r \leq q_i$ จะเลือก $\frac{1}{s}$

6.6 ทำขั้นตอนที่ 1.6.4 - 1.6.5 ซ้ำเป็นจำนวน n ครั้งเพื่อสร้างสมาชิก n ตัว

สามารถแสดงได้ดังตัวอย่างต่อไปนี้

พิจารณากลุ่มประชากรที่มีสมาชิก 5 ตัว ($n=5$), มีค่าความเหมาะสมดังตาราง

ค่าความเหมาะสม = $\sum_{j=1}^n f_j = 28+18+14+9+26 = 95$ ความน่าจะเป็นของสมาชิกที่นำมา

คัดสรรกับความน่าจะเป็นสะสมแสดงดังตาราง 1

ตาราง 1 แสดงตัวอย่างการคัดสรร

Chromosome#	1	2	3	4	5
Fitness, f	28	18	14	9	26
Probability, p_i	$28/95=0.295$	0.189	0.147	0.095	0.274
Cumulative probability, q_i	0.295	0.484	0.631	0.726	1.000

สร้างตัวเลขแบบสุ่มสมมติว่าสุ่มได้ $r = 0.585$ มีค่า $q_2 = 0.484 < 0.585 \leq q_3 = 0.631$

7. การสลับสายพันธุ์ (Crossover)

การสลับสายพันธุ์ (Crossover) เป็นการนำโครโมโซมมาแลกเปลี่ยนยีนซึ่งกันและกันเพื่อให้โครโมโซมมีการเปลี่ยนแปลงไปเป็นคำตอบที่เหมาะสมที่สุดในการแก้ไขปัญหา โดยทำการสุ่มเลือกโครโมโซมจำนวน 2 โครโมโซม กำหนดให้เป็นโครโมโซมพ่อ 1 โครโมโซมและโครโมโซมแม่ 1 โครโมโซม เมื่อทำการแลกเปลี่ยนโครโมโซมแล้วจะได้โครโมโซมใหม่ขึ้นมา 2 ชุด คือโครโมโซมลูกชุดที่ 1 และโครโมโซมลูกชุดที่ 2 การสลับสายพันธุ์แต่ละครั้งไม่ได้หมายความว่าโครโมโซมทุกตัวจะมีการเปลี่ยนแปลงที่ดีขึ้นเสมอ และการสลับสายพันธุ์มีข้อที่ควรระวังคือการที่สุ่มเลือกโครโมโซมขึ้นมาเพื่อทำการสลับสายพันธุ์นั้นถ้าโครโมโซมที่ถูกสุ่มเลือกทั้งโครโมโซมพ่อและโครโมโซมแม่ 2 โครโมโซมเหมือนกันก็จะทำให้โครโมโซมรุ่นลูกไม่เกิดการเปลี่ยนแปลงใดๆ ในการแลกเปลี่ยนโครโมโซม ทำให้เสียเวลาในการค้นหาคำตอบ การสลับสายพันธุ์สามารถทำได้หลายวิธี ดังนี้

7.1 การสลับสายพันธุ์แบบ one point (Murata,Ishibuchi,&Tanaka,1996)

จะสุ่มเลือกโครโมโซมขึ้นมา 2 โครโมโซมเพื่อกำหนดเป็นโครโมโซมพ่อและโครโมโซมแม่และทำการกำหนดยีนในส่วนแรกของโครโมโซมพ่อและทำการถ่ายทอดไปยังโครโมโซมลูกชุดที่ 1 โดยส่วนที่เหลือของโครโมโซมลูกชุดที่ 1 จะได้รับการถ่ายทอดมาจากโครโมโซมแม่ และโครโมโซมลูกชุดที่ 2 จะได้รับการถ่ายทอดยีนในส่วนแรกมาจากโครโมโซมแม่ ส่วนที่เหลือของโครโมโซมลูกชุดที่ 2 จะได้รับการถ่ายทอดมาจากโครโมโซมพ่อ

โครโมโซมพ่อ [a**b**cdefgh] โครโมโซมแม่ [e**h**adbcgf]

โครโมโซมลูกชุดที่ 1 [a**b**cehdgf] โครโมโซมลูกชุดที่ 2 [e**h**abcdgf]

7.2 การสลับสายพันธุ์แบบ two point แบบที่หนึ่ง

(Murata,Ishibuchi,&Tanaka,1996)

ทำการสุ่มโครโมโซมพ่อและแม่เพื่อทำการสลับสายพันธุ์และสุ่มเลือกยีน 2 จุดของโครโมโซมพ่อและแม่ที่ต้องการทำการสลับสายพันธุ์ เพื่อถ่ายทอดยีน 2 ส่วนของโครโมโซมพ่อ และทำการถ่ายทอดไปยังโครโมโซมลูกชุดที่ 1 โดยส่วนที่เหลือของโครโมโซมลูกชุดที่ 1 จะได้รับการถ่ายทอดมาจากโครโมโซมแม่ และโครโมโซมลูกชุดที่ 2 จะได้รับการถ่ายทอดยีน 2 ส่วนมาจากโครโมโซมแม่ ที่เหลือของโครโมโซมลูกชุดที่ 2 จะได้รับการถ่ายทอดมาจากโครโมโซมพ่อ

โครโมโซมพ่อ [a**b**cdefgh] โครโมโซมแม่ [e**h**adbcgf]

โครโมโซมลูกชุดที่ 1 [a**b**edc**g**fh] โครโมโซมลูกชุดที่ 2 [e**h**abcdgf]

7.3 การสลับสายพันธุ์แบบ two point แบบที่สอง

(Murata,Ishibuchi,&Tanaka.,1996)

ทำการสุ่มโครโมโซมพ่อและแม่เพื่อทำการสลับสายพันธุ์และสุ่มเลือกยีนส์ 2 จุด ของโครโมโซมพ่อและแม่ ที่ต้องการทำการสลับสายพันธุ์โดยจุดเริ่มต้นและจุดลงท้ายเป็น ช่วงติดกัน ถ่ายทอดยีนส์โครโมโซมพ่อไปยังโครโมโซมลูกชุดที่ 1 โดยส่วนที่เหลือของโครโมโซมลูก ชุดที่ 1 จะได้รับการถ่ายทอดมาจากโครโมโซมแม่ และโครโมโซมลูกชุดที่ 2 จะได้รับการถ่ายทอด ยีนส์ 2 ส่วนมาจากโครโมโซมแม่ ส่วนที่เหลือของโครโมโซมลูกชุดที่ 2 จะได้รับการถ่ายทอดมาจาก โครโมโซมพ่อ

โครโมโซมพ่อ [a**b**cdefgh] โครโมโซมแม่ [eha**d**bcbgf]

โครโมโซมลูกชุดที่ 1 [ha**c**def bg] โครโมโซมลูกชุดที่ 2 [efa**d**bcbgh]

7.4 การสลับสายพันธุ์แบบ enhanced edge recombination

(Starkweather et al.,1991)

การสลับสายพันธุ์แบบ enhanced edge recombination นั้นมีการสร้าง ลิสต์โดยสร้างโดยเริ่มดูจากยีนส์ตัวแรกในโครโมโซมพ่อ ว่าตัวข้างเคียงของโครโมโซมคือยีนส์ อะไรแล้วจึงนำมาไว้ในลิสต์และต้องตรวจสอบว่าในโครโมโซมแม่ยีนส์ข้างเคียงที่อยู่ในโครโมโซม แม่เป็นตัวเดียวกับโครโมโซมพ่อหรือไม่ถ้าใช่ให้ใส่วงเล็บและให้สิทธิ์ในการถ่ายทอดไปสู่รุ่นลูก ก่อนยีนส์ตัวอื่น ทำการเปรียบเทียบสร้างลิสต์ไปที่ละยีนส์

โครโมโซมพ่อ [a**b**cdefgh]

โครโมโซมแม่ [bafcdhge]

โครโมโซมลูกชุดที่ 1 [a**b**cdhgf e]

โครโมโซมลูกชุดที่ 2 [bafcdhge]

gene: edge List
a:(b),f,h
b:(a),c,e
c:b,(d),f
d:(c),e,h
e:b,d,f,g
f:a,c,e,g
g:e,f,(h)
h:a,d,(g)

8. การผ่าเหล่า (Mutation)

การผ่าเหล่า (Mutation) เป็นการทำให้โครโมโซมเกิดความแตกต่างและเป็นการเพิ่มโอกาสในการค้นหาคำตอบที่ดีขึ้น โดยมีวิธีการผ่าเหล่าดังนี้ สุ่มเลือกโครโมโซมต้นแบบจำนวน 1 โครโมโซมและสุ่มยีนส์ (Genes) ในโครโมโซมเพื่อทำการสลับตำแหน่ง กลายเป็นโครโมโซมลูก การผ่าเหล่าสามารถทำได้หลายวิธีดังนี้

8.1 การผ่าเหล่าแบบ Inversion (Goldberg,D.E., 1989)

สุ่มเลือกโครโมโซมขึ้นมาจำนวน 1 โครโมโซมกำหนดตำแหน่งเริ่มต้นและลงท้าย ในการสลับตำแหน่งของยีนส์ เพื่อไปเป็นโครโมโซมลูก

โครโมโซมต้นแบบ [a b c d e f g h]

โครโมโซมลูก [a b f e d c g h]

8.2 การผ่าเหล่าแบบ Binary Mutation (บังอร กลับบ้านเกาะ, 2000)

สุ่มเลือกโครโมโซมขึ้นมาจำนวน 1 โครโมโซมและสุ่มเลือกยีนส์ 1 ตำแหน่ง ทำการเปลี่ยนค่าของยีนส์ตัวนั้น

โครโมโซมต้นแบบ [a b c d e f g h]

โครโมโซมลูก [a b c e e f g h]

8.3 การผ่าเหล่าแบบ Adjacent two genes change (Murata et al.,1996)

สุ่มเลือกโครโมโซมขึ้นมา 1 ตัวแล้วทำการสุ่มยีนส์ที่อยู่ติดกันมา 1 คู่แล้วทำการสลับตำแหน่งของยีนส์ทั้งสอง

โครโมโซมต้นแบบ [a b c d e f g h]

โครโมโซมลูก [a c b d e f g h]

เจเนติกส์อัลกอริทึมแบบกระจาย (Distributed Genetic Algorithm)

เจเนติกส์อัลกอริทึมแบบกระจาย คือ การนำเจเนติกส์อย่างง่ายมาประมวลผลพร้อมกัน ซึ่งช่วยลดเวลาในการประมวลผล และค้นหาคำตอบดีขึ้น วิธีการทำงานของเจเนติกส์อัลกอริทึมแบบกระจาย นำประชากรทั้งหมดมาแบ่งประชากรให้เป็นประชากรกลุ่มย่อย (Subpopulation) และแต่ละกลุ่มสามารถทำงานตามกระบวนการของเจเนติกส์ได้อย่างเป็นอิสระ ซึ่งช่วยให้ประชากรย่อยแต่ละกลุ่มมีการทำงานที่เป็นอิสระต่อกัน และยังคงรักษาสมาชิกตัวที่มีความเหมาะสมสูงไว้ และควบคุมว่าจะอพยพแลกเปลี่ยนประชากรระหว่างกลุ่มอย่างไร เจเนติกส์อัลกอริทึมแบบกระจายมีหลายวิธี สามารถอธิบายเจเนติกส์อัลกอริทึมแบบกระจายบางแบบ ดังนี้ (Shyh-Chang Lin et al, 1994)

1. Micro-Grain

Micro-Grain ต่างจากวิธีการอื่นคือ มีกลุ่มประชากรเพียงกลุ่มเดียว การประมวลผลมาจากการใช้โนดประมวลผลหลายโนดในการคำนวณค่าความเหมาะสมของโครโมโซมแต่ละตัว Micro-Grain จะส่งหน่วยประมวลผลในแต่ละที่โดยแยกออกจากกัน ทำให้เวลาในการคำนวณ (Evaluation time) ของประชากรทั้งหมดเท่ากับเวลาที่ใช้ในการคำนวณค่าของโครโมโซมตัวที่ใช้เวลานานที่สุด ถ้าใช้จำนวนของหน่วยประมวลผลน้อยลงกว่าคำตอบที่มีอยู่ แต่ละโนดก็ต้องประมวลผลประชากรกลุ่มย่อย ทำให้เวลาในการคำนวณทั้งหมดมีค่าเทียบเท่ากับเวลาในการคำนวณประชากรกลุ่มย่อยกลุ่มที่ต้องการเวลานานที่สุด การดำเนินงานทาง genetic operation จะทำที่ Master เพียงโนดเดียว Master จะเป็นตัวที่ควบคุมระบบ วิธี Micro-Grain มีประโยชน์เหมาะสมสำหรับปัญหาที่มีการคำนวณ ตามฟังก์ชันคำนวณที่ต้องใช้เวลานานที่สุดเมื่อเทียบกับการดำเนินงานทางเจเนติกส์ชนิดอื่น วิธีการ Micro-Grain มีข้อดีในเบื้องต้น คือ เร็วกว่า Sequential เจเนติกส์อัลกอริทึม

2. Fine-Grain

วิธีการ Fine-Grain จะกำหนดโครโมโซมหนึ่งโครโมโซมของประชากรให้กับตัวประมวลผลแต่ละตัว ซึ่งจะประมวลผลพร้อมกัน โครโมโซมแต่ละโครโมโซมเป็นส่วนหนึ่งของประชากรย่อยหลายกลุ่ม คุณสมบัติการเป็นสมาชิกหาได้จากลักษณะการเชื่อมต่อเป็นเครือข่ายของตัวประมวลผล ดังนั้น ประชากรทั้งหมด จึงถูกมองได้ว่าเป็นประชากรกลุ่มย่อยเล็ก ๆ หลาย

เครือข่ายเป็นตัวกำหนดว่าการแยกกลุ่มของจำนวนประชากรนั้นแยกจากกันมากน้อยเท่าใด และความหลากหลายของโครโมโซม แต่ละตัวในกลุ่มประชากร ปัญหาหลักในการแก้ปัญหาด้วย Fine-Grain คือ ลักษณะการเชื่อมต่อของเครือข่าย (Network topology) หากมีการเชื่อมต่อระหว่างเพื่อนบ้านสูง ก็จะมีการแพร่กระจายโครโมโซมตัวที่มีค่าความเหมาะสมสูงได้มากยิ่งขึ้น

3. Coarse-Grain

วิธีนี้คล้ายกับ Fine-Grain ตรงที่มีกลุ่มประชากรย่อยที่เป็นอิสระ และแลกเปลี่ยนโครโมโซมระหว่างกลุ่มเหมือนกัน แต่ Coarse-Grain จะมีจำนวนโครโมโซมในกลุ่มประชากรย่อยเป็นจำนวนมาก ไม่ได้มีเพียงโครโมโซมเดียว ผลที่ได้คือ ความถี่ของการอพยพระหว่างกลุ่มประชากรย่อยน้อยกว่าวิธี Fine-Grain มาก และผลกระทบที่เกิดขึ้นเนื่องจากการแลกเปลี่ยนระหว่างกลุ่มประชากรย่อยน้อยกว่า เพราะมีกลุ่มประชากรย่อยขนาดใหญ่ Coarse-Grain ยังแบ่งออกเป็นประเภทย่อยได้หลายประเภท ในที่นี้สามารถแยกประเภท Coarse-Grain ด้วยวิธีการสามวิธี คือ แบ่งตามวิธีการอพยพ แบ่งตามวิธีการเชื่อมต่อ และแบ่งตามคุณสมบัติที่เหมือนกัน (homogeneity) ของตัวประมวลผล

3.1 แบ่งตามประเภทการอพยพ

ประเภทการอพยพ เป็นการบอกว่าโครโมโซมมีการอพยพระหว่างโหนดบ่อยแค่ไหน ภายในเงื่อนไขเวลาเท่าใด และยังสามารถแบ่ง Coarse-Grain ตามประเภทการอพยพได้ 3 ชนิดคือ

3.1.1 Isolated Island Genetic algorithm

ประเภทนี้ไม่มีการอพยพระหว่างประชากรกลุ่มย่อย เป็นแบบจำลอง Coarse-Grain ที่ง่ายที่สุด

3.1.2 Synchronous Island Genetic algorithm

การอพยพระหว่างกลุ่มประชากรย่อยสองกลุ่มซึ่งโครโมโซมกัน ด้วยประเภทการอพยพแบบซิงโครไนซ์ (Synchronization migration) กลุ่มประชากรจะดำเนินการตามกระบวนการของเจเนติกส์ด้วยอัตราเท่ากันก่อนที่การเปลี่ยนแปลงจะเกิดขึ้น ตามจำนวนเจเนเนอเรชัน และอื่น ๆ เป็นตัวคำนวณการซิงโครไนซ์ของกลุ่มประชากรย่อย ฮาร์ดแวร์ที่นำมาทำการประมวลผลต้องสนับสนุนการซิงโครไนซ์แต่การทำซิงโครไนซ์ในสภาพแวดล้อมของ Distributed workstation สามารถทำให้เกิด Work loads ที่ไม่สมดุล ความเร็วของเครื่องที่ต่างกัน และโหนดที่

3.1.3 Asynchronous Island Genetic algorithm

Asynchronous Island Genetic algorithm ประเภทนี้ยอมให้การอพยพขึ้นกับเหตุการณ์เดียว ไม่เกี่ยวกับสถานะของกระบวนการของเจเนติกส์ของประชากรกลุ่มย่อย รูปแบบของ Asynchronous เป็นรูปแบบหนึ่งของการอพยพที่พบได้ทั่วไปในธรรมชาติ เนื่องจากสิ่งแวดล้อมต่างกันเป็นตัวทำให้ความเร็วของวิวัฒนาการต่างกัน

3.2 แบ่งตามประเภทการเชื่อมต่อ

ประเภทการเชื่อมต่อในดประมวลผลเข้าด้วยกันในเทอมของระดับของการเชื่อมต่อ (Degree of connectivity) และลักษณะการเชื่อมต่อ (Topology of connection) ส่งผลกระทบต่อประสิทธิภาพของ Coarse-Grain สามารถแบ่ง Coarse-Grain ตามประเภทการเชื่อมต่อได้สองประเภทดังนี้

3.2.1 Static Connection Scheme

ประเภทการเชื่อมต่อแบบนี้ จะกำหนดการเชื่อมต่อระหว่างโนดในตอนเริ่มต้นการทำงาน และไม่มีการเปลี่ยนแปลง ระหว่างการประมวลผลยังคงรักษาลักษณะการเชื่อมต่อ ลักษณะการเชื่อมต่อมีหลายชนิด คือ แบบ line, ring, mesh, n-cubes และอื่น ๆ ลักษณะการเชื่อมต่อเป็นตัวกำหนดว่าโนดแต่ละโนด สามารถแลกเปลี่ยนโครโมโซมกันได้ และลักษณะการเชื่อมต่อเป็นรูปแบบสถิต

3.2.2 Dynamic Connection Scheme

ลักษณะการเชื่อมต่อโนดสามารถเปลี่ยนแปลงได้ระหว่างช่วงเวลาทำงานมีเหตุผลพื้นฐานสองประการที่ยอมให้เปลี่ยนแปลงลักษณะการเชื่อมต่อระหว่างเวลาทำงาน ประการแรก การเปลี่ยนแปลงทำให้การประมวลผลของเครื่อง Workstation เป็นไปได้ในทางปฏิบัติ ถ้ากระบวนการของเจเนติกส์ ของโนดใดโนดหนึ่งหยุด แล้วจะเกิดการกำหนดการเชื่อมต่อเครือข่ายขึ้นใหม่ เพื่อให้ดำเนินกระบวนการต่อไปได้ Coarse-Grain ยอมรับต่อเหตุการณ์จำนวนหนึ่งที่เกิดขึ้นก่อนจะสูญเสียกระบวนการเจเนติกส์ ประการที่สอง การกำหนดการเชื่อมต่อขึ้นใหม่จะเกิดขึ้นเมื่อมีการเปลี่ยนแปลงในเจเนติกส์ของกลุ่มประชากรย่อย สิ่งหนึ่งที่เป็นจุดบกพร่องของ Coarse-Grain คือ การแทรกโครโมโซมตัวใหม่จากประชากรกลุ่มย่อยอื่นอาจไม่มีประสิทธิภาพ โครโมโซมตัวใหม่ อาจไม่เข้ากับประชากรกลุ่มย่อยโดยสิ้นเชิง ดังนั้นโครโมโซมเหล่านี้ก็จะถูกละทิ้ง หรือมีอิทธิพลเหนือโครโมโซมตัวอื่นในประชากรกลุ่มย่อยนั้น เพื่อหลีกเลี่ยงสิ่งเหล่านี้ ประชากรกลุ่มย่อยอาจจะเริ่มกระบวนการโดยไม่ต้องมีเพื่อนบ้าน และเมื่อเกิดการอพยพ จึงจะหาเพื่อนบ้านจากความเหมือน

มาเปรียบเทียบกับโครโมโซมตัวใดก็ตามที่มี Hamming distance ใกล้ที่สุด ก็เป็นเพื่อนบ้าน วิธีการนี้ทำให้เกิดลักษณะการเชื่อมต่อตามระยะทาง (Distance connection topology) ที่เปลี่ยนแปลงตลอดเวลา และรักษาการแลกเปลี่ยนกับกลุ่มประชากรย่อยที่เหมือนกันหรือต่างกัน เท่านั้นอยู่ตลอดเวลา

3.3 แบ่งตามคุณสมบัติที่เหมือนกัน (Node Homogeneity)

คุณสมบัติที่เหมือนกัน วัดจากความเหมือนของกระบวนการ เจเนติกส์

อัลกอริทึม ของโนดต่าง ๆ ว่าเหมือนกันอย่างไร

3.3.1 Homogeneous Island Genetic algorithm

เจเนติกส์บนโนดแต่ละโนดใช้พารามิเตอร์ เช่น ขนาดประชากร อัตราการเกิดการสลับสายพันธุ์ อัตราการผ่าเหล่า ช่วงเวลาอพยพ เป็นต้น ตัวดำเนินการเจเนติกส์ ฟังก์ชันวัตถุประสงค์ (Objective function) และวิธีการเข้ารหัสเหมือนกันทุกประการ PGA's ข้อดีข้อหนึ่งของวิธีนี้คือ นำมาประยุกต์ได้ง่าย

3.3.2 Heterogeneous Island Genetic algorithm

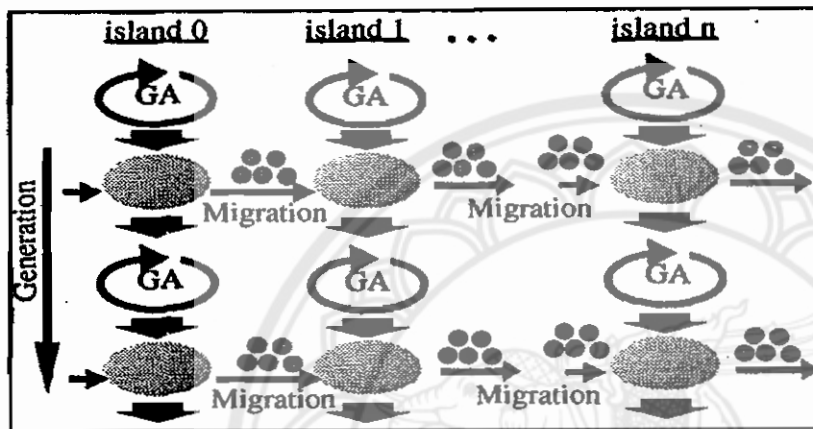
Heterogeneous Island Genetic algorithm ยอมให้วิวัฒนาการของกลุ่มประชากรย่อย ผ่านเจเนติกส์ มีพารามิเตอร์ ตัวดำเนินการเจเนติกส์ ฟังก์ชันวัตถุประสงค์ (Objective function) และวิธีการเข้ารหัสต่างกัน โดยทั่วไปจะหาการตั้งค่าพารามิเตอร์ในตอนเริ่มต้นของเจเนติกส์นั้นยาก โดยเฉพาะในกรณีที่เป็นปัญหาเป็นแบบที่ต้องการหาค่าความเหมาะสมแบบหลายวัตถุประสงค์ (Multiple objective optimization problem) ซึ่งการตั้งค่าพารามิเตอร์หลายตัว ทำให้เจเนติกส์ค้นหาไปยังส่วนที่แตกต่างไปใน search space สิ่งสำคัญกว่าคือ ปัญหาหลายปัญหามารถเข้ารหัสในเจเนติกส์โดยใช้วิธีการต่างกัน วิธีเหล่านี้มีข้อดีและข้อเสียต่างกัน การเปลี่ยนแปลงการตั้งค่าพารามิเตอร์และวิธีการเข้ารหัส ทำให้เกิดปัญหาในการแลกเปลี่ยนระหว่างประชากรแต่ละกลุ่มแต่ ปัญหาเหล่านี้สามารถแก้ไขได้

4. พารามิเตอร์ที่เป็นตัวกำหนดกลไกการอพยพ

4.1 ขนาดของการอพยพ (Migration size)

ขนาดของการอพยพ (Migration size) คือ จำนวนของโครโมโซมที่เป็นผู้อพยพในการอพยพแต่ละครั้งไปยังกลุ่มประชากรอื่น เช่น อัตราการอพยพเป็น 5 % หมายความว่า ถ้า

ออกไปก็จะเป็น 5 โครโมโซม และในทำนองเดียวกันก็จะมีจำนวนโครโมโซม ที่อพยพเข้ามาเป็น 5 โครโมโซม ด้วยเช่นกัน ทั้งนี้เพื่อให้จำนวนโครโมโซม ของแต่ละกลุ่มประชากรยังคงมีจำนวนรวมเป็น 100 เท่าเดิม แสดงได้ดังภาพ 2



ภาพ 2 แสดงช่วงระยะเวลาการอพยพและอัตราการอพยพ (Tomoyuki et al., 1999)

4.2 ความถี่ในการอพยพ (Migration frequency)

ความถี่ในการอพยพ (Migration frequency) คือ จำนวนเจเนอเรชันที่คั่นกลางระหว่างการอพยพหนึ่งจนถึงเวลาของการเกิดการอพยพครั้งต่อไป การเกิดการอพยพสามารถกำหนดได้ทั้งแบบคงที่หรือแบบสุ่ม โดยทั่วไปของการเกิดการอพยพนั้นจะเป็นเวลาที่เทียบกับจำนวนเจเนอเรชันในกระบวนการเจเนติกส์ ซึ่งต่อไปจะเรียกการเกิดการอพยพว่า ช่วงระยะเวลาการอพยพ (Migration interval) ตัวอย่างเช่น การอพยพเกิดขึ้นทุก 5 เจเนอเรชัน จะเรียกว่า มีช่วงระยะเวลาการอพยพเท่ากับ 5 เจเนอเรชัน เป็นต้น

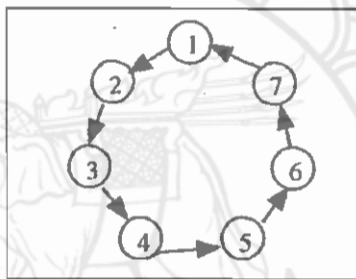
4.3 วิธีการอพยพ (Migration strategy)

วิธีการอพยพ (Migration strategy) คือ การคัดเลือกโครโมโซม ที่จะเป็นผู้อพยพภายในประชากรกลุ่มย่อยต้นทาง การอพยพโครโมโซม ระหว่างเกาะ มีด้วยกันหลายรูปแบบ เช่น แบบ stepping stone model แบบ random migration model และ unrestricted migration เป็นต้น วิธีการคัดเลือกโครโมโซมในการอพยพสำหรับเจเนติกส์อัลกอริทึมแบบกระจายมีหลายรูปแบบ เช่น การเลือกโครโมโซมตัวที่ดีในกลุ่มประชากรย่อยเท่านั้นเป็นโครโมโซมที่จะถูกอพยพ (D.Whitley,

et.al., 1997) หรือ เลือกโครโมโซมจำนวนหนึ่งมาแบบสุ่ม และใช้วิธีการคัดเลือกแบบ tournament เพื่อคัดสรรโครโมโซมที่ผ่านเข้ารอบไว้เพื่อการอพยพ (V.S. Gordon and D.Whitley., 1993) เป็นต้น

4.3.1 การอพยพแบบ Stepping stone model (J.Nang & K.Matsuo, 1994)

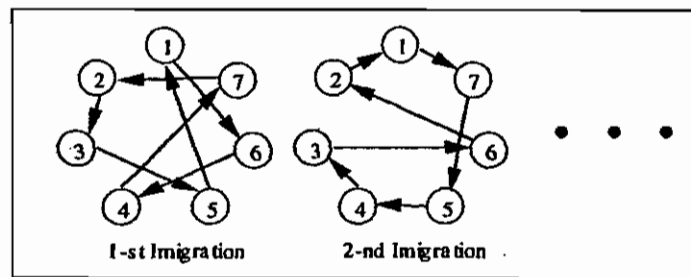
จำนวนของโครโมโซม จากเกาะหนึ่งจะถูกส่งไปยังอีกเกาะหนึ่งอย่างเป็นลำดับ โดยเกาะซึ่งเป็นเกาะเป้าหมายนั้นจะถูกกำหนดไว้แน่นอน สำหรับทุกช่วงระยะการอพยพ เช่น โครโมโซม จากเกาะที่ 1 จะถูกส่งไปยังเกาะที่ 2 และ โครโมโซม จากเกาะที่ 2 จะถูกส่งไปยังเกาะที่ 3 และต่อ ๆ ไป ลำดับของการอพยพนี้จะมีลำดับเหมือนเดิมสำหรับทุกช่วงระยะการอพยพ ซึ่งแสดงไว้ในภาพ 3 การอพยพแบบ stepping stone model นี้ถูกออกแบบไว้เหมาะสำหรับใช้กับโครงข่ายที่มีการประมวลผลแบบขนานแบบแมสซีฟ (massive parallel computing networks)



ภาพ 3 การอพยพแบบ Stepping stone mode

4.3.2 การอพยพแบบ Random migration model (J.Nang & K.Matsuo, 1994)

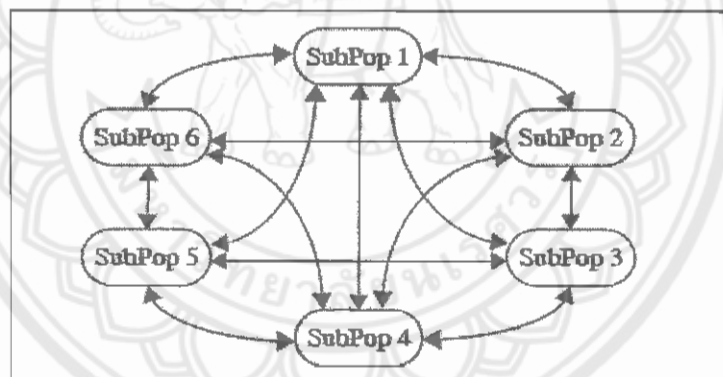
จำนวนของโครโมโซมจากเกาะหนึ่งจะถูกส่งไปยังอีกเกาะหนึ่ง ซึ่งเกาะเป้าหมายนั้นจะถูกกำหนดไว้แบบสุ่มทุกครั้งที่เกิดการอพยพ เช่น การอพยพรอบแรก ลำดับของเกาะในการส่งโครโมโซมสุ่มเป็น 1 -> 6 -> 4 -> 7 -> 2 -> 3 -> 5 -> 1 เมื่อเกิดการอพยพครั้งที่สอง ลำดับของเกาะในการส่งโครโมโซมจะสุ่มเป็น 1 -> 7 -> 5 -> 4 -> 3 -> 6 -> 2 -> 1 และจะสุ่มลำดับของเกาะสำหรับการอพยพทุกครั้ง ซึ่งแสดงไว้ในภาพ 4 การอพยพแบบ random migration model นี้เหมาะสำหรับเครื่อง PC ที่มีการแบ่งคลัสเตอร์เพื่อประมวลแบบขนาน (PC cluster parallel machines)



ภาพ 4 การอพยพแบบ Random migration model

4.3.3 การอพยพแบบ Unrestricted migration (Andrew Chipfield et. al.)

วิธีนี้ individuals อาจจะอพยพจากประชากรย่อยกลุ่มหนึ่งไปยังกลุ่มใดก็ได้ สำหรับประชากรกลุ่มย่อย กลุ่มผู้อพยพเข้าก็คือมาจากประชากรย่อยกลุ่มอื่น แล้วจึงนำวิธีการเลือกผู้อพยพมาเลือกผู้ที่จะอพยพออก



ภาพ 5 การอพยพแบบ Unrestricted migration

5. การเลือกผู้อพยพ (Migrant selection) การเลือกผู้อพยพมีวิธีการที่เป็นไปได้ 3 วิธี

คือ

- 5.1 เลือกผู้อพยพด้วยวิธีการสุ่ม
- 5.2 เลือกผู้อพยพที่มีค่าความเหมาะสมสูงที่สุดตามลำดับ
- 5.3 เลือกผู้อพยพด้วยวิธี roulette wheel

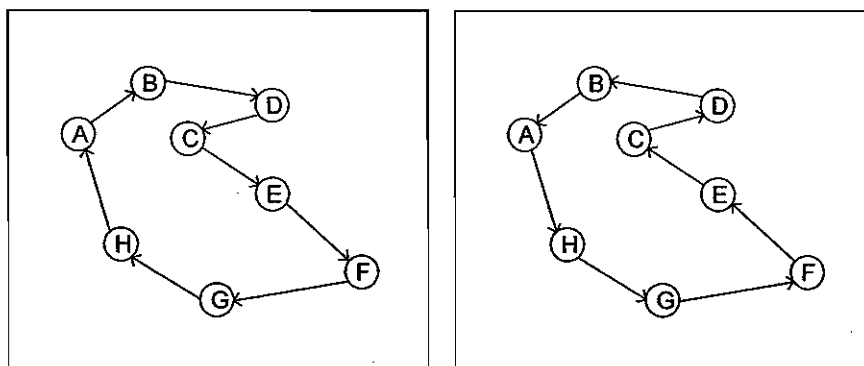
6. กลุ่มประชากรย่อยปลายทาง (Migration Destination)

การเลือกประชากรกลุ่มย่อยที่เป็นปลายทางของการอพยพแต่ละครั้ง ขึ้นอยู่กับสถาปัตยกรรมที่เราเลือกใช้ ซึ่งมีอิทธิพลต่อประสิทธิภาพของระบบ

ปัญหาการเดินทางของเซลส์แมน (Traveling Salesman Problem : TSP)

ปัญหาการเดินทางของเซลส์แมน (Traveling Salesman Problem หรือ TSP) “เซลส์แมนต้องเดินทางผ่านเมืองที่กำหนดไว้จำนวน n เมือง ให้ครบทุกเมือง โดยเซลส์แมนจะเดินทางผ่านเมืองแต่ละเมืองได้เพียงครั้งเดียวเท่านั้นและสุดท้ายจะต้องกลับมายังเมืองเริ่มต้นที่เริ่มออกเดินทาง” ซึ่งรูปแบบหรือลำดับของการเดินทางนั้นมีได้หลากหลายรูปแบบ และรูปแบบจะยิ่งมากขึ้นอย่างมากหาก n มีค่ามากขึ้น ปัญหาคือรูปแบบการเดินทางรูปแบบใดเป็นรูปแบบการเดินทางที่เหมาะสมที่สุด (optimum) นั่นคือใช้ระยะทางสั้นที่สุด ซึ่งหมายถึงการใช้เวลาน้อยที่สุด และทำให้ประหยัดค่าใช้จ่าย (A.J.Hoffman and P.Wolfe 1985)

ปัญหาการเดินทางของเซลส์แมนเป็นปัญหาหนึ่งซึ่งท้าทายนักคณิตศาสตร์และนักวิทยาศาสตร์สาขาคอมพิวเตอร์เป็นอย่างมาก เพราะว่าเป็นปัญหาที่ง่ายต่อการอธิบาย แต่ยากต่อการหาค่าที่เหมาะสมที่สุด (Optimum) ซึ่งสำหรับปัญหา TSP แบบ n โหนด ปัญหา TSP สามารถแบ่งได้เป็น 2 แบบคือแบบ สมมาตร (Symmetrical) กับแบบไม่สมมาตร (Asymmetrical) ปัญหา TSP โดยทั่วไปเป็นแบบสมมาตร นั่นคือสำหรับเมืองสองเมืองใด ๆ เช่น เมือง A กับเมือง B ถ้าระยะทางจากเมือง A ไปถึงเมือง B เท่ากับระยะทางจากเมือง B ไปถึงเมือง A จะเรียกว่าเป็นปัญหา TSP แบบสมมาตร และในทางตรงข้าม ถ้าหากว่าระยะทางจากเมือง A ไปถึงเมือง B ไม่เท่ากับระยะทางจากเมือง B ไปถึงเมือง A จะเรียกว่าเป็นปัญหา TSP แบบไม่สมมาตร ดังนั้น สำหรับจำนวนเมือง n เมือง ระยะทางรวมของการเดินทางที่เป็นไปได้ทั้งหมดจะเท่ากับ $(n-1)!$ แบบ สำหรับปัญหา TSP แบบไม่สมมาตร



(ก) การเดินทาง ABCDEFGHA

(ข) การเดินทาง AHGFECDBA

ภาพ 6 แสดงการเดินทางผ่านเมืองต่าง ๆ ของเซลล์แมน

จากภาพ 6 จะเห็นว่าระยะทางที่เซลล์แมนเดินทางในทิศทางย้อนกลับจะเท่ากัน สำหรับ ปัญหา TSP แบบสมมาตร ดังนั้น ระยะทางของการเดินทางของเซลล์แมนที่เป็นไปได้ทั้งหมดจะ เท่ากับ $(n-1)!/2$ แบบ

งานวิจัยที่เกี่ยวข้อง

Theodore C Belding (1995) ได้เสนอ Island Model โดยการทำงานของ Model นี้ เหมือนกับหมู่เกาะของ Island โดยเรียกว่า distributed Ga's เป็นการใช้ distributed Ga's เป็นการเชื่อมความสัมพันธ์ของค่าความเหมาะสม

David Patrick, Peter Green and Trevor York (1997) เป็นการนำ A Distributed genetic algorithm มาใช้กับ unix work stations ในการใช้ TCP/IP client-server ในการสร้าง multithread multiprocessor genetic algorithm เพื่อการอพยพข้อมูล โดยการส่ง multiple workstations ผ่าน local area network ไปยังอีก local area network หนึ่งให้รวดเร็วมากยิ่งขึ้น โดยสามารถส่งผ่านข้อมูลได้เร็วขึ้น 95% โดยผ่าน 16 สถานี

Matthew T McMahon (1998) A Distributed genetic algorithm with migration for the design of composite laminate structures เป็นการนำ Simple genetic algorithm และ Distributed genetic algorithm มาทำการเปรียบเทียบผ่าน Composite laminate structures ว่า อัลกอริทึมตัวใดใช้เวลาในการประมวลผลน้อย เพิ่มประสิทธิภาพการทำงาน และความน่าเชื่อถือ

Alessandro Bevilacqua (2005) ประยุกต์ใช้ Distributed genetic algorithm กับการ motion detection system เพื่อปรับปรุงเพิ่มประสิทธิภาพในการตรวจสอบความเคลื่อนไหวจาก Parameters ทั้งหมดที่มี โดยใช้เวลาในการประมวลผลน้อยลงและมีประสิทธิภาพที่เพิ่มขึ้น

Wayne Pullan (2003) ได้เสนอการประยุกต์นำจีเนติกอัลกอริทึมมารวมกับการ local optimization heuristics เพื่อใช้กับปัญหาการเดินทางของเซลล์แมน และแสดงได้ว่าเป็นวิธีที่มีประสิทธิภาพมากในการหาผลเฉลยที่ใกล้เคียงค่าเหมาะที่สุด โดย local optimization heuristics ช่วยในการลดขนาดของโดเมนของปัญหาลง ในขณะที่จีเนติกอัลกอริทึมช่วยจัดการเกิดเส้นทางที่ไม่เหมาะสมลงและยังช่วยกำเนิดกลุ่มของเส้นทางที่เหมาะสมรองอีกด้วย ซึ่งให้ผลลัพธ์ที่ดีเมื่อนำไปเปรียบเทียบกับปัญหา TSP ตามแบบเกณฑ์มาตรฐานต่าง ๆ

งานวิจัยนี้ยังเสนอแนวทางการพัฒนาต่อไปในอนาคตอีกหลายประการ เช่น ประการแรก การปรับปรุงอัลกอริทึมของ local optimization (2-OPT) โดยรวมกับแบบ 3-OPT หรือ Lin-Kernighan heuristic ประการที่สอง การปรับปรุงการ crossover รวมถึงการปรับ variants และการประมวลผลแบบขนานมาใช้กับจีเนติกอัลกอริทึม

Ekkasit Tiamkaew, Sushil J. Louis, and Angkul Kongmunvattana (2002) กระทำการประเมินสมรรถนะเชิงเปรียบเทียบของจีเนติกอัลกอริทึมแบบขนานภายใต้การโปรแกรมแบบขนานที่เป็นที่นิยม 2 แบบด้วยกัน คือ แบบ message passing กับแบบ shared memory การประเมินสมรรถนะดังกล่าวนี้กระทำบนกลุ่มของเครื่องคอมพิวเตอร์ที่หาซื้อได้ทั่วไปที่นำมาต่อเป็นโครงข่าย โดยมีจุดประสงค์เพื่อปรับแก้ความเข้าใจผิด ๆ ที่ว่า การโปรแกรมภายใต้โมเดลแบบ shared memory นั้นมักจะนำไปสู่โทษเชิงสมรรถนะเป็นสำคัญ เมื่อเปรียบเทียบกับกรณีแบบ message passing ของโปรแกรมแบบเดียวกัน โดยใช้โบลีของการอินเตอร์เฟสแบบ message passing แบบมาตรฐานสำหรับการทำ message passing และใช้โปรแกรมที่เป็นที่รู้จักกันดี คือ TradeMarks สำหรับกระจายการทำงานไปยังระบบที่มีการ share memory สำหรับการ share memory ตลอดการประเมินนั้นได้คำนึงถึงพารามิเตอร์ของ GA หลายตัว เช่นขนาดประชากร และความซับซ้อนของฟิตเนสฟังก์ชัน และสังเกตผลกระทบต่อสมรรถนะนั้น ในการประเมินเชิงเปรียบเทียบนั้นยังได้ประเมินสภาพขนาดปัญหาของ GA แบบขนานแต่ละแบบ โดยติดตามแนวโน้มเชิงสมรรถนะของทั้งสองแบบเมื่อแปรเปลี่ยนขนาดปัญหาของระบบไป จากผลทดลองแสดงให้เห็นว่าแบบ software share-memory ไม่เพียงแต่เฉพาะลดความซับซ้อนของการโปรแกรมแบบขนานลงเท่านั้น แต่ยังส่งผลให้สมรรถนะดีขึ้นอีกด้วย (คือภายใน 3% ของแบบ message passing)

Nuno Neves, Anthony-Trung Nguyen, and Edgar L. Torres (1996) ใช้จินตนาการอัลกอริทึมแบบขนาน กับปัญหาการหาค่าเหมาะที่สุดแบบไม่เชิงเส้นขนาดใหญ่มาก (กรณีนี้มี search space มาก) โดยได้พัฒนาโปรแกรมขึ้นมาใหม่โดยพัฒนามาต่อมาจาก GENESIS 5.0 โดยใช้ PVM 3.3 ซึ่งโปรแกรมใหม่มีชื่อว่า VMGENESIS และได้ใช้ศึกษาปัญหากำลังสองน้อยสุดแบบไม่เชิงเส้น (nonlinear least-square problem) บนโครงข่ายคอมพิวเตอร์ และเสนอวิธีการอพยพขึ้นมาสองแบบคือ แบบวงแหวนตรรก (logical ring) และแบบ multicast และใช้ความถี่ของการอพยพที่แตกต่างกัน 3 ความถี่ คืออพยพทุก ๆ 10, 100, และ 1000 เจนเนอเรชัน ซึ่งผลลัพธ์เชิงสมรรถนะแสดงให้เห็นว่า speedups จะมีความเป็นเชิงเส้นถ้าหาก DGA แบบพื้นฐานถูกเอามารวมเข้ากับ a simple dynamic load-balancing mechanism ผลการทดลองยังแสดงให้เห็นอีกว่าคุณภาพของการค้นหาคำตอบนั้นเปลี่ยนอย่างมีนัยยะสำคัญกับจำนวนของหน่วยประมวลผลที่เกี่ยวข้องกับการคำนวณ และกับความถี่ของการติดต่อสื่อสาร (อพยพ) และสรุปว่าจะได้ผลลัพธ์เชิงสมรรถนะดีที่สุดเมื่อ การอพยพเป็นแบบวงแหวนตรรก ความถี่ของการอพยพถูกกำหนดให้เป็นความถี่กลาง โดยการทดลองได้ใช้เครื่อง 16 เครื่องบนโครงข่ายสำหรับการทดลอง

Tomoyuki Hiroyasu, Mitsunori Miki and Masami Negami (1999) เสนอและศึกษาถึงผลกระทบของการอพยพของประชากรด้วยอัตราการอพยพเป็นแบบสุ่ม กับ DGA หรือเรียกว่า DGA/rmr (randomized migration rate) โดยนำ DGA/rmr นี้มาใช้จำลองเชิงตัวเลขกับฟังก์ชันทดสอบที่เป็นมาตรฐานสองฟังก์ชัน คือ Rastrigin function และ Rosenbrock function ซึ่งสามารถสรุปผลได้ดังนี้

1. ใน DGA เมื่อขนาดของประชากรมีขนาดใหญ่ ผลลัพธ์จะไม่ขึ้นอยู่กับการอพยพและอัตราการอพยพ
2. เมื่อขนาดประชากรเล็ก ผลลัพธ์จะขึ้นอยู่กับการอพยพและอัตราการอพยพ ดังนั้นจะเป็นการยากที่จะหาช่วงระยะเวลาการอพยพและอัตราการอพยพที่เหมาะสมที่สุดสำหรับระบบนี้
3. ใน DGA/rmr จะให้ผลลัพธ์ที่ดีที่ทุก ๆ ช่วงระยะเวลาการอพยพสำหรับทั้ง Rastrigin function และ Rosenbrock function ซึ่งผลลัพธ์เหล่านี้จะทำให้เห็นว่า การที่จะให้ได้ผลลัพธ์ที่ดีนั้นไม่จำเป็นต้องกำหนดทั้งอัตราการอพยพและช่วงระยะเวลาการอพยพเอาไว้
4. ข้อเสียเดียวของ DGA/rmr ที่พบคือ จำนวนของประชากรย่อยในแต่ละเกาะจะเปลี่ยนไปทุกครั้งที่มีการอพยพเกิดขึ้น ซึ่งการเปลี่ยนแปลงจำนวนประชากรย่อยนี้จะทำให้สูญเสียเวลาไปในการซิงโครไนซ์สำหรับในอัลกอริทึมนี้ ซึ่งกำลังวิจัยเพื่อพยายามแก้ปัญหาข้อนี้

James P. Cohoon and Shailesh U. Hegde and Worthy N. Martin and Dana S. Richards ทำการวิจัยเกี่ยวกับการนำ DGA มาแก้ปัญหาเกี่ยวกับ floor plan design ในการออกแบบ VLSI

Tomoyuki Hiroyasu, Mitsunori Miki and Sinya Watanabe (1999) เสนอ DGA แบบใหม่ใช้ในการแก้ปัญหา multiobjective optimization problem โดยนำ island model มาใช้กับ DGA และนำ Pareto-optimum solution มาแบ่งกันใช้ (sharing) ในกลุ่มประชากรทั้งหมด วิธีการนี้ทำให้มีความแม่นยำสูงและมีความหลากหลายของโครโมโซมมาก ผลกระทบที่เกิดจากการแบ่งประชากรเป็นกลุ่มย่อย ทำให้ความแม่นยำสูง ผลกระทบที่เกิดจากการแบ่งกันใช้ (sharing effect) นำไปสู่ความหลากหลายของโครโมโซมสูง เราตรวจสอบผลกระทบเหล่านี้โดยทำการทดลองทางตัวเลข ซึ่งมีฟังก์ชันวัตถุประสงค์ (objective function) มากกว่า 3 ฟังก์ชัน

David Patrick, Peter Green and Trevor York (1997) การพัฒนา CNIX multiprocessor environment ซึ่งออกแบบมาให้ใช้ในการประมวลผล DGA บน clusters ของ UNIX workstation. CNIX อิงบนพื้นฐานของแบบจำลอง TCP/IP client-server model และมีฟังก์ชันไลบรารี C++ สอง ฟังก์ชัน และทำการทดลองสามชุดด้วยกัน โดยนำ DGA มารันบนเครื่องคอมพิวเตอร์ตั้งแต่ 1 – 16 เครื่อง แต่ละเครื่องมีระบบปฏิบัติการไม่เหมือนกันเพียงแต่ต้องการ compiler ของ C++ แบบมาตรฐานเท่านั้น และมีวิธีรับส่งโครโมโซมแตกต่างกัน พบว่า speedup ที่ได้โดยเฉลี่ยอยู่ระหว่าง 85%-90% และมี speedup สูงที่สุด 93%

Shyh-Chang Lin, W.F. Punch III and E.D. Goodman (1994) ศึกษาเกี่ยวกับ premature convergence โดยเลือกเอาปัญหา graph partitioning problem มาศึกษา โดยทำการทดลองเปรียบเทียบ DGA และ SGA การทดลองนำ SGA มาแก้ปัญหา The Two 3-CCCs Partition Problem ด้วยวิธีการคัดสรรต่างกัน 5 วิธี และทำการทดลองเปรียบเทียบ DGA ทั้งหมด 8 แบบ โดยปรับให้แต่ละแบบมีจำนวนเกาะเป็น 1 5 10 และ 25 เกาะ โดยขนาดของกลุ่มประชากรย่อยหาได้จากนำประชากรทั้งหมด (5000) หารด้วยจำนวนเกาะ แล้ววัดค่าความเหมาะสมที่ดีที่สุดเฉลี่ย (average best fitness) มาเปรียบเทียบกัน และเสนอวิธีการ DGA แบบใหม่เรียกว่า the Injection Island GA (iiGA)

Wayne F. Boyer and Gurdeep S. Hura (2005) ปัญหาที่นำมาศึกษาคือ การหา optimal matching and scheduling of interdependent tasks ในสภาพแวดล้อมแบบ distributed heterogeneous computing (DHC) environments เป็นปัญหาที่เรียกว่า NP-hard problem ในระบบ DHC เวลาในการประมวลผลขึ้นกับเครื่องที่นำมาใช้งาน งานวิจัยชิ้นนี้เสนอ

อัลกอริทึมที่นำมาใช้กับ GA แบบใหม่คือ non-evolutionary random scheduling (RS) algorithm สำหรับการจับคู่และจัดตารางหน้าที่ในระบบ DHC และทำการทดลองเปรียบเทียบกับ GA และ PSGA เดิมที่มีอยู่ โดยทดลองเกี่ยวกับจำนวนเครื่องที่มีผลต่อประสิทธิภาพของอัลกอริทึมที่เสนอ เปรียบเทียบกับแบบเก่า และผลกระทบจากจำนวนเครื่องที่มีต่อเวลาในการประมวลผล นอกจากนี้ยังทำการทดลองเกี่ยวกับการเปลี่ยนขนาดของอินพุตที่นำมาทดสอบในขณะที่จำนวนเครื่องคงที่ ที่ 8 เครื่อง

Mir M. Atiqullah (2002) การวิเคราะห์สมรรถนะของ DGA เมื่อมาประยุกต์ใช้กับปัญหา design optimization problem 3 ปัญหา โดยต้องการศึกษาว่าจำนวนของตัวประมวลผล (processor) มีผลต่อสมรรถนะของระบบอย่างไร การวัดสมรรถนะของระบบทำได้โดยวัด parallel time

วัด speedup และวัดประสิทธิภาพ (efficiency) ผลการทดลองที่ได้ชี้ให้เห็นว่าสมรรถนะของระบบ ขึ้นอยู่กับขนาดของประชากร สถาปัตยกรรมของเครื่อง และจำนวนเครื่องที่นำมาประมวลผล

M. Rebaudengo and M. Sonza Reorda (1992) ผลกระทบของการแปรค่าพารามิเตอร์เกี่ยวกับการอพยพของ DGA โดยเลือกปัญหาการเดินทางของเซลล์แมนมาทำการทดลอง การทดลองเกี่ยวกับการอพยพทำโดยปรับพารามิเตอร์สองตัว ได้แก่ ปรับความถี่ของการอพยพ และปรับขนาดของผู้อพยพ และการทดลองเปลี่ยนวิธีการเลือกผู้อพยพ จากการทดลองปรับความถี่ของการอพยพ พบว่าเมื่อความถี่ลดลง ค่าความเหมาะสมเฉลี่ยเมื่อเทียบกับค่าความเหมาะสมที่ดีที่สุด ดีขึ้น และเมื่อทดลองปรับขนาดของผู้อพยพพบว่า เมื่อผู้อพยพมีจำนวนลดลงแล้วทำให้สมรรถภาพของระบบดีขึ้น สำหรับการทดลองเปลี่ยนวิธีการเลือกผู้อพยพ ได้ทำการทดลองเปลี่ยนวิธีการเลือกผู้อพยพสามวิธีด้วยกัน คือ ครั้งแรก เลือกผู้อพยพโดยวิธีสุ่ม ครั้งที่สองเลือกโครโมโซมที่มีค่าความเหมาะสมสูงที่สุดเป็นผู้อพยพ และวิธีที่สามเลือกผู้อพยพด้วยวิธี roulette wheel selection ผลการทดลองที่ได้ พบว่า วิธีการที่ดีที่สุดคือการเลือกผู้อพยพโดยวิธีสุ่ม นอกจากนี้ ยังทำการทดลองปรับจำนวนโปรเซสเซอร์ของ DGA พบว่า จำนวนโปรเซสเซอร์มากขึ้นกลับทำให้ความเร็วในการประมวลผลช้าลง ซึ่งสวนทางกับกรณีอุดมคติ

Wen-Yang Lin, Tzung-Pei Hong and Shu-Min Liu (2004) ศึกษาพารามิเตอร์เกี่ยวกับการอพยพของ DGA ที่ส่งผลกระทบต่อสมรรถภาพของ DGA โดยเสนอการปรับ migration interval และปรับอัตราการอพยพ เพื่อให้สมรรถนะของระบบดีขึ้น แล้วทดลองกับปัญหา 0/1 knapsack โดยนำสมรรถนะของ DGA ที่ปรับ migration interval และปรับอัตราการ

๘ QA
๗๖.๔
.A43
๐๖๕๓
๒๕๕๐

20 พ.ย. 2550



๖๖๗๘๖๒๙๗

อพยพมาเปรียบเทียบกับแบบที่ migration interval และอัตราการอพยพคงที่ การวัดสมรรถนะของระบบวัดจากค่าความเหมาะสมที่ดีที่สุด และวัดความหลากหลายของประชากร (population diversity) พบว่าการปรับ migration interval และปรับอัตราการอพยพ ได้ผลที่ทัดเทียมกับแบบที่ migration interval และอัตราการอพยพคงที่

