

# Various kinds of genetic operators

## 1. Selection

Selection คือการคัดเลือก string ที่เหมาะสมเพื่อสร้างประชากรในรุ่นถัดไป โดยพิจารณาจาก ค่า fitness ของ string ไต ๆ ซึ่งสำหรับ string ที่มีค่า fitness มาก ก็จะมีโอกาสอยู่รอดในรุ่นถัดไปมากขึ้น

วิธีการแบบดั้งเดิมที่ใช้ในการเลือกประชากรในรุ่นถัดไปก็คือ “Roulette wheel” selection ซึ่งขั้นตอนในการเลือกนั้น เริ่มจากแบ่งวงกลมตามสัดส่วนของค่า fitness ของแต่ละ string ซึ่ง string ที่มีค่า fitness มาก ก็จะได้พื้นที่ใน roulette wheel มากไปด้วย จากนั้นจึงเลือกประชากรโดยการ วนซ้ำตามจำนวนประชากร เพื่อเลือกตำแหน่งใน roulette wheel แบบสุ่ม จะได้ว่า string ที่มีพื้นที่ใน roulette wheel มากก็จะมีโอกาสถูกเลือกมากไปด้วย

วิธีการเลือกแบบดั้งเดิมนั้นมีปัญหอยู่สองอย่างคือ อย่างแรก ในตอนเริ่มต้นของวิวัฒนาการนั้น ประชากรทั้งหลายต่างมีค่า fitness ต่ำ ดังนั้น ถ้าหากว่ามี string อยู่ตัวหนึ่งที่มีค่า fitness มากกว่าตัวอื่น ๆ จะทำให้ประชากรรุ่นถัดไป ลู่เข้าสู่ string ตัวนั้นเร็วมาก ทำให้เกิดปัญหา local maxima นั่นคือทำให้วิวัฒนาการนั้นไม่สามารถก้าวกระโดดไปยังคำตอบที่ดีกว่าได้ ปัญหาย่างที่สองคือ เนื่องจากการวิวัฒนาการได้ลู่เข้าสู่คำตอบคำตอบหนึ่งเร็วเกินไป ทำให้ประชากรในรุ่นหลัง ๆ มี string ที่รูปร่างคล้าย ๆ กัน ดังนั้น การ crossover หรือ mutation จึงไม่อาจสร้างประชากรรุ่นถัดไปที่มีความหลากหลายได้ ดังนั้นจึงได้มีการคิดค้นวิธีการคัดเลือกใหม่ ๆ เพื่อจะแก้ปัญหาสองอย่างนี้

## Scaling mechanisms

การ Scale ค่า fitness คือการพยายามปรับปรุงค่า fitness ของ string เพื่อให้มีความเหมาะสมในการคัดเลือกมากขึ้น ซึ่งการ Scale นั้นทำได้โดยคำนวณ Linear scaling ตามสมการ

$$f' = af + b$$

เมื่อ  $f$  เป็นค่า fitness,  $f'$  เป็นค่า fitness ที่ scale แล้ว ส่วน  $a$  และ  $b$  เป็นค่าคงที่ที่เหมาะสม จะเห็นได้ว่าในบางรุ่นของประชากร ที่ string บางตัวมีค่า fitness สูงกว่า ตัวอื่นมาก ๆ เราก็สามารถเลือกค่า  $a$  และ  $b$  ที่เหมาะสมเพื่อ scale และ shift ค่า fitness เพื่อให้ ค่า fitness มากสุดไม่เกิน 1.5 หรือ 2 เท่า ของค่า fitness เฉลี่ยได้ วิธีนี้จะทำให้ string ที่มีค่า fitness น้อย มีโอกาสถูกเลือกมากขึ้น แต่วิธีการ scale นี้ก็มีโอกาสทำให้เกิดค่า fitness ใหม่ที่มีค่าติดลบได้ ดังนั้นจึงต้องเลือกค่า  $a$  และ  $b$  อย่างเหมาะสม หรืออาจจะแก้ปัญหาโดยการตัด string ที่มีค่า fitness ติดลบ ออกไปจากประชากรก็ได้

การทำ Linear scaling ค่า fitness นั้น ปัญหาหลักก็คือ ไม่รู้จะเลือกค่า  $a$  และ  $b$  เป็นเท่าไรดี ดังนั้นจึงมีวิธี scale อีกวิธีหนึ่งเรียกว่า Sigma Scaling วิธี Sigma Scaling จะทำการพิจารณา standard deviation ( $\sigma$ ) ของค่า fitness ทั้งหมดเสียก่อนจากนั้นจึงหาค่า fitness ด้วยสมการ

$$f' = f - (\bar{f} - c\sigma)$$

ซึ่ง  $\bar{f}$  เป็นค่าเฉลี่ยของค่า fitness ทั้งหมด,  $\sigma$  เป็น standard deviation ของค่า fitness ของประชากรทั้งหมด ส่วน  $c$  เป็นค่าคงที่มีค่าได้ตั้งแต่ 1 ถึง 3

จากสมการข้างต้นนี้เป็นการ shift ค่า fitness เพื่อให้อัตราส่วนระหว่าง ค่า fitness มากสุด และ ค่า fitness เฉลี่ย มีความเหมาะสม ส่วน ค่า fitness ที่มีค่าน้อยกว่า  $c\sigma$  ก็จะถูกตัดทิ้งไป วิธีนี้จะเป็นการช่วยรับประกันได้ว่า ค่า fitness ที่อยู่ในช่วง  $\pm c$  standard deviations จะถูกพิจารณา

### Rank-based selection

อีกวิธีหนึ่งที่ใช้ในการแก้ปัญหาการคัดเลือกคือ Rank-based selection ซึ่งเป็นการควบคุมการเลือก string โดยไม่สนใจค่า fitness ของ string ว่ามีค่าเท่าไร แต่จะสนใจเพียงแค่ลำดับของ string โดย string ในประชากรทั้งหมดจะถูกนำมาจัดเรียงลำดับจากนั้นจึงกำหนดความน่าจะเป็นในการถูกเลือกให้ใหม่ ตามลำดับของ string

วิธีนี้เป็นการเปิดโอกาสให้ string ที่มีค่า fitness มากแบบกระโดด ไม่ถูกเลือกมากเกินไป และทำให้ string ที่มีค่า fitness น้อยมาก มีโอกาสถูกเลือกบ้าง แต่วิธีนี้ก็มี overhead ในการ sort string ด้วยเช่นกัน

### Tournament selection

วิธี Tournament selection เป็นการคัดเลือกประชากรโดยเลียนแบบการแข่งขันแบบ Tournament วิธีการคัดเลือก จะเริ่มจากการสุ่มเลือก string มาสองตัว จากนั้นก็สุ่มตัวเลข  $r$  ตั้งแต่ 0 ถึง 1 ถ้า  $r$  น้อยกว่า  $m$  ก็เลือก string ที่ค่า fitness ดีกว่า แต่ถ้า  $r$  มากกว่า  $m$  ก็เลือก string ที่ค่า fitness ต่ำกว่า ซึ่งค่า  $m$  นั้นจะเป็นความน่าจะเป็นที่ string ที่ค่า fitness ดีกว่า จะชนะ string ที่ค่า fitness แยกว่า ซึ่งใน  $k$ -ary tournament นั้น  $k$  strings ที่ดีที่สุดจะถูกเลือกเพื่อเป็นประชากรในรอบถัดไป

ข้อดีของวิธีนี้คือ ไม่จำเป็นต้อง sort string แล้วยังสามารถคำนวณแบบ parallel ได้ง่ายอีกด้วย

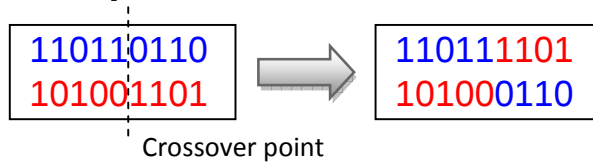
## 2. Recombination

Recombination คือการสร้าง string ตัวใหม่ จาก string ตัวเดิม ซึ่ง string ตัวใหม่ที่ถูกสร้างขึ้นนั้นจะถูกนำไปวิเคราะห์ค่า fitness ว่ามีความเหมาะสมจะเป็นประชากรรุ่นถัดไปมากแค่ไหน ซึ่ง string ตัวใหม่ ที่สร้างขึ้นมานั้นจะถูกนำไปคัดเลือก เพื่อเป็นประชากรรุ่นถัดไปต่อไป จะเห็นได้ว่ากระบวนการ Recombination นั้น มีความสำคัญในการกำหนด แนวทางการวิวัฒนาการของประชากร ซึ่งวิธีการปรับปรุงพันธุกรรมของ string นั้น ก็เหมือนกับการ ใส่ Bias ในการค้นหาคำตอบของปัญหา ดังนั้นการเลือกวิธี Recombination ที่เหมาะสมสำหรับปัญหานั้น ๆ จะทำให้การค้นหาคำตอบมีความรวดเร็วและเกิดประสิทธิภาพ

วิธี Recombination ใน GA นั้นมีอยู่สองแบบหลัก ๆ ด้วยกันคือ Crossover และ Mutation

## One-Point Crossover

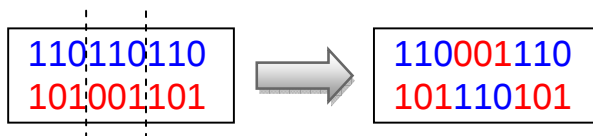
เป็นการสร้าง strings ตัวใหม่ จาก strings พ่อแม่จำนวนสองตัว วิธีการของ One-Point Crossover ก็คือการสุ่มเลือกตำแหน่ง ใน string พ่อและแม่มา 1 ตำแหน่ง จากนั้นจึงทำการสลับข้อมูลใน string ของพ่อและแม่ ตามตำแหน่งที่สุ่มเลือก เพื่อสร้าง string ลูกใหม่ สองตัว



ปัญหาหลักของ One-Point Crossover คือมี Bias ที่ตำแหน่งท้ายของ string เนื่องจาก ไม่ว่าจะเลือก Crossover Point ตำแหน่งไหน bit ท้ายสุดของ string จะถูก Crossover เสมอ

## Two-Point Crossover

จะสร้าง strings ตัวใหม่จาก strings พ่อแม่ โดยเลือกตำแหน่งใน string สำหรับ Crossover มาสองตำแหน่ง จากนั้นก็สลับข้อมูลในช่วงระหว่างกลาง



ซึ่งจะเห็นได้ว่า Two-Point Crossover สามารถแก้ปัญหา ends of strings bias ของ One-Point Crossover ได้

## Multipoint crossover

คล้ายกับ Two-Point Crossover แต่จะแบ่งส่วนในการ Crossover มากขึ้น

## Uniform Crossover

วิธีนี้ไม่จำเป็นต้องเลือก Crossover point เนื่องจากทุก bits มีโอกาสสลับกันได้หมด โดยมีความน่าจะเป็นในการสลับถูก fix ไว้ที่ค่าใดค่าหนึ่ง และการสลับในตำแหน่งหนึ่งนั้นจะไม่ขึ้นกับการสลับในตำแหน่งอื่น ๆ

วิธี Uniform crossover นั้นจะมี positional bias น้อยสุด เนื่อง ความน่าจะเป็นที่ bit จะถูกสลับไม่ขึ้นกับตำแหน่งของมัน แต่ว่า Uniform crossover จะมี distributional bias มากเนื่องจาก จำนวน bits ที่ถูกสลับในแต่ละครั้งไม่ uniform ซึ่งจะตรงกันข้ามกับแบบ One-Point Crossover ซึ่งมี positional bias มาก และ distributional bias น้อย

## Inversion

เป็นการปรับปรุงพันธุกรรมของ string โดยการเลือกกลับ ช่วงบางช่วงใน string ดังภาพ

