บทที่ 2

ทฤษฎีคิวอิงและฮิวริสติกอัลกอริทึม

ความนำ

ในบทนี้ได้กล่าวถึงทฤษฎีต่างๆที่ใช้ในการจำลองปัญหาการจัดเส้นทางในโครงข่ายสื่อสารที่ให้บริการแบบคอนเนกชันโอเรียนเตด โดยจะแบ่งกล่าวเป็นหัวข้อย่อยๆดังนี้ ส่วนแรกจะกล่าวถึงทฤษฎีคิวอิง (Kleinrock, 1975) ที่จะนำมาจำลองปัญหาในการคำนวณความน่าจะเป็นที่แพ็กเกตของข้อมูลจะเกิดการสูญหายเนื่องจากเกิดการบล็อกขึ้นที่บัฟเฟอร์ด้านขาออกของสวิตช์ โดยได้กล่าวถึงแบบจำลอง M/M/1/K ที่จะนำมาเป็นแบบจำลองของโนดภายในโครงข่าย ส่วนต่อมาจะกล่าวถึงอัลกอริทึมต่างๆที่ใช้ในการค้นหาคำตอบที่เหมาะที่สุดได้แก่ ยีนส์นิติกอัลกอริทึม, ทาบูเซอร์ช และอีโวลูชันนารีคอมพิวติง หรืออีโวลูชันนารีสตราทีจี ซึ่งถูกนำไปดัดแปลงและปรับปรุงให้ได้เป็นฮิวริสติกอัลกอริทึม ในส่วนสุดท้ายจะกล่าวถึงฮิวริสติกอัลกอริทึมโดยละเอียด

ทฤษฎีคิวอิง (Kleinrock, 1975 และ Hayes, 1987)

การศึกษาทฤษฎีคิวอิงนั้นจะเป็นพื้นฐานในการวิเคราะห์และออกแบบให้โครงข่ายสื่อสารโทรคมนาคมมีประสิทธิภาพสูงขึ้น (Hayes, 1987) การจัดระเบียบคิวอิงไม่เพียงจะเกิดขึ้นในสวิตช์ภายในโครงข่ายสื่อสารแต่ยังเกิดขึ้นในระบบของการคำนวณด้วยโปรเซสเซอร์หลายๆตัว และระบบที่มีหน่วยความจำหลายๆชุด นอกจากนี้แล้วทฤษฎีคิวอิงยังสามารถที่จะนำไปประยุกต์ใช้ในระบบท่อส่งน้ำมัน โรงงานผลิตเรือ และ ระบบการจราจร เป็นต้น ในโครงข่ายข้อมูลพื้นที่กว้าง(wide-area data network) นั้น แพ็กเกตที่เข้ามายังโนดจะถูกจัดเข้าแถวรอการจัดส่งไปยังเส้นทางที่ต้องการ หากเส้นทางที่จะส่งผ่านเกิดความคับคั่งหรือโนดกำลังดำเนินการใดๆอยู่ ระยะเวลาในการดำเนินการของโนด เช่น เวลาที่ใช้ในการอ่านส่วนหัว(header) ของแพ็กเกต การตรวจ

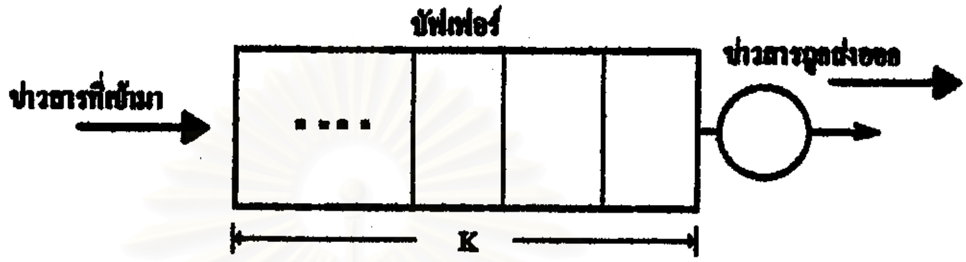
สอบความผิดพลาด และการอ่านข่าวสารในการจัดเส้นทาง เป็นต้น จะเป็นเวลาที่ต้องนำมาพิจารณาเพื่อจะสามารถระบุถึงอัตราการให้บริการ(service rate) ของโนดได้ (Hayes, 1987)

ในการพิจารณาถึงระยะเวลาในการรอคอยการให้บริการแก่แพ็กเกตในคิวจะขึ้นกับกระบวนการเข้ามาของแพ็กเกต และความจุของช่องสัญญาณหรืออัตราที่ให้บริการได้ ในระบบคิวอิงจะสามารถที่จะเขียนแบบจำลองระบบแทนด้วยการระบุชนิดของกระบวนการเข้ามา กระบวนการให้บริการ จำนวนผู้ให้บริการ(server) และขนาดของบัฟเฟอร์ในระบบคิวอิงได้ด้วยกลุ่มของตัวอักษรและตัวเลข (Hayes, 1987) เช่น M / G / 2 / 5 โดย M ตัวแรกจะหมายถึงกระบวนการเข้ามาของแพ็กเกตเป็นแบบสุ่มโดยมีสถิติเป็นการกระจายแบบปัวส์ซง(poisson distribution) ตัวอักษร G ตัวที่สองจะหมายถึงการให้บริการที่มีการกระจายทางสถิติของเวลาเป็นกรณีทั่วไป (general service time distribution) เลข 2 คือจำนวนของผู้ให้บริการในระบบคิวอิง และ ขนาดของบัฟเฟอร์จะเก็บแพ็กเกตได้ 5 แพ็กเกต เป็นต้น

กรณีของคิวอิงระบบ M/M/1/K นั้น จะได้ว่าตัวอักษร M ตัวแรกหมายถึงกระบวนการเข้ามาของแพ็กเกตเป็นแบบสุ่มที่มีการกระจายแบบปัวส์ซง ตัวอักษร M ตัวที่สองจะหมายถึงการให้บริการที่มีการกระจายเวลาในการให้บริการเป็นแบบเอ็กโปเนนเชียล(exponential service time distribution) และใช้จำนวนของผู้ให้บริการเพียงหนึ่งตัว (single server) โดยในที่นี้จะมีขนาดของบัฟเฟอร์เป็น K แพ็กเกต ซึ่งสามารถแสดงระบบได้ในรูปที่ 2.1 จากระบบดังรูปจะสามารถที่จะอธิบายระบบได้ด้วยกระบวนการเกิด-ตาย (birth-death process) (Kleinrock, 1975) ซึ่งเป็นกระบวนการมาร์คอฟชนิดหนึ่งซึ่งมีสเตตการเปลี่ยนแปลงเพียงสองสเตตคือ สเตตการเกิดซึ่งทำให้ความยาวของคิวในบัฟเฟอร์ยาวขึ้นหนึ่งหน่วยแพ็กเกตเนื่องจากมีการเข้ามาของแพ็กเกต และสเตตการตายซึ่งทำให้ความยาวของคิวในบัฟเฟอร์ลดลงหนึ่งหน่วยแพ็กเกตเนื่องจากแพ็กเกตในบัฟเฟอร์ถูกให้บริการไป ในระบบคิวอิงระบบนี้จะมีการจัดระเบียบคิวเป็นชนิดเข้าก่อนได้รับบริการก่อน (first in first served : FIFS) และปราศจากสิทธิก่อน(Priority) ระบบสามารถแทนจำนวนแพ็กเกตในคิวได้ด้วยตัวแปรสุ่มที่เป็นค่าดีสครีต $n(t)$ หากอัตราเฉลี่ยการเข้ามาของแพ็กเกต เป็น λ หน่วยแพ็กเกตต่อวินาที และมีอัตราเฉลี่ยการให้บริการกับแพ็กเกตในบัฟเฟอร์เป็น μ หน่วยแพ็กเกตต่อวินาที ในช่วงเวลาที่เปลี่ยนแปลงไปสั้นๆ dt วินาที แล้วจะได้ว่า $n(t)$ จะแสดงได้ด้วยห่วงโซ่มาร์คอฟที่เวลาต่อเนื่อง(continuos time Markov chain) ซึ่งเขียนสมการความน่าจะเป็นที่สเตตใดๆที่เวลา $t = t + dt$ วินาทีแสดงได้ด้วย $P_n(t+dt)$ โดยเทียบกับที่เวลา t วินาทีได้ในสมการที่ 2.1

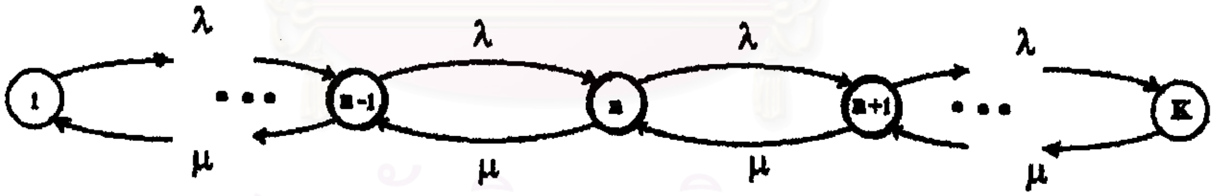
$P_n(t+dt) = P_n(t)$ [ความน่าจะเป็นที่ย้ายมาจากสเตต n ที่เวลา t ไปยังสเตต n ที่เวลา $t + dt$ วินาที]
+ $P_{n-1}(t)$ [ความน่าจะเป็นที่ย้ายมาจากสเตต $n\text{-}1$ ที่เวลา t ไปยังสเตต n ที่เวลา $t + dt$ วินาที]
+ $P_{n+1}(t)$[ความน่าจะเป็นที่ย้ายมาจากสเตต $n+1$ ที่เวลา t ไปยังสเตต n ที่เวลา $t + dt$ วินาที]

(2.1)



รูปที่ 2.1 ระบบคิวอิงที่มีขนาดบัฟเฟอร์เป็น K

ในสมการนี้จะทำการสมมุติให้ความน่าจะเป็นของการเข้ามา การให้บริการ และสเตตของบัฟเฟอร์เป็นอิสระต่อกัน จากสมการจะพบว่าค่าของ $P_n(t+dt)$ จะขึ้นกับ $P_n(t)$ $P_{n-1}(t)$ และ $P_{n+1}(t)$ เท่านั้น แต่จะไม่ขึ้นกับ $P_{n-2}(t)$ หรือ $P_{n+2}(t)$ และค่าความน่าจะเป็นที่สเตตอื่นๆ จากข้างต้นจะสามารถแสดงสเตตของห่วงโซ่มาร์คอฟได้ดังรูปที่ 2.2 (Hayes, 1987)



รูปที่ 2.2 แสดงสเตตไดอะแกรมของระบบคิวอิงที่มีบัฟเฟอร์ขนาด K

การกระจายแบบปัวส์ซงของการเข้ามาของแพ็กเกต (James, 1993) สามารถเขียนได้ดังสมการที่ 2.2

$$P(\text{มีแพ็กเกต k หน่วยที่เข้ามาในช่วงเวลา T วินาที}) = \frac{(\lambda T)^k e^{-\lambda T}}{k!} \tag{2.2}$$

โดยที่ค่า k = 0, 1, 2, ...

จำนวนเฉลี่ยของจำนวนแพ็กเกตที่เข้ามาในช่วงเวลา T วินาทีจะได้เท่ากับ λT หน่วยแพ็กเกตและจะได้ค่าความน่าจะเป็นที่จะมีแพ็กเกตหนึ่งแพ็กเกตเข้ามาในช่วงเวลา T วินาทีดังนี้

$$P(T) = \lambda T \cdot e^{-\lambda T} = (\lambda T)\left[1 + (-\lambda T) + (-\lambda T)^2 + (-\lambda T)^3 + \ldots\right]$$

หรือ

$$P(T) = (\lambda T)\left[1 - \lambda T + (\lambda T)^2 - (\lambda T)^3 + \ldots\right] \tag{2.3}$$

และจะได้ค่าเฉลี่ยของอัตราการเข้ามาของแพ็กเกตเป็น λ แพ็กเกตต่อวินาที

จากสมการที่ 2.3 ในช่วงเวลาแคบมากๆ dt วินาทีจะมีค่าความน่าจะเป็นที่จะมีแพ็กเกตเข้ามาหนึ่งหน่วยแพ็กเกตโดยประมาณดังนี้

$$\begin{aligned} P_1(dt) &= (\lambda \cdot dt)\left[1 - \lambda \cdot dt + \lambda^2 (dt)^2 - \lambda^3 (dt)^3 + \ldots\right] \cong (\lambda \cdot dt)(1 - \lambda \cdot dt) \\ &\cong \lambda \cdot dt - \lambda^2 dt^2 \cong \lambda \cdot dt \end{aligned} \tag{2.4}$$

โดยที่ค่า dt มีค่าเข้าใกล้ศูนย์

และจะได้ค่าประมาณของค่าความน่าจะเป็นที่จะไม่มีแพ็กเกตเข้ามาในช่วงเวลาสั้นๆ dt วินาที จากสมการที่ 2.2 จะสามารถเขียนได้ดังนี้

$$P_0(dt) = e^{-\lambda dt} = \left[1 + (-\lambda \cdot dt) + (-\lambda \cdot dt)^2 + (-\lambda \cdot dt)^3 + \ldots\right]$$

$$\cong 1 - \lambda dt \tag{2.5}$$

เวลาที่ใช้ในการให้บริการส่งแพ็กเกตหนึ่งแพ็กเกตออกไปซึ่งมีฟังก์ชันความหนาแน่นความน่าจะเป็นแบบเอ็กโปเนลเชียล (James, 1993) จะได้สมการดังนี้คือ

$$p(t) = \mu \cdot e^{-\mu t} \tag{2.6}$$

ซึ่งจะได้ค่าเฉลี่ยทางเวลาที่ใช้ในการให้บริการส่งแพ็กเกตหนึ่งแพ็กเกตออกไปเป็น $\frac{1}{\mu}$ วินาที หรืออัตราเฉลี่ยที่ใช้ส่งแพ็กเกตออกจากบัฟเฟอร์เป็น μ แพ็กเกตต่อวินาที

จากสมการที่ 2.6 จะสามารถหาค่าความน่าจะเป็นที่จะมีแพ็กเกตถูกส่งออกจากบัฟเฟอร์ในช่วงเวลาสั้นๆ *dt* วินาทีได้ดังนี้

$$P_s(dt) = \int_0^{dt} p(t)dt = \int_0^{dt} \mu \cdot e^{-\mu \cdot t} dt = 1 - e^{-\mu \cdot dt}$$

$$= 1 - \left(1 + \left(-\mu \cdot dt\right) + \left(-\mu \cdot dt\right)^2 + \left(-\mu \cdot dt\right)^3 + \ldots\right) \cong \mu \cdot dt \tag{2.7}$$

และค่าความน่าจะเป็นที่จะไม่มีแพ็กเกตถูกส่งออกจากบัฟเฟอร์ในช่วงเวลาสั้นๆ *dt* วินาทีคือ

$$P_{ns}(dt) = 1 - \mu \cdot dt \tag{2.8}$$

พิจารณาสมการที่ 2.4, 2.5, 2.7 และ 2.8 ร่วมกับรูปที่ 2.2 จะได้ว่า

1) ค่าความน่าจะเป็นที่ย้ายมาจากสเตต n ที่เวลา t ไปยังสเตต n ที่เวลา $t + dt$ วินาที = ค่าความน่าจะเป็นที่จะไม่มีแพ็กเกตที่เข้ามาเพิ่มและไม่มีแพ็กเกตถูกส่งออกไปจากบัฟเฟอร์ $= (1-\lambda \cdot dt)(1-\mu \cdot dt)$

2) ค่าความน่าจะเป็นที่ย้ายมาจากสเตต n-1 ที่เวลา t ไปยังสเตต n ที่เวลา $t + dt$ วินาที = ค่าความน่าจะเป็นที่จะมีแพ็กเกตที่เข้ามาเพิ่มและไม่มีแพ็กเกตถูกส่งออกไปจากบัฟเฟอร์ $= (\lambda \cdot dt)(1-\mu \cdot dt)$

3) ค่าความน่าจะเป็นที่ย้ายมาจากสเตต n+1 ที่เวลา t ไปยังสเตต n ที่เวลา $t + dt$ วินาที = ค่าความน่าจะเป็นที่จะไม่มีแพ็กเกตที่เข้ามาเพิ่มและมีแพ็กเกตถูกส่งออกไปจากบัฟเฟอร์ $= (1-\lambda \cdot dt)(\mu \cdot dt)$

ดังนั้นสมการที่ 2.1 จะสามารถเขียนใหม่ได้ดังนี้

$$\begin{aligned} &P_n(t+dt) = \\ &\quad P_n(t)(1-\lambda \cdot dt)(1-\mu \cdot dt) + P_{n-1}(t)(\lambda \cdot dt)(1-\mu \cdot dt) + P_{n+1}(t)(1-\lambda \cdot dt)(\mu \cdot dt) \end{aligned} \tag{2.9}$$

กำหนดให้ $dt \to 0$ สมมุติให้ค่าของค่าความน่าจะเป็น $P_n(t)$ มีค่าที่ต่อเนื่องในเชิงเวลา ซึ่งจะได้ว่าหากทำการแสดงสมการในรูปของอนุกรม Taylor (James, 1993) ของ $P_n(t+dt)$ โดยพิจารณาเฉพาะสองเทอมแรกซึ่งมีค่าที่เห็นเด่นชัดกว่าค่าที่ได้จากเทอมอื่นๆ จะสามารถทำการประมาณค่าของ $P_n(t+dt)$ ได้ใหม่ดังนี้

$$P_n(t+dt) \cong P_n(t) + \frac{dP_n(t)}{dt}(dt) \tag{2.10}$$

จากสมการที่ 2.9 และ 2.10 จะได้

$$\begin{aligned} &P_n(t) + \frac{dP_n(t)}{dt}(dt) = \\ &\quad P_n(t)(1-\lambda \cdot dt)(1-\mu \cdot dt) + P_{n-1}(t)(\lambda \cdot dt)(1-\mu \cdot dt) + P_{n+1}(t)(1-\lambda \cdot dt)(\mu \cdot dt) \end{aligned}$$

หรือ

$$P_n(t)(-\lambda\cdot dt-\mu\cdot dt+\lambda\cdot\mu\cdot dt^2)+P_{n-1}(t)(\lambda\cdot dt-\lambda\cdot\mu\cdot dt^2)+P_{n+1}(t)(\mu\cdot dt-\mu\cdot\lambda\cdot dt^2)$$
$$=\frac{dP_n(t)}{dt}(dt)$$

จะได้เป็น

$$-(\lambda+\mu)P_n(t)+\lambda\cdot P_{n-1}(t)+\mu\cdot P_{n+1}(t)\cong\frac{dP_n(t)}{dt} \tag{2.11}$$

หากพิจารณาค่าที่สถิติไม่แปรตามเวลา (Stationary Statistics) แล้ว ค่าความน่าจะเป็นจะไม่ขึ้นกับเวลา จะได้ว่า $\frac{dP_n(t)}{dt}=0$ จากสมการที่ 2.11 จะได้

$$\lambda\cdot P_{n-1}+\mu\cdot P_{n+1}=(\lambda+\mu)P_n \tag{2.12}$$

สมการที่ 2.12 เป็นสมการอนุรักษ์การไหล (flow conservation) และจากรูปที่ 2.2 ที่สเตตเริ่มแรก จะได้ว่า

$$\mu P_1=\lambda P_0 \tag{2.13}$$

จากสมการที่ 2.12 และ 2.13 จะสามารถหาค่าความน่าจะเป็นที่มีแพ็กเกตอยู่ในบัฟเฟอร์จำนวน n แพ็กเกตได้โดยกำหนดให้โหลด $\rho=\frac{\lambda}{\mu}$ และ $\rho\le 1$ ดังนั้นจะได้ว่า

$$P_{n+1}=(\rho+1)P_n-\rho\cdot P_{n-1} \tag{2.14}$$

$$P_1=\frac{\lambda}{\mu}P_0=\rho P_0$$

$$P_2=(\rho+1)P_1-\rho P_0=(\rho+1)\rho P_0-\rho P_0=\rho^2 P_0$$

$$P_3 = (\rho+1)P_2 - \rho P_1 = (\rho+1)\rho^2 P_0 - \rho P_0 = \rho^3 P_0$$

.

.

.

$$P_n = \rho^n P_0 \tag{2.15}$$

ระบบที่มีคิวอิงแบบ M/M/1/K นั้น ขนาดของบัฟเฟอร์จะมีค่าคงที่เท่ากับ K แพ็กเกต และหากแพ็กเกตที่เข้ามาหลังจากที่บัฟเฟอร์เต็มแล้ว แพ็กเกตนั้นจะถูกบล็อกทิ้งไป ดังนั้นสเตตที่เป็นไปได้ของระบบคิวอิงแบบนี้จะเป็นไปได้ตั้งแต่ 0 จนถึง K จากคุณสมบัติของค่าความน่าจะเป็นจะได้

$$\sum_{n=0}^{K} P_n = 1 \tag{2.16}$$

จากสมการที่ 2.15 จะได้สมการที่ 2.16 ใหม่เป็น

$$\sum_{n=0}^{K} \rho^n P_0 = \left[\frac{1-\rho^{K+1}}{1-\rho}\right] P_0 = 1$$

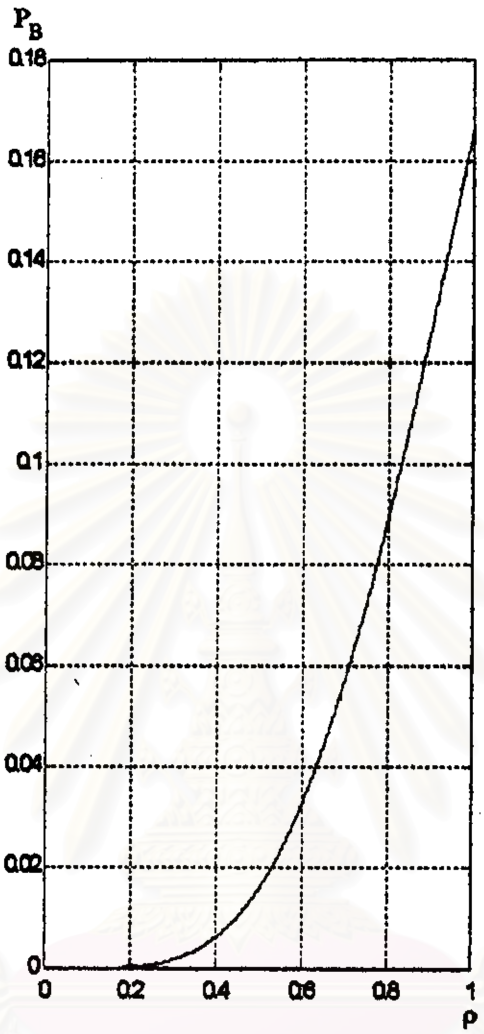
ซึ่งจะได้สมการใหม่เป็น

$$P_0 = \frac{1-\rho}{1-\rho^{K+1}} \qquad \text{และ} \qquad P_n = \frac{1-\rho}{1-\rho^{K+1}}\rho^n \tag{2.17}$$

จากสมการที่ 2.17 จะสามารถหาค่าความน่าจะเป็นที่แพ็กเกตที่เข้ามาจะเกิดการบล็อกขึ้น ซึ่งก็คือค่าความน่าจะเป็นที่แพ็กเกตในบัฟเฟอร์จะมีจำนวนเท่ากับขนาดของบัฟเฟอร์

$$P_B = \frac{(1-\rho)\rho^K}{1-\rho^{K+1}} \tag{2.18}$$

และจะได้กราฟที่แสดงความสัมพันธ์ระหว่างค่าความน่าจะเป็นที่จะเกิดการบล็อกขึ้นที่บัฟเฟอร์เทียบกับโหลด(อัตราส่วนของอัตราการเข้ามาต่ออัตราการให้บริการ) ได้ดังรูปที่ 2.3 เมื่อกำหนดให้ขนาดบัฟเฟอร์เป็น 5 แพ็กเกต

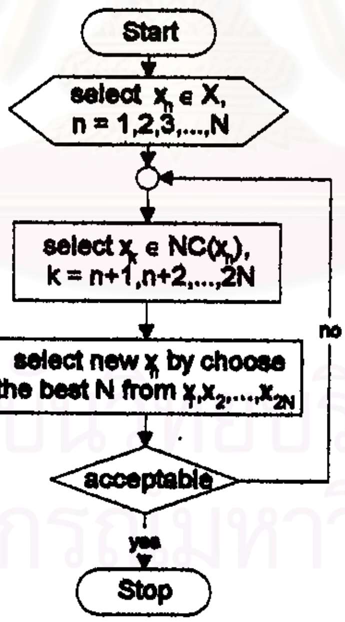


รูปที่ 2.3 แสดงความสัมพันธ์ระหว่างค่าความน่าจะเป็นที่จะเกิดการบล็อกขึ้นที่บัฟเฟอร์เทียบกับโหลดที่ขนาดต่างๆ

จากรูปกราฟและสมการที่ 2.18 จะพบว่า โอกาสที่แพ็กเกตข้อมูลจะเกิดบล็อกขึ้นที่บัฟเฟอร์ที่ด้านออกของสวิตช์แปรตามขนาดของอัตราการเข้ามาของแพ็กเกต

<u>ทฤษฎีอีโวลูชันนารีคอมพิวติง(Evolutionary Computing)</u> (Glover และ Laguna, 1993)

อีโวลูชันนารีคอมพิวติงหรืออีโวลูชันนารีสตราตีจี เป็นอัลกอริทึมที่ใช้ในการค้นหาคำตอบที่เหมาะที่สุดที่ถูกคิดขึ้นมาอัลกอริทึมแรก(Glover และ Laguna, 1993) อัลกอริทึมจะเริ่มจากการเลือกกลุ่มคำตอบที่เป็นไปได้มาจำนวน N คำตอบ จากนั้นทำการเลือกคำตอบที่ใกล้เคียงกับคำตอบในกลุ่มเดิมขึ้นมาใหม่อีก N คำตอบ ซึ่งได้คำตอบทั้งหมดเป็น 2N คำตอบ ทำการเลือกคำตอบที่เหมาะที่สุดจำนวน N คำตอบจากคำตอบในกลุ่มคำตอบ 2N คำตอบ จากนั้นทำการวนขั้นตอนซ้ำจนกว่าจะได้คำตอบที่เหมาะสมดังรูปไฟล์ชาร์ตรูปที่ 2.4 โดย x_n คือกลุ่มคำตอบในรอบการประมวลผลนั้นๆ เซต X เป็นเซตของคำตอบที่เป็นไปได้ทั้งหมด และ NC(x) คือเซตของคำตอบที่ใกล้เคียงกับคำตอบ x จากกระบวนการของอัลกอริทึมจะพบว่าการค้นหาคำตอบจะมุ่งเน้นไปทางการลู่เข้าหาคำตอบเท่านั้น ขณะที่ความหลากหลายของคำตอบที่ค้นหาในแต่ละรอบการประมวลผลยังไม่มากพอ จึงทำให้คำตอบของรอบการค้นหาในรอบท้ายๆจะซ้ำๆกัน ซึ่งคำตอบที่ซ้ำๆนี้จะเป็นคำตอบที่ใกล้กับคำตอบที่เหมาะที่สุด เป็นผลให้การค้นหาคำตอบใช้จำนวนรอบการประมวลผลหลายรอบกว่าจะทำให้คำตอบที่ได้ในรอบการประมวลผลท้ายๆหลุดออกจากคำตอบที่ซ้ำๆกันหลายๆรอบการประมวลผลมาเป็นคำตอบที่เหมาะที่สุดได้ ซึ่งกระบวนการนี้ต้องการส่วนของอัลกอริทึมที่สามารถหลีกเลี่ยงการได้คำตอบที่ซ้ำๆกับคำตอบของรอบการประมวลผลที่แล้วมา

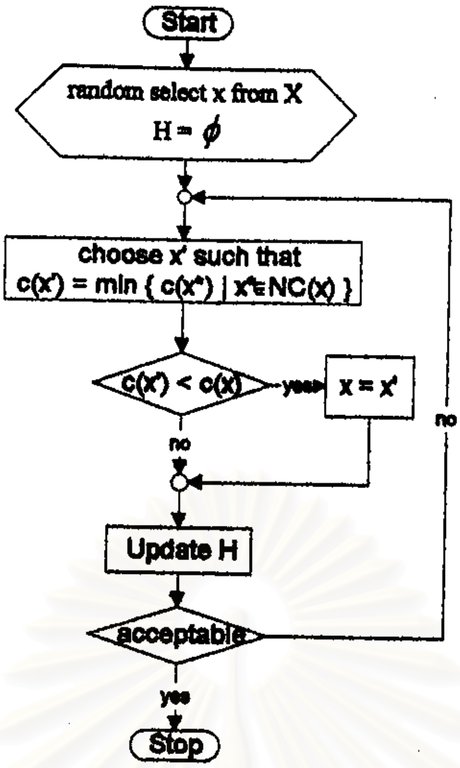


<u>รูปที่ 2.4</u> ไฟล์ชาร์ตแสดงขั้นตอนของอีโวลูชันนารีคอมพิวติง

ทฤษฎีทาบูเซอร์ช(Tabu Search)

ทาบูเซอร์ช (Glover และ Laguna, 1993) เป็นวิธีการในการแก้ปัญหาในการหาคำตอบที่เหมาะที่สุดในวงกว้างวิธีการหนึ่ง โดยมีขั้นตอนในการดำเนินการที่ไม่ซับซ้อนและเหมาะสมสำหรับนำไปใช้แก้ปัญหาที่จัดอยู่ในรูป Combinatorial optimization problem (Yagiura และ Tbaraki, 1996) ทาบูเซอร์ชเป็นอัลกริทึมที่มีรากฐานการพัฒนามาจากอัลกอริทึมอีโวลูชันนารีคอมพิวติง หากจะแตกต่างตรงจำนวนของคำตอบที่นำมาใช้หาคำตอบในรอบการค้นหาต่อไป และได้เพิ่มเทคนิคเพื่อป้องกันปัญหาการเกิดคำตอบซ้ำๆกันหลายๆรอบในช่วงรอบการประมวลผลรอบท้ายๆได้ ในขบวนการของทาบูเซอร์ชนั้นจำเป็นที่จะต้องใช้หน่วยความจำในการเก็บคำตอบของการประมวลผลในรอบการประมวลผลที่ผ่านมาแล้ว ซึ่งจะถูกเรียกว่า ทาบูลิสต์(Tabu list) หรือ ประวัติทาบู(Tabu History) คำตอบในแต่ละรอบของการประมวลผลจะได้มาจากคำตอบที่เหมาะที่สุดในเซตของคำตอบที่เป็นคำตอบที่ใกล้เคียงกับคำตอบในรอบการประมวลผลที่ผ่านมา และคำตอบในเซตนี้จะต้องไม่ซ้ำกับคำตอบที่มีอยู่เดิมในทาบูลิสต์ (Glover, 1989)

หากกำหนดให้ c(x) เป็นฟังก์ชันวัตถุประสงค์ที่ต้องการหาคำตอบ $x \in X$ ที่เหมาะที่สุด โดย X เป็นเซตของคำตอบที่เป็นไปได้ทั้งหมด เซต H เป็นเซตของคำตอบในรอบการประมวลผลที่ผ่านมา และ NC(H, x) เป็นเซตของคำตอบที่ใกล้เคียงกับคำตอบ x และไม่มีสมาชิกของเซตที่ซ้ำกับคำตอบที่เป็นสมาชิกในเซต H แล้ว จะสามารถแสดงขั้นตอนการหาคำตอบที่เหมาะที่สุดได้ดังโฟล์ชาร์ตรูปที่ 2.5



<u>รูปที่ 2.5</u> ไฟล์ชาร์ตแสดงขั้นตอนของทาบูเซอร์ช

จะพบว่าอัลกอริทึมนี้จะเน้นทางด้านความหลากหลายของคำตอบในแต่ละรอบการค้นหา โดยเฉพาะในช่วงรอบการประมวลผลรอบท้ายๆซึ่งต่างจากผลของอีโวลูชันนารีคอมพิวติง และจำนวนของคำตอบที่จะนำมาสร้างกลุ่มคำตอบที่ใกล้เคียงมีเพียงคำตอบเดียวเท่านั้นซึ่งจะทำให้ความหลากหลายของคำตอบในแต่ละรอบการค้นหาไม่กระจายครบคลุมคำตอบทั้งหมดได้ ซึ่งจะทำให้จำนวนรอบการประมวลผลที่มากหลายรอบกว่าคำตอบจะลู่เข้าหาคำตอบที่เหมาะที่สุด (Glover, 1990)

<u>ทฤษฎียีนส์นิติกอัลกอริทึม(Genetic Algorithm)</u> (Goldberg, 1991)

ยีนส์นิติกอัลกอริทึมเป็นวิธีในการทำให้เหมาะที่สุด (Optimization) โดยจะจำลองคำตอบที่ต้องการค้นหาคำตอบที่เหมาะที่สุดในรูปของโครโมโซม (Goldberg, 1991) ซึ่งในโครโมโซมจะประกอบไปด้วยยีนส์หลายๆยีนส์ ซึ่งขั้นตอนของยีนส์นิติกอัลกอริทึมจะประกอบด้วยวิธีการทำยีนส์โอเปอเรชัน กล่าวคือมีการพัฒนาการของยีนส์จนกระทั่งได้ยีนส์ที่ดีที่สุดตามวัตถุประสงค์ที่

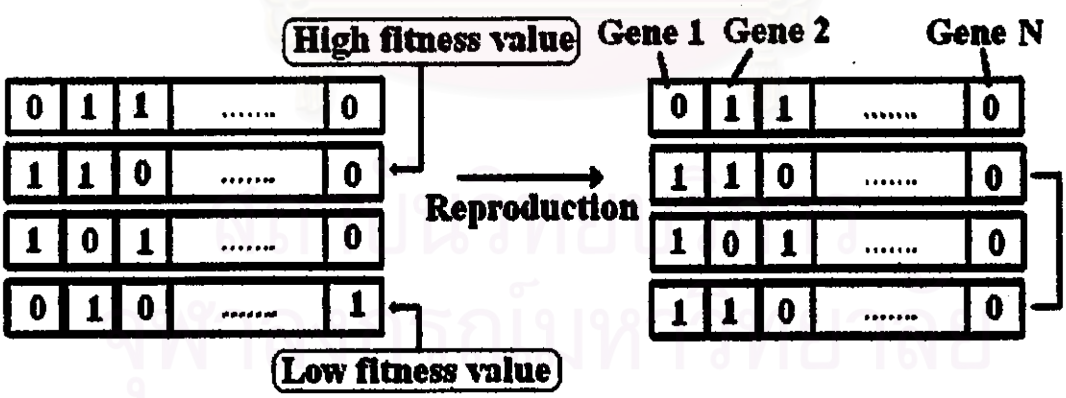
ต้องการ การประเมินค่าว่ายีนส์หรือโครโมโซมมีคุณภาพมากน้อยเพียงใดขึ้นอยู่กับค่าความเหมาะสม (Fitness) ซึ่งเป็นค่าที่ใช้ในการประเมินประสิทธิภาพของโครโมโซมนั้นๆ ในแต่ละโครโมโซมประกอบไปด้วยยีนส์หลายๆยีนส์ซึ่งจะมีจำนวนขึ้นอยู่กับความยาวของโครโมโซม เช่น ถ้าโครโมโซมหนึ่งมีความยาว N จะมียีนส์จำนวน N ตัวเรียงตามตำแหน่งตั้งแต่ตำแหน่งที่ 1 จนถึงตำแหน่งที่ N ถ้าใช้เลขฐาน 2 ในการเข้ารหัสสำหรับแต่ละโครโมโซม โครโมโซมความยาว N จะมีจำนวนโครโมโซมที่เป็นไปได้ทั้งสิ้น 2^N รูปแบบของโครโมโซมจะสามารถแสดงได้ดังรูปที่ 2.6

gene 1	gene2	gene3	gene 4	gene 5	gene6	gene7	gene8
1	0	1	1	0	1	0	1

รูปที่ 2.6 รูปแบบโครโมโซม

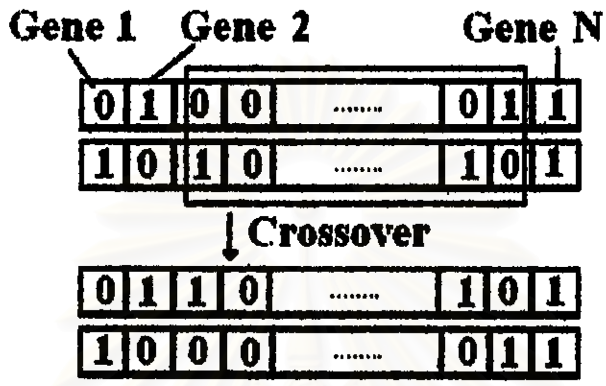
ขั้นตอนในการทำยีนส์โอเปอเรชันแบ่งออกเป็นขั้นตอนดังนี้

1) การทำรีโปรดักชัน (Reproduction) คือการเพิ่มจำนวนและการลดจำนวนของโครโมโซมเนื่องจากโครโมโซมแต่ละตัวมีค่าความเหมาะสมที่แตกต่างกัน โครโมโซมที่มีค่าความเหมาะสมน้อยเป็นโครโมโซมที่มีโอกาสลดจำนวนลงในขณะที่โครโมโซมที่มีค่าความเหมาะสมมากมีโอกาสเพิ่มจำนวนมากขึ้น ดังแสดงในรูปที่ 2.7 เป็นการทำรีโปรดักชันของโครโมโซม 2 โครโมโซมที่มีค่าความเหมาะสมที่แตกต่างกัน



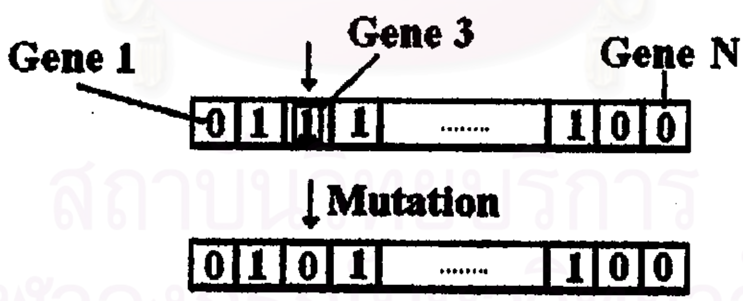
รูปที่ 2.7 การทำรีโปรดักชัน

2) การทำครอสโอเวอร์ (Crossover Operation) เป็นการแลกเปลี่ยนยีนส์ระหว่างโครโมโซมสองโครโมโซมที่แตกต่างกันโดยมี วัตถุประสงค์เพื่อที่จะได้รับโครโมโซมใหม่สองโครโมโซมที่แตกต่างไปจากโครโมโซมเดิม ดังแสดงในรูปที่ 2.8 เป็นการทำครอสโอเวอร์ แบบ 2 จุด ซึ่งภายหลังจากทำการครอสโอเวอร์แล้วยีนส์ที่อยู่ระหว่างจุดที่ทำการครอสโอเวอร์นั้นจะแลกเปลี่ยนกันได้เป็นโครโมโซมใหม่ขึ้นมา



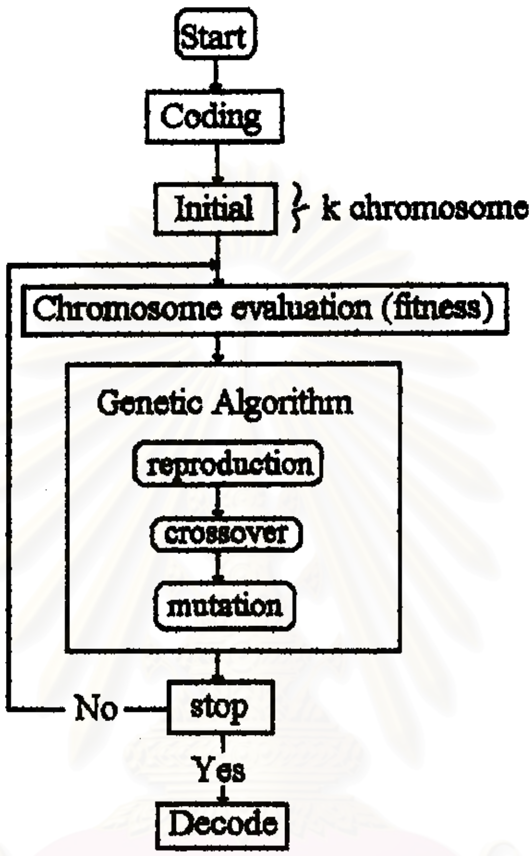
รูปที่ 2.8 การทำครอสโอเวอร์

3) การทำมิวเตชัน (Mutation Operation) มีวัตถุประสงค์เพื่อเปลี่ยนค่ายีนส์ในโครโมโซมหนึ่งให้เป็นค่าใหม่ ดังนั้นโครโมโซมที่ผ่านการมิวเตชันไปแล้วจะได้เป็นโครโมโซมใหม่ ดังแสดงในรูปที่ 2.9 เป็นการแสดงการทำมิวเตชันของยีนส์หนึ่งยีนส์



รูปที่ 2.9 การทำมิวเตชัน

ขั้นตอนของการทำยีนส์ไอเปอเรชันจะกระทำซ้ำในขั้นตอนที่ 1) ถึง 3) จนกว่าค่าเฉลี่ยของความเหมาะสมของประชากร (population) มีการเปลี่ยนแปลงที่น้อยมาก บล็อกไดอะแกรมการทำงานของยีนส์นิติกอัลกอริทึมสามารถแสดงได้ในรูปที่ 2.10 รายละเอียดเกี่ยวกับยีนส์ไอเปอเรชันสามารถศึกษาเพิ่มเติมได้จาก Goldberg (1991)



<u>รูปที่ 2.10</u> บล็อกไดอะแกรมการทำงานของยีนส์นิติกอัลกอริทึม

<u>ทฤษฎีฮิวริสติกอัลกอริทึม(Heuristic Algorithm) (วิธีที่เสนอ)</u>

ฮิวริสติกอัลกอริทึมที่ใช้ในการค้นหาคำตอบที่เหมาะที่สุดนี้ ถูกพัฒนามาจากหลักการของอีโวลูชันนารีคอมพิวติง, ทาบูเซอร์ช และยีนส์นิติกอัลกอริทึม โดยฮิวริสติกอัลกอริทึมจะอาศัยหลักการของอีโวลูชันนารีคอมพิวติงในการสร้างเวกเตอร์ใหม่จากเวกเตอร์แม่ โดยทำการเลือกเวกเตอร์ที่ใกล้เคียงกับเวกเตอร์แม่ ใช้หลักการบัฟเฟอร์ภายนอกเข้ามาเพิ่มความหลากหลายของเวก

เตอร์ในแต่ละรอบการประมวลผลซึ่งเป็นคุณสมบัติของทาบูเซอร์ช การเข้ารหัสเวกเตอร์ และการใช้เวกเตอร์แทนคำตอบของการค้นหา รวมถึงการคำนวณค่า fitness ของแต่ละเวกเตอร์ได้มาจากหลักการของยีนส์นิติกอัลกอริทึม

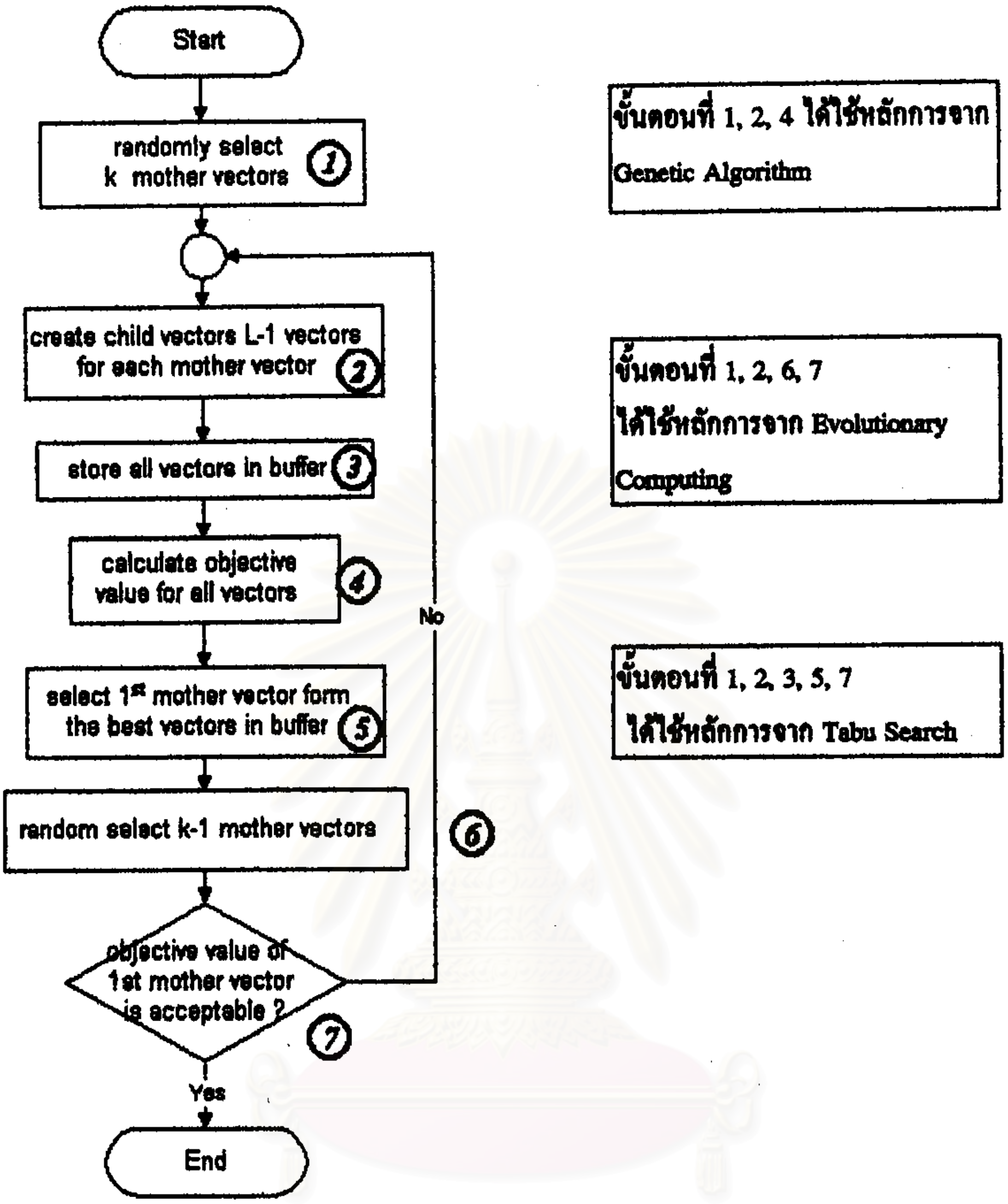
รูปที่ 2.11 เป็นโฟลว์ชาร์ตของฮิวริสติกอัลกอริทึมที่เสนอใช้ในการค้นหาคำตอบที่เหมาะที่สุด โดยจะสามารถอธิบายขั้นตอนได้ 4 ขั้นตอนดังนี้

ขั้นตอนแรก ทำการสร้างเวกเตอร์ขึ้นมาจำนวน k เวกเตอร์ที่แตกต่างกันเรียกเวกเตอร์ k เวกเตอร์นี้ว่าเวกเตอร์แม่(mother vector) ซึ่งเวกเตอร์เหล่านี้จะถูกเลือกมาจากเวกเตอร์คำตอบที่เป็นไปได้ทั้งหมด การจัดรูปแบบคำตอบให้อยู่ในรูปของเวกเตอร์จะคล้ายกับการจัดรูปแบบคำตอบให้อยู่ในรูปแบบโครโมโซมซึ่งใช้ในยีนส์นิติกอัลกอริทึม

ขั้นตอนที่สอง ทำการสร้างเวกเตอร์ขึ้นมาใหม่จากเวกเตอร์แม่แต่ละเวกเตอร์ ซึ่งจะเรียกว่าเวกเตอร์ลูก(child vector) โดยสร้างเวกเตอร์ลูกขึ้นมาที่ไม่ซ้ำกัน $L-1$ เวกเตอร์ต่อเวกเตอร์แม่หนึ่งเวกเตอร์ ทำการเก็บเวกเตอร์ใหม่และเวกเตอร์แม่ลงในบัฟเฟอร์ขนาด $L \cdot k$ เวกเตอร์ และทำการหาค่าวัตถุประสงค์สำหรับทุกๆเวกเตอร์ใหม่ การสร้างเวกเตอร์ลูกขึ้นมาใหม่จากเวกเตอร์แม่จะเป็นลักษณะเดียวกับการสร้างเวกเตอร์ที่ใกล้เคียงกับเวกเตอร์เดิมในทาบูเซอร์ช และอีโวลูชันนารี คอมพิวติง และจะคล้ายกับการทำมิวเตชันในยีนส์นิติกอัลกอริทึม การทำให้คำตอบที่ทำการค้นหาในแต่ละรอบการประมวลผลมีความหลากหลายของคำตอบมากยิ่งขึ้นนั้น การสร้างเวกเตอร์ลูกขึ้นมาใหม่จึงจำเป็นที่จะต้องไม่ซ้ำกับเวกเตอร์ลูกเดิมที่มีอยู่ ซึ่งวิธีการนี้จะได้มาจากแนวคิดการใช้ทาบูลิสต์ในทาบูเซอร์ช สำหรับการหาค่าวัตถุประสงค์ก็จะกระทำในลักษณะเดียวกับขั้นตอนการหาค่าวัตถุประสงค์ของยีนส์นิติกอัลกอริทึม

ขั้นตอนที่สาม เลือกเวกเตอร์ที่มีค่าวัตถุประสงค์ที่น้อยที่สุดจากเวกเตอร์ทั้งหมดที่เก็บในบัฟเฟอร์มาเป็นเวกเตอร์แม่เวกเตอร์แรกในรอบถัดไปและทำการสุ่มเลือกเวกเตอร์แม่ที่เหลืออีก k-1 เวกเตอร์จากเวกเตอร์ทั้งหมดที่เก็บในบัฟเฟอร์โดยไม่ซ้ำกัน ในขั้นตอนนี้จะกระทำคล้ายกับหลักการที่ใช้ในการเลือกคำตอบเริ่มต้นของรอบการประมวลผลรอบต่อไปของทาบูเซอร์ช

ขั้นตอนที่สี่ ตรวจสอบค่าวัตถุประสงค์ของเวกเตอร์ที่มีค่าน้อยที่สุดในบัฟเฟอร์ว่าน้อยพอที่จะสามารถยอมรับได้หรือไม่ หากไม่สามารถยอมรับค่าได้ก็จะกลับไปทำตั้งแต่ขั้นตอนที่สองใหม่ หากสามารถยอมรับค่าได้ เวกเตอร์นั้นจะถูกนำไปถอดรหัสเพื่อเป็นคำตอบที่เหมาะที่สุด ขั้นตอนนี้จะเป็นกระทำแบบเดียวกับยีนส์นิติกอัลกอริทึม, ทาบูเซอร์ช และอีโวลูชันนารี คอมพิวติง



<u>รูปที่ 2.11</u> ไฟล์ชาร์ตของฮิวริสติกอัลกอริทึมที่ได้เสนอ

จากขั้นตอนของฮิวริสติกอัลกอริทึมพบว่าฮิวริสติกอัลกอริทึมมีความคล้ายคลึงกับทาบูเซอร์ช อีโวลูชันนารีคอมพิวติงและขั้นตอนการมิวเทชันในยีนส์นิติกอัลกอริทึม ทั้งนี้อาศัยสมมุติฐานที่ว่าคำตอบที่เหมาะสมกว่าในรอบถัดไปจะเป็นคำตอบที่ใกล้เคียงกับคำตอบในรอบปัจจุบัน และหากทำการสร้างคำตอบที่ใกล้เคียงกับคำตอบเดิมหลายคำตอบมากขึ้นก็จะมีความเป็นไปได้ที่

จะได้คำตอบที่เหมาะสมกว่ามากขึ้นซึ่งอาจพิจารณาได้จากความหลากหลายของคำตอบในแต่ละรอบการประมวลผลของอัลกอริทึม(Yagiura และ Tbaraki, 1996)

เนื่องจากทาบูเซอร์ชใช้คำตอบเดียวในแต่ละรอบในการหาคำตอบที่ใกล้เคียงจึงทำให้การลู่เข้าหาคำตอบจำเป็นต้องใช้จำนวนรอบการประมวลผลที่มาก(Glover, 1990) แต่ในขั้นตอนของทาบูเซอร์ชได้ใช้ทาบูลิสทำให้แต่ละคำตอบที่ได้ในแต่ละรอบการประมวลผลต่างกัน(Glover, 1989) ดังนั้นจำนวนคำตอบใหม่ที่เกิดขึ้นในแต่ละรอบเท่ากับ 1 คำตอบ ฮิวริสติกอัลกอริทึมได้ดัดแปลงแนวคิดในการหาคำตอบที่ใกล้เคียงจากทาบูเซอร์ชมาเป็นการใช้จำนวนคำตอบในการหาคำตอบที่ใกล้เคียงใน 1 รอบการประมวลผลเท่ากับจำนวนของเวกเตอร์แม่จึงทำให้มีความหลากหลายของคำตอบมากกว่าทาบูเซอร์ช ทำให้โอกาสที่จะพบคำตอบที่เหมาะสมกว่ามีมากขึ้น นอกจากนี้การใช้ทาบูลิสของทาบูเซอร์ชมีความยุ่งยากต่อการเขียนโปรแกรมเพราะต้องเขียนโปรแกรมในส่วนที่คอยตรวจสอบคำตอบในแต่ละรอบกับคำตอบเดิมที่เก็บไว้ในทาบูลิส และเวลาส่วนใหญ่ที่เสียไปในขณะประมวลผลก็คือเวลาที่โปรแกรมใช้ตรวจสอบคำตอบในทาบูลิส(Yagiura และ Tbaraki, 1996) ฮิวริสติกอัลกอริทึมจึงไม่ใช้ส่วนของทาบูลิสมาทำการเพิ่มความหลากหลายของคำตอบแต่ทำการเพิ่มความหลากหลายของคำตอบด้วยวิธีอื่นแทน

โดยแนวคิดที่จะใช้จำนวนคำตอบที่ใช้หาคำตอบที่ใกล้เคียงมากกว่า 1 นั้นได้มาจากขั้นตอนของอีโวลูชันนารีคอมพิวติง แต่ได้ทำการดัดแปลงให้ 1 คำตอบเดิมในฮิวริสติกอัลกอริทึมสามารถสร้างคำตอบที่ใกล้เคียงได้มากกว่า 1 คำตอบใหม่ซึ่งต่างจากอีโวลูชันนารีคอมพิวติงซึ่งการสร้างคำตอบที่ใกล้เคียงกับคำตอบเดิมจะสร้างเพียง 1 คำตอบใหม่ต่อ 1 คำตอบเดิม(Glover และ Laguna, 1993) ซึ่งก็เป็นผลให้ใช้จำนวนรอบการประมวลผลลดลง นอกจากนี้แล้วการเลือกกลุ่มคำตอบที่เหมาะสมในแต่ละรอบของอีโวลูชันนารีคอมพิวติงจะได้จากคำตอบที่เหมาะสมที่สุดครึ่งแรกจากกลุ่มคำตอบที่สร้างจากคำตอบที่ใกล้เคียงคำตอบเดิม ทำให้คำตอบในกลุ่มคำตอบที่ใกล้เคียงในรอบการประมวลผลท้ายๆมีคำตอบที่ซ้ำกันมากซึ่งทำให้โอกาสที่จะพบคำตอบที่เหมาะที่สุดลดลง(Yagiura และ Tbaraki, 1996) แต่ฮิวริสติกอัลกอริทึมจะเลือกคำตอบที่ดีที่สุดในกลุ่มคำตอบที่ใกล้เคียงเพียงคำตอบเดียว และเลือกคำตอบที่เหลือในกลุ่มจากคำตอบที่เป็นคำตอบที่ใกล้เคียงโดยการสุ่มเลือก ทำให้คำตอบที่ใกล้เคียงในรอบการประมวลผลท้ายๆไม่เกิดการซ้ำกัน และเวลาที่ใช้ในการประมวลผล 1 รอบของฮิวริสติกอัลกอริทึมจะมากกว่าอีโวลูชันนารีคอมพิวติงเพียงเล็กน้อย แต่เมื่อเทียบกับจำนวนรอบที่ลดลงพบว่าเวลาที่ใช้ทั้งหมดของฮิวริสติกอัลกอริทึมน้อยกว่าอีโวลูชันนารีคอมพิวติงมาก

การสร้างคำตอบที่ใกล้เคียงของฮิวริสติกอัลกอริทึมได้จากขั้นตอนมิวเตชันในยีนส์นิติกอัลกอริทึม จากขั้นตอนของยีนส์นิติกอัลกอริทึมพบว่าขั้นตอนที่ใช้ในการสร้างคำตอบที่ใกล้เคียงคือขั้นตอนการทำครอสโอเวอร์และขั้นตอนการมิวเตชัน(Tanterdtid, 1997) ในรอบการประมวลผลรอบท้ายๆพบว่าคำตอบในกลุ่มคำตอบจะซ้ำกันมาก ดังนั้นการสร้างคำตอบที่ใกล้เคียงกับคำตอบเดิมโดยการทำครอสโอเวอร์จะไม่สามารถที่จะได้คำตอบใหม่ที่ไม่ซ้ำกับคำตอบเดิม การสร้างคำตอบใหม่จึงขึ้นกับการทำการมิวเตชันเท่านั้น จากขั้นตอนของยีนส์นิติกอัลกอริทึมพบว่าค่าวัตถุประสงค์ในแต่ละรอบการประมวลผลมีค่าแกว่งไปมาทำให้ยากต่อการตัดสินใจยุติการประมวลผล (Goldberg, 1991) เนื่องจากการที่ค่าวัตถุประสงค์แกว่งขึ้นๆลงๆทำให้ใช้จำนวนรอบในการประมวลผลมากกว่าฮิวริสติกอัลกอริทึม การพัฒนาโปรแกรมประมวลของยีนส์นิติกอัลกอริทึมทำได้ยากกว่าฮิวริสติกอัลกอริทึม เพราะในขั้นตอนการทำครอสโอเวอร์นั้น การเขียนส่วนโปรแกรมที่ทำหน้าที่สลับยีนส์ภายในโครโมโซมทำได้ยาก และเวลาที่เสียไปในการประมวลผลคือเวลาในการทำครอสโอเวอร์ หากโครโมโซมยิ่งยาวมากเท่าไรเวลาที่ใช้ในการสลับยีนส์ภายในโครโมโซมจะยิ่งมากขึ้น(Goldberg, 1991)

จากทฤษฎีและขั้นตอนการค้นหาคำตอบของอัลกอริทึมที่ใช้สำหรับค้นหาคำตอบที่เหมาะที่สุดข้างต้นนี้ พบว่าถ้าหากทำการเลือกคำตอบที่ใช้เริ่มต้นอัลกอริทึมให้ใกล้กับคำตอบที่เหมาะที่สุดมากๆได้ ก็จะทำให้จำนวนรอบในการค้นหาคำตอบที่เหมาะที่สุดลดลงได้