บทที่ 4

เทคนิคการหาค่าเหมาะสมที่สุด

Linear Programming

Linear Programming เป็นเทคนิคการหาค่าเหมาะสมที่สุดสำหรับปัญหาที่มี ออปเจกทีฟฟังก์ชั่น และสมการเงื่อนไขเป็นฟังก์ชั่นที่เป็นเชิงเส้น โดยสมการเงื่อนไขนั้น เป็นแบบเท่ากับ หรือไม่เท่ากับก็ได้

1. รูปแบบมาตรฐานของ Linear Programming

ลักษณะปัญหาทาง Linear Programimg จะเขียนในรูปมาตรฐาน โดยจะเป็นการหาค่าต่ำสุดของออปเจกทีฟฟังก์ชั่น สมการเงื่อนไขทั้งหมดจะเป็นแบบเท่ากับ และตัวแปรออกแบบทั้งหมดต้องมากกว่า · หรือเท่ากับศูนย์ ซึ่งจากมาตรฐานข้างต้นสามารถเขียนเป็นรูปแบบคณิตศาสตร์ได้ทั้ง แบบสเกลลา และแบบเวกเตอร์

แบบสเกลลา

หาค่าต่ำสุดของ

$$f(x_1, x_2, \ldots, x_n) = c_1x_1 + c_2x_2 + \ldots + c_nx_n$$

โดยมีสมการเงื่อนไข

$$\begin{aligned} a_{11}x_1 + a_{12}x_2 + \ldots + a_{1n}x_n &= b_1 \\ a_{21}x_1 + a_{22}x_2 + \ldots + a_{2n}x_n &= b_2 \\ &\vdots \\ a_{m1}x_1 + a_{m2}x_2 + \ldots + a_{mn}x_n &= b_m \end{aligned}$$

และ $x_1, x_2, \ldots, x_n \geq 0$

แบบเวกเตอร์

หาค่าต่ำสุดของ $C^T.\underline{X}$

โดยมีสมการเงื่อนไข

$$a.\underline{X} = b$$

และ $$\underline{X} \geq 0$$

โดย $$\underline{X} = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix}$$

$$a = \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & & & \\ a_{m1} & a_{m2} & \cdots & a_{mn} \end{bmatrix}$$

$$b = \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_m \end{bmatrix} \qquad c = \begin{bmatrix} c_1 \\ c_2 \\ \vdots \\ c_m \end{bmatrix}$$

2. การแปลงให้อยู่ในรูปแบบมาตรฐาน

โดยทั่วไปปัญหาทาง Linear Programming นั้นอาจจะไม่อยู่ในรูปแบบมาตรฐานก็ได้ แต่ Linear Programming ก็ยังสามารถใช้กับปัญหาดังกล่าวได้โดยการแปลงรูปแบบของปัญหานั้นให้อยู่ในรูปมาตรฐาน

ก. เมื่อต้องการหาค่าสูงสุดของออปเจกทีฟฟังก์ชั่น

ค่าสูงสุดของฟังก์ชั่น $f(x_1,x_2, \ldots ,x_n)$ จะมีค่าเท่ากับค่าต่ำสุดของฟังก์ชั่น $-f(x_1,x_2, \ldots , x_n)$ ดังนั้นเมื่อต้องการหาค่าสูงสุดของออปเจกทีฟฟังก์ชั่นก็สามารถทำได้โดย หาค่าต่ำสุดของออปเจกทีฟฟังก์ชั่นที่เป็นลบนั้น ดังนี้

หาค่าสูงสุดของ

$$f = c_1x_1+c_2x_2+ \ldots +c_nx_n$$

แปลงเป็น หาค่าต่ำสุดของ

$$f' = -f = -c_1x_1-c_2x_2- \ldots -c_nx_n$$

ข. เมื่อเงื่อนไขเป็นแบบไม่เท่ากับ

ถ้าเงื่อนไขที่ต้องการเป็น แบบน้อยกว่า หรือเท่ากับก็สามารถแปลงเป็นเงื่อนไขแบบเท่ากับได้ โดยกำหนดตัวแปรที่มากกว่าเท่ากับศูนย์บวกเข้าไป ซึ่งเรียก ตัวแปรที่บวกเข้าไปว่า ตัวแปร Slack

เงื่อนไขแบบน้อยกว่า หรือเท่ากับ

$$a_{k1}x_1+a_{k2}x_2+ \ldots +a_{kn}x_n \leq b_k$$

แปลงเป็นแบบเท่ากับได้

$$a_{k1}x_1+a_{k2}x_2+ \ldots +a_{kn}x_n+x_{n+1} = b_k$$

ถ้าเงื่อนไขที่ต้องการเป็นแบบมากกว่า หรือเท่ากับ จะแปลงรูปโดยลบสมการเงื่อนไขด้วย ตัวแปร Slack

เงื่อนไขแบบมากกว่าหรือเท่ากับ

$$a_{k1}x_1+a_{k2}x_2+ \ldots +a_{kn}x_n \geq b_k$$

แปลงเป็นแบบเท่ากับได้

$$a_{k1}x_1+a_{k2}x_2+ \ldots +a_{kn}x_n-x_{n+1} = b_k$$

โดย x_{n+1} = ตัวแปร Slack

ค. เมื่อตัวแปรออกแบบน้อยกว่าศูนย์

โดยทั่วไป ค่าของตัวแปรออกแบบจะมากกว่า หรือเท่ากับศูนย์ แต่อาจมีบางกรณีที่ตัวแปรออกแบบอาจมากกว่า เท่ากับ หรือน้อยกว่าศูนย์ก็ได้ ดังนั้นถ้าไม่ได้กำหนดว่าตัวแปรออกแบบต้องมากกว่า หรือเท่ากับศูนย์ ตัวแปรออกแบบนั้นอาจเป็นค่าลบก็ได้ ในกรณีนี้สามารถแปลงตัวแปรออกแบบ โดยให้ตัวแปรออกแบบนี้เท่ากับผลต่างของตัวแปรที่มากกว่า หรือเท่ากับศูนย์ 2 ตัว

ค่า x_j เป็นตัวแปรออกแบบที่มีค่าเป็นลบได้

แปลง $x_j = x_j' - x_j''$

เมื่อ $x_j', x_j'' \geq 0$

วิธี Simplex

การแก้ปัญหา Linear Programming นั้นคือ การหา Basic Feasible Solution ที่ทำให้ออปเจกทีฟฟังก์ชั่นมีค่าสูงสุด หรือต่ำสุดนั้นเอง ในกรณีที่มีตัวแปร หรือมีสมการเงื่อนไขมากก็จะมี Basic Feasible Solution มากด้วย การที่ต้องหา Basic Feasible Solution ทุกชุดแล้วพิจารณาหาชุดที่ให้ค่าออปเจกทีฟฟังก์ชั่นมีค่าเหมาะสมที่สุดนั้น จึงเป็นการแก้ปัญหา Linear Programming ที่ยุ่งยาก และเสียเวลามาก

วิธี Simplex เป็นการแก้ปัญหาด้วยการหา Basic Feasible Solution เป็นชุด ๆ โดย Basic Feasible Solution แต่ละชุดที่หาได้จะให้ค่าออปเจกทีฟฟังก์ชั่นที่ดีขึ้นเรื่อย ๆ จนได้ค่าเหมาะสมที่สุด

1. Simplex Algorithm

วิธี Simplex จะเริ่มด้วยการจัดออปเจกทีฟฟังก์ชั่น และสมการเงื่อนไขแบบเท่ากับ ให้อยู่ในรูปแบบ Canonical ที่มี Basic Feasible Solution ดังแสดงในสมการ 4.1

$$
\begin{aligned}
&1.x_1+0.x_2+ \ldots +0.x_m+a'_{1,m+1}x_{m+1}+ \ldots +a'_{1,n}x'_n = b_1' \\
&0.x_1+0.x_2+ \ldots +0.x_m+a'_{2,m+1}x_{m+1}+ \ldots +a'_{2,n}x'_n = b_2' \\
&\vdots \\
&0.x_1+0.x_2+ \ldots +0.x_m -f + C'_{m+1}x_{m+1} + \ldots +C'_{mn}x_n = -f'_0
\end{aligned}
$$

----- 4.1

โดย a'_{ij}, C'_j, b'_i และ f'_0 เป็นค่าคงที่ ส่วน (-f) เป็นฟังก์ชั่นของตัวแปร Basic ของรูปแบบ Canonical

จากสมการ 4.1 ถ้าให้ $x_i = 0$; $i = m+1, m+2, \ldots, n$ จะได้

$$x_i = b'_i \quad , \quad i = 1,2,\ldots,m$$

$$f = f'_0$$

x_i ; $i=1,2, \ldots ,m$ จะเป็น Basic Feasible Solution เมื่อ $b_i \geq 0$; $i=1,2, \ldots ,m$

ถ้า C'_j ; $j = m+1, m+2, \ldots, n$ ในสมการ 4.1 มีค่าเป็นบวก จะได้ Basic Feasible Solution ที่เป็น Optimal Basic Solution แต่ถ้า C'_j มีค่าเป็นลบ ค่าของออปเจกทีฟฟังก์ชั่น ยังสามารถทำให้ลดลงได้อีก โดยเปลี่ยนตัวแปร Non Basic x_j ให้เป็นตัวแปร Basic และเปลี่ยนตัวแปร Basic ตัวใดตัวหนึ่งเป็นตัวแปร Non Basic

ในกรณีที่มี $C'_j < 0$ หลายตัวนั้นจะต้องเปลี่ยนตัวแปร Non-Basic x_j ที่มีค่า C'_j น้อยที่สุดเป็นตัวแปร Basic จากสมการ 4.1 ถ้าให้ C'_s เป็น C'_j ที่น้อยที่สุด และ x_s เป็น x_j จะได้

$$\begin{aligned}
x_1 &= b'_1 - a_{1s}x_s \quad , \quad b'_1 \geq 0 \\
x_2 &= b'_2 - a_{2s}x_s \quad , \quad b'_2 \geq 0 \\
x_m &= b'_m - a_{ms}x_s \quad , \quad b'_m \geq 0 \\
f &= f'_0 + C'_s x_s \quad , \quad C'_s < 0
\end{aligned}$$

----- 4.2

ในสมการ 4.2 ถ้าค่าของ $a_{is} < 0$; $i = 1,2, \ldots ,n$ ถือว่าเป็น Unbounded Solution แต่ถ้ามี $a_{is} > 0$ ค่า x_s มากที่สุดที่ไม่ทำให้ x_i ทั้งหมดน้อยกว่าศูนย์ แต่จะทำให้ออปเจกทีฟมีค่าลดลงมากที่สุดจะเท่ากับ b'_i/a_{is}

ในกรณีที่มี $a_{is} > 0$ หลายตัว อัตราส่วน b'_i/a_{is} ที่น้อยที่สุดจะให้ค่า x_s ที่มีค่ามากที่สุด ซึ่งจะทำให้ออปเจกทีฟมีค่าลดลงมากที่สุด ถ้าให้ b'_r/a_{rs} เป็น b'_i/a_{is} ที่น้อยที่สุด และ x^*_s เป็น x_s ที่มีค่ามากที่สุด เมื่อนำค่า x^*_s มาลบกับสมการ 4.2 ซึ่งจะได้ผลลัพธ์ดังนี้

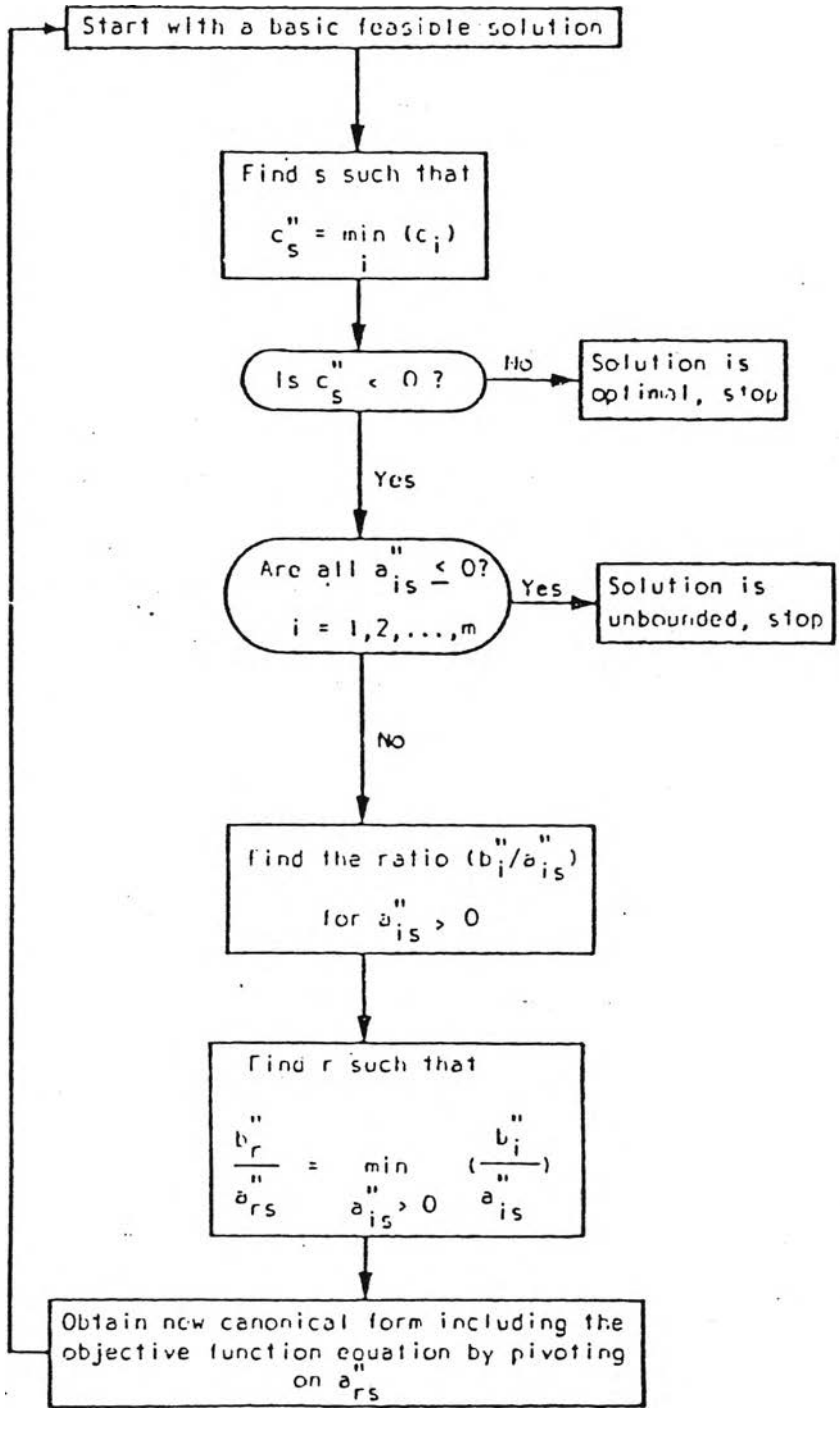
$$
\begin{aligned}
x_s &= x_s^* \\
x_i &= b'_i - a_{is}\, x_s^* > 0 \\
&\qquad\qquad i = 1,2, \ldots ,m \text{ และ } i = r \\
x_r &= 0 \\
x_j &= 0 \ , \ j = m+1, m+2, \ldots n \text{ และ } j = s \\
f &= f'_s + C'_s x_s^* \ , \quad C_s x_s \ , C_s < o
\end{aligned}
$$

----- 4.3

จากสมการ 4.3 จะเห็นว่าค่าออปเจกทีฟฟังก์ชั่นจะมี ค่าลดลงเมื่อเปรียบกับค่าออปเจกทีฟฟังก์ชั่นในสมการ 4.1 Basic Feasible Solution ที่ได้ใหม่นี้จะถูกตรวจสอบว่าเป็น Optimal Basic Solution หรือไม่ โดยสังเกตจาก C'_j ในรูปแบบ Canonical ว่าเป็นบวกหมดทุกตัวหรือไม่ ซึ่งถ้าไม่ การเปลี่ยนชุด Basic Feasible Solution จะมีขึ้นอีกจนกว่าจะได้ Optimal Basic Solution รูป 4.1 จะแสดงแผนผังขั้นตอนของ Simplex Algorithm

2. วิธี Simplex แบบสองเฟส (Two Phases of Simplex Method)

ในการแก้ปัญหา Linear Programming โดยใช้ Simplex Algorithm นั้นขั้นแรกต้องจัดออปเจกทีฟฟังก์ชั่น และสมการเงื่อนไขให้อยู่ในรูปแบบ Canonical ที่สามารถหา Basic Feasible Solution ได้ก่อน แล้วจึงเปลี่ยน Basic Feasible Solution ที่ได้นี้ไปยังชุดที่ให้ค่าออปเจกทีฟฟังก์ชั่น ที่มีค่าลดลง



รูปที่ 4.1 แสดงแผนผังขั้นตอนของ Simplex Algorithm

ในกรณีปัญหา Linear Programming มีสมการเงื่อนไขเป็นแบบเท่ากับ และแบบมากกว่าหรือเท่ากับ จะทำให้ Linear Programming นี้ไม่มีตัวแปร Slack และ มีตัวแปร Slack ที่เป็นลบตามลำดับซึ่งจะไม่สามารถสร้างรูปแบบ Canonical ที่ให้ Basic Feasible Solution ได้

วิธี Simplex แบบสองเฟส จะสามารถนำมาแก้ปัญหาลักษณะนี้ได้โดยวิธี Simplex แบบสองเฟสนี้จะแบ่งขั้นตอนการคำนวณเป็น 2 เฟส ในเฟสแรกจะสร้าง Auxiliary Problem ขึ้นมาโดยเพิ่มตัวแปรลงไปในสมการเงื่อนไขที่ไม่มีตัวแปร Slack หรือมีตัวแปร Slack เป็นลบ ตัวแปรที่เพิ่มลงไปนี้เรียกว่า ตัวแปร Artificial ซึ่งเมื่อเพิ่มตัวแปร Artificial เข้าไปแล้วจะจัดรูปแบบ Canonical ที่ให้ Basic Feasible Solution ได้แล้วเริ่มใช้ Simplex Algorithm เพื่อทำให้ค่าของ Auxiliary Function มีค่าต่ำสุด ซึ่งเมื่อได้ค่าต่ำสุดของ Auxiliary Problem นั้นจะได้ Basic Feasible Solution ชุดแรกของปัญหา Linear Programming ด้วยซึ่งถือว่าจบเฟสแรก ในเฟสที่สองนั้น เมื่อได้ Basic Feasible Solution ชุดแรกแล้วก็ใช้ Simplex Algorithm เพื่อหาค่าต่ำสุดของออปเจกทีฟฟังก์ชั่นต่อไปจนได้ Optimal Basic Solution การคำนวณของวิธี Simplex แบบสองเฟส อธิบายเป็นขั้นตอน ได้ดังนี้

ก. จัดปัญหา Linear Programming ในรูป

$$
\begin{aligned}
a_{11}x_1+a_{12}x_2+ \ldots +a_{1n}x_n &= b_1\\
a_{21}x_1+a_{22}x_2+ \ldots +a_{2n}x_n &= b_2\\
&\vdots\\
a_{m1}x_1+a_{m2}x_2+ \ldots +a_{mn}x_n &= b_m\\
C_1 x_1+c_2x_2 + \ldots +C_n x_n &= f
\end{aligned}
$$

----- 4.4

โดย $b_i \geq 0$; $i = 1,2, \ldots,m$

ข. เพิ่มตัวแปร Artificial $y_1, y_2, \ldots, y_m$ ซึ่งมากกว่า หรือเท่ากับศูนย์ ซึ่งจะได้

$$\begin{aligned}
a_{11}x_1 + a_{12}x_2 + \ldots + a_{1n}x_n + y_1 &= b_1 \\
a_{21}x_1 + a_{22}x_2 + \ldots + a_{2n}x_n + y_2 &= b_2 \\
&\vdots \\
a_{m1}x_1 + a_{m2}x_2 + \ldots + a_{mn}x_n + y_m &= b_m \\
C_1x_1 + C_2x_2 + \ldots + C_nx_n + (-f) &= 0
\end{aligned}$$

----- 4.5

ค. เริ่มเฟสแรกสร้าง Auxiliary Problem (w) โดย w เท่ากับ ผลรวมของตัวแปร Artificail ได้

$$w = y_1 + y_2 + \ldots + y_m$$

ง. หาค่า $x_i \geq 0$; i=1, 2, ...,n และ $y_i \geq 0$ i=1, 2, ... ,m ที่ทำให้ w มีค่าต่ำสุด และ เป็นไปตามสมการ 4.5 ดังนี้

$$\begin{aligned}
a_{11}x_1 + a_{12}x_2 + \ldots + a_{1n}x_n + y_1 &= b_1 \\
a_{21}x_1 + a_{22}x_2 + \ldots + a_{2n}x_n + y_2 &= b_2 \\
&\vdots \\
a_{m1}x_1 + a_{m2}x_2 + \ldots + a_{mn}x_n + y_m &= b_m \\
C_1x_1 + C_2x_2 + \ldots + C_nx_n + (-f) &= 0 \\
y_1 + y_2 + \ldots + y_n + (-w) &= 0
\end{aligned}$$

----- 4.6

จากสมการ 4.6 ยังไม่ใช่รูปแบบ Canonical แต่สามารถแปลงให้เป็นรูปแบบ Canonical ที่มี $y_1, y_2, \ldots, y_m$ เป็นตัวแปร Basic ได้

โดยลบสมการสุดท้ายของสมการ 4.6 ด้วยผลรวมของ m สมการแรกจะได้รูปแบบ Canonical ดังนี้

$$
\begin{aligned}
a_{11}x_1 + a_{12}x_2 + \ldots + a_{1n}x_n + y_1 &= b_1 \\
a_{21}x_2 + a_{22}x_2 + \ldots + a_{2n}x_n + y_2 &= b_2 \\
a_{m1}x_m + a_{m2}x_m + \ldots + a_{mn}x_m + y_m &= b_m \\
C_1x_1 + C_2x_2 + \ldots + C_nx_n + (-f) &= o \\
d_1x_1 + d_2x_2 + \ldots + d_nx_n + (-w) &= -wo
\end{aligned}
$$

----- 4.7

เมื่อ

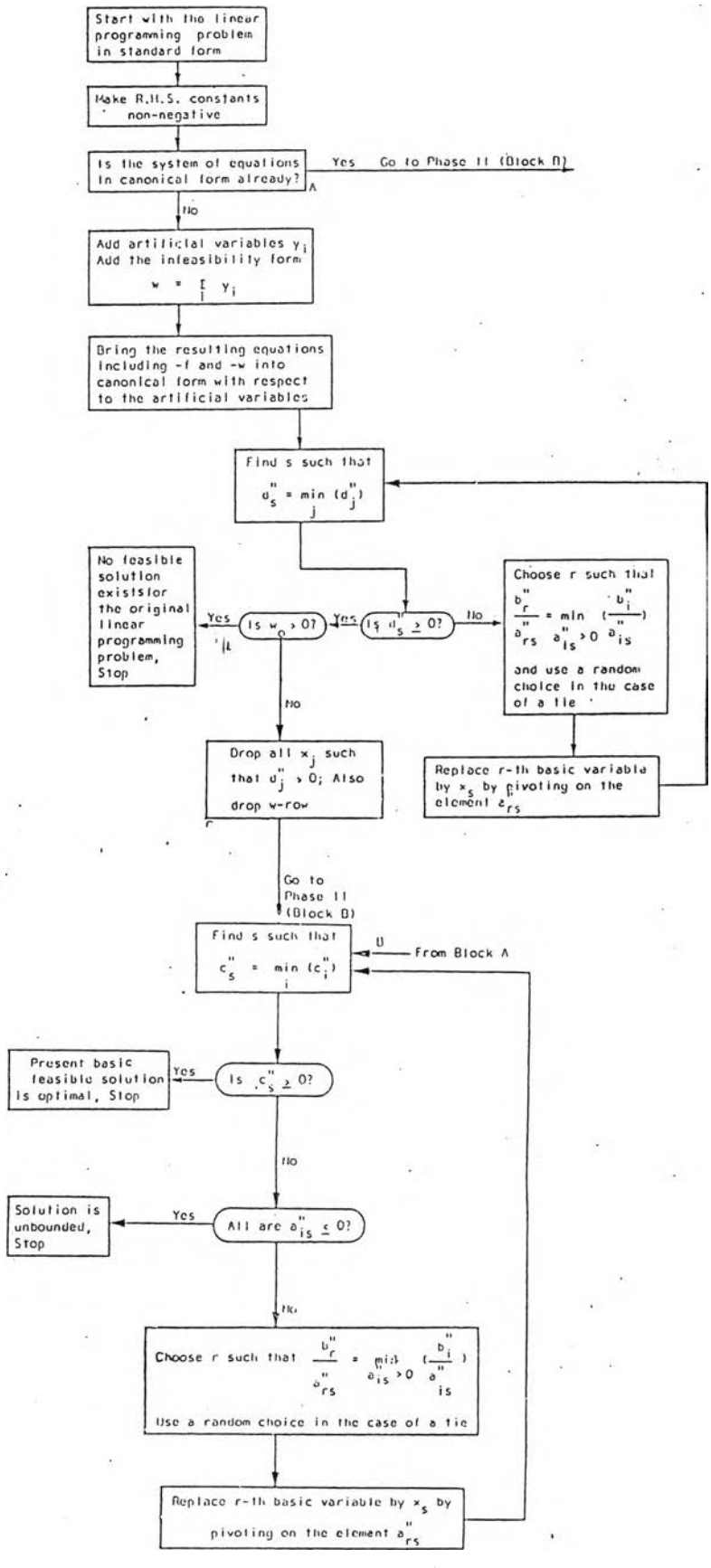
$$d_i = -(a_{1i} + a_{2i} + \ldots + a_{mi}) \ , \ i = 1,2,\ldots,n$$

$$-wo = -(b_1 + b_2 + \ldots b_m)$$

จากสมการ 4.7 จะได้ Basic Feasible Solution ที่จะใช้หาค่าต่ำสุดของ w โดยใช้ Simplex Algorithm

จ. ในการหาค่าต่ำสุดของ w นั้น ถ้าค่า w ที่ได้มีค่ามากกว่าศูนย์แล้วแสดงว่าไม่มี Feasible Solution สำหรับปัญหา Linear Programming นี้แต่ถ้าค่าต่ำสุดที่ได้นี้ เท่ากับศูนย์ รูปแบบ Canonical ที่ให้ผลลัพธ์นี้จะให้ Basic Feasible Solution ของ Linear Programming ซึ่งหลังจากนี้สามารถตัดสมการ w และคอลัมน์ของ $y_1, y_2, \ldots, y_m$ ไปได้

ฉ. เริ่มเฟสสองโดยใช้ Simplex Algorithm จัดรูปแบบ Canonical เมื่อจบเฟสแรก เพื่อให้ได้ Optimal Basic Solution ซึ่งรูปที่ 4.2 จะแสดงขั้นตอนการคำนวณด้วยวิธี Simplex แบบสองเฟส



รูปที่ 4.2 แสดงขั้นตอนการคำนวณ วิธี Simplex แบบสองเฟส

Non-Linear Programming

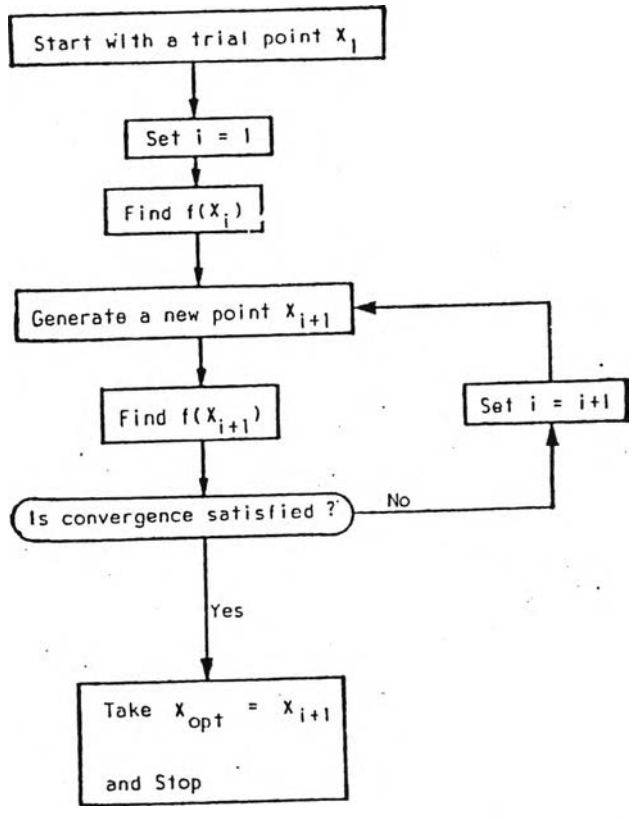
Non-Linear Programming เป็นเทคนิคการหาค่าเหมาะสมที่สุด ในกรณีที่ ออปเจกทีฟฟังก์ชั่น และสมการเงื่อนไข เป็นฟังก์ชั่นที่ไม่เป็นเชิงเส้น สำหรับ Non-Linear Programming นี้จะแบ่งเป็น 2 แบบ ตามลักษณะของปัญหา คือ Non-Linear Programming แบบไม่มีสมการเงื่อนไข และแบบมีสมการเงื่อนไข

1. Non-Linear Programming แบบไม่มีสมการเงื่อนไข

การแก้ปัญหา Non-Linear Programming แบบไม่มีสมการเงื่อนไข คือ การหาค่าตัวแปรออกแบบ ($\underline{X}$) ที่ทำให้ออปเจกทีฟฟังก์ชั่นซึ่งเป็นฟังก์ชั่นไม่เป็นเชิงเส้นมีค่าสูงสุด หรือต่ำสุด เทคนิคการแก้ปัญหา Non-Linear Programming แบบไม่มีสมการเงื่อนไข สามารถแบ่งออกได้ 2 ประเภท ตามลักษณะการคำนวณ คือ ประเภท Gradient Method และ Derivative Free Method

Gradient Method จะมีทั้งการคำนวณ และหาอนุพันธ์ย่อยของออปเจกทีฟฟังก์ชั่น ส่วน Derivative Free Method จะมีแต่การคำนวณออปเจกทีฟฟังก์ชั่นอย่างเดียว แต่การแก้ปัญหาทั้ง 2 ประเภทนี้จะมีขั้นตอนทั่ว ๆ ไปที่คล้ายกัน คือ จะกำหนดจุดเริ่มต้นซึ่งถือว่าเป็นจุดที่ดีสุดเริ่มต้น แล้วเคลื่อนจุดที่กำหนดให้นี้ไปยัง จุดที่ให้ค่าออปเจกทีฟฟังก์ชั่นที่ดีขึ้น รูปที่ 4.3 สามารถแสดงขั้นตอนทั่ว ๆ ไป ของการแก้ปัญหา Non-Linear Programming แบบไม่มีสมการเงื่อนไข

จากรูปที่ 4.3 การแก้ปัญหาทั้ง 2 ประเภทมีที่เหมือนกัน คือ จะเริ่มต้นด้วยการกำหนดจุด x_1 เพื่อเริ่มขั้นตอนการแก้ปัญหา ส่วนที่จะแตกต่างกันก็ คือ วิธีการสร้างจุดใหม่ x_{i+1} จาก x_i และการทดสอบว่า จุดใหม่นี้เป็นจุดที่ดีที่สุด หรือไม่



รูปที่ 4.3 แสดงขั้นตอนการแก้ปัญหา Non-Linear Programming แบบไม่มีสมการเงื่อนไข

2. Non-Linear Programming แบบมีสมการเงื่อนไข

Non-Linear Programming แบบมีสมการเงื่อนไข คือ การหาค่าตัวแปรออกแบบ ($\underline{X}$) ที่ทำให้ออปเจกทีฟฟังก์ชั่นที่เป็นฟังก์ชั่นไม่เป็นเชิงเส้นมีค่าเหมาะสมที่สุด และค่าตัวแปรออกแบบนี้จะต้องทำให้สมการเงื่อนไขที่กำหนดเป็นจริงด้วย เทคนิคในการแก้ปัญหา Non-Linear Programming แบบมีเงื่อนไขนี้มีหลายเทคนิค ซึ่งจะแบ่งเป็นประเภทใหญ่ได้ 2 ประเภท คือ ประเภท Feasibility Check และ Modified Objective Function

Feasibility Check นั้นจะคล้ายกับ การแก้ปัญหาแบบไม่มีสมการเงื่อนไข แต่จะเพิ่มขั้นตอนการตรวจสอบสมการเงื่อนไขว่าเป็นจริง หรือไม่เข้าไป ส่วนวิธี Modified Objective Function จะใช้การรวมสมการเงื่อนไขเข้าไว้ในออปเจกทีฟฟังก์ชั่น แล้วสร้างเป็นปัญหาแบบไม่มีเงื่อนไข

เทคนิคการหาค่าเหมาะสมที่สุดของ Rosenbrock

1. แบบไม่มีสมการเงื่อนไข

การแก้ปัญหา Non-Linear Programming แบบไม่มีสมการเงื่อนไขของเทคนิค Rosenbrock จะใช้สำหรับหาค่าต่ำสุดของออปเจกทีฟฟังก์ชั่น $f(\underline{X})$ โดยจะเป็นลักษณะ Derivate Free Method คือ ไม่ต้องหาอนุพันธ์ย่อยของ $f(\underline{X})$ เพื่อกำหนดทิศทางการเคลื่อนที่ เทคนิคของ Rosenbrock นี้จะเปลี่ยนทิศทางการเคลื่อนที่ด้วยการหมุนระบบโคออร์ดิเนต โดยจะต้องกำหนดทิศทางการเคลื่อนที่เริ่มต้นให้ ส่วนระยะทางในการเคลื่อนที่เมื่อกำหนดให้ตอนเริ่มต้นแล้วจะมีการกำหนดระยะทางการเคลื่อนที่ต่อไปด้วยตัวเทคนิคเอง

ก. ขั้นตอนการคำนวณ

1. ก่อนจะเริ่มหาค่าต่ำสุดจะต้องกำหนด ระยะการเคลื่อนที่เริ่มต้น และทิศทางการเคลื่อนที่ เริ่มต้นของตัวแปรแต่ละตัวก่อน คือ

$\underline{S}_1, \underline{S}_2, \ldots, \underline{S}_n$ = เวกเตอร์ของทิศทางการเคลื่อนที่

$\lambda_1, \lambda_2, \ldots, \lambda_n$ = ระยะทางในการเคลื่อนที่ไปในทิศทาง $\underline{S}_i$, $i = 1,2, \ldots,n$

n = จำนวนตัวแปร

2. เริ่มขั้นตอนแรกโดยการกำหนดจุดเริ่มต้น ($\underline{X}^\circ$) โดยให้ถือว่า จุดเริ่มต้นนั้นเป็นจุดที่ดีที่สุดจุดแรก

3. เริ่มการเคลื่อนที่จากจุดเริ่มต้น ($\underline{X}^\circ$) ไปตามทิศทาง $\underline{S}_1$ ด้วยระยะ λ_1 จะได้จุดใหม่ ($\underline{X}^1$) ซึ่งเขียนเป็นสมการได้ดังนี้

$$\underline{X}^1 = \underline{X}^\circ + \lambda_1 . \underline{S}_1$$

4. เปรียบเทียบค่าออปเจกทีฟฟังก์ชั่น ระหว่างจุดเก่า และจุดใหม่ ถ้า $f(\underline{X}^1) < f(\underline{X}^\circ)$ จะถือว่าเป็นการเคลื่อนที่ที่สำเร็จ ให้จุด $\underline{X}^1$ เป็นจุดที่ดีที่สุดจุดใหม่ และให้คูณ λ_1 ด้วย α ($\alpha > 1$) สำหรับค่า α Rosenbrock แนะนำให้มีค่าเท่ากับ 3 แต่ถ้าในการเปรียบเทียบได้ว่า $f(\underline{X}^1) > f(\underline{X}^\circ)$ จะถือว่าเป็นการเคลื่อนที่ที่ล้มเหลวให้คูณ λ_1 ด้วย $-\beta$ ($0 < \beta < 1$) และจุดที่ดีที่สุดยังเป็นจุดเดิม คือ $\underline{X}^\circ$

5. ให้ดำเนินการตามข้อ 3 และ 4 ในลักษณะคล้ายกันในทิศทางที่เหลือ กล่าวคือ ให้เคลื่อนที่จากจุดที่ดีที่สุดไปตามทิศทาง $\underline{S}_2$ $\underline{S}_3, \ldots, \underline{S}_n$ ด้วยระยะ $\lambda_2, \lambda_3, \ldots, \lambda_n$ ตามลำดับ โดยเคลื่อนที่ทีละทิศทาง

6. ขั้นตอนแรกจะสิ้นสุดเมื่อ การเคลื่อนที่ในทุกทิศทางประสบความสำเร็จอย่างน้อยหนึ่งครั้ง และประสบความล้มเหลวอย่างน้อยหนึ่งครั้ง แล้วคำนวณขั้นตอนที่สองต่อไป โดยทิศทางการเคลื่อนที่จะเปลี่ยนไป และระยะทางการเคลื่อนที่จะต้องกลับไปมีค่าเท่ากับ ค่าที่ได้กำหนดไว้ครั้งแรก สำหรับทิศทางการเคลื่อนที่ชุดใหม่จะหาได้ด้วยวิธี Gram-Schmidt ดังนี้

กำหนดให้ $\underline{P}_1, \underline{P}_2, \ldots, \underline{P}_n$ เป็นเวกเตอร์ของทิศทางอิสระโดย

$$\underset{n\times n}{P} = [P_1, P_2, \ldots, P_n]$$

$$= \underset{n\times n}{[S_1^{(j)}, S_2^{(j)}, \ldots, S_n^{(j)}]} \begin{bmatrix} \lambda_1 & & & \\ \lambda_2 & \lambda_2 & 0 & \\ \lambda_3 & \lambda_3 & \Delta_3 & \\ \vdots & & & \\ \lambda_n & \lambda_n & \lambda_n \ldots & \lambda_n \end{bmatrix}_{n\times n} \quad \text{---- 4.8}$$

โดย Δk คือ ผลรวมของระยะที่มีการเคลื่อนที่สำเร็จ

ในแต่ละทิศทาง $\underline{S}_i$, $i = 1, 2, \ldots, n$

จากสมการที่ 4.8 $\underline{P}_1$ จะเป็นเวกเตอร์ที่ต่อเชื่อมระหว่างจุดเริ่มต้น และจุดสุดท้ายใน สเตจที่ j ส่วน $\underline{P}_2$ จะเป็นผลรวมของระยะทางที่มีการเคลื่อนที่ ที่สำเร็จในทุกทิศทางการเคลื่อนยกเว้นทิศทางแรก ($\underline{S}_1$) ดังนั้น $\underline{P}_1, \underline{P}_2, \ldots, \underline{P}_n$ จะสามารถสร้างทิศทางการเคลื่อนที่ชุดใหม่ได้ด้วยวิธี Gram-Schmidt ดังนี้

$$\underline{S}_i = \frac{\underline{Q}_i}{[Q_i]} \quad ; i = 1,2, \ldots,n$$

โดย

$$Q_1 = \underline{P}_1$$

$$Q_{i+1} = \underline{P}_{i+1} - \sum_{m=1}^{i} [\underline{P}^T_{i+1} \cdot \underline{S}_m^{(j+1)}] \underline{S}_m^{(j+1)}$$

$$i= 0,1,2, \ldots,n-1$$

7. กำหนดให้จุดที่ดีที่สุดที่ได้จากขั้นตอนแรกเป็นจุดที่ดีที่สุดเริ่มต้นของขั้นตอนที่สองต่อไปแล้ว คำนวณซ้ำจากขั้นตอนที่ 1 จนกระทั่งค่าของออปเจกทีฟฟังก์ชั่นมีการเปลี่ยนแปลงน้อยกว่าที่กำหนด ขั้นตอนแผนผังการคำนวณวิธี Rosenbrock แสดงได้ดังรูปที่ 4.4

2. แบบมีสมการเงื่อนไข

เทคนิคของ Rosenbrock ที่ใช้แก้ปัญหา Non-Linear Programming แบบมีสมการเงื่อนไข จะมีลักษณะเป็น Feasible Check เทคนิคนี้จะใช้หาค่าต่ำสุด หรือสูงสุดของออปเจกทีฟฟังก์ชั่นที่เป็นฟังก์ชั่นไม่เป็นเชิงเส้น และมีเงื่อนไขแบบไม่เท่ากับ คือ

หาค่าเหมาะสมที่สุด $F(x_1, x_2, \ldots, x_n)$

โดย $G_k < X_k < H_k \quad , \quad k = 1,2, \ldots, M$

เมื่อ $X_{n+1}, \ldots, X_M$ คือ ฟังก์ชั่นของตัวแปร

$x_1, x_2, \ldots, x_n$

Gk และ Hk คือ ค่าคงที่ หรือฟังก์ชั่นของตัวแปร

$x_1, x_2, \ldots, x_n$

เทคนิค Rosenbroch แบบมีสมการเงื่อนไขนี้จะมีขั้นตอน การคำนวณเหมือนกันแบบไม่มีสมการเงื่อนไข แต่จะมีการตรวจสอบสมการเงื่อนไขและ โซนขอบเขต (Boundary Zone) เพิ่มเติมเข้ามา โดยโซนขอบเขตนี้จะกำหนดให้เท่ากับ

ขอบเขตล่าง $G_k < X_k < (G_k + (H_k - G_k) \cdot 10^{-4})$

ขอบเขตบน $H_k > X_k < (H_k - (H_k - G_k) \cdot 10^{-4})$

เมื่อ $k = 1,2, \ldots, n.$

ก. ขั้นตอนการคำนวณ

เทคนิค Resenbrock แบบมีสมการเงื่อนไขนี้ควรกำหนดจุดเริ่มต้นที่เป็นไปตามเงื่อนไข และไม่อยู่ในโซนขอบเขต สำหรับขั้นตอนการคำนวณนั้นจะเหมือนแบบไม่มีสมการเงื่อนไข นอกจากว่าเมื่อดำเนินการถึงการคำนวณค่า ออปเจกทีฟฟังก์ชั่น ให้ดำเนินการต่อไปดังนี้

1. กำหนดให้ F° เป็นค่าออปเจกทีฟฟังก์ชั่นที่ดีที่สุดในขณะนี้ และจุดที่ให้ค่าออปเจกทีฟฟังก์ชั่นนี้ จะต้องให้สมการเงื่อนไขเป็นจริง และให้ F^{*} เป็นค่าออปเจกทีฟฟังก์ชั่นที่ดีที่สุดในขณะนี้ และจุดที่ให้ค่านี้จะต้องให้สมการเงื่อนไขเป็นจริง และไม่ละเมิดโซนขอบเขต ซึ่งทั้ง F° และ F^{*} จะมีเท่ากันที่จุดเริ่มต้น

2. ถ้าค่าออปเจกทีฟฟังก์ชั่นที่คำนวณได้ F มีค่าที่ไม่ดีกว่า F° (ให้ค่าสูงกว่าในกรณีหาค่าต่ำสุด) หรือ สมการเงื่อนไขไม่เป็นจริง จะถือว่าเป็นการเคลื่อนที่ที่ล้มเหลว และดำเนินการตามเทคนิคแบบไม่มีสมการเงื่อนไขต่อไป

3. ถ้าจุดที่ให้ค่าออปเจกทีฟฟังก์ชั่น F นี้ให้ค่าสมการเงื่อนไขที่อยู่ในโซนขอบเขต ออปเจกทีฟฟังก์ชั่นจะถูกเปลี่ยนแปลงดังนี้

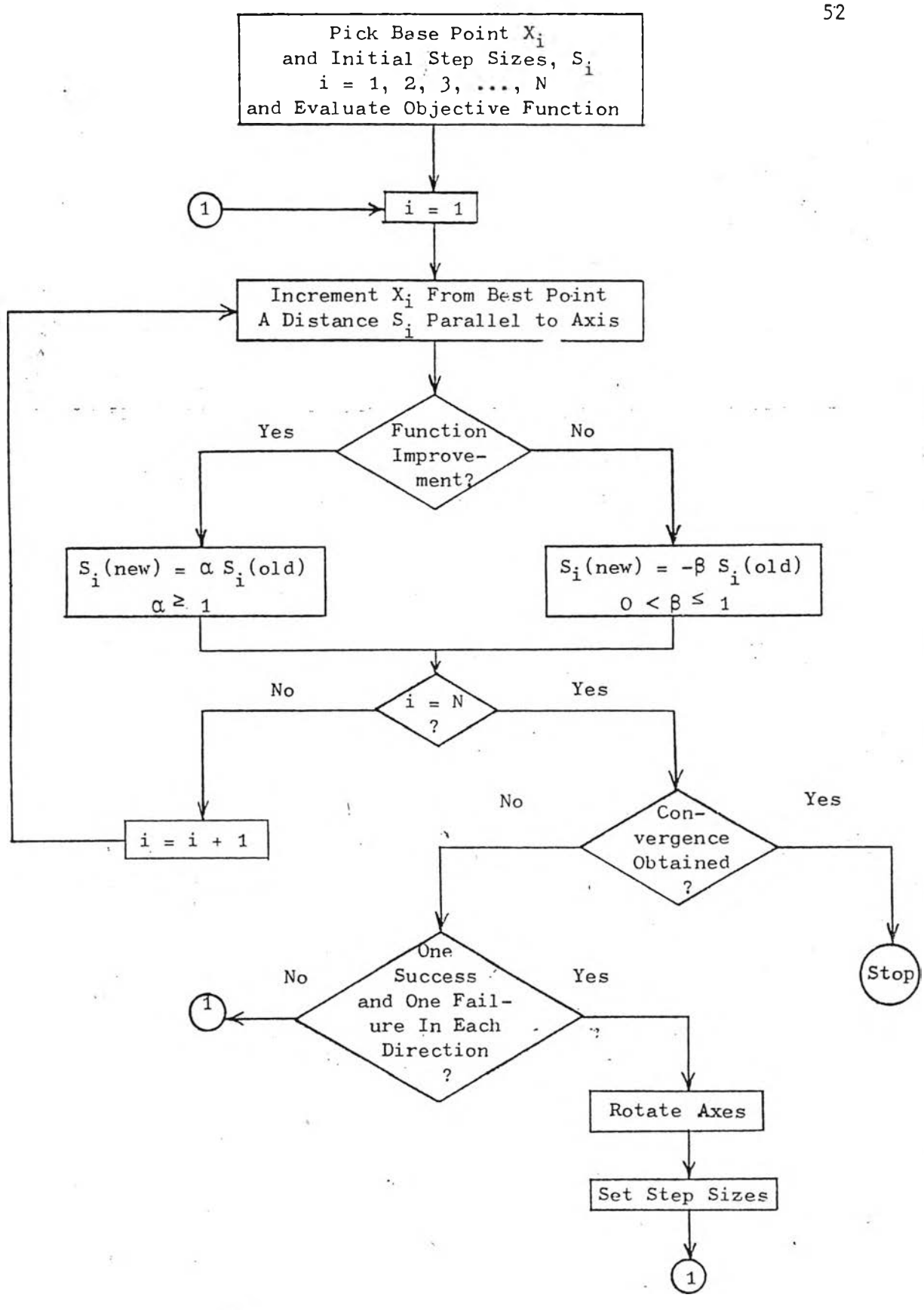
$$F(new) = F(old) - (F(old) - F^{*})\ (3.\Lambda - 4.\Lambda^{3} + 2.\Lambda^{3})$$

เมื่อ

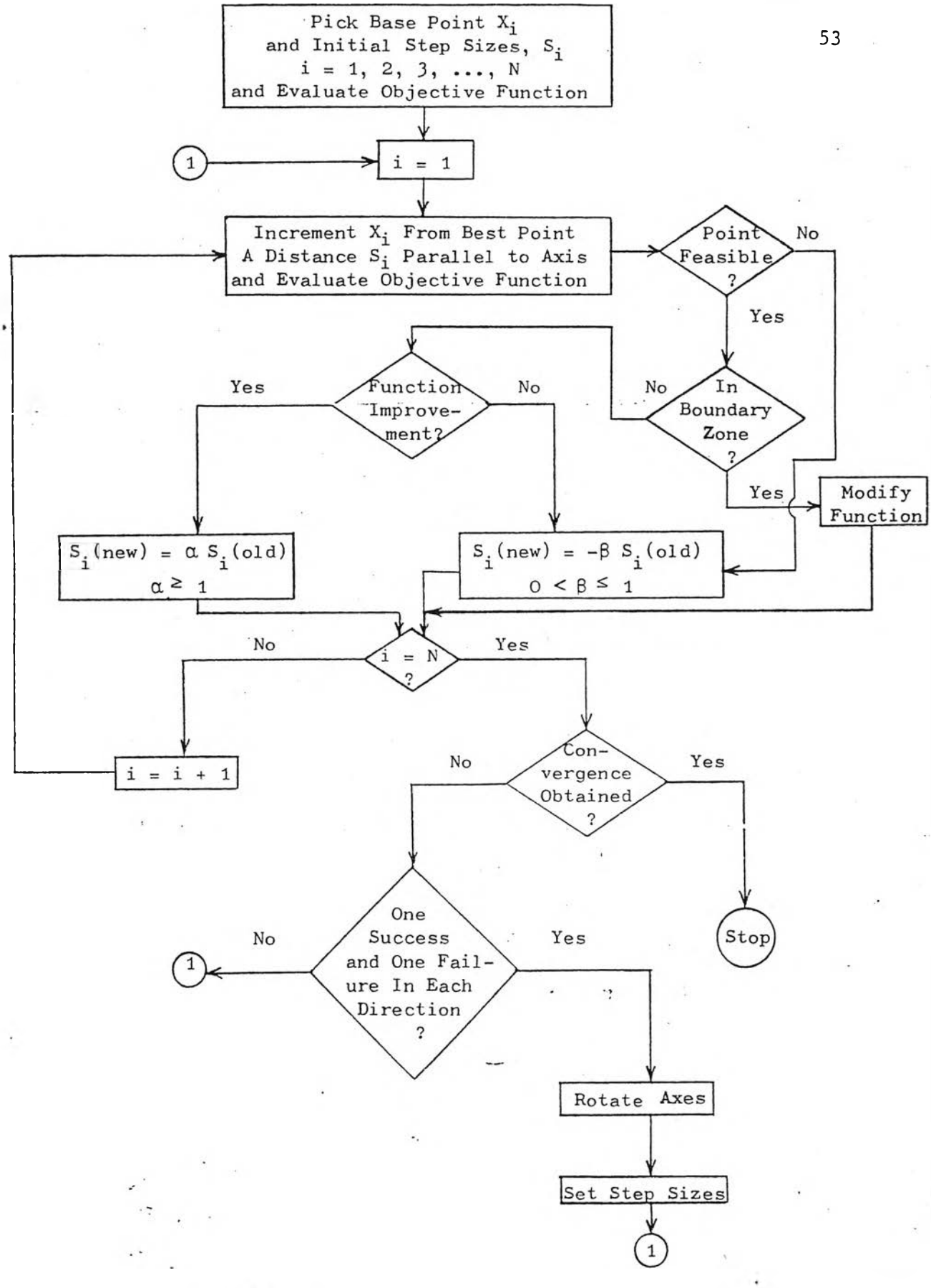
$$\Lambda = \frac{G_k + (H_k - G_k).\ 10^{-\Delta} - X_k}{(H_k - G_k)\ .\ 10^{-\Delta}} \qquad \text{กรณีขอบเขตล่าง}$$

$$= \frac{X_k - (H_k - ((H_k - G_k).10^{-\Delta}}{(H_k - G_k).10^{-\Delta}} \qquad \text{กรณีขอบเขตบน}$$

4. ถ้าค่าออปเจกทีฟฟังก์ชั่นได้จาก จุดที่ไม่ละเมิดโซนขอบเขต และสมการเงื่อนไข กำหนดให้ F^{*} มีค่าเท่ากับ F° แล้วคำนวณขั้นตอนต่อไปจนกระทั่ง การเปลี่ยนแปลงค่าออปเจกทีฟฟังก์ชั่นน้อยกว่าที่กำหนด รูปที่ 4.5 แสดงขั้นตอนการคำนวณเทคนิค Rosenbrock แบบมีสมการเงื่อนไข



รูปที่ 4.4 แสดงแผนผังการคำนวณด้วยเทคนิคของ Rosenbrock แบบไม่มีสมการเงื่อนไข



รูปที่ 4.5 แสดงแผนผังการคำนวณด้วยเทคนิคของ
วิธี Rosenbrock แบบมีสมการเงื่อนไข