บทที่ 3

การออปติไมซ์เซชันแบบหลายเป้าหมาย
(Multiobjective Optimization)

3.1 บทนำ

ปัญหาออปติไมซ์เซชัน(Optimization Problem) เป็นปัญหาที่เกี่ยวกับการหาค่าสูงสุดหรือค่าต่ำสุด ของฟังก์ชันของตัวแปรตั้งแต่หนึ่งตัวขึ้นไปโดยที่ตัวแปร (หรือฟังก์ชัน) เหล่านี้ต้องเป็นไปตามเงื่อนไข (Constraint) ที่กำหนด เทคนิคของการทำออปติไมซ์เซชันเหล่านี้เป็นที่รู้จักกันมานานแล้วกว่า 150ปี [12]และได้ถูกนำไปใช้ในการหาคำตอบในปัญหาทางวิทยาศาสตร์และทางวิศวกรรมศาสตร์จำนวนมากและให้ผลเป็นที่น่าพอใจ ตลอดจนได้มีการพัฒนาวิธีใหม่ๆ ขึ้นมาหลายวิธี แต่ในวิทยานิพนธ์ดังกล่าว เราจะกล่าวถึงการแก้ปัญหาโดยใช้เทคนิคการทำออปติไมซ์เซชันแบบหลายเป้าหมาย(Multiobjective Optimization)

กล่าวโดยทั่วไปแล้วการทำออปติไมซ์ ก็คือการแบ่งสรรทรัพยากรที่มีอยู่อย่างจำกัดอย่างดีที่สุดเพื่อให้เป็นไปตามจุดประสงค์ที่ตั้งไว้ เนื่องจากเราสามารถจัดสรรทรัพยากรต่างๆ ได้แตกต่างกันการจัดสรรคำตอบที่เป็นไปตามเงื่อนไขต่างๆ อีกทั้งให้ได้ผลตามจุดประสงค์ที่เหมาะสมที่สุดนั้นจะเรียกว่าคำตอบเหมาะที่สุด (Optimal Solution)

การศึกษาการจ่ายโหลดอย่างประหยัดที่คำนึงถึงข้อจำกัดของการปล่อยก๊าซพิษจากโรงไฟฟ้า อาศัยเทคนิคการทำออปติไมซ์เซชันแบบหลายเป้าหมาย มาช่วยหาผลลัพธ์ หรือค่าที่เหมาะสมที่สุดซึ่งหมายถึงผลลัพธ์ที่มีค่าใช้จ่ายประหยัดที่สุดและปล่อยก๊าซพิษน้อยที่สุด ซึ่งขั้นตอนในการคำนวณออปติไมซ์เซชันแบบหลายเป้าหมาย [12] มีรายละเอียดดังต่อไปนี้

3.2 แบบจำลองของการออปติไมซ์เซชันแบบหลายเป้าหมาย (Multiobjective Optimization Model)

การออปติไมซ์เซชันแบบหลายเป้าหมาย หรือ Vector Minimization เป็นการโปรแกรมทางคณิตศาสตร์ที่ใช้ในการแก้ปัญหาที่มีหลายเป้าหมาย เพื่อใช้ในการหาคำตอบที่เหมาะสมที่สุด

แบบจำลองทางคณิตศาสตร์ของการออปติไมซ์เซชันแบบหลายเป้าหมาย มีโครงสร้างดังนี้

1. มีฟังก์ชันเป้าหมาย (Objective function) คือ การแสดงความสัมพันธ์ของตัวแปรเป็นสมการหรือฟังก์ชันที่จะทำการออปติไมซ์ เพื่อให้ได้ คำตอบสูงสุดหรือต่ำสุด (maximize or minimize)

2. มีสมการแสดงเงื่อนไข (Constraints) ซึ่งแสดงถึงเงื่อนไขบังคับในการออปติไมซ์ แบ่งออกได้เป็น 2 ชนิดคือ เงื่อนไขบังคับแบบอสมการ (Inequality constraint) และเงื่อนไขบังคับแบบสมการ (Equality constraint)

3. ตัวแปรทุกตัวต้องมีค่ามากกว่าหรือเท่ากับศูนย์ (All positive value)

ปัญหาในการออปติไมซ์เซชันแบบหลายเป้าหมายนั้น ประกอบด้วย จำนวนตัวแปรในการตัดสินใจ n ค่า (n decision variables) , m เงื่อนไข (m constraints) และ p เป้าหมาย (p objectives) โดย

สมการเป้าหมาย คือ

$$\begin{aligned}\text{minimize } \mathbf{Z}(x_1,x_2,\ldots,x_n) &= [Z_1(x_1,x_2,\ldots,x_n), Z_2(x_1,x_2,\ldots,x_n),\ldots,\\ &\quad Z_p(x_1,x_2,\ldots,x_n)] \end{aligned} \tag{3.1}$$

สมการหรืออสมการเงื่อนไข :

$$g_i(x_1,x_2,\ldots,x_n) \le 0 \tag{3.2}$$

$$\text{หรือ } g_i(x_1,x_2,\ldots,x_n) \ge 0$$

$$\text{หรือ } g_i(x_1,x_2,\ldots,x_n) = 0 \;;\; i = 1,2,\ldots,m$$

$$x_j \ge 0 \;;\; j = 1,2,\ldots,n$$

โดย $\mathbf{Z}(x_1, x_2, \ldots, x_n)$: ฟังก์ชันหลายเป้าหมาย

$Z_1(\,), Z_2(\,), \ldots, Z_p(\,)$: เป็นสมการเป้าหมาย p สมการ

x_j : เป็นค่าตัวแปรที่แทนในสมการเป้าหมาย

เรามีตัวแปรที่จะทำการพิจารณาอยู่ n ตัว การเพิ่มค่าตัวแปรตัวหนึ่งตัวใดมีผลทำให้ตัวแปรตัวอื่นๆ ที่เกี่ยวข้องกันลดค่าลงไปด้วยภายใต้ขอบข่ายที่กำหนดเป็นสมการหรืออสมการ โดยเครื่องหมายทางคณิตศาสตร์คือ =(เท่ากับ) , ≤ (น้อยกว่าหรือเท่ากับ) และ ≥ (มากกว่าหรือเท่ากับ)

3.3 การหาผลลัพธ์ของการออปติไมซ์เซชันแบบหลายเป้าหมาย

รูปแบบแทนระบบปัญหาของการออปติไมซ์เซชันแบบหลายเป้าหมาย สามารถเขียนได้ดังนี้

สมการเป้าหมาย : Max. (Min) $Z = Z_1, Z_2, \ldots, Z_p$

$$= C_{1i}X_i, C_{2i}X_2, \ldots, C_{pi}X_i$$

สมการหรืออสมการเงื่อนไข $a_{ij}x_j \quad (\leq, =, \geq) \quad b_i$

$$\vdots \qquad \vdots \qquad \vdots$$

$$a_{mj}x_j \quad (\leq, =, \geq) \quad b_i$$

และ $x_j \geq 0$

คำตอบสำหรับปัญหาแบบการออปติไมซ์เซชันแบบหลายเป้าหมาย :

คำตอบ(noninferior solution) คือ $\mathbf{X} = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_{n-1} \\ x_n \end{bmatrix}$

ในการหาคำตอบของ การออปติไมซ์เซชันแบบหลายเป้าหมาย โดยทั่วไปมีอยู่ 4 วิธี คือ

1). วิธีการ weighting

2). วิธีการ constraint

3). วิธีการ non-inferior set-estimation(NISE)

4). วิธีการ multiple-objective simplex

สำหรับวิธีที่ 1-3 นั้น จะทำการแปลงปัญหาแบบหลายเป้าหมาย ให้เป็น แบบเป้าหมายเดียว ก่อน จากนั้นจึงนำวิธีการออปติไมซ์ เช่น การโปรแกรมเชิงเส้นตรง โดยวิธีซิมเพล็กซ์ (simplex method) มาแก้ปัญหา ส่วนวิธีที่ 4 จะไม่มีการแปลงปัญหาให้เป็นแบบเป้าหมายเดียว แต่จะเป็นการแก้ปัญหาแบบหลายเป้าหมายโดยตรง ซึ่งจะยากและซับซ้อน รายละเอียดของแต่ละวิธีการมีดังต่อไปนี้

3.4 วิธีการ Weighting

วิธีการ weighting [12] เป็นวิธีที่จะแปลงปัญหาแบบหลายเป้าหมาย ให้เป็นแบบเป้าหมายเดียว โดยการคำนวณค่าน้อยที่สุดของปัญหา (minimization) แบบ 2 ฟังก์ชันเป้าหมาย สามารถแสดงให้อยู่ในรูปสมการได้ดังนี้

$$\text{minimize} \quad \mathbf{Z} = [Z_1, Z_2] \qquad (3.3)$$

จากปัญหาแบบ 2 เป้าหมายนั้นเราสามารถเปลี่ยนเป็นปัญหาที่มีเป้าหมายเดียว ได้โดยอาศัยการถ่วงน้ำหนัก(weight) ให้แก่ Z_1 และ Z_2

$$\text{minimize} \quad \mathbf{Z}(w_1, w_2) = w_1 Z_1 + w_2 Z_2 \qquad (3.4)$$

โดย w_1 และ w_2 คือค่าถ่วงน้ำหนักของ Z_1 และ Z_2 ตามลำดับ

จากหลักการที่ว่า เมื่อทำการหารฟังก์ชันเป้าหมายด้วยเลขจำนวนจริงบวกใด ๆ แล้วคำตอบไม่เปลี่ยน ดังนั้นจะเอา w_1 หารตลอดสมการ (3.4) จะได้

$$\text{minimize} \quad \mathbf{Z}(w_1, w_2) = Z_1 + (w_2 / w_1) Z_2 \qquad (3.5)$$

แทน $(w_2 / w_1) = w$ ดังนั้นสมการทั่วไป จะเป็น

$$\text{minimize} \quad \mathbf{Z}(w_1, w_2) = Z_1 + w Z_2 \qquad (3.6)$$

จะเห็นว่า เราสามารถกำหนดค่าถ่วงน้ำหนัก(weight) สำหรับฟังก์ชันเป้าหมายหนึ่งๆ เป็น 1 หรือเลขจำนวนจริงบวกใด ๆ

<u>ขั้นตอนในการทำวิธีการ weighting</u>

1). ทำการออปติไมซ์แต่ละ ฟังก์ชันเป้าหมาย คือ Z_1 1 ครั้ง โดย $wZ_2 = 0$ และ ออปติไมซ์ เฉพาะ Z_2 โดย $Z_1 = 0$ เพื่อหาช่วงคำตอบ

2).หลังจากนั้นทำการออปติไมซ์ฟังก์ชันเป้าหมาย (Z) ซึ่งได้ ถ่วงน้ำหนัก Z_2 โดยใช้ค่าถ่วงน้ำหนักที่ต่างๆ กัน

ในการทำออปติไมซ์แต่ละฟังก์ชันเป้าหมายหรือฟังก์ชันหลายเป้าหมายที่ได้ใช้ค่าถ่วงน้ำหนักมาแล้วนั้น ปัญหาทั้งหมดที่กล่าวมาจะกลายเป็นปัญหาแบบเป้าหมายเดียว ดังนั้นในการหาคำตอบสำหรับปัญหาแบบนี้จะใช้วิธีการโปรแกรมเชิงเส้น (Linear programming) เนื่องจากฟังก์ชันเป้าหมายและ สมการเงื่อนไขเป็นฟังก์ชันแบบเชิงเส้น

3.5 วิธีการ Constraint

วิธีการ constraint [12] ก็เป็นอีกวิธีหนึ่งที่ทำการแปลงปัญหาแบบหลายเป้าหมาย ให้เป็นแบบเป้าหมายเดียว โดยวิธีนี้จะทำการออปติไมซ์ฟังก์ชันเป้าหมายเพียง 1 เป้าหมาย ส่วนฟังก์ชันเป้าหมายที่เหลือจะให้มีค่าเท่ากับค่าหนึ่งๆ

<u>รูปแบบทางคณิตศาสตร์ของวิธีการ constraint</u>

ปัญหาหลายเป้าหมาย ซึ่งมี p เป้าหมาย

$$\text{minimize} \quad \mathbf{Z}(x_1, x_2, \ldots, x_n)$$

$$= [Z_1(x_1, x_2, \ldots, x_n), Z_2(x_1, x_2, \ldots, x_n), \ldots, Z_p(x_1, x_2, \ldots, x_n)]$$

โดย $(x_1, x_2, \ldots, x_n) \in F_d$

F_d : เซตของเงื่อนไขบังคับ (constraint set)

<u>ปัญหาแบบ constraint</u>

$$\text{minimize} \quad \mathbf{Z}_h(x_1, x_2, \ldots, x_n) \tag{3.7}$$

โดย $(x_1, x_2, \ldots, x_n) \in F_d$ (3.8)

$$\mathbf{z}_k(x_1, x_2, \ldots, x_n) \geq L_k$$

$$k=1,2,\ldots,h-1,h+1,\ldots,p \qquad (3.9)$$

L_k : เงื่อนไขบังคับด้านขวาของฟังก์ชันเป้าหมาย k

<u>ขั้นตอนในการทำวิธีการ constraint</u>

<u>ขั้นตอนที่1</u>

สร้าง payoff table

1). หาคำตอบของแต่ละ ฟังก์ชันเป้าหมาย สำหรับ p ฟังก์ชันเป้าหมาย

$$x^k = (x^k{}_1, x^k{}_2, \ldots, x^k{}_n)$$

2). คำนวณหาค่าของแต่ละฟังก์ชันเป้าหมาย สำหรับแต่ละ p คำตอบ (p optimal solution) :

$Z_1(x^k), Z_2(x^k), \ldots, Z_p(x^k)$, $k=1,2,\ldots,p$ ซึ่งจะได้ค่าทั้งหมด p ค่า สำหรับแต่ละ pฟังก์ชันเป้าหมาย

3). ทำเป็นตาราง โดยให้แถว(row) เป็นค่า $x^1, x^2, \ldots, x^p$ และคอลัมน์(column) เป็นของฟังก์ชันเป้าหมาย

4). ในแต่ละ k_{th} คอลัมน์ ให้ค่ามากที่สุด = M_k และค่าน้อยที่สุด = n_k โดย $k=1,2,\ldots,p$

Payoff Table สำหรับปัญหาที่มี p ฟังก์ชันเป้าหมาย

	$Z_1(x^k)$	$Z_2(x^k)$	...	$Zp(x^k)$
x^1	$Z_1(x^1)$	$Z_2(x^1)$	...	$Zp(x^1)$
x^2	$Z_1(x^2)$	$Z_2(x^2)$	...	$Zp(x^2)$
.	.	.		.
.	.	.		.
.	.	.		.
x^p	$Z_1(x^p)$	$Z_2(x^p)$	...	$Zp(x^p)$

ขั้นตอนที่2

แปลงปัญหาแบบหลายเป้าหมาย ให้เป็นปัญหาแบบ constraint ดังเช่น (3.7)-(3.9)

ขั้นตอนที่3

ค่า n_k และ M_k ที่ได้จาก ขั้นตอนที่1 จะแทนช่วง $n_k \leq Z_k \leq M_k$

ขั้นตอนที่4

หาคำตอบของปัญหาแบบ constraint ซึ่งจะได้จากขั้นตอนที่2 สำหรับทุกค่าของ L_k , $k=1,2,...,h-1,h+1,...,p$ เมื่อ

$L_k = n_k + [t/(r-1)]*(M_k-n_k)$ $\quad ,t=0,1,2,...,(r-1)$

โดย r : จำนวนค่าความแตกต่างของ L_k

3.6 วิธีการ Noninferior Set Estimation (NISE)

วิธีการ NISE [12] เป็นอีกวิธีหนึ่งที่ทำการแปลงปัญหาแบบหลายเป้าหมายให้เป็นแบบเป้าหมายเดียว โดยวิธีนี้จะทำการออปติไมซ์แต่ละฟังก์ชันเป้าหมาย เพื่อหาจุดที่เป็นคำตอบ(noninferior extreme point) และหาคุณสมบัติของเส้นระหว่างจุด extreme point

ขั้นตอนในการทำวิธีการ NISE (สำหรับ 2 ฟังก์ชันเป้าหมาย)

ขั้นตอนที่1

ทำการออปติไมซ์ แต่ละฟังก์ชันเป้าหมาย (Z_1,Z_2) จะได้คำตอบ 2 ค่า (Noninferior extreme point) คือ P_1 และ P_2 ตามลำดับ และให้ $S_1=P_2$, $S_2=P_1$ และ n=2 คำนวณ ค่า γ_{12}

ขั้นตอนที่2S_i

ถ้า $\gamma_{i,i+1} \leq T$ สำหรับ i= 1,2,...,n-1 ให้หยุด

โดย $\gamma_{i,i+1}$: ค่าความยาวของเส้นระหว่างขอบเขตล่างS_i และขอบเขตบนS_{i+1}

T : ค่าความผิดพลาดสูงสุดที่จะยอมรับได้

n : จำนวนคำตอบ

(ถ้าการประมาณค่ามีความแม่นยำเพียงพอ ซึ่งจะประกอบด้วย S_i , i = 1,2,...,n และหาค่าเส้นที่เชื่อมระหว่างจุดทั้งหลาย) มิฉะนั้นให้เริ่มขั้นตอนที่3

<u>ขั้นตอนที่3</u>

หาค่า $\gamma_{i,i+1}$, i= 1,2,....,n-1 สำหรับค่าที่มากที่สุด สำหรับค่าที่ i และ i+1 ซึ่งจะให้ค่าความคลาดเคลื่อนมากที่สุดที่จะยอมรับได้ ใช้สมการดังนี้

$$\text{minimize } Z(x_1,\ldots,x_n ; i+1)$$

$$= [Z_2(S_i) - Z_2(S_{i+1})]Z_1(x_1,\ldots,x_n)$$

$$+ [Z_1(S_{i+1}) - Z_1(S_i)]Z_2(x_1,\ldots,x_n)$$

ซึ่งค่า ฟังก์ชัน ข้างบนนี้ จะเป็นสมการของเส้นที่เชื่อมระหว่าง S_i และ S_{i+1}

หาค่า $B_{i,i+1}$ (ค่าของ $Z(x_1,\ldots,x_n ; i,i+1)$ ที่จุด S_i และ S_{i+1} ก่อนที่จะหาคำตอบของฟังก์ชันเป้าหมายที่ S_i หรือ S_{i+1} ถ้าคำตอบของฟังก์ชันเป้าหมาย $Z\ (x_1,\ldots,x_n;i,i+1) = B_{i,i+1}$ ให้ $\gamma_{i,i+1} = 0$ และกลับไปที่ขั้นตอนที่2 ถ้า $Z(x_1,\ldots,x_n;i,i+1) > B_{i,i+1}$ ให้หาคำตอบใหม่ (noninferior solution) ใหม่ (P_{n+1}) จากนั้นไปขั้นตอนที่ 4

<u>ขั้นตอนที่4</u>

จัดลำดับ P_t ,t =1,2,....,n+1 ค่า i จะเป็นค่ามากที่สุดของ t เช่น $Z_2(S_t) \geq Z_2(P_{n+1})$ ในการหาคำตอบใหม่ จะได้ Z_2 ค่าใหม่ ซึ่งจะเป็นค่ามากที่สุดถัดไปจาก $Z_2(S_i)$ ซึ่งเราสามารถใช้รูปแบบ ได้ดังนี้

$$S'_t = S_t \qquad t=1,2,\ldots,i$$

$$S'_{i+1} = P_{n+1}$$

$$S'_{t+1} = S_t \qquad t=i+1,\ldots,n$$

γ' terms :

$$\gamma'_{t,t+1} = \gamma_{t,t+1} \qquad t=1,2,\ldots,i-1 \quad (\text{ถ้า } i \geq 1)$$

$$\gamma'_{t+1,t+2} = \gamma_{t,t+1} \qquad t=i+1,\ldots,n-1 \quad (\text{ถ้า } i \leq n-2)$$

คำนวณ $\gamma'_{i,i+1}$ และ $\gamma'_{i+1,i+2}$ โดยเพิ่มค่า n ขึ้นอีก 1 และกลับไปที่ขั้นตอนที่2

3.7 วิธีการ Multiobjective Simplex

วิธีการ Multiobjective Simplex [12] เป็นวิธีที่ไม่ต้องมีการแปลงปัญหาแบบหลายเป้าหมายให้เป็นแบบเป้าหมายเดียว โดยการแก้ปัญหาวิธีนี้จะทำที่ฟังก์ชันหลายเป้าหมายโดยตรง ซึ่งวิธีนี้จะยุ่งยากและซับซ้อน

สำหรับปัญหาแบบหลายเป้าหมาย ที่มี m เงื่อนไขบังคับ , p ฟังก์ชันเป้าหมาย ซึ่งจะใช้ multiobjective simplex tableau ดังนี้

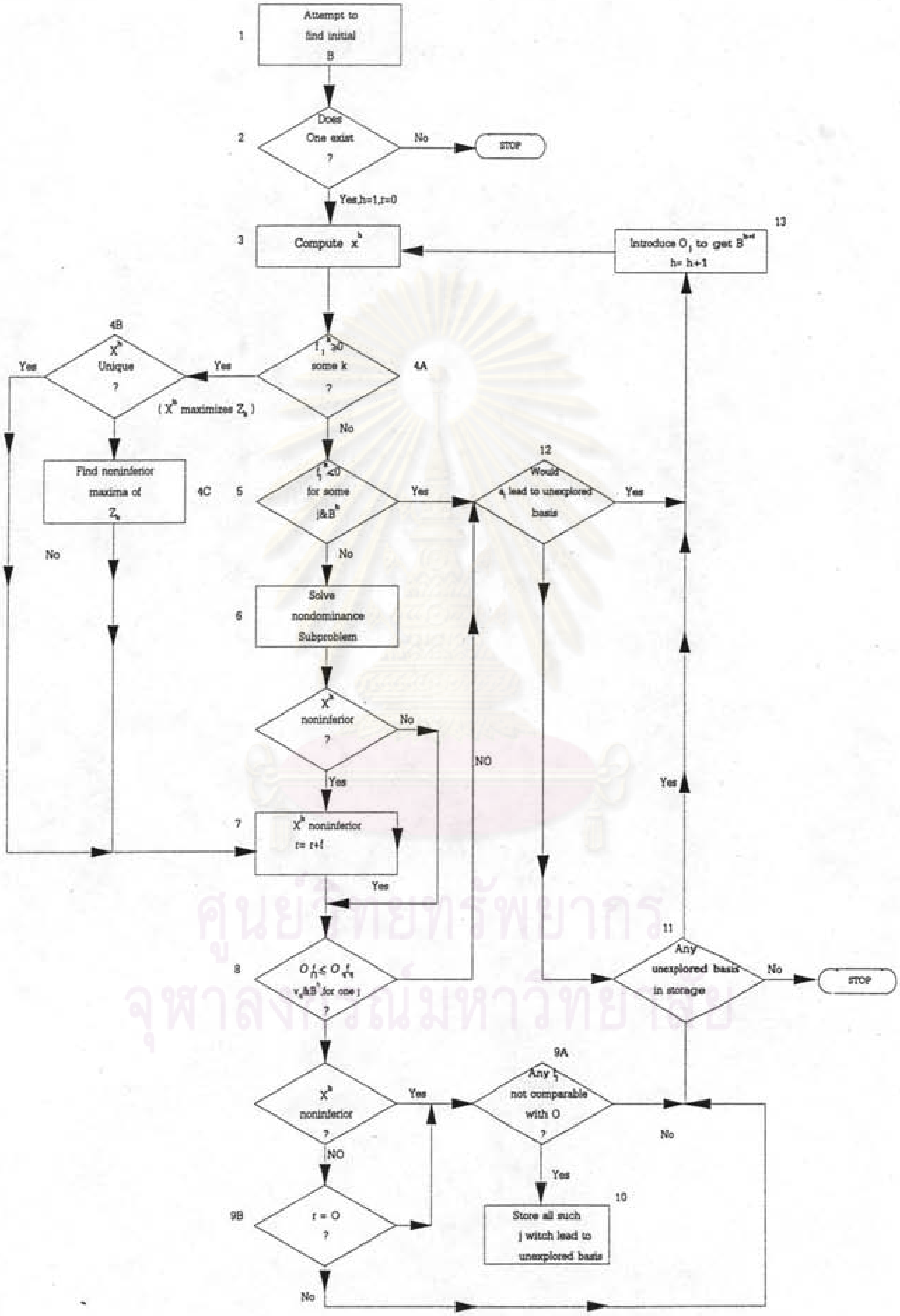
Partial Multiobjective Simplex Tableau สำหรับปัญหาทั่วไป

	a_1	a_2	...	a_n	a_{n+1}	...	a_{n+m}	a_0
c^1_j	c^1_1	c^1_2	...	c^1_n	0	...	0	0
c^2_j	c^2_1	c^2_2	...	c^2_n	0	...	0	0
.								
.								
c^p_j	c^p_1	c^p_2	...	c^p_n	0	...	0	0
Basic columns								
f^1_j	f^1_1	f^1_2	...	f^1_n	f^1_{n+1}	...	f^1_{n+m}	$f^1_0 = Z_1$
f^2_j	f^2_1	f^2_2	...	f^2_n	f^2_{n+1}	...	f^2_{n+m}	$f^2_0 = Z_2$
.								
.								
f^p_j	f^p_1	f^p_2	...	f^p_n	f^p_{n+1}	...	f^p_{n+m}	$f^p_0 = Z_p$

สำหรับขั้นตอนวิธีการคำนวน Multiobjective Simplex ได้แสดงไว้ในรูปที่ 3.1

จากที่กล่าวมาทั้งหมด 4 วิธี สำหรับในวิทยานิพนธ์นี้จะเลือกใช้วิธี weighting ด้วยเหตุผลที่ว่า

1. วิธีนี้จะให้คำตอบ (noninferior solution) เสมอสำหรับปัญหาแบบ convex
2. สำหรับค่าถ่วงน้ำหนัก (weight) ที่แต่ละฟังก์ชันเป้าหมายจะให้คำตอบที่เหมาะสมและสามารถเพิ่มความสำคัญของฟังก์ชันเป้าหมายโดยเพิ่มหรือลดค่าถ่วงน้ำหนักที่ฟังก์ชันเป้าหมายนั้น



รูปที่ 3.1 แสดงขั้นตอนวิธีของ MULTIOBJECTIVE SIMPLEX METHOD