บทที่ 3
ขั้นตอนวิธีใหม่

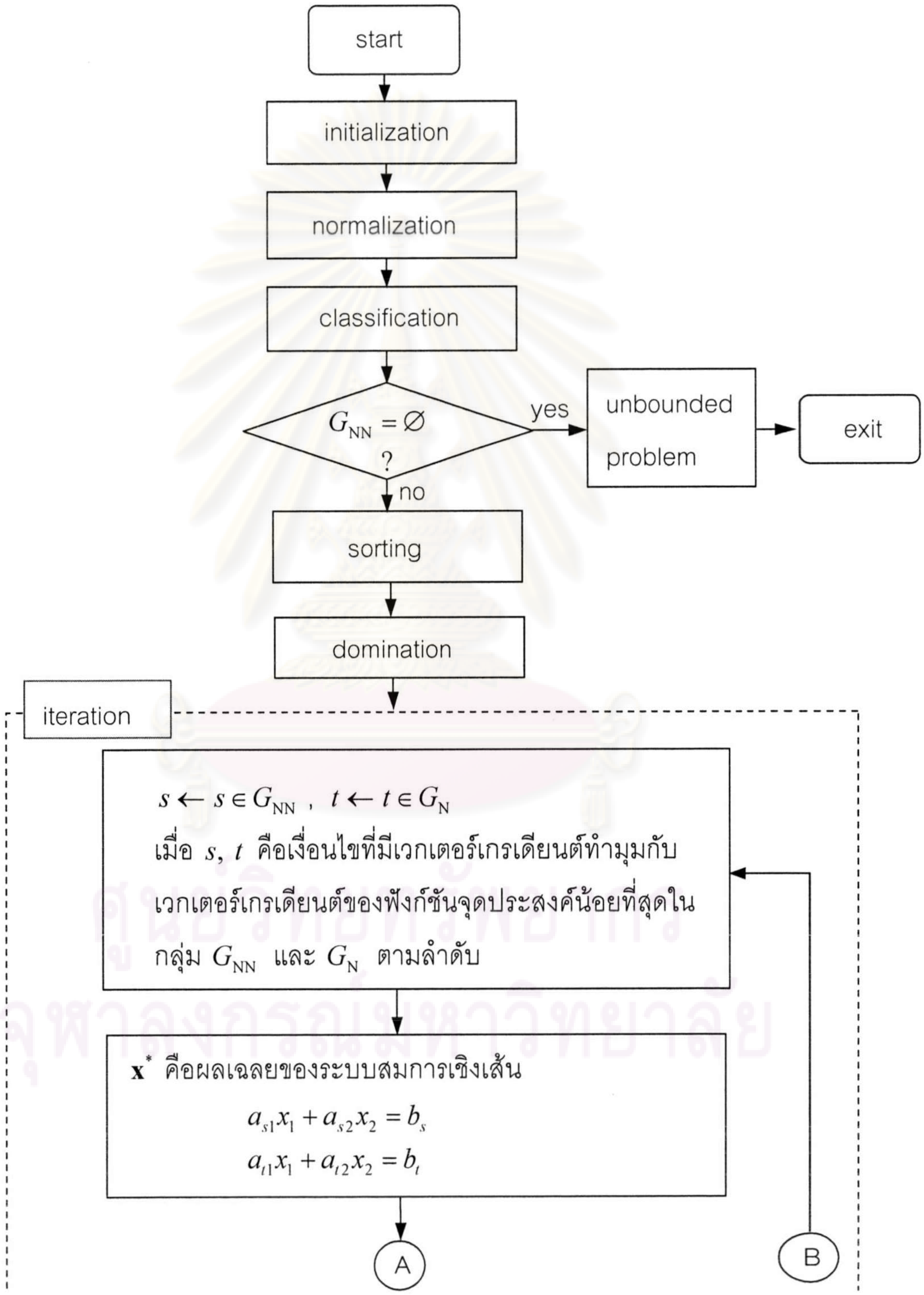
ในบทนี้จะกล่าวถึงขั้นตอนวิธีที่พัฒนาขึ้นมาเพื่อใช้หาผลเฉลยของปัญหากำหนดการเชิงเส้นที่อยู่ในรูปแบบที่ 1.1 เรียกขั้นตอนดังกล่าวว่าขั้นตอนวิธีใหม่ (novel algorithm) เนื่องจากว่าปัญหากำหนดการเชิงเส้นที่กำหนดให้เป็นปัญหาที่มีบริเวณที่เป็นไปได้เสมอทำให้ได้ว่าปัญหาดังกล่าวอาจจะเป็นปัญหาที่มีผลเฉลยที่เหมาะที่สุดเสมอ หรืออาจเป็นปัญหาที่ไม่มีขอบเขต จากทฤษฎีบท 2.1 ทำให้เราสามารถตรวจสอบได้ว่าปัญหาที่กำหนดให้เป็นปัญหาที่มีผลเฉลยที่เหมาะที่สุดเสมอหรือไม่ ในขณะที่บทแทรก 2.1 ทำให้เราสามารถตรวจสอบได้ว่าปัญหาที่กำหนดให้เป็นปัญหาที่ไม่มีขอบเขตหรือไม่

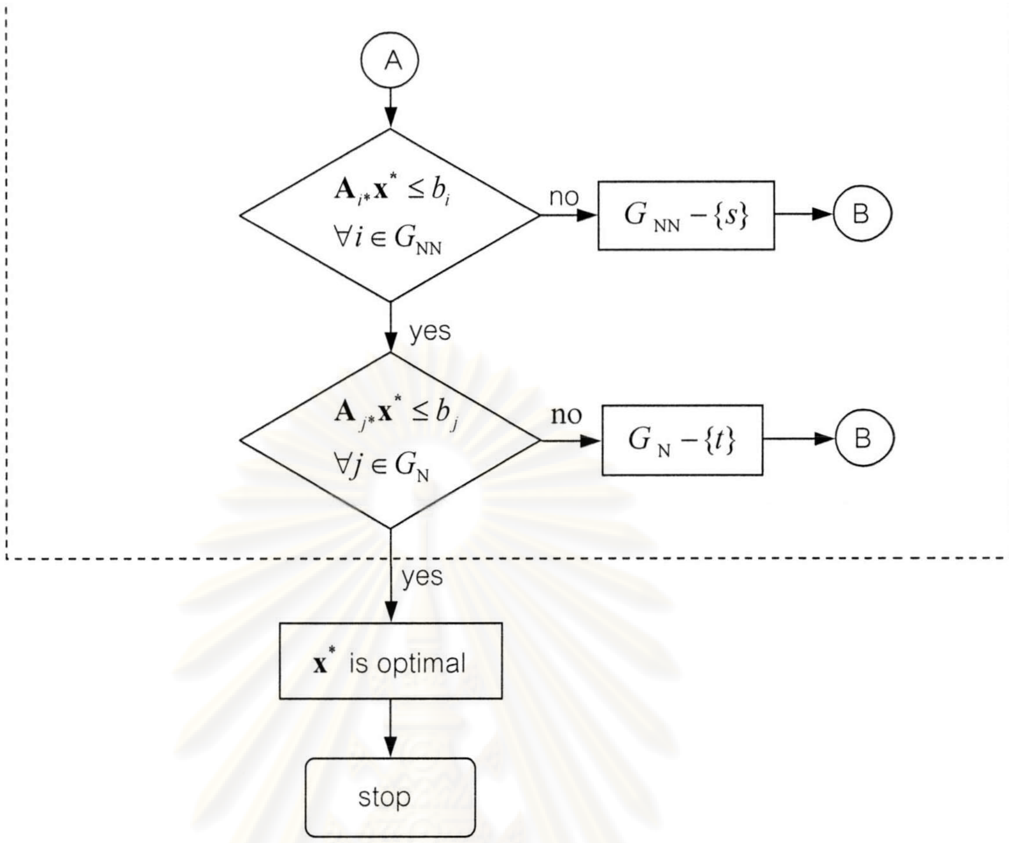
ขั้นตอนวิธีใหม่ประกอบไปด้วยขั้นตอนย่อยดังต่อไปนี้ 1.) normalization เป็นขั้นตอนที่ทำให้ b_i ซึ่งเป็นส่วนประกอบของเวกเตอร์ด้านขวามือ ($\mathbf{b}$) มีค่าเท่ากับ 1 ถ้า $b_i > 0$ หรือเท่ากับ -1 ถ้า $b_i < 0$ ด้วยการหารตลอดเงื่อนไขบังคับที่ i ด้วย $|b_i|$ 2.) classification เป็นขั้นตอนการแบ่งกลุ่มของเงื่อนไขบังคับออกเป็นสองกลุ่ม โดยการอาศัยผลคูณเชิงสเกลาร์ของเวกเตอร์เกรเดียนต์ของเงื่อนไขบังคับกับเวกเตอร์ $\mathbf{u}$ ซึ่งเป็นเวกเตอร์ที่ตั้งฉากกับเวกเตอร์เกรเดียนต์ของฟังก์ชันจุดประสงค์ 3.) sorting เป็นขั้นตอนการจัดเรียงเงื่อนไขบังคับในแต่ละกลุ่มที่ได้จากขั้นตอนที่ 2 ในการจัดเรียงได้อาศัยมุมระหว่างเวกเตอร์เกรเดียนต์ของเงื่อนไขบังคับกับเวกเตอร์ เกรเดียนต์ของฟังก์ชันจุดประสงค์ โดยจัดเรียงข้อมูลจากเงื่อนไขบังคับที่มีเวกเตอร์เกรเดียนต์ทำมุมน้อยที่สุดกับเวกเตอร์เกรเดียนต์ของฟังก์ชันจุดประสงค์ ไปยังเงื่อนไขบังคับที่มีเวกเตอร์ เกรเดียนต์ทำมุมมากที่สุดกับเวกเตอร์เกรเดียนต์ของฟังก์ชันจุดประสงค์สำหรับแต่ละกลุ่ม 4.) domination ขั้นตอนนี้จะลดทอน redundant constraint ที่มีเวกเตอร์เกรเดียนต์ขนานกันและมีทิศทางเดียวกัน โดยใช้ทฤษฎีบท 2.2 เพื่อบอกว่าเงื่อนไขใดสามารถกำจัดออกไปได้โดยที่ไม่มีผลกระทบต่อผลเฉลยของปัญหาเดิม แต่ถึงอย่างไรก็ตามการลดทอนโดยใช้ทฤษฎีบท 2.2 อาจจะไม่ได้ทำให้ redundant constraint หมดไป ขั้นตอนสุดท้ายคือขั้นตอนการกระทำซ้ำเพื่อหาผลเฉลยของปัญหาโดยอาศัยข้อมูลจากสมาชิกในสองกลุ่มดังกล่าวที่ผ่านการลดทอน redundant constraint ลงแล้ว ถึงแม้ว่าจะยังคงมี redundant constraint เหลือแต่ redundant constraint ที่เหลือดังกล่าวไม่มีผลต่อผลเฉลยของปัญหา

3.1 ขั้นตอนวิธีใหม่

ฟังก์ชันหลัก (main function)

ฟังก์ชันหลักของขั้นตอนวิธีใหม่ จะเป็นการเตรียมข้อมูลเบื้องต้น(initialization) ของปัญหา และเรียกใช้ฟังก์ชันย่อย ซึ่งมีแผนภาพการทำงานดังต่อไปนี้





รูปที่ 3.1

ขั้นตอน normalization

กำหนดให้ $\mathbf{A}$ คือเมทริกซ์สัมประสิทธิ์ที่มีมิติ $m \times 2$ และ $\mathbf{b}$ คือเวกเตอร์ด้านขวามือที่มีขนาด $m \times 1$ ของปัญหากำหนดการเชิงเส้น เมื่อ m คือจำนวนเงื่อนไขบังคับ และ $n = 2$ คือมิติของปัญหา การทำงานของขั้นตอนนี้แสดงดังฟังก์ชันย่อย Normalize() ต่อไปนี้

Normalize()

1 กระทำซ้ำขั้นตอนที่ 2 ถึง 4 สำหรับ $i = 1, 2, \cdots, m$

2 ถ้า $b_i \ne 0$ แล้ว

3 $\mathbf{A}_{i*} \leftarrow \mathbf{A}_{i*} / |b_i|$

4 $b_i \leftarrow b_i / |b_i|$

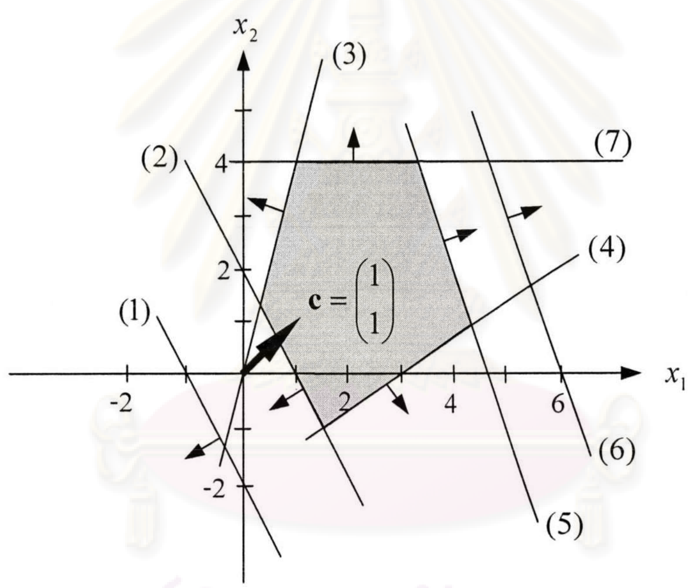
ตัวอย่าง 3.1

พิจารณาปัญหากำหนดการเชิงเส้นใน 2 มิติที่กำหนดดังต่อไปนี้

Maximize $x_1 + x_2$

subject to

$$
\begin{array}{rcrcrl}
-2x_1 & - & x_2 & \le & 2 & \quad ....(1) \\
-2x_1 & - & x_2 & \le & -2 & \quad ....(2) \\
-4x_1 & + & x_2 & \le & 0 & \quad ....(3) \\
2x_1 & - & 3x_2 & \le & 6 & \quad ....(4) \\
3x_1 & + & x_2 & \le & 14 & \quad ....(5) \\
3x_1 & + & x_2 & \le & 18 & \quad ....(6) \\
 & & x_2 & \le & 4 & \quad ....(7)
\end{array}
$$



รูปที่ 3.2

จะได้เมทริกซ์สัมประสิทธิ์ **A** และเวกเตอร์ด้านขวามือ **b** ดังนี้

$$
\mathbf{A} = \begin{pmatrix} -2 & -1 \\ -2 & -1 \\ -4 & 1 \\ 2 & -3 \\ 3 & 1 \\ 3 & 1 \\ 0 & 1 \end{pmatrix} \qquad \mathbf{b} = \begin{pmatrix} 2 \\ -2 \\ 0 \\ 6 \\ 14 \\ 18 \\ 4 \end{pmatrix}
$$

เมื่อใช้ขั้นตอนวิธีใหม่แก้ปัญหานี้ โดยการส่งค่าของเมทริกซ์ $\mathbf{A}$ และเวกเตอร์ $\mathbf{b}$ เข้าไปในขั้นตอน normalization จะได้เมทริกซ์ $\mathbf{A}$ และเวกเตอร์ $\mathbf{b}$ ที่ผ่านการ normalize ดังต่อไปนี้

$$\mathbf{A} = \begin{pmatrix} -2/2 & -1/2 \\ -2/2 & -1/2 \\ -4 & 1 \\ 2/6 & -3/6 \\ 3/14 & 1/14 \\ 3/18 & 1/18 \\ 0 & 1/4 \end{pmatrix} = \begin{pmatrix} -1 & -1/2 \\ -1 & -1/2 \\ -4 & 1 \\ 1/3 & -1/2 \\ 3/14 & 1/14 \\ 1/6 & 1/18 \\ 0 & 1/4 \end{pmatrix}$$

และ

$$\mathbf{b} = \begin{pmatrix} 2/2 \\ -2/2 \\ 0 \\ 6/6 \\ 14/114 \\ 18/18 \\ 4/4 \end{pmatrix} = \begin{pmatrix} 1 \\ -1 \\ 0 \\ 1 \\ 1 \\ 1 \\ 1 \end{pmatrix}$$

ขั้นตอน classification

จากขั้นตอน normalization เราจะได้เมทริกซ์ $\mathbf{A}$ และเวกเตอร์ $\mathbf{b}$ ที่ผ่านการ normalize แล้ว หลังจากนั้นเราจะส่งเมทริกซ์ $\mathbf{A}$ และเวกเตอร์ $\mathbf{b}$ ดังกล่าว พร้อมด้วยเวกเตอร์ $\mathbf{c} = \begin{pmatrix} c_1 \\ c_2 \end{pmatrix}$ ซึ่งเป็นเวกเตอร์เกรเดียนต์ของฟังก์ชันจุดประสงค์ เข้าไปในขั้นตอน classification

กำหนดเวกเตอร์ $\mathbf{u}$ ดังต่อไปนี้ $\mathbf{u} = \begin{pmatrix} -c_2 \\ c_1 \end{pmatrix}$

พิจารณาส่วนประกอบของเวกเตอร์ $\mathbf{c}$ ซึ่งเป็นค่าสัมประสิทธิ์ของตัวแปร x_1 และ x_2 ที่ปรากฏในฟังก์ชันจุดประสงค์ ถ้า $c_1 = c_2 = 0$ แล้วจะทำให้ค่าของฟังก์ชันจุดประสงค์ (z) เท่ากับศูนย์เสมอ หมายความว่าทุกผลเฉลยที่เป็นไปได้เป็นผลเฉลยที่เหมาะที่สุด ดังนั้นจะพิจารณาเฉพาะกรณีที่

$\mathbf{c} \neq \mathbf{0}$ เท่านั้นซึ่งทำให้ $\mathbf{u} \neq \mathbf{0}$ ด้วย และเมื่อพิจารณาความสัมพันธ์ระหว่างเวกเตอร์ $\mathbf{c}$ และ $\mathbf{u}$ จากผลคูณเชิงสเกลาร์ของ $\mathbf{c}$ และ $\mathbf{u}$ จะได้ดังนี้

$$
\begin{aligned}
\mathrm{c} \cdot \mathrm{u} = \| \mathrm{c} \| \ \| \mathrm{u} \| \cos\theta &= -c_1 c_2 + c_1 c_2 \\
&= 0
\end{aligned}
$$

ทำให้ได้ว่า $\theta = \dfrac{\pi}{2}$ ซึ่งเป็นการแสดงว่าเวกเตอร์ $\mathbf{c}$ ตั้งฉากกับเวกเตอร์ $\mathbf{u}$ เสมอ

ขั้นตอน classification เรียกใช้ฟังก์ชันย่อย Classify() ดังต่อไปนี้

Classify()

1 $\mathbf{u} \leftarrow \begin{pmatrix} -c_2 & c_1 \end{pmatrix}^{\mathrm{T}}$

2 $countG_{\mathrm{N}} \leftarrow 0$ และ $countG_{\mathrm{NN}} \leftarrow 0$

3 กระทำซ้ำขั้นตอนที่ 4 ถึง 9 สำหรับ $i = 1, 2, \cdots, m$

4   ถ้า $\mathbf{A}_{i*} \cdot \mathbf{u} < 0$ แล้ว

5         $countG_{\mathrm{N}} \leftarrow countG_{\mathrm{N}} + 1$

6         $G_{\mathrm{N}}\left[countG_{\mathrm{N}}\right] \leftarrow i$

7   มิฉะนั้น

8         $countG_{\mathrm{NN}} \leftarrow countG_{\mathrm{NN}} + 1$

9         $G_{\mathrm{NN}}\left[countG_{\mathrm{NN}}\right] \leftarrow i$

คำสั่งในบรรทัดที่ 1 คือการกำหนดค่าของเวกเตอร์ $\mathbf{u}$ เท่ากับ $\begin{pmatrix} -c_2 & c_1 \end{pmatrix}^{\mathrm{T}}$

คำสั่งในบรรทัดที่ 2 คือการกำหนดค่าของตัวแปร $countG_{\mathrm{N}}$ และ $countG_{\mathrm{NN}}$ ให้เท่ากับ 0

คำสั่งในบรรทัดที่ 3 คือการสั่งให้กระทำซ้ำคำสั่งในบรรทัดที่ 4 ถึง บรรทัดที่ 9 สำหรับ $i = 1, 2, \cdots, m$

คำสั่งในบรรทัดที่ 4 ถึงบรรทัดที่ 9 เป็นการตรวจสอบว่า ถ้าผลคูณเชิงสเกลาร์ดังกล่าวมีค่าน้อยกว่าศูนย์ แล้วจะกำหนดให้เงื่อนไขบังคับที่ i เป็นสมาชิกของกลุ่ม G_{N} แต่ถ้ามีค่ามากกว่าหรือเท่ากับศูนย์ แล้วจะกำหนดให้เป็นสมาชิกของกลุ่ม G_{NN}

หมายเหตุ: $countG_{\mathrm{N}} + countG_{\mathrm{NN}} = m$ เมื่อ $countG_{\mathrm{N}}, countG_{\mathrm{NN}}$ คือขนาดของกลุ่ม G_{N} และ G_{NN} ตามลำดับ และ m เท่ากับจำนวนเงื่อนไขบังคับของปัญหากำหนดการเชิงเส้น

ตัวอย่าง 3.2

จากตัวอย่าง 3.1 เมื่อนำเมทริกซ์ $\mathbf{A}$ และเวกเตอร์ $\mathbf{b}$ ที่ผ่านการ normalize แล้วพร้อมด้วยเวกเตอร์เกรเดียนต์ของฟังก์ชันจุดประสงค์ ($\mathbf{c} = (1 \quad 1)^{\mathrm{T}}$) เข้าไปในขั้นตอน classification เพื่อจัดกลุ่มของเงื่อนไขบังคับ กำหนดให้ $\mathbf{u} = (-1 \quad 1)^{\mathrm{T}}$ จะได้ว่า

$$\mathbf{A} = \begin{pmatrix} -1 & -1/2 \\ -1 & -1/2 \\ -4 & 1 \\ 1/3 & -1/2 \\ 3/14 & 1/14 \\ 1/6 & 1/18 \\ 0 & 1/4 \end{pmatrix} = \begin{pmatrix} \mathbf{A}_{1*} \\ \mathbf{A}_{2*} \\ \mathbf{A}_{3*} \\ \mathbf{A}_{4*} \\ \mathbf{A}_{5*} \\ \mathbf{A}_{6*} \\ \mathbf{A}_{7*} \end{pmatrix}$$

เงื่อนไขที่ i	$\mathbf{A}_{i*}$	$\mathbf{A}_{i*} \cdot \mathbf{u}$	เป็นสมาชิกของกลุ่ม G_{N} หรือ G_{NN}
1	$(-1 \quad -1/2)$	$1/2$	G_{NN}
2	$(-1 \quad -1/2)$	$1/2$	G_{NN}
3	$(-4 \quad 1)$	5	G_{NN}
4	$(1/3 \quad -1/2)$	$-5/6$	G_{N}
5	$(3/14 \quad 1/14)$	$-1/7$	G_{N}
6	$(1/6 \quad 1/18)$	$-1/9$	G_{N}
7	$(0 \quad 1/4)$	$1/4$	G_{NN}

ตาราง 3.1 การแบ่งกลุ่มเงื่อนไขบังคับออกเป็น 2 กลุ่มคือ G_{N} และ G_{NN}

จากตารางที่ 3.1 จะได้ว่าเงื่อนไขลำดับที่ 1, 2, 3 และ 7 อยู่ในกลุ่ม G_{NN} และเพื่อความสะดวกในการกล่าวอ้างจึงกำหนดให้ G_{NN} เก็บข้อมูลเป็นแบบแถวลำดับ (array) ในหนึ่งมิติ ดังนั้นจะได้ว่า

ตำแหน่งของข้อมูล $\rightarrow$	1	2	3	4
$G_{\mathrm{NN}} =$	1	2	3	7

สำหรับการกล่าวอ้างถึงข้อมูลสามารถกระทำได้ดังเช่น $G_{\mathrm{NN}}[1]=1$ หมายถึงสมาชิกในตำแหน่งที่ 1 ของ G_{NN} มีค่าเท่ากับ 1 (เงื่อนไขบังคับลำดับที่หนึ่งของปัญหากำหนดการเชิงเส้น) หรือ $G_{\mathrm{NN}}[4]=7$ หมายถึงสมาชิกในตำแหน่งที่ 4 ของ G_{NN} มีค่าเท่ากับ 7 (เงื่อนไขบังคับลำดับที่เจ็ดของปัญหากำหนดการเชิงเส้น)

ในลักษณะเดียวกันจะได้ว่าเงื่อนไขบังคับลำดับที่ 4, 5 และ 6 อยู่ในกลุ่ม G_{N} ซึ่งกำหนดเป็นแบบแถวลำดับดังต่อไปนี้

ตำแหน่งของข้อมูล $\rightarrow$	1	2	3
$G_{\mathrm{N}}=$	4	5	6

ขั้นตอน sorting

กำหนดให้ $\boldsymbol{\alpha}$ และ $\boldsymbol{\beta}$ เป็นเวกเตอร์ที่มีขนาดเท่ากับค่าของตัวแปร $countG_{\mathrm{N}}$ และ $countG_{\mathrm{NN}}$ ตามลำดับ และกำหนดดังต่อไปนี้

$$\boldsymbol{\alpha}=\begin{pmatrix}\alpha_1 & \alpha_2 & \cdots & \alpha_i & \cdots & \alpha_{countG_{\mathrm{N}}}\end{pmatrix}^{\mathrm{T}} \quad \text{และ} \quad \boldsymbol{\beta}=\begin{pmatrix}\beta_1 & \beta_2 & \cdots & \beta_i & \cdots & \beta_{countG_{\mathrm{NN}}}\end{pmatrix}^{\mathrm{T}}$$

เมื่อ $$\alpha_i=\cos^{-1}\left(\frac{\mathbf{A}_{k*}\cdot\mathbf{c}}{\|\mathbf{c}\|\|\mathbf{A}_{k*}\|}\right) \quad ; \quad k=G_{\mathrm{N}}[i] \qquad \text{.......................................(3.1)}$$

และ $$\beta_i=\cos^{-1}\left(\frac{\mathbf{A}_{l*}\cdot\mathbf{c}}{\|\mathbf{c}\|\|\mathbf{A}_{l*}\|}\right) \quad ; \quad l=G_{\mathrm{NN}}[i] \qquad \text{.......................................(3.2)}$$

ในการจัดเรียงข้อมูลเราใช้การจัดเรียงแบบเร็ว (quick sort) โดยเรียกผ่านชุดคำสั่งย่อยที่ชื่อ QuickSort(T,first,last) เมื่อ T คือ G_{N} และ G_{NN} ในขณะที่ first และ last หมายถึงตำแหน่งของข้อมูลตัวแรกและตัวสุดท้ายของ T จะเห็นว่าการจัดเรียงเป็นการจัดเรียงบน G_{N} และ G_{NN} แต่ค่าที่นำมาพิจารณาเปรียบเทียบในการจัดเรียงจะเป็นค่ามุมที่อยู่ในเวกเตอร์ $\boldsymbol{\alpha}$ และ $\boldsymbol{\beta}$ ซึ่งสัมพันธ์กับ G_{N} และ G_{NN} ตามลำดับ โดยเรียงลำดับจากเงื่อนไขบังคับที่มีค่ามุมน้อยที่สุดไปยังเงื่อนไขบังคับที่มีค่ามุมมากที่สุด จากขั้นตอนการเรียงลำดับ ทำให้ได้ว่า

$$\alpha_i \le \alpha_{i+1} \;;\; i = 1,2,\cdots,countG_{\mathrm{N}} - 1$$

และ

$$\beta_i \le \beta_{i+1} \;;\; i = 1,2,\cdots,countG_{\mathrm{NN}} - 1$$

หรือกล่าวอีกนัยหนึ่งคือ มุมระหว่างเวกเตอร์เกรเดียนต์ของฟังก์ชันจุดประสงค์ กับเวกเตอร์เกรเดียนต์ของเงื่อนไขบังคับที่ $G_{\mathrm{N}}[i]$ มีค่าน้อยกว่าหรือเท่ากับมุมระหว่างเวกเตอร์เกรเดียนต์ของฟังก์ชันจุดประสงค์กับเวกเตอร์เกรเดียนต์ของเงื่อนไขบังคับที่ $G_{\mathrm{N}}[i+1]$ เสมอสำหรับ $i = 1,2,\cdots,countG_{\mathrm{N}} - 1$ และในทำนองเดียวกันมุมระหว่างเวกเตอร์เกรเดียนต์ของฟังก์ชันจุดประสงค์กับเวกเตอร์เกรเดียนต์ของเงื่อนไขบังคับที่ $G_{\mathrm{NN}}[i]$ มีค่าน้อยกว่าหรือเท่ากับมุมระหว่างเวกเตอร์เกรเดียนต์ของฟังก์ชันจุดประสงค์กับเวกเตอร์เกรเดียนต์ของเงื่อนไขบังคับที่ $G_{\mathrm{NN}}[i+1]$ เสมอสำหรับ $i = 1,2,\cdots,countG_{\mathrm{NN}} - 1$

```
QuickSort(T, first, last)
ถ้า first < last แล้ว
      pivot ← Partition(T, first, last)
      QuickSort(T, first, pivot − 1)
      QuickSort(T, pivot + 1 ,last)
```

```
Partition(T, first, last)
pivot ← T[first]
pivotPos ← first
สำหรับ next = first +1 ถึง last
      ถ้า T[next] < pivot แล้ว
            pivotPos ← pivotPos + 1
            T[pivotPos] ↔ T[next]
T[pivotPos] ↔ T[first]
return pivotPos
```

จากตัวอย่างที่ 3.1 จะได้เวกเตอร์ $\boldsymbol{\alpha}$ และ $\boldsymbol{\beta}$ ดังนี้

$$\boldsymbol{\alpha} = (2.306 \quad 2.306 \quad 1.943 \quad 1.047)^{\mathrm{T}} \quad \text{และ} \quad \boldsymbol{\beta} = (1.710 \quad 0.886 \quad 0.886)^{\mathrm{T}}$$

เมื่อ $\alpha_i = \cos^{-1}\left(\dfrac{\mathbf{A}_{k*} \cdot \mathbf{c}}{\| \mathbf{c} \| \| \mathbf{A}_{k*} \|}\right) \quad ; \quad k = G_{\mathrm{N}}[i] = 1, 2, 3, 7$

และ $\beta_i = \cos^{-1}\left(\dfrac{\mathbf{A}_{l*} \cdot \mathbf{c}}{\| \mathbf{c} \| \| \mathbf{A}_{l*} \|}\right) \quad ; \quad l = G_{\mathrm{NN}}[i] = 4, 5, 6$

ขั้นตอน domination

ขั้นตอน domination จะลดทอน redundant constraint ลงโดยอาศัยทฤษฎีบท 2.2 กล่าวคือ ถ้ามีเงื่อนไข 2 เงื่อนไขที่มีเวกเตอร์เกรเดียนต์ขนานกันและทิศทางเดียวกันสมมุติว่าเป็นเงื่อนไขที่ i และ j แล้วจะลดทอนเงื่อนไขที่ j ถ้า $H_i^- \subseteq H_j^-$ มิฉะนั้นจะลดทอนเงื่อนไขที่ i

รูปแบบของการเรียกใช้ฟังก์ชันย่อย Dominate() ในการลดทอนเงื่อนไขที่เป็น redundant constraint ในกลุ่ม G_{N} คือ Dominate(G_{N} , $\boldsymbol{\alpha}$, $\mathbf{b}$) และในทำนองเดียวกันสามารถเรียกใช้ฟังก์ชันย่อย Dominate() เพื่อลดทอนเงื่อนไขที่เป็น redundant constraint ในกลุ่ม G_{NN} คือ Dominate(G_{NN} , $\boldsymbol{\beta}$, $\mathbf{b}$)

Dominate(G , θ , $\mathbf{b}$)

1 size_G $\leftarrow$ **size_of** G

2 $loop \leftarrow 0$ และ $i \leftarrow 1$

3 ถ้า size_G $\neq 0$ แล้ว

4 กระทำซ้ำขั้นตอนที่ 5 ถึง 38 เมื่อ $i \leq$ size_G

5 $loop \leftarrow loop + 1$

6 $count \leftarrow 0$

7 $fix \leftarrow i$

8 $k \leftarrow G[i]$

9 $MaxNorm \leftarrow \| \mathbf{A}_{k*} \|$

10 สำหรับ $j = i + 1$ ถึง size_G กระทำซ้ำขั้นตอนที่ 11 ถึง 36

11 $s \leftarrow G[j]$

12 ถ้า $\theta_i = \theta_j$ แล้ว

13 $count \leftarrow count + 1$

14 ถ้า $b_i b_j > 0$ แล้ว

15 ถ้า $b_j > 0$ แล้ว

16 ถ้า $\| \mathbf{A}_{s*} \| > MaxNorm$ แล้ว

17 $MaxNorm \leftarrow \| \mathbf{A}_{s*} \|$

18 $fix \leftarrow j$

19 มิฉะนั้น

20 ถ้า $\| \mathbf{A}_{s*} \| < MaxNorm$ แล้ว

21 $MaxNorm \leftarrow \| \mathbf{A}_{s*} \|$

22 $fix \leftarrow j$

23 มิฉะนั้น ถ้า $b_i b_j < 0$ แล้ว

24 ถ้า $b_j < 0$ แล้ว

25 $MaxNorm \leftarrow \| \mathbf{A}_{s*} \|$

26 $fix \leftarrow j$

27 มิฉะนั้น

28 $MaxNorm \leftarrow \| \mathbf{A}_{k*} \|$

29 $fix \leftarrow i$

30 มิฉะนั้น ถ้า $b_i > 0$ แล้ว

31 $MaxNorm \leftarrow \| \mathbf{A}_{s*} \|$

32 $fix \leftarrow j$

33 มิฉะนั้น $b_i < 0$ แล้ว

34 $MaxNorm \leftarrow \| \mathbf{A}_{k*} \|$

35 $fix \leftarrow i$

36 มิฉะนั้น ไปยังขั้นตอนที่ 37

37 $G[loop] \leftarrow G[fix]$

38 $i \leftarrow i + count + 1$

คำสั่งในบรรทัดที่ 1 คือการกำหนดให้ตัวแปร size_G มีค่าเท่ากับขนาดของกลุ่ม G

คำสั่งในบรรทัดที่ 2 คือการกำหนดค่าให้ตัวแปร $loop$ และ i ให้มีค่าเท่ากับ 0 และ 1 ตามลำดับ

คำสั่งในบรรทัดที่ 3 เป็นการตรวจสอบเงื่อนไขการทำงาน ถ้า size_G$\neq 0$ แล้วให้ทำคำสั่งในบรรทัดที่ 4

คำสั่งในบรรทัดที่ 4 คือคำสั่งในการวนซ้ำคำสั่งในบรรทัดที่ 5 ถึง 38 ถ้า $i \leq$ size_G

คำสั่งในบรรทัดที่ 5 คือการกำหนดค่าให้ตัวแปร $loop$ เท่ากับ $loop+1$

คำสั่งในบรรทัดที่ 6 คือการกำหนดค่าให้ตัวแปร $count$ เท่ากับ 0

คำสั่งในบรรทัดที่ 7 คือการกำหนดค่าให้ตัวแปร fix เท่ากับ i

คำสั่งในบรรทัดที่ 8 คือการกำหนดค่าให้ตัวแปร k เท่ากับ $G[i]$ ซึ่งเป็นหมายเลขเงื่อนไขบังคับในลำดับที่ i ของ G

คำสั่งในบรรทัดที่ 9 คือการกำหนดค่าให้ตัวแปร $MaxNorm$ เท่ากับ $\|\mathbf{A}_{k*}\|$ ซึ่งเป็น Euclidean norm ของเวกเตอร์ $\mathbf{A}_{k*}$

คำสั่งในบรรทัด 10 คือการวนซ้ำคำสั่งในบรรทัดที่ 11 ถึงบรรทัดที่ 36

คำสั่งในบรรทัดที่ 11 คือการกำหนดค่าให้ตัวแปร s เท่ากับ $G[j]$ ซึ่งเป็นหมายเลขเงื่อนไขบังคับในลำดับที่ j ของ G

คำสั่งในบรรทัดที่ 12 คือการตรวจสอบว่าเวกเตอร์เกรเดียนต์ของเงื่อนไขบังคับที่ k และเวกเตอร์เกรเดียนต์ของเงื่อนไขบังคับที่ s มีทิศทางเดียวกันหรือไม่ โดยพิจารณาจากมุม θ_i และ มุม θ_j ถ้า $\theta_i = \theta_j$ แสดงว่ามีทิศทางเดียวกัน แล้วให้ทำคำสั่งในบรรทัดที่ 13 ถึงบรรทัดที่ 35 มิฉะนั้นให้ทำคำสั่งบรรทัดที่ 37

คำสั่งบรรทัดที่ 13 คือการกำหนดค่าให้ตัวแปร $count$ เท่ากับ $count+1$

คำสั่งในบรรทัดที่ 14 คือการตรวจสอบว่าเวกเตอร์ด้านขวามือของเงื่อนไขบังคับที่ k และ s มีค่าเครื่องหมายเหมือนกันหรือไม่ โดยพิจารณาจาก b_i และ b_j ถ้า $b_i b_j > 0$ แสดงว่ามีเครื่องหมายเหมือนกัน ให้ทำคำสั่งในบรรทัดที่ 15 ถึง บรรทัดที่ 22 มิฉะนั้นให้ทำคำสั่งในบรรทัดที่ 23 ถึงบรรทัด 35

คำสั่งในบรรทัดที่ 15 คือการตรวจสอบว่าเวกเตอร์ด้านขวามือของเงื่อนไขบังคับที่ k และ s มีค่าเครื่องหมายบวกเหมือนกันหรือไม่ โดยพิจารณาจาก $b_j > 0$ ถ้า $b_j > 0$ ($b_i > 0$ โดยปริยาย) ให้ทำคำสั่งในบรรทัดที่ 16 ถึงบรรทัดที่ 18

คำสั่งในบรรทัดที่ 16 ถึงบรรทัดที่ 18 คือการตรวจสอบว่า norm ของ $\mathbf{A}_{s*}$ มีค่ามากกว่า $MaxNorm$ หรือไม่ ถ้ามากกว่าแล้วให้ตัวแปร $MaxNorm$ เท่ากับ $\|\mathbf{A}_{s*}\|$ และให้ตัวแปร fix เท่ากับ j

คำสั่งในบรรทัดที่ 20 ถึงบรรทัดที่ 22 จะเรียกใช้เมื่อผลจากบรรทัดที่ 15 ไม่จริง คือการตรวจสอบว่า norm ของ $\mathbf{A}_{s*}$ มีค่าน้อยกว่า $MaxNorm$ หรือไม่ ถ้าน้อยกว่าให้ตัวแปร $MaxNorm$ เท่ากับ $\| \mathbf{A}_{s*} \|$ และให้ตัวแปร fix เท่ากับ j

คำสั่งในบรรทัดที่ 23 คือการตรวจสอบว่าเวกเตอร์ด้านขวามือของเงื่อนไขบังคับที่ k และ s มีค่าเครื่องหมายตรงข้ามกันหรือไม่ โดยพิจารณาจาก b_i และ b_j ถ้า $b_i b_j < 0$ แสดงว่ามีเครื่องหมายตรงข้ามกัน ให้ทำคำสั่งในบรรทัดที่ 24 ถึงบรรทัดที่ 35 มิฉะนั้นให้ทำคำสั่งในบรรทัดที่ 36

คำสั่งในบรรทัดที่ 24 ถึงบรรทัดที่ 29 คือการตรวจสอบว่าถ้า $b_j < 0$ แล้วให้ $MaxNorm$ มีค่าเท่ากับ $\| \mathbf{A}_{s*} \|$ และให้ตัวแปร fix มีค่าเท่ากับ j มิฉะนั้นให้ $MaxNorm$ มีค่าเท่ากับ $\| \mathbf{A}_{k*} \|$ และให้ตัวแปร fix มีค่าเท่ากับ i

คำสั่งในบรรทัดที่ 30 ถึงบรรทัดที่ 35 คือการตรวจสอบว่าถ้า $b_i > 0$ แล้วให้ $MaxNorm$ มีค่าเท่ากับ $\| \mathbf{A}_{s*} \|$ และให้ตัวแปร fix มีค่าเท่ากับ j มิฉะนั้นให้ $MaxNorm$ มีค่าเท่ากับ $\| \mathbf{A}_{k*} \|$ และให้ตัวแปร fix มีค่าเท่ากับ i

คำสั่งในบรรทัดที่ 36 คือการสั่งให้ทำคำสั่งในบรรทัดที่ 37

คำสั่งในบรรทัดที่ 37 คือการกำหนดค่าให้ตัวแปรแถวลำดับ $G[loop]$ เท่ากับ $G[fix]$

คำสั่งในบรรทัดที่ 38 คือการกำหนดค่าให้ตัวแปร i เท่ากับ $i + count + 1$

ขั้นตอน iteration

ขั้นตอนนี้จะเป็นขั้นตอนการวนซ้ำเพื่อให้ได้ค่าผลเฉลยของปัญหา ซึ่งในกระบวนการวนซ้ำแต่ละรอบนั้นจะตรวจสอบค่า $flagN$ และ $flagNN$ ว่ามีค่าเป็นจริงหรือไม่ ถ้าจริงหมายความว่าจะเปลี่ยนเงื่อนไขที่จะนำมาพิจารณาในรอบนั้น แต่ถ้าไม่จริงก็ยังคงเงื่อนไขเดิมไว้ หลังจากได้เงื่อนไข s จากกลุ่ม G_{NN} และเงื่อนไขบังคับ t จากกลุ่ม G_{N} แล้วเราจะตรวจสอบว่าปัญหากำหนดการเชิงเส้นดังกล่าวเป็นปัญหาที่ไม่มีขอบเขตหรือไม่ ด้วยการตรวจสอบว่ามุมระหว่างเวกเตอร์เกรเดียนต์ของเงื่อนไขบังคับทั้งสองมีค่าเท่ากับ π หรือไม่ ถ้ามีค่าเท่ากับ π แสดงว่าเป็นปัญหาแบบไม่มีขอบเขตและให้หยุดการทำงาน แต่ถ้าไม่เท่ากับ π เราจะคำนวณหาผลเฉลยของระบบสมการเชิงเส้นต่อไปนี้

$$\begin{aligned} a_{s1}x_1 + a_{s2}x_2 &= b_s \\ a_{t1}x_1 + a_{t2}x_2 &= b_t \end{aligned} \qquad \text{.....................................(3.3)}$$

เนื่องจากเส้นตรงทั้งสองเส้นมีเวกเตอร์เกรเดียนต์ที่ไม่ขนานกันดังนั้นเส้นตรงทั้งสองเส้นจึงไม่ขนานกันด้วยและจากบทที่ 2 เรื่องการมีผลเฉลยเสมอสำหรับระบบสมการเชิงเส้น 2 ตัวแปร 2 สมการที่

ไม่ขนานกันดังนั้นระบบสมการเชิงเส้น (3.3) จะต้องมีผลเฉลยเสมอ และสมมุติให้ผลเฉลยของระบบสมการดังกล่าวเป็น $\mathbf{x}^* = \begin{pmatrix} x_1^* & x_2^* \end{pmatrix}^{\mathrm{T}}$ ด้วยหลักเกณฑ์คราเมอร์จะได้ว่า

$$x_1^* = \frac{\begin{vmatrix} b_s & a_{s2} \\ b_t & a_{t2} \end{vmatrix}}{\begin{vmatrix} a_{s1} & a_{s2} \\ a_{t1} & a_{t2} \end{vmatrix}} = \frac{(b_s a_{t2} - b_t a_{s2})}{(a_{s1}a_{t2} - a_{s2}a_{t1})} \quad \text{และ} \quad x_2^* = \frac{\begin{vmatrix} a_{s1} & b_s \\ a_{t1} & b_t \end{vmatrix}}{\begin{vmatrix} a_{s1} & a_{s2} \\ a_{t1} & a_{t2} \end{vmatrix}} = \frac{(b_t a_{s1} - b_s a_{t1})}{(a_{s1}a_{t2} - a_{s2}a_{t1})}$$

เมื่อได้ $\mathbf{x}^*$ แล้วต่อไปจะตรวจสอบว่า $\mathbf{x}^*$ เป็นผลเฉลยที่เป็นไปได้ (feasible solution) หรือไม่ โดยตรวจสอบว่า $\mathbf{x}^*$ สอดคล้องกับทุกเงื่อนไขที่อยู่ในกลุ่ม G_{N} และทุกเงื่อนไขในกลุ่ม G_{NN} หรือไม่ สำหรับกลุ่มที่มีเงื่อนไขที่ $\mathbf{x}^*$ ไม่สอดคล้องจะต้องเปลี่ยนเงื่อนไขสำหรับกลุ่มนั้นในรอบต่อไป แต่ถ้า $\mathbf{x}^*$ สอดคล้องทุกเงื่อนไขในกลุ่ม G_{N} และกลุ่ม G_{NN} แสดงว่า $\mathbf{x}^*$ เป็นผลเฉลยที่เหมาะที่สุด และให้หยุดการทำงาน

Iterate()

1 [initialize] $flagN \leftarrow \mathbf{true}$, $flagNN \leftarrow \mathbf{true}$, $runN \leftarrow 1$, $runNN \leftarrow 1$,
$num\_loop \leftarrow \mathbf{size\_of\_}G_{\mathrm{N}} + \mathbf{size\_of\_}G_{\mathrm{NN}}$
2 กระทำซ้ำขั้นตอนที่ 3 ถึง 44 สำหรับ $l = 1, 2, \cdots, num\_loop$
3 ถ้า $flagNN = \mathbf{true}$ แล้ว
4 $s \leftarrow G_{\mathrm{NN}}[runNN]$
5 $s\_angle \leftarrow \boldsymbol{\alpha}[runNN]$
6 ถ้า $flagN = \mathbf{true}$ แล้ว
7 $t \leftarrow G_N[runN]$
8 $t\_angle \leftarrow \boldsymbol{\beta}[runN]$
9 ถ้ามุมระหว่างเวกเตอร์เกรเดียนต์ของเงื่อนไขบังคับ s และ t เท่ากับ π แล้ว
10 หยุดการทำงาน
11 มิฉะนั้น
12 $Det \leftarrow a_{s1}a_{t2} - a_{s2}a_{t1}$
13 $x_1^* \leftarrow (b_s a_{t2} - b_t a_{s2}) / Det$
14 $x_2^* \leftarrow (b_t a_{s1} - b_s a_{t1}) / Det$
▷ Checking feasibility with G_{N}
15 $flagN = \mathbf{false}$
16 กระทำซ้ำขั้นตอน 17 ถึง 29 สำหรับ $j = runN, runN+1, \cdots, \mathbf{size\_of\_}G_{\mathrm{N}}$

17 $temp \leftarrow \mathbf{A}_{j1}x_1^* + \mathbf{A}_{j2}x_2^*$

18 ถ้า $b_j = 1$ และ $temp > 1$ แล้ว

19 $flagN \leftarrow \mathbf{true}$

20 $runN \leftarrow runN + 1$

21 break

22 ถ้า $b_j = -1$ และ $temp > -1$ แล้ว

23 $flagN \leftarrow \mathbf{true}$

24 $runN \leftarrow runN + 1$

25 break

26 ถ้า $b_j = 0$ และ $temp > 0$ แล้ว

27 $flagN \leftarrow \mathbf{true}$

28 $runN \leftarrow runN + 1$

29 break

▷ Checking feasibility with G_{NN}

30 $flagNN = \mathbf{false}$

31 กระทำซ้ำขั้นตอนที่ 32 ถึง 44 สำหรับ $j = runNN, runNN + 1, \cdots, \mathbf{size\_of\_}G_{\mathrm{NN}}$

32 $temp \leftarrow \mathbf{A}_{j1}x_1^* + \mathbf{A}_{j2}x_2^*$

33 ถ้า $b_j = 1$ และ $temp > 1$ แล้ว

34 $flagNN \leftarrow \mathbf{true}$

35 $runNN \leftarrow runNN + 1$

36 break

37 ถ้า $b_j = -1$ และ $temp > -1$ แล้ว

38 $flagNN \leftarrow \mathbf{true}$

39 $runNN \leftarrow runNN + 1$

40 break

41 ถ้า $b_j = 0$ และ $temp > 0$ แล้ว

42 $flagNN \leftarrow \mathbf{true}$

43 $runNN \leftarrow runNN + 1$

44 break

45 กลับไปยังขั้นตอนที่ 16

คำสั่งในบรรทัดที่ 1 คือการกำหนดค่าเริ่มต้นให้กับตัวแปร $flagN$ และ $flagNN$ ให้มีค่าเริ่มต้นเป็นจริง(**true**) เสมอ กำหนดให้ตัวแปร $runN$ และ $runNN$ มีค่าเท่ากับ 1 และตัวแปร num_loop มีค่าเท่ากับ size_of_G_{N} +size_of_G_{NN}

คำสั่งในบรรทัดที่ 2 คือคำสั่งในการกระทำซ้ำบรรทัดที่ 3 ถึงบรรทัดที่ 44 สำหรับ $l = 1, 2, \cdots, num\_loop$

คำสั่งในบรรทัดที่ 3 ถึงบรรทัดที่ 5 คือการตรวจสอบว่า ถ้า $flagNN = \mathbf{true}$ แล้วให้ตัวแปร s เท่ากับ $G_{\mathrm{NN}}[runNN]$ และ ตัวแปร s_angle เท่ากับ $\boldsymbol{\alpha}[runNN]$

คำสั่งในบรรทัดที่ 6 ถึงบรรทัดที่ 8 คือการตรวจสอบว่า ถ้า $flagN = \mathbf{true}$ แล้วให้ตัวแปร t เท่ากับ $G_{\mathrm{N}}[runN]$ และ ตัวแปร t_angle เท่ากับ $\boldsymbol{\beta}[runN]$

<u>หมายเหตุ :</u> คำสั่งในบรรทัดที่ 3 ถึงบรรทัดที่ 8 เป็นการกำหนดเงื่อนไขบังคับที่เป็นตัวแทนในแต่ละกลุ่ม โดยที่ s เป็นตัวแทนจากกลุ่ม G_{NN} และ t เป็นตัวแทนจากกลุ่ม G_{N}

คำสั่งในบรรทัดที่ 9 ถึงบรรทัดที่ 14 คือการตรวจสอบว่ามุมระหว่างเวกเตอร์เกรเดียนต์ของเงื่อนไขบังคับ s และเงื่อนไขบังคับ t มีค่าเท่ากับ π หรือไม่ ถ้าเท่ากับ π แล้วให้หยุดการทำงาน ซึ่งหมายความว่าปัญหากำหนดการเชิงเส้นเป็นปัญหาแบบไม่มีขอบเขต มิฉะนั้นให้ตัวแปร Det มีค่าเท่ากับ $a_{s1}a_{12} - a_{s2}a_{t1}$ และตัวแปร x_1^* มีค่าเท่ากับ $(b_s a_{t2} - b_t a_{s2}) / Det$ และตัวแปร x_2^* มีค่าเท่ากับ $(b_t a_{s1} - b_s a_{t1}) / Det$

<u>หมายเหตุ:</u> คำสั่งในบรรทัดที่ 12 ถึงบรรทัดที่ 14 เป็นคำสั่งในการหาผลเฉลยของระบบสมการที่สร้างจากเงื่อนไขบังคับ s และ t

คำสั่งในบรรทัดที่ 15 คือการกำหนดค่าให้ตัวแปร $flagN$ มีค่าเป็นเท็จ(**false**)

คำสั่งในบรรทัดที่ 16 คือคำสั่งในการกระทำซ้ำบรรทัดที่ 17 ถึงบรรทัดที่ 29 สำหรับ $j = runN,\ runN+1,\ \cdots,\ \mathbf{size\_of\_}G_{\mathrm{N}}$

คำสั่งในบรรทัดที่ 17 คือการกำหนดค่าให้ตัวแปร $temp$ เท่ากับ $\mathbf{A}_{j1}x_1^* + \mathbf{A}_{j2}x_2^*$

คำสั่งในบรรทัดที่ 18 ถึงบรรทัดที่ 21 คือการตรวจสอบว่า ถ้า $b_j = 1$ และ $temp > 1$ แล้วกำหนดให้ตัวแปร $flagN$ มีค่าเป็น **true** และตัวแปร $runN$ เท่ากับ $runN + 1$ และให้ข้ามไปทำคำสั่งในบรรทัดที่ 30

คำสั่งในบรรทัดที่ 22 ถึงบรรทัดที่ 25 คือการตรวจสอบว่า ถ้า $b_j = -1$ และ $temp > -1$ แล้วกำหนดให้ตัวแปร $flagN$ มีค่าเป็น **true** และตัวแปร $runN$ เท่ากับ $runN + 1$ และให้ข้ามไปทำคำสั่งในบรรทัดที่ 30

คำสั่งในบรรทัดที่ 26 ถึงบรรทัดที่ 29 คือการตรวจสอบว่า ถ้า $b_j = 0$ และ $temp > 0$ แล้วกำหนดให้ตัวแปร $flagN$ มีค่าเป็น **true** และตัวแปร $runN$ เท่ากับ $runN + 1$ และให้ข้ามไปทำคำสั่งในบรรทัดที่ 30

คำสั่งในบรรทัดที่ 30 คือการกำหนดค่าให้ตัวแปร $flagNN$ มีค่าเป็นเท็จ(**false**)

คำสั่งในบรรทัดที่ 31 คือคำสั่งในการกระทำซ้ำบรรทัดที่ 32 ถึงบรรทัดที่ 44 สำหรับ $j = runNN,\ runNN + 1,\ \cdots,\ \mathbf{size\_of\_}G_{\mathrm{NN}}$

คำสั่งในบรรทัดที่ 32 คือการกำหนดค่าให้ตัวแปร $temp$ เท่ากับ $\mathbf{A}_{j1}x_1^* + \mathbf{A}_{j2}x_2^*$

คำสั่งในบรรทัดที่ 33 ถึงบรรทัดที่ 36 คือการตรวจสอบว่า ถ้า $b_j = 1$ และ $temp > 1$ แล้วกำหนดให้ตัวแปร $flagNN$ มีค่าเป็น **true** และตัวแปร $runNN$ เท่ากับ $runNN + 1$ และให้ข้ามไปทำคำสั่งในบรรทัดที่ 45

คำสั่งในบรรทัดที่ 37 ถึงบรรทัดที่ 40 คือการตรวจสอบว่า ถ้า $b_j = -1$ และ $temp > -1$ แล้วกำหนดให้ตัวแปร $flagNN$ มีค่าเป็น **true** และตัวแปร $runNN$ เท่ากับ $runNN + 1$ และให้ข้ามไปทำคำสั่งในบรรทัดที่ 45

คำสั่งในบรรทัดที่ 41 ถึงบรรทัดที่ 44 คือการตรวจสอบว่า ถ้า $b_j = 0$ และ $temp > 0$ แล้วกำหนดให้ตัวแปร $flagNN$ มีค่าเป็น **true** และตัวแปร $runNN$ เท่ากับ $runNN + 1$ และให้ข้ามไปทำคำสั่งในบรรทัดที่ 45

คำสั่งในบรรทัดที่ 45 คือคำสั่งให้กลับไปทำยังบรรทัดที่ 16

3.2 บทวิเคราะห์การทำงาน

ในฟังก์ชันหลักเรียกใช้ฟังก์ชันย่อย Normalize() หนึ่งครั้งในขั้นตอน classification เมื่อพิจารณาการทำงานของฟังก์ชันย่อย Normalize() แล้วจะเห็นว่ามีการวนซ้ำ m ครั้ง ดังนั้นจึงมีระยะเวลาการทำงานไม่เกิน $O(m)$ เมื่อ m คือจำนวนเงื่อนไขบังคับทั้งหมด

มีการเรียกใช้ฟังก์ชันย่อย Classify() ในขั้นตอน classification หนึ่งครั้ง และทำงานโดยวนซ้ำ m ครั้งเช่นกัน เพราะว่าทุกเงื่อนไขจะมีการจำแนกว่าควรจะอยู่ในกลุ่ม G_{N} หรือ G_{NN} ดังนั้นระยะเวลาการทำงานจะไม่เกิน $O(m)$

มีการเรียกใช้ฟังก์ชัน QuickSort() สองครั้งในขั้นตอน sorting ครั้งแรกเป็นการจัดเรียงลำดับในกลุ่ม G_{N} และครั้งที่สองสำหรับกลุ่ม G_{NN} ดังนั้นระยะเวลาการทำงานจะขึ้นกับขนาดของทั้งสองกลุ่มว่ามีข้อมูลในการจัดเรียงมากน้อยเพียงใด แต่เนื่องจากผลรวมของสมาชิกของทั้งสองกลุ่ม

เท่ากับ m ดังนั้นระยะเวลาการทำงานจึงมีค่าไม่เกินระยะเวลาการจัดเรียงข้อมูลที่มี m ตัว และด้วยวิธีการจัดเรียงแบบ quick sort จะได้ระยะเวลาการจัดเรียงไม่เกิน $O(m \lg m)$

ในขั้นตอน domination มีการเรียกใช้ฟังก์ชันย่อย Dominate() สองครั้ง ครั้งแรกสำหรับกลุ่ม G_{N} ครั้งที่สองสำหรับกลุ่ม G_{NN} ในขั้นตอนการทำงานเกิดการวนซ้ำภายนอกเริ่มต้นบรรทัดที่ 4 และมีการวนซ้ำ m ครั้ง และการวนซ้ำภายในเริ่มต้นบรรทัดที่ 10 แต่ละการวนซ้ำภายนอกจะวนซ้ำภายใน m ครั้ง และผลรวมของขนาดของทั้งสองกลุ่มเท่ากับ m ดังนั้นเวลาการทำงานจึงมีค่าไม่เกิน $O(m^2)$

ในขั้นตอนสุดท้ายคือขั้นตอน iteration เรียกใช้ฟังก์ชันย่อย Iterate() หนึ่งครั้ง และในขั้นตอนการทำงานของฟังก์ชันย่อยดังกล่าวมีการวนซ้ำภายนอกเริ่มต้นที่บรรทัดที่ 1 ซึ่งวนซ้ำ m ครั้ง และการวนซ้ำภายในมีสองครั้งเริ่มต้นที่บรรทัดที่ 16 และ บรรทัดที่ 31 ตามลำดับ ซึ่งทั้งสองครั้งรวมกันแล้วเท่ากับ m ครั้ง ดังนั้นตลอดทั้งฟังก์ชันย่อยมีระยะเวลาในการทำงานไม่เกิน $O(m^2)$

เมื่อพิจารณาระยะเวลาการทำงานรวมทั้งหมดจะได้

$$O(m) + O(m) + O(m \lg m) + O(m^2) + O(m^2) \le O(m^2)$$

ดังนั้นสรุปได้ว่าระยะเวลาการทำงานของขั้นตอนวิธีใหม่ไม่เกิน $O(m^2)$ ซึ่งเป็นแบบกำลังสอง