

ต่อไปนี้เป็นกรอธิบายสมการและตัวแปรที่ใช้ในสองวิธีที่โปรแกรมนี้ใช้ในการมอบหมายกิจกรรมให้พนักงาน ได้แก่ **Linear Programming** (ใช้ในฟังก์ชัน `assign_single_activity_from_df`) และ **Greedy Algorithm** (ใช้ในฟังก์ชัน `use_greedy_algorithm`) โดยจะแยกอธิบายสำหรับแต่ละวิธีอย่างชัดเจน

1. Linear Programming (ใช้ PuLP)

ตัวแปร

• ตัวแปรตัดสินใจ (Decision Variables):

- $x_{i,j}$: ตัวแปรไบนารี (0 หรือ 1) ที่ระบุว่าพนักงาน i (EMPLOYEE_ID) ได้รับมอบหมายให้ทำกิจกรรม j (Activity) หรือไม่
 - $x_{i,j} = 1$: พนักงาน i ได้รับมอบหมายกิจกรรม j
 - $x_{i,j} = 0$: พนักงาน i ไม่ได้รับมอบหมายกิจกรรม j
- $i \in E$: E คือเซตของพนักงานทั้งหมดในสำนักงาน ($E = \{\text{EMPLOYEE_ID_1}, \text{EMPLOYEE_ID_2}, \dots\}$)
- $j \in A$: A คือเซตของกิจกรรมทั้งหมดในสำนักงาน ($A = \{\text{Activity_1}, \text{Activity_2}, \dots\}$)

• ตัวแปรข้อมูลนำเข้า (Input Parameters):

- $v_{i,j}$: สัดส่วนการทำงานของพนักงาน i ในกิจกรรม j (จากคอลัมน์ `values` ใน DataFrame, ค่าอยู่ในช่วง $[0, 1]$)
- r_j : สัดส่วนภาพรวมของส่วนงานสำหรับกิจกรรม j (คำนวณจาก $r_j = \frac{\sum_{i \in E} v_{i,j}}{\sum_{j \in A} \sum_{i \in E} v_{i,j}}$)
- w_{ind} : น้ำหนักของสัดส่วนงานของพนักงาน (individual_weight, ค่าเริ่มต้น 0.7)
- w_{sec} : น้ำหนักของสัดส่วนภาพรวมของส่วนงาน (section_weight, ค่าเริ่มต้น 0.3)
- t_j : เป้าหมายจำนวนพนักงานที่ควรได้รับมอบหมายในกิจกรรม j (คำนวณจาก $t_j = \text{round}(|E| \cdot r_j)$, โดย $|E|$ คือจำนวนพนักงานทั้งหมด)
- $s_{i,j}$: คะแนนสำหรับการมอบหมายพนักงาน i ให้ทำกิจกรรม j
 - คำนวณจาก:

$$s_{i,j} = (v_{i,j} \cdot w_{\text{ind}}) + (r_j \cdot w_{\text{sec}}) \quad (1)$$

- S_i : เซตของพนักงานที่มีเงื่อนไขพิเศษ (พนักงานที่ทำกิจกรรม j เดียวด้วยสัดส่วน 100%, คือ $v_{i,j} = 1.0$) พร้อมกิจกรรมที่ต้องมอบหมายให้ (j)

สมการ

1. ฟังก์ชันวัตถุประสงค์ (Objective Function):

- เป้าหมายคือเพิ่มคะแนนรวมของการมอบหมายให้สูงสุด:

$$\text{Maximize} \quad \sum_{i \in E} \sum_{j \in A} s_{i,j} \cdot x_{i,j} \quad (2)$$

- อธิบาย: ต้องการให้การมอบหมายมีคะแนนรวมสูงสุด โดยคะแนน $s_{i,j}$ คำนวณจากความเหมาะสมของพนักงานในกิจกรรมนั้น (จากสัดส่วนส่วนตัวและภาพรวมของส่วนงาน)

2. ข้อจำกัด (Constraints):

- พนักงานแต่ละคนต้องได้รับมอบหมายเพียง 1 กิจกรรม:

$$\sum_{j \in A} x_{i,j} = 1 \quad \forall i \in E \quad (3)$$

- อธิบาย: ทุกพนักงานต้องได้รับมอบหมายกิจกรรมหนึ่งและเพียงหนึ่งกิจกรรมเท่านั้น

- จำนวนพนักงานที่ได้รับมอบหมายในแต่ละกิจกรรมต้องเท่ากับเป้าหมาย:

$$\sum_{i \in E} x_{i,j} = t_j \quad \forall j \in A \text{ where } t_j > 0 \quad (4)$$

- อธิบาย: จำนวนพนักงานที่ได้รับมอบหมายในกิจกรรม j ต้องตรงกับเป้าหมาย t_j (เฉพาะกิจกรรมที่มีเป้าหมายมากกว่า 0)

- เงื่อนไขพิเศษสำหรับพนักงานที่ทำกิจกรรมเดียว 100%:

$$x_{i,j} = 1 \quad \forall (i,j) \in S_i \quad (5)$$

- อธิบาย: พนักงานที่มีเงื่อนไขพิเศษ (ทำกิจกรรม j ด้วยสัดส่วน 100%) ต้องได้รับมอบหมายกิจกรรม j นั้น

- ตัวแปรเป็นไบนารี:

$$x_{i,j} \in \{0, 1\} \quad \forall i \in E, j \in A \quad (6)$$

- อธิบาย: $x_{i,j}$ ต้องเป็น 0 หรือ 1 เท่านั้น

3. การปรับเป้าหมาย (ถ้าจำเป็น):

- หากผลรวมของ t_j ไม่เท่ากับจำนวนพนักงานทั้งหมด ($\sum_{j \in A} t_j \neq |E|$) จะมีการปรับ t_j :

- ถ้า $\sum_{j \in A} t_j < |E|$: เพิ่ม t_j ของกิจกรรมที่มี r_j สูงสุด
- ถ้า $\sum_{j \in A} t_j > |E|$: ลด t_j ของกิจกรรมที่มี r_j ต่ำสุด

- สูตรการปรับ:

$$\text{Difference} = |E| - \sum_{j \in At_j} \quad (7)$$

- ถ้า $\text{Difference} > 0$: $t_j_max += \text{Difference}$
- ถ้า $\text{Difference} < 0$: ลด t_j จาก j ที่มี r_j ต่ำสุดจนครบ Difference

2. Greedy Algorithm

ตัวแปร

- ตัวแปรข้อมูลนำเข้า (Input Parameters):

- ตัวแปรส่วนใหญ่เหมือนกับ Linear Programming:

- v_i, j : สัดส่วนการทำงานของพนักงาน i ในกิจกรรม j
- r_j : สัดส่วนภาพรวมของส่วนงานสำหรับกิจกรรม j
- w_ind : น้ำหนักของสัดส่วนงานของพนักงาน (0.7)
- w_sec : น้ำหนักของสัดส่วนภาพรวมของส่วนงาน (0.3)
- t_j : เป้าหมายจำนวนพนักงาน ในกิจกรรม j
- s_i, j : คะแนนสำหรับการมอบหมายพนักงาน i ให้ทำกิจกรรม j

$$s_i, j = (v_i, j \cdot w_ind) + (r_j \cdot w_sec) \quad (8)$$

- S_i : เซตของพนักงานที่มีเงื่อนไขพิเศษและกิจกรรมที่ต้องมอบหมาย

- ตัวแปรเพิ่มเติม:

- AE : เซตของพนักงานที่ได้รับมอบหมายแล้ว ($AE \subseteq E$)
- AA_j : จำนวนพนักงานที่ได้รับมอบหมาย ในกิจกรรม j (เริ่มต้นเป็น 0)
- $Assignments$: Dictionary ที่เก็บผลการมอบหมาย ($Assignments[i] = j$ หมายถึงพนักงาน i ได้รับมอบหมายกิจกรรม j)

- ตัวแปรชั่วคราว:

- Q : Priority Queue (หรือรายการที่เรียงลำดับ) ที่เก็บการมอบหมายที่เป็นไปได้ทั้งหมดในรูปแบบ:
 - $\{\text{emp_id} : i, \text{activity} : j, \text{score} : s_i, j\}$
 - เรียงตาม s_i, j จากมากไปน้อย

สมการ

1. การคำนวณคะแนน:

- คะแนนสำหรับการมอบหมายพนักงาน i ให้ทำกิจกรรม j :

$$s_i, j = (v_i, j \cdot w_ind) + (r_j \cdot w_sec) \quad (9)$$

- อธิบาย: คะแนนนี้ใช้ในการตัดสินใจเลือกการมอบหมายที่ดีที่สุด

2. การกำหนดเป้าหมายจำนวนพนักงาน:

- จำนวนพนักงานเป้าหมายสำหรับกิจกรรม j :

$$t_j = \text{round}(|E| \cdot r_j) \quad (10)$$

- อธิบาย: คำนวณจากสัดส่วนภาพรวมของส่วนงาน คล้ายกับ Linear Programming

3. ขั้นตอนการมอบหมาย (Algorithmic Steps, ไม่ใช่สมการโดยตรง):

- ขั้นตอน 1: มอบหมายพนักงานที่มีเงื่อนไขพิเศษ:

$$\text{If } (i, j) \in S_i \text{ then } Assignments[i] = j, \quad AE = AE \cup \{i\}, \quad AA_j += 1 \quad (11)$$

- ขั้นตอน 2: สร้าง Priority Queue:

$$Q = [\{\text{emp_id} : i, \text{activity} : j, \text{score} : s_i, j\} \quad \forall i \in E \setminus AE, j \in A] \quad (12)$$

- เรียง Q ตาม s_i, j จากมากไปน้อย

- ขั้นตอน 3: มอบหมายพนักงานที่เหลือตามคะแนน:

$$\text{For each } q \in Q : \quad (13)$$

$$\text{If } q.\text{emp_id} \notin AE \text{ and } AA_q.\text{activity} < t_q.\text{activity} : \quad (14)$$

$$Assignments[q.\text{emp_id}] = q.\text{activity}, \quad AE = AE \cup \{q.\text{emp_id}\}, \quad AA_q.\text{activity} += 1 \quad (15)$$

- ขั้นตอน 4: มอบหมายพนักงานที่ยังไม่ได้รับมอบหมาย:

$$\text{For each } i \in E \setminus AE : \tag{16}$$

$$j^* = \arg \max_{j \in As_i, j} \tag{17}$$

$$\text{Assignments}[i] = j^*, \quad AE = AE \cup \{i\}, \quad AA_j^* + 1 = 1 \tag{18}$$

4. การสร้างผลลัพธ์:

- สำหรับแต่ละการมอบหมาย $\text{Assignments}[i] = j$:
 - คำนวณข้อมูลผลลัพธ์:

$$\text{Result} = \{\text{EMPLOYEE_ID} : i, \text{NAME}, \text{ASSIGNED_ACTIVITY} : j, \text{ORIGINAL_VALUE} : v_i, j, \text{SCORE} : s_i, j\}$$

- สรุปจำนวนพนักงานในแต่ละกิจกรรม:

$$\text{ASSIGNED_COUNT_j} = AA_j \tag{20}$$

$$\text{MATCHING_j} = (\text{ASSIGNED_COUNT_j} == t_j) \tag{21}$$

เปรียบเทียบสมการและตัวแปร

ด้าน	Linear Programming	Greedy Algorithm
ตัวแปรตัดสินใจ	x_i, j (ไบนารี)	ไม่มีตัวแปรตัดสินใจแบบชัดเจน ใช้ Assignments เพื่อเก็บผลการมอบหมาย
คะแนน	$s_i, j = (v_i, j \cdot w_ind) + (r_j \cdot w_sec)$	เหมือนกัน
เป้าหมาย	Maximize $\sum_{i \in E} \sum_{j \in As_i, j} x_i, j$	เลือกการมอบหมายที่มี s_i, j สูงสุดตามลำดับ
ข้อจำกัด	- พนักงาน 1 คนต่อ 1 กิจกรรม - จำนวนพนักงานในกิจกรรม = t_j - เงื่อนไขพิเศษ	- พิจารณาเงื่อนไขพิเศษก่อน - จำกัด $AA_j \leq t_j$ - มอบหมายทุกพนักงาน
วิธีการแก้ปัญหา	ใช้ PuLP เพื่อหาคำตอบที่เหมาะสมที่สุด	ใช้การเรียงลำดับคะแนนและเลือกตามลำดับ (Greedy)
ผลลัพธ์	$x_i, j = 1$ ระบุการมอบหมาย	$\text{Assignments}[i] = j$ ระบุการมอบหมาย

หมายเหตุ

- **Linear Programming** เหมาะสำหรับการหาคำตอบที่เหมาะสมที่สุด (optimal) แต่ใช้เวลาและทรัพยากรการคำนวณมากกว่า และอาจล้มเหลวหากข้อจำกัดขัดแย้งกัน
- **Greedy Algorithm** เร็วกว่าและรับประกันว่าจะมอบหมายทุกพนักงาน แต่ผลลัพธ์อาจไม่เหมาะสมที่สุด (suboptimal)
- ทั้งสองวิธีใช้ คะแนน s_i, j และ เป้าหมาย t_j ที่คำนวณจากข้อมูลเดียวกัน เพื่อให้ผลลัพธ์สอดคล้องกับความต้องการของส่วนงาน

Ref. https://grok.com/share/bGVnYWN5_92ee1668-a129-4951-bdc1-d2fc92f21fb0