517 311 การวิเคราะห์และการออกแบบขั้นตอนวิธี

Greedy Algorithms

26 September 2023

วัตถุประสงค์

- อธิบายหลักการของอัลกอริทึมเชิงละโมบ (Greedy Algorithm)
- แสดงขั้นตอนการแก้ปัญหาด้วยการใช้อัลกอริทึมเชิงละโมบ

อัลกอริทึมเชิงละโมบ

อัลกอริทึมเชิงละโมบ (Greedy Algorithm) เป็นวิธีการในการแก้ปัญหาการหาค่าเหมาะสมที่สุด (Optimization Problem)

- มีลักษณะการทำงานเป็นขั้นตอนซ้ำ ๆ ในการเลือกส่วนของคำตอบสำหรับแต่ละส่วนย่อย ๆ ของปัญหา
- ปัญหาย่อยจะใช้กฎเชิงละโมบ (Greedy Rule)
- เมื่อเลือกคำตอบของส่วนย่อยทั้งหมดแล้วจะได้คำตอบของปัญหาใหญ่ซึ่งในหลายกรณี
- การเลือกคำตอบที่ดีที่สุดสำหรับปัญหาย่อยนั้น โดยไม่ได้คำนึงถึงปัญหาที่ใหญ่ขึ้นมา
- การพิสูจน์ว่าวิธีการใช้กฎเชิงละโมบให้คำตอบที่ถูกต้อง ในกรณีทั่วไป

ตัวอย่าง 1

ปัญหาการแลกเหรียญ (Coin Changing Problem) กำหนดให้เหรียญมี n แบบ โดยแต่ละเหรียญมีมูลค่าเป็นดังนี้ $d_1, d_2, \ldots, d_n$ ซึ่งมูลค่าเหรียญเป็นจำนวนเต็ม โดยที่ $d_1 > d_2 > \ldots > d_n$ และ $d_n = 1$ ในแต่ละมูลค่าของเหรียญมีจำนวนเหรียญไม่จำกัด กำหนดค่าที่ต้องการแลกเท่ากับ A เป็นจำนวนเต็มบวก ผลลัพธ์ที่ต้องการคือ หาวิธีการรวมมูลค่าของเหรียญให้เท่ากับ A โดยใช้จำนวนเหรียญที่น้อยที่สุด กำหนดให้ c_i คือ จำนวนเหรียญของเหรียญที่มีมูลค่าเท่ากับ d_i เมื่อ $1 \leq i \leq n$ จำนวนเหรียญสามารถแสดงเป็นสมการดังนี้

$$\sum_{i=1}^{n} c_i \tag{1}$$

เมื่อ $\sum_{i=1}^{n} c_i d_i = A$ โดยค่าของสมการที่ 1 มีค่าน้อยที่สุด

แนวคิด

- ขั้นแรกก็เลือกพิจารณาเหรียญที่มูลค่าสูงสุดก่อน
- พิจารณาว่ามูลค่าของเหรียญที่เลือกมาเกินกว่าจำนวนที่ต้องการแลกหรือไม่
- พิจารณามูลค่าส่วนที่เกิน มาพิจารณาเหรียญที่จะแลกในลำดับต่อไปจนได้จำนวนครบตามที่ต้องการแลก

กำหนด มูลค่าของเหรียญอยู่ 3 ชนิด คือ 10, 5 และ 1 มูลค่าที่ต้องการแลก $A = 27$

- เลือกเหรียญที่มีมูลค่า 10 จำนวนสองเหรียญ
- เลือกเหรียญที่มีมูลค่า 5 จำนวนหนึ่งเหรียญ
- เลือกเหรียญที่มีมูลค่า 1 จำนวนสองเหรียญ
- จำนวนเหรียญรวมคือ 5 เหรียญ

คุณสมบัติ

ปัญหาที่จะแก้ด้วยอัลกอริทึมเชิงละโมบจะต้องมีคุณสมบัติ

- คุณสมบัติในการเลือกแบบละโมบ (Greedy Choice Property) หมายถึง วิธีการเลือกคำตอบที่ดีที่สุดในส่วนของปัญหาย่อยหรือสถานการณ์ปัจจุบัน จะต้องเป็นส่วนหนึ่งของคำตอบของปัญหาทั้งหมดด้วย
- หลักของความเหมาะสมที่สุดหรือโครงสร้างย่อยเหมาะสมที่สุดเหมือนในกำหนดการพลวัต

ปัญหาจัดตารางแบบช่วง

- กำหนดให้มีกิจกรรมจำนวน n กิจกรรมที่ใช้ทรัพยากรร่วมกัน
- กิจกรรมแทนด้วยเซต $A = \{a_1, a_2, \ldots, a_n\}$ แต่ละกิจกรรม a_i เมื่อ $1 \leq i \leq n$
- ต้องการใช้ทรัพยากรในระหว่างเวลา $[s_i, f_i)$ หมายถึง เวลาเริ่มต้น (Start Time) ที่เวลา s_i โดยรวม s_i ด้วย แต่เวลาสิ้นสุด (Finish Time) f_i แต่ไม่รวมที่จุด f_i ซึ่งมีเงื่อนไขว่า $s_i < f_i$ หมายความว่า ช่วงเวลาต้องไม่เป็นศูนย์
- สำหรับกิจกรรม 2 กิจกรรม แทนด้วย a_i และ a_j มีความสอดคล้องกัน ถ้า 2 กิจกรรมนี้ไม่มีการใช้ทรัพยากรในเวลาที่ซ้อนกัน
  - $f_i \leq s_j$ หมายถึง กิจกรรม a_i สิ้นสุดก่อนหรือพร้อมกับกิจกรรม a_j จะเริ่ม
  - $f_j \leq s_i$ หมายถึง กิจกรรม a_j สิ้นสุดก่อนหรือพร้อมกับกิจกรรม a_i จะเริ่ม
- เลือกเซตของกิจกรรมที่มีจำนวนมากที่สุดซึ่งทุกกิจกรรมมีความสอดคล้องกันหรือไม่ซ้อนกัน

ตัวอย่าง

i	1	2	3	4	5	6	7	8	9	10	11
s_i	1	5	9	13	3	7.5	11	2	11	0	11
f_i	4	8	12	16	7	9.5	17	7	15	6	14

การเลือกกิจกรรมมาใส่เป็นคำตอบ

- แนวคิดในการเลือกกิจกรรม a_{i_1} แล้วจะนำกิจกรรมอื่นที่ขัดแย้งกับกิจกรรม a_{i_1} ออกไป
- จากนั้นจะเลือกกิจกรรม a_{i_2} ที่ไม่ขัดแย้งกับกิจกรรม a_{i_1} แล้วลบกิจกรรมอื่นที่ขัดแย้งกับ a_{i_2} ออก
- ทำซ้ำกระบวนการ เลือกกิจกรรมใหม่ a_j ขึ้นมา แล้วลบกิจกรรมที่ขัดแย้งกับกิจกรรมใหม่ a_j นี้ จนไม่มีกิจกรรมเหลืออยู่

ส่วนที่ต้องตัดสินใจต่อไปคือ วิธีเลือกกิจกรรม $a_{i_1}, a_{i_2}, \ldots$

อัลกอริทึม

$Q \leftarrow \{a_1\}$
$j \leftarrow 1$
$n \leftarrow |A|$
for $i = 2$ **to** n **do**
    if $s_i \geq f_j$ **then**
        $Q \leftarrow Q \cup \{a_i\}$
        $j \leftarrow i$
    end if
end for

ตัวอย่าง 2

จงแสดงการหาคำตอบของปัญหาจัดตารางแบบช่วง เมื่อกำหนดกิจกรรมโดยมีเวลาเริ่มต้นและเวลาสิ้นสุดดังต่อไปนี้

i	1	2	3	4	5	6	7	8	9	10	11
s_i	1	3	0	6	3	6	7	9	9	2	13
f_i	4	5	7	8	9	10	11	12	13	14	15

เริ่มต้น $Q = \{a_1\}$, $j = 1$, $n = 11$

- $j = 1$ และ $i = 2$ ดังนั้น $s_2 = 3$ และ $f_1 = 4$ ซึ่ง $s_2 < f_1$ ไม่เลือกกิจกรรมนี้
- $j = 1$ และ $i = 3$ ดังนั้น $s_3 = 0$ และ $f_1 = 4$ ซึ่ง $s_3 < f_1$ ไม่เลือกกิจกรรมนี้
- $j = 1$ และ $i = 4$ ดังนั้น $s_4 = 6$ และ $f_1 = 4$ ซึ่ง $s_4 \geq f_1$ เลือกกิจกรรมนี้
- ดังนั้น $Q = \{a_1, a_4\}$ และ $j = 4$

- $j = 4$ และ $i = 5$ ดังนั้น $s_5 = 3$ และ $f_4 = 8$ ซึ่ง $s_5 < f_4$ ไม่เลือกกิจกรรมนี้
- $j = 4$ และ $i = 6$ ดังนั้น $s_6 = 6$ และ $f_4 = 8$ ซึ่ง $s_6 < f_4$ ไม่เลือกกิจกรรมนี้
- $j = 4$ และ $i = 7$ ดังนั้น $s_7 = 7$ และ $f_4 = 8$ ซึ่ง $s_7 < f_4$ ไม่เลือกกิจกรรมนี้
- $j = 4$ และ $i = 8$ ดังนั้น $s_8 = 9$ และ $f_4 = 8$ ซึ่ง $s_8 \geq f_4$ เลือกกิจกรรมนี้
- ดังนั้น $Q = \{a_1, a_4, a_8\}$ และ $j = 8$
- $j = 8$ และ $i = 9$ ดังนั้น $s_9 = 9$ และ $f_8 = 12$ ซึ่ง $s_9 < f_8$ ไม่เลือกกิจกรรมนี้
- $j = 8$ และ $i = 10$ ดังนั้น $s_{10} = 2$ และ $f_8 = 12$ ซึ่ง $s_{10} < f_8$ ไม่เลือกกิจกรรมนี้
- $j = 8$ และ $i = 11$ ดังนั้น $s_{11} = 13$ และ $f_8 = 12$ ซึ่ง $s_{11} \geq f_8$ เลือกกิจกรรมนี้
- ดังนั้นคำตอบคือ $Q = \{a_1, a_4, a_8, a_{11}\}$

ตัวอย่าง 3

จงแสดงการหาคำตอบของปัญหาจัดตารางแบบช่วง เมื่อกำหนดกิจกรรมโดยมีเวลาเริ่มต้นและเวลาสิ้นสุดดังต่อไปนี้

i	1	2	3	4	5	6	7
s_i	1	3	5	6	3	8	7
f_i	12	9	7	10	5	11	8

เนื่องจากตารางของโจทย์ยังไม่ได้เรียงลำดับตามเวลาสิ้นสุดให้ เป็น $B = \{b_1, b_2, \ldots, b_7\}$

i	1	2	3	4	5	6	7
s_i	3	5	7	3	6	8	1
f_i	5	7	8	9	10	11	12
B	b_5	b_3	b_7	b_2	b_4	b_6	b_1

จงพิสูจน์ว่า คำตอบต้องมีจำนวนกิจกรรม 4 กิจกรรม ต้องยกตัวอย่างกิจกรรมในเซตของคำตอบด้วย

ปัญหาถุงเป้ (Knapsack Problem)

- มีสิ่งของจำนวน n ชิ้น แทนด้วย $a_1, a_2, \ldots, a_n$
- โดยสิ่งของชิ้นที่ i จะมีมูลค่า v_i และน้ำหนัก w_i
- ถุงเป้ที่มีความสามารถบรรจุน้ำหนักได้ C
- ให้เลือกของใส่เข้ามาในถุงให้มีมูลค่ามากที่สุดและน้ำหนักรวมไม่เกิน C โดย $w_i \leq C$ เมื่อ $1 \leq i \leq n$
- กำหนดให้ x_i เป็นอัตราส่วนการเลือกของสิ่งนั้น โดยที่ $0 \leq x_i \leq 1$ ซึ่งต้องการหาวิธีการเลือกสิ่งของที่ทำให้ค่าของ P มีค่ามากที่สุด โดย P มีนิยามดังนี้

$$P = \sum_{i=1}^{n} x_i v_i \tag{2}$$

โดยที่ $\sum_{i=1}^{n} x_i w_i \leq C$

รูปแบบปัญหา

- ถ้า $x_i \in \{0, 1\}$ เรียกว่าปัญหาถุงเป้แบบ 0/1 (0/1 Knapsack Problem)
- หาก x_i มีค่าในลักษณะเป็นจำนวนเต็มตั้งแต่ 0 ถึง 1 นั้นคือ $0 \leq x_i \leq 1$ เรียกว่า ปัญหาถุงเป้แบบต่อเนื่อง (Continuous Knapsack Problem)
- อัลกอริทึมเชิงละโมบสามารถแก้ปัญหาถุงเป้แบบต่อเนื่อง

อัลกอริทึม

```
for i = 1 to n do
    x[i] ← 0
end for
weight ← 0
value ← 0
i ← 1
while (weight < C) AND (i ≤ n) do
    if weight + w[i] ≤ C then
        x[i] ← 1
        weight ← weight + w[i]
        value ← value + v[i]
    else
        x[i] ← (C − weight)/w[i]
        value ← x[i] ∗ v[i]
        weight ← C
    end if
    i ← i + 1
end while
```

- กำหนดว่าสิ่งของ $a_1, a_2, \ldots, a_n$ เรียงจากค่าอัตราส่วนของมูลค่าต่อน้ำหนักจากมากไปน้อยแล้ว
- ให้สิ่งของอยู่ในเซตที่เรียกว่า A
- ทำให้มูลค่าของสิ่งของเป็น $v_1,v_2,\ldots,v_n$ แทนด้วยอาเรย์ $v[1], v[2], \ldots, v[n]$ และ
- น้ำหนักของสิ่งของแสดงด้วย $w_1,w_2,\ldots,w_n$ แทนด้วยอาเรย์ $w[1], w[2], \ldots, w[n]$
- สำหรับค่าอัตราส่วนที่เลือกสิ่งของชิ้นที่ i แทนด้วย x_i ในรูปของอาเรย์คือ $x[i]$ ซึ่งเป็นค่าที่สอดคล้องกับการเรียงลำดับสิ่งของไปด้วย
- ตัวแปร *weight* เก็บน้ำหนักของสิ่งของที่เก็บไปไว้
- ตัวแปร *value* เก็บมูลค่าของสิ่งของที่ใส่ถุงไว้

ตัวอย่าง 4

จงแสดงการแก้ปัญหาถุงเป้ที่ถุงเป้บรรจุได้เท่ากับ 100 โดยกำหนดสิ่งของพร้อมกับน้ำหนักและมูลค่าเป็นตารางต่อไปนี้

i	v_i	w_i
1	20	10
2	30	20
3	66	30
4	40	40
5	60	50

i	v_i	w_i	v_i/w_i	ลำดับในเซต A
1	20	10	2	a_2
2	30	20	1.5	a_3
3	66	30	2.2	a_1
4	40	40	1	a_5
5	60	50	1.2	a_4

- $i = 1$ เลือก a_1 แล้ว $x[1] = 1$, $weight = 30$ และ $value = 66$
- $i = 2$ เลือก a_2 แล้ว $x[2] = 1$, $weight = 30 + 10 = 40$ และ $value = 66 + 20 = 86$
- $i = 3$ เลือก a_3 แล้ว $x[3] = 1$, $weight = 40 + 20 = 60$ และ $value = 86 + 30 = 116$
- $i = 4$ เลือก a_4 แล้ว $x[4] = (100 - 60)/50 = 0.8$, $weight = 100$ และ $value = 116 + 0.8 * 60 = 164$ เนื่องจาก $weight$ มีค่าเท่ากับ C ดังนั้น ออกจากคำสั่ง *while*

ผลลัพธ์คือมูลค่าที่ได้คือ 164 โดยที่ อัตราส่วนของสิ่งของที่เลือก $x = (1, 1, 1, 0.8, 0)$

ตัวอย่าง 5

กำหนดสิ่งของพร้อมกับมูลค่าเป็นตารางต่อไปนี้

i	v_i	w_i	v_i/w_i
1	60	10	6
2	100	20	5
3	120	30	4

โดยถุงเป้บรรจุได้เท่ากับ 50

ปัญหาถุงเป้แบบต่อเนื่องใช้อัลกอริทึมเชิงละโมบได้

- $i = 1$ เลือก a_1 แล้ว $x[1] = 1$, $weight = 10$ และ $value = 60$
- $i = 2$ เลือก a_2 แล้ว $x[2] = 1$, $weight = 20 + 10 = 30$ และ $value = 100 + 60 = 160$
- $i = 3$ เลือก a_3 แล้ว $x[3] = (50 - 30)/30 = 2/3$, $weight = 50$ และ $value = 160 + ((2/3) * 120) = 160 + 80 = 240$ เนื่องจาก $weight$ มีค่าเท่ากับ C ดังนั้น ออกจากคำสั่ง *while*

มูลค่ารวมเท่ากับ $60 + 100 + 80 = 240$ โดยมีอัตราส่วนในการเลือกเป็น $(1, 1, 2/3)$

ข้อสังเกต

- ปัญหาถุงเป้แบบ 0/1 หากใช้แนวคิดอัลกอริทึมเชิงละโมบเหมือนกับปัญหาถุงเป้แบบต่อเนื่องแล้ว จะทำการตัดส่วนที่เป็นเศษส่วนออก จะได้การเลือกเลือกชิ้นที่ 1 และ 2 ทั้งชิ้นมูลค่ารวมเท่ากับ 160 โดยตัดชิ้นที่ 3 ออกเนื่องจากเลือกได้ไม่เต็มชิ้น
- แต่หากเลือกชิ้นที่ 2 และ 3 มีมูลค่ารวมเท่ากับ $100 + 120 = 220$ ซึ่งได้มูลค่ามากกว่า
- ดังนั้น อาจจะกล่าวว่าการใช้อัลกอริทึมเชิงละโมบไม่สามารถแก้ปัญหาถุงเป้แบบ 0/1 ได้

ปัญหาการแลกเหรียญ

$i \leftarrow 1$
while $A > 0$ **do**
  $c[i] \leftarrow A$ **div** $d[i]$
  $A \leftarrow A - c[i] * d[i]$
  $i \leftarrow i + 1$
end while

$A = 24$

- เมื่อกำหนดให้มูลค่าของเหรียญที่ใช้แลกเป็น $10, 5, 1$
- เมื่อกำหนดให้มูลค่าของเหรียญที่ใช้แลกเป็น $10, 6, 1$
- สรุปเกี่ยวกับการใช้อัลกอริทึมเชิงละโมบในการแก้ปัญหาการแลกเหรียญอย่างไร