

אלגוריתמים כלכליים

מיכאל טרושקין

* שאלה 5: מציאת שיפור פארטו חלש בגרף ההחלפות

כתבו פונקציה בפייתון, המקבלת מטריצה המייצגת חלוקת חפצים לאנשים, ומטריצה המייצגת הערכות של אנשים לחפצים, ומעגל בגרף הצריכה. הפונקציה מחשבת שיפור פארטו חלש המבטל לפחות קשת אחת במעגל.

```
def find_weak_pareto_improvement(
    allocation: List[List[float]],
    valuations: List[List[float]],
    cycle: list
):
```

המעגל `cycle` הוא מהצורה: [שחקן, חפץ, שחקן, חפץ, ...]. למשל: [0, 13, 2, 12, 1, 11, 0]. הוא מעגל שבו שחקן 0 משתף את חפץ 11 עם שחקן 1, שחקן 1 משתף את חפץ 12 עם שחקן 2, וכו'.

הפלט צריך להיות החלוקה החדשה – לאחר השיפור.

בדקו והדגימו את תשובתכם.

האלגוריתם שעשיתי מוצא שיפור פראטו חלש או חזק.

זה לא משנה לאלגוריתם.

שלבי האלגוריתם:

1. בדוק את שווי המעגל המכוון בכיוון הראשוני שהביאו, כלומר הערך של שחקן 1, לחפץ הראשון, לחלק לערך השחקן השני וכו'...
2. סובב את המעגל, כך שכפל כל היחסים קטן שווה ל-1.
3. עבור על המעגל ובדוק מה הערך הכי קטן ששחקן כלשהו מסוגל לקבל מהקודם שלו, ושמור אותו כ *minget*, שמור ב *minlose* את הערך שאותו שחקן יעבד על ידי העברת כל החלק שלו לשחקן העוקב.
4. סובב את המעגל כך שהשחקן הראשון זה השחקן שיעביר את כל החלק שלו להבא בתור. כעת אנו יודעים שהוא מעביר לשחקן הבא *minget* ומעבד *minlose*.
5. שחקן 1, שולח לשחקן 2 את הכל.
6. שחקן 2, שולח לשחקן הבא בתור חלק ששווה ערך ל *minget* כלומר החלק שהוא קיבל (או את המקסימום שהוא יכול) . וכו'...

כל שחקן שולח לשחקן הבא בתור חלק מחפץ ששווה ערך למה שקיבל עד עכשיו, בגלל שהמעגל הוא עם יחס ערכים קטן שווה ל-1, נובע שהשחקן הראשון יקבל לפחות *minlose*, שזה שווי החתיכה שהוא שלח מבחינתו.

האלגוריתם:

הפונקציה הנ"ל מקבלת את המטריצות החלוקה והשווי והמעגל, ומחשבת את הערך המינימום שיכול לאשלח, והערך המינימום שזה עולה לאותו שחקן, וגם מחזירה את המעגל החדש המסובב.

```

def find_cycle_value(item_partitions, item_values, cycle):
    # Find the value of the cycle, i.e player1_value / player2_value etc...
    # Find the minimum amount that can be transfered, i.e if every player transfer this much then someone will be left with no item.
    val = 1
    min_get_val = 100

    min_index = 0
    for i in range(0, len(cycle)-2, 2):
        first_player = cycle[i]
        next_player = cycle[i+2]
        item = cycle[i+1]
        first_pv = item_values[item, first_player]
        next_pv = item_values[item, next_player]
        val *= (first_pv / next_pv)

    cycle = list(np.array(cycle))
    if val > 1:
        cycle.reverse()
    min_get_val = 100
    for i in range(0, len(cycle)-2, 2):
        first_player = cycle[i]
        next_player = cycle[i+2]
        item = cycle[i+1]
        first_pv = item_values[item, first_player]
        next_pv = item_values[item, next_player]
        # print("index i :", i, " | fp :", first_player, " | item :", item, " | np :", next_player)
        # print(first_pv, next_pv)

        fpp = item_partitions[item, first_player]
        npp = item_partitions[item, next_player]

        if min_get_val > fpp * next_pv:
            min_index = i
            min_get_val = fpp * next_pv
            min_lose_val = fpp * first_pv

    shifted_cycle = []
    for i in range(min_index, len(cycle)):
        shifted_cycle.append(cycle[i])
    for i in range(1, min_index):
        shifted_cycle.append(cycle[i])
    shifted_cycle.append(cycle[min_index])
    return val, min_get_val, min_lose_val, shifted_cycle

```

✓ 0.3s

למשל עבור הדוגמא הבא:

```

[[0.      0.18341908 0.32308974 0.49349118]
 [0.23394398 0.      0.22138323 0.54467279]
 [0.38432847 0.23134307 0.      0.38432847]
 [0.31534001 0.32896326 0.      0.35569673]
 [0.55340708 0.23669644 0.      0.20989648]]
[[19  7 33 25]
 [ 6  8  8 11]
 [36 21 13 20]
 [14 39 41 20]
 [25 25  5 24]]

```

כאשר כל עמודה זה שחקן, ושורה זה החפץ, במקרה הזה 4 שחקנים ו 5 חפצים, והמטריצה השנייה היא הערכים שכל שחקן מביא לחפצים.

נראה שיש את מעגל השיתופים הבא:

```
cycle = [2, 0, 3, 4, 0, 1, 2]
```

כלומר שחקן 2, משתף חפץ 0 עם שחקן 3, שחקן 3 משתף את 4 עם 0, שחקן 0 את 1 עם 2.

```

    find_cycle_value(item_partitions, item_values, cycle)
✓ 0.3s

(1.0521885521885521,
 1.8715518331128973,
 1.403663874834673,
 [0, 1, 2, 0, 3, 4, 0])

```

הערך הראשון מתאר מה היחס הערכים עבור המעגל המקורי, כלומר במקרה הזה נסובב אותו, השני אומר שאם שחקן 0 ישלח את כל חפץ 1 לשחקן 2, אז שחקן 2 יקבל 1.87 בערך. והשלישי אומר ששחקן 0 יעבד 1.4 בערך.

```

def find_pareto_improvement(item_partitions, item_values, cycle):
    # Get the min_val and val.
    val, min_get_val, min_lost_val, cycle = find_cycle_value(item_partitions, item_values, cycle)
    item_partitions = item_partitions.copy()
    sum_improvement = 0

    prev_val = min_lost_val
    for i in range(0, len(cycle)-2, 2):
        first_player = cycle[i]
        next_player = cycle[i+2]
        item = cycle[i+1]
        first_pv = item_values[item, first_player]
        next_pv = item_values[item, next_player]

        fpp = item_partitions[item, first_player]
        npp = item_partitions[item, next_player]

        val = min_get_val
        if(fpp / npp < 1):
            val = min_get_val

        max_transfer_amount = prev_val / first_pv
        transfer_amount = min(max_transfer_amount, fpp)

        gets = transfer_amount * next_pv
        prev_val = gets

        print(first_pv, next_pv)
        print("player", first_player, "transfers", transfer_amount, "to player", next_player)
        print("player", first_player, "loses", transfer_amount * first_pv, "\nplayer", next_player, "gets", gets)
        print("player", first_player, "has", fpp - transfer_amount, "of item", item)
        sum_improvement += transfer_amount * next_pv - transfer_amount * first_pv
        item_partitions[item, first_player] -= transfer_amount
        item_partitions[item, next_player] += transfer_amount
    print("total cycle improvement :", sum_improvement)
    return item_partitions

```

האלגוריתם עצמו, בכל שלב תביא לשחקן הבא חתיכה מהחפץ שלך ששוות ערך למה שקיבלת מהשחקן הקודם, במידה ואין מספיק שלח הכל.

על הדוגמא מלפני זה נקבל את השיפור הבא:

```
item_partitions_improved = find_pareto_improvement(item_partitions, item_values, cycle)
item_partitions_improved
✓ 0.2s

6 8
player 0 transfers 0.23394397913911216 to player 2
player 0 loses 1.403663874834673
player 2 gets 1.8715518331128973
player 0 has 0.0 of item 1
33 25
player 2 transfers 0.05671369191251204 to player 3
player 2 loses 1.8715518331128973
player 3 gets 1.4178422978128011
player 2 has 0.26637604821288174 of item 0
24 25
player 3 transfers 0.059076762408866716 to player 0
player 3 loses 1.4178422978128011
player 0 gets 1.476919060221668
player 3 has 0.15081971556117466 of item 4
total cycle improvment : 0.07325518538699494

array([[0.          , 0.18341908, 0.26637605, 0.55020487],
       [0.          , 0.          , 0.45532721, 0.54467279],
       [0.38432847, 0.23134307, 0.          , 0.38432847],
       [0.31534001, 0.32896326, 0.          , 0.35569673],
       [0.61248384, 0.23669644, 0.          , 0.15081972]])
```

שזה שיפור פראטו חזק:

```
print(calculate_val(item_partitions, item_values, num_items, num_players))
print(calculate_val(item_partitions_improved, item_values, num_items, num_players))
✓ 0.3s

[33.48942579 24.88911616 12.43302726 38.16669961]
[33.56268098 24.88911616 12.43302726 38.16669961]
```

דוגמא נוספת:

```
num_items2 = 7
num_players2 = 5
item_partitions2 = create_partitions(0.3, 0.7, num_items2, num_players2)
print(item_partitions2)
```

9] ✓ 0.4s

```
[[0.      0.34511722 0.17378638 0.34511722 0.13597918]
 [0.39326306 0.60673694 0.      0.      0.      ]
 [0.31055667 0.      0.20399954 0.17488712 0.31055667]
 [0.      0.39708776 0.      0.      0.60291224]
 [0.38164499 0.23671002 0.38164499 0.      0.      ]
 [0.30918862 0.19162968 0.30918862 0.      0.18999308]
 [0.17136661 0.16416879 0.25578644 0.15289172 0.25578644]]
```

```
item_values2 = create_item_values(num_items2, num_players2)
print(item_values2)
```

1] ✓ 0.3s

```
[100 100 100 100 100]
[[20 14  2  9 19]
 [ 4 13 18 11 23]
 [ 7  4  3  7 20]
 [12 21 23 16  5]
 [25 30 23 14 24]
 [11 15 20 19  3]
 [21  3 11 24  6]]
```

```
cycle2 = [3, 0, 4, 6, 1, 4, 2, 2, 3]
```

3] ✓ 0.4s

```

    item_partitions_improved2 = find_pareto_improvement(item_partitions2, item_values2, cycle2)
25] ✓ 0.3s
· 0.7673593214432672 1.5347186428865345
  6 3
  player 4 transfers 0.25578644048108906 to player 1
  player 4 loses 1.5347186428865345
  player 1 gets 0.7673593214432672
  player 4 has 0.0 of item 6
  30 23
  player 1 transfers 0.025578644048108907 to player 2
  player 1 loses 0.7673593214432672
  player 2 gets 0.5883088131065048
  player 1 has 0.21113138033801201 of item 4
  3 7
  player 2 transfers 0.19610293770216827 to player 3
  player 2 loses 0.5883088131065048
  player 3 gets 1.3727205639151778
  player 2 has 0.007896604850938849 of item 2
  9 19
  player 3 transfers 0.1525245071016864 to player 4
  player 3 loses 1.3727205639151778
  player 4 gets 2.8979656349320417
  player 3 has 0.19259271167934167 of item 0
  total cycle improvment : 1.3632469920455073

```

```

    print(calculate_val(item_partitions2, item_values2, num_items2, num_players2))
    print(calculate_val(item_partitions_improved2, item_values2, num_items2, num_players2))
26] ✓ 0.3s

```

```

· [20.28784722 31.52631659 18.73482937 7.99966612 13.91399691]
  [20.28784722 31.52631659 18.73482937 7.99966612 15.2772439 ]

```

שגם זהו שיפור פראטו חזק.

דוגמא ספציפית יותר

```

    item_partitions3 = np.zeros((2,2))
    item_partitions3[0,0] = 0.01
    item_partitions3[0,1] = 0.99
    item_partitions3[1,0] = 0.01
    item_partitions3[1,1] = 0.99
    item_values3 = np.zeros((2,2))
    item_values3[0,0] = 100
    item_values3[0,1] = 2
    item_values3[1,0] = 100
    item_values3[1,1] = 2
    print(item_partitions3)
    print(item_values3)
[427] ✓ 0.4s

... [[0.01 0.99]
      [0.01 0.99]]
      [[100.  2.]
      [100.  2.]]

    cycle3 = [0, 0, 1, 1, 0]
[428] ✓ 0.2s

    item_partitions_improved3 = find_pareto_improvement(item_partitions3, item_values3, cycle3)
[429] ✓ 0.4s

... 0.02 1.0
    100.0 2.0
    player 0 transfers 0.01 to player 1
    player 0 loses 1.0
    player 1 gets 0.02
    player 0 has 0.0 of item 0
    2.0 100.0
    player 1 transfers 0.01 to player 0
    player 1 loses 0.02
    player 0 gets 1.0
    player 1 has 0.98 of item 1
    total cycle improvment : 0.0

    print(calculate_val(item_partitions3, item_values3, 2, 2))
    print(calculate_val(item_partitions_improved3, item_values3, 2, 2))
[430] ✓ 0.2s

... [2.  3.96]
      [2.  3.96]

```

ודוגמא שבא הערך הנמוך ביותר ששחקן כלשהו מעריך עבור החלק שלו, גדול ממש מהחלק שיכול לקבל משחקן אחר כלשהו

```

item_partitions4 = np.zeros((2,2))
item_partitions4[0,0] = 0.01
item_partitions4[0,1] = 0.99
item_partitions4[1,0] = 0.999
item_partitions4[1,1] = 0.001
item_values4 = np.zeros((2,2))
item_values4[0,0] = 1
item_values4[0,1] = 50
item_values4[1,0] = 1
item_values4[1,1] = 50
print(item_partitions4)
print(item_values4)
✓ 0.2s

[[0.01  0.99 ]
 [0.999 0.001]]
[[ 1. 50.]
 [ 1. 50.]]

item_partitions_improved4 = find_pareto_improvement(item_partitions4, item_values4, cycle3)
item_partitions_improved4
✓ 0.3s

50.0 1.0
player 1 transfers 0.001 to player 0
player 1 loses 0.05
player 0 gets 0.001
player 1 has 0.0 of item 1
1.0 50.0
player 0 transfers 0.001 to player 1
player 0 loses 0.001
player 1 gets 0.05
player 0 has 0.009000000000000001 of item 0
total cycle improvment : 0.0

array([[0.009, 0.991],
       [1.    , 0.   ]])

print(calculate_val(item_partitions4, item_values4, 2, 2))
print(calculate_val(item_partitions_improved4, item_values4, 2, 2))
✓ 0.3s

[ 1.009 49.55 ]
[ 1.009 49.55 ]

```

שני הדוגמאות הנ"ל מביאות שיפור פראטו חלש עם פחות חלוקות.