

2010 年度 大問 5

hari64boli64 (hari64boli64@gmail.com)

2025 年 4 月 23 日

1 問題

$$\begin{aligned} & \max \sum_{i=1}^n p_i x_i \\ & \text{subject to } \sum_{i=1}^n s_i x_i \leq S \quad (*) \\ & x_i \in \{0, 1\} \quad (i = 1, \dots, n) \quad (*) \end{aligned}$$

2 解答

5.py が解答。

slow とあるのは、正当性の検証目的であり、実際には不要。

Listing 1 answer

```
1 import numpy as np
2 import random
3 from itertools import product
4
5
6 def makeProblem(problem: str):
7     if problem == "P":
8         N = 5
9         S = 5
10        Ps = [2, 3, 2, 1, 3]
11        Ss = [2, 3, 1, 2, 1]
12        return N, S, Ps, Ss
13    elif problem == "Q":
14        N = random.randint(1, 5)
15        S = random.randint(1, 10)
```

```

16         Ps = [random.randint(1, 10) for _ in range(N)]
17         Ss = [random.randint(1, 10) for _ in range(N)]
18         return N, S, Ps, Ss
19     elif problem == "R":
20         N = random.randint(1, 5)
21         S = random.randint(1, 10)
22         W = random.randint(1, 10)
23         Ps = [random.randint(1, 10) for _ in range(N)]
24         Ss = [random.randint(1, 10) for _ in range(N)]
25         Ws = [random.randint(1, 10) for _ in range(N)]
26         return N, S, W, Ps, Ss, Ws
27     else:
28         raise ValueError("problem must be P, Q or R")
29
30
31 def slowP(N, S, Ps, Ss):
32     """全探索による
33     bit(P)の解法解法を説明すると、
34
35
36     1.  $0 \sim 2^N$ までの-1を生成するbit これは、各がかどうかを表すx_i1
37     2. それに対応するについての制約条件を計算するbitx
38     3. 制約条件を満たすならば、それに対応するについての目的関数の値を計算するbitx
39     4. で計算した値の最大値を答えとする3.というものである。これは、全探索の計算量が
40
41
42     bit0( $2^N$ )であり、指数時間アルゴリズムとなっている。
43     """
44     ans = -np.inf
45     for bit in range(1 << N):
46         constraint = sum([Ss[i] * bool(bit & (1 << i)) for i in range(N)])
47         if constraint <= S:
48             obj = sum([Ps[i] * bool(bit & (1 << i)) for i in range(N)])
49             ans = max(ans, obj)
50     return ans
51
52
53 def slowQ(N, S, Ps, Ss):
54     """全探索による
55     (Q)の解法上とほぼ同様
56
57
58     """
59     ans = -np.inf

```

```

60     for Xs in product(range(0, 10 + 1), repeat=N):
61         constraint = sum([Ss[i] * Xs[i] for i in range(N)])
62         if constraint <= S:
63             obj = sum([Ps[i] * Xs[i] for i in range(N)])
64             ans = max(ans, obj)
65     return ans
66
67
68 def slowR(N, S, W, Ps, Ss, Ws):
69     """全探索による
70     (R)の解法上とほぼ同様
71
72
73     """
74     ans = -np.inf
75     for bit in range(1 << N):
76         constraint1 = sum([Ss[i] * bool(bit & (1 << i)) for i in
77                             range(N)])
78         constraint2 = sum([Ws[i] * bool(bit & (1 << i)) for i in
79                             range(N)])
80         if constraint1 <= S and constraint2 <= W:
81             obj = sum([Ps[i] * bool(bit & (1 << i)) for i in
82                         range(N)])
83             ans = max(ans, obj)
84     return ans
85
86
87 def solveP():
88     """
89     (1),(2)の解答これは
90     p1...,pn, s1...,sn について多項式時間, S について指数時間アルゴリズム
91     である。
92
93     """
94     N, S, Ps, Ss = makeProblem("P")
95     print(f"{N=},{S=},{Ps=},{Ss=}")
96     print(f"{slowP(N, S, Ps, Ss)=}")
97
98     As = [[None for _ in range(S + 1)] for _ in range(N + 1)]
99     for s in range(1, S + 1):
100         As[0][s] = -np.inf
101         As[0][0] = 0
102
103     for j in range(1, N + 1):
104         for s in range(S + 1):
105             if s < Ss[j - 1]:
106                 As[j][s] = As[j - 1][s]
107             else:
108                 As[j][s] = max(As[j - 1][s], Ps[j - 1] + As[j -

```

```

104         1][s - Ss[j - 1]])
105
106     print("As=", *As, sep="\n")
107     print("ans=", max(max(a) for a in As))
108
109 def solveQ():
110     """
111     (3)の解答これは
112     p1...,pn, s1...,sn について多項式時間, S について指数時間アルゴリズム
113     である。
114     """
115     N, S, Ps, Ss = makeProblem("Q")
116     print(f"{N=},{S=},{Ps=},{Ss=}")
117     print(f"{slowQ(N, S, Ps, Ss)=}")
118
119     As = [[None for _ in range(S + 1)] for _ in range(N + 1)]
120     for s in range(1, S + 1):
121         As[0][s] = -np.inf
122     As[0][0] = 0
123
124     for j in range(1, N + 1):
125         for s in range(S + 1):
126             # ここが漸化式
127             if s < Ss[j - 1]:
128                 As[j][s] = As[j - 1][s]
129             else:
130                 As[j][s] = max(
131                     As[j - 1][s],
132                     # 以下が増えたこれは、を x_i1の範囲内で変化させてい
133                     # る~10
134                     max(
135                         Ps[j - 1] * x + As[j - 1][s - Ss[j - 1] *
136                             x]
137                         for x in range(1, 10 + 1)
138                         if s - Ss[j - 1] * x >= 0
139                     ),
140                 )
141
142     # print("As=", *As, sep="\n")
143     print("ans=", max(max(a) for a in As))
144
145 def solveR():
146     """
147     (4)の解答これは
148     p1...,pn, s1...,sn, w1...,wn について多項式時間, S について指数時間
149     アルゴリズムである。

```

```

147     """
148     N, S, W, Ps, Ss, Ws = makeProblem("R")
149     print(f"{N=},{S=},{W=},{Ps=},{Ss=},{Ws=}")
150     print(f"{slowR(N, S, W, Ps, Ss, Ws)=}")
151
152     # の引数にを追加すれば良いAW
153     As = [[[None for _ in range(W + 1)] for _ in range(S + 1)]
154            for _ in range(N + 1)]
155     for s in range(S + 1):
156         for w in range(W + 1):
157             if s != 0 or w != 0:
158                 As[0][s][w] = -np.inf
159     As[0][0][0] = 0
160
161     for j in range(1, N + 1):
162         for s in range(S + 1):
163             for w in range(W + 1):
164                 if s < Ss[j - 1] or w < Ws[j - 1]:
165                     As[j][s][w] = As[j - 1][s][w]
166                 else:
167                     As[j][s][w] = max(
168                         As[j - 1][s][w],
169                         Ps[j - 1] + As[j - 1][s - Ss[j - 1]][w -
170                             Ws[j - 1]],
171                     )
172
173     # print("As=", *As, sep="\n")
174     print("ans=", max(max(max(a) for a in aa) for aa in As))
175
176 def main():
177     solveP()
178     print("=" * 10)
179     solveQ()
180     print("=" * 10)
181     solveR()
182
183 if __name__ == "__main__":
184     main()

```

Listing 2 output

```

1 N=5,S=5,Ps=[2, 3, 2, 1, 3],Ss=[2, 3, 1, 2, 1]
2 slowP(N, S, Ps, Ss)=8
3 As=
4 [0, -inf, -inf, -inf, -inf, -inf]
5 [0, -inf, 2, -inf, -inf, -inf]

```

```

6  [0, -inf, 2, 3, -inf, 5]
7  [0, 2, 2, 4, 5, 5]
8  [0, 2, 2, 4, 5, 5]
9  [0, 3, 5, 5, 7, 8]
10 ans= 8
11 =====
12 N=3,S=3,Ps=[7, 10, 4],Ss=[1, 4, 1]
13 slowQ(N, S, Ps, Ss)=21
14 ans= 21
15 =====
16 N=4,S=2,W=8,Ps=[5, 7, 7, 10],Ss=[2, 8, 7, 2],Ws=[6, 2, 6, 10]
17 slowR(N, S, W, Ps, Ss, Ws)=5
18 ans= 5

```

3 知識

特に無し