Vol.2018-AL-168 No.5
2018/5/25

L1 距離制約をもつ分離凸資源配分問題に対するアルゴリズム

南川 智都[1,a)]　塩浦 昭義[1,b)]

概要：分離凸資源配分問題とは，複数の活動に対して有限個の離散的な資源を割り当てる問題であり，凸関数で表される経費・損失が最小になるように割り当てることを目的とする．本稿では，所与のベクトルから解ベクトルへの L1 距離が定数以下，という制約の下での分離凸資源配分問題を考え，厳密解を求めるアルゴリズムを提案する．まず，L1 距離制約の下での最も単純な分離凸資源配分問題がポリマトロイド制約付き資源配分問題の特殊ケースと見なせることを示す．この結果に基づき，最適解が多項式時間で求められることを示す．さらに，より一般的な資源配分問題に L1 距離制約を加えた場合，ポリマトロイド制約付き資源配分問題の枠組みに当てはまらない事を具体例により示す．

1. はじめに

資源配分問題とは，いくつかの離散的な資源を複数の活動へ最適に配分する方法を求める問題である．各活動に配分された資源に応じた経費・損失が生じる状況下で，その和を最小化することを目的とする．この問題に対する研究は，1953 年に行われた Koopman [9] の研究からはじまり，半世紀以上の間，様々な研究が行われてきた．資源・制約として様々なものが考えられることから，投資計画，要員計画，生産計画，最適軍備計画などの多岐に渡る応用例がある．

本稿では，各活動の目的関数が配分量のみに依存し，かつ凸関数の場合，すなわち分離凸関数で表されるケースを扱う．目的関数が分離凸関数となる資源配分問題は，分離凸資源配分問題とよばれる．とくに，配分される資源の総量以外に制約がない，最も単純な分離凸資源配分問題を**単純資源配分問題**とよぶ．N 個の資源を n 個の活動に配分する単純資源配分問題は，以下のように定式化される．

$$\begin{aligned} \text{Minimize} \quad & \sum_{i=1}^{n} f_i(x_i) \\ \text{subject to} \quad & \sum_{i=1}^{n} x_i = N, \\ & x \in \mathbb{Z}_+^n. \end{aligned}$$

ここで，$f_i : \mathbb{R} \to \mathbb{R}$ は凸関数 $(i = 1, \ldots, n)$ であり，$\mathbb{Z}_+^n$ は各成分が非負整数の n 次元ベクトルの集合を表す．また，$N \geq n$ とする．各 f_i は陽に与えられるか，または与えられた引数 x_i に対して関数値 $f_i(x_i)$ を定数時間で返す関数値評価オラクルで与えられるとする．

単純資源配分問題を解く基本的なアルゴリズムとして，Gross [6] による貪欲アルゴリズムが知られている．このアルゴリズムの時間計算量は $\mathrm{O}(N \log n)$ である．単純資源配分問題の入力サイズは $\mathrm{O}(n + \log N)$ であり，Gross のアルゴリズムは入力サイズの多項式時間で抑えられない．そのため，入力の数値が大きくなると，実行に時間がかかってしまうという問題がある．Galil, Megiddo [5] と加藤ら [8] は，同時期に，入力サイズの多項式時間で動作するアルゴリズムを開発した．この 2 つの研究におけるアルゴリズムの時間計算量は，共に $\mathrm{O}(n(\log N)^2)$ である．現在知られている最も高速なアルゴリズムは，Frederickson, Johnson [2] によるものであり，$\mathrm{O}(n \log(N/n))$ 時間で最適解を求める．この $\mathrm{O}(n \log(N/n))$ という時間計算量は，一般的な計算モデルにおいては，これ以上改善できないことが証明されている [7].

単純資源配分問題は最も基本的な資源配分問題であるが，一般化上界制約，入れ子制約，木制約，ネットワーク制約などの制約が追加された，より一般的な資源配分問題についても，これまで扱われてきた．これらの制約の共通の一般化として知られるのがポリマトロイド制約であり，ポリマトロイド制約をもつ分離凸資源配分問題は劣モジュラ資源配分問題とよばれる．劣モジュラ資源配分問題に対しては，多項式時間で動作するスケーリングアルゴリズムが Hochbaum [7] によって提案されている (森口，塩浦 [10] も参照).

本稿では，単純資源配分問題に L1 距離制約を追加した分離凸資源配分問題を考える．L1 距離制約とは，所与の

[1] 東京工業大学工学院　経営工学系
東京都目黒区大岡山 2-12-1
[a)] minamikawa.n.aa@m.titech.ac.jp
[b)] shioura.a.aa@m.titech.ac.jp

ベクトルから解ベクトルへの L1 距離が定数以下，という制約であり，L1 距離制約付き分離凸資源配分問題 (以下 L1SRA とよぶ) は以下のように定式化される．

$$\begin{aligned} \text{Minimize} \quad & \sum_{i=1}^{n} f_i(x_i) \\ \text{subject to} \quad & \sum_{i=1}^{n} x_i = N, \\ & \|x-y\|_1 \le K, \\ & x \in \mathbb{Z}_+^n. \end{aligned}$$

ここで K は非負の整数，y は $\sum_{i=1}^{n} y_i = N$ を満たす n 次元非負整数ベクトルである．また，$E = \{1, 2, \ldots, n\}$ とおく．

L1 距離制約をもつ資源配分問題は，与えられた資源を再配分する際に現れる．当初の資源配分量がベクトル y で与えられ，その後，資源の再配分が必要とされるとき，新たな資源配分量と当初の資源配分量をできるだけ近い値にしたい，という要請がしばしばある．そのような条件を表現しているのが，L1 距離を用いた制約 $\|x-y\|_1 \le K$ である．このような制約を用いている例として，バイクシェアリングサービスにおける自転車の配置 [3] が挙げられる．

本稿の目的は，問題 L1SRA の組合せ構造を明らかにし，その多項式時間可解性を示すことである．そのために，L1SRA が劣モジュラ資源配分問題として定式化できることを示す．劣モジュラ資源配分問題は多項式時間で解けることが知られているので，この結果より，L1SRA の多項式時間可解性が直ちに導かれる．さらに本稿では，劣モジュラ資源配分問題に対する既存のアルゴリズムである貪欲アルゴリズムとスケーリングアルゴリズムを L1SRA に適用し，それぞれ $O(N \log n)$ 時間および $O(n \log n \log(N/n))$ 時間の実装が可能であることを示す．

また本稿では，L1SRA より一般的な問題として，単純資源配分問題より一般的な資源配分問題に L1 距離制約を加えた問題を考える．このような問題は劣モジュラ資源配分問題として定式化できないことを具体例により示す．

本稿ではさらに，L1SRA に関連する問題として，L1SRA から非負制約を削除した問題 (以下 L1SRA$^-$ とよぶ) について議論する．L1SRA$^-$ についても，劣モジュラ関数を用いた制約付きの資源配分問題として定式化可能であることを示すとともに，貪欲アルゴリズムとスケーリングアルゴリズムを適用できることを示す．

2. 資源配分問題

1 節で述べた単純資源配分問題は，資源配分の総量にのみ制約がある，最も基本的な資源配分問題である．本節では，単純資源配分問題より一般的な資源配分問題を紹介する．また，資源配分問題に対する既存のアルゴリズムを説明する．

一般化上界制約においては，問題の変数に対する分割が与えられ，それぞれの変数集合に対して，変数の値の和に対する上界が与えられる．つまり，E の分割 $\{S_1, S_2, \ldots, S_m\}$ および非負整数 $b_1, b_2, \ldots, b_m$ に対し，$x(S_j) \le b_j$ $(j = 1, 2, \ldots, m)$ と表される制約が一般化上界制約である．ここで $S \subseteq E$ に対して $x(S) = \sum_{i \in S} x_i$ とおく．一般化上界制約付き資源配分問題は以下の通り定式化される．

$$\begin{aligned} \text{Minimize} \quad & \sum_{i=1}^{n} f_i(x_i) \\ \text{subject to} \quad & \sum_{i=1}^{n} x_i = N, \\ & x(S_j) \le b_j \quad (j = 1, 2, \ldots, m), \\ & x \in \mathbb{Z}_+^n. \end{aligned}$$

一般化上界制約より一般的な制約として，入れ子制約や木制約が知られている．さらに，これらの制約は全て劣モジュラ性をもつ集合関数によって表される制約の特殊ケースとみなすことができる．集合関数 $f : 2^E \to \mathbb{Z}$ が**劣モジュラ**であるとは，**劣モジュラ不等式**

$$f(S) + f(T) \ge f(S \cup T) + f(S \cap T) \quad (\forall S, T \in 2^E) \tag{1}$$

を満たすことをいう．また，$S \subseteq T$ を満たす任意の $S, T \in 2^E$ が不等式 $f(S) \le f(T)$ を満たすとき，集合関数 f は**単調非減少**であるという．**ポリマトロイド制約**とは，$\rho(\emptyset) = 0$ を満たす単調非減少な劣モジュラ関数 ρ について

$$x(S) \le \rho(S) \quad (\forall S \in 2^E) \tag{2}$$

となる制約のことである．単純資源配分問題にポリマトロイド制約を加えた問題を，**劣モジュラ資源配分問題**という．劣モジュラ資源配分問題においては，一般性を失うことなく $\rho(E) = N$ とする．

以下では，劣モジュラ資源配分問題に対する基本的なアルゴリズムとして，貪欲アルゴリズム [1] とスケーリングアルゴリズム [7] を紹介する．これらのアルゴリズムは 3 節以降で L1SRA を解くアルゴリズムとして利用される．

最初に貪欲アルゴリズム **procedure** GREEDY を説明する (Algorithm 1 参照)．各 $i \in E$ および $x_i \in \mathbb{Z}_+$ に対して，関数 f_i の増分を

Algorithm 1 procedure GREEDY

```
x := (0, 0, ..., 0)^⊤, E' := E, k := 0;
while k ≤ N do
    d_j(x_j + 1) を最小化する j = j* を求める;
    if x + e(j*) はポリマトロイド制約を満たす then
        x_{j*} := x_{j*} + 1, k := k + 1;
    else
        E' := E' − {j*};
    end if
end while
return x := x*;
```

Algorithm 2 procedure SM-INCREMENT(s, l)

```
1: x := l, E' := E, δ := 0;
2: while E' ≠ ∅ do
3:     d_j(x_j + 1) を最小にする j = j* を求める;
4:     if x + s·e(j*) はポリマトロイド制約を満たす then
5:         x := x + s·e(j*);
6:     else if x + e(j*) はポリマトロイド制約を満たす then
7:         α := max{α' | x + α'·e(j*)
8:                         はポリマトロイド制約を満たす };
9:         x := x + α·e(j*), E' := E' − {j*};
10:    end if
11: end while
12: return x^(s) := x;
```

$$d_i(x_i) = f_i(x_i) - f(x_i - 1)$$

とおく．また，$e(j^*)$ は第 j^* 成分のみ 1 の単位ベクトルを表す．このアルゴリズムでは，原点 $x = (0, 0, \ldots, 0)^\top$ を初期解とし，各反復では，ベクトル x がポリマトロイド制約を満たしたまま，目的関数値の増分 $d_i(x_i)$ が最小となるようにベクトルのある成分 x_i を 1 増加させる．この手順を，ベクトル x の成分和が N となるまで繰り返す．このアルゴリズムの時間計算量は $\mathrm{O}(N(\log n + F))$ となる．ここで，F は所与の解がポリマトロイド制約を満たすかどうかの判定に必要な時間を表す．

次に，劣モジュラ資源配分問題を解くスケーリングアルゴリズム [7] について述べる．整数格子点上で定義された関数 $f : \mathbb{Z}^n \to \mathbb{R}$ の定義域上の点を飛び飛びに取って，大雑把に表現した関数は，元の関数と形が似ているため，最小解が近くにある可能性が高い．この性質を利用し，貪欲アルゴリズムでは 1 ずつ増加させていた変数を，複数単位ずつ増加させることを考える．

このアルゴリズムはメインルーチンである **procedure** SCALING とサブルーチンである **procedure** SM-INCREMENT(s, l) によって構成される (Algorithm 2,3 参照)．**procedure** SM-INCREMENT(s, l) は初期解を l としてベクトルのある成分を s 単位ずつ増加させる．**procedure** SCALING は，**procedure** SM-INCREMENT(s, l) を繰り返し実行する．その際，初期ステップサイズを $\lceil N/2n \rceil$ として，各反復ではステップサイズ s を減らしつつ，l を更新していく．**procedure** SCALING の時間計算量は $\mathrm{O}(n(\log n + F)\log(N/n))$ である．

最後に，劣モジュラ資源配分問題と深く関連する問題が前出の貪欲アルゴリズムやスケーリングアルゴリズムで解けることを説明する．(単調非減少とは限らない) $\rho(\emptyset) = 0$ を満たす劣モジュラ関数 $\rho : 2^E \to \mathbb{Z}$ を用いて (2) の形で表現される制約を**劣モジュラ制約**とよぶ．劣モジュラ制約をもつ以下の問題を，本稿では**劣モジュラ制約付き分離凸最小化問題**とよぶ．

Algorithm 3 procedure SCALING

```
1: s := ⌈N/2n⌉, l := (0, 0, ..., 0)^⊤;
2: while s ≥ 2 do
3:     SM-INCREMENT(s, l) を実行し，出力を x^(s) とする．
4:     各 i ∈ E について l_i := max{x_i^(s) − s, 0} とする．
5:     s := ⌈s/2⌉;
6: end while
7: SM-INCREMENT(1, l) を実行し，出力を x* とする．
8: return x*;
```

$$\begin{aligned} \text{Minimize} \quad & \sum_{i=1}^{n} f_i(x_i) \\ \text{subject to} \quad & x(E) = \rho(E), \\ & x(S) \le \rho(S) \quad (\forall S \in 2^E), \\ & x \in \mathbb{Z}^n. \end{aligned}$$

劣モジュラ制約付き分離凸最小化問題においては，ベクトルの非負条件がないことに注意する．

劣モジュラ制約付き分離凸最小化問題についても，アルゴリズムの初期解を適切に変更することで，前述の貪欲アルゴリズムやスケーリングアルゴリズムによって解けることが知られている ([4] など参照)．初期解としては，ある最適解 x^* の下界となるベクトル b (つまり，$x^* \ge b$ を満たすベクトル) を用いればよい．ベクトル b の候補としては，劣モジュラ制約付き分離凸最小化問題のすべての実行可能解 x に対して $x \ge b$ を満たすベクトル，たとえば

$$b_i = \rho(E) - \rho(E \setminus \{i\}) \qquad (i \in E) \qquad (3)$$

が挙げられる．貪欲アルゴリズムやスケーリングアルゴリズムの時間計算量はパラメータ N の代わりに $\tilde{N} = \rho(E) - b(E)$ に依存し，それぞれ $\mathrm{O}(\tilde{N}(\log n + F))$ および $\mathrm{O}(n(\log n + F)\log(\tilde{N}/n))$ となる．ここで，F は所与の解が劣モジュラ制約を満たすかどうかの判定に必要な時間を表す．

3. L1SRA の構造と解法

本節では L1SRA が劣モジュラ資源配分問題として定式化できることを証明するとともに．この問題が効率的に解けることを示す．$k = \lfloor (K/2) \rfloor$ とおき，集合関数 $\rho : 2^E \to \mathbb{Z}$ を次のように定義する．

$$\rho(S) = \begin{cases} 0 & (S = \emptyset \text{ のとき}), \\ N & (S = E \text{ のとき}), \\ \min(N, y(S) + k) & (\text{それ以外}). \end{cases}$$

以下の定理は，L1SRA がこの ρ に関する劣モジュラ資源配分問題として定式化できることを述べている．

定理 1. (i) ρ は単調非減少劣モジュラ関数である．
(ii) L1 距離制約付き分離凸資源配分問題の許容解領域 $R \subseteq \mathbb{Z}^n$ は，以下の $R_\rho \subseteq \mathbb{Z}^n$ に等しい：

$$R_\rho = \{x \in \mathbb{Z}_+^n \mid x(S) \le \rho(S)\ (\forall S \in 2^E),\ x(E) = N\}.$$

証明. [(i) の証明] 定義から明らかに ρ は単調非減少である．ρ の劣モジュラ性を示すために，任意の 2 つの集合 $S,T \in 2^E$ に対して劣モジュラ不等式 (1) を満たすことを証明する．S または T が $\emptyset$ または E のときは劣モジュラ不等式は自明に成り立つので，以下では S および T がともに非空な E の真部分集合の場合について考える．

まず $f(S) = N$ の場合は，$f(S \cup T) = N$ であるから

$$\begin{aligned} &f(S) + f(T) = N + \min(N, y(T) + k) \\ &\ge N + \min(N, y(S \cap T) + k) = f(S \cup T) + f(S \cap T) \end{aligned}$$

が成り立つ．$f(T) = N$ の場合も同様である．次に，$f(S) = y(S) + k$ かつ $f(T) = y(T) + k$ のときは，

$$\begin{aligned} &f(S) + f(T) \\ &= (y(S) + k) + (y(T) + k) \\ &= (y(S \cup T) + k) + (y(S \cap T) + k) \\ &\ge \min(N, y(S \cup T) + k) + \min(N, y(S \cap T) + k) \\ &= f(S \cup T) + f(S \cap T) \end{aligned}$$

となる．よって，ρ は劣モジュラ不等式を満たす．

[(ii) の証明] まず，任意の $x \in R$ に対して $x \in R_\rho$ が成り立つことを示す．R は L1SRA の実行可能解なので，x は $x(E) = N$ を満たす非負ベクトルである．よって，あとは x がポリマトロイド制約 $x(S) \le \rho(S)$ を満たすことを示せば良い．ポリマトロイド制約は $S = \emptyset$ および $S = E$ のときは自明に成り立つので，以下では $\emptyset \subset S \subset E$ の場合を考え，$x(S) \le \rho(S) = \min(N, y(S) + k)$ が成り立つことを証明する．

この不等式は 2 つの不等式 $x(S) \le N$ および $x(S) \le y(S) + k$ と等価である．前者の不等式は $x(E) = N$ であることから導かれる．後者の不等式 $x(S) \le y(S) + k$ を示すために，$x \in R$ が満たす L1 距離制約 $\|x - y\|_1 \le K$ に注目する．$x(E) = y(E)$ であることから $\|x - y\|_1$ は偶数であり，$\|x - y\|_1 \le 2\lfloor (K/2) \rfloor = 2k$ が成り立つ．また，ベクトル $x - y$ の成分和はゼロであるので，

$$\sum_{i \in E} \max(0, x(i) - y(i)) = \sum_{i \in E} \max(0, -x(i) + y(i))$$

が成り立つ．よって，

$$\begin{aligned} 2k &\ge \|x - y\|_1 \\ &= \sum_{i \in E} \max(0, x(i) - y(i)) + \sum_{i \in E} \max(0, -x(i) + y(i)) \\ &= 2 \sum_{i \in E} \max(0, x(i) - y(i)) \end{aligned}$$

が得られる．この不等式より，

$$\begin{aligned} x(S) - y(S) &\le \sum_{i \in S} \max(0, x(i) - y(i)) \\ &\le \sum_{i \in E} \max(0, x(i) - y(i)) \le k \end{aligned}$$

が導かれる．よって，$x \in R_\rho$ が示された．

次に，任意の $x \in R_\rho$ に対して $x \in R$ が成り立つこと，つまり x が L1SRA のすべての制約を満たすことを示す．$x \in R_\rho$ なので， L1SRA の制約のうち，L1 距離制約以外の制約が成り立つことは容易に分かる．以下では，x が L1 距離制約を満たすことを示す．

集合 $S_+, S_- \subseteq E$ を

$$S_+ = \{i \in E \mid x_i \ge y_i\}, \quad S_- = \{i \in E \mid x_i < y_i\}$$

と定義する．$S_+ \cap S_- = \emptyset$ および $S_+ \cup S_- = E$ であることに注意する．また，$x(E) = y(E)$ なので，S_+ は必ず非空となる．このとき，

$$\begin{aligned} \|x - y\|_1 &= \sum_{i \in S_+} (x_i - y_i) - \sum_{j \in S_-} (x_j - y_j) \\ &= (x(S_+) - y(S_+)) - (x(S_-) - y(S_-)) \\ &= x(S_+) - x(S_-) - y(S_+) + y(S_-) \end{aligned}$$

が成り立つ．ここで $x(S_+) + x(S_-) = y(S_+) + y(S_-) = N$ であることより，等式

$$\|x - y\|_1 = 2x(S_+) - 2y(S_+) \tag{4}$$

を得る．この等式を用いて $\|x - y\|_1 \le K$ を示す．

$S_+ = E$ のときは $x(E) = y(E) = N$ なので，(4) より $\|x - y\|_1 = 0 \le K$ となる．つまり，x は L1 距離制約を満たす．次に，$S_+ \ne E$ の場合を考える．x は ρ に関するポリマトロイド制約を満たすので，

$$\begin{aligned} x(S_+) \le \rho(S_+) &= \min\left(N, y(S_+) + k\right) \\ &= y(S_+) + \min\left(N - y(S_+), k\right) \end{aligned}$$

を得る．よって (4) より，

$$\|x - y\|_1 \le 2 \min\left(N - y(S_+), k\right) \le 2k \le K.$$

よって x は L1 距離制約を満たす．これにより，$x \in R_\rho$ が示された． □

定理 1 より，2 節で紹介した貪欲アルゴリズムおよびスケーリングアルゴリズムを L1SRA に適用できることがわかる．以下では，これらのアルゴリズムを L1SRA に特化すると，$O(N \log n)$ および $O(n \log n \log(N/n))$ という時間計算量を実現できることを示す．そのためには，アルゴリズムの各反復で得られたベクトル x が，ポリマトロイド制約式 (2) を満たすかどうかの判定が定数時間で行えることを示せば良い．

制約 $x(E) \le \rho(E)$ を定数時間でチェックするには，アルゴリズムの実行時に常に $x(E)$ の値を保持していれば可能である．集合 S が非空な E の真部分集合の場合には，ポリマトロイド制約の式 $x(S) \le \rho(S) = \min(N, y(S)+k)$ $(\emptyset \subset \forall S \subset E)$ をチェックするには，2 つの不等式 $x(S) \le N$ と $x(S) \le y(S)+k$ をすべての S に対してチェックすればよい．ベクトル x は非負なので，制約 $x(E) \le \rho(E)$ が満たされていれば，$x(S) \le N$ は自動的に満たされる．一方，不等式 $x(S) \le y(S)+k$ は

$$\sum_{i \in S}(x_i - y_i) \le k$$

と書けることに注意する．左辺の最大値は $\sum_{i:x_i>y_i}(x_i - y_i)$ となるので

$$\sum_{i:x_i>y_i}(x_i - y_i) \le k$$

が成り立つかどうかを調べればよい．この式の成立を調べるために，各反復で $\sum_{i:x_i>y_i}(x_i - y_i)$ の値を保持しておけばよい．$y(E) = N$ であり，アルゴリズムの実行中には $x(E) \le N$ が成り立つので，集合 $S^* = \{i \in E \mid x_i > y_i\}$ は全体集合 E にはならないことに注意する．ベクトル x が更新された際の値 $\sum_{i:x_i>y_i}(x_i - y_i)$ の更新を定数時間で行うことは難しくない．

したがって，アルゴリズム実行時にベクトル x がポリマトロイド制約を満たすかどうかの判定は $F = \mathrm{O}(1)$ 時間で行うことができる．よって，貪欲アルゴリズムとスケーリングアルゴリズムの時間計算量は，それぞれ $\mathrm{O}(N \log n)$ および $\mathrm{O}(n \log n \log(N/n))$ となる．

本節の最後に，L1SRA より一般的な問題として，単純資源配分問題より一般的な資源配分問題に L1 距離制約を加えた問題を考える．そのような問題は劣モジュラ資源配分問題として定式化できないことを具体例により示す．

証明のために，一般化上界制約付き資源配分問題に L1 距離制約を追加した問題を考える．この問題は以下のように定式化される (2 節参照)．

$$\begin{aligned} \text{Minimize} \quad & \textstyle\sum_{i=1}^{n} f_i(x_i) \\ \text{subject to} \quad & x(E) = N, \\ & \|x - y\|_1 \le K, \\ & x(S_j) \le b_j \quad (j = 1, 2, \ldots, m), \\ & x \in \mathbb{Z}_+^n. \end{aligned}$$

この問題の (実行可能領域の) 具体例として，$n = 4$ の場合の以下の例を考える．

$$\begin{aligned} & x_1 + x_2 + x_3 + x_4 = 4, \\ & |x_1 - 1| + |x_2 - 1| + |x_3 - 1| + |x_4 - 1| \le 2, \\ & x_1 + x_2 \le 2,\ x_3 + x_4 \le 4, \\ & x \in \mathbb{Z}_+^4. \end{aligned}$$

この実行可能領域がポリマトロイド制約で表現できるならば，

$$\rho(S) = \max\{x(S) \mid x \text{ は実行可能解 }\} \quad (S \in 2^E)$$

で定義される集合関数 ρ は劣モジュラ関数になるはずである [4]．

3 番目の制約式より

$$\rho(\{1,2\}) \le 2$$

となる．$(0,2,1,1)$ および $(1,1,2,0)$ が実行可能解であることから，

$$\rho(\{2\}) = 2,\ \rho(\{2,3\}) = 3,\ \rho(\{1,2,3\}) = 4$$

が成り立つことが確かめられる．したがって，$S = \{1,2\}, T = \{2,3\}$ に対して

$$\rho(S) + \rho(T) \le 5 < 6 = \rho(S \cup T) + \rho(S \cap T)$$

となり，ρ は劣モジュラ不等式 (1) を満たさない．このことから，この問題は劣モジュラ資源配分問題として定式化できない．

4. 非負制約のない L1SRA の構造と解法

本節では L1SRA に密接に関連した問題として，非負制約を削除した下記の問題 L1SRA$^-$

$$\begin{aligned} \text{Minimize} \quad & \sum_{i=1}^{n} f_i(x_i) \\ \text{subject to} \quad & \sum_{i=1}^{n} x_i = N, \\ & \|x - y\|_1 \le K, \\ & x \in \mathbb{Z}^n \end{aligned}$$

について考える．ここで N は (非負とは限らない) 整数，K は非負の整数，y は $\sum_{i=1}^{n} y_i = N$ を満たす n 次元の (非負とは限らない) 整数ベクトルである．本節では，問題 L1SRA$^-$ が劣モジュラ制約つき分離凸最小化問題として定式化できることを示す．さらに，この事実をふまえ，問題 L1SRA$^-$ が多項式時間で解けることを示す．

L1SRA$^-$ の再定式化のために，集合関数 $\mu : 2^E \to \mathbb{Z}$ を次のように定義する．

$$\mu(S) = \begin{cases} 0 & (S = \emptyset \text{ のとき}), \\ N & (S = E \text{ のとき}), \\ y(S) + k & (\text{それ以外}). \end{cases}$$

この関数 μ の劣モジュラ性は，3 節の関数 ρ の劣モジュラ性と同様に証明できる．ρ と異なり，μ は単調非減少ではないことに注意する．

また，3 節で用いた証明と同様のやり方で，関数 μ に関する劣モジュラ制約および制約 $x(E) = \mu(E)$ を満たす解

集合が L1SRA$^-$ の実行可能領域に一致することを証明できる．つまり，L1SRA$^-$ は劣モジュラ制約付き分離凸最小化問題として定式化できることがわかる．

この事実および 2 節で述べたことから，L1SRA$^-$ に対しても貪欲アルゴリズムとスケーリングアルゴリズムを適用可能である．その際，ある最適解 x^* の下界となるベクトル b を初期解として用いる必要がある．例えば，式 (3) で与えられる b を用いると，

$$\begin{aligned} b_i &= N - \{y(E \setminus \{i\}) + k\} \\ &= N - \{y(E) - y(i) + k\} \\ &= N - N + y(i) - k = y(i) - k \end{aligned}$$

となる．したがって

$$\tilde{N} = \rho(E) - b(E) = \rho(E) - y(E) + nk = nk = \mathrm{O}(nK)$$

であり，貪欲アルゴリズムとスケーリングアルゴリズムの時間計算量はそれぞれ $\mathrm{O}((n \log n)K)$ および $\mathrm{O}(n \log n \log K)$ となる．この時間計算量を削減するために，以下では $b(E)$ の値がより小さい b を求めることを目指す．

補題 2. 以下の最適化問題 SRA$^-$ の最適解を x' とする．

$$\begin{aligned} \text{Minimize} \quad & \sum_{i=1}^{n} f_i(x_i) \\ \text{subject to} \quad & \sum_{i=1}^{n} x_i = N - k, \\ & x \le y, \\ & x \in \mathbb{Z}^n. \end{aligned}$$

このとき，$x^* \ge x'$ を満たす L1SRA$^-$ の最適解 x^* が存在する．

証明. ベクトル x^* は，L1SRA$^-$ の最適解であり，すべての最適解の中で $\sum_{i=1}^{n} \max(0, x'_i - x^*_i)$ の値が最大であるとする．ここで $\sum_{i=1}^{n} \max(0, x'_i - x^*_i) \le 0$ ならば $x^* \ge x'$ が成り立つので，$\sum_{i=1}^{n} \max(0, x'_i - x^*_i) > 0$ と仮定して矛盾を導く．

仮定より，$x'_h > x^*_h$ となる $h \in E$ が存在する．また，

$$\sum_{i=1}^{n} x'_i = N - k < N = \sum_{i=1}^{n} x^*_i$$

が成り立つので，$x'_j < x^*_j$ となる $j \in E$ が存在する．ここで

$$S_- = \{i \in E \mid x'_i < x^*_i\}, \quad S_+ = \{i \in E \mid x'_i \ge x^*_i\}$$

とおく．以下では，$x'_j < y_j$ を満たす $j \in S_-$ が存在することを示す．

すべての $i \in S_-$ に対して $x'_i \ge y_i$ が成り立つとする．x' は SRA$^-$ の実行可能解なので，$x'_i \ge y_i$ は等号で成り立つ．よって，$x'_i < y_i$ ならば $i \in S_+$ となる．このことより，

$$\begin{aligned} \sum_{i=1}^{n} (y_i - x'_i) &= \sum_{i \in S_+} (y_i - x'_i) \\ &< \sum_{i \in S_+} (y_i - x^*_i) \\ &\le \sum_{i \in E} \max(0, y_i - x^*_i) \end{aligned}$$

が得られる．2 番目の不等式は S_+ の定義と $h \in S_+$ による．また，ベクトル x^* は L1 距離制約 $\|x^* - y\|_1 \le K$ を満たすことから，$\sum_{i=1}^{n} \max(0, y_i - x^*_i) \le k$ が成り立つ (定理 1 の証明参照)．したがって，$\sum_{i=1}^{n} (y_i - x'_i) < k$ となるが，この不等式より

$$\sum_{i=1}^{n} x'_i > \sum_{i=1}^{n} y_i - k = N - k$$

となり，x' が SRA$^-$ の実行可能解であることに矛盾する．よって，$x'_j < y_j$ を満たす $j \in S_-$ が存在する．

凸関数の性質より

$$\begin{aligned} f_h(x^*_h + 1) - f_h(x^*_h) &\le f_h(x'_h) - f_h(x'_h - 1), \\ f_j(x'_j + 1) - f_j(x'_j) &\le f_j(x^*_j) - f_j(x^*_j - 1) \end{aligned}$$

が成り立つ．これらの不等式より，

$$\begin{aligned} & f(x^*) + f(x') \\ & \ge f(x^* + e(h) - e(j)) + f(x' - e(h) + e(j)) \end{aligned}$$

が導かれる．ここで $f(x) = \sum_{i=1}^{n} f_i(x_i)$ $(x \in \mathbb{Z}^n)$ とおく．不等式 $y_h \ge x'_h > x^*_h$ が成り立つので，ベクトル $x^* + e(h) - e(j)$ は L1 距離制約を満たし，ゆえに L1SRA$^-$ の実行可能解であることがわかり，不等式 $f(x^*) \le f(x^* + e(h) - e(j))$ を得る．また，$x'_j < y_j$ なので $x' - e(h) + e(j)$ は SRA$^-$ の実行可能解であり，不等式 $f(x') \ge f(x' - e(h) + e(j))$ が得られる．以上の不等式より，$f(x^* + e(h) - e(j)) = f(x^*)$ が成り立ち，ベクトル $x^* + e(h) - e(j)$ もまた L1SRA$^-$ の最適解であることがわかるが，$x'_h > x^*_h$ であることから，これは x^* の選び方に矛盾する．よって，$x^* \ge x'$ を満たす L1SRA$^-$ の最適解 x^* が存在する． □

補題 2 より，L1SRA$^-$ は SRA$^-$ の最適解を初期解として利用することで，貪欲アルゴリズムおよびスケーリングアルゴリズムを用いて解くことができる．その際の時間計算量を算定する．

SRA$^-$ は本質的に単純資源配分問題と等価な問題であるので，SRA$^-$ の最適解は $\mathrm{O}(n \log(K/n))$ 時間で求められる．よって，SRA$^-$ の最適解を貪欲アルゴリズムおよびスケーリングアルゴリズムの初期解 b として使うと，パラメータ $\tilde{N}$ の値は

$$\tilde{N} = \rho(E) - b(E) = N - (N - k) = k = \mathrm{O}(K)$$

となるため，貪欲アルゴリズムの時間計算量は $O(K \log n) + O(n \log(K/n)) = O(K \log n + n \log(K/n))$ となり，スケーリングアルゴリズムの時間計算量は $O(n \log n \log(K/n)) + O(n \log(K/n)) = O(n \log n \log(K/n))$ となる．

参考文献

[1] A. Federgruen and H. Groenevelt. The greedy procedure for resource allocation problems: Necessary and sufficient conditions for optimality. *Operations Research*, Vol. 34, pp. 909–918, 1986.
[2] G.N. Frederickson and D.B. Johnson. The complexity of selection and ranking in $X + Y$ and matrices with sorted columns. *Journal of Computer and System Sciences*, Vol. 24, pp. 197–208, 1982.
[3] D. Freund, S.G. Henderson, and D.B. Shmoys. Minimizing multimodular functions and allocating capacity in bike-sharing systems. *Proceedings of Conference on Integer Programming and Combinatorial Optimization (IPCO)*, pp. 186–198, 2017.
[4] S. Fujishige. *Submodular Functions and Optimization, 2nd Edition.* Elsevier, Amsterdam, 2005.
[5] Z. Galil and N. Megiddo. A fast selection algorithm and the problem of optimum distribution of effort. *Journal of ACM*, Vol. 26, pp. 58–64, 1979.
[6] O. Gross. *A class of discrete type minimization problems.* Rand Corporation, Santa Monica, 1956.
[7] D.S. Hochbaum. Lower and upper bounds for the allocation problem and other nonlinear optimization problems. *Mathematics of Operations Research*, Vol. 19, pp. 390–409, 1994.
[8] N. Katoh, T. Ibaraki, and H. Mine. A polynomial time algorithm for the resource allocation problem with a convex objective function. *Journal of Operational Research Society*, Vol. 30, No. 5, pp. 449–455, 1979.
[9] O. Koopman. The optimum distribution of effort. *Journal of Operations Research Society of America*, Vol. 1, No. 2, pp. 52–63, 1953.
[10] S. Moriguchi and A. Shioura. On Hochbaum's proximity-scaling algorithm for the general resource allocation problem. *Mathematics of Operations Research*, Vol. 29, pp. 394–397, 2004.