

ある整数資源配分ゲームのシュタッケルベルク均衡

防衛大学校・情報工学科 宝崎 隆祐 (Ryusuke Hohzaki)
Department of Computer Science, National Defense Academy
防衛省・航空幕僚監部 長嶋 真一 (Shinichi Nagashima)
Air Staff Office, Ministry of Defense

1 はじめに

資源の最適配分に関する問題、いわゆる最適資源配分問題に関する多くの研究では、制約は主として資源量に課せられており、また目的関数も変数に対し分離型のものが多い[8]。資源配分問題の応用分野としては、目標割当問題や搜索問題といったものがある。目標割当問題では、1種類のみミサイルを様々な価値を持つ複数目標に割り当てた Manne の研究[5]や、それを更に複数種ミサイルに拡張した Lemus の研究[4]、Nickel の研究[6]があるが、いずれも制約がミサイル数に関するものであるため、コスト制約に比較すると取扱いが簡単である。搜索問題に関する Kadane の研究[7]は、次のような問題である。複数セルに隠れた静止目標の存在分布を知った上で、搜索者は毎回いずれかのセルを1回のみ搜索をすることができるが、搜索実施にはコストが掛かるため、限られたコスト制約の下で行う有限回の搜索において探知確率を最大にする搜索戦略を求める。Hohzaki ら[3]は、以上のような目標割当問題や搜索問題をコスト制約下での離散資源の最適配分問題へ一般化した問題を考えているが、これも1人のプレイヤーが行うべき一方的な最適化問題である。

ここで取り扱う問題はA, B 2人のプレイヤーによる先手、後手のある2人ゼロ和の離散資源配分ゲームである。この報告では、問題に対するシュタッケルベルク均衡を求める解法を提案するが、長年研究されてきた整数ナップサック問題に関する知見を利用する。

次の節でモデルの前提を厳密に記述することから始め、コスト制約と指数型の目的関数をもつ整数計画問題に定式化する。さらに、この問題がマックスミニ型の整数ナップサック問題に帰着できることを示し、3節ではその解法として3つの厳密解法と1つの近似解法を提案する。

2 モデルの記述と定式化

A, B 2人のプレイヤーによる先手、後手のある次のような資源配分ゲームを考える。

- (A1) プレイヤーAは価値 V を分割して T カ所の施設に隠そうとしている。これらの施設に $i = 1, \dots, T$ と番号を付与する。
- (A2) プレイヤーBは費用 C を使って m 種類の整数資源を手に入れた後、この搜索資源を使って T カ所の施設を探す。ただし、 j タイプの搜索資源の単価は c_j であり、その1単位を用いた搜索により施設 i に隠された価値が発見される確率は $\beta_{ij} > 0$ である。
- (A3) まずプレイヤーBが費用 C により搜索資源を準備し、プレイヤーAはそれを見て価値 V を分割して施設に隠蔽する。その後、搜索資源による施設の搜索が行われる。
- (A4) プレイヤーAは発見価値の期待値を小さくすることに、プレイヤーBは大きくすることに興味がある。

仮定にあるパラメータ C, c_j は自然数、 V, β_{ij} は実数であるとする。

問題を定式化するため、プレイヤーAの戦略を施設 i への価値の隠蔽量 $v_i \geq 0$ とする。もちろん、 $\sum_{i=1}^T v_i = V$ である。プレイヤーBの搜索資源構成戦略としてタイプ j の整数搜索資源の施設 i への割当を $x_{ij} \in \mathbb{Z}$ とすると、コスト制約は $\sum_{j=1}^m c_j \sum_{i=1}^T x_{ij} \leq C$ と表すことができる。ここで、施設の集合を $I \equiv \{1, \dots, T\}$ 、資源の

種類の集合を $J \equiv \{1, \dots, m\}$ と記すと、プレイヤー A 及び B の戦略の実行可能領域 D 及び A はそれぞれ次のようになる。

$$D = \{(v_1, \dots, v_T) \in \mathbf{R}^T \mid v_i \geq 0, i = 1, \dots, T, \sum_{i=1}^T v_i = V\} \quad (1)$$

$$A = \{\{x_{ij}\} \in \mathbf{Z}^{I \times J} \mid x_{ij} \geq 0, i = 1, \dots, T, j = 1, \dots, m, \sum_{j=1}^m c_j \sum_{i=1}^T x_{ij} \leq C\} \quad (2)$$

j タイプ探索資源 1 単位による施設 i での発見確率は $0 < \beta_{ij} \leq 1$ であるため、 $\alpha_{ij} > 0$ なる係数を使って $\beta_{ij} \equiv 1 - \exp(-\alpha_{ij})$ と表記できる。施設 i に投入された資源 $\{x_{ij}, j \in J\}$ による発見確率は $1 - \prod_{j=1}^m (1 - \beta_{ij})^{x_{ij}} = 1 - \exp(-\sum_{j=1}^m \alpha_{ij} x_{ij})$ となる。故に、発見できる期待価値は

$$\sum_{i=1}^T v_i \left(1 - \exp\left(-\sum_{j=1}^m \alpha_{ij} x_{ij}\right) \right) = V - \sum_{i=1}^T v_i \exp\left(-\sum_{j=1}^m \alpha_{ij} x_{ij}\right)$$

となる。右辺第 1 項は定数であるから、以後右辺第 2 項の示す未発見価値の期待値 $E(v, x)$ を支払関数として議論してゆく。ただし、 $v = \{v_i, i \in I\}$, $x = \{x_{ij}, i \in I, j \in J\}$ であり、 $E(v, x)$ は次式で与えられる。

$$E(v, x) = \sum_{i=1}^T v_i \exp\left(-\sum_{j=1}^m \alpha_{ij} x_{ij}\right) \quad (3)$$

問題は、支払関数 $E(v, x)$ に関して、プレイヤー B が先手で探索資源配分戦略 x を決めるミニマイザー、プレイヤー A が後手で価値配分 v を決定するマキシマイザーであるゲームであるから、支払関数のミニマックス値であるゲームの値 G は次式により評価される。

$$G = \min_{\{x_{ij}\} \in A} \max_{\{v_i\} \in D} \sum_{i=1}^T v_i \exp\left(-\sum_{j=1}^m \alpha_{ij} x_{ij}\right)$$

実行可能領域 D の制約条件を考察すれば、 v に関する最大化は次のように変形できる。

$$\max_{\{v_i\} \in D} \sum_{i=1}^T v_i \exp\left(-\sum_{j=1}^m \alpha_{ij} x_{ij}\right) = V \max_i \exp\left(-\sum_{j=1}^m \alpha_{ij} x_{ij}\right)$$

等式は、 $\exp(-\sum_{j=1}^m \alpha_{ij} x_{ij}) < \max_i \exp(-\sum_{j=1}^m \alpha_{ij} x_{ij})$ なるすべての $i \in I$ に対して $v_i = 0$ とし、右辺の最大値を与える施設 i に対してはそれらの総量が V となるように適当に価値配分をすることにより実現される。上の最終式を x に関してさらに最小化してゲームの値を求める問題は、次式で与えられる。

$$(P_0) \quad G = \min \left\{ V\lambda \mid \exp\left(-\sum_{j=1}^m \alpha_{ij} x_{ij}\right) \leq \lambda, i = 1, \dots, T, \{x_{ij}\} \in A \right\} \quad (4)$$

制約式 $\exp(-\sum_{j=1}^m \alpha_{ij} x_{ij}) \leq \lambda$ は、 $-\log \lambda \leq \sum_{j=1}^m \alpha_{ij} x_{ij}$ と変形できるから、 $\mu \equiv -\log \lambda$ により λ を置き換えると次式を得る。

$$G = \min \left\{ \frac{V}{\exp \mu} \mid \mu \leq \sum_{j=1}^m \alpha_{ij} x_{ij}, i = 1, \dots, T, \{x_{ij}\} \in A \right\} = \frac{V}{\exp \xi_C} \quad (5)$$

ただし、非負の整数全体を $\mathbf{Z}^+$ とすれば、 ξ_C は次式により与えられる。

$$\begin{aligned} (P_1) \quad \xi_C &= \max \left\{ \mu \mid \mu \leq \sum_{j=1}^m \alpha_{ij} x_{ij}, i = 1, \dots, T, \{x_{ij}\} \in A \right\} \\ &= \max \left\{ \min_{i \in I} \sum_{j=1}^m \alpha_{ij} x_{ij} \mid \sum_{i \in I} \sum_{j \in J} c_j x_{ij} \leq C, x_{ij} \in \mathbf{Z}^+, i \in I, j \in J \right\} \end{aligned} \quad (6)$$

I が1つの要素から成るときは、式(6)はいわゆる整数ナップサック問題と呼ばれる問題であるから、ナップサック問題の拡張という観点からすれば、(6)式はマックスミニ型整数ナップサック問題と呼ぶべき問題となる。すなわち、 T 個のナップサックに m 種類の品物を総コスト C の制約の下でいくつでもよいから詰め込む。 j タイプの品物1個のコストは c_j であり、これを i 番目のナップサックに入れた場合の価値は α_{ij} である。このとき、1つのナップサックに詰め込む価値の最小をできるだけ大きくして、ナップサック毎に入れた価値ができるだけ公平になるような詰め方を考える問題である。以上の議論から次の定理が成り立つ。

定理 1 原問題(P_0)はNP完全である。プレイヤーBの最適戦略はマックスミニ型整数ナップサック問題(P_1)の最適解 x^* により与えられ、プレイヤーAの最適戦略は $\xi_C < \sum_{j=1}^m \alpha_{ij} x_{ij}^*$ なる施設 i には価値を隠さず、総価値 V を $\xi_C = \sum_{j=1}^m \alpha_{ij} x_{ij}^*$ なる施設 i の間に任意に分割して隠すことである。

定理1から、以後はマックスミニ型整数ナップサック問題(P_1)の解法を議論してゆくが、その際自明な解を持つ場合はできるだけその対象から外したいため、まず以下の補助定理を述べよう。ただし、次のような記号を定義しておく。搜索資源単価の最小値を c_0 とし、この最小単価をもつ資源のタイプの集合を J_0 、各 $i \in I$ に対し J_0 の中で最大の係数 α_{ij} をもつ資源のタイプを $j_0(i)$ で表す。

$$c_0 = \min_{j=1, \dots, m} c_j, \quad J_0 = \{j \in J \mid c_j = c_0\}, \quad j_0(i) = \arg \max_{j \in J_0} \alpha_{ij}$$

補助定理 1 (i) $C < Tc_0$ ならば、最適値は $\xi_C = 0$ であり、最適解の1つは $x_{ij}^* = 0$ となる。

(ii) $Tc_0 \leq C < 2Tc_0$ ならば、最適値は $\xi_C = \min_{i \in I} \alpha_{ij_0(i)}$ である。また、最適解の1つは各 $i \in I$ に対し $x_{ij_0(i)}^* = 1$, $x_{ij}^* = 0$, $j \neq j_0(i)$ で与えられる。

(証明) (i)のケースは、総コストが少ないため少なくとも1つの施設への資源配分ができない場合であり、最適値は0となる。(ii)のケースは、すべての施設に対し最低価格 c_0 の資源を1単位のみ購入可能な場合であり、各施設 i に対し J_0 内の購入可能な資源により得られる最大係数 $\alpha_{ij_0(i)}$ の中で、最小値をもつ施設により最適値が決定される。 Q.E.D.

以下では、補助定理1の自明な場合以外、すなわち総費用が $C \geq 2Tc_0$ であることを暗黙に仮定して議論を進めてゆく。

3 マックスミニ型整数ナップサック問題の解法

3.1 動的計画法による解法

前節で議論したようにマックスミニ型整数ナップサック問題(P_1)は古典的な整数ナップサック問題を部分問題として含んでいるから、これを利用できる。施設 n に対する総コスト D の制約をもつ整数ナップサック問題とその最適値を

$$(IKP_n(D)) \quad \mu_n(D) = \max_{\{x_{nj}, j \in J\}} \left\{ \sum_{j=1}^m \alpha_{nj} x_{nj} \mid \sum_{j=1}^m c_j x_{nj} \leq D, x_{nj} \in \mathbb{Z}^+ \right\} \quad (7)$$

と表記しよう。さらに、目標 $i = 1, \dots, n$ に対する総コスト D 制約のマックスミニ型ナップサック問題の実行可能領域 $S_n(D)$ と最適値 $F_n(D)$ を次で定義する。

$$S_n(D) \equiv \left\{ \{x_{ij}, i = 1, \dots, n, j \in J\} \mid \sum_{i=1}^n \sum_{j \in J} c_j x_{ij} \leq D, x_{ij} \in \mathbb{Z}^+, i = 1, \dots, n, j \in J \right\}$$

$$F_n(D) \equiv \max_{\{x_{ij}\} \in S_n(D)} \min_{i=1, \dots, n} \sum_{j=1}^m \alpha_{ij} x_{ij}$$

このとき、整数の集合 $W \equiv \{\sum_{j=1}^m c_j x_j \mid \sum_{j=1}^m c_j x_j \leq C, x_j \in \mathbb{Z}^+\}$ を用いれば、次の漸化式が成り立つ。

$$\begin{aligned}
 F_n(D) &= \max_{\{x_{nj}, j \in J\}} \min \left\{ \sum_{j=1}^m \alpha_{nj} x_{nj}, F_{n-1}(D - \sum_{j=1}^m c_j x_{nj}) \right\} \\
 &= \max_{d \in W} \min \left\{ \sum_{j=1}^m \alpha_{nj} x_{nj}, F_{n-1}(D - d) \right\} \\
 &= \max_{d \in W} \min \left[\max_{\{x_{nj}, j \in J\}} \left\{ \sum_{j=1}^m \alpha_{nj} x_{nj} \mid \sum_{j=1}^m c_j x_{nj} = d, x_{nj} \in \mathbb{Z}^+ \right\}, F_{n-1}(D - d) \right] \\
 &= \max_{d \in W} \min \left[\max_{\{x_{nj}, j \in J\}} \left\{ \sum_{j=1}^m \alpha_{nj} x_{nj} \mid \sum_{j=1}^m c_j x_{nj} \leq d, x_{nj} \in \mathbb{Z}^+ \right\}, F_{n-1}(D - d) \right] \\
 &= \max_{d=1, \dots, D} \min \{ \mu_n(d), F_{n-1}(D - d) \} \\
 &= \max_{c_0 \leq d \leq D - (n-1)c_0} \min \{ \mu_n(d), F_{n-1}(D - d) \} \quad (8)
 \end{aligned}$$

最後の式は、補助定理1の議論から、施設 n への最低割当コスト c_0 と施設 $1 \sim n-1$ 全体への最低割当コスト $(n-1)c_0$ を確保する範囲内で d を動かすべきこと加味して導いた。

この漸化式を $n = 1, \dots, T$ と変化させながら、各 n に対しては $D = nc_0, nc_0 + 1, \dots, C$ に対して繰り返し計算し、最終的に $F_T(C)$ を求めれば終了する。繰り返し計算の初期値としては、 $n = 1$ に対しては次式を用いればよい。

$$F_1(D) = 0, D = 1, \dots, c_0 - 1, \quad F_1(D) = \mu_1(D), D = c_0, \dots, C$$

ただし、実際の数値計算では、各 $n = 1, \dots, T$ に対して $IKP_n(C)$ を一度解けばその過程で $\mu_n(D)$, $D = 1, \dots, C$ がいわゆる整数ナップサック関数として得られるから、その値をセーブしておき、漸化式(8)の $\mu_n(d)$ にはその値をロードして用いればよい。整数ナップサック問題 $IKP_n(C)$ では、総コスト C と資源の単価構成 c_j はどの施設 n に関しても同じであることを考えると、従来の解法による計算時間は n に大きくは依存しないことが予想されるためその計算時間を $knap(C)$ とすれば、整数ナップサック問題を解くために費やされる計算時間は $T \cdot knap(C)$ としてよい。

さて、(8)式では、各 d に対し内側の比較演算 $\min$ を1回と外側 $\max$ に関わる比較演算1回がなされる。 d は $D - (n-1)c_0 - c_0 + 1 = D - nc_0 + 1$ 回変化する。更にこのインデックス D は $nc_0 \leq D \leq C$ の範囲で動き、最後に $n = 2, \dots, T$ と変化する。したがって、比較演算の回数は全体で

$$\sum_{n=2}^T \sum_{D=nc_0}^C 2(D - nc_0 + 1) = \frac{c_0^2}{6}(2T^3 + 3T^2 + T - 6) - \frac{c_0}{2}(2C + 3)(T^2 + T - 2) + (C + 1)(C + 2)(T - 1)$$

となる。上の式から比較演算の計算量は $O(TC^2 + T^2Cc_0 + T^3c_0^2)$ のオーダーとなるが、一般的に総費用 C は c_0 に比べ十分大きい。以上をまとめると、提案した動的計画法によるマックスミニ型ナップサック問題 (P_1) の計算時間はオーダー $O(T \cdot knap(C) + TC^2 + T^2Cc_0 + T^3c_0^2)$ となる。

3.2 事前判定法

ここでは、まず整数ナップサック問題に関してこれまで培われてきた知見[1, 2]を解説し、それを使って問題 (P_1) の解法の詳細を述べる。次のような古典的な整数ナップサック問題を議論しよう。

$$(IKP(d)) \quad \mu(d) = \max_x \sum_{j=1}^m a_j x_j, \quad s.t. \quad \sum_{j=1}^m c_j x_j \leq d, \quad x_j \in \mathbb{Z}^+, j \in J \quad (9)$$

ただし、品物には $a_1/c_1 \geq a_2/c_2 \geq \dots \geq a_m/c_m$ となるようすでに番号付けがされている。総コスト d を $0 \leq d \leq C$ で変化させたときの最適値 $\mu(d)$ の変化は整数ナップサック関数と呼ばれ、容易に想像できるように単調で非減少、右連続な階段関数となる。この階段関数が不連続に変化する d の値を $(IKP(C))$ を解く過程で求めるアルゴリズムはすでにあり、小さい順にならべたこの d のリスト $f(1), f(2), \dots$ を階段リストと呼ぶこととしよう。また、 $(IKP(d))$ の最適解を x_j^* とすると、 x_j^* が正となる最小の添字 j を $g(d)$ とする。ただし、最適解が複数あればそのような添字 j を最小にする最適解 x_j^* を想定する。すなわち、

$$g(d) = \begin{cases} \min\{j \mid x_j^* > 0\}, & x^* \neq 0 \\ m, & x^* = 0 \end{cases} \quad (10)$$

このとき、 $x_j^* > 0$ ならば解 $(x_1^*, \dots, x_{j-1}^*, x_j^* - 1, x_{j+1}^*, \dots, x_m^*)$ は $IKP(d - c_j)$ の最適解となり、 $x_j^* > 0$ なるすべての j に対し $g(d) \leq g(d - c_j)$ が成り立つ。したがって、 $j \leq g(d - c_j)$ で $d - c_j \geq 0$ を満たす j の中に、総コスト $d - c_j$ の問題 $(IKP(d - c_j))$ の最適解 $\tilde{x}^*$ の j 成分 $\tilde{x}_j^*$ にだけ 1 を足した解が問題 $(IKP(d))$ の最適解となるものが存在する。すなわち、

$$\mu(d) = \max\{a_j + \mu(d - c_j) \mid j \leq g(d - c_j), d - c_j \geq 0\} \quad (11)$$

上式の最大値を与える添字 j の最小値が $g(d)$ である。

また、 $D^* \leq d$ なるすべての d に対しては $\mu(d) = a_1 + \mu(d - c_1)$ 、すなわち、あるコスト D^* 以上のコスト制約 d では、常に $IKP(d - c_1)$ の最適解の x_1^* に 1 を足すことにより $IKP(d)$ の最適解が得られるという周期性を示す。この周期性は、あるコスト D^* に対し

$$g(d) = 1, \quad d = D^*, D^* + 1, \dots, D^* + \max_j c_j - 1 \quad (12)$$

となることを確認すれば十分である。以上が、整数ナップサック問題に関してすでに知られている知見である。以上を踏まえ、原問題 (P_1) の解法を提案する。

総コスト d をもつ問題 (P_1) の最適値 ξ_d が得られている段階で、さらに最適値を増加させるためには $\mu_i(d_i) = \xi_d$ となっている施設 i への割当コスト d_i をすべて適切な量まで増加させなければならない。そのために、各施設 $i \in I$ の整数ナップサック問題 $IKP_i(C)$ を解く際に得られる階段リスト $f_i(\cdot)$ を利用して、必要な増加コストを事前に判定しつつ計算を進めてゆくもので、この解法を事前判定法と呼ぶ。

Step1. 施設 $i \in I$ のそれぞれに関する整数ナップサック問題 $IKP_i(C)$ を解き、整数ナップサック関数値 $\mu_i(D), D = 1, \dots, C$ と階段リスト $f_i(\cdot)$ を得る。 $l_i = 1, i \in I, d \equiv \sum_{i \in I} f_i(l_i)$ とおく。このとき、 d の値は T_{c_0} となっているはずである。

Step2. $\mu = \min_{i \in I} \mu_i(f_i(l_i)), K = \{n \mid \mu_n(f_n(l_n)) = \mu\}$ とおく。

Step3. $s = \sum_{n \in K} (f_n(l_n + 1) - f_n(l_n))$ を計算する。

もし $d + s > C$ ならば終了、最適値は μ である。 $d_i^* = f_i(l_i)$ が施設 i への最適な割当コストであり、整数ナップサック問題 $IKP_i(d_i^*)$ を解くことにより施設 i に対するタイプ $j \in J$ の資源の最適な割当量 $x_{ij}^*, j \in J$ を求める。

もし $d + s \leq C$ ならば、 $d = d + s, l_i = l_i + 1, i \in K$ として Step0b に戻る。

Step0c での処理から分かるとおり、施設 i への割当コストは階段リスト $f_i(1), f_i(2), \dots$ の示す値へ一度に増加され、総コストが s だけ増加するごとに最適値も必ず増加してゆく。

この事前判定法が正しく最適値を導くことは次のように示すことができる。まず、Step0c で増加させるある施設 $n \in K$ への割当コストの一部を代わりに他の施設への割当コストの増加分に使用した場合、 $\mu_n(d_n)$ は増加せず目的関数値は現在の暫定値 μ のままとなる。また、 $n \notin K$ である施設 n の割当コストを減らし、その分を他の施設の割当コストの増加分に使用した場合、施設 n の現在値 $\mu_n(f_n(l_n))$ は前の値 $\mu_n(f_n(l_n - 1))$ に

減少する。割当コスト $f_n(l_n - 1)$ がかって Step0cにおいて増加されたということは、現在の暫定値 μ に対し $\mu_n(f_n(l_n - 1)) < \mu$ であることを意味するから、この場合の目的関数値は現在の暫定値 μ より減少することになる。以上により、各施設への割当コストを上記アルゴリズムで行うことが目的関数値を最大にする方法である。アルゴリズムが終了するまでに Step0bを実行するごとに計算される d の値により、原問題 (P_1) の目的関数値が不連続に変化する点、原問題のいわばナップサック関数を構成する不連続点が羅列される。

3.3 逐次増分法

事前判定法では、各施設 $i \in I$ に対する割当コストを C として $IKP_i(C)$ を解いた際に得られる整数ナップサック関数を利用するものであるが、実際に (P_1) が解かれた段階での各施設への最適な割当コストは C よりかなり小さくなることが予想される。ここで述べる逐次増分法では、各施設への割当コストを階段リストを用いずに逐次に増加させながら解を求める方法であり、その際 (10)~(12) 式で表される知識を利用する。したがって、各施設に対する整数ナップサック問題の解法を事前には用いず、アルゴリズムの中に直接組み込む。

施設 i に関して、パラメータ比 α_{ij}/c_j , $j \in J$ を大きい順に並べた際の添字 j の並びは $l_i(1), \dots, l_i(m)$ であるとする。すなわち、 $\alpha_{il_i(1)}/c_{l_i(1)} \geq \alpha_{il_i(2)}/c_{l_i(2)} \geq \dots \geq \alpha_{il_i(m)}/c_{l_i(m)}$ とする。また、 $c_{\max} = \max_{j \in J} c_j$ とおく。

Step1. すべての施設 $i \in I$ に対し次のような初期設定を行う。

$\mu_i(0) = \mu_i(1) = \dots = \mu_i(c_0 - 1) = 0$, $\mu_i(c_0) = \alpha_{ij_0(i)}$ とし, $g_i(0) = g_i(1) = \dots = g_i(c_0 - 1) = m$, $g_i(c_0) = \min\{k \mid c_{l_i(k)} = c_0\}$ とおく。さらに, $r(i) = 0$, $d_i = c_0$, $D_i^* = c_0$ とおく。

Step2. $d = \sum_{i \in I} d_i$ を計算する。もし $d > C$ ならば, Step0dにいき終了する。そうでなければ, d_i の値を $\tilde{d}_i$ に保存する。すなわち, $\tilde{d}_i = d_i$, $i \in I$ とする。 $\mu = \min_{i \in I} \mu_i(d_i)$, $K = \{n \mid \mu_n(d_n) = \mu\}$ とする。ここで $d + |K| > C$ ならば終了, Step0dにいく。

Step3. 各 $n \in K$ に対し次を実行した後, Step0bに戻る。ただし, 途中で $\sum_{i \in I} d_i > C$ となれば Step0dにいき終了する。

$d_n = d_n + 1$ とし以下の手順で $\mu_n(d_n)$ を計算することを, $\mu_n(d_n - 1) < \mu_n(d_n)$ となるまで繰り返す。

(i) $r(n) = 1$ ならば, $\mu_n(d_n) = \mu_n(d_n - c_{l_n(1)}) + \alpha_{nl_n(1)}$ とする。

(ii) $r(n) = 0$ ならば,

$$j^* = \arg \max_j \{\mu_n(d_n - c_{l_n(j)}) + \alpha_{nl_n(j)} \mid d_n - c_{l_n(j)} \geq 0, j \leq g_n(d_n - c_{l_n(j)})\}, \quad (13)$$

$$\mu_n(d_n) = \mu_n(d_n - c_{l_n(j^*)}) + \alpha_{nl_n(j^*)}, \quad (14)$$

$$g_n(d_n) = j^* \quad (15)$$

とおく。 $j^* = 1$ かつ $d_n = D_n^* + c_{\max} - 1$ ならば, $r(n) = 1$ とする。 $j^* \neq 1$ ならば, D_n^* を $D_n^* = d_n$ により更新する。

Step4. 最適値は μ である。各施設 $i \in I$ について $IKP_i(\tilde{d}_i)$ を解いて求めた解 $\{x_{ij}^*, j \in J\}$ を最適解として終了する。

Step0aで初期設定し, Step0cで更新を行う $g_n(d_n)$, D_n^* は, 施設 n に関する (12) 式の $g(d)$, D^* に相当し, この式が成立すれば $r(n) = 1$ において最適値の周期性が始まったことを示す。また, (13), (14) 式は施設 n に関する (11) 式の演算に相当する。このアルゴリズムにおいても, 総コスト d を逐次に増加させながら, 最も小さい目的関数値をもつ施設の集合 K への割当コストを目的関数値が増加するまで増やすという基本プロセスを繰り返すことは事前判定法と同様であるため, その妥当性は明らかである。

3.4 近似解法

これまで提案した解法は厳密解法であり、当然のことながら問題のNP完全性は、問題のサイズによってはその計算時間を耐え難いものにする可能性がある。ここでは、厳密解であることをあきらめ、多項式計算時間で近似解を出す近似解法について議論する。

まず、問題 (P_1) の変数を実数緩和した次の問題を考えよう。

$$(RLP_1) \quad \hat{\mu} = \max \min_{i \in I} \sum_{j=1}^m \alpha_{ij} x_{ij} \quad (16)$$

$$s.t. \quad \sum_{i=1}^T \sum_{j=1}^m c_j x_{ij} \leq C, \quad (17)$$

$$x_{ij} \geq 0, \quad x_{ij} \in \mathbf{R}, \quad i \in I, \quad j \in J. \quad (18)$$

この問題の最適解は次で与えられる。

定理 2 緩和問題 (RLP_1) の最適値 $\hat{\mu}$ 、最適解 $\hat{x}$ は次式で与えられる。ただし、 $s(i) \equiv \arg \max_{j \in J} \{\alpha_{ij}/c_j\}$ とする。

$$\hat{\mu} = \frac{C}{\sum_{k=1}^T c_{s(k)}/\alpha_{ks(k)}} \quad (19)$$

$$\hat{x}_{is(i)} = \frac{C/\alpha_{is(i)}}{\sum_{k=1}^T c_{s(k)}/\alpha_{ks(k)}}, \quad i \in I, \quad \hat{x}_{ij} = 0, \quad i \in I, \quad j \neq s(i), \quad j \in J \quad (20)$$

(証明) $\rho_{ij} = c_j x_{ij}$ とおくと、問題 (RLP_1) は、

$$\hat{\mu} = \max_{\{\rho_{ij}\}} \min_{i \in I} \left\{ \sum_{j=1}^m \frac{\alpha_{ij}}{c_j} \rho_{ij} \mid \sum_{i=1}^T \sum_{j=1}^m \rho_{ij} \leq C, \quad \rho_{ij} \geq 0, \quad i \in I, \quad j \in J \right\}$$

となるが、さらに、 $\gamma_{ij} \equiv \alpha_{ij}/c_j$ 、 $\hat{d}_i \equiv \sum_{j=1}^m \rho_{ij}$ とおけば、次のように変形できる。

$$\begin{aligned} &= \max_{\{\hat{d}_i, \rho_{ij}\}} \min_{i \in I} \left\{ \sum_{j=1}^m \gamma_{ij} \rho_{ij} \mid \sum_{i=1}^T \hat{d}_i \leq C, \quad \hat{d}_i = \sum_{j=1}^m \rho_{ij}, \quad \rho_{ij} \geq 0, \quad i \in I, \quad j \in J \right\} \\ &= \max_{\{\hat{d}_i, \rho_{is(i)}\}} \min_{i \in I} \left\{ \gamma_{is(i)} \rho_{is(i)} \mid \sum_{i=1}^T \hat{d}_i \leq C, \quad \hat{d}_i = \rho_{is(i)} \geq 0, \quad i \in I \right\} \\ &= \max_{\{\hat{d}_i\}} \min_{i \in I} \left\{ \gamma_{is(i)} \hat{d}_i \mid \sum_{i=1}^T \hat{d}_i \leq C, \quad \hat{d}_i \geq 0, \quad i \in I \right\} \\ &= \max_{\{\hat{d}_i, \mu\}} \left\{ \mu \mid \mu \leq \gamma_{is(i)} \hat{d}_i, \quad i \in I, \quad \sum_{i=1}^T \hat{d}_i = C, \quad \hat{d}_i \geq 0, \quad i \in I \right\} \end{aligned} \quad (21)$$

最終式の最適値は次式により与えられることは明らかである。

$$\hat{\mu} = \gamma_{is(i)} \hat{d}_i, \quad i \in I, \quad \sum_{i=1}^T \hat{d}_i = C.$$

この連立方程式を解けば、

$$\hat{\mu} = \frac{C}{\sum_{k=1}^T 1/\gamma_{ks(k)}} = \frac{C}{\sum_{k=1}^T c_{s(k)}/\alpha_{ks(k)}}, \quad \hat{d}_i = C \frac{c_{s(i)}/\alpha_{is(i)}}{\sum_{k=1}^T c_{s(k)}/\alpha_{ks(k)}}$$

となるが、(21)式の変形では $\hat{d}_i = \rho_{is(i)}$ 、 $i \in I$ 、 $\rho_{ij} = 0$ 、 $j \neq s(i)$ 、 $j \in J$ の設定がなされていたことを考えると、最適解 $\{\hat{x}_{ij}\}$ は定理により与えられる。 **Q.E.D.**

近似解法として、定理 2 を用いた次のような極めて簡便な貪欲法を提案しよう。その手順の第 1 は、まず緩和問題 (RLP_1) の実数最適解 $\{\hat{x}_{ij}\}$ を、切り捨てにより整数解 $\{x_{ij}\} = \{\lfloor \hat{x}_{ij} \rfloor\}$ とする。その後、未使用コスト $C - \sum_i \sum_j c_j x_{ij}$ を用いて、事前判定法、逐次増分法の Step0b で求めた施設集合 K の目的関数を増加させるために最小単価 c_0 の資源を逐次的に増やしてゆこうとするものである。そのアルゴリズムを述べれば次のようになる。

Step1. 定理 2 を用いて緩和問題 (RLP_1) の実数最適解 $\{\hat{x}_{ij}\}$ を求める。その解を切り捨て、 $\{x_{ij}\} = \{\lfloor \hat{x}_{ij} \rfloor\}$ により整数化する。 $C_R = C - \sum_{i \in I} \sum_{j \in J} c_j x_{ij}$, $\mu_i = \sum_{j \in J} \alpha_{ij} x_{ij}$, $i \in I$ を計算する。

Step2. $\mu = \min_{i \in I} \mu_i$, $K = \{n \mid \mu_n = \mu\}$ を求める。

Step3. もし $|K| c_0 > C_R$ ならば、 μ を近似値、 $\{x_{ij}\}$ を近似解として終了する。

そうでなければ、各 $n \in K$ に対し次式による更新を行い、 $C_R = C_R - |K| c_0$ として、Step2 に戻る。

$$x_{njo(n)} = x_{njo(n)} + 1, \quad \mu_n = \mu_n + \alpha_{njo(n)}$$

4 おわりに

この論文では、先行者、追従者のいる一般的な離散資源配分ゲームのシュタッケルベルク均衡を議論した。このタイプの問題は、目標割当問題や搜索問題、査察問題をはじめとして多くの応用例をもつ。同様に、一方的な最適化問題として研究されてきた他のタイプの資源配分問題を複数の意思決定者間のゲームとして考え直すニーズは至る所に存在する。また、資源配分問題の中でも特に離散最適化問題に属する問題は、ナップサック問題をはじめ問題構造の単純さの割には実用的な解法の提案が困難であり、ゲームによる定式化とともにその解法を提案することは、これから推進されるべき研究方向の 1 つであると思われる。今回の研究はそのような研究の 1 つである。

参考文献

- [1] P.C. Gilmore and R.E. Gomory: The theory and computation of Knapsack function. *Opns. Res.*, **14** (1966) 1045–1074.
- [2] R.S. Garfinkel and G.L. Nemhauser: *Integer programming*. (John Wiley & Sons, 1972).
- [3] R. Hohzaki and K. Iida: An integer resource allocation problem with cost constraint. *Journal of the Operations Research Society of Japan*, **41** (1998) 470–482.
- [4] F. Lemus and K.H. David: An optimal allocation of different weapons to a target complex. *Opns. Res.*, **11** (1963) 787–794.
- [5] A.S. Manne: A target-assignment problem. *Opns. Res.*, **7** (1959) 258–260.
- [6] R.H. Nickel and M. Mangel: Weapon acquisition with target uncertainty. *Naval Research Logistics*, **32** (1985) 567–588.
- [7] J.B. Kadane: Discrete search and the Neyman-Pearson lemma. *J. of Mathematical Analysis and Applications*, **22** (1968) 156–171.
- [8] T. Ibaraki and N. Katoh: *Resource allocation problem*. (The MIT Press, London, 1988).