A-018

多品種フローゲームのコアの要素を求める多項式時間アルゴリズム
Polynomial-Time Algorithm for a Element in the Core of a Multicommodity Flow Game

唐澤 一寛 † Kazuhiro Karasawa　　山田 敏規 † Toshinori Yamada

1. はじめに

インターネットは多くの自律システムから構成されている．現在自律システム間のルーティングプロトコルとしてはBGP(Border Gateway Protocol)が多く用いられている．自律システムはBGPによって互いの経路情報を交換・告知することで，動的なルーティングを可能としている．自律システムのトラフィックにはその自律システム内のノードを始点または終点とするローカル・トラフィックと，自律システムを通過する中継トラフィックがある．各自律システムではローカル・トラフィックの要求を優先し中継トラフィックをできるだけ回避しようとするが，これがネットワークの不安定性に結び付くかもしれない．

Papadimitriou[1]はBGPにおけるルーティングの安定性の概念を以下のような提携ゲームとして定式化した．グラフ G を $1,2,\ldots,n$ とラベル付けされた n 点からなる無向グラフとする．G の各点は自律システムを表す，ゲームのプレイヤーである．G の点集合を $V(G)$ で，辺集合を $E(G)$ で表す．G の任意の点 i について，c_i を点 i の容量とする．c_i は自律システムのサブネットワークの容量を抽象化したものである．また，G の任意の点 i,j について，$d_{i,j}$ を点 i と j の間の通信要求とする．f_p をパス p に沿ったフローの量とする．このとき，点 i,j 間のフロー $f_{i,j}$ は

$$f_{i,j} = \sum_{p\in P_{i,j}} f_p$$

と表せる．ここで，$P_{i,j}$ は点 i と j を結ぶパスの集合を表す．点 i を通るパスの集合を $P(i)$ とする．任意の点 i について $\sum_{p\in P(i)} f_p \le c_i$ が成り立ち，任意の点 i,j について $f_{i,j} \le d_{i,j}, f_p \ge 0$ が成り立つならば $f = (f_p)$ は G,d,c における実現可能な多品種フローと呼ばれる．ゲームの結果は実現可能な多品種フローである．$\mathcal{F}(G,d,c)$ を実現可能な多品種フローの集合，$\mathcal{F}_S(G,d,c)$ を S によって誘導される G の部分グラフにおける実現可能な多品種フローの集合とする．フロー f による点 i の利得 $u_i(f)$ を

$$u_i(f) = \sum_{j\ne i} f_{i,j}$$

とし，f の利得ベクトルを $u(f) = (u_1(f), u_2(f), \ldots, u_n(f))$ とする．多品種フローの値 $\mathrm{value}(f)$ を

$$\mathrm{value}(f) = \sum_{i,j} f_{i,j} = \frac{1}{2}\sum_{i\in[1,n]} u_i(f)$$

とする．全ての $f' \in \mathcal{F}(G,d,c)$ に対して $\mathrm{value}(f) \ge \mathrm{value}(f')$ であるような $f \in \mathcal{F}(G,d,c)$ を最大多品種フローと呼ぶ．G,d,c における多品種フローゲームのコア $core(G,d,c)$ は，任意の $S \subset V(G)$ と $f' \in \mathcal{F}_S(G,d,c)$ について，$u_k(f') \le u_k(f)$ となる $k \in S$ が存在する $f \in \mathcal{F}(G,d,c)$ の利得ベクトル $u(f)$ の集合である．

MarkakisとSaberi[2]はコアの非空性を証明した．

定理 1 [2] 容量 c，通信要求 d である任意のグラフ G について $core(G,d,c) \ne \emptyset$ である．

しかしながら，定理1の証明は構成的ではなく，$u(f) \in core(G,d,c)$ である多品種フロー f を見つけることは重要な未解決問題である．

この小文で，G がパス，完全グラフまたはスターである場合に，$u(f) \in core(G,d,c)$ である多品種フロー f を見つけるための多項式時間アルゴリズムを与える．さらに，このアルゴリズムによって得られる多品種フローは最大多品種フローであることも示す．

なお，証明は紙面の都合上省略する．

2. 多品種フローゲームのコアの要素を求める多項式時間アルゴリズム

2.1 パス

任意の正の整数 i と j について，$i \le j$ ならば $[i,j] = \{i, i+1, \ldots j\}$ とする．このとき n 点パス L_n は以下のように定義されるグラフである．

$$V(L_n) = [1,n]; \quad E(L_n) = \{(i,j) : |i-j| = 1\}$$

パス上の多品種フローゲームにおける実現可能な多品種フロー f は図1のアルゴリズムによって多項式時間で求められる．

定理 2 図1のアルゴリズムは f を $O(n^3)$ 時間で求める．

例として，4点パス L_4 において通信要求が $d_{1,2} = 1, d_{1,3} = 4, d_{1,4} = 3, d_{2,3} = 0, d_{2,4} = 2, d_{3,4} = 2$ で容量

† 埼玉大学 Saitama University

```
for i ← 1 to n do C_i ← c_i;
for j ← 2 to n do
  for i ← j − 1 to 1 do begin
    f_{i,j} ← min{C_i, C_{i+1}, ..., C_j, d_{i,j}};
    for k ← i to j do C_k ← C_k − f_{i,j};
  end
```

図 1　パス P_n 上の多品種フローゲームのコアの要素を求めるアルゴリズム

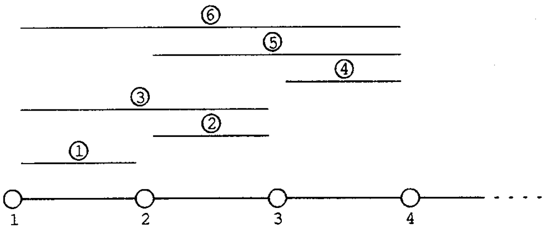

図 2　図 1 のアルゴリズムによるフローの割り当て順

が $c_1 = 3, c_2 = 6, c_3 = 5, c_4 = 4$ であるような多品種フローゲームについて考える．最初に，図 1 のアルゴリズムの 1 行目で，点 i の容量 c_i の値が C_i に代入される．C_i はその時点での点 i の残り容量を表している．2 行目以降のループにおいてフロー $f_{i,j}$ は図 2 のような順序で決定されていく．L_4 の場合には $f_{1,2}, f_{2,3}, f_{1,3}, f_{3,4}, f_{2,4}, f_{1,4}$ の順である．$f_{i,j}$ には $C_i, C_{i+1}, \ldots, C_j$ と $d_{i,j}$ の中の最小の値が選ばれる．L_4 で d, c が上記のようになっている場合には，まず，$f_{1,2}$ が $f_{1,2} = \min\{3, 6, 1\} = 1$ と計算される．その後で $C_1 = C_1 - f_{1,2} = 2$ と $C_2 = C_2 - f_{1,2} = 5$ の計算が行われて，$f_{1,2}$ の分だけ残り容量が減らされる．次に，$f_{2,3}$ が $f_{2,3} = \min\{5, 5, 0\} = 0$ となる．この場合には $f_{2,3} = 0$ であるので C の値は変わらない．さらに，$f_{1,3} = \min\{2, 5, 5, 4\} = 2$ となり，残り容量は $C_1 = 0, C_2 = 3, C_3 = 3$ に更新される．同様にして計算を続けると，$f_{1,2} = 1, f_{1,3} = 2, f_{1,4} = 0, f_{2,3} = 0, f_{2,4} = 1, f_{3,4} = 2$ となる．このとき利得は，$u(f) = \{3, 2, 4, 3\}$ となる．

図 1 による多品種フロー f は，$u(f) \in core(L_n, d, c)$ であるような最大多品種フローである．

定理 3 $u(f) \in core(L_n, d, c)$ である．さらに，f は最大多品種フローである．

2.2 完全グラフ

完全グラフ K_n は n 個の点 $V(K_n)$ と，辺 $E(K_n) = \{(i, j) : i, j \in V(K_n), i \neq j\}$ からなるグラフである．f は以下の線形計画問題の解とする．

$$\begin{aligned} \text{maximize :} \quad & \sum_{i,j} f_{i,j} \\ \text{subject to :} \quad & f_{i,j} \leq d_{i,j} \\ & \sum_{j \neq i} f_{i,j} \leq C_i \end{aligned}$$

定理 4 f は多項式時間で求められる．時間で求める．

定理 5 $u(f) \in core(K_n, d, c)$ である．さらに，f は最大多品種フローである．

2.3 スター

n 点スター S_n は n 個の点 $V(S_n) = \{0, 1, 2, \ldots, n-1\}$ と辺 $E(S_n) = \{(0, i) : i \in V(S_n) - 0\}$ からなるグラフである．図 3 のアルゴリズムによって，スター S_n 上の多品種フローゲームにおける実現可能な多品種フロー f が多項式時間で求められる．

定理 6 図 3 のアルゴリズムは f を $O(n^2)$ 時間で求める．

```
for i ← 0 to n − 1 do C_i ← c_i;
for j ← 0 to n − 2 do
  for i ← j + 1 to n − 1 do begin
    f_{i,j} ← min{C_0, C_i, C_j, d_{i,j}};
    C_i ← C_i − f_{i,j};   C_j ← C_j − f_{i,j};
    if j ≠ 0 then
      C_0 ← C_0 − f_{i,j};
  end
```

図 3　スター S_n 上の多品種フローゲームのコアの要素を求めるアルゴリズム

スター S_n 上の多品種フローゲームにおいて，全てのフローは中央の点を通る．したがって，中央の点の利得となるフローを優先して割り当てることで，$u(f) \in core(S_n, d, c)$ であるような最大多品種フロー f を得ることができる．

定理 7 $u(f) \in core(S_n, d, c)$ である．さらに，f は最大多品種フローである．

参考文献

[1] C. H. Papadimitriou. Algorithms, games, and the internet. *Annual ACM Symposium on the Theory of Computing*, pages 749–753, 2001

[2] E. Markakis and A. Saberi. On the Core of the Multicommodity Flow Game. *ACM conference on Electronic commerce*, pages 93–97, 2003