

パイ分割問題に対するカット数最小の無羨望メカニズム
An envy-free mechanism with minimum number of cuts for the pie-cutting problem

梅田 博之[†] 浅野 孝夫[‡]
Hiroyuki Umeda Takao Asano

1. はじめに

パイとは始点と終点を同一視した閉区間 $[0, b_p]$ であり, $P = [0, b_p]$ とかく. またプレイヤー集合を N とする. 各プレイヤー $i \in N$ は評価区間 V_i をもつ. このとき, P をどのように分割し, 各プレイヤー i に割り当てるかを考える問題がパイ分割問題である.

この問題に対し, Barbanel, J. B. ら [1] はプレイヤー数を3以下と制限すれば, カット数が $|N|$ であり, 無羨望性を満たす割当を返すメカニズムを提案した. 一方, 類似問題であるケーキ分割問題に対し, Alijani, R. ら [2] はプレイヤー数を制限せずに, 上記2つの性質に加え, 戦略的操作不可能性を満たす割当を返すメカニズムを提案した. そこで梅田, 浅野 [3] は, 後者の結果をパイ分割問題へ応用し, 評価区間に制約を加えた状態において, 3つの性質を満たすメカニズムを提案した.

本研究はこれの発展として, パイ分割問題に対し評価区間を制約せずに, カット数が高々 $2(|N| - 1)$ であり, 無羨望性・戦略的操作不可能性を満たすメカニズムを提案する.

2. 準備

まず, P -区間を定義する. パイ P 上の座標は, 12時の位置が0であり, 時計回りに大きくなり, 1週すると0に戻る. この座標上で始点 a , 終点 b の区間を $[a, b]$ とかく. また, その大きさを $|[a, b]|$ と表す.

次に評価区間およびそれに関する定義を行う. 全てのプレイヤー i は, 評価区間 $V_i = [\alpha_i, \beta_i]$ をもつ. 図1(1)の例では, $V_1 = [0, 1], V_2 = [1/8, 7/8], V_3 = [1/4, 3/4]$ である. $T \subseteq N$ に対し, $V_T = (V_i)_{i \in T}$ と表す. このとき, V_T のドメイン (V_T の要素全てを含む極小な P -区間) を

$\text{DOM}(V_T) = \text{argmin}_{[a,b] \subseteq P} \{ |[a,b]| \mid V_i \subseteq [a,b], \forall i \in T \}$ と定義する. V_T の密度を $\Phi(V_T) = \text{DOM}(V_T)/|T|$ とする. $\Phi(V_T) < \Phi(V_{T'}), \forall T' \subseteq T$ が成り立つとき V_T は既約であるという.

次にメカニズム M による割当について述べる. M は V_N を入力とし, 割当 $A^M(V_N) = (A_1^M(V_N), A_2^M(V_N), \dots, A_{|N|}^M(V_N))$ を決定する. 割当は P の分割である.

効用とは, 評価区間と割当の共通部分の大きさである.

具体的には $U_i^M(V_N) = |V_i \cap A_i^M(V_N)|$ で表す.

最後に, カット数と2つの性質を定義する.

定義1 (カット数) $\#A_i^M(V_N)$ は $A_i^M(V_N)$ に含まれる P -区間の数としたとき, M のカット数は, $\sum_{i \in N} \#A_i^M(V_N)$ である.

定義2 (無羨望性) メカニズム M が無羨望性を満たすとは, $\forall i \in N$ に対し

$$|V_i \cap A_i^M(V_N)| \leq U_i^M(V_N), \forall o \in N - \{i\}, \forall V_N$$

が成立することである.

定義3 (戦略的操作不可能性) メカニズム M が戦略的操作不可能性を満たすとは, 任意の入力 V_N に対して, どの $i \in N$ でも, V_i をどのような V_i' に変えても

$$|V_i \cap A_i^M(V_i', V_{N-\{i\}})| \leq U_i^M(V_i, V_{N-\{i\}})$$

が成立することである.

3. 提案メカニズム

3.1 一般化拡張プロセス

ここでは提案メカニズム P-EFGISM の構成要素である一般化拡張プロセスを示し, その後アンロックについて示す.

そのためまずチェーンを定義する. これは各プレイヤー $i \in N$ への割当候補に対応する, P -区間 $C_i = [a_i, b_i]$ の隙間ない列である. N 上の置換を σ_j とする. そのとき, $K_j = (C_{\sigma_j(1)}, C_{\sigma_j(2)}, \dots, C_{\sigma_j(n_j)})$ において, $b_{\sigma_j(i')} = a_{\sigma_j(i'+1)}$, $1 \leq \forall i' \leq n_j - 1$ が成立するとき, K_j をチェーンと呼ぶ. また, $b_{\sigma_j(n_j)} = \beta_{\sigma_j(n_j)}$ が成立するとき, K_j はロックされたという.

それでは, 以下に一般化拡張プロセスを示す.

アルゴリズム1 一般化拡張プロセス
<pre> U-Exp(T, V_T, P) $K = \phi$; For each $i \in T$ $K_i \leftarrow [\alpha_i, \alpha_i]$; $K \leftarrow K \cup \{K_i\}$; 図1(1) While K内いずれのチェーンもロックされていない do For each $K_j \in K$ do For $i' = 1$ to n_j do $C_{\sigma_j(i')} \leftarrow [a_{\sigma_j(i')} + (i' - 1)\delta, (b_{\sigma_j(i')} + i'\delta) \bmod P]$; (2) if K_jがロックされている 図1(4) if G^{K_j}が閉路をもつ // G^{K_j}については3.1後半 for $i' = 1$ to n_j do $P_{\sigma_j(i')} \leftarrow P_{\sigma_j}(\delta^{K_j(i')})$ 図1(5) Else return K_j; 図1(6) Elseif $b_{\sigma_j(n_j)} = a_{\sigma_{j'}(1)}, \exists K_{j'} \in K$ $K_j \leftarrow K_j \cup K_{j'}$; $K \leftarrow K - K_{j'}$; 図1(3) </pre>

次に, 図1を用いて一般化拡張プロセスの動作を概説する. ここでは, 大きさが1のパイが入力されたとする. まず初期化を行う(1). 次に拡張を行う(2). 拡張とは全てのプレイヤー i への割当候補の大きさを同時に δ だけ大きくすることを意味する. これを行っていくと, チェーン K_1 が K_2 に, K_2 が K_3 に衝突するため, これらを結合する(3). 結合した K_1 を拡張し続けると, ロックされる(4). これは, これ以上拡張を続けると, プレイヤー3が評価していない区間を割当候補にすることを意味する. そこで, チェーン内のプレイヤーの順番を1, 2, 3 から 1, 3, 2 へ変更し(アンロック) (5), 拡張を続ける. それが不可能な場合ロックされたチェーンを返し, アルゴリズムは終了する(6).

[†] Hiroyuki Umeda, Information and System Engineering Course, Graduate School of Science and Engineering, CHUO University

[‡] Takao Asano, Department of Information and System Engineering, Faculty of Science and Engineering, CHUO University

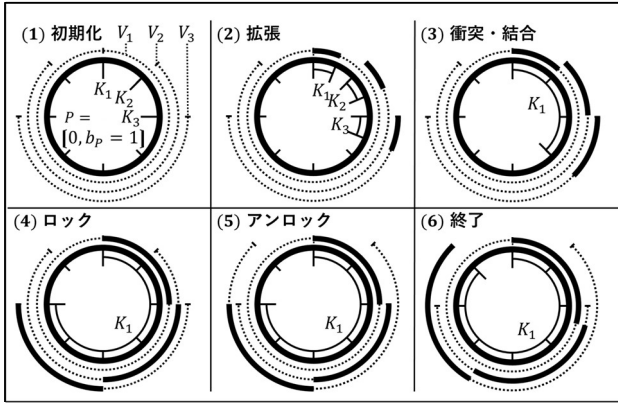


図1 一般化拡張プロセス

以降は、アンロックをいかに行うかを述べる。

まず、ロックされたチェーン K_j に対し、アンロックグラフ $G^{K_j} = (V, E_l \cup E_r)$ を構成する。点集合を $V = \{v_{\sigma_j(1)}, v_{\sigma_j(2)}, \dots, v_{\sigma_j(n_j)}\}$ と定義する。また辺集合を

$$E_l = \{(v_{\sigma_j(i)}, v_{\sigma_j(i')}) \mid (i' > i') \cap (\beta_{\sigma_j(i')} - \lfloor \alpha_{\sigma_j(i')} \rfloor \leq \beta_{\sigma_j(i)} - \lfloor \alpha_{\sigma_j(i)} \rfloor)\}$$

$$E_r = \{(v_{\sigma_j(i)}, v_{\sigma_j(i')}) \mid (i' < i') \cap (\alpha_{\sigma_j(i')} + \lfloor \alpha_{\sigma_j(i')} \rfloor > \alpha_{\sigma_j(i)} + \lfloor \alpha_{\sigma_j(i)} \rfloor)\}$$

とする。図1(4)の K_1 に対して、 G^{K_1} は図2ようになる

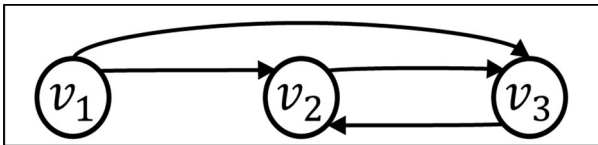


図2 図1(4)の K_1 に対するアンロックグラフ

アンロックは G^{K_j} を用いて行う。 G^{K_j} 上にて $v_{\sigma_j(n_j)}$ を含む閉路を考える。閉路に含まれる点 $v_{\sigma_j(i')}$ の中で、最小の i' を l とする。このとき、置換 δ^{K_j} を $\delta^{K_j}(i') = i', 1 \leq \forall i' < l, \delta^{K_j}(l) = n_j, \delta^{K_j}(i') = i' - 1, l < \forall i' \leq n_j$ と定義する。図2では、 $l = 2$ であり、 $\delta^{K_j}(1) = 1, \delta^{K_j}(2) = 3, \delta^{K_j}(3) = 2$ となり、図1(5)のようにアンロックできる。

3.2 P-EFGISM

本節では、P-EFGISMの構成要素である、Shrink操作・Allocate操作を示す。次にP-EFGISMを紹介し、最後にその性質およびその証明概要を示す。

Shrink操作は、パイ P から $[a, b]$ を除き、 V_T を再構成する操作である。Shrink($P, [a, b], V_T$)とかく。図3に実際に適用した様子を示す。(7)から(9.1)の部分を除くと、 V_1 は(8.1)のようになる。紙面の都合上、詳細は省略する。

次にAllocate操作を示す。 G^{K_j} 上で $v_{\sigma_j(i')}$ からのパスが存在する点集合を $V(v_{\sigma_j(i')})$ とかけば、以下のようになる。

<p>アルゴリズム 2 Allocate</p> <p>Allocate(T, V_T, P)</p> <p>$h \leftarrow U\text{-Exp}(T, V_T, P);$</p> <p>If $DOM(h) = DOM(V_T)$ //U-Expで割当が求まった場合</p> <p> For each $[a_i, b_i] \in h$ do</p> <p> $A_i^M(V_N) \leftarrow [a_i, b_i];$</p> <p>Else</p> <p> $L = v_{\sigma_h(i')} \in V(v_{\sigma_h(n_h)})$において最小の$i'$</p> <p> $x = (\beta_{\sigma_h(n_h)} - (k - L + 1)\Phi(V_T) + P) \bmod P ;$</p> <p> For $i = L$ to n_h do</p> <p> $V_{\sigma_h(i)} \leftarrow V_{\sigma_h(i)} \cap [x, \beta_k];$ 図3(7)</p> <p> $T' = \{\sigma_h(L), \sigma_h(L + 1), \dots, \sigma_h(n_h)\};$</p>
--

<p>$V_{T-T'} \leftarrow \text{Shrink}(P, DOM(V_{T'}), V_{T-T'});$ 図3(8.1)</p> <p>Allocate($T - T', V_{T-T'}, [0, P - DOM(V_{T'})];$ 図3(8.2)</p> <p>Allocate($T', V_{T'}, [0, DOM(V_{T'})];$ 図3(9.2)</p>
--

実行例を示す。まず、一般化拡張プロセスを行う図1(6)。それに対し、評価区間を縮小する図3(7)。これから、 $DOM(V_{\{2,3\}})$ をシュリンクし、 V_1 を再構成する図3(8.1)。これと、 $P = [0, |P| - |DOM(V_{\{2,3\}})|]$ で子問題1を作成する図3(8.2)。また、図3(7)から、 V_2, V_3 と、 $P = [0, |DOM(V_{\{2,3\}})|]$ で子問題2を作成する図3(9.2)。

Allocateの性質として本研究では、以下を証明した。補題1 既約な V_T に対し、Allocate(T, V_T, P)は以下をみたす。

- $\sum_{x \in A_i^M(V_N)} |x| = \Phi(V_T), A_i^M(V_N) \subseteq V_i, \forall i \in T$
- $(A_i^M(V_N))_{i \in T}$ は $DOM(V_T)$ を高々 $2(|N| - 1)$ カットで分割したものである。

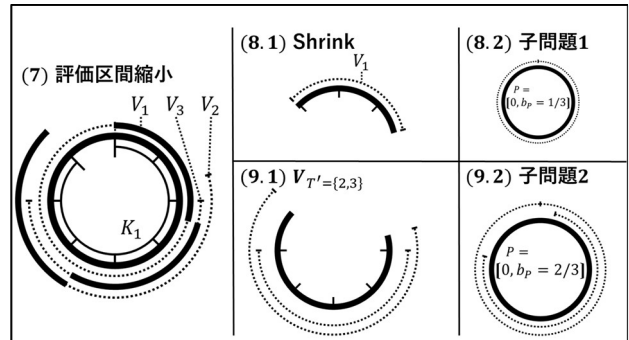


図3 Allocate

最後にP-EFGISMおよびその性質を以下に表す。

<p>アルゴリズム 3 P-EFGISM</p> <p>P-EFGISM(N, V_N, P)</p> <p>If $P \neq \phi$</p> <p> $V_T = \text{argmin}_{T' \subseteq N} \{\Phi(V_{T'})\} // V_T$は必ず既約になる</p> <p> Allocate(T, V_T, P)</p> <p> $V_{N-T} \leftarrow \text{Shrink}(P, DOM(V_T), V_{N-T})$</p> <p> P-EFGISM($N - T, V_{N-T}, [0, P - DOM(V_T)]$)</p>

これに対し、補題1を用いて、以下の性質を証明できる。定理1 P-EFGISMは無羨望性・戦略的操作不可能性を満たす割当を高々 $2(|N| - 1)$ のカット数で達成する。

証明概要

(カット数は高々 $2(|N| - 1)$) 二つの子問題に分割していることに注目することで、帰納法により示すことができる。

(無羨望性・戦略的操作不可能性) Chenら[4]は、各ステップで密度最小の V_T を探し、それに対して、 $DOM(V_T)$ を $\sum_{x \in A_i^M(V_N)} |x| = \Phi(V_T), \forall i \in T$ かつ $A_i^M(V_N) \subseteq V_i, \forall i \in T$ を満たすように割当を行うメカニズム M は無羨望性・戦略的操作不可能性をもつことを示した。ゆえに補題1より、P-EFGISMは2つの性質をみたす。

参考文献

- [1] J. B. Barbanel, S. J. Brams, W. Stromquist, "Cutting a Pie is Not a Piece of Cake", *American Mathematical Monthly*, Vol.116, No.6 (2009).
- [2] R. Alijani, M. Farhadi, M. Ghodsi, M. Seddighin, A. S. Tajik, "Envy-free mechanisms with minimum number of cuts", *AAAI-17*, pp.312-318, (2017).
- [3] 梅田 博之, 浅野 孝夫, "ホールケーキ分割問題に対するカット数最小の無羨望メカニズム", IPSJ 第80回全国大会, (2018).
- [4] Y. Chen, J. K. Lai, D. C. Parkes, A. D. Procaccia, "Truth, justice, and cake cutting", *Games and Economic Behavior*, Vol.77, No.1 (2013).