弱欲張り算法

高畑貴志　中村政隆
東京大学大学院総合文化研究科　東京大学教養学部
広域科学専攻広域システム科学系　システム科学系
〒153 東京都目黒区駒場 3-8-1
{takashi,nakamura}@klee.c.u-tokyo.ac.jp

我々は, 欲張り算法を拡張し, 弱欲張り算法を考案した. このアルゴリズムは, マトロイドやその拡張であるデルタマトロイドよりも広いクラスでの, 組合せ最適化問題を正しく解くことができる. ある種の２対２交換可能性を満たす離散システムは, このクラスに属する. このシステムの, ランク関数による特徴付けを示す. また, このシステムの具体例で, デルタマトロイドにならないものを提示する.

WEAKLY GREEDY ALGORITHM

Takashi Takabatake　Masataka Nakamura
Graduate Division of International and Interdisciplinary Studies,　Department of System Studies
University of Tokyo, Komaba, Meguro-ku, Tokyo 153, Japan
{takashi,nakamura}@klee.c.u-tokyo.ac.jp

We develop weakly greedy algorithm as an extension of the greedy algorithm. It gives a correct answer of a combinatorial optimization problem on a discrete system in a properly wider class than the class of matroids and delta-matroids. Discrete systems with a certain 2 to 2 exchangeability belong to this class. We characterize these systems in terms of their rank function. We give two examples of these systems which are not delta-matroids.

1 はじめに

マトロイドは, 多様な組み合わせ構造を統合する重要な概念である [4]. マトロイドとその拡張であるデルタマトロイドやジャンプシステム等では, ある種の組合せ最適化問題が, 欲張り算法により常に正しく解けることが知られている [1, 2, 3]. これらの組合せ構造は, 「欲張り」で特徴付けられているとも言える.

我々は欲張り算法を拡張し, 弱欲張り算法と名付けたアルゴリズムを考案した. このアルゴリズムは, デルタマトロイドを真に含むクラスでの組み合わせ最適化問題に適用可能である. 特に, ある種の 2 対 2 交換可能性が保証されている離散システムがこのクラスに属す. 我々はこの離散システムをペアデルタマトロイドと名付け, ペアデルタマトロイドを特徴付けるランク関数の性質を記述した.

2 デルタマトロイドと欲張り算法

有限集合 S と空でない S の部分集合族 $\mathcal{F}$ との組 $(S,\mathcal{F})$ を離散システムと呼ぶ. S を台集合と呼び,$\mathcal{F}$ の要素を独立集合と呼ぶ.

離散システム $(S,\mathcal{F})$ と, 重み関数 $w : S \to \mathrm{R}$ が与えられた時, 次の形式の問題を考えることができる.

$w(X)$[1]を最大にする $X \in \mathcal{F}$ を求めよ.

以後, この問題を単に最適化問題と呼び,$(S,\mathcal{F},w)$ で表す.

最適化問題の解を求めるのに, ある種の欲張り算法が使われることがある.

欲張り算法

Input An optimization problem $(S,\mathcal{F},w)$.

Output A subset $X \in \mathcal{F}$ which maximizes $w(X)$.

Step1 Consider an optimization problem $(S,\mathcal{F}\Delta N,w')$, where $w'(i) = |w(i)|$ $(i = 1,\ldots,n)$. Sort the support $S = \{1,2,\ldots,n\}$ so that $w'(i) \geq w'(i+1)$ $(1 \leq i \leq n-1)$, where n denotes $|S|$.

Step2 $X := \phi$.

Step3 For $i := 1$ to n repeat,

$$X := X \cup \{i\} \text{ if there exists } F \in \mathcal{F}\Delta N \text{ such that } X \cup \{i\} \subseteq F.$$

[1] $w(X)$ は $\sum_{i\in X} w(i)$ を表わす.

End Return $X\Delta N$ as a solution.
(注意) $X\Delta Y$ は,$(X\cup Y)\setminus(X\cap Y)$ を表し, $\mathcal{F}\Delta N$ は $\{F\Delta N|F\in\mathcal{F}\}$ を表す.

欲張り算法は, 常に最適化問題の正解を与えるわけではない. しかし,$(S,\mathcal{F})$ がマトロイドを拡張した, デルタマトロイドと呼ばれる離散システムならば, 欲張り算法は常に正しい答えを与える [1, 3].

定義 2.1 デルタマトロイドとは, 次の条件を満たす離散システム $(S,\mathcal{F})$ とする.

For all $X,Y\in\mathcal{F}$ and for all $x\in X\Delta Y$, there exists $y\in X\Delta Y$ such that $X\Delta\{x,y\}\in\mathcal{F}$.

欲張り算法には, ある集合 A を含み別の集合 B と交わらない独立集合が存在するかどうかを, 判断する手続きが必要になる.Bouchet はこの手続きを分離オラクルと名付けた [1]. 欲張り算法は, 分離オラクルを台集合の位数と同じ回数使用して終了する.

3 弱欲張り算法

欲張り算法は, ある順序のもとで, 最初に来る独立集合を答えとして返す, アルゴリズムといえる. 非負の重み関数の下で, 集合 X,Y の順序は,$X\backslash Y,Y\backslash X$ に属する, 最も重い要素の重みの比較によって決まる.

弱欲張り算法は, この順序を,$X\backslash Y,Y\backslash X$ それぞれから, 最も重い 2 つずつの要素の, 重みの和の比較で決定される順序に置き換えて, その新しい順序のもとで, 最初に来る独立集合を答えとして返すアルゴリズムである.

このアルゴリズムで, 台集合の部分集合を線形に繋げた, リストを使用する. $L=(X_1,\ldots,X_n)$ は, リスト L が $X_1,\ldots,X_n$ を, この順に含むリストであることを意味する.Y を追加するとは,$L:=(X_1,\ldots,X_n,Y)$ とすることであり,X_i を削除するとは $L:=(X_1,\ldots,X_{i-1},X_{i+1},\ldots,X_n)$ を, 意味する.$L=()$ は, リスト L が空であることを示す. リスト L に含まれる集合の数を $|L|$ で表す.

このアルゴリズムでも, 分離オラクルを使用する. 集合 A を含み, 集合 B と交わりを持たない独立集合が存在する時,(A,B) は,potentially feasible であると呼ぶことにする.

弱欲張り算法

Input An optimization problem $(S,\mathcal{F},w)$.
Output A subset $X\in\mathcal{F}$ which maximizes $w(X)$.

Step1 Consider an optimization problem $(S, \mathcal{F}\Delta N, w')$, where $w'(i) = |w(i)|$ $(i = 1, \ldots, n)$. Sort the support $S = \{1, 2, \ldots, n\}$ so that $w'(i) \geq w'(i+1)$ $(1 \leq i \leq n-1)$, where n denotes $|S|$.

Step2 $L(0) := (\phi)$.

Step3 For $i := 1, \ldots, n$ repeat,

> **Sub-Step1** $L(i) := ()$.
>
> **Sub-Step2** For $j := 1, \ldots, |L(i-1)|$ repeat,
>
> > Let A_j be the j'th subset of $L(i-1)$.
> >
> > Let A_j^+ (A_j^-) denote $A_j \cup \{i\}$ (A_j).
> >
> > And let B_j^+ (B_j^-) denote $\{1, \ldots, i\} \backslash A_j^+$ $(\{1, \ldots, i\} \backslash A_j^-)$.
> >
> > If (A_j^+, B_j^+) is potentially feasible then;
> >
> > > Add A_j^+ to $L(i)$.
> >
> > end if.
> >
> > If (A_j^-, B_j^-) is potentially feasible then; p
> >
> > > `flag` := TRUE.
> > >
> > > While $L(i)$ contains subsets except A_j^+ that contain i and `flag` = TRUE, repeat;
> > >
> > > > Let X be the first subset in $L(i)$ that contain i.
> > > >
> > > > If $w'(A_j) > w'(X)$ then;
> > > >
> > > > > Delete X from $L(i)$.
> > > >
> > > > else;
> > > >
> > > > > `flag` := FALSE.
> > > >
> > > > end if.
> > >
> > > end while.
> > >
> > > If no subsets in $L(i)$ except A_j^+ contain i, then add A_j^- to $L(i)$.
> >
> > end if.

End Find an independent subset $A \in L(n)$ which maximizes $w'(A)$. Return $A\Delta N$ as a solution.

このアルゴリズムは,デルタマトロイドを真に含むクラス上で定義される最適化問題を,任意の重み関数の下で正しく解く.我々は次の定理を証明した.

定理 3.1 離散システムが $(S,\mathcal{F})$ が,次の条件を満たすならば,弱欲張り算法は任意の重み関数 $w : S \to R$ の下で,最適化問題 $(S,\mathcal{F},w)$ を正しく解く.

> *For all* $X,Y \in \mathcal{F}$ *and for all* $x,y \in X\Delta Y (x \neq y)$, *there exist* $s,t \in X\Delta Y (s \neq t)$ *such that either* $X\Delta\{x,y,s,t\} \in \mathcal{F}$ *or* $X\Delta\{x,s\}$, $X\Delta\{x,t\} \in \mathcal{F}$.

デルタマトロイドが定理 3.1の条件を満たすことは,簡単に確かめられる.従って,弱欲張り算法は,欲張り算法の拡張であるといえる.

次に,このアルゴリズムに必要な時間のオーダーを考察する.台集合の位数を n で表す.分離オラクルは 1 ステップと考える.

最も主要な Step3 では,Sub-Step1,2 が n 回繰り返されるが,1 は一定の時間で終了する. 2 では,1 つの手続きが $L(i-1)$ 回繰り返される.この手続きは,部分集合の対が potentially feasible かどうかを,分離オラクルを用いて決定する.2 回以下の追加と何回かの削除が,リスト $L(i)$ に実施される.ところで Sub-Step2 を通じて,削除は $2|L(i-1)|$ 回以下である.従って,Sub-Step2 では,$O(|L(i-1)|)$ の時間が掛かる.$|L(i)|$ は Step3 の各繰り返しで,1 づつしか増えないので,$|L(i)| \leq i$ である.結局,Step3 全体で,$O(n^2)$ の時間が掛かる.

従って,弱欲張り算法は $O(n^2)$ の時間を必要とするアルゴリズムである.

4 ペアデルタマトロイド

この節では,弱欲張り算法が必ず成功する離散システムのうち,2 対 2 交換可能性が保証される,興味深いクラスについて考察する.

定義 4.1 離散システム (S,F) が,ペアデルタマトロイドであるとは,$\mathcal{F}$ が次の条件を満たすことである.

> For all $X,Y \in \mathcal{F}$ and for all $x,y \in X\Delta Y (x \neq y)$, there exist $s,t \in X\Delta Y$ such that $X\Delta\{x,y,s,t\} \in \mathcal{F}$.

デルタマトロイドが,ペアデルタマトロイドであるのを確かめるのは容易である.

ペアデルタマトロイドはランク関数のある性質により特徴付けられる.その性質を簡単に記述するために,ここでは既存の定義 [3] とは異なる[2]定義を用いる.

[2]本質的には同じものであり,簡単な式により互いに翻訳できる.

定義 4.2 離散システム $(S,\mathcal{F})$ のランク関数とは,(1) で定義される $S\times S$ から非負の整数への関数 r とする.

$$r(A,B)=\max_{X\in\mathcal{F}\Delta B}(|A\cap X|) \tag{1}$$

このランク関数を用いて,ペアデルタマトロイドは特徴付けできる.

定理 4.1 r をペアデルタマトロイド $(S,\mathcal{F})$ のランク関数とすると, 次の性質が満たされる. 逆に, もし離散システム $(S,\mathcal{F})$ のランク関数 r が次の性質を満たすならば, そのシステムはペアデルタマトロイドである.

> *For all* $A,B\subseteq S$, $e_1,e_2,e_3\in S$, *and* $p,q\in A$ $(p\neq q)$ *that satisfy* $r(A,B)=|A|\geq 2$, $r(A\cup\{e_i\},B)=r(A,B)$ $(i=1,2,3)$ *implies* $r((A\cup\{e_1,e_2,e_3\})\backslash\{p,q\},B)\leq r(A,B)$.

ペアマトロイドのクラスはデルタマトロイドを含むが, それ以外の組合せ構造も含む. その例を二つ挙げる.

例 4.1 有限集合 S の部分集合のうち, 位数が 4 の倍数であるものの集まりを $\mathcal{F}$ とする. 離散システム $(S,\mathcal{F})$ は, デルタマトロイドではないが, ペアデルタマトロイドになる.

例 4.2 S をある人々の集合とする. 偶数人の男と偶数人の学生を含む S の部分集合族を $\mathcal{F}$ とする. この時,$(S,\mathcal{F})$ は, デルタマトロイドである.

参考文献

[1] André Bouchet. Greedy algorithm and symmetric matroids. *Mathematical Programming*, 38:147–159, 1987.

[2] André Bouchet and William H. Cunningham. Delta-matroids, jump systems and bisubmodular polyhedra. preprint, 1991.

[3] R.Chandrasekaran and Santosh N.Kabadi. Pseudomatroids. *Discrete Mathematics*, 71(3):205–217, 1988.

[4] András Recski. *Matroid Theory and its Applications in Electric Network Theory and in Statics.* Springer-Verlag, 1989.

[5] Takashi Takabatake. Weakly greedy algorithm, weak-Δ-matroid and its subclasses. Master's thesis, University of Tokyo, 1995.