

要素間に先行順位をもつシステムの配置問題

正員 加地 太一 (小樽商科大学)

正員 大内 東 (北海道大学)

Assignment Problem for System with Precedence Relation among Elements

Taichi Kaji, Member (Otaru University of Commerce), Azuma Ohuchi, Member (Hokkaido University)

We think about the problem to assign the elements to an ordered sequence of stations such that the precedence relations are satisfied. At this time, we obtain the best evaluation value decided under a certain constrained condition to the assignment. This kind of problem includes the line balancing problem etc. In addition, this problem can adjust to various problems by changing constrained condition and objective function. These problems can be shown as a problem for sequential partitions of the nodes of a directed acyclic graph into subsets. We especially consider problem for finding a minimum total cost of the cut edge under the restriction of the size of block.

In this paper, we propose the general framework for sequential partitions of directed acyclic graphs. And we describe an efficient algorithm that can be used to reduce computational requirements and, possibly storage. We estimate that complexity of the algorithm is the polynomial order, if structure of directed acyclic graphs is near parallel.

キーワード：配置問題、系列分割、動的計画法、無閉路有向グラフ、先行順位

1. はじめに

先行順位のある要素をその先行順位の制限を無視することなく、いくつかのグループに配置する問題を考える。このとき、各グループに課せられた制約条件のもとで、最良の評価値をもつ配置を求めるものとする。典型的な事例としてライン・バランスングの問題^{(1),(4)}、工場配置の問題などがある。例えば、ライン・バランスングの問題は先行順位の制限を無視せずに、要素作業を作業ステーションの数が最小になるように各作業ステーションに割り当てることであり、工場配置問題では工程の先行順位を無視することなく各工場間の中間製品の全移動コストの最小化を達成する配置を求める。この種の問題は各グループに課せられた制約条件、配置の評価関数などを変えることにより、種々の配置問題に適応させることが可能である。これらの問題はすべて無閉路有向グラフの頂点を先行順位の関係を無視せずに各成分に配置する基本的分割問題の範疇に属するものと考えられる。この基本的分割問題を系列分割問題⁽²⁾と言うことにする。

本論文では、系列分割問題の一般的定式化と、その実例として無閉路有向グラフを系列分割するとき生ずるカット・エッジのコストの総和を最小化する問題を取り扱う。今後、この問題を“総カット値を最小化する系列分割問題”と呼ぶ。その事例として、次のような問題が考えられる。先行順位を持ついくつかの生産プロセスからなるシステム

を考える。各要素プロセスは、その先行プロセスの生産物を入力として受け取り生産活動を行い、その産出物をそれを必要とする後続プロセスに渡す。また、各プロセスは生産活動のために一定の資源量を必要とするが、1ステーションあたり使用できる資源量は定まっているとする。中間製品の移動には各ステーション間で、ある輸送コストを必要とするが、同一ステーション内の輸送コストは無視できるものとする。このとき、各生産プロセスを輸送コストの総和が最小になるように、先行順位の制限を無視せずに、ステーションごとに利用可能な資源量の範囲内で各ステーションに配置する問題が挙げられる。

本論文では、上記の問題に対する解を与えることを目標に、まず単一の入口と出口を持つ無閉路有向グラフ $D(V, E)$ に対して、系列を保持する部分グラフ、頂点集合 V の切断、 V の互いに素な部分集合の切断による分離などの定義と記法を説明し、系列分割の概念を導入する。さらに、任意の系列分割が切断の鎖によって一意に定まることを示す。次に第3章では単一の入口と出口を持つ無閉路有向グラフの各頂点に重み、各辺にコストを付与したネットワークを構成することによって総カット値を最小化する系列分割問題を考察する。さらに、問題の定式化を容易にするために、分割を一意に表す切断決定子を導入し、切断の鎖を切断決定子列で一意に表現する。また、評価関数を定義し問題の定式化を行う。さらに、第4章では、すべての切断を決定するために切断を表す切断決定子を節点とする木を生成す

る。このための分枝規則を与える基本操作について述べ、基本操作を高速に実現するための関数を導入する。次に無駄のない探索方法を実現するために木を縮約し既約グラフとする。この既約グラフ上で動的計画法にもとづき、この問題の性質を加味した最適化算法を構成する。最後に第5章では細分化禁止則による効果を示し、さらに並列に近い構造のグラフに対して、算法の計算量が多項式オーダーになることと、中間エッジの挿入がこの計算量の負担を軽減することについて述べる。

2. 無閉路有向グラフの系列分割

先行順位をもつシステムをその先行順位を考慮して配置する問題はシステムの要素を頂点、その先行順位を有向辺で表す無閉路有向グラフ $D(V, E)$ 上での先行順位を保持した分割と考えられる。

単一の入口と出口を持つ無閉路有向グラフ $D(V, E)$ が与えられたとき、 D の有向辺が定める V 上の関係の反射的かつ推移的な閉包をとって得られる順序関係を \preceq とする。順序関係 \preceq は D が無閉路であることから反対称性をみだし、半順序関係となる。このようにして、 D から得られる半順序集合を (V, \preceq) で表す。また、 $v \preceq u$ かつ $v \neq u$ を $v < u$ で表す。

定義 1. 空でない部分集合 $A \subset V$ から誘導された $(3)D$ の部分グラフを $\mathcal{D}(A)$ で表す。 $\mathcal{D}(A)$ の任意の2点を結ぶ D 内の有向路がすべて $\mathcal{D}(A)$ の有向路となるとき、 $\mathcal{D}(A)$ は系列を保持する D の部分グラフであるという。

補題 1. $\mathcal{D}(A)$ が系列を保持する D の部分グラフであるとき、 $\mathcal{D}(A)$ から導かれる半順序関係 \preceq_A は A に制限された D の半順序関係 \preceq に等しく、 $(A, \preceq) = (A, \preceq_A)$ となる。

定義 2. V の部分集合 A が、 $A^\circ \times A$ から選んだ2元対 (x, y) に関して、 x と y が \preceq において比較可能ならば常に $x < y$ が成立するとき、 $\mu = (A^\circ, A)$ を A によって定まる V の切断という。そして、 A をこの切断の上組、 A° を下組という。

定義 3. 2つの切断 (A°, A) 、 (B°, B) に対して $A \supseteq B$ が成立するとき、 (A°, A) は (B°, B) の前にあるといい、 $(A^\circ, A) \prec (B^\circ, B)$ で表す。また $A \supset B$ のとき、真に前にあるといい $(A^\circ, A) \prec \neq (B^\circ, B)$ で示す。

定義 4. V の互いに素な部分集合 X と Y がそれぞれ V のある切断の下組と上組に含まれるならば、 X と Y は切断により分離されるといい、 $X|Y$ で表す。 X と Y について $X|Y$ または $Y|X$ が成り立つとき、 X と Y は分離可能であるという。

$X|Y$ は定性的には“ X が Y より前にある”ことを、また X と Y を結ぶ辺が存在するときには“それらの辺はすべて同じ向きをもつ”ことを意味している。ここで、無閉路有向グラフの系列分割を次の様に定義する。

定義 5. $D(V, E)$ の頂点集合 V の分割 $\{V_1, V_2, \dots, V_k\}$ がその任意の成分集合 V_i, V_j に対し、 $i < j$ のとき $V_i | V_j$ が成立し、かつ $V_1 | V_2 | \dots | V_k$ を満たすように並べることができるとき、この分割を $D(V, E)$ の系列分割という。

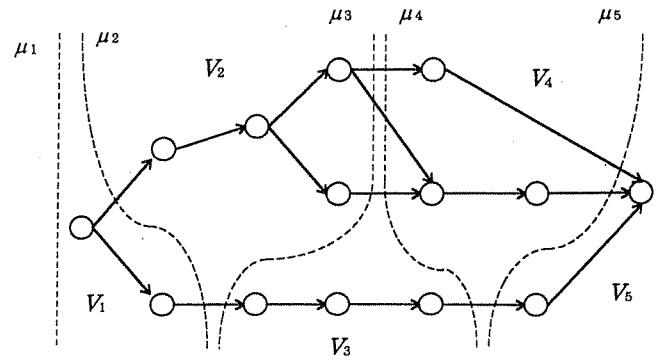


図1 無閉路有向グラフの系列分割

Fig.1. Sequential partitions of a directed acyclic graph.

このとき、各分割成分 V_i は系列保持の性質をもつ。図1は無閉路有向グラフの系列分割の一例であり、切断を破線で表し、その右側が上組、左側が下組となることを示す。

系列分割 $V_1 | V_2 | \dots | V_k$ が与えられたとき、各成分集合 V_i に上組 $A_i = V_i \cup V_{i+1} \cup \dots \cup V_k$ 、下組 $A_i^\circ = V_1 \cup V_2 \cup \dots \cup V_{i-1}$ からなる切断 $\mu_i = (A_i^\circ, A_i)$ を対応づけると、 $A_i \supseteq A_{i+1}$ が成り立つことより、関係“ $\prec \neq$ ”に関する切断の列

$$\mu_1 \prec \neq \mu_2 \prec \neq \dots \prec \neq \mu_k \quad \dots \quad (1)$$

が得られる。このような引き続き2つの切断 μ_i, μ_{i+1} が常に関係 $\prec \neq$ で結ばれている切断の単調列のことを切断の鎖と呼ぶ。逆に、切断の鎖(1)が与えられたとき、隣り合う切断の上組をそれぞれ A_i, A_{i+1} として、 $V_i = A_i - A_{i+1}$ を求めると、系列分割 $V_1 | V_2 | \dots | V_k$ が定まる。以上のように、無閉路有向グラフの系列分割と切断の鎖は一対一に対応しており、任意の系列分割は切断の鎖によって一意に定まる。

3. 総カット値を最小化する系列分割問題の定式化

総カット値を最小化する系列分割問題で考えるネットワークは、多重辺をもたない単一の入口と出口を持つ n 個の頂点からなる無閉路有向グラフ $D(V, E)$ であり、すべての頂点 $v \in V$ には重み $w(v)$ が、各有向辺 $(v_1, v_2) \in E$ にはコスト $c(v_1, v_2)$ が付与されている。これらの値は任意の $v \in V$ および $(v_1, v_2) \in E$ について、各々条件 $0 < w(v) \leq B, c(v_1, v_2) \geq 0$ を満たす整数であり、また B はブロックサイズと呼ばれる問題に固有な正の整数である。

無閉路有向グラフ $D(V, E)$ から導かれる半順序集合 (V, \preceq) の切断を $\mu = (A^\circ, A)$ とすると、 A に制限した半順序集合 (A, \preceq) の極小元の集合を切断 μ の切断決定子 γ と呼ぶ。このように、切断が与えられたとき切断決定子 γ は一意に定まり、逆に、 γ が与えられたとき、 $A = \bigcup_{u \in \gamma} \{v | v \succeq u\}$ とするこ

とによって切断 (A°, A) が復元でき、切断と切断決定子の関係は1対1である。それゆえ、切断の鎖 $\mu_1, \mu_2, \dots, \mu_k$ は切断決定子の列 $\gamma_1, \gamma_2, \dots, \gamma_k$ によって表現できる。また、この

鎖から生成される系列分割の成分集合の一つを V_i とすると、それから誘導される D の部分グラフ $\mathcal{D}(V_i)$ をブロックと呼ぶ。この切断の鎖で $i < j$ のとき、 μ_i, μ_j の上組集合 A_i, A_j の差集合 $A_i - A_j$ を記号 $[\gamma_i, \gamma_j]$ で表す。この記法によれば、 $V_i = [\gamma_i, \gamma_{i+1}]$ と書ける。また、 V_i に属するすべての頂点 v の重み $w(v)$ の総和 $||[\gamma_i, \gamma_{i+1}]|| = \sum_{v \in V_i} w(v)$ をブロック長

という。一方、どのブロックにも属さない D の辺の集合をカット・セットという。カット・セットに属する辺のコストの総和は

$$f_c(\gamma_1, \gamma_2, \dots, \gamma_k) = \sum_{i=1}^{k-1} C(\gamma_i, \gamma_{i+1}) \quad \dots (2)$$

と書き表すことができる。ここで、 $C(\gamma_i, \gamma_{i+1})$ は切断決定子 γ_i の後ろに γ_{i+1} を置いたときの増分コストと呼ばれる量であり、始点が V_i の中にあり、終点が V_j の中にあるすべてのカット辺のコストの総和を表している。

$$C(\gamma_i, \gamma_{i+1}) = \sum_{\substack{v \in [\gamma_i, \gamma_{i+1}] \\ u \in P}} c(v, u) \quad \dots (3)$$

ここで、 $P = V_{i+1} \cup \dots \cup V_k$ である。

以上より、カット値の総和を最小にする系列分割問題はブロック長がブロックサイズ B を越えないという制約のもとで、カット・セットに属する辺のコストの総和を最小にする系列分割を求める問題であり、制約条件 $||[\gamma_i, \gamma_{i+1}]|| \leq B$ を満たすすべての切断決定子列の集合の上で、目的関数 $f_c(\gamma_1, \gamma_2, \dots, \gamma_k)$ を最小化する問題として定式化される。

4. 算法構成

本章では切断の鎖の生成過程を可能解に至る部分解の列挙のプロセスと考えて、すべての切断からなる構造木を構成し、これより同一の切断決定子が重複しない縮約した既約グラフを導出する。この上で問題固有の性質を取り入れた動的計画法による算法について論じる。まず、本章で用いる記号等について述べる。

点 v を終点(始点)とする辺 (u, v) ((v, u))を v の流入辺(流出辺)と呼ぶ。 V の切断を $\mu = (A^c, A)$ とすると、点 $v \in A$ に対して D での v の入り次数を $d^-(v)$ で、 A^c に始点をもつ v への流入辺の個数を $d^-_{\mu}(v)$ で表す。 $d^-_{\mu}(v) > 0$ を満たす点 v を A の境界点といい、その集合を $\partial A = \{v | v \in A, d^-_{\mu}(v) > 0\}$ とする。また、 $A - \partial A$ を A の内点の集合という。 A を定義域とする関数 $h_{\mu}(v) = d^-(v) - d^-_{\mu}(v)$ を定義する。その値は上組に始点をもつ v への流入辺の個数を表す。半順序集合 (A, \preceq) の極小元 v は $h_{\mu}(v) = 0$ を満たす点であり、切断決定子は定義より $\gamma = \{v | v \in A, h_{\mu}(v) = 0\}$ と書き表される。 A と ∂A および γ の間には包含関係 $\gamma \subseteq \partial A \subseteq A$ が成立する。また、境界条件に関する記述の統一性とプログラムの便宜上、入口の前にダミーの頂点を一個加え、切断 $\mu_{k+1} = (V, \emptyset)$ を表す γ_{k+1} を切断決定子列の末尾に常に加える。

<4. 1> 切断決定子の生成

切断 $\mu = (A^c, A)$ に対応する切断決定子を γ とする。このとき、下組 A^c の頂点数を μ または γ のレベルと呼ぶ。前後関係 \preceq について μ より後にあり、かつレベルが1だけ大であるような切断を μ' とする。 μ' は (A, \preceq) の極小元の一つを選び、それを下組 A^c に移すことによって得られる。すなわち、 $v \in \gamma$ をピボットとして選び、

$$A' = A - \{v\} \quad \dots (4)$$

と置けば、

$$\mu' = ((A')^c, A') \quad \dots (5)$$

となる。この操作を v をピボットとする切断 μ に対する基本操作という。 (A, \preceq) において、ピボット v を始点とするすべての辺の終点の集合を W とする。すなわち、 $v < u$ であるが $v < x < u$ をみたす x が存在しないことを $v \ll u$ で表すと、 W は次のように書ける。

$$W = \{u | u \in A, v \ll u\} \quad \dots (6)$$

基本操作を行うと関数 $h_{\mu}()$ は定義域を A' とする関数 $h_{\mu'}()$ として新たに定義され、

$$u \in A' - W \text{ のとき、 } h_{\mu'}(u) = h_{\mu}(u)$$

$$u \in W \text{ のとき、 } h_{\mu'}(u) = h_{\mu}(u) - 1 \quad \dots (7)$$

となる。

補題 2. 切断 μ のレベルが1以上ならば、次式が成り立つ。

$$\gamma = \{w | w \in \partial A, h_{\mu}(w) = 0\}$$

証明: $\partial A \subseteq A$ より、 $\gamma = \{w | w \in A, h_{\mu}(w) = 0\} \supseteq \{w | w \in \partial A, h_{\mu}(w) = 0\}$ である。逆に $v \in \gamma$ とすれば $h_{\mu}(v) = 0$ であり、また D が単一の入口を持ち、 μ のレベルが1以上であることから、 $d^-(v) > 0$ が成立し、 $d^-_{\mu}(v) = d^-(v) > 0$ となる。これから、 $v \in \partial A$ が導かれ、逆の包含関係が満たされる。□

この補題と、関係 $\gamma \subseteq \partial A \subseteq A$ から、 (A, \preceq) および $(\partial A, \preceq)$ の極小元集合は γ と一致する。次に基本操作により変換された境界点の集合と切断決定子は以下の2つの定理によって導かれる。

定理 1. v をピボットとする切断 μ に対する基本操作により、 μ が μ' に変換されるとき、 ∂A は $\partial A' = (\partial A - \{v\}) \cup W$ に変換される。

証明: まず、 $\partial A' \supseteq (\partial A - \{v\}) \cup W$ を示す。右辺を次のように互いに素な部分集合に分ける。

$$(\partial A - \{v\}) \cup W = ((\partial A - \{v\}) - W) \cup W$$

$u \in (\partial A - \{v\}) - W$ のとき、基本操作でピボット v を下組に移しても、点 u への下組からの流入辺の個数は変わらない、それゆえ $d^-_{\mu'}(u) = d^-_{\mu}(u) > 0$ となる。また、 $u \in W$ のとき u への流入辺として (v, u) が常に存在するので、 $d^-_{\mu'}(u) \geq 1 > 0$ となる。いずれにしても u は $\partial A'$ に含まれる。逆に、 $\partial A' \subseteq (\partial A - \{v\}) \cup W$ を示す。“ $u \in \partial A'$ に対して、 $u \in (\partial A - \{v\}) \cup W$ ”と仮定すると、この仮定は“ $u \in A - \{v\}$ かつ $d^-_{\mu'}(u) > 0$ に対して、 $u \in (\partial A - \{v\})$ かつ $u \in W$ ”と書き直される。 $u \in A - \{v\}$ と $u \in (\partial A - \{v\})$ から $u \in A - \partial A$ となり、 $d^-_{\mu}(u) = 0$ が導かれる。また $u \in W$ であるから、 $d^-_{\mu'}(u) = d^-_{\mu}(u) = 0$ であり、 $d^-_{\mu'}(u) > 0$ に矛盾する。□

この定理1と上記の(7)式から、 v をピボットとする基本操作により μ が μ' に変換されるとき、 $h_{\mu}()$ の定義域を A から

∂A に縮小して得られる制限関数 $h_\mu(u) | \partial A$ は次の(8)式で与えられる $h_{\mu'}(u) | \partial A'$ に変換されることを容易に示すことができる。

$$\begin{aligned}
 & u \in (\partial A - \{v\}) - W \text{ に対して, } h_{\mu'}(u) | \partial A' = h_\mu(u) | \partial A \\
 & u \in W \text{ のときは} \\
 & u \in \partial A \text{ ならば, } h_{\mu'}(u) | \partial A' = h_\mu(u) | \partial A - 1 \\
 & u \in A - \partial A \text{ ならば, } h_{\mu'}(u) | \partial A' = d^-(u) - 1
 \end{aligned} \quad \dots \dots (8)$$

ここで、 μ の上組からの流入辺の個数が1である W の点の集合 W^* を

$$W^* = \{u | u \in W, h_\mu(u) = 1\} \quad \dots \dots (9)$$

と定義すると、次の定理が成立する。

定理 2. v をピボットとする切断 μ に対する基本操作により μ が μ' に変換されるとき、 γ が γ' に変換されるならば、

$$\begin{aligned}
 \gamma' &= \{u | u \in A', h_{\mu'}(u) = 0\} \\
 &= \{u | u \in \partial A', h_{\mu'}(u) = 0\} \quad \dots \dots (10) \\
 &= (\gamma - \{v\}) \cup W^*
 \end{aligned}$$

となる。

証明: 定理 1 で与えられた式の右辺を次のように互いに素な集合に分ける。

$$\begin{aligned}
 \partial A' &= (\partial A - \{v\}) \cup W \\
 &= ((\partial A - \{v\}) - W) \cup W
 \end{aligned}$$

ここで、関数 $h_\mu(u)$ の変換式(7)と補題 2 を用いると次の計算式が成り立つ。

$$\begin{aligned}
 & \{u | u \in \partial A', h_{\mu'}(u) = 0\} \\
 &= \{u | u \in (\partial A - \{v\}) - W, h_{\mu'}(u) = 0\} \\
 & \quad \cup \{u | u \in W, h_{\mu'}(u) = 0\} \\
 &= \{u | u \in (\partial A - \{v\}) - W, h_\mu(u) = 0\} \\
 & \quad \cup \{u | u \in W, h_\mu(u) = 1\} \\
 &= (\gamma - \{v\}) \cup W^* \quad \square
 \end{aligned}$$

(10)式より、 ∂A に制限した情報とそれに対応する関数 $h_\mu(u)$ を表す情報のみを持つことによって、 γ' は γ から、 v からの出次数に線形な計算負担で計算可能である。また ∂A をもつことによりコスト計算も効果的に行うことが可能になる。

次に、すべての切断を決定するために切断を節点とする木を生成する。切断 μ に対応する切断決定子 γ の点 v をピボットとして基本操作を行うと、 v の選択に応じて、1レベル高い切断 $\mu_1, \mu_2, \dots, \mu_r$ が作り出される。これらを親節点 μ を展開して得られる子節点と考える。親から子への展開操作を繰り返すと高さ $n+1$ の多分岐平衡木 T が生成できる。ただし、 n は頂点数である。図 3 に図 2 のグラフから作られた木 T を示す。ここでは切断のかわりに切断決定子によって節点を表している。この木の葉節点はレベル値が n の切断 (V, \emptyset) を表しており、また各節点にその節点を得るために用いたピボットをラベルとして付けると、根から葉に至る経路上の節点のピボットの列は頂点集合 V のある特定の一列化に対応する。

< 4. 2 > 既約グラフの生成

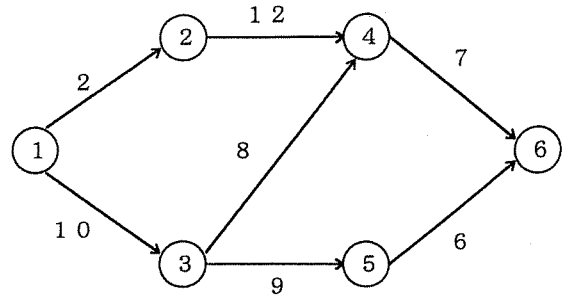


図 2 無閉路有向グラフ
Fig.2. A directed acyclic graph.

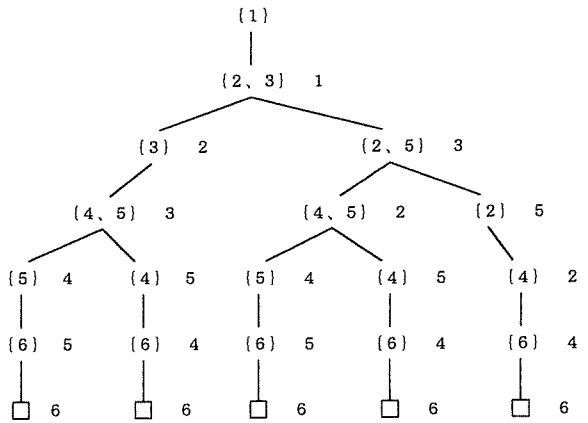


図 3 すべての切断決定子を生成する多分岐平衡木
Fig.3. Multiway balanced tree that produce all cut generators.

上で述べた切断を節点とする木 T では、同じ切断が多数重複して生成される。しかし、同一の切断を根とする部分木はすべて同じ構造を持つので、これらを重ねて一つの節点とし、既約グラフとして表現できる。図 4 に図 3 に対する既約グラフを示す。既約グラフは、辺の向きが基本操作における親から子への向きであり、入口と出口がそれぞれ1個で、レベル間隔1のレベル構造を持つ無閉路有向グラフである。

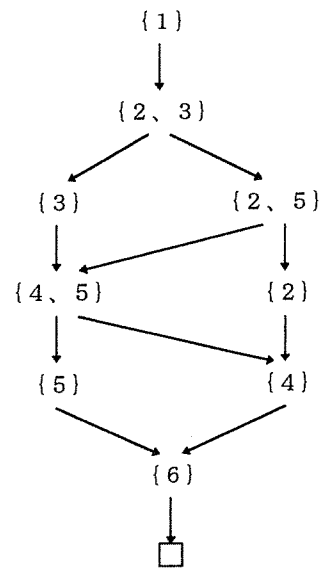


図 4 すべての切断決定子を含む既約グラフ
Fig.4. Reduced graph of cut generators.

もとの木 T で同じ切断を表わす節点はすべて同じ深さのレベル上にあるので、横型展開法を用いて効率良く既約グラフを生成することができる。この横型展開法に

基づく算法を図5に示す。この算法では、表現するのに多くのデータ量を必要とする切断を直接使う代わりに、より少ないデータ量で表現できる切断決定子を用いて節点を表わしている。このとき、計算過程で生成された各節点とその下組の頂点の重さの累積値を、親節点のピボットの重さを漸化的に加算していくことによって求め、それを記憶しておく。また横型展開の過程を制御するために、レベル値をキーとする優先順位待ち行列をオープンリストOpenListとして構成する。この算法の第5行 "αをピボットとしてγ_iからγ_gを生成する"、第7行の"γ_g ∈ OpenListを判断する"という二つの操作を効率良く実行することはプログラム実装上重要である。前者による計算負担の総量は前章の議論より無閉路有向グラフのエッジ数に比例する。後者のOpenListの判断については、次のようなOpenList上での節点の分布の性質を利用して効率的な判断が可能である。横型展開の構成法より、OpenList上で各活性化節点はレベル値について常に昇順に並び、現在のγ_iから生成されたγ_gと同レベルの切断決定子はOpenListの末尾に連続的にまとまって存在する性質を利用すると、γ_g ∈ OpenListの判断はOpenListを末尾から見ていくことによって効率的に行うこ

- 1: OpenList := γ_s;
- 2: While (OpenList ≠ φ) begin
- 3: "OpenListの先頭要素 γ_t を取り出す";
- 4: for (α ∈ γ_t) begin
- 5: "αをピボットとしてγ_tからγ_gを生成する";
- 6: "γ_gからγ_tへのポインタを確保する";
- 7: if (not(γ_g ∈ OpenList))
- 8: "γ_gをOpenListの末尾に加える";
- 9: end
- 10: end;

図5 既約グラフを生成する算法

Fig.5. An algorithm of production for reduced graph of cut generators.

- 1: f(γ_s) = 0;
- 2: while (Ω ≠ φ) begin
- 3: "Ωの先頭要素 γ_i を取り出す";
- 4: f(γ_i) = min_{γ_j} {f(γ_j) + C(γ_j, γ_i)};
- 5: "γ_iから最小値を与えるγ_jへのポインタを確保する";
- 6: end;

図6 動的計画法による算法

Fig.6. An algorithm using dynamic programming.

とができる。さらに、この性質を積極的に利用してより効率のよい判断も可能である。たとえば、ハッシュ法等を用いて計算効率をO(1)まで減ずることが可能である。

<4.3> 既約グラフ上の最適化算法

切断(∅, V)をμ_sとおき、切断の単調列μ_s <= ... <= μ_iに対応する切断決定子の列{γ_s, ..., γ_i}を系列分割問題の部分解と考える。最初の要素がγ_s、最後の要素がγ_iであるようなすべての可能部分解{γ_s, ..., γ_i}の集合をΞ(γ_i)で表す。ここで、部分解{γ_s, ..., γ_i}は既約グラフ上で、節点γ_sを入口とし節点γ_iを出口とする部分グラフの一つの系列分割を与える。部分解{γ_s, ..., γ_j, ..., γ_i}がΞ(γ_i)の中での最小コストを実現するならば、部分解{γ_s, ..., γ_j}が同時にΞ(γ_j)の中での最小コストを実現しなければならない。すなわち、最適性の原理が成立する。したがって、Ξ(γ_i)を状態と考えた動的計画法、もしくは最適性の原理を取り入れた分枝限定法^{(3),(6),(7)}によって効率のよい厳密解法を構築することができる。

図6に動的計画法による算法を示す。この算法に従い、順次最良部分解を確定していくことによって最適解が求まる。ここでf(γ_i)はΞ(γ_i)に属する部分解の最小コストを示す関数であり、またΩは既約グラフ上でレベルの小さい順に並べた(同レベル間においては順不同)切断決定子のリストを表現するものであり、既約グラフより容易に取り出せる。このとき、第4行のγ_j (<= γ_i)に関する最小値は、既約グラフ上で親節点を辿ることによって得られる節点γ_jから根γ_sに至るすべての経路上で| [γ_j, γ_i] | ≤ Bにより制約された範囲内で計算する。すなわちγ_jとγ_iのすでに既知である、それぞれの頂点に付与した“下組の頂点の重さの累積値”の差がBを越えない範囲内で、節点γ_jを出発点として縦型グラフ探索などを用い探索することとなる。プログラム全体の流れでは、既約グラフの切断決定子を表す各頂点に{γ_s, ..., γ_i}を表す部分解を割付け、根よりレベル順にf(γ_i)を確定していく。無閉路有向グラフの出口を表す最後の切断決定子γ_{k+1}が確定されたなら、最適解のコストが求り、計算過程で定めたポインタを逆順に辿ることによって最適解の切断決定子列による分割が求まる。

また、増分コストの定義式は(3)式であるが、この定義式を以下の漸化式で代替し、図6の第4行の計算過程で漸化的に使用することによって、計算効率を上げることが可能である。ピボットをvとする基本操作により切断決定子γ_jがγ_mに後退したとき、増分コストC(γ_i, γ_m)は次式のように書ける。

$$C(\gamma_i, \gamma_m) = C(\gamma_i, \gamma_j) + \sum_{t \in [\gamma_m, \gamma_{k+1}]} c(v, t) - \sum_{s \in [\gamma_i, \gamma_j]} c(s, v) \dots \dots (11)$$

ここで、 $\sum_{t \in [\gamma_m, \gamma_{k+1}]} c(v, t)$ はピボットvからの流出辺の総コストであり、 $\sum_{s \in [\gamma_i, \gamma_j]} c(s, v)$ は [γ_i, γ_j]内の要素を始点とするピボットvへの流入辺の総コストである。また、 [γ_i, γ_j]内の

頂点の認識は、無閉路有向グラフの隣接行列の推移的閉包を始めに一度求めておくことによって容易に実現できる。これによって、(11)式または定義式(3)においても効果的な計算が可能となり、その計算の総量はエッジ数に比例する⁽²⁾。

また、全カット値を最小化する本問題では、以下に述べるように図6の4行目の計算式における最小値を求める過程で制約として、細分化禁止則⁽⁷⁾を組み込むことによって計算効率の向上が可能である。すなわち、部分解を表す単調増大な切断決定子列

$$\gamma_1, \gamma_2, \dots, \gamma_i, \gamma_j, \gamma_m$$

に対して制約式

$$|[\gamma_i, \gamma_m]| \leq B$$

が満たされるならば、この部分解より、 γ_j を無視した

$$\gamma_1, \gamma_2, \dots, \gamma_i, \gamma_m$$

に対応する部分解のコストが低い。したがって、はじめの分割を考慮する必要はない。これを細分化禁止則という。この規則を取り入れることによって漸近的計算量は等しいが探索空間の実際の絞り込みによって最適化の計算の負担を軽減する算法を構成する。

5. 並列グラフに対する算法の特性評価

本算法の計算量に寄与する主な要因は既約グラフの生成と、そのグラフを利用した最適解の導出過程である。両過程における計算量は切断決定子の生成数に大きく依存する。したがって、この切断決定子の生成数に着目し考察する。ここで着目する切断決定子数は無閉路有向グラフの構造によって大きく影響され、理論的に切断決定子の生成数を求めることは困難である。しかし、無閉路有向グラフをある特定な構造に限定することによって推定は可能である。一

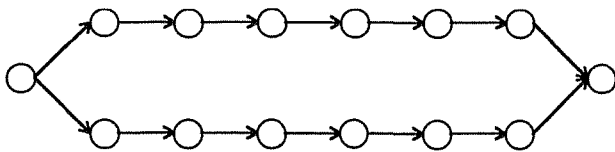


図7 2並列なグラフ

Fig.7. Digraph of only 2 paths.

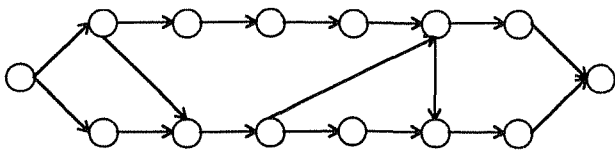


図8 中間エッジをもつ2並列なグラフ

Fig.8. Digraph of only 2 paths with crossing edges.

表1 頂点数に対しての細分化禁止則の評価 (ブロックサイズ=10)

Table 1. Results of experiments for number of nodes by rule of prohibition of partitioning.

頂点数	100	150	200	250	300	350	400
全最小値探索数	57027	105591	162242	265119	281770	371671	598996
細分化禁止則による削除数	5848	11589	14925	24833	27457	30565	52341

表2 ブロックサイズに対しての細分化禁止則の評価 (頂点数=100)

Table 2. Results of experiments for block size by rule of prohibition of partitioning.

ブロックサイズ	4	8	12	16	20	24	28
全最小値探索数	14500	40812	74689	112849	152985	194358	236110
細分化禁止則による削除数	624	4431	11648	17949	33963	52796	65231

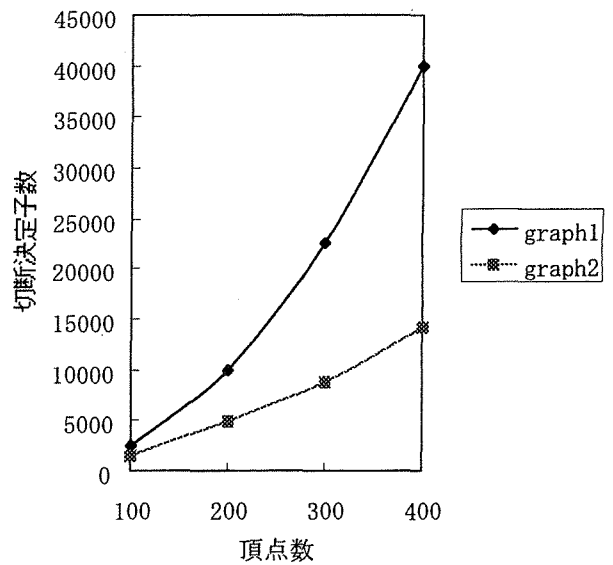


図9 頂点数と切断決定子数

graph1: 中間エッジがない2並列グラフ

graph2: 中間エッジをもつ2並列グラフ

Fig.9. Number of nodes vs. number of cut generators.

例として単一の入口と出口を持つ m 並列な構造をもち、並列線路間の接続辺(中間エッジ)が存在しないグラフ(図7参照)を対象として考察する。

まず、このとき切断決定子の要素数 $|\gamma|$ は並列数 m 以下となり、また前章で述べたエッジ数に比例する計算の負担は高々 $O(n^2)$ で、考えられる切断決定子数以下であることを考慮すると、既約グラフの生成過程においては4. 2章で述べた構成により、その計算量は切断決定子数に比例する。次に、既約グラフを利用した最適解の導出過程では無閉路有向グラフが並列構造を持つことによって、図6の4行目の計算過程のある段における最小値探索の数はブロックサイズ B のみに依存した定数以下となり、 B が定数であることより同様にその計算量は切断決定子数に比例する。すなわ

ち、本算法の全体の計算量は切断決定子数に比例する。このとき、切断決定子数は組合せ的考察により n^m に比例する数(2並列の場合は正確に $n^2/4+2$ を越えることはない)となる。したがって、その計算量は $O(n^m)$ と考えられ、多項式的増大を示す。さらに並列線路間に接続辺が存在する場合、切断決定子数が減少し計算量の負担はさらに減少する傾向が示せる。特に、入口に近い中間エッジは大きな減少をもたらす。図8に示すような中間エッジが頂点数の10%の割合でランダムに生成する2並列グラフに対して数値実験を試みた結果、切断決定子数の40から65%の減少が見られた。図9に示すように中間エッジが存在しない場合 $O(n^2)$ となるのに対して上記の多数の数値実験の平均値は $O(n^{1.58})$ の特性を示している。

また、細分化禁止則による候補者の絞り込みが最適解の導出過程に対して非常に有効であることを確かめた。それによると細分化禁止則の適用により約10%から30%の削除効果もたらされ、しかもこの効果はブロックサイズの増加とともに向上する傾向が判明した。ここでは2並列グラフ(中間エッジ数 $=0.2 \times$ 頂点数)に対する一例を表1に示す。このときブロックサイズを10に固定し、頂点数を100から400に変化させた場合に対して図6の4行目における最小値探索数の総量(全最小値探索数)と、そのうち細分化禁止則により削除された最小値探索数を示している。表2は頂点数100の2並列グラフ(中間エッジ数 $=0.2 \times$ 頂点数)でブロックサイズを変化させた場合の同様な事項について表示している。これより、実質的な計算時間の軽減のために細分化禁止則の導入が薦められる。

6. おわりに

本論文では、要素が先行順位をもつシステムの系列性を保持した群配置問題について検討している。この問題は無閉路有向グラフの系列分割問題として定式化することができ、切断の鎖や、切断決定子列の概念を用いて問題を表現することが可能である。一般的な系列分割問題に対する最適化算法を構成するにあたって、全ての切断決定子とそれらの関係を表現する必要がある。この関係の最少データ表現として既約グラフを用いることができ、一方、既約グラフは各切断決定子の展開を基本操作によって行う横型展開法により、効率よく形成可能であることを明らかにした。また、基本操作による節点の展開は本来集合計算によって実行されるが、集合の特性関数にあたる関数 $h_{\mu}() \mid \partial A$ を導入することにより計算過程を簡素化することが可能である。

要素間に先行順位をもつシステムにおいて並列構造が認められないとき、本来、本問題は指数計算時間を要するが、これに対して並列構造が顕著に認められるシステム構造に対しては十分実用に耐える算法が実現できることを示した。

このような要素が先行順位関係をもつシステムの群配置問題はコスト関数と制約条件をそれぞれの問題の特性に合

わせることによって、多様な問題へ適応させることが可能である。

(平成7年5月15日受付、同7年11月6日再受付)

文献

- (1) Betts, J. and Mahmoud, K.I.: "A Method for Assembly Line Balancing", *Engineering Costs and Production Economics*, Vol.18, pp.55-64(1989).
- (2) Kernighan, B.W.: "Optimal Sequential Partitions of Graphs", *J.ACM*, Vol.18, No.1, pp.34-40(1971).
- (3) Morin, T.L. and Marsten, R.E.: "Branch-and-Bound Strategies for Dynamic Programming", *Operations Research*, Vol.24, No.4, pp.611-627(1976).
- (4) Salvendy, M.E.: "The Assembly Line Balancing Problem", *The Journal of Industrial Engineering*, May-June, pp.18-25(1955).
- (5) 秋山、西関: 「グラフとダイグラフの理論」, 共立出版(1981).
- (6) 茨木 俊秀: 「組合せ最適化一分枝限定法を中心として」, 産業図書(1983).
- (7) 加地 太一, 大内 東: 「最適系列分割問題に対する効率的な分枝限定法の構築と諸特性解析」, *情報処理学会論文誌*, Vol.35, No.3, pp.364-372(1994).