論　　文

要素間に先行順位をもつシステム要素の配置問題に対する厳密解法と近似解法の提案

加　地　太　一*

先行順位を持ついくつかの生産プロセスを各ステーション間の移動でのみ必要とする輸送コストの総和が最小になるように，ステーションごとに利用可能な資源量の範囲内で配置する問題が考えられる．この問題に対して系列グラフ分割問題を利用しモデル化を行い，合わせて厳密解法と近似解法を提案する．本来，本問題は指数的計算時間を要するが，要素間に先行順序をもつシステムにおいて並列構造が認められるシステム構造に対しては十分実用に耐える厳密解法が実現可能であることを示す．さらに，ランダムなグラフにも対応できるよう，複合的近傍構造を導入したTabu Search法を用い効果的な解を導出する近似解法を構成し，その計算時間もグラフの構造などに影響されることなく，ほぼ頂点数に線形に増加する傾向が示されることを述べる．

Exact and Approximation Approach of Assignment Problem for System with Precedence Relationships among Elements

Taichi KAJI*

We think about the problem to assign the elements to an ordered sequence of stations such that the precedence relationships are satisfied. This problem can be shown as a problem for sequential partitions of the nodes of a directed acyclic graph into subsets. We describe an efficient exact algorithm that can be used to reduce computational requirements and, possibly storage. And, we presents approximation algorithm using the tabu search approach with effective neighbourhood structure, that is effective in obtaining near-optimal solutions to this problem. The running time of the procedure is proportional to the number of nodes in the graph.

1. はじめに

本論文の対象として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，その具体的な事例として以下の問題を考察する．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接続プロセスに渡す．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を挙げる．

この問題に対して本論文では無閉路有向グラフを先行順位を無視することなく分割するときに生ずるカット・エッジのコストの総和を最小化する問題としてモデル化を行う．今後，この問題を“総カット値を最小化する系列分割問題”と呼ぶこととする．さらに，動的計画法を適用し，無閉路有向グラフの構造に並列構造が認められるとき，多項式オーダーの実用時間内で計算可能な有効な厳密解法を提案する．しかし，無閉路有向グラフがランダムな構造を有するとき指数的オーダーとなり実質的に計算が困難となる．そこで，ランダムなグラフにも対応できるよう，昨今，種々の問題で優れた成果を示しているTabu Search法〔2〕，〔3〕，〔10〕，〔11〕を用い，本問題に有効な近似解法の構成を示す．ただし，本問題の場合，系列性を保存する多分割問題であり，また，解の成分集合の個数，各成分集合の要素数が不定であり，次の解への最良移動の決定は最良分割数の決定と合わせて複雑な組合せ探索を必要とするため，効果的な近傍構造が構成しにくい．これに対して複合的な解の近傍移動により解に大きな変化をほどこし近似解法の性能を引き出す．

* 小樽商科大学(Otaru University of Commerce)
受付：1996年5月8日，再受付(1回)
受理：1996年10月22日

2. 問題のモデル化と定式化

先行順位をもつシステムで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気閉路有向グラフ $D(V, E)$ 上での先行順位を保持した分割と考えられ，以後，この D を対象として問題を考察する．ここで，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，任意の頂点 $v, u \in V$ に対して v から u への有向路が存在するならば，順序関係 $v \preceq u$ が成立するという．順序関係 $\preceq$ は D が無閉路であることから反対称性をみたし，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 $v \prec u$ で表す．この問題は要素間の先行順位関係を無視せずに頂点集合 V を互いに素で網羅的な部分集合の族 $\{V_1, V_2, \cdots, V_k\}$ に分割し，各部分集合を先行順位関係にもとづいて先行順に並べることができる．これを系列分割と呼び，この定義をより正確に以下にまとめる．

定義 1. 空でない部分集合 $A \subset V$ から誘導された〔6〕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．

A とその補集合 A^c に関して，任意の $x \in A^c$, $y \in A$ が $\preceq$ について比較可能ならば常に $x \prec y$ が成立するとき，(A^c, A) を A によって定まる V の切断，A をその切断の上組，A^c を下組という．また，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．さらに，2 つの切断 $(A^c, A), (B^c, B)$ に対して $A \supseteq B$ が成立するとき，(A^c, A) は (B^c, B) の前にあるといい，$(A^c, A) \prec = (B^c, B)$ で表す．

$X|Y$ は直観的には“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”ことを表している．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 2. $D(V, E)$ の頂点集合 V の分割 $\{V_1, V_2, \cdots, V_k\}$ 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 $V_1|V_2|\cdots|V_k$ 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．また，各 V_i をブロックと呼ぶ．

この定義より，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．各成分集合 V_i に上組 $A_i = V_i \cup V_{i+1} \cup \cdots \cup V_6$，下組 $A_i{}^c = V_1 \cup V_2 \cup \cdots \cup V_{i-1}$ からなる切断 $\mu_i = (A_i{}^c, A_i)$ を対応づけ，切断の順序列 $\mu_1, \mu_2, \cdots, \mu_6$ を作ると隣り合う切断の上組 A_i, A_{i+1} から，ブロック $V_i = A_i - A_{i+1}$ となり，系列分割 $V_1|V_2|\cdots|V_6$ が得られる．同じ関係式は，切断の系列が $\mu_1 \prec = \mu_2 \prec = \cdots \prec = \mu_6$ を満たす切断の鎖に対して常に成立する．

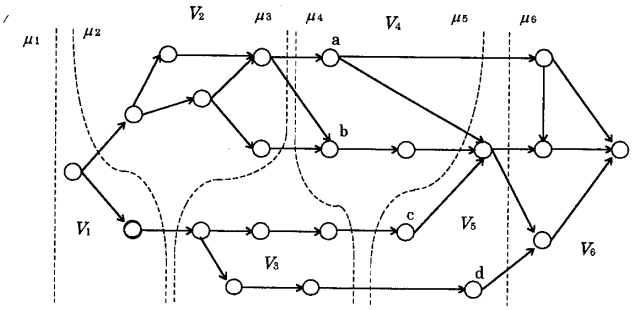


図 1 無閉路有向グラフの系列分割

系列分割上で，各成分に課せられた制約条件のもとで，最良の評価値をもつ分割を求める問題を系列分割問題と呼ぶ．本論文の問題は“総カット値を最小値する系列分割問題”を扱うことにより問題の解決が図られる．以下にその定式化について述べる．

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 $D(V, E)$ として与えられ，$D(V, E)$ のすべての頂点 $v \in V$ には重み $w(v)$ が，各有向辺 $(u, v) \in E$ にはコスト $c(u, v)$ が付与されている．これらの値は，すべての $u, v \in V$ について，条件 $0 < w(v) \leq B$，および $c(u, v) \geq 0$ を満たす整数であり，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．総カット値を最小化する系列分割問題では，容量制約 $|V_i| = \sum_{v \in V_i} w(v) \leq B$ のもとで，切断される辺のコストの総和を最小にする分割を求める問題として定式化できる．以後，この問題を単に最適系列分割問題と呼ぶ．

3. 厳密解法の提案

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が重複しない縮約した既約グラフを導出する．この上で基本的な動的計画法による算法について論じる．

まず，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$\mu=(A^c, A)$とするとき，上組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．**図1**における切断μ_4に対応する切断決定子は頂点集合$\{a, b, c, d\}$である．切断は切断決定子により一意に表現可能である．系列分割の各部分集合$V_i=A_i-A_{i+1}$に対応する切断決定子を用いて$V_i=[\gamma_i, \gamma_{i+1})$と表現する．さらに，切断$\mu=(A^c, A)$に対応する切断決定子を$\gamma$とするとき，下組$A^c$の頂点数を$\mu$または$\gamma$のレベルと呼ぶ．前後関係$\prec=$について$\mu$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，

$$A'=A-\{v\} \quad (1)$$

と置けば，

$$\mu'=((A')^c, A') \quad (2)$$

となる．この操作をvをピボットする切断μに対する基本操作という．この処理はvからの出次数に線形な計算負担で計算可能である．

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$\mu_1, \mu_2, \cdots, \mu_r$が作り出される．切断$(\emptyset, V)$を根とし，基本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．例として，**図2**のエッジに重みを持つ無閉路有向グラフに対して，基本操作によって生成される木Tを**図3**に示す．このとき**図2**の頂点番号は各要素プロセス，エッジに付与された数は各ステーション間の輸送コストにあたり，**図3**の各頂点は切断決定子に対応する．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．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．横型展

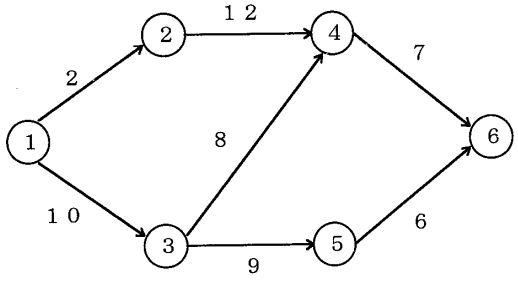


図2　無閉路有向グラフ

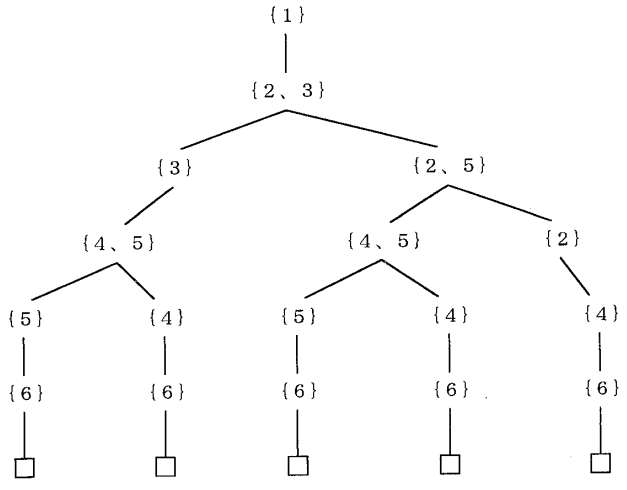


図3　すべての切断決定子を生成する多分岐平衡木

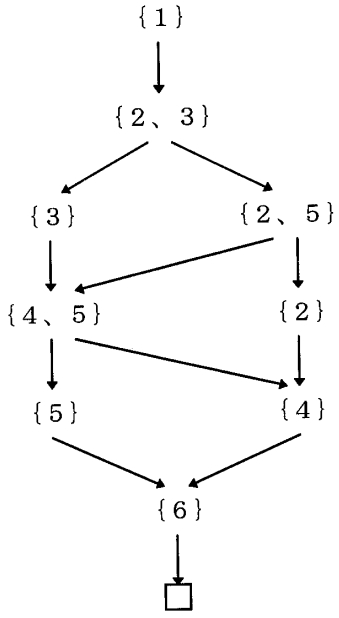


図4　すべての切断決定子を含む既約グラフ

```
1: OpenList:=γs;
2: While (OpenList≠φ) begin
3:    "OpenListの先頭要素γtを取り出す";
4:       for (α∈γt) begin
5:          "αをピボットとしてγtからγgを生成する";
6:          "γgからγtへのポインタを確保する";
7:          if (not(γg∈ OpenList))
8:             "γg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

```
1: f(γs) = 0;
2: while (Ω ≠ φ) begin
3:    "Ωの先頭要素γiを取り出す";
4:    f(γi) = min_γj {f(γj)+C(γj, γi)};
5:    "γiから最小値を与えるγjへのポインターを
                                    確保する";
6: end;
```

図6　動的計画法による算法

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_tからγ_gを生成する”による計算負担の総量は無閉路有向グラフのエッジ数に比例する．また，第7行の“$\gamma_g \in$ OpenList を判断する”は，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し，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．次に，切断決定子の列$\{\gamma_s, \cdots, \gamma_i\}$を系列分割問題の部分解と考え，最後の要素が$\gamma_i$であるようなすべての可能部分解$\{\gamma_s, \cdots, \gamma_i\}$の集合を$\Xi(\gamma_i)$で表す．部分解$\{\gamma_s, \cdots, \gamma_j, \cdots, \gamma_i\}$が$\Xi(\gamma_i)$の中で最小コストを実現するならば，部分解$\{\gamma_s, \cdots, \gamma_j\}$が同時に$\Xi(\gamma_j)$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．

これより，この既約化グラフに対して図6の動的計画法による算法にしたが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$f(\gamma_i)$は$\Xi(\gamma_i)$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$\gamma_j(\prec=\gamma_i)$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_iから根γ_sに至るすべての経路上で$|[(\gamma_j, \gamma_i)| \leq B$により制約された範囲内で計算する．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まり，計算過程で定めたポインター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，コストの計算は以下の漸化式を用い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，コスト式は次式のように書け，その計算の総量はエッジ数に比例する．

$$C(\gamma_i, \gamma_m) = C(\gamma_i, \gamma_j) + \sum_{t \in [\gamma_m, \gamma_{k+1})} c(v, t) - \sum_{s \in [\gamma_i, \gamma_j)} c(s, v) \quad (3)$$

本問題ではその特徴により，動的計画法の各状態での部分解の処理数はブロックサイズBに依存し，組織化して記憶し高速に読み出しを可能にすることができる．また，分枝限定法で横形探索法を用い，優越関係による限定操作を用いると動的計画法の手順と等価〔7〕となり最適性の原理を利用できることはすでに知られているが，プログラム的単純さおよび簡便さなどにより動的計画法を採用した．

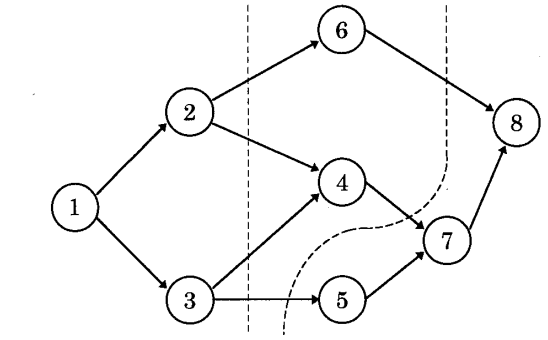
本算法の計算量は切断決定子の生成数に大きく依存し，図6の4行目の計算過程のある段における最小値探索の数はブロックサイズBのみに依存した定数以下となることを考慮すると，無閉路有向グラフがm並列構造を持つ場合，切断決定子数の組合わせがn^mに比例することから$O(n^m)$と考えられる．

4. 近似解法の提案

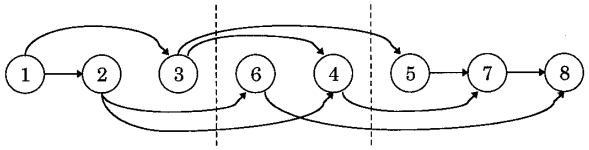
上記の厳密解法ではグラフに並列構造が認められるとき有効であるが，ランダムグラフに対しては実用的な時間内での計算は困難である．本章ではランダムな無閉路有向グラフに対応できるよう，近年，いくつかの成果を示している Tabu Search 法〔11〕の考えを取り入れた近似解法の構成を試みる．Tabu Search 法は Fred Glover〔2〕，〔3〕によって提案された局所探索法の変形である．その大まかな戦略は探索過程で以前に探索した解に再び戻る解のサイクリングをタブーリストを設けることによって禁止する処置をとることである．Tabu Search 法の構造上，解の近傍の構成法，およびタブーリストの要素，タブーリストの構築等が，この算法にとって重要な役割〔8〕，〔10〕を担い，大きな影響を及ぼす．

本問題はその特徴として，解の成分集合の個数，各成分集合の要素数がともに不定であり，その個数を決定しなければならない．これに対して，頂点移動を組織的かつ連続的に行い，さらに局所的な最適化を取り入れた複合的近傍移動を導入することにより解決をはかる．

本近似解法の場合，解xから近傍解x'への移動が主体となるため，近傍解の生成を容易にするデータ構造が必要となる．無閉路有向グラフの系列分割を表す解xは，その無閉路有向グラフをトポロジカルソートすることによって得られる一つの一列化グラフとその頂点列に対して分割点(ブレイク・ポイント〔4〕)を用いて一意に表される．たとえば，図7の(a)を表す解は(b)で示す一列化グラフとブレイク・ポイント列$\{1, 6, 5\}$によって一意に表される．また，一列化された頂点列上で，連続的に並んだ頂点部分列$v_p, v_{p+1},$



(a) 無閉路有向グラフの系列分割



(b) (a)に対する一列化グラフの系列分割

図7 無閉路有向グラフと一列化グラフの関係

…, v_q から誘導される部分グラフは，常に系列を保持する D の部分グラフであり，この部分グラフからなるブロックをブレイク・ポイントによる表示 $[v_p, v_{q+1})$ を用いて表すこととする．任意の解を表す一列化グラフとブレイク・ポイント列はこのとき複数存在するが，最適値への収束は一つの代表例から出発して計算すればよい．以上より，本近似解法の近傍移動で行われる頂点の移動は一列化上の順番を意識するのみで構成でき，また，以下で述べる部分的最適化へ容易に当てはめることができるなどの利点が生じる．

さらに，解の探索の過程で一度探索した解を再び探索するサイクリングの現象を防ぎ，無駄な探索を行わないようにするため，タブーリフトを設ける．本問題の場合，頂点の移動によって近傍を構成するので，その移動した頂点を属性と考える．ここで，$V_i|V_j$ である V_i のある頂点が V_j に移動したとき，その頂点をタブーリスト tabu-to-left に格納する．tabu-to-left に格納されたこの頂点は V_j から V_i への移動が禁止されることとなる．同様に，V_j の任意の頂点が V_i へ移動したとき，その頂点をタブーリスト tabu-to-right に保存し，V_i から V_j の移動を禁断する．この禁断の期間はパラメータ tabulength により設定し，探索が無駄に終わった場合，ふたたび元の探索解に戻ってそこから探索をやり直すという可能性を残す．

次に，近傍の構成を説明する．まず，直接隣接する2つのブロック $V_i=[b_i, b_{i+1})$, $V_{i+1}=[b_{i+1}, b_{i+2})$ に対して V_i に含まれている移動可能な頂点の中で，タブーリスト tabu-to-right 上にあるものを除き，移動によるコスト変化量が最小となる頂点 v_0 を見出し，これを V_{i+1} に移動する．この処理を解 $x=(V_1, V_2, \cdots, V_k)$ に対してある演算子を作用させた結果とみなし，$x'=\overrightarrow{N_i}$(tabu-to-right)$\cdot x$ で表わす．また，V_i から V_{i-1} への左移動に関しても，tabu-to-left の考慮のもとで，最小移動コスト変化量をあたえる頂点 v_0 を V_{i-1} へ移動する同様な演算子 $\overleftarrow{N_i}$(tabu-to-left) を用いて表わす．$\overrightarrow{N_i}$(tabu-to-right)，および $\overleftarrow{N_i}$(tabu-to-left) を作用させた過程で伴うコスト変化量は次式を採用する．

$$\overrightarrow{\delta}(v, V_i, V_{i+1})=\sum_{s\in V_i} c(s, v)-\sum_{t\in V_{i+1}} c(v, t) \quad (4)$$

$$\overleftarrow{\delta}(v, V_i, V_{i-1})=\sum_{t\in V_i} c(v, t)-\sum_{s\in V_{i-1}} c(s, v) \quad (5)$$

次に，解の近似度をできるだけ早く高めるために，ブロックの容量制約を無視して上記の left-to-right 移動による改善を次のように反復して行う．

$$x'=\overrightarrow{N}_{k-1}\text{(tabu-to-right)}\cdot\overrightarrow{N}_{k-2}\text{(tabu-to-right)}\cdots\overrightarrow{N_1}\text{(tabu-to-right)}\cdot x$$

これを多重 left-to-right 移動と呼び，$x'=\overrightarrow{multiN}$(tabu-to-right)$\cdot x$ で表わす．その結果に次の right-to-left 移動の反復による改善を行う．

$$x'=\overleftarrow{N_2}\text{(tabu-to-left)}\cdot\overleftarrow{N_3}\text{(tabu-to-left)}\cdots\overleftarrow{N_k}\text{(tabu-to-left)}\cdot x$$

これを多重 right-to-left 移動と呼び，$x'=\overleftarrow{multiN}$(tabu-to-left)$\cdot x$ で表す．また，多重 left-to-right 移動，多重 right-to-left 移動だけでは，解の容量制約に関する実行可能性が失われるばかりでなく，成分集合の個数，大きさに大きな変化をもたらすことは望めない．そこで，実行可能性を回復し，部分解の個数を決定し，さらに近似度を高めるためにブレイク・ポイントの移動を試みる．多重移動によって得られた解 $x=(V_1, V_2, \cdots, V_k)$ がある一列化グラフとブレイク・ポイントの単調増加列 $\{b_1, b_2, \cdots, b_k\}$ により表されているとする．このとき，ブレイク・ポイントの移動，新たな付加，および削除などによって実行可能性を回復して得られる実行可能解 x' の集合を新たに解 x の近傍解の集合 λ と解釈し，x から λ の中で最も低いコストを示す解 x' への移行を示す関数を $dp(x)$ とする．ここで $dp(x)$ は Kernighan の一列化グラフの最適系列分割問題を解く算法〔4〕，〔9〕がそのまま利用可能である．この算法の計算量は $O(n)$ であり，与えられた一列化グラフのもとで最適なブレイク・ポイント列 $\{b_1', b_2', \cdots, b_{k'}'\}$ を見出す．以上の考えにもとづき，**図8** に Tabu Search 法の算法を示す．

終了判定基準としては，ある反復回数 iteration を繰り返す方法が考えられ，求まる最良解が 2000 前後

```
Procedure Tabu_Search
begin
    "初期解をxとする。"
   x* = x;
   tabulength := positive integer ;
   t :=1;
   while stop-criterion <> yes do begin
      x' := multiN(tabu - to - right) · x ;
      L_t := "解移動(x,x')により移動した頂点の集合" ;
      tabu-to-left := tabu-to-left ∪ L_t;
      x'' := multiN(tabu - to - left) · x' ;
      R_t := "解移動(x',x'')により移動した頂点の集合" ;
      tabu-to-right := tabu-to-right ∪ R_t ;
      x''' := dp(x'') ;
      if f(x''') < f(x*) then x* := x''' ;
      x := x''' ;
      tabu-to-right := tabu-to-right - (R_{t-tabulength} - ∪_{i=t-tabulength+1}^{t} R_i);
      tabu-to-left := tabu-to-left - (L_{t-tabulength} - ∪_{i=t-tabulength+1}^{t} L_i);
      t := t + 1 ;
   end;
   best solution := x*
end;
```

図8　Tabu Search 法のアルゴリズム

の反復回数以内で決定される場合が多いことから，iteration を 2000 前後とする(並列グラフの場合，1000 以内で求まることが多いので 1000 前後とする)．さらに，精度を考慮した場合，上記の値の 2 倍程度に設定する．tabulength に対しては 0 から 100 まで変化させ調べた結果，その適正値はブロックサイズに依存し，ブロックサイズの 1/10 から 1/5 の値に設定することが望ましい．

5. 数 値 実 験

この問題における並列グラフに対して，厳密解が実用時間内で算出可能であることを示し，あわせて，近似解法との比較により，近似解法の解の近似度とその特徴について論じる．計算環境は CPU が Intel 486 DX 2 および OS として Linux を採用している．まず，**表 1** はブロックサイズ，エッジコスト，中間エッジ(並列線路間の接続辺)を変化させた場合の頂点 200 の 2 並列グラフに対するコストの比較であり，パラメータ iteration は 2000 とする．**表 1** よりエッジコストがすべて 1 である fix の場合，近似解法の結果は最適解にほとんど近い値が最適解そのものとなり相対誤差〔8〕が 5 %以内を維持する．また，エッジコストがすべて同一な値に対しての実験においても同様な傾向にあることが示されている．しかし，エッジコストの値が幅広く分散する random(1 から 10) の場合，近似度は前記に比べてやや悪くなり相対誤差 5 %

表 1　2 並列グラフに対する近似解と厳密解

厳密解法 (コスト値)	近似解法 (コスト値)	相対誤差(%)	エッジコスト	ブロック サイズ	中間エッジ数
20	20	0	fix	10	0
113	118	4.4	fix	10	100
5	5	0	fix	40	0
80	81	1.2	fix	40	100
46	49	6.5	random	10	0
547	598	9.3	random	10	100
7	7	0	random	40	0
396	409	3.2	random	40	100

表 2　頂点数による影響と計算時間

頂点数	50	100	150	200	250	300
厳密解法 (コスト値)	28	58	87	113	145	175
厳密解法の時間(ms)	1129	7759	17078	35488	61457	94255
近似解法 (コスト値)	28	58	90	118	150	181
近似解法の時間(ms)	1483	3098	4900	6644	8469	9968
相対誤差(%)	0	0	3.4	4.4	3.4	3.4

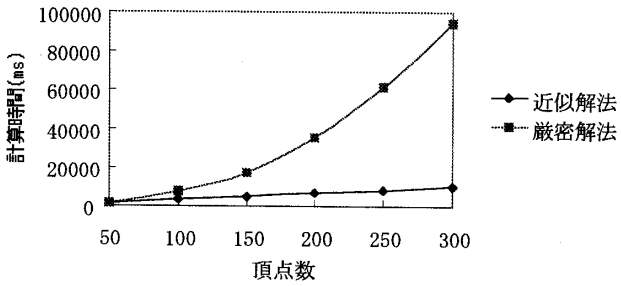


図 9　頂点数と計算時間

以上となる場合がある．これはカットされる辺のコスト値が大きな範囲を持つため，カットの選びかたにより，解の値に大きな影響を及ぼす結果となるためである．しかし，ブロックサイズが大であると，そのカットの選びかたに多様性があるので最良な値を選びやすく，上記の影響は現れにくい．3 並列等の同様な実験でも同じ傾向が示された．しかし，エッジコストの分散による影響は強くでる傾向がある．

次に，中間エッジが頂点数の 50%となる 2 並列グラフに対して，頂点数に対する計算時間とコストの変化を**表 2** で示し，特に時間の増加傾向を**図 9** で表す．このときのパラメータ iteration は 1000 とする．この実験より，近似解法では最適解かほぼそれに近いコストの値が求まり，同様に相対誤差 5%以内となることが示された．さらに，厳密解法は 3.で述べた通り，2 並列の場合，ほぼ n^2 傾向の計算時間が認められ，本来，指数関数的に計算時間を要する本問題において，並列構造が認められるとき実用時間内で計算可能なことが示された．また，近似解法ではほぼ線形の計算時間で求まることが示され，グラフの構造およびその他の戦略に依存することなく頂点数にほぼ線形な計算が可能である．

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モデル化することができ，その問題の表現法と定式化について提案した．本来，本問題は，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，指数的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厳密解法が実現できることを述べた．さらに，ランダムなグラフにも対応できるよう，複合的近傍構造を導入したTabu Search法を用い近似解法を提案した．この解法では，グラフのエッジコストの値がほぼ同一である値を示す場合，相対誤差5％以内の解(最適解が求まる場合もある)が求まる．また，その計算時間もグラフの構造などに影響されることなく，ほぼ頂点数に線形に増加する傾向が示され，本算法は今回の問題に対して有効な近似解法を与えるものと考えられる．

さらに，ここで示す先行順位関係をもつシステムの配置問題はコスト関数と制約条件をそれぞれ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．今後の問題として目的関数，制約条件を変化させることによって構成できるライン・バランシング問題〔1〕，〔5〕などへの適用および効果について検討を試みたい．

参考文献

〔1〕 Betts, J. and Mahmoud, K. I.: "A Method for Assembly Line Balancing," *Engineering Costs and Production Economics*, pp. 55-64, Vol. 18, (1989)

〔2〕 Glover, F.: "Tabu Search Part I," *ORSA J. C.*, pp. 190-206, Vol. 1, No. 3, (1989)

〔3〕 Glover, F.: "Tabu Search Part II," *ORSA J. C.*, pp. 4-32, Vol. 2, No. 1, (1990)

〔4〕 Kernighan, B. W.: "Optimal Sequential Partitions of Graphs," *J. ACM*, pp. 34-40, Vol. 18, No. 1, (1971)

〔5〕 Salveson, M. E.: "The Assembly Line Balancing Problem," *The Journal of Industrial Engineering*, pp. 18-25, May-June, (1955)

〔6〕 秋山　仁, 西関隆夫:「グラフとダイグラフの理論」, 共立出版, (1981)

〔7〕 茨木俊秀:「組合せ最適化」, 産業図書, (1983)

〔8〕 茨木俊秀:「離散最適化法とアルゴリズム(岩波講座応用数学)」, 岩波書店, (1993)

〔9〕 加地太一, 大内　東: "最適系列分割問題に対する効率的分枝限定法の構築と諸特性解析," 情報処理学会論文誌, pp. 364-372, Vol. 35, No. 3, (1994)

〔10〕 久保幹雄: "Local Search から Simulated Annealing, Tabu Search へ", モダンヒューリスティックス(平成6年度第2回ORセミナー), pp. 1-32, (1994)

〔11〕 藤沢克樹, 久保幹雄, 森戸　晋: "Tabu Search のグラフ分割問題への適用と実験的解析", 電学論c, pp. 430-437, Vol. 114, No. 4, (1994)