Tabu Search を用いた
無閉路有向グラフ系列分割問題の近似解法

加 地 太 一

1．はじめに

先行順位のある要素をその先行順位を保持したまま，いくつかのステーションに配置する問題を考える。このとき，配置に対するある制約条件のもとで，この配置にともなって決定される最良な評価値を求めるものとする。この種の問題としては，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作業ステーションに割り当てるライン・バランシングの問題などがある[1),6)]。さらに，制約条件，コスト関数を変えることによって，各種の問題に適応できる。

今回は特に次のような問題を取り上げいくつかの検討を試みる。プロセスと考えられる要素がある資源量を要求し，1つのステーションにおいて使用できる資源量が定まっている場合，ステーション間のプロセスの移動において，あるコスト量がかかるものとする。ただし，同一のステーション間での移動においてはコストが無視できるものとする。このとき，各プロセスをステーションにコストの総和が最小になるよう先行順位の制限を無視せずに配置する。この問題はライン・バランシング問題のコスト関数を変えた問題でもある。また，応用例としては前後関係のある一般化したデータのパイプラインの問題などが考えられる。たとえば，メモリと2次記憶間のプロセスの移動コストを辺コストとし，ディスク時間を最小化する問題，およびパートにおける年次計画などでの利用が計られる。

これらの問題は基本的に無閉路有向グラフの頂点を先行順位の関係を無視せ

ずに分割する問題として表すことが可能であり，この無閉路有向グラフの系列分割問題に置き換え，本問題を考察する。前回[8]，無閉路有向グラフの系列分割問題に対しては探索空間を宿約し動的計画法を用いた算法により，m 並列に近い構造を持つ頂点数 n のグラフに対して計算量 $O(n^m)$ で求められることを示した。しかし，この算法ではランダムなグラフに対しては指数的オーダーの計算時間を必要とし，実質的な時間内では計算が不可能である。したがって，我々は本問題に対し効果的な時間内で計算可能な近似解法の適用を試みる。その近似解法として昨今，種々の問題で優れた成果を示している Tabu Search[3),4),5),10),11),12)] を採用することとする。Tabu Search は Fred Glover 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。

本稿ではまず，無閉路有向グラフの系列分割問題についての諸定義を定め，Tabu Search の戦略について述べる。さらに本問題を Tabu Search へ適用する場合の近傍等の基本的考えを示す。次に具体的な算法の実現と工夫点について論ずる。最後に特性評価および挙動解析を行うための数値実験の結果とその考察について明らかにする。

2．諸定義

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 上の順序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。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 から導かれる半順序集合を（$V, \preceq$）で表す。

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

定義 2．　Vの部分集合Aが，$A^c \times A$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$x \prec y$が成立するとき，(A^c, A)をAによって定まるVの切断という。そして，Aをこの切断の上組，A^cを下組という。

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。また真に前にあることを記号（$A^c, A) \prec\neq (B^c, B$）で示す。

定義 3．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 4．　$D(V, E)$の頂点集合Vの分割が$V_1|V_2|\cdot\cdot\cdot|V_k$を満たすように順序づけられるとき，この分割を$D(V, E)$の系列分割という。

このとき，各分割成分V_iは系列保存の性質をもつ。図1は無閉路有向グラフの系列分割の一例である。

本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$D(V, E)$として与えられており，$D(V, E)$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$0 < w(v) \leqq B$，$c(u, v) \geqq 0$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。このとき無閉路有向グラフの系列分割において，$|V_i| = \sum_{v \in V_i} w(v) \leq B$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。

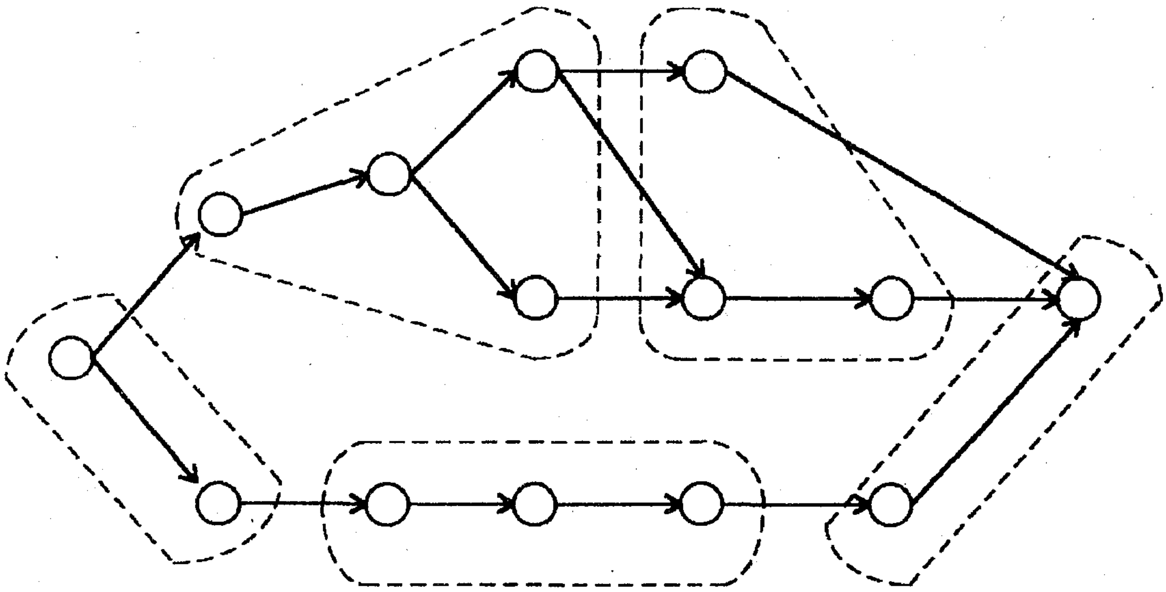

図1　無閉路有向グラフの系列分割

3．Tabu Search の基本的構造

Tabu Search は人間の記憶の構造を利用した解への探索方法であり，解のサイクリングを避けるためにタブーリストという記憶域を設け，一部の探索を禁止する処置をとる。すなわち，Tabu Search は最近の s 個の探索解をタブーリストに記憶しておき，それらを候補から除く，あるいは最近の s 個の探索解で生じた変数 x_i の変化方向をタブーリストに記憶し，これらの変数の逆方向への変化を禁止する処置をとる。ここで，タブーリストを α とすると，Tabu Search では現在の解 x^{now} の近傍集合 $N(x^{now})$ から α を除き，その中から最小コストとなる解 x^{next} を選択する。また近傍集合から α を除いた新たな集合を $N(\alpha, x^{now})$ とする。すなわち，$N(\alpha, x^{now})=\{N(x^{now})-\alpha\}$ となる。その移動を行う関数を以下に示す。

$$move(x^{now})=\begin{cases} x^{next}, & if \cos t(x^{next}) \leq \cos t(x)\ for\ all\ x \in N(\alpha,\ x^{now}) \\ \phi, & if\ N(\alpha,\ x^{now})=\phi \end{cases}$$

また α の長さを length とし，以下の算法構成で基本的なTabu Searchを記述する[10),11),12)]。

```
t := 0;
x_0 := initial solution;
α := φ;
length := a positive integer;
while stopping−criterion <> yes do begin
    x_{t+1} := move (x_t);
    α := α ∪ x_t − x_{t−length}
    t := t+1;
end;
```

Tabu Searchは非常に柔軟性の高い枠組みであり，上の算法にさらにいくつかの戦略を付加して構成する。

4．近傍とタブーの基本的考え

Tabu Search の算法を構成するにあたって重要な要素はその問題における解の近傍とタブーの要素の構成である。以下にその基本的な考えを述べる。

無閉路有向グラフ$D(V,\ E)$ の系列分割を $\{V_1,\ V_2,\ \cdots,V_k\}$ とする。V_iによって誘導された$D(V,E)$の部分グラフを$\Delta(V')$とする。このとき，$\Delta(V')$ の有向辺が定めるV'の関係において，極大元および極小元となる頂点$v\in V'$は系列分割の性質を変えることなく，他のあるブロックV_jに移動できる。vが極大元のときは$D(V,\ E)$ でのvからの流出辺が$V_{i+1},\ \cdots,\ V_{j-1}$内の頂点に流入することがなければ，$v$は$V_j$に移動可能である。同様に，$v$が極小元のときは流入辺が$V_{j-1},\ \cdots,\ V_{i-1}$内の頂点から流出しなければ$V_j$に移動可能となる。この$v$の$V_i$から$V_j$への移動を$e\,(v\ ;\ i,j)$ で表わす。また，V_jの直前あるいは直後に新しい空ブロックを作り，前記と同様な条件が満たされれば，そこにvを移動して系列を拡大するができる。このvの空ブロックへの移動を$a\,(v\ ;\ i,j)$ で表す。さらに，$\mathrm{e}\,(v\ ;\ i,j)$，$a\,(v\ ;\ i,j)$ による頂点の移動によってV_iが空になったときには，このV_iを取り除いて系列

を縮小することもできる。

与えられた系列分割 $\{V_1,\ V_2,\ \cdots,\ V_k\}$ に移動 $e(v\ ;\ i,j)$ または，$a(v\ ;\ i,j)$を行うことによって，新しい系列分割が得られる。このようにして，得られた系列分割の集合が系列分割 $\{V_1,\ V_2,\ \cdots,\ V_k\}$ の近傍である。この移動による系列分割の生成にあたって，移動によって空となったブロックは系列から取り外す約束にしておく。

次に，タブーの要素としては解そのものとはせず，通常，解の移動に関係する頂点および辺などを属性として，これをタブーの要素とする。本問題の場合，頂点の移動による近傍が構成されるので，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の任意の頂点が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 に格納されている属性である頂点はV_jからV_iへの移動が禁止されることとなる。同様な形式で，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。

5. Tabu Searchへの実現

本問題において無閉路有向グラフそのものをあつかう場合，分割の表現などに複雑なデータ構造を必要とする。また基本構成に述べるような解の近傍構造を実現するためには複雑な処理が要求される。それゆえに，算法の構成，計算の負担に著しい影響を及ぼす。これに対して我々は無閉路有向グラフを一列化したグラフに変換し，その上で分割を考え，さらに基本構成の近傍を効果的な処理方法で実現する。

5．1　無閉路有向グラフと一列化グラフ

無閉路有向グラフはいつでも一列化(トポロジカルソート)が可能であるが，結果は一意でなく複数の一列化が可能である。本問題はこの一列化グラフとの関係において以下の特徴的な性質がある。

定理.　無閉路有向グラフDの任意の系列分割はDのある一列化グラフの順序を保持した分割（一列化グラフの系列分割）を指定することによって定まる。

証明：　Dの系列分割は正規表現のもとで$V_1|V_2|\cdots|V_k$をみたす。各V_iによって誘導されるVの部分グラフを一列化グラフD_iに書き直し，D_iを添字の順に左から右に一列に並べ，それらをカットセットに属する辺で結びつけると，Dの一列化グラフが得られる。ここで，この一列化による頂点の番号付けと，各D_iの最左端の頂点番号からなる単調増加部分列から一列化の系列分割を作れば，これは与えられた無閉路有向グラフの系列分割と一致する。

以上から本問題においては無閉路有向グラフを一列化グラフに変換する。その一列化グラフ上での分割を考えることによって，データの処理を容易に行うことが可能となる。ここで，一列化されたある頂点列を$\{v_1, v_2, \cdots, v_n\}$とする。このとき，任意の頂点$v_p, v_q$；$(p < q)$をブレイク・ポイント[2)]$b_i$，$b_j$とすることによって部分集合$V_i = [b_i, b_j) = \{v_p, v_{p+1}, \cdots, v_{q-1}\}$を決定でき，ブレイク・ポイントの集合$\{b_1, b_2, \cdots, b_k\}$によって一意的に系列分割$\{V_1, V_2, \cdots, V_k\}$が表現できる。したがって，無閉路有向グラフの先行順位を無視しない頂点列とブレイク・ポイントの集合のデータを管理することによって容易に解の表現が可能となり，これを利用して効果的なTabu Searchを構成する。

5．2　近傍移動の実現

本問題では部分集合の個数，任意の部分集合の要素数は不定である。これが近傍の構造を複雑にしている。これに対して一列化グラフの頂点列とブレイク・ポイントの集合によるデータ表現の観点から，効果的な近傍移動を実現する。このとき，4章の基本的考えで述べた近傍に若干の制限を加える。
基本構成の近傍$e(v\,;i,j)$，$a(v:i,j)$を実現するためには，部分集合間の頂点の移動，あるいはブレイク・ポイントの移動，およびブレイク・ポイントの

付加，削除などによって構成される。本問題を構成するにあたっては上記の頂点の移動とブレイク・ポイントの処理の2つの操作を分け，複合することによって次の解への移動を決定する。

まず，頂点の移動は基本的には2つの部分集合間の左，右移動によって構成する。ただし，最適解の近似を強めるために複数の部分集合間での移動を実現する。直接前後する2つの分割集合$V_i=[b_m,\ b_{m+1})$，$V_j=[b_{m+1},\ b_{m+2})$に対してV_iの移動可能な頂点のうちコスト変化量が最小となる頂点をV_jへ移動する。このときタブーリスト上に含まれている頂点は移動を禁止する。この処理を関数$\overrightarrow{N}(V_i,\ V_j,\ tabu\text{-}to\text{-}right)$とする。ただし移動可能な頂点がなければ，$V_i$，$V_j$はそのままとする。また，$V_j$の移動可能な頂点を$V_i$へ移動する同様な処理を関数$\overleftarrow{N}(V_i,\ V_j,\ tabu\text{-}to\text{-}left)$とする。

近似を強めるために，$\overrightarrow{N}(V_i,\ V_j,\ tabu\text{-}to\text{-}right)$の処理をすべての部分集合に順次適合した以下の処理

$$\overrightarrow{N}(V_1,\ V_2, tabu\text{-}to\text{-}right),\ \overrightarrow{N}(V_2,\ V_3, tabu\text{-}to\text{-}right),\ \cdots,\ \overrightarrow{N}(V_{k-1},\ V_k, tabu\text{-}to\text{-}right)$$

を多次元left−to−right移動と呼び，

関数$\overrightarrow{multiN}(V_1,\ V_2, \cdots, V_k,\ tabu\text{-}to\text{-}right)$で表す。同様な

$\overleftarrow{N}(V_i,\ V_j,\ tabu\text{-}to\text{-}left)$の順次適合した以下の処理

$$\overleftarrow{N}(V_k,\ V_{k-1},\ tabu\text{-}to\text{-}left),\ \overleftarrow{N}(V_{k-1},\ V_{k-2},\ tabu\text{-}to\text{-}left),\ \cdots,\ \overleftarrow{N}(V_2,\ V_1, tabu\text{-}to\text{-}left)$$

を多次元right-to-left移動と呼び，関数

$\overleftarrow{multiN}(V_1,\ V_2, \cdots, V_k,\ tabu\text{-}to\text{-}left)$で表す。

また，多次元left-to-right移動，多次元right-to-left移動だけでは部分集合の個数，大きさの大きな変化は望めない。したがってさらに近似を強めるためにブレイク・ポイントの移動を試みる。ある一列化グラフに対してブレイク・ポイントの単調増加列$\{b_1, b_2,\ \cdots, b_k\}$を決定することによって部分集合$\{V_1, V_2,\ \cdots, V_k\}$である解$x$が示される。このとき，ブレイク・ポイントを移動および付加，削除などによって得られる$\{b'_1, b'_2,\ \cdots, b'_{k'}\}$で決定され

る部分集合$V'_1, V'_2, \cdots, V'_{k'}$である近傍解の集合を$\lambda$と定義し，これによって部分集合の大きさ，個数が変化する。またその変化にともなって頂点の部分集合間の移動も行われる。ただし，$x \in \lambda$でもあるとする。このとき，λの中で最も低いコストを示す解x'への移行を$\vec{\lambda}$と定義する。ここで，$\vec{\lambda}$を求めることはKernighanの一列化グラフの最適系列分割問題を解く算法[2),9)]と等しく$O(n)$で構築できる。

以上，頂点列の並び，およびブレイク・ポイントの位置に観点をおき，3つの近傍移動の処理に分割し構成した。この処理を一反復過程中において，$\overrightarrow{multiN}(V_1, V_2, \cdots, V_k, \textit{tabu-to-right})$, $\overleftarrow{multiN}(V_1, V_2, \cdots, V_k, \textit{tabu-to-left})$と順次行い，再び合成することによって，容易な処理で効果的な近似効果を持つ近傍移動を作成する。

5．3　タブーリストの実現

解移動（x，x'）に対しての逆向きの移動の禁止の属性として，（x，x'）のとき移動する頂点をあてることとし，多次元的left-to-right移動と多次元的right-to-left移動の2つの移動形式に対して2つのタブーリストTabu-to-leftとTabu-to-rightを設けた。タブーリストは待ち行列構造によって構成されるが，プログラム上では頂点集合に対応する一次元配列を用意し，配列の添え字は頂点に対応するものとする。このとき，初期化の段階で配列のすべての要素を0にしておき，移動対象となる頂点の要素にある正の数を入れ，反復毎に1以上の要素の値に対して1減ずることによって，1以上の頂点を禁断対象とする。ある正の数としてはV_iからV_{i+1}（V_{i+1}からV_i）へ，頂点vを移動したときのコストの変化量が負および正の2つの場合に対して異なる値を用意する。すなわち，変化量が改善されたならばTabuLength 1の値，そうでないのならばTabuLength 2の値を与える。これらの値の適正値は事前の数値実験により判定する。

5．4　コスト計算

$\overrightarrow{N}(V_i, V_{i+1}, \mathit{tabu\text{-}to\text{-}right})$ の処理の過程で計算する頂点 $v \in V_i$ の V_i から V_{i+1} への移動に伴うコスト変化量 $\overrightarrow{\delta}(v, V_i, V_{i+1})$，および $\overleftarrow{N}(V_i, V_{i-1}, \mathit{tabu\text{-}to\text{-}left})$ における頂点 $v \in V_i$ に関する同様なコストの変化量 $\overleftarrow{\delta}(v, V_i, V_{i-1})$ は以下の計算式を用いることによって高速に計算することが可能である。

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

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

5．5　終了判定基準

終了判定基準としては，ある反復回数を繰り返す方法，または局所解が更新された段階から，この値が更新されなくなってからの経過時間がパラメーター Stop を越えたときに，算法が終了する方法などが考えられる。今回のタブーリストでは後者を採用しパラメータは2000から3000反復前後とする。

5．6　近似度の改善

現在の段階で求まった近似解に対して，本問題の特徴により，以下の方法を用いさらに近似度を改善することが可能である。

5．6．1　局所最適解からの再出発

本問題で得られる局所最適解の頂点列と最適解の頂点列の構造は部分的な配置が異なるのみで，解の構造としては近い関係にある。すなわち，分割された部分集合の部分的一致が見られる。この性質より，以前に探索した良好な解の情報を積極的に使うことにより，さらに良好な解を探索する戦略が考えられる。この考えを取り入れ，現在得られた局所最適解を初期解とおき，タブーリストをすべて空として再び探索を進めることは有効な結果をもたらす可能性が強い。

5．6．2　2頂点交換法

解xを表す一列化グラフ上の任意の隣接する2頂点v_i，v_jに対して，v_iとv_jの間の切断を考慮したとき得られるコストをcとする。このとき，頂点v_iとv_jが交換可能であり，しかも交換された場合のコストc'が$c'<c$となる場合，v_iとv_jの交換を行う。この操作をすべての交換可能な2頂点に対して行うことによって，コストが改善される可能性を含む新たな一列化グラフが生成される。この一列化グラフに対して，前記の一列化グラフの最適系列分割問題の解法により最適なブレイク・ポイントの位置を求め，解x'を生成する。このx'のコストが改善されているならば，x'を近似解xとし，以上の処理を改善される間繰り返す。この方法を2頂点交換法と呼ぶ。

6．数値実験

本章では上記で構成したTabu Searchの算法に対しての特性評価および挙動解析を行う。まず，Tabu Searchを効果的に動作させるために，最も重要なパラメータは禁止リストの長さである。このパラメータの値は事前の数値実験から考察することとなる。本問題においては2つの禁止リストTabu-to-leftおよびTabu-to-rightを設けているが禁止リストの長さのパラメータは両禁止リストとも共通とする。ただし，任意の頂点の移動に関与するコスト変化量が改善された場合TabuLength 1，そうでない場合TabuLength 2と2つのパラメータを準備する。最初にTabuLength 1の適正値について考える。TabuLength 2の値を0に固定し，値の変化に対してのコストの値の変化を数値実験で確かめ考察する。図2，3，4は頂点数500のランダムグラフに対しての数値実験の一例であり，それぞれブロックサイズが10，30，50の場合を扱う。この実験より，ブロックサイズ10の場合，2前後が適正であり，ブロックサイズ30の場合，7から14であり，ブロックサイズ50の場合，14前後となった。TabuLength 1の適正値はこの結果よりブロックサイズの大きさに影響されると考えられる。すなわち，ブロックサイズの1／5から1／3の大きさ

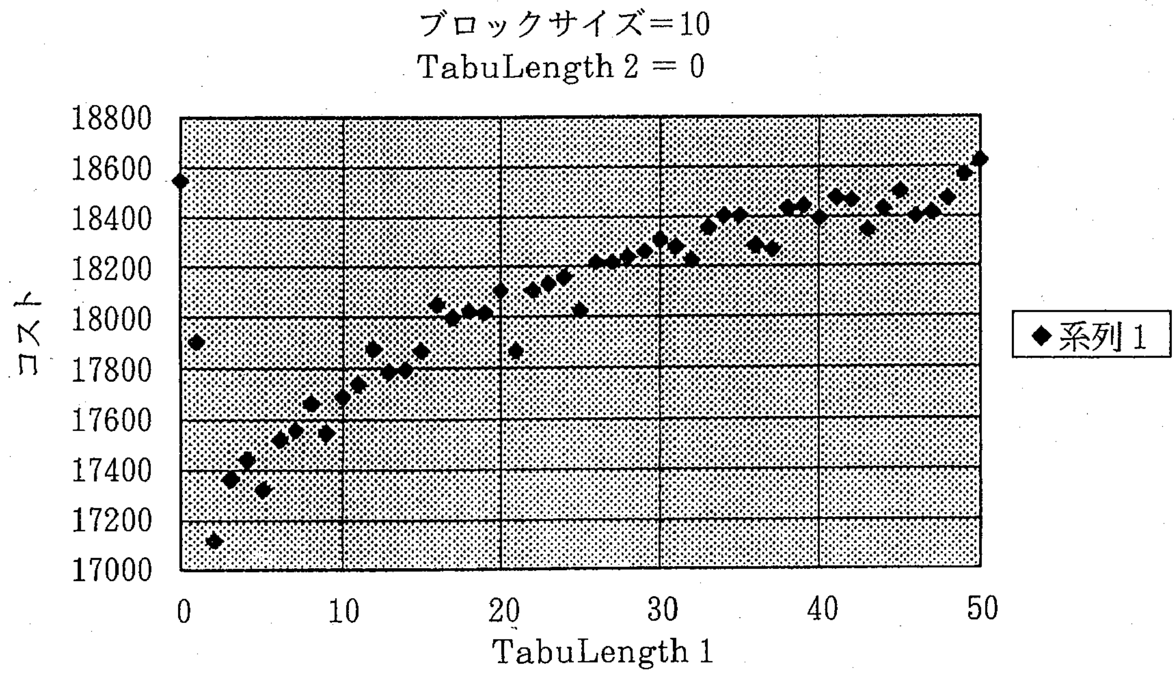

図 2　禁止リストの長さによるコストの変化

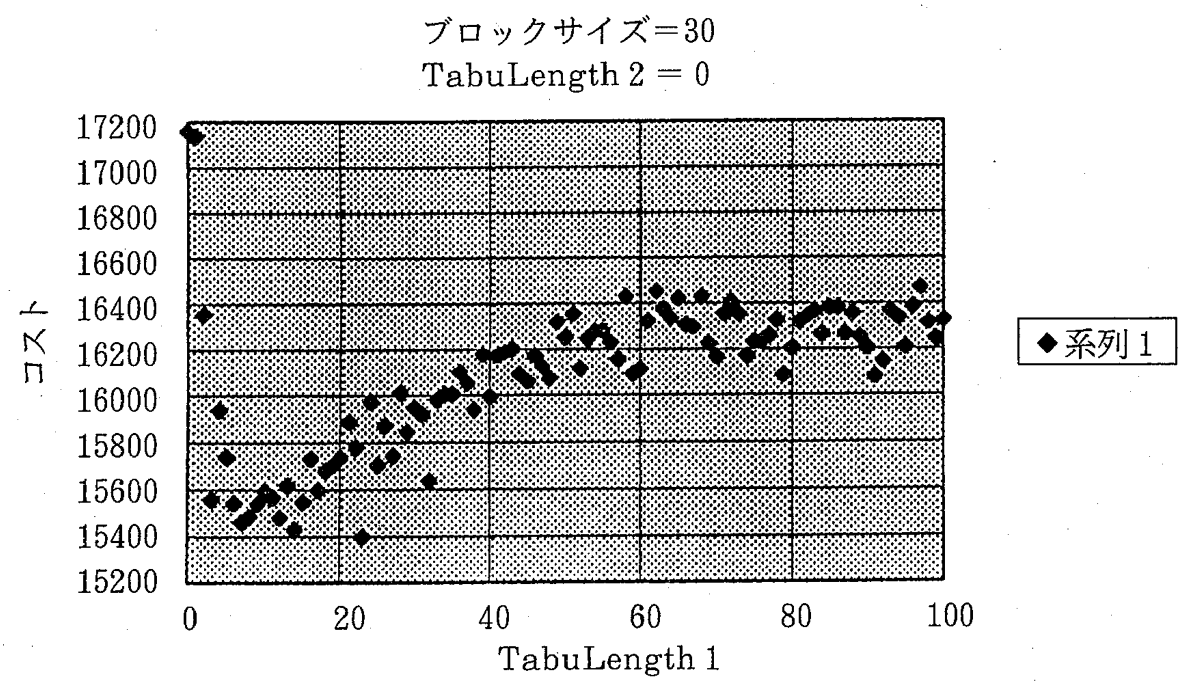

図 3　禁止リストの長さによるコストの変化

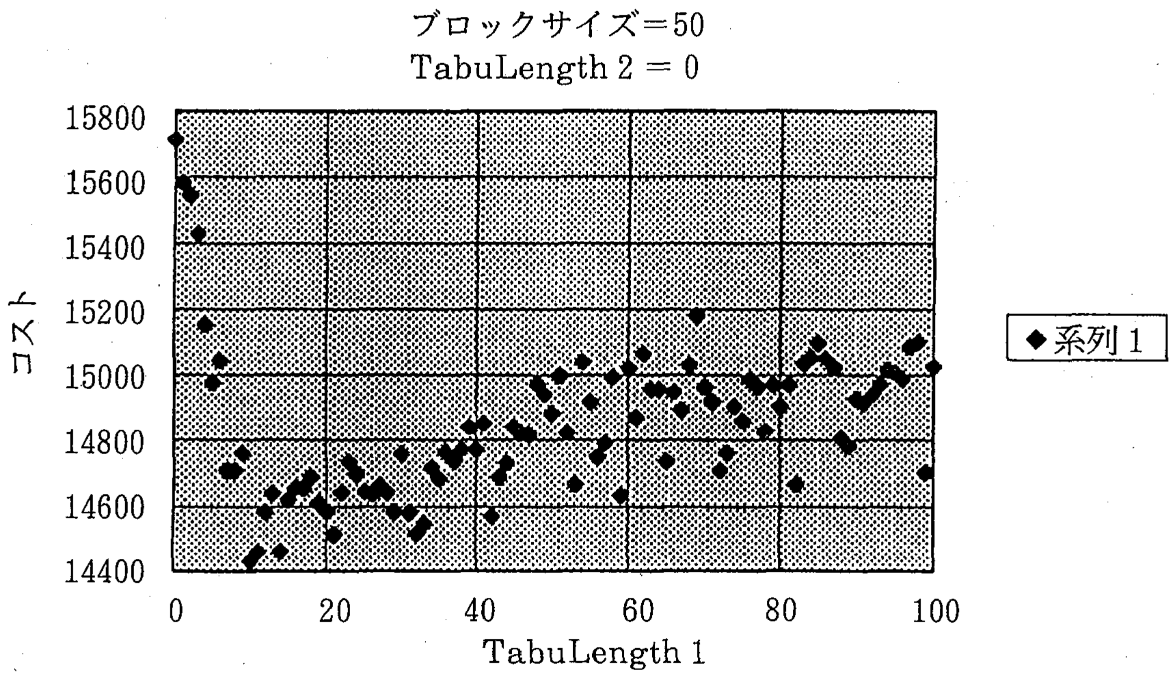

図4　禁止リストの長さによるコストの変化

をTabuLength 1の適正値とすることが望ましい。

次にTabuLength 2の適正値を考察する。図5，6，7，8は図4で使用したグラフで，同様にブロックサイズ50に対するTabuLength 1とコストの変化を表す数値実験であり，それぞれ，TabuLength 2の値が4，6，8，10である。図4に比べると明らかにTabuLength 2によって大きなコストの改善の効果を得ることがわかる。また図5，6，7，8を比較することによって，この場合，TabuLength 2は8前後が適正であり，その他の実験も含めて考察するとTabuLength 1のやや低めの値を適正値とすることが望ましい。

最後に2並列グラフに対しては厳密解を求めることが可能であるので，Tabu Searchとの比較を試み，局所解の近似度について論じる。表1はブロックサイズ，エッジコスト，中間エッジを変化させた場合の頂点数200の2並列グラフのコストの比較である。表1よりエッジコストがすべて1の場合，Tabu Searchの結果は最適解にほとんど近い値か最適解そのものとなる。しかし，エッジコストの値が幅広く変化する場合，近似度は前記に比べて悪くな

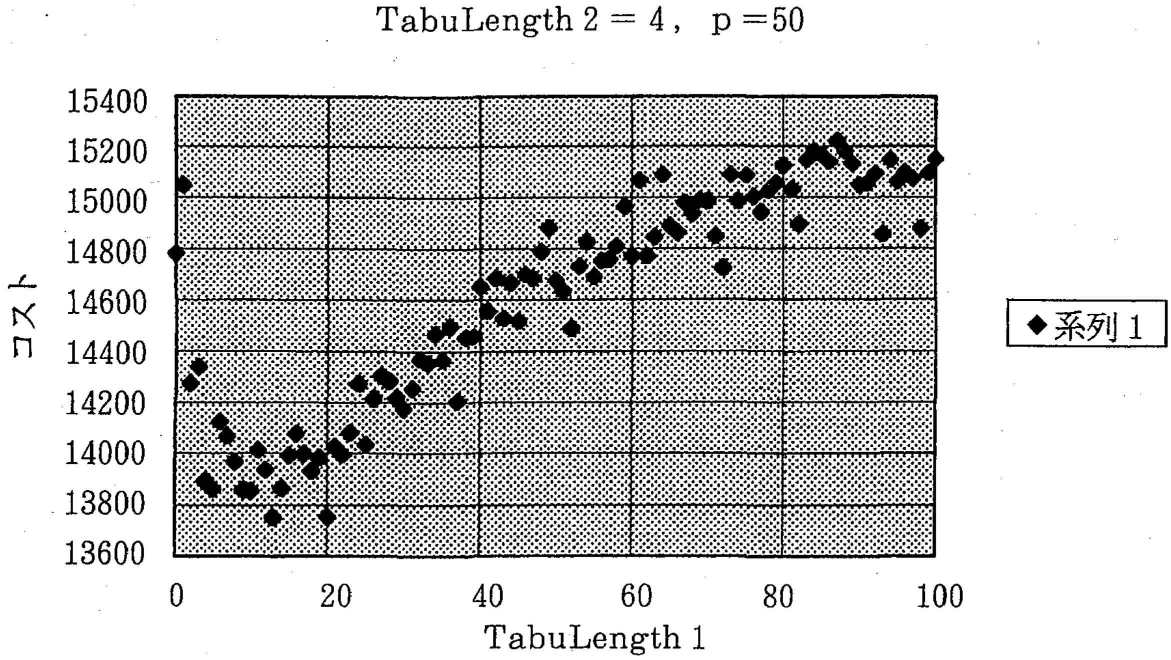

図5　禁止リストの長さによるコストの変化

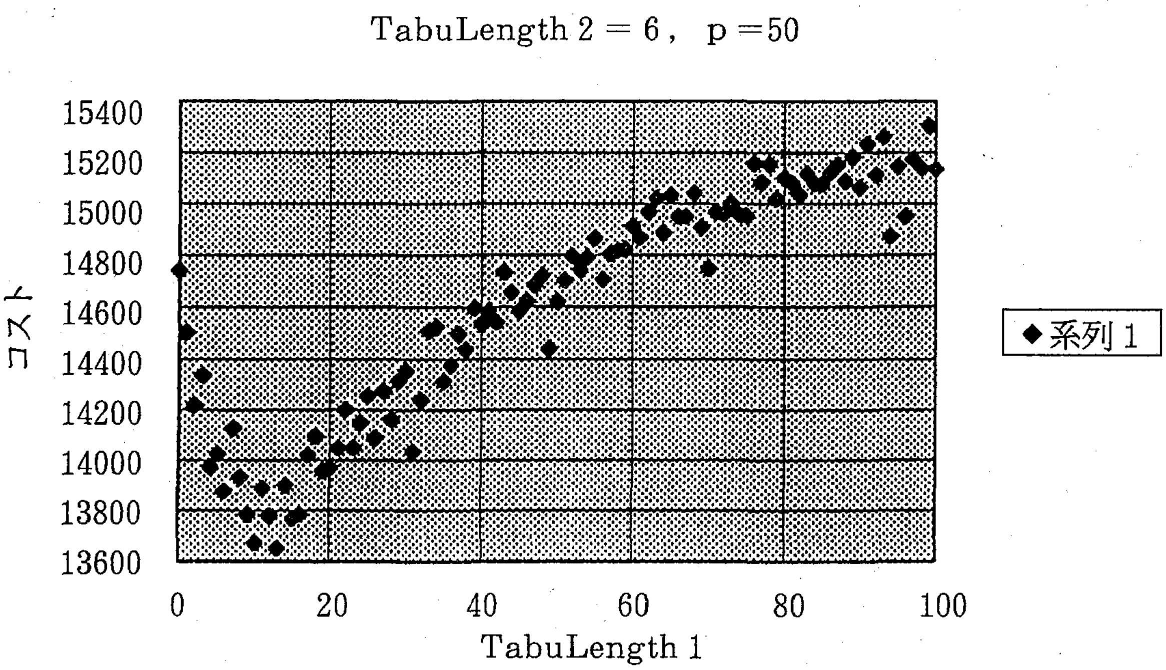

図6　禁止リストの長さによるコストの変化

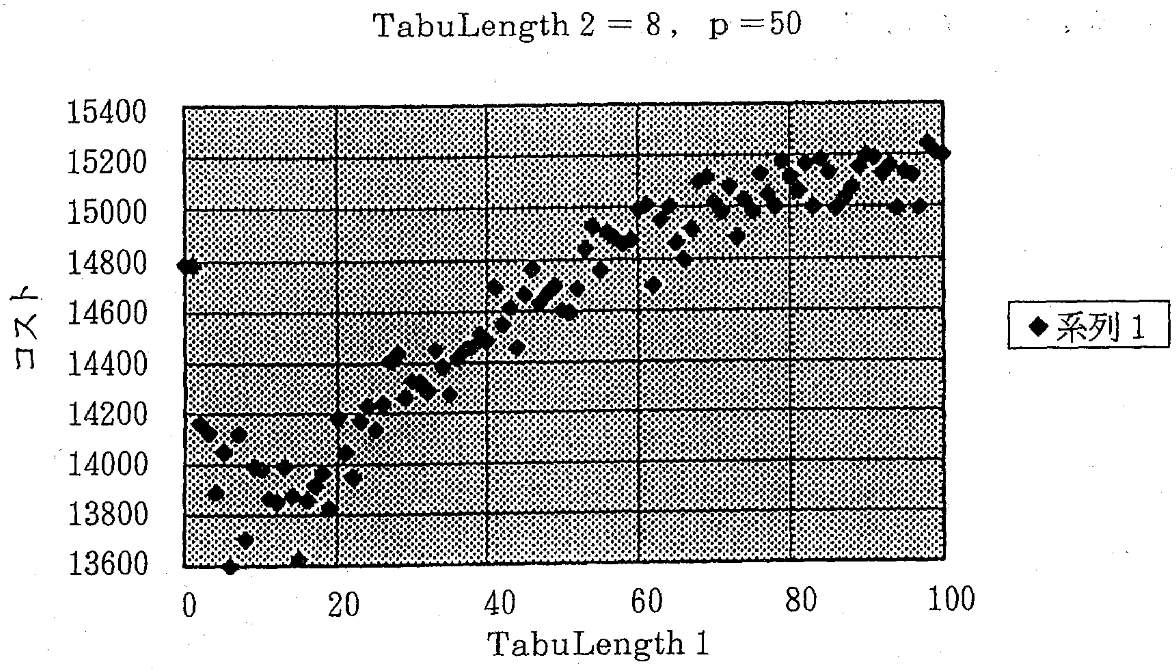

図 7　禁止リストの長さによるコストの変化

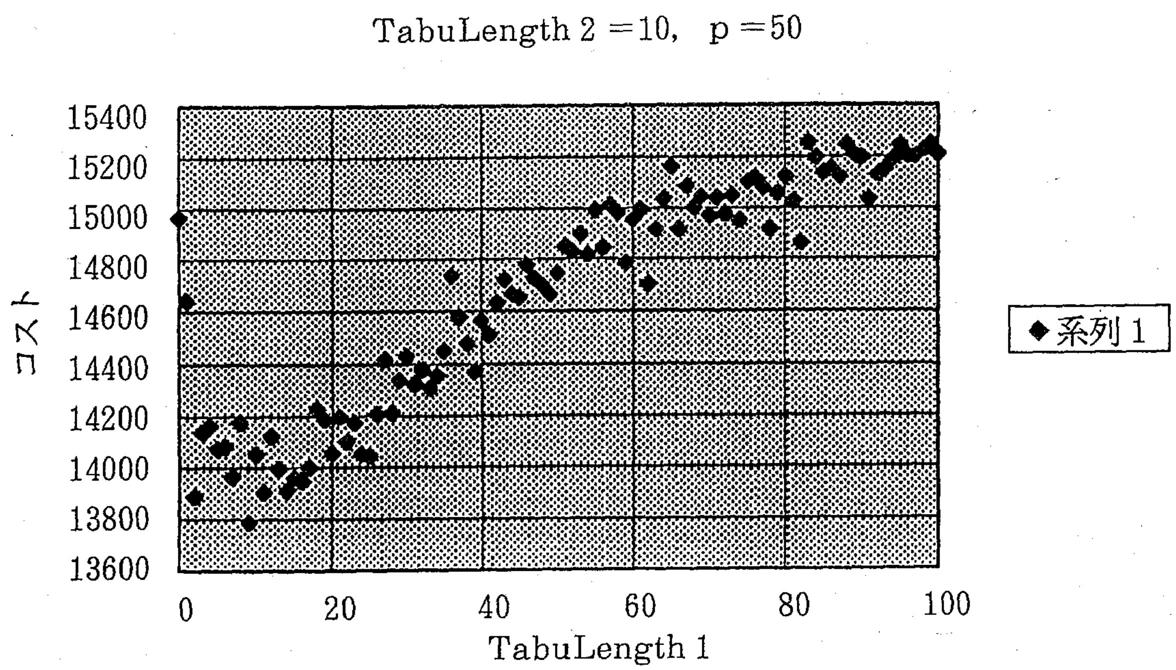

図 8　禁止リストの長さによるコストの変化

表1　2並列グラフに対する厳密解とTabu Searchによるコスト

グ ラ フ	a	b	c	d	e	f	g	h
エッジコスト	1				1から10			
ブロックサイズ	10	10	40	40	10	10	40	40
中間エッジ	0	20	0	20	0	20	0	20
厳密解	20	40	5	19	51	146	6	87
Tabu Search	20	41	5	20	58	168	6	89

る傾向がある。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，上記の影響は現れない。また，中間エッジが入ることによるグラフの複雑性には本解法は影響を受けない。

本算法の計算時間は頂点数200のグラフに対しては一反復6.2msとなり，2000から3000の反復で計算が一般に終了するので，約12から18秒の計算時間を要することとなる。これは近似解を求めるのに十分に効果的な時間内での処理が行えることを意味する。

7．おわりに

本稿では，無閉路有向グラフの系列分割問題に対する効果的なTabu Searchの適用を試みた。無閉路有向グラフの複雑なデータ構造を一列化グラフによる単純なデータ構造へ変換することによって高速処理および記憶容量の削減を実現した。また，効果的な近傍構造を構成することによって，解の移動の多様性を広げるとともにより近似度の高い解を探索することを可能とした。さらに，禁断リストにより解のサイクリングを避けるとともに探索空間の効果的な削減を行うことによって高速な処理を実現した。次に数値実験により算法

の特性および挙動の分析を行い，パラメーターの適正化に対する傾向を明らかにした。また，全般的にTabu Searchは無閉路有向グラフの系列分割問題に対して近似度の高い結果を得ることが示され，最良解の探索に効果がある算法と結論できうる。

最後に，以上の数値実験は小樽商科大学情報処理センターのArgoss5270（SUN）およびDEC3000上で行った。

参 考 文 献

1）Betts,j. and Mahmoud,k.I.:A Method for Assembly Line Balancing, Engineering Costs and Production Economics, Vol.18, pp55−64(1989).

2）Kernighan,B.W. : Optimal Sequential Partitions of Graphs, J.ACM, Vol.18, No.1, pp.34−40 (1971).

3）Glover, F.: Tabu Search I, ORSA J.C. ,Vol.1, No3, pp190−206 (1989).

4）Glover, F.: Tabu Search II, ORSA J.C. ,Vol.2, No1, pp4−32 (1990).

5）Reeves,C.R : Mondarn Heuristic Techniqus for Combinatorial Problems, BlackWell (1993).

6）Salveson,M.E.:The Assembly Line Balancing Problem, The Journal of Industrial Engineering, May−June, pp.18−25 (1955).

7）茨木俊秀：　組合わせ最適化−分枝限定法を中心として−，産業図書（1983).

8）加地：半順序の最適系列分割問題の構造と算法構成，商学討究，Vol.45，No.2，pp185−204（1994).

9）加地，大内：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，Vol.35，No.3，pp.364−372（1994).

10）久保：巡回セールスマン問題への招待，日本ＯＲ，Vol39，No3，pp156−162（1994).

11）日本ＯＲ学会第30回シンポジウム：モダンヒューリスティクの新展開−Gnetic Algorithm, Simulated Annealing, Tabu Search, Neural Net法は本当に有効か−，(1994).

12）藤沢，久保，森戸：Tabu Searchのグラフ分割問題への適用と実験的解析，電学論c，Vol.114, No4, pp430−437（1994).