

Tabu Search法による無閉路有向グラフの 最適系列分割問題の解法

正員 加地 太一 (小樽商科大学)

正員 大内 東 (北海道大学)

A Tabu Search Approach to the Optimal Sequential Partitions of Directed Acyclic Graphs
Taichi Kaji, Member (Otaru University of Commerce), Azuma Ohuchi, Member (Hokkaido University)

Tabu search is a novel technique for solving combinatorial optimization problems. The process in which the tabu search method seeks to transcend local optimality is based on an evaluation function which chooses the highest evaluation move in terms of objective function and tabu restrictions. This paper presents a tabu search algorithm for finding a minimum cost partition of the nodes of a directed acyclic graph into subsets of a given size, subject to the constraint that the precedence relation are satisfied. A standard tabu search approach can't find good solutions for this problem, because the problem is complex multiple partitioning problems in which number of subsets and number of nodes in each subset are unsettled. For this problem, we use appropriate data structure for this method, and develop effective neighbourhood structure and heuristics. And, we assess the effectiveness of the developed algorithm. The results show that this algorithm is effective in obtaining near-optimal solution to this problem. the running time of the procedure is proportional to the number of nodes in the graph.

キーワード：Tabu Search法、近傍構造、最適系列分割問題、無閉路有向グラフ、近似解法

1. はじめに

無閉路有向グラフの頂点を先行順位の関係を見捨てずに各成分に分割する問題を系列分割問題と呼ぶ。本論文では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い、この問題を“総カット値を最小化する系列分割問題”と呼ぶことにする。その応用事例として、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は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、各生産プロセスをステーションに輸送コストの総和が最小になるように、先行順位の制限を見捨てずに、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。このモデルは2次記憶装置を中間データの中継に用いる多プロセスからなるデータ処理系などに適用できるものと考えられる。また、コスト関数や制約条件を問題の特性に合わせて変えることによって、ライン・バランス問題^{(1),(7)}の問題など多様な問題に適応させる

ことが可能である。

この総カット値を最小化する系列分割問題に対して、動的計画法を適用することによって厳密解法^{(9),(10)}が構成できる。この解法は無閉路有向グラフの構造に強く依存しており、並列構造が認められるとき多項式オーダーであるが、それでも頂点数、および並列数の増加に対して計算、および記憶容量に重い負担がかかることは否めない。また、ランダムな構造を持つ無閉路有向グラフでは指数オーダーの計算時間を必要とし、この厳密解法では頂点数の多いランダムグラフに対して実用的な時間内での計算は不可能である。これに対して我々は昨今、種々の問題で優れた成果を示しているTabu Search法^{(3),(4),(5),(6),(8),(12),(13),(14),(16)}を用い、本問題の近似解法の構成を試みる。Tabu Search法は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。

すでに、Tabu Search法により優れた効果を示したグラフの2分割問題に関する研究⁽¹⁶⁾があるが、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、より複雑化したグラフ分割問題となる。したがって、標準的なTabu Search法(近傍構造の設定等)の適用では効果的な結果は得られ

にくい。この複雑な構造に対して、本論文では、無閉路有向グラフの系列化グラフとその上で与えた分割を表現するブレイク・ポイントの集合によって解である系列分割を表現し、系列化グラフ上での系列分割を維持する頂点の移動、およびブレイク・ポイントの移動、付加、削除により、新しい解への移動を表現し、これらを複合統括して本問題に効果的な独自の近傍移動を実現し、Tabu Search法による算法を構成する。この複合的な近傍移動により解に大きな変化をほどこしTabu Search法の性能を引き出す。また、解の近似度を増すために、本問題の特徴を利用したヒューリスティックな戦略を組み込む。特に、ヒューリスティックな知識を取り入れたコスト変化量の計算式、また求められた最良解の情報を利用することによって、さらに良好な解を探索する戦略などに効果が認められた。最後に本算法について数値実験を行った結果を示し、その性質、性能、および戦略の効果を明らかにするとともに、Tabu Search法の本問題への有効性と、より複雑なグラフ分割問題への利用可能性について述べる。

2. 最適系列分割問題

先行、後続の関係をもつ要素集合の構造は、その要素を頂点とし、要素とその後続要素からなる要素の有向対を辺とする無閉路有向グラフ $D(V, E)$ によって表すことができる。ここで V は頂点集合、 E は有向辺の集合である。これから扱う系列分割の問題では単一の入口と出口を持つ無閉路有向グラフ $D(V, E)$ を考察の対象とする。 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だし、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。関係 \preceq は要素間の先行順位関係を表しており、 $v \preceq u$ は v が u より先行する順位にあることを示している。また、 $v \preceq u$ かつ $v \neq u$ を $v \prec u$ で表す。ここで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 $\{V_1, V_2, \dots, V_k\}$ 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とを、これを系列分割と呼ぶ。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⁽¹⁵⁾ 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。

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。

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。

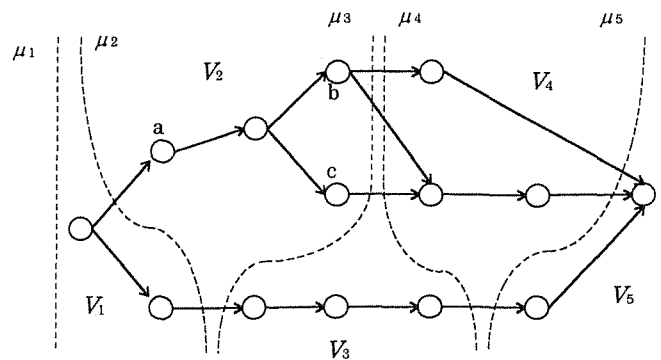


図1 無閉路有向グラフの系列分割
Fig. 1. Sequential partitions of a directed acyclic graph.

$X|Y$ 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 $\{V_1, V_2, \dots, V_k\}$ 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ことを、この分割を $D(V, E)$ の系列分割という。

このとき、各成分 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 $A_i = V_i \cup V_{i+1} \cup \dots \cup V_k$ 、下組 $A_i^c = V_1 \cup V_2 \cup \dots \cup V_{i-1}$ からなる切断 $\mu_i = (A_i^c, A_i)$ を対応づけ、切断の順序列 $\mu_1, \mu_2, \dots, \mu_k$ を作ると隣り合う切断の上組 A_i, A_{i+1} から、 $V_i = A_i - A_{i+1}$ となり、系列分割 $V_1 | V_2 | \dots | V_k$ が得られる。

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。本論文では“総カット値を最小化する系列分割問題”を扱う。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。以後、この問題を単に最適系列分割問題と呼ぶ。

3. Tabu Search法の概略

Tabu Search法は局所探索法の変形である。通常の局所

探索法の場合、探索の過程で現在の解 x から再びもとの解 x に戻るサイクリングが生じる可能性がある。こ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 法は最近の 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属性の変化をタブーリストに記憶し、これらの属性の逆方向への変化を禁止する処置をとる⁽⁸⁾。ここで、タブーリストを α とすると、Tabu Search 法では現在の解 x^{now} の近傍集合 $N(x^{now})$ から α を除いた集合 $N(\alpha, x^{now}) = N(x^{now}) - \alpha$ を構成し、その中から最小コストとなる解 x^{next} を選択し探索を進める。ただし通常は、 α の要素として解そのものを扱わず解の移動に関連した属性を用いる。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、リストの構築等が、この算法にとって重要な役割^{(14),(16)}を担い、大きな影響を及ぼす。この構成によりサイクリングをとまなう無駄な探索を避け、 $N(\alpha, x^{now})$ 中の最良な解を選択することにより近似度をより高める方向へと解を移動させる。また、算法の構造が単純に構成できるため実際の計算時間が少なくすむ。および、Tabu Search 法は非常に柔軟性の高い枠組みであり、その特性を失うことなく種々の戦略を組み合わせ性能を上げやすい等などの利点を持っている。

4. 近傍構造とタブーリストの基本的考察

Tabu Search法の算法を構成するにあたって重要な構成要素は解の近傍構造とタブー要素の構成である。無閉路有向グラフ $D(V, E)$ の系列分割を V_1, V_2, \dots, V_k として、問題の解を $x = (V_1, V_2, \dots, V_k)$ で表す。このとき、 V_i によって誘導された $D(V, E)$ の部分グラフを $\mathcal{D}(V_i)$ とすると、 $\mathcal{D}(V_i)$ の有向辺が定める V_i の順序関係において、極大元または極小元となる頂点 $v \in V_i$ は系列分割の性質を変えることなく、他のある成分 V_j に移動できる。図1の部分グラフ $\mathcal{D}(V_2)$ に対して、極大元は頂点 b, c にあたり、極小元は頂点 a にあたる。まず説明のために記号 $d^+(v, V_i)$ と $d^-(v, V_i)$ を導入する。 $v \in V$ とし、 v を始点とする D の有向辺で成分集合 V_i 中に終点を持つ辺の個数を $d^+(v, V_i)$ で、また V_i 中に始点を持ち v を終点とする有向辺の個数を $d^-(v, V_i)$ で示す。 $v \in V_i$ が (V_i, \leq) の極大元または極小元である条件は、それぞれ $d^+(v, V_i) = 0$ 、また $d^-(v, V_i) = 0$ で書き表すことができる。 v が極大元るとき、 $i < m$ に対して、 $d^+(v, V_i) = d^+(v, V_{i+1}) = \dots = d^+(v, V_{m-1}) = 0$ かつ $d^+(v, V_m) > 0$ が成立するならば、 v は $V_j \in \{V_{i+1}, \dots, V_m\}$ に移動可能である。これは、 $D(V, E)$ での v からの流出辺が $V_i, V_{i+1}, \dots, V_{m-1}$; $m < i$ 内の頂点に流入せず、 V_m に初めて流入する状況を表している。図1の部分グラフ $\mathcal{D}(V_2)$ に対する場合、極大元 b, c はともに V_3 と V_4 へ移動可能である。同様に、 v が極小元るときは $m < i$ として $d^-(v, V_i) = d^-(v, V_{i-1}) = \dots = d^-(v, V_{m+1}) = 0$ かつ $d^-(v, V_m) > 0$

のとき、 v は $V_j \in \{V_m, \dots, V_{i-1}\}$ に移動可能である。これは、 v への流入辺が $V_{m+1}, \dots, V_{i-1}, V_i$; $m < i$ 内の頂点から流出していない状況に対応している。この V_i から V_j への v の移動を $e(v; V_i, V_j)$ で表す。また、同様な条件のもとで、 v が極大または極小元るとき、 V_j の直前または直後の位置までに新しい空の成分を作り、そこに v を移動して系列を拡大することができる。この v の空の成分への移動を $a(v; V_i, V_j)$ で表す。さらに、 $e(v; V_i, V_j)$, $a(v; V_i, V_j)$ 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。

与えられた解 $x = (V_1, V_2, \dots, V_k)$ に移動 $e(v; V_i, V_j)$ 、または $a(v; V_i, V_j)$ を行うことによって、新しい解 x' が得られ、このような x' の集合が与えられた解 x の近傍集合を形成する。

次に、タブーの要素としては解そのものではなく、通常、解の移動に関連した属性値をタブー要素とするのが都合がよい^{(13),(14)}。本問題の場合、頂点の移動によって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。

5. Tabu Search法による近似解法の実現

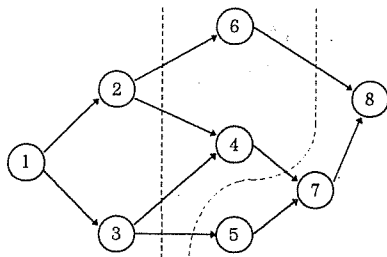
まず、系列分割問題で無閉路有向グラフを直接利用した算法構成をとると、集合演算を主体とする複雑な処理が要求され、プログラムの計算時間に無駄が多い。そこで無閉路有向グラフを系列化グラフで表現することによって、系列分割の計算処理に適した形にする。このようにすることによって、容量制約を満たす最良のブロック分割を求める局所的な最適化のアルゴリズムに適合しやすくなり、問題自体も容易に表現できるなどの利点が生じる。

次に、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、また各成分集合は頂点の重みの和がブロックサイズ B を越えてはならないという容量制約を受けている。そこで、移動 $e(v; V_i, V_j)$, $a(v; V_i, V_j)$ を用いた標準的な Tabu Search法の構成には幾つかの問題点がある。まず、現在の解から次の解への最適移動を決定する際の探索の組合せの複雑さ、特に移動 $a(v; V_i, V_j)$ はコストを悪化させる方向にのみ働くので、何等かの先見がなければこの移動を積極的に利用することができないなどの問題点がある。これに対して、本論文ではグラフ分割問題における従来の近傍操作である `left-to-right` 移動、`right-to-left` 移動⁽¹⁴⁾の考えを拡張し、近傍移動の複合処理をほどこすことによって解に大きな変化を与え、問題の解決を計っている。まず、現在の解から容量制約を無視して、`left-to-right` 移動、`right-to-left` 移動の複合移動により

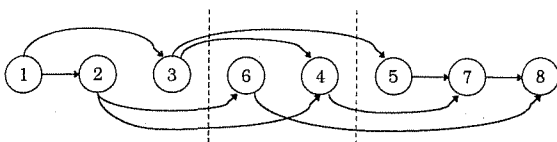
複数の頂点を移動させることによって、目的関数の値をできるだけ改善した系列分割を、探索領域を非実行可能解の範囲まで広げて求める。次に、この系列分割に対して容量制約を満たす最良のブロック分割を求めることによって、実行可能性を回復すると共にさらに近似度が改善された解を求める。以上の過程を複合近傍移動として構築し、この独自の複合近傍移動をTabu Search法の枠組みのもとで、十分近似度の良い解が得られるまで反復するという方策を採用する。

< 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 $D(V, E)$ の頂点集合 V はいつでも一列化(トポロジカルソート)が可能である。頂点をこの一列化の順に左から右に並べ換えて得られる同型なグラフを一列化グラフと呼ぶことにする。一列化された頂点列を $\{v_1, v_2, \dots, v_n\}$ とすると、一列化グラフの各頂点はこの順に左から右に並んでおり、すべての辺の向きは常に左から右に向かっている。また連続した頂点の並びからなる部分列 $\{v_p, v_{p+1}, \dots, v_q\}$ は系列を保持する V の部分集合となることを容易に示すことができる。それゆえ、無閉路有向グラフの系列分割は一列化グラフとその頂点列上の分割点(ブレイク・ポイント⁽²⁾)の並びを与えることによって生成できる(図2参照)。ここで部分列 $\{v_p, v_{p+1}, \dots, v_q\}$ をブロックと呼び、 $[v_p, v_q], [v_p, v_{q+1})$ などの記法を用いて表わす。ブレイク・ポイントの挿入位置を示すインデックスを $1 \leq p_1 < p_2 < \dots < p_k \leq n$ とし、ブレイク・ポイントの並びを $b_i = v_{p_i}; i=1, 2, \dots, k$ とすると、ブロック $[b_i, b_{i+1}); i=1, 2, \dots, k$ の系列は V の系列分割を生成し、 $[b_i, b_{i+1})$ はその成分集合 V_i となる。したがって、無閉路有向グラフの一列化された頂点列とブレイク・ポイントの集合を操作することによって解とその移動を表現することができ、Tabu Search法の構成を容易にする。



(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2. 無閉路有向グラフと一列化グラフの関係
Fig.2. Relation between a directed acyclic graph and a sequential graph.

< 5. 2 > 複合近傍移動の実現

本問題の特性から、移動 $e(v; V_i, V_j), a(v; V_i, V_j)$ の単純な適用による構成は処理の複雑化、解の近似度の向上の観点から問題が多い。そこで、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を用い、その上でブロック間の複合的な頂点の移動、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、複合的な近傍移動を実現し問題の解決をはかる。

まず、頂点の移動は本問題の特色に合わせて構成したleft-to-right移動、right-to-left移動を順次隣接する2つのブロック間に定義する。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。このとき移動可能な頂点が存在しなければ、 V_i, V_{i+1} はそのままとする。この処理を解 $x=(V_1, V_2, \dots, V_k)$ に対してある演算子を作用させた結果とみなし、 $x' = \bar{N}_i(\text{tabu-to-right}) \cdot x$ で表わす。この演算子の操作で、解 x の第 i 成分と第 $i+1$ 成分は $V'_i = V_i - \{v_0\}, V'_{i+1} = V_{i+1} \cup \{v_0\}$ と変化するが、その他の成分は全く変化を受けない。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 $\bar{N}_i(\text{tabu-to-left})$ を用いて表わす。

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。

$$x' = \bar{N}_{k-1}(\text{tabu-to-right}) \cdot \bar{N}_{k-2}(\text{tabu-to-right}) \dots \bar{N}_1(\text{tabu-to-right}) \cdot x$$

これを多次元left-to-right移動と呼び、

$x' = \overline{\text{multi}N}(\text{tabu-to-right}) \cdot x$ 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。

$$x' = \bar{N}_2(\text{tabu-to-left}) \cdot \bar{N}_3(\text{tabu-to-left}) \dots \bar{N}_k(\text{tabu-to-left}) \cdot x$$

これを多次元right-to-left移動と呼び、

$x' = \overline{\text{multi}N}(\text{tabu-to-left}) \cdot x$ 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、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、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。多次元移動によって得られた解 $x=(V_1, V_2, \dots, V_k)$ がある一列化グラフとブレイク・ポイントの単調増加列 $\{b_1, b_2, \dots, b_k\}$ 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。

Procedure Tabu_Search

```

begin
  “無閉路有向グラフを一系列化する。”
  “実行可能解を与えるブレイク・ポイントの列を設定し、
  初期解を  $x$  とする。”

  tabu-to-left :=  $\emptyset$ ;
  tabu-to-right :=  $\emptyset$ ;
   $x^* := x$ ;
  tabulength := positive integer;
   $t := 1$ ;
  while stop-criterion  $\leftrightarrow$  yes do begin
     $x' := \overline{multiN}(\text{tabu-to-right}) \cdot x$ ;
     $L_t :=$  “解移動( $x, x'$ )により移動した頂点の集合”;
    tabu-to-left := tabu-to-left  $\cup$   $L_t$ ;
     $x'' := \overline{multiN}(\text{tabu-to-left}) \cdot x'$ ;
     $R_t :=$  “解移動( $x', x''$ )により移動した頂点の集合”;
    tabu-to-right := tabu-to-right  $\cup$   $R_t$ ;
     $x''' := dp(x'')$ ;
    if  $f(x''') < f(x^*)$  then  $x^* := x'''$ ;
     $x := x'''$ ;

    tabu-to-right := tabu-to-right - ( $R_t - \text{tabulength} - \bigcup_{i=t-\text{tabulength}+1}^t R_i$ );

    tabu-to-left := tabu-to-left - ( $L_t - \text{tabulength} - \bigcup_{i=t-\text{tabulength}+1}^t L_i$ );

     $t := t + 1$ ;
  end;
  best solution :=  $x^*$ 
end;
```

図3 Tabu Search 法のアルゴリズム
Fig.3. An algorithm of Tabu Search method.

ここで $dp(x)$ は Kernighan の一系列化グラフの最適系列分割問題を解く算法^{(2),(11)}を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、与えられた一系列化グラフのもとで最適なブレイク・ポイント列 $\{b'_1, b'_2, \dots, b'_k\}$ を見出す。その結果、頂点のブロック間での移動によるコストの改善と、実行可能性の回復が行われる。

これらの処理を一反復過程において

$x' = dp(\overline{multiN}(\text{tabu-to-left}) \cdot \overline{multiN}(\text{tabu-to-right}) \cdot x)$ の順序で行い、局所的に最良な近似効果を持つ近傍移動を達成する。

以上の処理は、複数回の基本操作 $e(v; V_i, V_j)$, $a(v; V_i, V_j)$ によって、現在の実行可能解から近似度のより改善された可能解に短時間で導く操作を一反復過程ごとに行っている。また、基本操作による隣接していないブロックへの頂点移動も可能である。これらより本処理の反復が、頂点の移動可能性を高めてより多様性のある組合せを生成することとなり、良質の実行可能解の多くを生み出すことになる。この考えにもとづく Tabu Search 法の算法を図3に示す。

< 5. 3 > タブーリストの実現

解移動 (x, x') に対しての逆向き移動を禁止する属性として、(x, x') のとき移動する頂点をあてることとし、多次元 left-to-right 移動と多次元 right-to-left 移動の 2 つの移動形式

に対してそれぞれ 2 つのタブーリスト tabu-to-left と tabu-to-right を設け、そのとき関与する頂点の集合をタブーリストに記憶する。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⁽¹⁶⁾。このとき、初期化の段階で配列のすべての要素を 0 におき、移動対象となる頂点の要素にある正の数を入れ（すでに要素に値が入っていても、ある正の数を入れなおすこととする）、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 tabulength1 の値、そうでないのならば tabulength2 の値を与える。図3では tabulength1 と tabulength2 の値を同一な tabulength としている。tabulength の長さにより禁断の期間を限定しているのは、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ためである。この長さは理論的に推察することが困難であるため、通常、事前の数値実験の傾向によりその禁断の期間であるタブーリストの長さを判定する。

< 5. 4 > 頂点移動にともなうコスト変化量の計算

解に $\overline{N}_i(\text{tabu-to-right})$ を作用させた過程で生ずる頂点 $v \in V_i$ の V_i から V_{i+1} への移動に伴うコスト変化量 $\bar{\delta}(v, V_i, V_{i+1})$ 、および $\overline{N}_i(\text{tabu-to-left})$ における頂点 $v \in V_i$ に関する同様なコストの変化量 $\bar{\delta}(v, V_i, V_{i-1})$ 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。ただし、この計算において、無閉路有向グラフ上に存在しない辺については考慮しないものとする。

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

$$\bar{\delta}(v, V_i, V_{i-1}) = \sum_{t \in V_i} c(v, t) - \sum_{s \in V_{i-1}} c(s, v) \quad \dots \dots (2)$$

これらの式の使用は、グラフが並列構造を持つとき良好な結果に導くが、全く無秩序なランダムグラフに対しては不満足なことが多い。本問題の場合、一系列化グラフ上の頂点間の辺の長さ（頂点の添字の差）をより短くするというヒューリスティックな戦略が効果をもつことが経験により示される。これを反映するために、コスト変化量式を以下のように変形しその効果を確かめる。一系列化グラフ上で、 v からの流出辺（への流入辺）の端点となる頂点の中で最も左側（右側）に存在する頂点の直前（直後）に v を移動すると考えたとき、この v を左端点（右端点）とするブロックで、頂点の重さの総和がブロックサイズ B を越えないものの中で、重さの総和が最大であるものを $V^*(V)$ で表わす。このとき、 $V^*(V)$ によって表される以下の新たなコスト変化量を用いることによって、ランダムグラフにも、構造を持ったグラフにも対応できる高い近似度を達成する

ことが可能となる。

$$\bar{e}(v, V_i, V_{i+1}) = \sum_{s \in V_i} c(s, v) - \sum_{t \in V_{i+1}} c(v, t) \quad \dots \dots (3)$$

$$\bar{e}(v, V_i, V_{i-1}) = \sum_{t \in V_i} c(v, t) - \sum_{s \in V_{i-1}} c(s, v) \quad \dots \dots (4)$$

< 5. 5 > 終了判定基準

終了判定基準としては、ある反復回数 $iterate$ を繰り返す方法、または探索解が更新された段階から、この値が更新されなくなつてからの経過時間がパラメータ $stop$ を越えたときに、算法が終了する方法などが考えられる。求まる最良解が2000前後の反復回数以内で決定される場合が多いことから、前者の方法では $iterate$ を2000前後とし（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。）、後者の方法ではパラメータ $stop$ を1000前後とするのが望ましい。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。

< 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、分割されたブロックの部分的な一致が見られる。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い、それを出発点とし、さらに良好な解を探索することによって解を改善する戦略が考えられる。この考えを取り入れ、現在得られている最良解とその暫定値を初期値として、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高い⁽¹⁶⁾。また、再出発による解の決定は反復回数の初期の段階で決定される傾向がみられるので、この過程での終了判定基準に用いる反復回数 $iterate$ 、およびパラメータ $stop$ の値は前記で示した数の1/2の値を用いる。

< 5. 6. 2 > 2頂点交換法

ある一列化グラフとブレイク・ポイントの単調増加列により表現された系列分割において、この一列化グラフ上に有向辺で結ばれていない隣接する2頂点 v_i, v_{i+1} が存在するとき、一列化の性質を変えずに、これらの2頂点を交換することができ、この交換によって新しい一列化グラフが得られる。 v_i と v_{i+1} の間に切断を考慮したときに、もとの一列化グラフから得られるこの切断のカットエッジのコストを c 、2頂点を交換した新しい一列化グラフから得られる同様なカットエッジのコストを c' とし、もし $c' < c$ が成立するならば v_i と v_{i+1} の交換をおこなうという操作を考える。ブレイク・ポイントの移動により一層のコスト

改善を行う次の段階に進む前に、この操作をすべての交換可能な2頂点に対して行っておくと、次の改善過程で2頂点の交換の位置に丁度ブレイク・ポイントの移動または生成が生じたとき、よりコストが改善された新たな系列分割が作り出される。この予知にもとづく方法を2頂点交換法と呼ぶ。本論文では現在得られている最良解に2頂点交換法を適用し、それに続いて $dp()$ 操作で解の近似度を上げる方策を試みている。

6. 数値実験

本章では上記で構成した Tabu Search 法の算法に対して特性評価と挙動解析を行う。なお、以下の数値実験は小樽商科大学情報処理センター Argoss5270(SUN4)および DEC3000 上で行い、計算時間に関する数値実験は Argoss5270(SUN4)を使用した。まず、Tabu Search 法を効果的に動作させるための最も重要なパラメータは禁止リストの長さである。本問題においては2つの禁止リスト $tabu\text{-}to\text{-}left$ および $tabu\text{-}to\text{-}right$ 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。このとき、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 $tabulength1$ 、そうでない場合 $tabulength2$ と2つのパラメータを用いる方法が考えられる。しかし、 $tabulength1$ の値に対して多少低めに $tabulength2$ の値を設定することにより若干の効果はあるものの、著しい効果は認められずその効果は低い。したがって、両者とも同じ値である $tabulength$ とし、算法を構成している。この $tabulength$ の適正値を判定するため、 $tabulength$ の値の変化に対してコストの値の変化を数値実験で確かめる。

図4、5、6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（コスト変化量式は(1),(2)を使用、パラメータ $iterate$ は4000とし、近似度の改善は行わない）。この実験より、ブロックサイズ10の場合、 $tabulength$ の長さは1、または2が適正であり、ブロックサイズ30の場合、6前後であり、ブロックサイズ50の場合、10前後となった。 $tabulength$ 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、ブロックサイズの1/10から1/5の大きさを $tabulength$ の適正値とすることが望ましい。このとき、グラフの構造に対するこの適正値の影響は少ない。

次に5.4章で述べた改良型のコスト変化量の計算式(3)、(4)を用いることによる効果について確かめる。図5で用いたグラフに対して同一の条件のもとでコスト変化量式(3)、(4)を適用した場合の実験を行い、その結果を図7に示す。これより、 $tabulength$ が20以内の範囲で求まる良質な解において、特に著しい改善が見られた。したがって、この変化量式の採用により無秩序なランダムグラフに対する効果が得られることが判明した。ただし、計算時間には影響はない。

第5.6章で示される局所最適解からの再出発による近

似度の改善の効果は表1より考察する。表1では図5で用いたグラフに対して代表的な値に対して、上記と同一な条件のもとで再出発を適用しない場合のコスト値と、適用した場合のコスト値との比較を示している。これより、再出発の効果により解が改良されていることがわかる。ただし、解の近似度が悪い場合、著しい効果をもたらすが、良質な解の場合、改善率は低い傾向が見出される。しかし、2頂点交換法を適用した場合、これによる改善が示される例は少なく、また改善される場合もその改善率は低い傾向にある。

論文(16)で述べられている長期メモリの効果について実験の結果を加えておく。長期メモリは長期的な巡回を防ぐため、頻りに移動を繰り返す頂点に対して移動しにくくさせるものであり、任意の頂点の移動回数とパラメーターBIASの積をペナルティ関数としてコスト変化量に加算し実現する。上記と同様な条件で図5で用いられたグラフを使用した一例を表2に示すと、明らかに長期メモリによる改悪の現象が見られる。tabulengthが0、1のときは改善されているが、これはtabulengthの長さが0あるいは短いため、タブー効果が存在しない算法となり、これに対して長期メモリを設けることよりタブー効果が付加され、タブーサーチの戦略と同一の構成となるものと思われる。また、BIASを上げることによりさらに改悪される傾向が示されている。この結果、少なくとも本問題に対しては長期メモ

リーは近似度の悪化の要素となる。

この問題に対して、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法との比較を試み、解の近似度について論じる。この厳密解法は探索領域の削除を考慮した動的計画法にもとづく算法である。表3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並列グラフに対するコストの比較である。コスト変化量は(1),(2)を使用し、パラメータiterateは2000とする。さらに、近似度の改善のアルゴリズムを加えている。しかし、グラフ以外は最良解が反復回数1000以内で発見され、また、 f と h に対してのみ2頂点交換法が行われている。表3よりエッジコストがすべて1の場合、Tabu Search法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。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場合、近似度は前記に比べて悪くなる傾向がある。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する傾向がある。

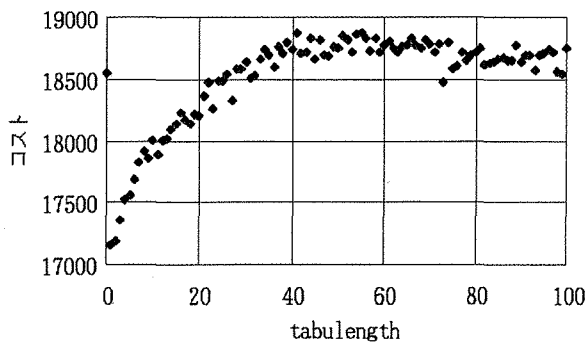


図4 ブロックサイズ10に対する
タブーリストの長さとおコスト
Fig.4. Tabu length vs. cost for block size 10.

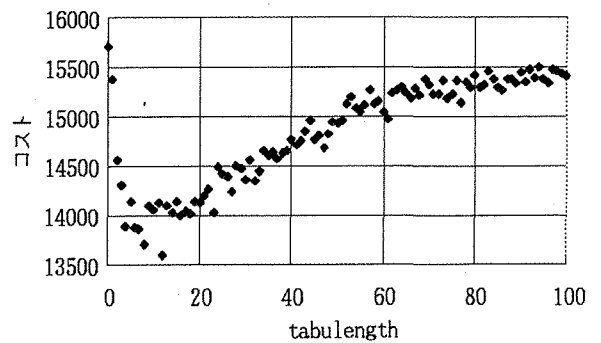


図6 ブロックサイズ50に対する
タブーリストの長さとおコスト
Fig.6. Tabu length vs. cost for block size 50.

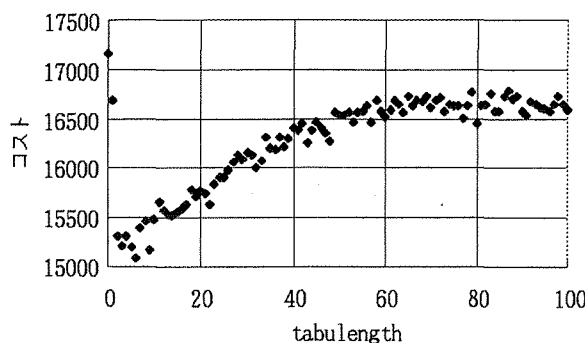


図5 ブロックサイズ30に対する
タブーリストの長さとおコスト
Fig.5. Tabu length vs. cost for block size 30.

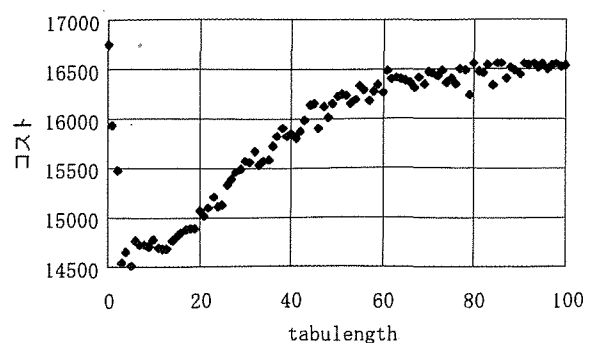


図7 $\bar{\epsilon}(\bar{\epsilon})$ に対するタブーリストの長さとおコスト
Fig.7. Tabu length vs. cost for $\bar{\epsilon}(\bar{\epsilon})$.

また、表4で並列路の間を結ぶ中間エッジが入ることによるグラフの複雑性に対して、厳密解法と比較したコスト、ならびに計算時間を併記し、近似度と計算の負担への影響を示す。本実験で使用する終了規則はパラメータiterateの値を1000とする。また、コスト変化量式は(1),(2)を使用し、解の改善は行っていない。これより中間エッジが増加することによるグラフの複雑性に本算法は影響を受けず、近似度(相対誤差)5%以内を維持する。

最後に、2並列グラフに対しての頂点数に対する計算時間とコストの変化を表5で示し、特に時間の増加傾向を図8で表す。このときのパラメータiterateは1000であり、コスト変化量式は(1),(2)を採用する。また、近似度の改善は行っていない。この実験より、本算法では最適解かほぼそれに近いコストの値が求まることが示される。さらに、厳密解法はべき乗で計算時間を要するのに対して、Tabu Search法はほぼ線形の計算時間で求まることが示される。また厳密解法では頂点数の大きいランダムグラフに対しては実質的に計算が困難であるが、本算法では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。これは反復数を限定することにより、一反復中に要する計算時間がすべての頂点の探索とエッジコストの参照および頂点の移動に依存し、その値がほぼ頂点数に比例するためである。

7. おわりに

本論文では無閉路有向グラフの系列分割問題に対するTabu Search法の適用とその有効性について検討している。本問題の近傍構造は基本的には $e(v; V_i, V_j)$ 、 $a(v; V_i, V_j)$ で実現できる。しかし、本問題は系列性を保持した多分割問題であり、成分集合の個数、各成分集合の要素数は不定である複雑なグラフ分割問題であるため、 $e(v; V_i, V_j)$ 、 $a(v; V_i, V_j)$ の単純な適用では計算時間、近似度ともにより良い結果は望めない。これに対して、本論文では一列化グラフとブレイク・ポイントの集合を用いたデータ構造を利用し、近傍移動の設計では本問題の構成に合わせて変形したleft-to-right移動、およびright-to-left移動を連続的に適用し、さらに局所的な最適化を取り入れ、複合的な処理方法を用い効果的な近傍移動を実現した。従来の近傍移動に対して、この複合的な近傍移動の採用により解に大きな変化をほどこしTabu Search法の性能をより強く引き出し、その特徴も維持する結果となった。さらに、問題独自のヒューリスティックな知識を取り入れることによって、より近似度の高い算法が実現できた。また、今回の数値実験より多くの知見を得ることができた。タブーリストの長さの適正值に関しては本問題の場合、ブロックサイズの大きさに強く依存する結果を得た。次に、ヒューリスティックな戦略を用いた場合の効果を示し、同時に解の改善の性質を明ら

表1 再出発による近似度の評価

Table 1. Effect for re-start.

Algo1: 再出発を用いない場合、 Algo2: 再出発を用いた場合

tabulength	0	1	5	6	8	10	20	40	60	80	100
Algo1	17161	16696	15190	15081	15460	15478	15772	16416	16528	16449	16598
Algo2	17161	16696	15180	15042	15342	15441	15487	16173	16127	16036	16103

表2 長期メモリーによる近似度の評価

Table 2. Effect for long-term memory function.

Algo1: 長期メモリーを使用しない場合、 Algo2: 長期メモリーを使用した場合

tabulength	0	1	5	6	8	10	20	40	60	80	100
Algo1	17161	16696	15190	15081	15460	15478	15772	16416	16528	16449	16598
Algo2	16007	15708	15637	15848	16058	16002	16378	16942	17160	17141	17129

表3 2並列グラフに対する近似解と厳密解

Table 3. Approximate and exact optimal solution for digraph of only 2 paths.

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
厳密解法	20	40	5	19	51	146	6	87
Tabu Search	20	41	5	20	58	168	6	89

表4 中間エッジ数による影響と計算時間

Table 4. Effect and running times by increasing crossing edges.

中間エッジ数	0	50	100	150	200	250	300
厳密解法	20	69	115	162	203	253	298
厳密解法の時間(ms)	23820	20352	25689	22075	24747	26843	25812
Tabu Search	20	70	119	167	211	260	307
Tabu Searchの時間(ms)	7023	9699	8443	7687	8493	12014	8083

表5 頂点数による影響と計算時間

Table 5. Effect and running times by increasing number of nodes.

頂点数	50	100	150	200	250	300
厳密解法	5	10	15	20	25	30
厳密解法の時間(ms)	699	3808	11729	23638	45947	76949
Tabu Search	6	11	15	20	25	31
Tabu Searchの時間(ms)	1419	3049	4709	6449	8228	9898

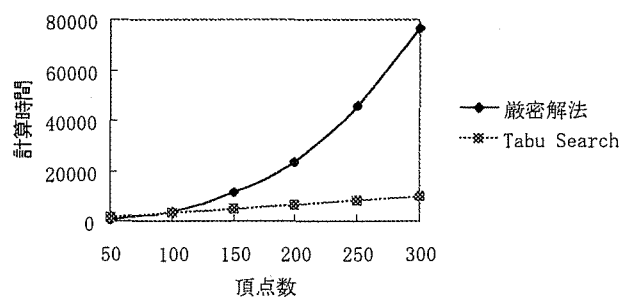


図8 頂点数と計算時間 (ms)

Fig. 8. Number of nodes vs. computing times.

かにした。本実験から、タブーリストの適正值、および長期メモリーの効果など、2分割問題では示されていない性質、および異なる性質などの知見を得たことは以後の研究にとって興味深い。さらに、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。同時に、Tabu Search法はより複雑なグラフ分割問題に対しても優れた効果が実現できる有効な手法であることが示された。さらに、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・バランシング問題などへのTabu Search法の適用および効果について検討を試みたい。

(平成7年7月3日受付、同8年2月20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) Glover, F.: "Tabu Search Part I", ORSA J.C., Vol.1, No.3, pp.190-206(1989).
- (4) Glover, F.: "Tabu Search Part II", ORSA J.C., Vol.2, No.1, pp.4-32(1990).
- (5) Glover, F.: "A User's Guide to Tabu Search", Annals of Operations Research, 41, pp.3-28(1993).
- (6) Reeves, C.R.: "Modern Heuristic Techniques for Combinatorial Problems", BlackWell(1993).
- (7) Salvesson, M.E.: "The Assembly Line Balancing Problem, The Journal of Industrial Engineering", May-June, pp.18-25(1955).
- (8) 茨木: 「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: 「無閉路有向グラフにおける系列分割問題の算法構成」, 情報処理学会第47回全国大会講演論文集(1), pp.83-84(1993).
- (10) 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, pp.185-204(1994).
- (11) 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).
- (12) 久保: 「巡回セールスマン問題への招待」, 日本OR, Vol.39, No.3, pp.156-162(1994).
- (13) 久保: 「モダンヒューリスティックスの新展開—Gnetic Algorithm, Simulated Annealing, Tabu Search, Neural Net 法は本当に有効か—」, 日本OR学会第30回シンポジウム(1994).
- (14) 久保: 「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).
- (15) 西関, 秋山: 「グラフとダイグラフの理論」, 共立出版(1981).
- (16) 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, No.4, pp.430-437(1994).