

(4): スケジューリングと Tabu Search

高山 裕志, 久保 幹雄, 森戸 晋

1 はじめに

本稿で紹介する Tabu (Taboo) Search (禁断探索法) は, 数あるメタ解法の中でも新参者の部類に属し, そのため知名度はいま一步であるが, 最近では, 最も強力な近似アルゴリズムの一つと認識されつつあるメタ解法の有望株である. 特に, スケジューリングや配置問題に代表される極めて困難な組合せ最適化問題に対して, 次々と世界記録を更新していることは特筆に値する. ここでは, Tabu Search で用いられる諸概念の簡単な紹介と, スケジューリング問題の中でも最も困難であると言われているジョブショップ問題への適用について述べる.

2 Tabu Search 入門

まず, Tabu Search の概要を説明するために, 表 1 に示す 4 ジョブからなる簡単なスケジューリング問題を考える. 各ジョブ i は決められた処理時間 p_i と納期 d_i を持ち, 全てのジョブを一台の機械で加工する.

表 1: 一機械総納期遅れ最小化問題

ジョブ	i	1	2	3	4
処理時間	p_i	1	2	3	4
納期	d_i	5	9	6	4

ジョブの集合 $\{1, 2, 3, 4\}$ から構成される $4!$ 個の順列を実行可能解 (または単に解) と呼び, その集合を \mathcal{F} と記す. 評価尺度 (目的関数, 費用関数) は, ジョブごとの納期からの遅れの合計値とする. すなわち順列 (π) に対する目的関数 $f(\pi)$ (π はジョブ j の処理の順番) は, $\sum_{i=1}^4 [\sum_{\pi(j) \leq \pi(i)} p_j - d_i]^+$ と定義される. ここで $[x]^+ = \max\{x, 0\}$ である.

一般に, 組合せ最適化問題とは与えられた実行可能解の集合の中で, 目的関数を最小にするものを見つけ

ることを目的とする. 目的関数値が最小になる解を大域的最適解と呼び, その集合を \mathcal{F}^* と記す. すなわち, $\mathcal{F}^* = \{x \in \mathcal{F} : f(x) \leq f(y), \forall y \in \mathcal{F}\}$ である.

ここで扱う例題は, 一般には一機械総納期遅れ最小化問題 (Graham や Lawler ら [15, 21] の記号を用いれば $1 \parallel \sum T_j$; 茨木 俊秀, "スケジューリングの新解法 (1), スケジューリング問題と計算の複雑さ," 本誌 1994 年 10 月号参照) と呼ばれ, 最近まで計算複雑性は未解決であったが, 1990 年に Texas 大学の Du と Leung によって, \mathcal{NP} -困難であることが証明された [10]. このことは, 最悪の場合の計算量が多項式時間で抑えられる最適解法が ($P \neq \mathcal{NP}$ の仮定の下では) 存在しないことを示し, 大規模な問題に対して Tabu Search 等の近似解法を適用する言い訳が成立する. なお, 計算複雑性の理論は, 全ての \mathcal{NP} -困難な問題が実用的な意味で解けないことを保証している訳ではないことに注意されたい. 総納期遅れ最小化問題に対しては効率的な最適化手法が数多く提案されているので, Tabu Search 等の近似解法は大規模問題もしくは計算時間が極端に限定されている場合に限られる.

Tabu Search をはじめとする多くの近似解法では, 解の間に適当に定められた "近さ" の概念を利用する. ここでは簡単のため, 二つの隣り合うジョブを交換することによって得られる解を "近い" 解と定義する. (実用的には任意の異なる二つのジョブを交換する方法が推奨される.) 近い解の集合は近傍と呼び, 解 $x \in \mathcal{F}$ の近傍を $N(x)$ と書く. 実行可能解を点とし, 枝集合を $\{(x, y) : x \in \mathcal{F}, y \in N(x)\}$ と定義することによって有向グラフが構成される. このようなグラフを近傍グラフと呼ぶ (図 1 参照: 例題では近傍は対称であるので無向グラフとして記してある). 近傍グラフ上で隣接する 2 点に対応する解は近傍になっている. 例えば解 1234* の近傍は 2134, 1324, 1243 である.

近傍グラフ上を目的関数値が減少する方向に進るこ

*ここでは順列 (1, 2, 3, 4) を単に 1234 と記す.

たかやま ひろし*, くぼ みきお**, もりと すずむ*

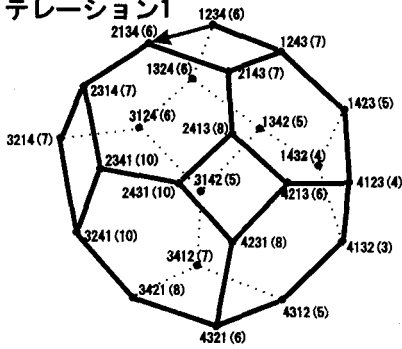
* 早稲田大学理工学部 工業経営学科

〒169 東京都新宿区大久保 3-4-1

** 東京商船大学 流通情報工学

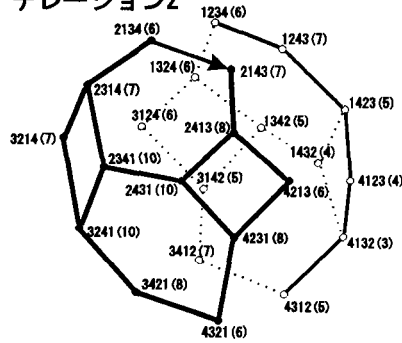
〒135 東京都江東区越中島 2-1-6

イテレーション1



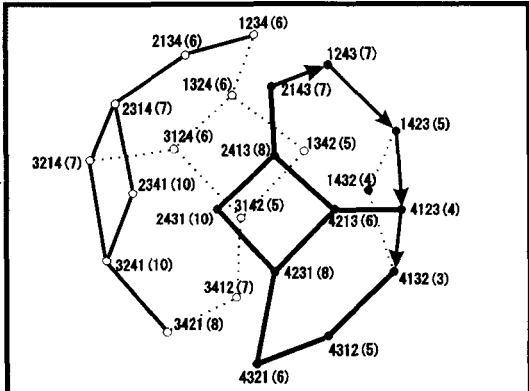
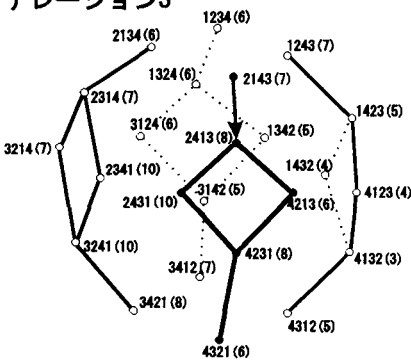
解1234から出発する。この解は局所最適であるので、Local Searchはここで停止する。
Tabu Searchにおいては、近傍から目的関数値の減少量が最大のもの(タイは適当に破る)へ移ることによって探索を継続する。この場合は、解2134へ移動する。

イテレーション2



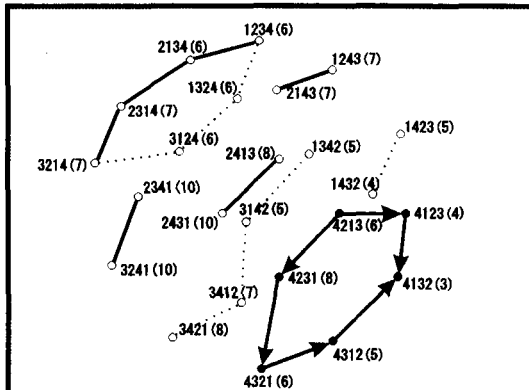
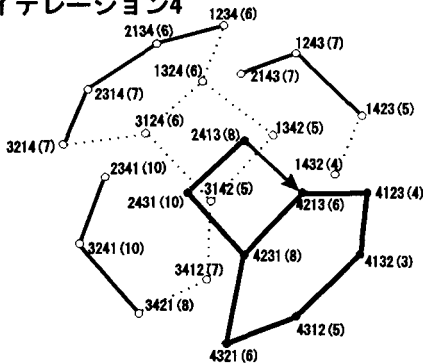
その結果としてジョブ2と1を交換する全ての移動が禁止される(イテレーション2)。禁止されていない移動から目的関数値の減少量が最大のものを選択することによって探索が進む。

イテレーション3



Tabu Lengthが1のときは、解2143から1243への移動が可能であるので、1423, 4123を経由して最適解4132に到達する。

イテレーション4



Tabu Lengthが2, 3のときは4123を経由し最適解4132に到達し、Tabu Lengthが4, 5のときは4231, 4321, 4312を経由し、やはり最適解を得る。
Tabu Lengthが6以上のときは解4321で移行可能な解がなくなるが、Aspiration Levelを使えば、解4213から4123への移動が可能になり、最適解に到達する。

図1: 一機械総納期遅れ最小化の例題における Tabu Search の動き。各点には、解に対応する順列と目的関数値(括弧内)が付してあり、二つの解が互いに近傍で、かつ Tabu List によって禁止されていないとき枝が張られている。黒丸(白丸)は現在の解から枝を経由して到達可能(不可能)な解を表し、矢線は解の移動を表す。

とによって解を改善する方法が、古くから実用的な解法として用いられていた Local Search である (例えば [23, § 19] 参照). Local Search は、近傍の中に目的関数値を減少させるものがないときに停止する. このような解を局所最適解と呼ぶ. すなわち, $f(x) \leq f(y), \forall y \in N(x)$ を満たすとき $x \in \mathcal{F}$ は局所最適解と呼ばれる.

Tabu Search も Local Search と同様に近傍グラフ上を辿ることによって近似解を得るが、以下の点で異なる. 1) Tabu Search においては、目的関数値を減少させるような近傍が見つからないときでも停止せず、目的関数値の増大が最小の解へ移動を行う. 2) 再び同じ解へ戻ること (巡回現象) を避けるために、ある解から別の解への移動に伴う何らかの情報を記憶しておき、その情報をもとに近傍グラフの枝を除去する.

移動に伴う情報を属性 (attribute) と呼び、属性を記憶するリストを Tabu List (禁断リスト) と呼ぶ. 属性の定義は曖昧であるが、例題における属性は交換した二つのジョブと定義できる (属性の定義から曖昧さをなくす方法については [17] 参照). 例えば、ジョブ 1, 2 を交換した後では、ジョブ 1, 2 が Tabu List に保持され、近傍グラフから対応する枝が除去される (図 1: イテレーション 1, 2 参照) このように Tabu Search においては、近傍グラフから枝を除去しながら最良の方向に探索を進めるが、枝の除去が進むにつれ、行き場を失ってしまう可能性がある.

これを防ぐために、通常、Tabu List の長さを有限とし、Tabu List を先入れ先出しの待ち行列 (キュー) 構造にする (キュー構造を用いない、より柔軟な実装方法については [17] 参照). Tabu List の長さは、Tabu Length と呼ばれ Tabu Search の最重要パラメータである.

また、移動が Tabu List によって禁止されていても何らかの基準を満たせば、その移動を許すという方法も提案されている. ここで言う基準は Aspiration Level [11, 12] と呼ばれ、その定義はやはり曖昧ではあるが、例えば、移動後の目的関数値が現在までの探索で求められた最良値を更新するならば、その移動が許されるといった方法である.

例題における Tabu Search の探索の様子を図 1 にあげる. 例題においては、Tabu Length を適正にするか、Aspiration Level を用いることによって、最適解 (4132) に到達することが分かる. この例題は、Tabu Search の動きを極端に単純化したものであるが、複雑な問題に対しても基本的な動きは同様である. 通常 Tabu Search では、他にも色々な工夫を導入し、探索を知的なものに洗練させていく. 例えば、長期的な巡回を避けるた

めに、移動に伴う属性を長期的に記憶しておく長期メモリや周期的に Tabu List を空にする方法などがある. これらの工夫については、Tabu Search の提案者である F. Glover らによる論文 [11, 12, 13, 14] を参照されたい.

Tabu Search の終了条件については一定のルールが確立しているという訳ではなく、アルゴリズム設計者や利用者の裁量の余地が大きい. 停止条件としてよく用いられるものに、計算時間やイテレーション数、最良解が更新されてからのイテレーション数などがあげられる. 後 (表 4) に示す筆者らの研究では、最良解が一定回数の反復で更新されないときに停止する方法を用いている. なお、終了条件に関する自由度ないしは曖昧さは批判にもつながっているが、筆者らはこのことを Tabu Search の強みと考えている.

3 ジョブショップ問題に対する適用

ここで扱うジョブショップ問題は複数の処理機械からなるシステムを対象とし、生産される製品の処理機械やその訪問順序などは既知であり、すべての製品が完成するまでの時間 (総所要時間) を最小にするように各機械上の製品の処理順序を決定する問題である.

3.1 ジョブショップ問題の従来の研究

ジョブショップ問題の解法は分枝限定法をはじめとする最適解法の研究が古くからなされてきた. Carrier と Pinson の分枝限定法 [8] は 20 年以上未解決であった Muth と Thompson の仕事数 10 機械数 $10(10 \times 10)$ [†] の問題の最適解を得ることに成功した.

近似解法の研究では処理可能な作業の中から実際に処理する作業を選択するための優先順序規則 (デイスバッティングルール) の研究 [24] が盛んであったが、1988 年に Adams らによって Shifting Bottleneck 法 [1] が提案されたのを契機としていわゆるメタ解法の適用が盛んになった. 表 2 はメタ解法を用いた最近の研究をまとめている. Tabu Search に関する 1992 年頃までの研究は Barnes と Laguna の文献 [5] に総括されている.

3.2 Tabu Search の適用例

本節では前述の一機械問題よりも複雑なジョブショップ問題に対して 3×3 の例題を用いて Tabu Search を適用する.

ここではスケジュールの表現法として離接グラフによる表現を用い、近傍と Tabu List の構造を示す.

[†]仕事数×機械数の形で表す.

表 2: 過去の研究

問題	著者	適用解法
$1 \mid r_i \mid \sum_i w_i \cdot C_i$	Laguna et al. [18]	Tabu Search
$1 \mid r_i \mid \sum_i w_i \cdot C_i$	Laguna & Glover [19]	Tabu Search
$P \mid d_i - C_i \geq 0 \mid \sum w_i \cdot (d_i - C_i)$	Laguna et al. [20]	Tabu Search
$F \parallel \sum F$	Krone & Steiglitz [16]	Local Search
$F \parallel C_{\max}$	Widmer & Hertz [30]	Tabu Search
$F \parallel C_{\max}$	Taillard [26]	Tabu Search
$J \parallel C_{\max}$	Taillard [25]	Tabu Search
$J \parallel C_{\max}$	Dell'Amico & Trubian [9]	Tabu Search
$J \parallel C_{\max}$	Laahorven et al. [28]	Simulated Annealing
$J \parallel C_{\max}$	Yamada et al. [31]	Simulated Annealing
$J \mid \text{工具制約} \mid C_{\max}(\S)$	Widmer [29]	Tabu Search
$J \mid \text{代替機械} \mid C_{\max}$	Brandimarte [7]	Tabu Search

§ $C(s) = \alpha \cdot \text{総所要時間} + \beta \cdot \text{工具交換回数} + \gamma \cdot \text{納期遅れ仕事数} + \sigma \cdot \text{期間内未完了仕事個数}$
 $(\alpha, \beta, \gamma, \sigma \text{は重み}, \alpha = 2, \beta = \gamma = \sigma = 1)$

離接グラフ表現： スケジュールの表現はガントチャートによる表現が馴染み深いですが、同一機械上で処理する任意の2つの作業の先行関係が決まれば一意に処理順序を決定することができる。このことを用いてスケジュールをグラフ表現することができる。このグラフは離接グラフ (Disjunctive Graph) と呼ばれる。

V : 作業を表す節点の集合で、 $V = \{0, 1, \dots, N+1\}$, 節点0, $N+1$ は始点と終点を示すダミー節点である。各節点上ではそれぞれが表す作業の加工時間分の重みがかけられる。

A : 技術的順序を表す有向弧の集合

E : 同一機械上で処理される作業の先行関係を決定するための離接弧の集合

離接グラフ $G = (V, A, E)$ ですべての離接弧の方向を閉路を含まないように決定したとき、このグラフは一つの実行可能スケジュールに対応する。節点0から $N+1$ への経路で、通過する節点上の重みの総和が最大の経路をクリティカルパスと呼び、その重みの総和(クリティカルパスの長さ)が対応するスケジュールの総所要時間になる。3×3の例を図2に示す。

近傍： 近傍はクリティカルパス上の離接弧を1対反転したスケジュールの集合とする。

これは Balas [4] の陰的列挙法をもとに Laarhovenら [28] が提案した近傍である。彼らはこの近傍を用いて Simulated Annealing を適用し、Taillard [25] も同じ近傍で Tabu Search を適用した。

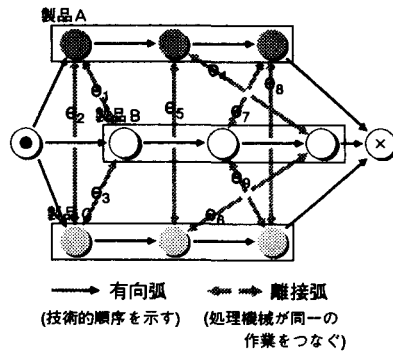


図 2: 離接グラフ $G = (V, A, E)$

Dell'Amico と Trubian [9] はこの近傍の拡張として複数の離接弧の反転を考慮した近傍を提案した。さらに同一機械上で連続して処理され、かつクリティカルパス上にある作業によって構成される部分仕事列に含まれる作業をその部分仕事列の先頭(または最後)に移す近傍を併用している(山田ら [31] の Simulated Annealing で後者の近傍が用いられている)。

Tabu List： 前述の近傍を用いたとき、離接弧の集合が属性の集合に対応する。そこで、離接弧の集合に対応する一次元配列を Tabu List(Ξ) とし、検索を以下のように行う。

1. 反転した離接弧 e に Tabu List の対応する箇所 ($\Xi(e)$) にその時点のイテレーション数 (t) と Tabu Length(TL) の和 $t + TL$ を書き込む
2. e を再び反転するとき Tabu List に書き込まれている数字と現在のイテレーション数 (t') を比較し、 $t' \geq t + TL$ のとき e の反転が許される

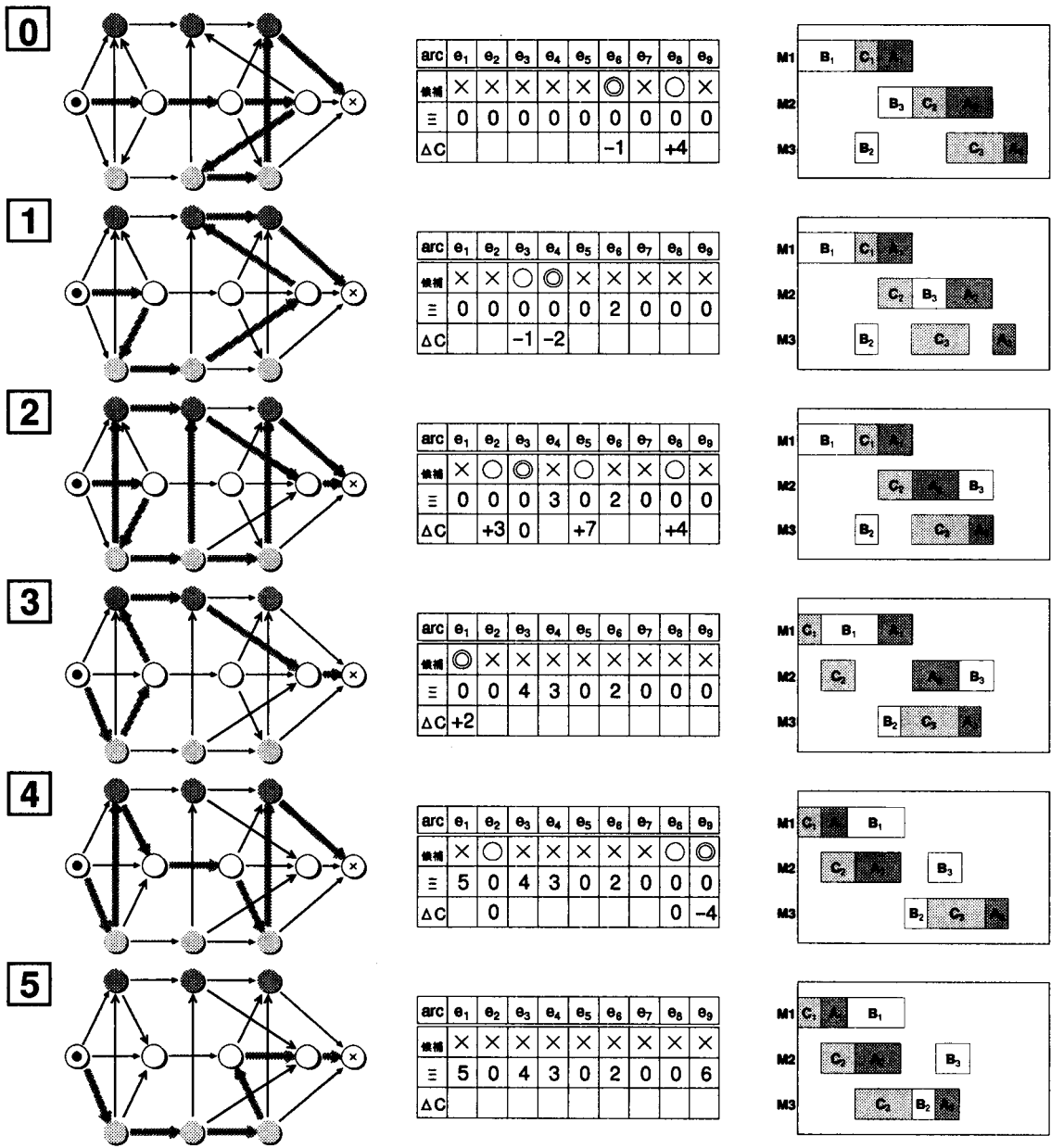


図3: ジョブショップ問題における Tabu Search の動き。左列は各イテレーションの離接グラフの状態とクリティカルパスを示している。中列は Tabu List の状態を示しており、反転が禁止されている離接弧の禁止期限(\equiv)と反転する離接弧の候補(反転が禁止されていないクリティカルパス上の離接弧)とその反転ともなう目的関数値の変化量(ΔC)を表している。離接弧の番号は図2に対応している。◎は実際に反転された離接弧、○は反転する候補となった離接弧(近傍の要素)である。右列は各イテレーションのスケジュールの状態をガントチャートで示している。目的関数値は20→19→17→17→19→15と変化する。イテレーション2ではクリティカルパスが複数存在する。イテレーション5終了時点で移行する解がなくなるが(クリティカルパス上の離接弧がすべて反転を禁止されている)、通常は Tabu List をクリアして探索を継続する。

Tabu Searchの挙動： 3×3問題を用いてTabu Searchの動きを実際に追ってみる。ここで用いた例題の数値データを表3に示す。この例題の各製品の機械への訪問順序は離接グラフ(図2)に対応している。

製品	処理時間			処理機械		
	1	2	3	1	2	3
A	3	4	2	1	2	3
B	5	2	3	1	3	2
C	2	3	5	1	2	3

図3はTabu Lengthの設定値を2としたときの探索開始から5イテレーション終了までの挙動を示している。Tabu Lengthが1の場合は、イテレーション2の e_3 の反転までは同様の挙動を示すが、イテレーション1で反転した e_4 をイテレーション3で再び反転し、 e_3, e_4 を交互に反転する巡回を起こす。

3.3 数値実験

パラメータに適正値を与えるための実験： Tabu Searchに限らず、多くのメタ戦略ではパラメータを必要とする。解法のチューニングではパラメータ値の適正な設定がきわめて重要である。主要なパラメータとしては、Tabu Length、終了条件を定めるパラメータ、ここでは述べなかったが「長期メモリ」にける重み等がある。図4はTabu Searchの最重要パラメータであるTabu Lengthの適正値を探るための実験結果である。ここでは、MuthとThompsonの10×10問題に対して、Tabu Length以外の条件を固定したときのTabu Lengthと得られた最良解との関係を示している。

図4からTabu Lengthが10前後で目的関数値が最小となっている。より大きなTabu Lengthを選択した場合でも解の悪化の度合は小さいが、クリティカルパス上のすべての離接弧が反転を禁止されてしまうことがあるため、適正範囲内で短めのTabu Lengthを選択する。

実験結果： 表4に筆者らの実験結果と他の近似解法の実験結果を示す。表4に掲載されている我々の実験結果は長期メモリなどの工夫を追加している。

なお、表4のApplegateとCookの実験結果(表内のAC)は彼らの作成したプログラム(C言語)[2]で再実験した。

遺伝的アルゴリズムを用いて系統的な数値実験を行った研究はほとんどみられず、MuthとThompsonの10×10問題を適用した山田と中野の研究[32]では10分程度の実行時間(SUN SPARCstation 2)を許した320

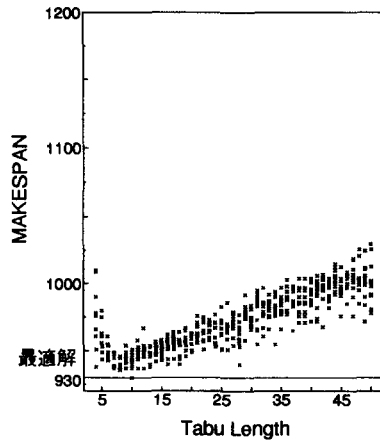


図4: Tabu Lengthと目的関数値の関係

回の試行で2回最適解を得ている。

4 おわりに

ごく一部の「理想的」問題を除いてスケジューリング問題はNP困難である。したがって、最適解を保証する厳密解法には限界があり、ある程度の規模の実際の問題を限られた時間の中で最適に準ずる形で解こうとすると、メタ解法とShifting Bottleneck系の解法が現実的な選択ということになる。中でも、Tabu SearchはSimulated Annealing法とともに古典的なLocal Searchの自然な変形であり、難しいとされるジョブシヨップ問題に対しても、これまでのところ一連のメタ解法の中でもっとも優れかつ安定した性能を誇っている解法といえる。

現実のスケジューリング問題は前述の一機械総納期遅れ最小化やジョブシヨップ問題のように単純ではなく、適用の場に固有な付加的条件を無視できない場合がほとんどである。Tabu Searchを始めとする多くのメタ解法の利点は、そうした現実的な付加的条件への対応が比較的容易にできることである。実際の問題解決にTabu Searchが有効かどうかの答えは、最終的にはこの方法を山積する現実問題の解決に適用して出されるものといえる。

謝辞

Applegate教授(Carnegie Mellon University)には、Shifting Bottleneck法とShuffle法のプログラムの提供を、また、Aarts博士(Philips Research Laboratories)には問題データの提供をいただいた。ここに感謝の意を表す。なお、本研究は一部文部省科学研究費一般研究C(05680344)、奨励研究A(06740145)の援助を受けている。

表 4: ベンチマーク問題に対する実験結果

問題	サイズ	最適解	Tabu Search				Simulated Annealing				その他			
			KTM		DT	Tai	LAL	YRN	MSS	ABZ	AC			
			C	t	C	C	C	t	C	C	C	t		
MT10	10 × 10	930	930.8	541.5	948.4	*930	933.4	930.8	2141	948	*930	952	17.5	
			*930		935		*930	*930				*930	*930	22.5
MT20	20 × 5	1165	1173.0	270.5	1166.8	—	1173.8	—	—	—	1178	1210	41.1	
			1173		*1165		*1165	—				—	1191	50.1
LA16	10 × 10	945	945.9	374.4	946.0	—	966.2	—	—	946	956	961	26.0	
			*945		*945		956	—				—	956	26.4
LA17	10 × 10	784	*784.0	30.9	*784.0	—	787.8	—	—	*784	787	*784	50.9	
			*784		*784		*784	—		—		*784	51.0	
LA18	10 × 10	848	*848.0	66.2	*848.0	—	861.2	—	—	*848	859	853	37.5	
			*848		*848		861	—		—		853	40.3	
LA19	10 × 10	842	844.9	104.9	846.6	—	853.4	—	—	*842	860	848	80.7	
			*842		*842		848	—		—		848	87.2	
LA20	10 × 10	902	*902.0	133.8	903.0	—	908.4	—	—	907	914	918	60.5	
			*902		*902		*902	—		—		911	62.9	
LA21	15 × 10	(1047) (1040)	1050.0	1720.2	1057.0	1047	1067.6	1054.6	4022	1071	1084	1072	99.0	
			1047		1048		1063	1050				1069	***	
LA22	15 × 10	927	931.0	671.9	936.6	—	944.2	—	—	*927	944	939	83.23	
			*927		933		938	—		—		933	165.7	
LA23	15 × 10	1032	*1032.0	5.0	*1032.0	—	*1032.0	—	—	*1032	*1032	*1032	62.3	
			*1032		*1032		*1032	—		—		*1032	62.4	
LA24	15 × 10	935	942.0	1170.4	943.8	—	966.6	947.2	2881	973	976	959	116.9	
			941		941		952	943				950	591.3	
LA25	15 × 10	977	980.5	1182.8	980.4	—	1004.4	987.8	4276	991	1017	1002	87.65	
			979		979		992	985				987	628.4	
LA26	20 × 10	1218	*1218.0	75.2	*1218.0	—	1219.0	—	—	*1218	1224	*1218	107.4	
			*1218		*1218		*1218	—		—		*1218	107.5	
LA27	20 × 10	(1240) (1235)	1247.3	919.8	1252.4	1240	1273.6	1266.0	7622	1274	1291	1269	235.3	
			1241		1242		1269	1262				1269	***	
LA28	20 × 10	1216	*1216.0	481.5	1216.8	—	1244.8	—	—	*1216	1250	1251	163.4	
			1216		*1216		1224	—		—		1223	***	
LA29	20 × 10	(1170) (1120)	1173.3	3042.9	1194.6	1170	1226.4	1193.2	9070	1196	1239	1217	149.7	
			1165		1182		1218	1188				1215	***	
LA30	20 × 10	1355	*1355.0	229.3	*1355.0	—	*1355.0	—	—	*1355	*1355	*1355	25.8	
			*1355		*1355		*1355	—		—		*1355	25.9	
LA36	15 × 15	1268	1279.3	3444.0	1289.4	—	1300.0	—	—	1292	1305	1314	191.9	
			1278		1278		1293	—		—		1314	192.2	
LA37	15 × 15	1397	1419.5	1638.0	1423.0	—	1442.4	—	—	1435	1423	1418	80.1	
			1412		1409		1433	—		—		1413	99.4	
LA38	15 × 15	(1202) (1184)	1210.5	3044.0	1210.0	1202	1227.2	1214.0	7439	1231	1255	1257	248.8	
			1202		1203		1215	1209				1233	263.8	
LA39	15 × 15	1233	1240.7	3279.1	1254.8	—	1258.2	—	—	1251	1273	1279	188.7	
			1239		1242		1248	—		—		1254	223.1	
LA40	15 × 15	1222	1233.7	6692.8	1235.4	—	1247.4	1248.8	5365	1235	1269	1246	150.7	
			1228		1233		1234	1235				1246	150.8	

KTM 筆者らによる Tabu Search, DT Dell'Amico と Trubian による Tabu Search [9], Tai Taillard による Tabu Search(最良値のみ) [25], LAL Laarhoven らによる Simulated Annealing [28], YRN Yamada らによる Simulated Annealing [31], MSS Matsuo らによる Controlled Search Simulated Annealing(最良値, 結果のみ [28]), ABZ Adams らによる Shifting Bottleneck 法 [1], AC:Applegate と Cook による Shifting Bottleneck 法(上段), Shuffle 法(下段) [3](LA27~29 は 3600 秒間の実行での暫定解) C:目的関数値(上段:平均値, 下段:最良値, AC を除く), t :実行時間(SUN SPARCstation 2, KTM と YRN は平均), * :最適値, — :結果未掲載

最適解の欄の括弧内の数字で上段は上界値, 下段は下界値を示している. MT は Muth と Thompson の問題, LA は Lawrence の問題を表す.

参考文献

- [1] J. Adams, E. Balas, and D. Zawack. The shifting bottleneck procedure for job shop scheduling problem. *Mgmt. Sci.*, 34:391-401, 1988.
- [2] D. Applegate. private communication, 1993.
- [3] D. Applegate and W. Cook. A computational study of the jobshop scheduling problem. *ORSA J. Comp.*, 3:149-156, 1991.
- [4] E. Balas. Machine sequencing via disjunctive graphs: an implicit enumeration algorithm. *Opns. Res.*, 17:941-957, 1969.
- [5] J. W. Barnes and M. Laguna. A tabu search experience in production scheduling. *Ann. Opns. Res.*, 41:141-156, 1993.
- [6] J. E. Beasley. OR-Library: distributing test problems by electronic mail. *J. Opnl. Res. Soc.*, 41:1069-1072, 1990.
- [7] P. Brandimarte. Routing and scheduling in a flexible job shop by tabu search. *Ann. Opns. Res.*, 41:157-183, 1993.
- [8] J. Carlier and E. Pinson. An algorithm for job shop scheduling problem. *Mgmt. Sci.*, 35:164-176, 1989.
- [9] M. Dell'Amico and M. Trubian. Applying tabu search to the job-shop scheduling problem. *Ann. Opns. Res.*, 41:231-252, 1993.
- [10] J. Du and J. Y. -T. Leung. Minimizing total tardiness on one machine is \mathcal{NP} -hard. *Math. Opns. Res.*, 15:483-495, 1990.
- [11] F. Glover. Tabu search I. *ORSA J. Comput.*, 1:190-206, 1989.
- [12] F. Glover. Tabu search II. *ORSA J. Comput.*, 2:4-32, 1990.
- [13] F. Glover. Tabu search. A chapter in *Modern Heuristic Techniques for Combinatorial Problems*, 1994,(to appear).
- [14] F. Glover, E. Taillard, and D. de Werra. A user's guide to tabu search. *Ann. Opns. Res.*, 41:3-28, 1993.
- [15] R. L. Graham, E. L. Lawler, J. K. Lenstra, and A. H. G. Rinnoy Kan. Optimization and approximation in deterministic sequencing and scheduling theory: a survey. *Ann. Dis. Math.*, 5:287-326, 1979.
- [16] M. J. Krone and K. Steiglitz. Heuristic Programming Solution of a Flowshop Scheduling Problem. *Opns. Res.*, 629-638, 1973.
- [17] M. Kubo, K. Fujisawa, and S. Morito. *The Life Span Method - A New Variant of Local Search -*. Technical Report 1, Tokyo University of Mercantile Marine, April 1993. The Institute of Statistics and Mathematics Cooperative Research Report 53 *Optimization - Modeling and Algorithm*, Feb., 1994, pp. 85-104.
- [18] M. Laguna, J. W. Barnes, and F. Glover. Tabu search methods for a single machine scheduling problem. *J. Intelligent Mfg.*, 2:63-74, 1991.
- [19] M. Laguna and F. Glover. Tabu search methods for a single machine scheduling problem. *Exp. Sys. Appl.*, 6:287-297, 1993.
- [20] M. Laguna and J. L. Gonzalez Velarde. A search heuristic for just-in-time scheduling in parallel machines. *J. Intelligent Mfg.*, 2:253-260, 1991.
- [21] E. L. Lawler, J. K. Lenstra, and A. H. G. Rinnoy Kan. Recent developments in deterministic sequencing and scheduling theory: a survey. *Deterministic and Stochastic Scheduling*, edited by Dempster E. L. Lawler, J. K. Lenstra, and A. H. G. Rinnoy Kan. Reidel Publishing Co., 1982.
- [22] J. F. Muth and G. L. Thompson. *Industrial Scheduling*. Prentice-Hall, 1963.
- [23] C. H. Papadimitriou and K. Steiglitz. *Combinatorial Optimization: Algorithms and Complexity*. Prentice-Hall, 1982.
- [24] S. S. Panwalkar and W. Iskander. A survey of scheduling rules. *Opns. Res.*, 23:45-61, 1977.
- [25] E. Taillard. Parallel taboo search for the job shop scheduling problem. Working paper ORWP 89/11 Département de Mathématiques, Ecole Polytechniques Fédérale De Lausanne, Switzerland, 1989.
- [26] E. Taillard. Some efficient heuristic methods for the flow shop sequencing problem. *European J. Opnl. Res.*, 47:65-79, 1990.
- [27] 高山 裕志, 久保 幹雄, 森戸 晋. Tabu Search のジョブショップスケジューリング問題への適用と実験的解析. 日本機械学会, 生産スケジューリングシンポジウム講演論文集, 152-157, 1993.
- [28] P. J. M. van Laarhorven, E. H. L. Aarts, and J. K. Lenstra. Job shop scheduling by simulated annealing. *Opns. Res.*, 40:113-125, 1992.
- [29] M. Widmer. Job shop scheduling with tooling constraints: a tabu search approach. *J. Opnl. Res. Soc.*, 42:75-82, 1991.
- [30] M. Widmer and A. Hertz. A new method for the flow shop sequencing problem. *European J. Opnl. Res.*, 41:186-193, 1989.
- [31] 山田 武士, B. E. Rosen, 中野良平. クリティカルブロックシミュレーション・アイデアによるジョブショップスケジューリング問題の解法. 電気学会論文誌 C(電子・情報・システム部門誌), 114-C:476-482, 1994.
- [32] 山田 武士, 中野良平. 遺伝的アルゴリズムによるジョブショップ問題の解法. 情報処理学会資料, 人工知能 81-8, 1-10, 1992.