岡山理科大学紀要 第35号 A pp. 123-132 (1999)

ジョブショップスケジューリング問題に対する反復局所探索法について

若宮 利治　　片山 謙吾*　　成久 洋之*
岡山理科大学大学院工学研究科修士課程情報工学専攻
*岡山理科大学工学部情報工学科

(1999年11月4日 受理)

1. まえがき

近年，システムの大規模化に伴い発見的手法の重要性が増している．そのような手法の中で，組合せ最適化問題に対しては局所探索法 (Local Search, LS) をベースにした手法がよく用いられ，実用的にも成果をあげている．

本論文では，組合せ最適化問題の中でも特に難しい問題の一つとされているジョブショップスケジューリング問題 (JSP) を取り上げ，この問題に対して LS の拡張である反復局所探索法 (Iterated Local Search, ILS)[4] を適用し，効率性について検討を行う．

2. ジョブショップスケジューリング問題

スケジューリングとは，ある目的を達成するため共通に使われる資源の時間配分を決定することである．JSP は，複数の異なる仕事を処理するために，機械群の時間的な割り当てを決定する問題であり，n 個の仕事を m 台の機械で処理することを考える ($n \times m$ JSP)．各仕事を処理する機械の順序 (技術的順序) は仕事ごとにあらかじめ与えられている．各機械上での仕事の処理を作業と呼ぶ．各作業は与えられた処理時間をかけて各機械上で中断なく処理される．ここで各機械はすべて異なり，同時に二つ以上の作業を処理することができないとする．すべての仕事を完成させるまでの時間を総作業時間 (makespan) と呼び，各仕事の技術的順序と，各作業の処理時間が与えられたもとで，makespan を最小にするような各機械上での仕事の処理順序をすべて決定するものである[3]．

JSP の解の表現方法として離接グラフ (Disjunctive Graph) がよく用いられる．JSP は離接グラフ $G = (V, C \cup D)$ を用いて次のように表現される．

- V は節点の集合であり，作業に対応する節点および二つの特殊な節点ソース (0) とシンク (*) からなる．
- C は節点間を結ぶ有向弧 (Conjuctive Arc) の集合で技術的順序を表す．
- D は離接弧 (Disjunctive Arc) の集合で，同一機械上の作業の対を表す．

各作業の処理時間は対応する節点に付与された重みによって表す．JSP の 3 × 3 問題の離接グラフを図 1 に示す．図 1 において円は節点，すなわち作業を表し，実線は有向弧，破線は離接弧を表す．なお，O_{ij} は機械 j での仕事 i の作業を意味し，P_{ij} は O_{ij} の処理時間を意味する．図 1 において，節点 O_{11} は機械 1 で処理される仕事 1 の作業を表し，以下同様に考える．仕事 1 は機械 1，機械 2，機械 3 の順番にスケジュールされることを意味し，仕事 2 は機械 1，機械 3，機械 2 の順番にスケジュールされ，仕事 3 は機械 2，機械 1，機械 3 の順番にスケジュールされることを意味する．

完全なスケジュールは同一機械上の作業の処理順序をすべて決定することによって得られる．離接グラフモデルにおいて，このことは D のすべての無向弧を有向弧に変えることに対応する．このとき D より得られた有向弧の集合を選択とよぶ．ある選択 S が実行可能スケジュールを表現していることと，有向グラフ $G(S) = (V, C \cup S)$ が閉路を持たないことは同値である．このとき S は完全選択とよばれる．

一つの完全選択から対応するスケジュールを生成する際，各作業を可能な限り早く加工して一意に得られるスケジュールをセミアクティブスケジュールとよぶ．完全選択と対応するセミアクティブスケジュールは同一視できるため，区別せず同一の記号 S を用いて表す．S の総作業時間 $L(S)$ は (0) から*に至る最も長い重みつきパスの長さによって与えられる．節点 (0) から (*) に至る最も長い重みつきの長さが makespan になる．

3. 反復局所探索法について

一般に，近似解法の研究においては多種多様な手法が試みられている．そのような近似解法における基本的な戦略は，LS がベースに置かれている．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．近い解の集合を近傍と呼び，実行可能解 x の近傍を $N(x)$ とする．本論文では，LS または Tabu Search(TS)[2] をそれぞれ組み込んだ二つの ILS (ILS, ILS+TS) および MSLS(MSLS, MSLS+TS) について比較検討する．なお，LS

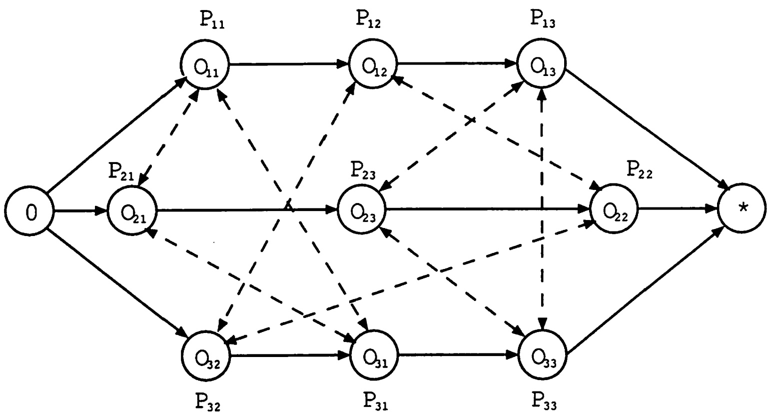

図 1　3 × 3 問題の離接グラフ

および TS で用いる近傍は，離接グラフ上の離接弧を 1 対反転したスケジュールの集合とする．以下，各々の探索法について簡単に説明する．

3.1 LS

与えられた解 x に対して，ある近傍操作を加えることでその近傍 $N(x)$ から新しい解 x' を生成し，その生成された解 x' が与えられた解 x よりも良い評価値を有すれば，その解 x' を x とみなし，再び近傍操作を施すことで近傍 $N(x)$ から新しい解を生成および評価する改善処理を繰り返すものである．この LS によって最終的に得られる解 x は，$N(x)$ の中に改善解が存在しなくなった時とされ，この x は近傍 $N(x)$ のもとで局所的に最適な解 (局所解) となる．つまりこの局所解の質は，LS で使用される近傍操作に大きく依存するものの，そこで使用される近傍操作ではこれ以上に評価値の良い解は存在しないことを意味する．このように LS の考え方は単純であり実装も容易であるが，一方で局所解に陥って抜け出せなくなってしまうという欠点がある．

3.2 TS

TS は禁断探索法と呼ばれ，組合せ最適化問題の近似解を求めるために設計されるメタヒューリスティック解法である．TS は LS と同様に与えられた解に近傍操作を施し，その近傍から新しい解を生成および評価する改善処理を繰り返すことによって近似解を得るが，以下の点で異なる．

- TS 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，良好な解を得るために目的関数値を増加させるような近傍への移動を行う．
- 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，その情報をもとに探索を行う．

移動に伴う情報を属性と呼び，属性を記憶するリストを Tabu List(禁断リスト) と呼ぶ．TS においては，禁断リストを考慮に入れながら最良の方向に探索を進めるが，禁断リストが多くなるにつれ，探索する場所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構造にする．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．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ているものに，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があげられる．

本研究での TS は，上述した LS と同様の近傍を用いて探索を行う．TS は，近傍解の生成に伴う情報を記憶する Tabu List を使用する．この Tabu List によって，以前探索された解へ戻ることを禁止することができる．本実験では，高山らの観測[2] を参考にし，Tabu Length を 10，終了までのイテレーションの回数を 1000 回とする．

3.3 MSLS

上述した LS によって得られる解はしばしば満足のいく上質な解ではないことがある．それは最適解の算出を保証する方法ではないため，最適解から離れ過ぎる解を算出してしまう傾向を否定できないからである．そこで，この LS を用いてより良好な解を得るための最も簡単な方法としてはこの LS を許容される時間内，繰り返し実行する方法がある．

この方法は極めて簡潔でありながら，比較的満足のいく解が得られ，実用上有効なアプローチの一つとされている．一般にこの方法はランダム多点局所探索法 (Random Multi-start Local Search, MSLS)[1)] と呼ばれる．

MSLS はランダムな解から探索が行われ，局所解を得たならば再びランダムな解を生成し，LS を繰り返し実行する方法で，ある与えられた時間または繰り返し回数に到達したときを処理の終了条件とし，最終的に複数回実行された LS で得られた解の中で最良な解を算出するものである．MSLS のアルゴリズムを以下に示す．

<u>MSLS アルゴリズム</u>

1. ランダムな解 x を生成する
2. 1 で発生した解 x に LS(または TS) を実施する
3. 2 で算出された解の評価値 $cost(x)$ を評価する
4. 1〜3 までの処理を与えられた時間または繰り返し回数に到達するまで繰り返す
5. 複数回実行された LS(または TS) の中で得られた最良の解を出力する

3.4 ILS

比較的良好な解の近くには，更により良い解があるという経験から MSLS のようにランダムな解を用いて再び局所解を探索するのではなく，得られた局所解から LS で使用される近傍操作とは異なる近傍操作を利用して局所解からの脱出を図り，再び探索を繰り返し行うものである．ILS のアルゴリズムを以下に示す．

<u>ILS アルゴリズム</u>

1. ランダムな解 x を生成する
2. 1 で発生した解 x に LS(または TS) を実施する
3. 2 で算出された解 x' に近傍操作 N1(または N2) を施す
4. 3 で得られた解 y に対して再び LS または TS を実施する
5. 4 で算出された解 y' が 2 で算出された解 x' よりも良い評価値を有すれば (つまり，$cost(y') < cost(x')$)，その解 y' を x' とみなす
6. 3〜5 までの処理を与えられた時間または繰り返し回数に到達するまで繰り返す
7. 複数回実行された LS(または TS) で得られた最良の解を出力する

本研究では，局所解から脱出を図るために 2 つの異なる近傍操作を考える．なお，異なる近傍操作は，現在までに算出された最良解に対して与えるものとする．ここで，異なる近傍操作とは，機械上で割り当てられる仕事をランダムに二つ選び，それらを入れ替える操作 ($N1$) と，割り当てられる仕事の区間をランダムに決め，その区間内の仕事を入れ替える操作 ($N2$) とする．$N1$ と $N2$ の近傍操作の例を図 2，図 3 に示す．

M1:	$\boxed{J1}$	J4	J3	J6	$\boxed{J2}$	J5		M1:	$\boxed{J2}$	J4	J3	J6	$\boxed{J1}$	J5
M2:	J2	$\boxed{J4}$	J6	J1	$\boxed{J5}$	J3		M2:	J2	$\boxed{J5}$	J6	J1	$\boxed{J4}$	J3
M3:	J3	J1	$\boxed{J2}$	J4	J5	$\boxed{J6}$	→	M3:	J3	J1	$\boxed{J6}$	J4	J5	$\boxed{J2}$
M4:	$\boxed{J3}$	J6	J4	J1	$\boxed{J2}$	J5		M4:	$\boxed{J2}$	J6	J4	J1	$\boxed{J3}$	J5
M5:	J2	$\boxed{J5}$	J3	J4	J6	$\boxed{J1}$		M5:	J2	$\boxed{J1}$	J3	J4	J6	$\boxed{J5}$
M6:	J3	J6	$\boxed{J2}$	$\boxed{J5}$	J1	J4		M6:	J3	J6	$\boxed{J5}$	$\boxed{J2}$	J1	J4

図 2　6 × 6 問題における $N1$ の近傍操作の適用例

M1:	J1	J4	J3	J6	J2	J5		M1:	J2	J6	J3	J4	J1	J5
M2:	J2	J4	J6	J1	J5	J3		M2:	J2	J5	J1	J6	J4	J3
M3:	J3	J1	J2	J4	J5	J6	→	M3:	J3	J1	J6	J5	J4	J2
M4:	J3	J6	J4	J1	J2	J5		M4:	J2	J1	J4	J6	J3	J5
M5:	J2	J5	J3	J4	J6	J1		M5:	J2	J1	J3	J4	J6	J5
M6:	J3	J6	J2	J5	J1	J4		M6:	J3	J6	J5	J2	J1	J4

図 3　6 × 6 問題における $N2$ の近傍操作の適用例

4. 実験結果

本論文では，6×6，10×10 および 20×5 のベンチマーク問題に対して，LS または TS をそれぞれ利用した MSLS 法および ILS 法について比較検討を行う．また，異なる近傍操作の有効性を見るために，$N1$，$N2$ の操作を，それぞれ 1 回，5 回，10 回行うものとし，近傍操作を行う確率を $N1$, $N2$ ともに 1.0 とする．さらに 10×10 問題に関して，Tabu Length を 5 から 50 まで 5 ずつ変化させたものに対する makespan の関係について検討を行う．

本実験で使用する計算機は，FUJITSU S-4/20H であり，計算打ち切り時間は，ILS においては 6×6 問題が 15 秒，10×10 問題が 500 秒，20×5 問題が 1000 秒とする．各探索法の試行回数は 5 回行う．また各問題例の最適解は，6×6 問題が 55，10×10 問題が 930，20×5 問題が 1165 である．使用言語は C 言語である．

表 1，表 2，表 3 は各ベンチマーク問題において MSLS と MSLS+TS，ILS と ILS+TS の 4 つの探索法に対する makespan の最良値と平均値および平均値の解の質を示したものである．図 4 は，10×10 問題において MSLS+TS の Tabu Length を 5 から 50 まで 5 ずつ変化させたものと makespan の関係を示したもので，各々の試行回数が 5 回のときのものを表している．

ここで，

$$\text{解の質} = \frac{\text{得られた解} - \text{最適解}}{\text{最適解}} \times 100\ (\%)$$

で求めるものとした．なお，計算打ち切り時間が本実験で設定した値よりも長ければ，本実験で得られた解よりもさらに良質な解が得られるものと考えられる．

表 1 MSLS と MSLS+TS の実験結果

	MSLS			MSLS+TS		
問題例	best	avg.	解の質	best	avg.	解の質
6×6	57	59.4	0.08	55	55	0.00
10×10	1094	1118.8	0.20	945	954.2	0.03
20×5	1370	1384.4	0.16	1196	1206.8	0.04

表 2 $N1$ を用いた ILS と ILS+TS の実験結果

1 回	ILS			ILS + TS		
問題例	best	avg.	解の質	best	avg.	解の質
6×6	55	58.4	0.06	55	55	0.00
10×10	1095	1110.6	0.19	950	962.8	0.04
20×5	1385	1411.2	0.21	1211	1223.8	0.05
5 回	ILS			ILS + TS		
問題例	best	avg.	解の質	best	avg.	解の質
6×6	56	58.2	0.06	55	55	0.00
10×10	1117	1126.4	0.21	937	951.2	0.02
20×5	1388	1400.2	0.20	1198	1213.8	0.04
10 回	ILS			ILS + TS		
問題例	best	avg.	解の質	best	avg.	解の質
6×6	58	58.6	0.07	55	55	0.00
10×10	1109	1119.8	0.20	961	968.8	0.04
20×5	1373	1401.2	0.20	1211	1218.8	0.06

表 3 $N2$ を用いた ILS と ILS+TS の実験結果

1 回	ILS			ILS + TS		
問題例	best	avg.	解の質	best	avg.	解の質
6×6	58	59.4	0.08	55	55	0.00
10×10	1111	1118.6	0.20	949	960.8	0.03
20×5	1416	1426.2	0.22	1201	1207.0	0.04
5 回	ILS			ILS + TS		
問題例	best	avg.	解の質	best	avg.	解の質
6×6	58	59.0	0.07	55	55	0.00
10×10	1110	1120.4	0.20	957	959.6	0.03
20×5	1368	1407.4	0.21	1210	1219.6	0.05
10 回	ILS			ILS + TS		
問題例	best	avg.	解の質	best	avg.	解の質
6×6	55	57.4	0.04	55	55	0.00
10×10	1084	1108.6	0.19	958	968.4	0.04
20×5	1391	1422.8	0.22	1198	1215.4	0.04

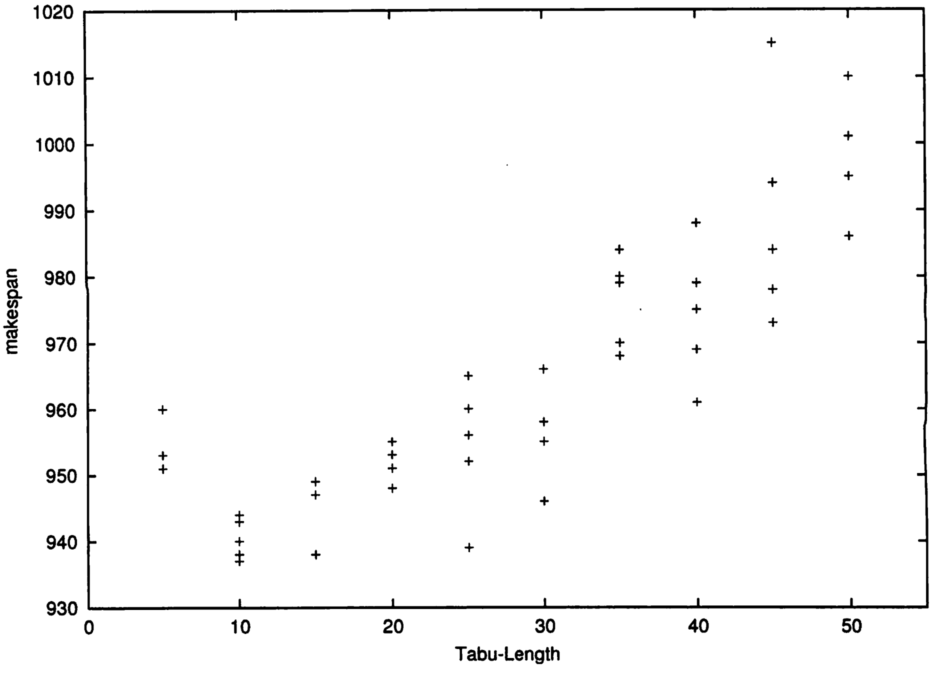

図 4　MSLS+TS の makespan と Tabu Length の関係

5. 考察

表 1〜3 より MSLS のほうが ILS よりもわずかに良質な解を得ているが，それほど解の質に大きな差が生じないことがわかる．これは，JSP において 2 つの探索法の形質が似ているため，解の質に大きな差が生じるほど影響を及ぼさないものであるといえる．表 2，表 3 より ILS+TS に関して，近傍操作が $N1$ のときは，近傍操作の回数が 5 回の場合に比較的解の質が良いことがわかり，近傍操作が $N2$ のときは，近傍操作が 1 回の場合が比較的良質な解が得られることがわかる．同様に ILS に関して，近傍操作が $N1$ のときは近傍操作の回数が 10 回の場合に比較的良質な解を得ており，近傍操作が $N2$ のときは近傍操作の回数が 10 回の場合に比較的良質な解を得ていることがわかる．このことから近傍操作の回数によって解の質には影響を及ぼすが，回数を増やせば必ずしも良質な解が得られるわけではないことがいえる．これは探索法や問題の大きさによって，良質な解を得る近傍操作やその回数が違ってくるものであると考えられる．

図 5〜10 は，本研究で得られた結果で，10 × 10 問題に ILS+TS を適用した場合において，2 つの異なる近傍操作の回数がそれぞれ 1 回，5 回，10 回のときの Tabu Length を 5 から 50 まで 5 ずつ変化させたものに対する makespan の関係を示したもので，各々の試行回数が 5 回のときのものを表しており，図 5〜10 の内容は，

- 図 5 は，近傍操作 $N1$ の回数が 1 回の場合の Tabu Length と makespan の関係
- 図 6 は，近傍操作 $N2$ の回数が 1 回の場合の Tabu Length と makespan の関係
- 図 7 は，近傍操作 $N1$ の回数が 5 回の場合の Tabu Length と makespan の関係
- 図 8 は，近傍操作 $N2$ の回数が 5 回の場合の Tabu Length と makespan の関係
- 図 9 は，近傍操作 $N1$ の回数が 10 回の場合の Tabu Length と makespan の関係
- 図 10 は，近傍操作 $N2$ の回数が 10 回の場合の Tabu Length と makespan の関係

を示したものである．

図 4〜図 10 より，ILS+TS のほうが MSLS+TS よりも Tabu Length を変化させたときに解の質の差が小さいことがわかる．これは，MSLS がランダムに解を繰り返し生成することから解の質の差にばらつきが生じることが考えられる．また，近傍操作や近傍操作の回数を変化させると解の質に影響を及ぼすが，解の質の差にそれほど大きな差が生じないことがわかった．これは，異なる二つの近傍操作の性質が，JSP に関してそれほど解の質の差が生じるほどの影響を及ぼさないものであるといえる．

6. むすび

本論文では，JSP を取り上げ，MSLS および ILS に LS または TS を組み込んだ探索法を適用し，比較検討した．その結果から，TS の方が LS よりも有効な探索法であることが確認された．また，$N1$ が $N2$ よりも比較的良質な解を得ることが観測され，異なる近傍操作を行う回数の違いによっても算出される解の質に影響を及ぼすことがわかった．しかし，良質な解が得られるかどうかは近傍操作の回数が多い少ないにかかわらず，問題の大きさや探索法および Tabu Length により影響を及ぼしていくものだと考えられる．Tabu Length については 10 くらいが最良な解が得られることが確認でき，Tabu Length の長さが 15 から増えていくと解の質が悪くなる傾向がみられる．これは Tabu Length をより大きくすると離接弧の反転をすべて禁止してしまうことがあるので，良質な解が得られる可能性が低くなるものであると思われる．今後は，他の探索法について検討する予定である．

参考文献

1) E.H.L Arts, P.J.M van Laarhoven, J.K.Lenstra, N.L.J.Ulder; A Computational Study of Local Search Algorithm for Job Shop Scheduling, ORSA Journal on Computing Vol.6, No.2, pp.118-125, 1994.

2) 髙山 裕志，久保 幹雄，森戸 晋; スケジューリングと Tabu Search, オペレーションズ・リサーチ，pp.47–54, 1995.

3) 山田 武士，中野 良平; 遺伝的局所探索法によるジョブショップスケジューリング問題の解法，情報処理学会誌，Vol.38，No.6，pp.1126–1138，1997.

4) 若宮，片山，成久; ジョブショップスケジューリング問題に対する反復局所探索法について，電気・情報関連学会第 50 回中国支部連合大会 講演論文集，1999.

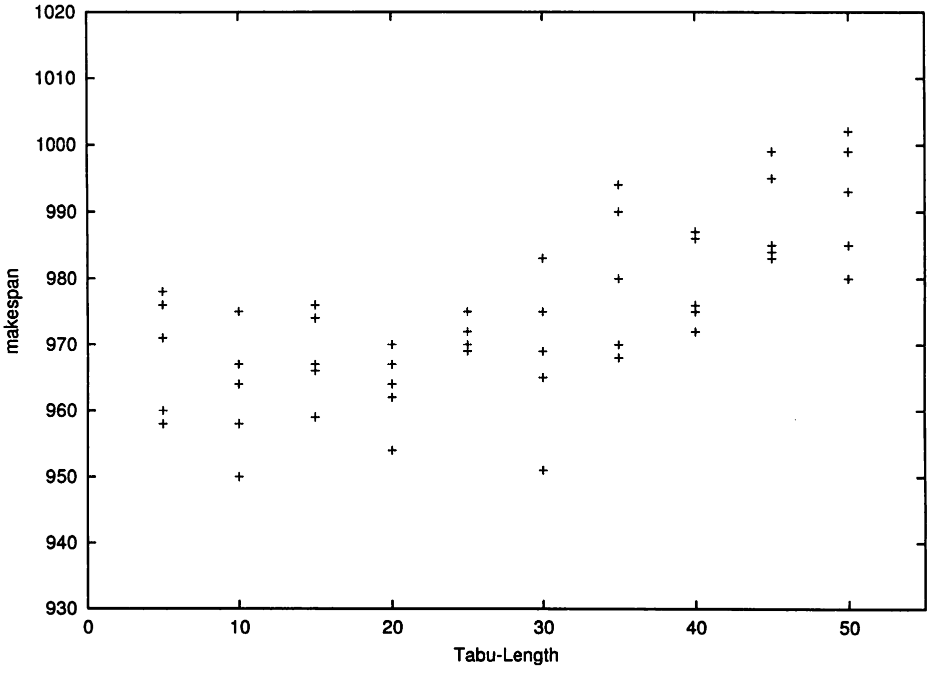

図 5　ILS+TS の $N1$ が 1 回の makespan と Tabu Length の関係

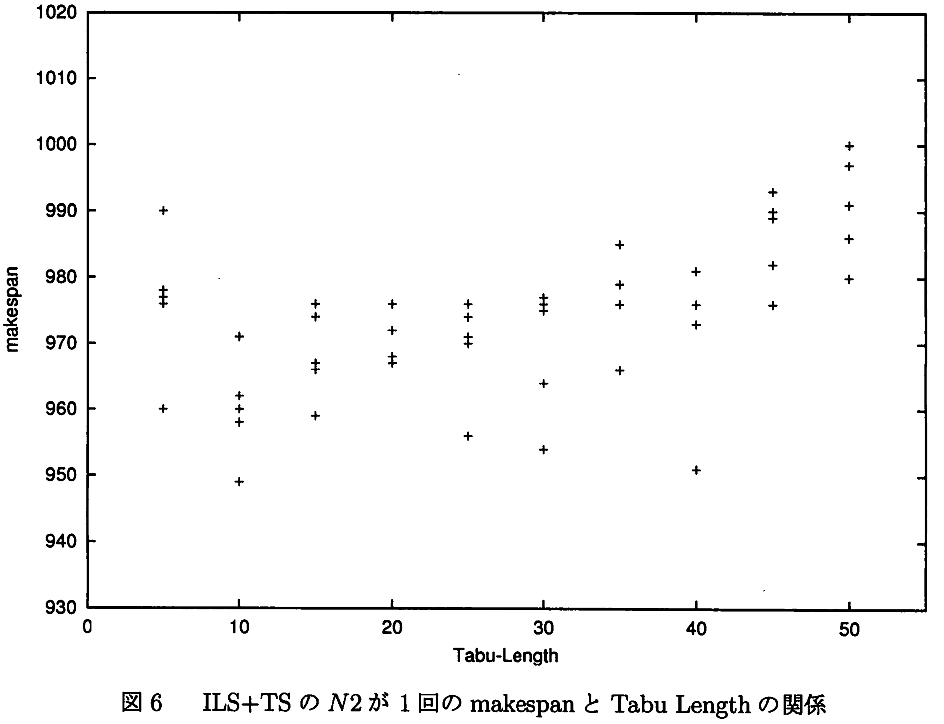

図 6　ILS+TS の $N2$ が 1 回の makespan と Tabu Length の関係

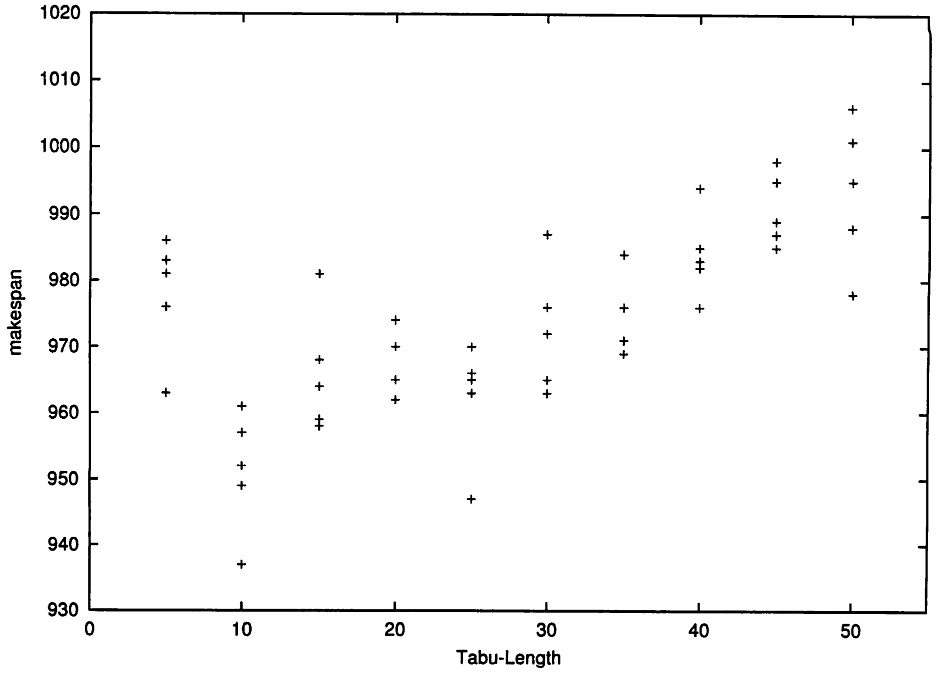

図 7　ILS+TS の $N1$ が 5 回の makespan と Tabu Length の関係

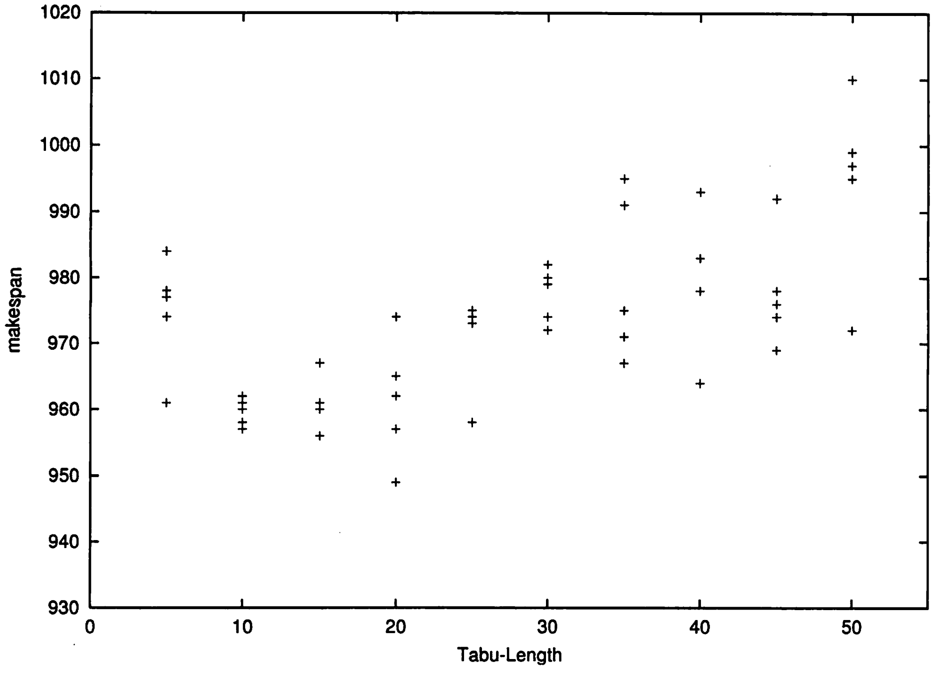

図 8　ILS+TS の $N2$ が 5 回の makespan と Tabu Length の関係

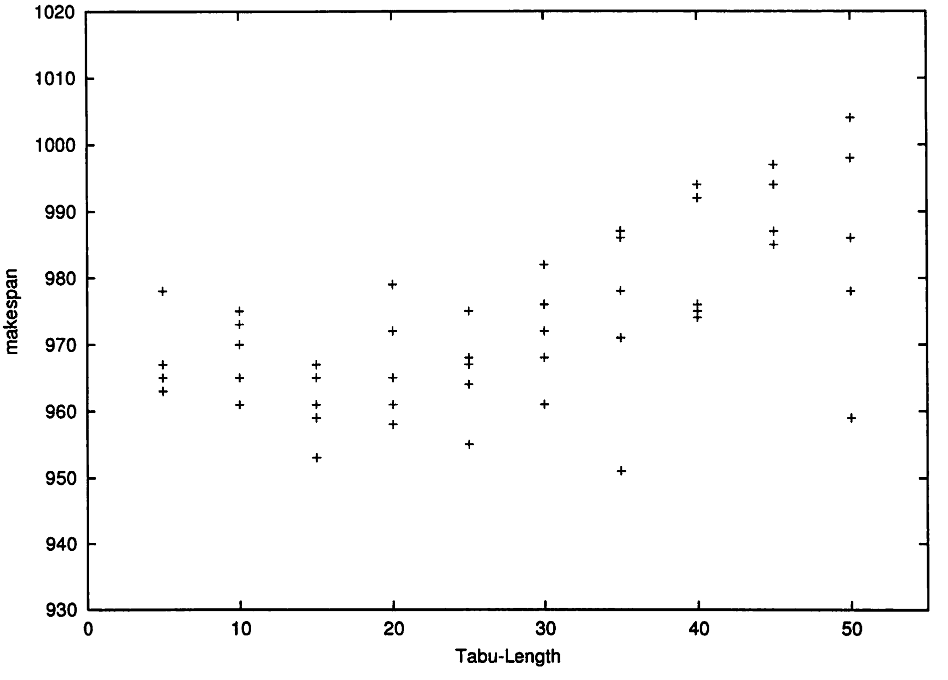

図 9　ILS+TS の $N1$ が 10 回の makespan と Tabu Length の関係

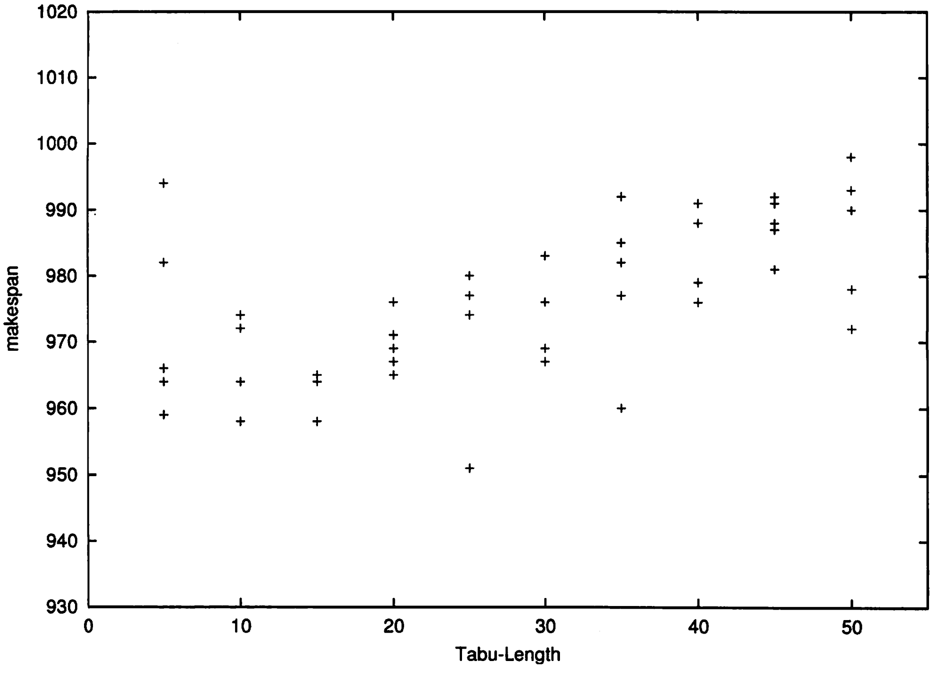

図 10　ILS+TS の $N2$ が 10 回の makespan と Tabu Length の関係

Some Comments on Iterated Local Searches for Job Shop Scheduling Problem

Toshiharu WAKAMIYA, Kengo KATAYAMA* and Hiroyuki NARIHISA*

Graduate School of Engineering

**Depertment of Information & Computer Engineering*

Okayama University of Science,

Ridai-cho 1-1, Okayama 700-0005, Japan

(Received November 4, 1999)

Almost all combinatorial optimization problems are NP-hard problems; it is impossible to obtain an exact optimal solution within polynomial time. However, heuristic algorithms are able to find near optimal solutions within reasonable time. Heuristic algorithms such as iterated local searches, simulated annealing, tabu searches and genetic algorithms are known to be able to find much better solutions than standard or traditional heuristics.

This paper presents iterated local searches for solving the job-shop scheduling problem. Generally the iterated local search consists of a local search phase and mutation phase. For both phases of the iterated local search, we investigate different suitable operations for several benchmark instances of the job-shop scheduling problem. Our experimental results, show that the search abilities of the iterated local searches depend on these operations.