商学論集　第67巻第1号　　1998年7月

【論　文】

探索空間平滑化つき近傍探索法による非一様型並列機械スケジューリング

董　　　彦　文

［概　要］本研究では，非一様型並列機械スケジューリング問題を対象として，探索空間平滑化つき近傍探索法を用いて，新たな近似解法を提案するとともに，従来の発見的解法や最適化解法と比較しながら提案解法の有効性を明らかにし，かつ提案解法の適用に必要なパラメータの選定についても考究する。

1. は じ め に

スケジューリングとは，与えられた仕事（ジョブ）をいつ，どの機械（生産機械，作業員など）に割当てるかを計画することである。スケジュールの評価基準，守るべき制約条件などによりスケジューリング問題はいろいろなタイプに分類されている。この中で，m ($m>1$) 台の機械と n 個のジョブが与えられ，各ジョブが任意の1台機械で加工できる場合のスケジューリング問題は，並列機械スケジューリング問題と呼ぶ。さらに，並列機械スケジューリング問題は等価機械型，一様機械型と非一様機械型に大別される。ここで，等価機械型は同じ機械がm台あり，各ジョブがどの機械でも同じ時間で加工できる場合である。一様機械型は，各ジョブの加工時間は機械ごとに異なるが，その比はジョブによらず一定であり，つまりジョブの加工時間が機械のみに依存する場合である。また，非一様機械型は，各ジョブの加工時間は機械ごとに異なり，かつその比も一定でなく，つまりジョブの加工時間が機械とジョブとの両方に依存する場合である。

総所要時間の最小化を目標関数とする非一様型並列機械スケジューリング問題は，並列処理システムのコンピュータ運用計画，複合工作機械を配置したFMS（フレキシブル生産システム）のスケジューリング，組立ロボットの動作計画などに幅広く応用できることがよく知られている。しかしながら，この問題はNP困難であり，問題の規模が大きくなると組合せ的爆発が起こり，実時間内に最適解を得ることは一般に不可能である。この非一様型並列機械スケジューリング問題に関しては，これまでにIbarraら〔1〕とDavisら〔2〕が発見的近似解法を，Lenstraら〔3〕が線形計画法に基づく近似解法を，野村ら〔4〕が分岐限定法に基づく最適化解法を提案しているものの，他のタイプのスケジューリング問題に比べてまだ研究が少なく，不十分なところが多い。

そこで，本研究では，非一様型並列機械スケジューリング問題を対象として，探索空間平滑化つき近傍探索法を用いて，新たな近似解法を提案するとともに，従来の発見的解法や最適化解法と比較しながら提案解法の有効性を明らかにし，かつ提案解法の適用に必要なパラメータの選定につい

ても考究する。

2. 問題の定義

本研究は，次の非一様型並列機械スケジューリング問題を対象とする。

(1) n 個のジョブ$\{J_i;\ i=1, 2, \dots, n, n>1\}$を，性能の異なる m 台の機械$\{M_j;\ j=1, 2, \dots, m, m>1\}$で加工する。

(2) 各ジョブは任意の1台の機械だけで加工できるが，加工時間は使用機械とジョブとの両方に依存し，機械 M_j でジョブ J_i を加工する場合の加工時間を t_{ij} とする。

(3) ジョブの加工時間は，正味加工時間と段取り時間を含む既知の数値であり，ジョブの加工順序によって変わらない。

(4) 各ジョブ間には先行関係がなく，任意の順序で加工できる。ただし，1つのジョブを分割して加工したり，1台の機械で2個以上のジョブを同時に加工したりすることはできない。

(5) 計画期間の初めに，n 個のジョブは全部着手可能である。また，計画期間にわたって，全部の機械が常に使用可能である。

(6) 目的関数は総所要時間の最小化とする。

この問題は次のとおりに定式化することができる。

問題P

Minimize $C=\mathrm{Max}\ \{C_j;\ j=1, 2, \dots, m\}$　　(1)

Subject to

$$C_j=\sum_{i=1}^{n} x_{ij}t_{ij} \quad (2)$$

$$\sum_{i=1}^{n} x_{ij}=1; \quad (3)$$

$$x_{ij}=\begin{cases}1; & M_j \text{ で } J_i \text{ を加工するとき}\\ 0; & \text{その他}\end{cases} \quad (4)$$

$i=1, 2, \dots, n;\ j=1, 2, \dots, m.$

3. 近傍探索法と探索空間の平滑化

3.1 単純近傍探索法

近傍探索法（Local Search，以下，単純近傍探索法またはLS法と呼ぶ）は，組合せ最適化問題に対する有力な近似解法としてよく知られている。まず，ランダムに生成するかまたは他の近似解法による近似解を求めて，1つの実行可能解を生成し初期解とする。次に，この解から“近い”新しい実行可能な候補解を生成して，この候補解での目的関数値が改善されれば，これを新しい解とする。目的関数値を改善させる候補解に次々と解を移動すれば，比較的良い近似解または最適解に到

達することができる。しかし，単純近傍探索法は，山登り法と同じ考え方であり，たいていの場合目的関数が多峰性を示すため，局所最適解を与えるにとどまり，広域最適解を見い出すことは期待できない。

3.2 探索空間平滑化つき近傍探索法

探索空間平滑化つき近傍探索法（Search Space Smoothing, 3S，以下 3SLS 法と呼ぶ）〔5〕は，これまでに提案されたものと違って，探索空間そのものを変換しながら，最適解または最適解に近い近似解を探索する方法である。その具体的な手順は次の通りである（図 1)。

［手順 1］ 与えられた原問題を P_0, P_0 の解空間を S_0 とする。S_0 に多数の局所最適点がある場合，まず，局所最適点が存在しないかまたは非常に少なく，かつ原問題の解空間 S_0 に近づく平滑化された探索空間（平滑空間）の系列$\{S_1, S_2, \ldots, S_k\}$を生成する。ただし，$S_1$ は S_0 にもっとも近い，S_k は S_0 にもっとも遠ざかる空間である。各平滑空間 S_i に対応する平滑問題は $P_i(i=1, 2, \ldots, k)$ と記す。

［手順 2］ 最も平滑化された（最も滑らかな）空間 S_k において，ランダムに生成するかまたは他の近似解法を用いて問題 P_k の近似解を求めて，1 つの実行可能解を生成し，s_{k+1} と記す。

［手順 3］ $i=k$ とする。

［手順 4］ s_{i+1} を探索の開始点として，空間 S_i を探索し，得られた最良解を s_i とする。

［手順 5］ $i=0$ ならば，終了する。この時点で得られた最良解 s_0 を原問題 P_0 の最良解とする。$i>0$ ならば，$i=i-1$ として，手順 4 へ戻る。

この方法では，局所最適点の多い原問題の解空間のかわりに，原問題解空間に近くてかつ原問題解空間より滑らかな平滑空間を探索することによって，原問題解空間の局所最適点を乗り越え，最適解を見つける可能性が非常に高くなる。また，最も滑らかな平滑空間から原問題の解空間へ次々

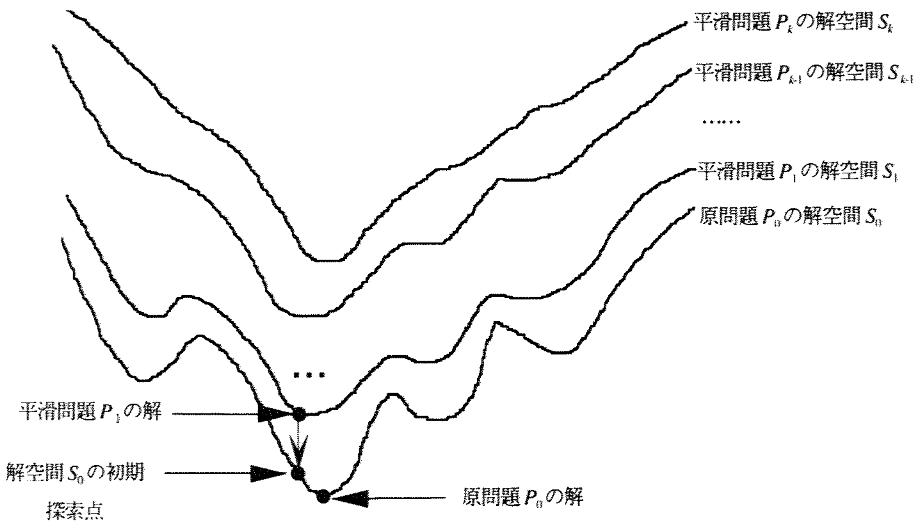

図 1. 平滑空間と問題解の探索

と探索空間を切り替えながら探索を行うことから，探索効率の向上が期待できる。

4. 探索空間平滑化つき近傍探索法によるスケジューリング解法

4.1 従来の研究と本研究の位置づけ

NP困難なスケジューリング問題は，組合せ的爆発が避けられず，実時間内に最適解を得ることができないため，より良い近似解を与える近似解法に関する研究は盛んに行われている。これまでにいろいろな発見的方法や優先規則などの近似解法が提案されている〔6〕。近年，モダン・ヒューリスティックスまたはメタ・ヒューリスティックスと称する一群の近似解法が注目されている〔7〕。主要なものとしては，近傍探索法，シミュレーテッドアニーリング法(SA法)，遺伝的アルゴリズム，タブー・サーチとニューラルネットがある。これらの方法は適用される問題とは独立し，予め用意された解法の枠組を用いてアルゴリズムを作成し，近似解を求めるところに特徴がある。スケジューリング問題への多くの適用事例により，これらの方法が解の探索能力，計算時間の両面について有効であることが示されている。

これまでに提案されたヒューリスティック解法では，問題の構造と解空間を固定したまま，探索方法に工夫することにより探索能力の向上と計算時間の短縮を図っている。これに対して探索空間平滑化つき近傍探索法は，探索方法としては近傍探索法と同じにシンプルなものを利用しながら，探索空間の平滑化を行い，問題の構造と解空間を変えることによって，より効率良く問題の最適解または最良解を見つけるアプローチである。この方法はすでに巡回セールスマン問題（TSP）に応用され，他の2つの近似解法と比べてその有効性が確認されているものの，探索空間の平滑化を行わない単純近傍探索法との比較は報告されていない〔5〕。またスケジューリング問題に適用する研究はまだ報告されていない。

そこで，本研究のねらいは，

- 探索空間平滑化つき近傍探索法をスケジューリング問題へ適用する事例を与えること；
- 単純近傍探索法および最適化解法と比較して，探索空間平滑化つき近傍探索法の有効性を明確にすること；
- スケジューリング問題に適合する探索空間の平滑化方策を与えること；
- この方法の適用に必要なパラメータの設定について考究すること

とする。

4.2 候補解の生成方策

1工程の非一様型並列機械スケジューリング問題に対して，総所要時間を最小化にするために，ジョブをできるかぎり加工時間の短い機械に割り当てるほかに，各機械の負荷(割り当てられたジョブの総加工量）のバランスを維持する必要がある。このため，次の二つの方策を提案して，近傍探索法を用いるのに必要な候補解を生成する。

［方策1］　確率的に2個のジョブを選んで，この2個ジョブの加工機械を交換する。

[方策 2]　確率的にジョブを 1 個選んで，このジョブの加工機械を，確率的に選んだ他の機械に切り換える。

4.3　平滑空間の生成

探索空間平滑化つき近傍探索法のポイントは，いかに平滑空間の系列を生成するかである。ここで，次の式 (5) を用いてジョブの加工時間を変換し，平滑化問題を生成する。

$$t_{ij}^{(\alpha)}=\begin{cases} t_a+\alpha(t_{ij}-t_a); & t_{ij}\geq t_a \\ t_a-\alpha(t_a-t_{ij}); & t_{ij}<t_a \end{cases} \tag{5}$$

ただし，$t_a=\sum_{i=1}^{n}\sum_{j=1}^{m}t_{ij}/(m\times n)$, α は $0\leqq\alpha\leqq 1$ を満たす平滑定数である。

問題 P における作業時間 t_{ij} を $t_{ij}^{(\alpha)}$ に置換えば，次の平滑化問題 P(α) を生成することができる。

問題P(α)

Minimize $C^{(\alpha)}=\mathrm{Max}\{C_j^{(\alpha)};\ j=1, 2, \ldots, m\}$　(6)

Subject to

$$C_j^{(\alpha)}=\sum_{i=1}^{n}x_{ij}t_{ij}^{(\alpha)} \tag{7}$$

$$\sum_{i=1}^{n}x_{ij}=1; \tag{8}$$

$$x_{ij}=\begin{cases} 1; & M_j \text{ で } J_i \text{ を加工するとき} \\ 0; & \text{その他} \end{cases} \tag{9}$$

$i=1, 2, \ldots, n;\ j=1, 2, \ldots, m.$

明らかに α の値を 0〜1 の範囲内で逐次に変更することにより，平滑化問題の系列が生成される。極端な場合としては，

(1)　$\alpha=0$ のとき：$t_{ij}^{(\alpha)}=t_a$ のため，P(α) はジョブの加工時間が同じである等価機械型の問題となる。このとき，ジョブの加工時間がすべて同じ値なので，問題の解空間は “平らな平面” であり，最適解は容易に求められる。

(2)　$\alpha=1$ のとき，$t_{ij}^{(\alpha)}=t_{ij}$ のため，P(α) は原問題 P となる。

4.3　提案解法

探索空間平滑化つき近傍探索法によるスケジューリング解法を以下のとおり提案する。

【提案アルゴリズム】

[ステップ 1]　$t_{ik}=\min\{t_{ij};\ j=1, 2, \ldots, m\}$ ならば，ジョブ J_i を機械 M_k で加工するように，各ジョブの加工機械を決めて，初期解 s^* を生成し，これに対応する目的関数値 (総所要時間) を C^* とする。また，$\alpha=\alpha_0(0<\alpha_0\leqq 1)$ とする。

[ステップ 2]　式 (5) を用いて $t_{ij}^{(\alpha)}$ を計算し，t_{ij} を $t_{ij}^{(\alpha)}(i=1, 2, \ldots, n;\ j=1, 2, \ldots, m)$ に置き換えた平滑化問題 P(α) を生成する。

［ステップ3］　s^* を初期点として，単純近傍探索法を用いて平滑化問題 P(α) を解く。

［ステップ3.1］　4.2節で述べた方策1を適用して，候補解 s' を生成し，そのときの目的関数値を C' とする。$C' \leqq C^*$ ならば，s^* を s' に置換え，$C^* = C'$ とする。$C' > C^*$ ならば，直接ステップ3.2へ。

［ステップ3.2］　4.2節で述べた方策2を適用して，候補解 s'' を生成し，そのときの目的関数値を C'' とする。$C'' \leqq C^*$ ならば，s^* を s'' に置換え，$C^* = C''$ とする。$C'' > C^*$ ならば，直接ステップ3.3へ。

［ステップ3.3］　ステップ3.1と3.2を定めた回数だけ繰り返してから，ステップ4へ。

［ステップ4］　$\alpha = \alpha + \Delta\alpha$ とする。$\alpha > 1$ ならば終了し，s^* を問題の最良解とする。$\alpha \leqq 1$ ならばステップ2へ戻る。

5.　計算例による考察

5.1　評価基準

次の式 (10) で目的関数の下界値を計算する。得られた解の目的関数値を C として，この解の良さを，式 (11) で定義する近似度 AR (Approximation Ratio) で評価する。むろん，AR の値が小さいほど良い解である。

$$LB = \left(\sum_{i=1}^{n} \min\{t_{ij};\ j = 1, 2, \ldots, m\}\right)/m \tag{10}$$

$$AR = (C - LB)/LB \tag{11}$$

5.2　計算例の生成とパラメータの設定

計算例は，ジョブの加工時間 t_{ij} を (1,100) の一様乱数で与えて，生成される。

$\Delta\alpha = 0.1$ とする。なお，生成された平滑空間を $S_1, S_2, \ldots, S_k$ として，その数が k であることを，平滑回数が k であるという。平滑回数を k とすれば，α_0 は次の式 (12) で決まる。

$$\alpha_0 = 1.0 - k \times \Delta\alpha \tag{12}$$

候補解生成の方策1と2をそれぞれ1回適用し，1つずつの候補解を生成し，良い候補解に解を移動することを，1回の探索とする。探索回数が同じ条件の下で，単純近傍探索法と提案解法による解の良さを比較するために，提案解法の探索回数SRT (Search Repeating Times) は，平滑空間 $S_1, S_2, \ldots, S_k$ と原問題の解空間 S_0 において行われた探索の総回数である。言い換えれば，もしSRT $= N$ とすれば，各空間において行われる探索の回数は，それぞれ $N/(k+1)$ となる。

また，候補解は一様乱数を用いて生成されるので，同じ問題でも毎回の実行で同じ解を得るとは限らない。このため，計算例の実行結果では，同じ問題の解を25回求めて，得られた解の目的関数値の最大値，平均値と最良値を与える。

表 1. 提案解法と最適化解法との比較（$m=5$）

ジョブ数	下界値	最適解(AR%)	提案解法による解			単純近傍探索法による解		
			最大値	平均値	最良値(AR%)	最大値	平均値	最良値(AR%)
10	33.9	50.9(51.3)	53.0	51.6	50.9(51.3)	53.0	51.6	50.9(51.3)
15	51.8	67.1(30.6)	68.3	68.6	67.1(30.6)	73.0	69.0	67.1(30.6)
20	71.9	86.6(20.6)	91.3	88.3	86.6(20.6)	96.7	89.2	86.7(20.7)
25	90.4	103.9(15.0)	112.2	105.7	103.9(15.0)	115.5	106.8	103.9(15.0)
30	109.3	123.6(13.2)	131.8	126.2	123.6(13.2)	135.1	127.2	123.7(13.3)
35	123.3	136.5(11.0)	146.7	139.2	136.5(11.0)	148.8	140.2	136.5(11.0)
40	138.3	148.9(7.7)	160.9	152.7	148.9(7.7)	160.9	153.2	149.2(7.86)

5.3 最適化解法との比較

$m=5$ として，$n=10, 15, 20, 25, 30, 35, 40$ の 7 組の計算例を生成し，分岐限定法に基づく最適化解法，単純近傍探索法と提案解法を用いて，求めた問題の解を表 1 に示す。ただし，

(1) $n=10, 15, 20, 25, 30$ の時の計算結果は，生成された 10 問問題の平均値である。$n=35, 40$ の時の計算結果は，分岐ノード数の上限を 3,100,000 とした場合に，最適化解法を用いて 20 問の中で最適解が求まった 10 問問題の平均値である。

(2) 分岐限定法に必要な分岐手順と下界の計算方法については，主に野村らの論文〔4〕を参考した。

(3) 単純近傍探索法と提案解法については，探索回数を $\mathrm{SRT}=n\times m\times 100$，平滑回数を $k=1$ ～3 と設定した。

表 1 から，ジョブ数が 40 以下の場合，提案解法を用いて 25 回の実行ですべての問題の最適解が求まった。C 言語で各解法のプログラムを組み，DOS/V パソコン (Pentium 90 MHz, Memory 16 MB) 上で実行した場合，$n=40$ の 1 問題あたりの平均計算時間は，最適化解法では 174 秒かかったのに対して，提案解法では 25 回計算の合計時間が 8 秒であった。

また，単純近傍探索法を用いても，25 回の実行で $n=20, 30$ の場合を除いて他の問題の最適解が得られた。しかし，得られた解の最大値と平均値を比べて，提案解法は単純近傍探索法より良い結果が得られ，提案解法の有効性は明らかである。さらに，探索回数が同じであるので，提案解法と単純近傍探索法の計算時間はほとんど変わらない。

5.4 発見的解法との比較

$m=3, 5$ として，$n=50, 100, 150, 200$ の 4 組の計算例をそれぞれ 10 問生成し，Ibarra と Kim の発見的解法(以下，IK 解法と呼ぶ)〔1〕，Davis と Jaffe の発見的解法(以下，DJ 解法と呼ぶ)〔2〕，単純近傍探索法と提案解法を用いて，求めた問題の解を表 2(a), 2(b) に示す。ただし，表 2 の計算結果は，同じ組の 10 問問題の平均値である。単純近傍探索法と提案解法では，探索回数を $\mathrm{SRT}=n\times$

表 2(a)．提案解法と発見的解法との比較（$m=5$）

ジョブ数	下界値	提案解法による解（AR%）			単純近傍探索法による解（AR%）			IK 解法 (AR%)	DJ 解法 (AR%)
		最大値	平均値	最良値	最大値	平均値	最良値		
50	179.2	202.0 (12.88)	195.2 (9.11)	191.3 (6.90)	204.3 (14.14)	196.8 (9.97)	191.7 (7.12)	225.1 (25.96)	221.5 (14.13)
100	343.4	368.6 (7.42)	360.7 (5.08)	354.8 (3.35)	371.0 (8.07)	362.4 (5.59)	355.8 (3.65)	393.6 (14.67)	406.6 (18.62)
150	514.6	538.6 (4.69)	530.9 (3.17)	525.1 (2.05)	540.7 (5.08)	531.7 (3.33)	525.4 (2.11)	559.6 (8.73)	585.4 (13.81)
200	686.8	713.1 (3.83)	704.6 (2.58)	698.4 (1.67)	716.4 (3.83)	706.5 (2.86)	698.9 (1.75)	729.2 (6.15)	768.8 (11.91)

表 2(b)．提案解法と発見的解法との比較（$m=3$）

ジョブ数	下界値	提案解法による解（AR%）			単純近傍探索法による解（AR%）			IK 解法 (AR%)	DJ 解法 (AR%)
		最大値	平均値	最良値	最大値	平均値	最良値		
50	434.9	463.0 (6.49)	452.5 (4.11)	447.4 (2.94)	463.4 (6.62)	453.1 (4.26)	447.4 (2.95)	476.3 (9.69)	509.8 (17.82)
100	851.3	874.3 (2.72)	865.7 (1.70)	859.2 (0.93)	874.6 (2.75)	865.9 (1.73)	859.8 (1.00)	890.7 (4.62)	926.6 (18.86)
150	1270.4	1295.7 (2.00)	1284.8 (1.13)	1275.7 (0.56)	1295.1 (1.95)	1284.7 (1.13)	1278.1 (0.60)	1311.8 (3.27)	1348.8 (6.21)
200	1693.1	1719.1 (1.54)	1709.3 (0.96)	1701.6 (0.51)	1720.5 (1.63)	1710.0 (1.01)	1702.0 (0.53)	1734.8 (2.47)	1840.3 (8.76)

$m\times$p(p=200, 300)，平滑回数を $k=1$～3 と設定した。

表 2 から，提案解法を用いて得られた解の最良値は，発見的解法の 1/3 以下であり，最大値も発見的解法より小さく，明らかに提案解法は発見的解法より有効である。また，提案解法による問題の最良解の近似度は，$m=5$ の場合 6.90～1.67% であり，$m=3$ の場合 2.94～0.51% である。問題の最適値が式 (11) で求めた下界値 LB より小さくないことを考慮して，提案解法による最良解は最適解に非常に近いと考えられる。

また，単純近傍探索法を用いても，発見的解法より非常に良い解が得られた。しかし，得られた解の最大値と平均値を比べて，提案解法は単純近傍探索法より良い結果が得られ，提案解法の有効性も明らかである。提案解法と単純近傍探索法による解の近似度の差が小さいため，さらに問題ごとの比較を行った結果，各組の 10 問問題の中で 7～8 問の問題は，両解法とも同じ解が得られ，あと 2～3 問の問題は，提案解法が単純近傍探索法より良い解を求めたことがわかった。また，n/m の値が大きくなるほど，両解法の差は小さくなることもわかった。

5.5 平滑回数と探索回数の設定についての考察

探索空間平滑化つき近傍探索法の有効性は，パラメータの選定に左右される。同様に，($m=3$, $n=50$), ($m=5$, $n=30$), ($m=5$, $n=50$) の 3 組の計算例をそれぞれ 10 問生成し，異なった平滑回数

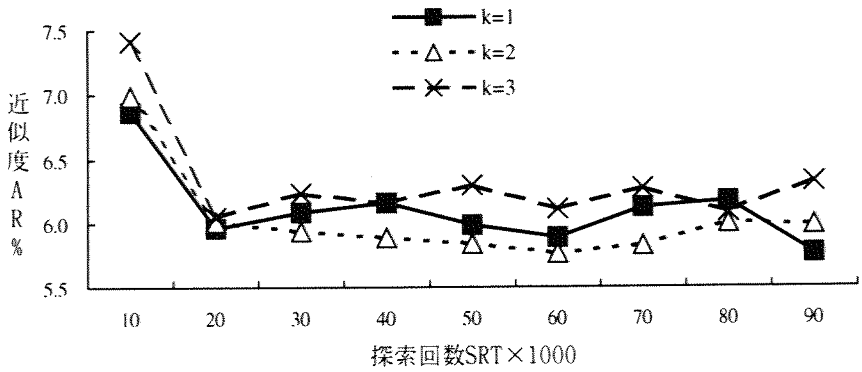

図 2(a). 近似度と平滑回数（$m=3$, $n=50$）

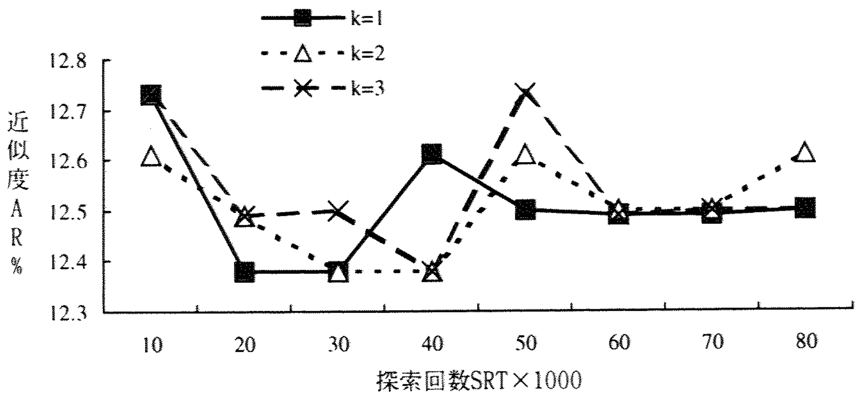

図 2(b). 近似度と平滑回数（$m=5$, $n=30$）

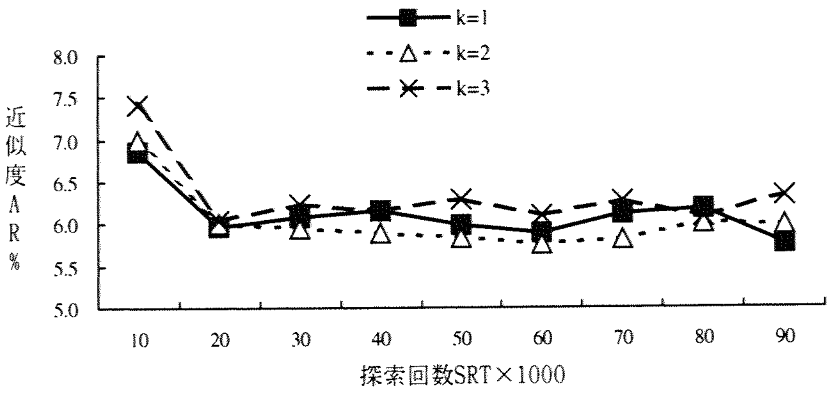

図 2(c). 近似度と平滑回数（$m=5$, $n=50$）

k と探索回数 SRT を与えて，各組の問題の最良解を求め，得られた結果をそれぞれ図 2(a)～(c) に示す。ただし，図 2 に示された近似度は，同じ組における 10 問計算例の最良解の近似度を求め，さらにこれらの算術平均値を計算したものである。

図 2 から次のことがわかる。

(1)　平滑回数 $k=1\sim3$ とすれば，ほとんどの場合において非常に良い解を得ることができる。さらに，問題によって多少異なるが，平滑回数 $k=2$ とするとき得られた解は比較的良い。このため，$k=2$ が適当な平滑回数である。その次は $k=1$ である。

(2)　図2には示されていないが，平滑回数 k が4以上（$k \geqq 4$）であれば，得られた近似度は図2に示した値より2〜3倍増えて，探索空間平滑化の効果は見られない。これは，探索空間平滑化つき近傍探索法は，各探索空間において単純近傍探索法を用いて最適点を探索するので，探索空間の数が増えると，局所最適点に陥る確率が高くなり，最終的に得られた解の品質は悪くなるからであると考えられる。

(3)　探索回数SRTを10,000以下とする場合，探索が不十分で得られた解は良くなかった。また，SRTを大幅に増えても，解の改善があまり顕著でなかったため，探索回数SRT＝(100〜500)$\times n \times m$ と設定すればよい。

5.6　$\Delta\alpha$ の設定についての考察

$\Delta\alpha$ の設定により各平滑空間と原問題の解空間における問題のジョブ加工時間の差が変わる。平

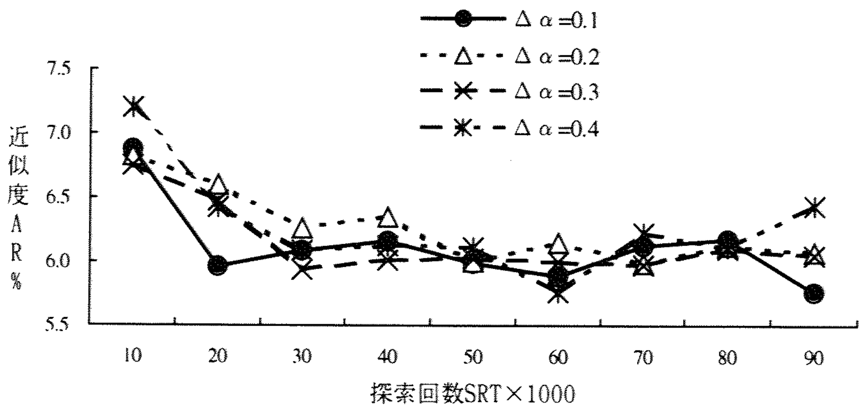

図3(a).　$\Delta\alpha$ と近似度（$k=1$）

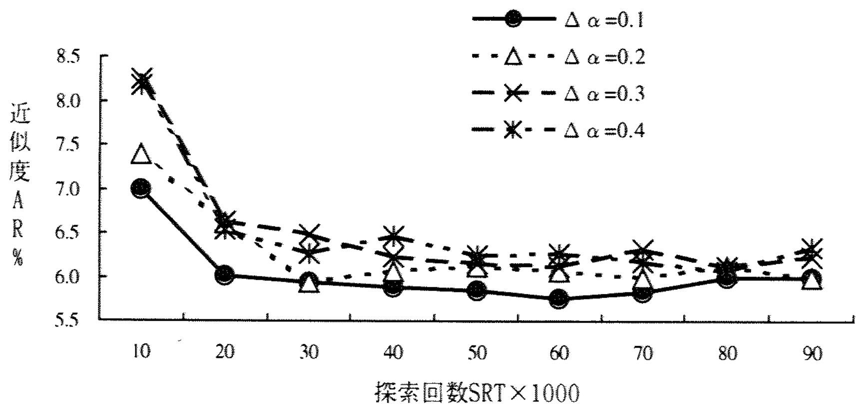

図3(b).　$\Delta\alpha$ と近似度（$k=2$）

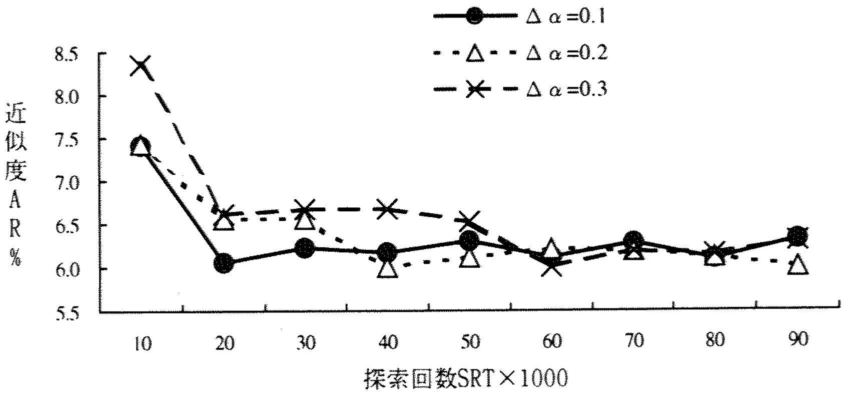

図3(c). $\Delta\alpha$ と近似度（$k=3$）

滑化の効果に対する $\Delta\alpha$ の影響を調べるため，($m=5, n=50$) の計算例を10問生成し，異なった $\Delta\alpha$ を与えて，各問題の最良解を求め，得られた結果をそれぞれ図3(a)～(c) に示す。同様に，図3に示された近似度は，同じ組の10問計算例の最良解を求め，得られた最良値の近似度の平均値である。

図3から，次のことがわかる。

(1) 探索回数が少ないとき，$\Delta\alpha=0.1$ として比較的良い解が得られる。

(2) $k=2$ の場合，$\Delta\alpha=0.1$ としたときの解は，顕著に良い。

$k=1$ と $k=3$ の場合，$\Delta\alpha$ の値と近似度の関連が明確ではないが，近似度の値が全体的に $k=2$ の場合より大きいことを考慮すれば，$\Delta\alpha$ の値を小さくした場合は，平滑化の効果が顕著である。しかし，$\Delta\alpha$ を0.05以下にすれば，平滑化の効果があまり見られなかった。このため，$\Delta\alpha=0.1$ が適当な値であると考える。

6. おわりに

本研究では，非一様型並列機械スケジューリング問題を検討し，探索空間平滑化つき近傍探索法を用いて，新たな近似解法を提案するうえ，その有効性を明かにした。得られた成果は次のように要約される。

(1) 総所要時間の最小化を目的関数とする場合，提案解法は，従来の発見的解法より有効であり，最適解に十分近い最良解を見つけることができる。また，単純近傍探索法に比べても，提案解法は，同じ計算時間でより良い近似解を得られ，提案解法の有効性は明らかである。

(2) 探索空間平滑化つき近傍探索法の適用にあたり，平滑回数=1～3, $\Delta\alpha=0.1$, 探索回数 $\mathrm{SRT}=n\times m\times(100\text{～}500)$ と設定すれば，十分良い解が求められる。

また，Guら〔5〕の論文では指数型の平滑化関数が用いられていたが，本研究では，対象としたスケジューリング問題に関して，指数型平滑化関数の平滑化効果は式(5)に示した線形型平滑化関数ほど顕著でないことがわかった。

探索空間平滑化つき近傍探索法については，研究課題がまだたくさん残っている。これまでの計

算結果では，少ない平滑回数で比較的良い解が得られたが，平滑回数と探索回数をともに増やせば，提案解法による解の良さはSA法と同様にパラメータの選定によって異なり，単純近傍探索法とSA法に対する提案解法の優位性は問題によっては損なわれる。それゆえ，もっと効果的平滑化関数あるいは候補解の生成方策を提案することが必要である。さらに，各平滑空間における最適点間の関連，特に探索空間平滑化のメカニズムを解明する必要がある。これらについては，今後の研究課題にしたい。

参 考 文 献

〔1〕 O.H., Ibarra & C.E., Kim: “Heuristic Algorithms for Scheduling Independent Tasks on Nonidentical Processors”, Journal of The Association of Computing Machinery, Vol. 24, No. 2, pp. 721～736 (1977).

〔2〕 E. Davis & J.M. Jaffe: “Algorithms for Scheduling Tasks on Unrelated Processors”, Journal of The Association of Computing Machinery, Vol. 28, No. 4, pp. 721～736 (1981).

〔3〕 J.K. Lenstra, D.B. Shmoys & E. Tardos: “Approximation Algorithms for Scheduling Unrelated Parallel Machines”, Mathematical Programming, No. 46, pp. 259-271 (1990).

〔4〕 野村弘光，羽田隆男：“分岐限定法による非一様型並列機械スケジューリング問題の解法”，日本経営工学会誌，Vol. 45, No. 2, pp. 155-161 (1994).

〔5〕 J. Gu & X. Huang: “Efficient Local Search with Search Space Smoothing: A Case Study of The Traveling Salesman Problem (TSP)”, IEEE Transaction on S.M.C., Vol. 24, No. 5, pp. 728-735 (1994).

〔6〕 T.C.E. Cheng & C.C.S. Sin: “A state-of-the-art review of parallel-machine scheduling research”, European Journal of Operational Research, Vol. 47, pp. 271～292 (1990).

〔7〕 黒田　充：“生産スケジューリングの方法 —— 体系化と状況適合モデル構築の試み”，経営システム，pp. 5-14, Vol. 5, No. 1 (1995).