

## フローショップスケジューリング問題の 地形解析と遺伝的局所探索による解法

山田 武士<sup>†</sup> Colin R. Reeves<sup>††</sup>

本論文では、総作業時間 (makespan) 最小・順列型フローショップスケジューリング問題 (PFSP) を題材とした探索空間の地形 (landscape) を解析する。特にクリティカルブロックに基づく近傍と、確率的局所探索を用いる場合を考える。例として最適解が既知な、比較的規模の小さい問題を取り上げ、ランダムに生成した局所最適解および真の最適解の相互の関係を調べたところ、目的関数である総作業時間と、各局所最適解から他の局所最適解への距離平均、各局所最適解から真の最適解への距離との間に強い正の相関が存在することが分かった。これらの相関は、局所最適解が探索空間内にクラスタとなって存在する、いわゆる「大谷構造」(big valley structure) が探索空間に存在することを示唆している。したがってこの大谷構造を有効に利用できる局所探索によってこの問題が効果的に解けることが期待できる。そこでジョブショップ問題解法として以前提案された MSXF-GA による PFSP 解法を提案した。さらに比較的難問なベンチマーク問題を用いた計算機実験により本方法の有効性を検証した。

### Landscape Analysis of the Flowshop Scheduling Problem and Genetic Local Search

TAKESHI YAMADA<sup>†</sup> and COLIN R. REEVES<sup>††</sup>

In this paper, the landscape for the permutation flowshop scheduling problem (PFSP) with stochastic local search and a critical block-based neighbourhood structure has been investigated. Numerical experiments using small benchmark problems with known global optima show that there are good correlations between the makespans of local optima, the average distances to other local optima and the distances to the global optima. These correlations suggest the existence of a 'big valley' structure, where local optima occur in clusters over the landscape. An approximation method for PFSP that would make use of this big valley structure is proposed by using a critical block-based neighbourhood structure, and a genetic local search method called MSXF-GA, previously developed for the job shop scheduling problem. Computational experiments using more challenging benchmark problems demonstrate the effectiveness of the proposed method.

#### 1. はじめに

遺伝的アルゴリズム (GA) は自然淘汰と遺伝の仕組みにヒントを得た最適化手法の 1 つであり、機械学習からシステム制御まで様々な分野に応用され成功を収めている。GA は組合せ最適化問題の近似的解法としても有用であるが、その場合、局所探索能力を補うために他の局所探索法 (LS)、主に近傍探索法と組み合わせることによって性能を著しく改善できることがよく知られている。このような解法はしばしば遺伝的局所探索 (Genetic Local Search: GLS) もしくは集団的局所探索 (population-based local search) と呼ばれる。

基本的に GLS は短期の局所探索を繰り返し適用する多スタート法、なかでも、新たな局所探索の初期解をそれまで行った複数の局所探索の結果に基づいて適応的に決定する、適応的多スタート法 (Adaptive Multi-Start methods: AMS) の一種と見なすことができる。著者らは以前、ジョブショップスケジューリング問題解法として、多段階探索交叉 (Multi-Step Crossover Fusion: MSXF) という交叉法とそれに基づく GLS 解法を提案し、成果を収めた<sup>15)</sup>。MSXF は従来の交叉オペレータとは異なり、一方の親そのものを新たな探索の初期値とし、もう一方の親に近づく方向を優先的に探索する局所探索 (navigated local search) を行うオペレータである。

一般に AMS もしくは GA によってある問題が効果的に解けるかどうかは、その探索空間  $S$  と、目的関

<sup>†</sup> NTT コミュニケーション科学研究所

NTT Communication Science Laboratories

<sup>††</sup> Coventry University, UK

数  $f$ , そして探索空間上に定義された距離  $d$  の組である  $\Lambda = (S, f, d)$  の構造に依存する. ここで  $\Lambda$  は地形 (landscape) と呼ばれる. 特に GA の場合, 目的関数  $f$  を fitness と呼ぶため, fitness landscape と呼ばれることもある. たとえば Boese ら<sup>3)</sup> は, あらかじめ AMS が有効であると分かっている巡回セールスマン問題やグラフ分割問題を取り上げ, ランダムに生成した局所解間の関係を調べた. その結果局所解どうし, あるいは局所解と既知の最適解はお互いに似通っている, すなわち, 問題に則して定義された距離  $d$  が比較的小さいことを発見した. 言い換えれば, 局所解はクラスタとなって存在する, いわゆる「大谷構造」(big valley structure) が存在する. 一方 Jones ら<sup>9)</sup> は, 多くの GA-easy な問題において適応度の分布と最適解への距離の分布の間には強い相関があることを示している. 最近では Reeves が単純な近傍探索オペレータを用いた場合について, 総作業時間 (makespan) 最小・順列型フローショップスケジューリング問題, 以降簡単に PFSP (Permutation Flowshop Scheduling Problems) と記す, の地形の解析を行っている<sup>10)</sup>.

本論文では, クリティカルブロックに基づく効果的な近傍探索オペレータと, 確率的な局所探索を用いた場合の PFSP の地形について解析し, その結果に基づいた効果的な PFSP 解法を提案する. 本論文の構成は次のようである. 2 章で本論文で扱う PFSP の定義と, PFSP の近傍構造であるクリティカルブロックに基づく代表近傍, そして距離の計算方法について, 3 章で探索空間の地形, 確率的近傍探索について説明し, クリティカルブロックに基づく代表近傍と確率的近傍探索を用いた場合の PFSP の探索空間の地形が大谷構造を持つかどうかを調べる. さらに 4 章では大谷構造を持つ場合の効率的な探索法について考察し, その 1 つとして MSXF 法について説明する. 5 章で MSXF を用いた PFSP の GA 解法 MSXF-GA を提案し, 最後に 6 章で実際に MSXF-GA を難問とされる Taillard のベンチマーク問題の一部に適用し, 提案法の有効性を検証する.

## 2. フローショップスケジューリング問題

本論文で扱うフローショップスケジューリング問題は特に順列型と呼ばれるものであり, 通常  $n/m/P/C_{\max}$  と表現される. ここで  $n$  は仕事数,  $m$  は機械数であり,  $n$  個の仕事  $J_1, \dots, J_n$  は  $m$  台の機械  $M_1, \dots, M_m$  上で仕事によらず同一の順序で処理される. 各仕事の各機械上での処理は作業と呼ばれ, 各機械は一度にたかだか 1 つの仕事を中断なしに, 定められた処

理時間をかけて処理する. 記号  $P$  は各機械が各仕事を処理する順序は機械によらないことを表し, その場合完成されたスケジュールは  $n$  個の仕事の順列によって一意的に表される (したがって順列型と呼ばれる). そこで本論文では解を仕事 (名) の順列と同一視し,  $S = J_{p_1} J_{p_2} \dots J_{p_n}$  と表したり, あるいは単に  $p_1 \dots p_n$  のように表す. この問題の目的はすべての仕事を処理するために必要な時間である総作業時間 (makespan) を最小にすることであり, これを記号  $C_{\max}$  で表す.

### 2.1 PFSP の近傍構造

一般に探索空間内の点  $x$  の近傍  $N(x)$  とは,  $x$  に対し遷移操作 (transition) と呼ばれる解の部分的な変更を 1 回だけ行って到達可能な点全体の集合である. PFSP の最も基本的な遷移操作としては互いに隣り合う作業を入れ替える交換操作 (exchange operator) があげられる. ある仕事をその処理位置から別の処理位置に移動させるシフト操作 (shift operator) もよく知られており, これを用いた探索は単純な遷移操作の中では比較的成功を収めている<sup>9)</sup>. より洗練されたオペレータとしては, クリティカルパスおよびクリティカルブロックに基づくものが知られており, これはジョブショップ問題で成功を収めた同様な方法をフローショップ問題にも取り入れたものである.

ここでクリティカルパスとは, 第 1 の機械  $M_1$  上での第 1 の仕事の処理を始点とし, 最後 ( $m$  番目) の機械  $M_m$  上での最後の仕事の処理を終点とする作業の処理順序列であり, このパス上では各作業の処理開始時刻は直前の作業の処理終了時刻に等しいように処理されているものをいう. したがって, このパスの全長はパス上の各作業の処理時間の和に等しく, さらにこれは総作業時間  $C_{\max}$  に等しい. クリティカルパス上の作業はそれらが処理される機械ごとにグループ化することができ, これらをクリティカルブロックと呼ぶ. したがって, クリティカルブロックとは, 同一機械上で処理されるクリティカルパス上の作業の部分列である. ただし PFSP の場合, 簡略化のためこの作業の部分列の代わりに対応する仕事の部分列を用いる. クリティカルブロックに基づく近傍とは, 各仕事をそれが属するブロックの外側へシフトする遷移のみ考慮することによって, シフト操作に基づく近傍を制限したものである.

クリティカルブロックに基づく近傍のサイズをさらに制限するため, Nowicki ら<sup>7)</sup> は代表近傍 (representative neighbourhood) という考え方を導入した. ここで代表近傍とは, もとになる近傍をクラスタに分割し,

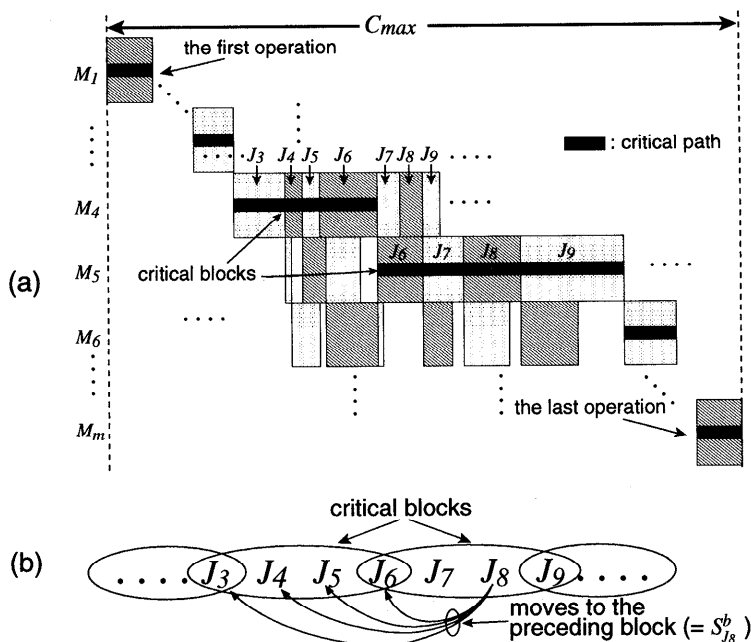


図1 クリティカルパスとクリティカルブロックの例  
 Fig. 1 An example of the critical path and blocks.

各クラスタ内の最も優秀な要素をそのクラスタの代表元とし、すべての代表元からなる集合を新たな近傍とする方法である。本論文では Nowicki らと同様に次のような方法を用いる。いま、クリティカルブロック内の任意の仕事 \$j\$ とし、仕事 \$j\$ を直後の (右隣の) クリティカルブロック内の任意の位置へ移動する遷移全体の集合を \$S\_j^a\$、直前の (左隣の) クリティカルブロック内へ移動する遷移全体の集合を \$S\_j^b\$ とする。\$S\_j^a, S\_j^b\$ の各要素をすべて評価し、最も評価値の高いもの、すなわち最も \$C\_{max}\$ の値の小さいものをそれぞれ \$s\_j^a, s\_j^b\$ とする。このとき、代表近傍はクリティカルブロック上のすべての仕事に関する \$\{s\_j^a, s\_j^b\}\$ によって得られる全スケジュールの集合として定義される。

図1にクリティカルパスとクリティカルブロックの例を示す。図1(a)は解の一部をガントチャートによって表現したものであり、クリティカルパスは図中の太線で表現されている。図において仕事 \$J\_3\$ から \$J\_6\$ まで、\$J\_6\$ から \$J\_9\$ までがそれぞれクリティカルブロックを形成することが分かる。作業の代わりに仕事を用いた、クリティカルブロックの簡略化された表現を図1(b)に示す。図においてたとえば \$J\_8\$ を直前のブロックに移動する合計4つの遷移あることが分かる。これらが \$S\_8^b\$ を形成する。

2.2 PFSP の距離尺度

仕事の順列で表された2つのスケジュール \$S, S'\$ の

類似度は、\$n\$ 文字の順列全体からなる空間上にしかるべき距離を定義することによって測定できる。たとえば次の3種類の距離がよく知られている。

隣接関係による距離 (adjacency-based: \$d\_{adj}\$)

仕事のペア \$i, j\$ が \$S, S'\$ 上ともにこの順序で隣接して処理されるような \$i, j\$ の個数を \$n\_{adj}\$ としたとき、\$d\_{adj}(S, S') = n - n\_{adj} - 1\$ (これを uni-directional と呼ぶ)。または条件を少し緩めて、\$i, j\$ または \$j, i\$ が \$S, S'\$ 上ともに隣接する場合の \$i, j\$ の個数を \$n\_{adj}\$ とする場合もある (bi-directional と呼ぶ)。

先行関係による距離 (precedence-based: \$d\_{pre}\$)

仕事のペア \$i, j\$ が \$S, S'\$ 上ともにこの順序で処理される、すなわち \$i\$ の処理が \$j\$ の処理より先行する \$i, j\$ の個数を \$n\_{pre}\$ としたとき、\$d\_{pre}(S, S') = n(n-1)/2 - n\_{pre}\$。

位置関係による距離 (position-based: \$d\_{pos}\$)

仕事 \$i\$ の \$S, S'\$ 上でのそれぞれの処理位置を \$\sigma\_i, \sigma'\_i\$ とするとき (すなわち \$s\_{\sigma\_i} = i\$)、\$d\_{pos}(S, S') = \sum\_{i=1}^n |\sigma\_i - \sigma'\_i|\$。

PFSP と同様に解が順列によって表される巡回セールスマン問題 (TSP) を例にとると、TSP では順列上の要素の絶対的な位置よりも要素相互の相対的な位置関係が重要であるため、隣接距離 \$d\_{adj}\$ (特に bi-directional なもの) がよく用いられる。一方 PFSP で

は仕事が処理される絶対的な位置関係も重要であるため、先行関係もしくは位置関係による距離  $d_{pre}, d_{pos}$  のほうがより適していると考えられる。

容易に分かるように  $d_{pre}$  の場合、任意のスケジュール  $s = s_1 s_2 \dots s_{n-1} s_n$  と、 $s$  の順序をすべて逆にしたスケジュール  $\hat{s} = s_n s_{n-1} \dots s_2 s_1$  との間の距離  $d_{pre}(s, \hat{s})$  において最大値  $n(n-1)/2$  をとる。一方  $d_{pos}$  の場合には  $d_{pre}$  と同様に  $d_{pos}(s, \hat{s})$  において最大となるが、 $n$  が奇数が偶数かによって最大値は異なる ( $n$  が奇数:  $(n^2 - 1)/2$ ,  $n$  が偶数:  $n^2/2$ )。また  $\hat{s}$  以外にも  $d_{pos}(s, t)$  が最大となるスケジュール  $t$  が存在する (たとえば  $d_{pos}(12345, 54321) = d_{pos}(12345, 45312)$ )。したがって、より扱いが複雑である。そこで本論文では主に  $d_{pre}$  を用いる。ただし  $d_{pre}, d_{pos}$  の2つの間には非常に強い相関があり、どちらを用いても本質的な違いはない。

### 3. 探索空間の地形解析

近年、Wolpert ら<sup>14)</sup> は目的関数の最適化をするようなアルゴリズム一般の性能に関する主張である “no free lunch theorem” を提示し関心を集めている。彼らの主張によれば、(一定の条件のもと) どんなアルゴリズムであっても、あらゆる目的関数について平均をとればアルゴリズムの性能はどれもみな等しい。すなわち一般にどんな問題に対しても遍く優れた解法というものは存在しえない。これを認めれば、ある特定の問題に関して解法 A が解法 B より優れているのは解法 A が対象となる問題の性質を解法 B よりもよりの確に活用しているからにはほかならないということになる。したがって問題に適した解法とは、対象となる問題の探索空間の構造上の特徴を何らかの形で反映したものでなければならない。以上をふまえて本章では探索空間の構造の尺度である、地形 (landscape) について考える。

#### 3.1 地形 (landscape) とは

地形 (landscape) とは探索空間の構造の複雑さの尺度である。直観的には  $f(x)$  の最適化における  $y = f(x)$  のグラフの様子と考えることができる。たとえばグラフの形状がすり鉢状で唯一の極値を持つとき  $f$  の地形は単純で、簡単な降下法によって容易に最適解が求まる。このような観点から地形  $\Lambda$  は探索空間  $S$  と目的関数  $f$  の間の関係と定義できる。しかし実際には特に組合せ最適化問題の場合、問題の複雑さ、難解さが、解法に用いられる探索オペレータにも依存するため、 $S$  と  $f$  だけでは不十分である。そこで Reeves ら<sup>11)</sup> は地形  $\Lambda$  を探索空間  $S$ 、目的関数  $f$ 、そして探

索空間上に定義された距離  $d$  の組として  $\Lambda = (S, f, d)$  と表現した。特に近傍探索の場合  $d$  として近傍探索オペレータ  $\mathcal{N}$  によって導入される距離  $d_{\mathcal{N}}$  を用いることにより、地形の複雑さと最適化の難しさとが関連付けられる。ただし、ここで探索空間内の任意の解  $s, t$  に対し  $d_{\mathcal{N}}(s, t)$  は、 $t$  から近傍探索オペレータによって  $s$  に至るまでの最小ステップ数として与えられる。

最近 Boese ら<sup>3)</sup> は、問題に応じた適切な近傍構造、および近傍探索オペレータを用いることによって  $\Lambda$  にいわゆる「大谷構造」(big valley structure) が発現することを、巡回セールスマン問題やグラフ分割問題の場合に明かにした。ある目的関数を最小にする問題を考えるとき、大谷構造とは探索空間の地形が大局的に見ると真の最適解である極小値に向かって1つの大きな谷を持つような構造をしていることである。もし問題の真の最適解と局所解の分布が分かっていたら、探索空間の地形が大谷構造を持つかどうかは距離として近傍探索オペレータ  $\mathcal{N}$  によって導入される距離  $d_{\mathcal{N}}$  を用いた場合の、各局所最適解の目的関数の値と、各局所最適解から他の局所最適解への距離 ( $d_{\mathcal{N}}$ ) 平均、各局所最適解から真の最適解への距離 ( $d_{\mathcal{N}}$ ) との間に正の相関があるかどうかによって調べることができる。

大谷構造においては局所最適解はクラスタとなって存在し、したがって、1つ優秀な解が発見されればさらに優秀な解がその比較的近くに存在する可能性がある。もし解くべき問題の  $\Lambda$  がこの性質を持てば、たとえば多スタート法において、新たな探索の初期値をランダムに選ぶのではなく現在までに得られた局所最適解に基づいて選ぶほうが有利だということになり、さらに4.1節で述べるように、経路再結合の考え方が有効であることを意味する。

#### 3.2 確率的近傍探索

実際の地形解析を行う前に、ここで用いる局所探索について簡単に説明する。いま、 $x$  の近傍を  $N(x)$  とし、目的関数  $V(x)$  を最小にする近傍探索 (Neighbourhood Search: NS) の概要を図2に示す。ただし図中の終了条件としては、たとえば「繰返し回数がある定数  $L$  に達するまで」などを用いる。

近傍探索は図2のステップ(1)で用いられる選択基準によって特徴づけられる。たとえば  $V(y) < V(x)$  となるような  $y \in N(x)$  をランダムに選ぶという最も単純な方法は「降下法」(descent method) と呼ばれ、また  $y$  として最小のものを選ぶ場合は「最良降下法」(best descent method) と呼ばれる。また次の

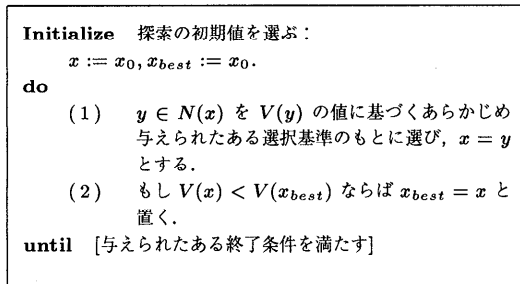


図 2 近傍探索アルゴリズム

Fig. 2 A neighbourhood search algorithm.

ような確率的な方法 (Metropolis 法) もよく知られている: すなわちランダムに選ばれた  $y \in N(x)$  を,  $V(y) < V(x)$  ならば確率 1 で選択し, そうでない場合は

$$P_T(y) = \exp\left(-\frac{\Delta V}{T}\right) \quad \text{ただし } \Delta V = V(y) - V(x) \quad (1)$$

で定義される確率  $P_T(y)$  で選択する. ここで  $P_T(y)$  は受理確率と呼ばれ, たとえばシミュレーテッドアニーリング (SA) では<sup>1)</sup>,  $T$  の値がアニーリングスケジュールに従って, 繰返し回数の増加とともに減少する (このとき  $T$  を温度パラメータと呼ぶ).

本論文では次に述べる 2 つの理由から, 主に確率的な探索を用いる. 理由の 1 つは確率的な探索のほうが一般に性能が良く, より実用的であるためである. もう 1 つは, 仮に単純な最急降下法を用いた場合, 局所解の数が爆発的に増加し, 解析が不可能であること, しかもこれら局所解のなかには比較的「浅い」局所解も「深い」局所解も区別なく含まれてしまうことである. ただし, ここで図 2 のような最小化における局所解の「深さ」とは, その局所解が解の悪化を許す最低  $N$  回の遷移によって回避できる場合の  $N$  の大きさのこととし,  $N$  が比較的小さいとき「浅い」,  $N$  が大きいとき「深い」ということにする. 比較的深い局所解は, 短期の確率的局所探索の結果得られる最良解であり, その深さは用いる温度パラメータ  $T$  に依存する. 以下では特に断らない限り, 深い局所解を単に局所解と書く.

SA は確率的な局所探索の代表的なものであり, スケジュールリングを含む多くの組合せ最適化に適用され成功を収めている. しかし 1 回の SA は非常に時間がかかるため, 後に説明するような 1 回の試行中何度も繰返し局所探索を行うような設定ではあまり実用的ではない. そこで本論文では SA において温度パラメータを定数に制限する ( $T = c$ ) 方法を用いることにす

る. 本来なら定数  $c$  は問題に応じて適応的に求めるべきであるが, 後述のように, 本論文では簡単のため予備実験に基づく経験的な値を採用する.

### 3.3 数値実験

クリティカルブロックと代表近傍による確率的な近傍探索を用いた場合の PFSP の地形を実験によって解析し, 大谷構造が存在するかどうかを調べる. ここで確率的な近傍探索は式 (1) の受理関数で  $T = c$  としたものを用いる. PFSP に対してより単純な近傍探索を用いた場合の同様な解析結果については文献 9) を参照されたい.

真の最適解が明らかになっている問題に対して, 大谷構造が存在するかどうかを調べるためには次のようにすればよい. すなわち, ランダムな局所最適解を  $N$  個生成し, 各局所最適解から他の  $N-1$  個の局所最適解までの距離の平均 MEAND と, 最も近い真の最適解までの距離 BESTD を計算し, 目的関数 OBJFN との間の相関を見る. この距離としては本来ならば先に説明した  $d_N$  を用いなければならないが, ここで取り上げる PFSP に対しては  $d_N$  の近似として先行関係による距離  $d_{pre}$  を用いることにする.  $d_N$  と  $d_{pre}$  の関係については後で論ずる. ここで取り上げた問題は Tallard のベンチマークとして知られるもののうち比較的規模の小さい 2 問題 ta011 (20 仕事 10 機械) と ta021 (20 仕事 20 機械) である. 両問題に対し図 2 で述べた近傍探索を, 繰返し回数  $L = 5000$ , 式 (1) の受理確率  $P_T = P_c (c = 5)$  とし乱数の初期値を変えながら  $N = 2500$  回適用した. その結果 ta011, ta021 のそれぞれに対し 1841, 2313 個の異なる局所最適解を得た. 今回の地形の解析には真の最適解が必要であるが, ta011, ta012 に関してそれらは既知であり,  $L$  を大きくとり, ある程度時間をかけた探索によって求めることができる. 時間をかけ繰返し行った予備実験の結果, ta011 問題の最適解は頻繁に発見されるものの, 異なる解は 2 つだけであり, 両者はとても類似していることが分かった ( $d_{pre} = 2$  であった). また同様に ta021 の最適解は 1 つだけしか発見されなかった. これはこの 2 つの問題に関しては異なる真の最適解は非常に少ないことを示唆している. ただし, すべての可能性を列挙しない限り異なる真の最適解の個数を結論付けることはできない. 一方で同様な実験を ta001 (20 仕事 5 機械) について行ったところ, 同程度の計算時間で 2500 個以上の異なる最適解が発見された.

図 3 において  $x$  軸は (a) 各局所最適解から他の局所解への  $d_{pre}$  の平均 (MEAND) および (b) 最も近

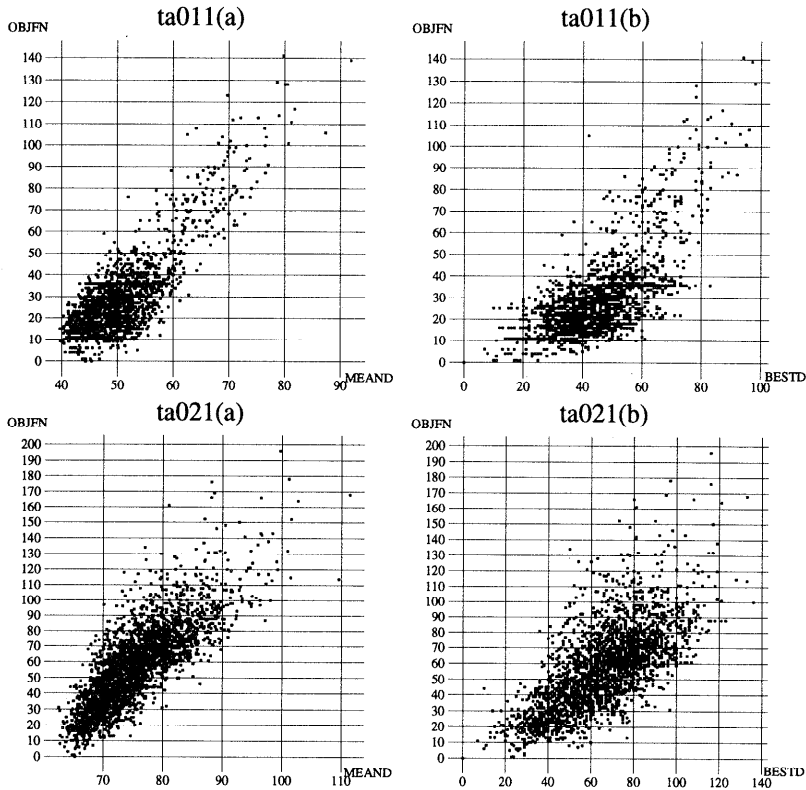


図 3 ta011 (20×10), ta021 (20×20) 問題に対してそれぞれ 2500 回の短期局所探索を行い合計 2313, 2313 個の異なる局所解を得た。これらを (a) 他の局所解からの距離の平均および (b) 真の最適解からの距離を  $x$  軸とし、目的関数の値 (真の最適解からの差) を  $y$  軸として描画した。

Fig. 3 1841 distinct local optima obtained from 2500 short term local search for the ta011 (20×10) problem and 2313 distinct local optima for the ta021 (20×20) problem are plotted in terms of (a) average distance from other local optima and (b) distance from global optima ( $x$ -axis), against their relative objective function values ( $y$ -axis).

い真の最適解までの距離 (BESTD) を表し、 $y$  軸は真の最適解の値を 0 としたときの各局所最適解の目的関数  $C_{max}$  の相対値である (OBJFN)。これらのグラフによってそれぞれの距離 MEAND, BESTD および OBJFN の間には明らかに正の相関が存在することが分かる。それぞれについて相関係数を計算したところ、ta011 (a): 0.74, ta011 (b): 0.50, ta021 (a): 0.62 and ta021 (b): 0.44 であった。これらの高い相関から、探索空間内で局所最適解は真の最適解を中心として放射線状に分布しており、真の最適解からの距離が遠くなるほど局所最適解の質が悪化する構造を持つと考えられる。したがって、ある局所解から少しだけ優れた次の局所解へと、1 カ所に捕捉されることなく段階的に移動しながら探索することができれば、次第に真の最適解に近づくことが可能はずである。

上記の解析では距離として  $d_{pre}$  を用いたが、本来な

らば近傍探索オペレータによって導入される距離  $d_N$  を用いるべきである。しかし実際にはある局所解から真の最適解へ到達する近傍探索オペレータの最小ステップ数を厳密に計算するのは非常に難しい。そこで  $d_N$  の近似として次のような  $d_s$  を実験的に計算した：任意の解  $y$  と真の最適解  $x_{bst}$  に対して  $d_s(y, x_{bst})$  を次のように定義する： $y$  を初期値とし ( $x = y$ )、現在の解  $x$  の近傍  $N(x)$  から  $x_{bst}$  に最も近い解 ( $d_{pre}(z, x_{bst})$  の最も小さい  $z \in N(x)$ ) を選択し、これを現在の解と置き換える ( $x = z$ ) ことを  $x$  が最適解に到達するまで繰り返し、その繰り返し回数を  $d_s(y, x_{bst})$  とする。

$d_s$  は  $d_N$  と必ずしも一致しないが、有効な下限を与えるものと思われる。

図 3 の各局所解について、先行関係による最適解までの距離  $d_{pre}$  (PREC) と近似的に求めた最適解まで

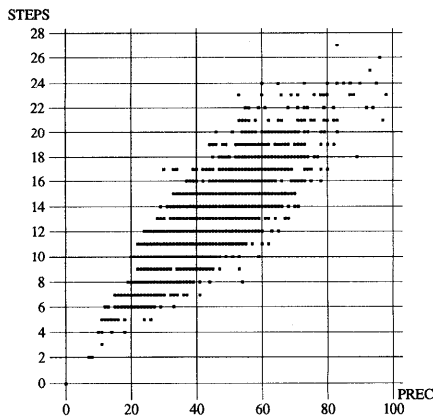


図4 先行関係による距離 (PREC) と最適解へ至る近似的なステップ数 (STEPS) との関係

Fig. 4 The relation between the precedence-based distance (PREC) and the approximate number of steps to the global optimum (STEPS).

のステップ数  $d_s$  (STEPS) との関係を描画したものを図4に示す。この図から両者の間には強い正の相関があることが分かる (相関係数は0.66)。したがって、 $d_{pre}$  は  $d_N$  の近似として十分成り立っていると考えられる。

以上によって少なくとも ta011, ta021 の2問題についてはクリティカルブロックに基づく代表近傍による地形において大谷構造が存在することが実験的に明らかになった。同様な実験は Tallard ベンチマーク中の同サイズ (20 仕事 10 機械および 20 仕事 20 機械) の他のすべての問題について行ったが、ほぼ同様な結果が得られた。

#### 4. 大谷構造と近傍探索

前章の実験によって、本論文で扱う PFSP の地形が少なくとも一部の問題について、大谷構造を持つことが明らかになった。本章では探索空間が大谷構造を持つと仮定したうえでの効率的な探索法について考える。

##### 4.1 経路再結合

仮に PFSP のような組合せ最適化問題と特定の探索オペレータが与えられているとし、それに対する探索空間の地形が大谷構造を持つとする。いま、たとえば品質のほぼ等しい複数の局所解がすでに得られているとすると、前章で述べた正の相関から、これら局所解の比較的近くにこれらより多少とも優れた新たな局所解が存在する傾向があることが分かる。したがって、2つの局所解に対し、一方を出発点とし、そこからもう一方へ至る経路を考えると、これをたどることによってその途中に新たな局所解が得られる可能性が

高い。この探索法自体のアイデアは (大谷構造とは独立に) 「拡散探索」 (scatter search) という名称で Glover によって提案された。提案当初はビット列空間のみを扱っていたが、より一般の空間への拡張として後に「経路再結合」 (path relinking) と呼ばれるようになり、最近では主にタブー探索との組合せで用いられることが多い<sup>4)</sup>。

経路再結合は一般的な考え方を示すもので、実際には両者を結ぶ経路は何通りもあり、またそれをたどる方法も複数考えうる。したがって、これらを状況に応じて個別に定めなければならない。たとえば Rana ら<sup>8)</sup> はビット列空間における「最適結合」 (optimal linking) という方法を提唱している。そこではビット列で表現された2つの解  $y_1, y_2$  に対し、 $y_1$  を初期値とし ( $x = y_1$ )、現在の解  $x$  の近傍  $N(x)$  から終点となる  $y_2$  へのハミング距離  $H(z, y_2)$  が最も小さい  $z \in N(x)$  に移動する ( $x = z$ ) ことを繰り返すことによって経路をたどる。これは  $y_1$  を始点とした単純な最急降下法に、選択基準としてもう一方の解  $y_2$  との類似度  $H(z, y_2)$  を加味したものにとらえることができる。PFSP においては2章で説明した位置関係による距離を用いるのが自然であり、それを用いることによって同様な方法を考えることができる。しかし今回3章で実験に用いたのは確率的な探索であり、経路再結合にも確率的な探索に基づくものを用いる。これによって確率的な探索で回避されていた浅い局所解の影響をここでも受けずにすむ。

先に著者ら<sup>15)</sup> はジョブショップスケジューリング問題 (JSSP) の GLS による解法として MSXF-GA 法を提案した。その中で提案された多段階探索交叉 (MSXF) という交叉オペレータがまさにこの経路再結合の条件を満たす。そこで今回の PFSP にも MSXF を用いることにする。文献15) によって MSXF-GA 法は JSSP に対して有効な解法であることが分かっているが、なぜ有効なのかその仕組みは十分解明されていない。しかも JSSP は PFSP にくらべ問題の構造がより複雑であるため、その解空間の地形を調べるのは難しい。したがって、JSSP と同様な解法によって PFSP が効率的に解けることを確認し、さらにその背景に大谷構造が存在することを明らかにしたい。

##### 4.2 多段階探索交叉

通常、交叉オペレータの役割は積木仮説 (building block hypothesis) との関わりで論じられることが多い。それによると、GA による最適化は、交叉と淘汰を繰り返すことによって解を表現する遺伝子の優秀な組合せが個体集団内に蓄積されることで達成される<sup>5)</sup>。

```

 $d(x, y)$  :  $x, y$  間の距離,  $N(x)$  :  $x$  の近傍,  $V(x)$  :  $x$  の目的関数とし,  $V(x)$  の最小化を考える.  $p_1, p_2$  を与えられた 2 つの解とする.
初期化  $x = q = p_1$  とおく.
do
  (1) 各要素  $y_i \in N(x)$  に対して  $d(y_i, p_2)$  を計算する.
  (2) 各  $y_i \in N(x)$  を  $d(y_i, p_2)$  の値の小さい順にソートし, 番号を付け替える.
  (3) do
    (a)  $i$  の値に反比例する確率で  $N(x)$  の要素  $y_i$  をランダムに選ぶ.
    (b) もし  $y_i$  が評価済みでなければ  $V(y_i)$  を計算する.
    (c) もし  $V(y_i) \leq V(x)$  なら確率 1 で  $y_i$  を受理. そうでない場合は確率  $P_c(y_i)$  で受理.
    (d)  $y_i$  の添字番号を  $i$  から  $n$  に変更し,  $y_k$  ( $k \in \{i+1, i+2, \dots, n\}$ ) の添字番号を  $k$  から  $k-1$  に変更する.
  until [ $y_i$  が受理]
  (4)  $x = y_i$  とおく.
  (5) もし  $V(x) < V(q)$  ならば  $q = x$  とおく.
until [与えられたある終了条件を満たす]
終了  $q$  を出力とする.

```

図 5 多段階探索交叉 (MSXF) による経路再結合  
Fig. 5 Path relinking by MSXF.

したがって、交叉オペレータは、両親の遺伝子的特徴を適切に子に遺伝する、遺伝子レベルでの類似度を保つものでなければならないということになる。議論を探索問題に限ると、遺伝子レベルでの類似度は遺伝子表現された個体間の距離を用いて測定することができる。ここで距離としては個体の類似性を十分反映したものであることが望ましく、たとえばビット列表現の場合はハミング距離、順列表現の場合は 2.2 節で与えられる距離を用いる。また両親の遺伝子的特徴を子に適切に遺伝させるとは、子を探索空間内の両親双方からの距離が比較的小さい領域、すなわち両親の間にある領域から選び出すということにほかならない。したがって、この場合の交叉オペレータの役割は：

(F1) 探索空間内の両親の中間の領域に探索を集中させ、

(F2) この領域の中から優秀な解の候補を選び出す。

という (F1), (F2) に集約されると考えられる。特に (F1) は経路再結合の考え方に非常によく似ている。(F2) で選び出すのは優秀な解の「候補」であって必ずしもつねに優秀である必要はない。いずれにせよ淘汰によってふるいにかけられ、優秀なものだけが残るからである。しかし探索の効率面から考えた場合、候補としてできるだけ優秀な個体が得られる可能性が十分高いほうがよいのは明らかである。

MSXF は上述の考察をふまえて考案された GLS に適した交叉オペレータの一種である。MSXF では一方の親を初期値として、優秀な解の候補を選び出すための短期の局所探索を行う (F2)。ただしここで他

方の親を参照点として用い、探索の方向がその親に近づく方向を優先するようにバイアスをかける (これを navigated local search と呼ぶ) ことによって探索を探索空間内の両親の中間にある領域に集中させる (F1)。ここで近傍探索としては式 (1) の受理関数 (ただし  $T = c$  (定数) とする) を用いた確率的探索が用いられる。MSXF は探索空間の近傍構造と距離さえ与えられればどのような組合せ最適化問題にも適用することができるという利点を持っている。図 5 に MSXF の手続きの概略を示す。ここで図 5 中の  $P_c(y_i)$  は 3.2 節の式 (1) で定義される受理確率において  $T = c$  (定数) としたものである。

このように MSXF は図 2 の近傍探索の拡張として提案されており、従来の交叉オペレータに比べるとむしろ一種の探索オペレータに近い。また、経路再結合の一種と考えることもできる。しかしここでは淘汰などと組み合わせた遺伝的局所探索の枠組みで扱っていること、(F1), (F2) の両特徴を保持していることなどを考慮してあえて交叉と呼ぶ。

#### 4.3 突然変異

経路再結合で 1 つ注意すべきことは  $p_1, p_2$  の距離がはじめから小さい場合、たどるべき経路がほとんど存在しないためこの操作が意味をなさない点である。MSXF ではこのような場合図 5 のステップ 2 を変更し、各  $y_i \in N(x)$  を  $d(y_i, p_2)$  の値の大きい順にソートすることにする。また終了段階において、もし  $q$  が  $p_1, p_2$  よりも改善された場合は  $q$  を、そうでない場合は手続きの過程で得られた  $p_1, p_2$  から最も離れた解を出力するように変更する。変更された新たな手続き



**個体集団の初期化** ランダムな仕事の順列の集合を生成し、これを初期個体集団とする。集団の各要素に対し局所探索を実行する。各個体をその総作業時間の小さいもの順にソートし、順位付けしておく。  
**do** (1) 個体集団より、総作業時間の短い優秀な個体ほど選ばれやすくなるよう、順位の高さに比例する確率でランダムに2個体  $p_1, p_2$  を選択する。  
 (2) 確率  $P_X$  でステップ (2a) か、ステップ (2b) のどちらかを選び実行する。  
 (a) もし  $p_1, p_2$  間の距離 ( $d_{pre}$ ) があらかじめ 定めたある値  $d_{min}$  より小さい場合は、 $p_1$  に対して MSMF を実行し  $q$  を生成する。そうでない場合、 $p_1, p_2$  に MSXF を適用し、新たなスケジュール  $q$  を生成する。  
 (b) 個体  $p_1$  に対し受理確率として 3.2 節の式 (1) で  $T = c$  (定数) とした  $P_c$  を用いた一定期間の近傍探索 (図 2) を行い、その結果新たなスケジュール  $q$  を得る。  
 (3) もし  $q$  の総作業時間が個体集団中の総作業時間の最大値より小さく、個体集団内に同一の総作業時間を持つ個体が存在しなければ、最大値をとる個体を  $q$  で置き換え、全体をソートしなおす。  
**until** [与えられたある終了条件を満たす]  
**終了** 個体集団中最良の個体を出力して終了。

図 6 MSXF-GA による PFSP 解法

Fig. 6 MSXF-GA for PFSP.

を多段階探索突然変異 MSMF (Multi-Step Mutation Fusion) と呼ぶ。この操作は探索の過程における多様性の維持のために役に立つ。また経路再結合の観点からは、MSXF が経路の内挿であったのに対し MSMF は経路の外挿に相当する。MSXF, MSMF のより詳しい内容については文献 15) を参照されたい。

## 5. MSXF-GA のフローショップ問題への適用

4.2 節ですでに説明したように、MSXF オペレータはすでに発見された局所解に基づいて新たな局所解を発見するようにデザインされている。MSXF を用いた探索法である MSXF-GA は、親である 2 つの局所解の中間の領域を重点的に、しかも一方に捕捉されることなく探索し、新たな局所解を発見する。そして大谷構造の仮定のもとでは、これを繰り返すことによって次第に非常に良い解が発見されるという仕組みになっている。そこですでに説明した代表近傍と先行関係による距離  $d_{pre}$  を用いることによって、MSXF-GA を用いた PFSP 解法を提案する。MSXF-GA を PFSP に適用するにあたっては、JSSP に適用した際の考え方をほぼそのまま用いることができる。ここで GA の淘汰のモデルとしては定常状態モデル<sup>12),13)</sup> を用いることとし、集団内より選ばれた各個体は局所探索と MSXF (もしくは MSMF) を確率的に切り替えることによって探索を行う。探索の結果新たに得られた個体は十分優秀でかつ集団内にすでに同一の個体が存在しない場合に限って受理される。図 6 に MSXF-GA の概要を示す。

## 6. 実験結果

前章で説明した MSXF-GA を実際に C 言語で実装し、DEC Alpha 600 5/226 を上で数値実験を行った。ただし実験で用いたパラメータの値はそれぞれ、集団サイズ 15,  $c = 3$ ,  $L = 1000$ ,  $d_{min} = n/2$ ,  $P_X = 0.5$  である。

まず Taillard のベンチマークのうち問題のサイズ比較的小さく、最適解が既知のもの (問題のサイズが  $20 \times 5$  から  $50 \times 10$  までのもの) に対し提案法を適用したところ、これらの問題に対しては最適解の約 1% の範囲内にある高品質の解がほとんど瞬時に求まることが分かった。

そこで最適解が知られていない、より大規模な問題に適用した結果を表 1 に示す。なお表中 Nowi 欄には比較のための Nowicki らのタブー探索による結果<sup>7)</sup> を示す。また同時に lb 欄, ub 欄にはそれぞれ OR-library<sup>2)</sup> に掲載されている理論的下界および現在知られている最も優れた解 (主に分枝限定法によって得られたもの) を示す。各問題について乱数の初期値を変えて計 30 回の実験を行った。各実験の終了条件は繰返し回数が  $L = 700$  に達した時点で終了とした。これは CPU 時間に換算すると  $50 \times 20$ ,  $100 \times 20$ ,  $200 \times 20$  問題のそれぞれに対し約 12, 21, 47 分であった。

表中、特に  $50 \times 20$  問題では、ほとんどの問題に対して文献 7) の結果を上回っているのみならず、太字で示されたいくつかの結果に対しては OR-library に登録されている最良解よりも優れていることが分かり、注目に値する。一方、これより規模の大きな  $100 \times 20$ ,  $200 \times 20$  問題の結果は  $50 \times 20$  問題の結果ほどは良くないが、MSXF の考え方が大谷構造のもと、遺伝的アル

表 1 Taillard によるベンチマーク問題による結果  
Table 1 Results of the Taillard benchmark problems.

50×20	best	avg	std	nowi	lb - ub
1	<b>3861</b>	3880	9.3	3875	3771-3875
2	<b>3709</b>	3716	2.9	3715	3661-3715
3	<b>3651</b>	3668	6.2	3668	3591-3668
4	<b>3726</b>	3744	6.1	3752	3631-3752
5	<b>3614</b>	3636	8.9	3635	3551-3635
6	3690	3701	6.7	3698	3667-3687
7	3711	3723	5.6	3716	3672-3706
8	<b>3699</b>	3721	7.7	3709	3627-3700
9	3760	3769	5.2	3765	3645-3755
10	3767	3772	4.2	3777	3696-3767
100×20	best	avg	std	nowi	lb - ub
1	6242	6259	9.6	6286	6106-6228
2	6217	6234	8.9	6241	6183-6210
3	6299	6312	7.8	6329	6252-6271
4	6288	6303	2.7	6306	6254-6269
5	6329	6354	11.3	6377	6262-6319
6	<b>6380</b>	6417	12.7	6437	6302-6403
7	6302	6319	11.0	6346	6184-6292
8	6433	6466	17.3	6481	6315-6423
9	6297	6323	11.4	6358	6204-6275
0	6448	6471	10.6	6465	6404-6434
200×20	best	avg	std	nowi	lb - ub
1	11272	11316	20.8	11294	11152-11195
2	11299	11346	21.4	11420	11143-11223
3	11410	11458	25.2	11446	11281-11337
4	11347	11400	29.9	11347	11275-11299
5	11290	11320	16.6	11311	11259-11260
6	11250	11288	23.4	11282	11176-11189
7	11438	11455	9.3	11456	11337-11386
8	11395	11426	16.4	11415	11301-11334
9	11263	11306	21.5	11343	11145-11192
10	11335	11409	31.3	11422	11284-11313

best, avg., std.: our best, average and standard deviation of makespan values

nowi: results of Nowicki and Smutnicki

lb, ub: theoretical lower bounds and best known makespans taken from OR-library

ゴリズムの性能向上にかなり貢献するという我々の仮説を支持するには十分である。なお 200×20 問題の結果がそれほど良くない理由の 1 つは、代表近傍計算の負荷大きくなるためと考えられる。ちなみに Nowicki らは仕事数と機械数の比  $n/m$  が大きい ( $n/m > 3$  である) 問題に対して代表近傍を用いることを断念し、より単純な近傍を用いている (すなわち、単に各仕事をそれが属するクリティカルブロックの先頭もしくは最後に移動するタイプの近傍を用いている)。また彼らは近傍内のすべての要素を評価する効率的な計算法を提案しているが、これはタブー探索には有効であっても我々の方法には直接適用可能でないため提案法では採用していない (より詳しい内容は文献 7) を参照のこと)。これらの問題に対してさらに良好な結果を得るのは今後の課題である。

所要時間を約 4 倍に増加させて再び各 50×20 問題

に提案法を適用した結果を表 2 に示す。今回は各問題に対して乱数の初期値を変えて計 10 回ずつ実験した。表から、計算時間の増加によって全体として得られる解の品質はさらに向上することが分かる。実際 6 つの問題に対して既知の最良解の記録を更新しており (したがって、合計 7 つの新たな最良解が得られた)、最良解が得られなかった問題に関してもその差はたかだか 1 単位時間にすぎない。また平均値も最良値からそれほど離れてはおらず、標準偏差も小さいことから、提案法が解法としての安定性に優れていることが分かる。これは比較的時間をかければ 1 回だけの実験で、非常に品質の良い解が得られる可能性が高いことを示唆するものである。

## 7. 結 論

PFSP に対しクリティカルブロックに基づく代表近

表 2 50×20 問題に対する, さらに時間をかけた実験結果  
(approx. 45 mins.)

Table 2 Results of longer runs (approx. 45 mins.) for the  
50×20 instances.

No.	best	avg.	std.	nowi	lb - ub
1	<b>3855</b>	3863	4.6	3875	3771-3875
2	<b>3708</b>	3713	2.3	3715	3661-3715
3	<b>3647</b>	3657	5.5	3668	3591-3668
4	<b>3731</b>	3735	2.9	3752	3631-3752
5	3614	3621	5.6	3635	3551-3635
6	<b>3686</b>	3692	5.0	3698	3667-3687
7	3707	3713	3.9	3716	3672-3706
8	3701	3712	6.6	3709	3627-3700
9	<b>3743</b>	3761	8.1	3765	3645-3755
10	3767	3768	1.0	3777	3696-3767

傍を用いた確率的近傍探索を行う場合に対して, 解空間の地形の構造を数値実験によって解析した. Taillard による 20×10 および 20×20 ベンチマーク問題を用いた数値実験の結果, 解空間の地形には局所解の解の品質と真の最適解までの距離, および他の局所解までの平均距離に正の相関がある, 「大谷構造」と呼ばれる大局的な構造が存在することが分かった. さらに「大谷構造」を生かした探索オペレータの利用によって, 得られる解の品質がいかに改善されるかを見るため, JSSP の GLS 解法として先に提案された MSXF を用い, クリティカルブロックに基づく代表近傍と組み合わせることによる PFSP 解法: MSXF-GA を提案した. MSXF-GA を Taillard のベンチマーク問題のうち 50×20, 100×20, 200×20 の各問題に適用したところ, そのうちのいくつかに対して最良解の上限を更新することができ, 提案法の有効性を検証することができた.

今後は得られる解の品質を落とすことなく計算時間を短縮することを目指したい. また, 他のタイプのスケジューリング問題 (たとえば各仕事の完了時刻の和を最小にするフローショップ問題:  $n/m/P/\sum C$  など) にも適用し, 提案法の適用性を検証したい.

### 参 考 文 献

- 1) Aarts, E.H.L. and Korst, J.H.M.: *Simulated Annealing and Boltzmann machines*, Wiley, Chichester (1989).
- 2) Beasley, J.E.: OR-Library: Distributing test problems by electronic mail, *E. J. of Oper. Res.*, Vol.41, pp.1069-1072 (1990).
- 3) Boese, K.D., Kahng, A.B. and Muddu, S.: A New Adaptive Multi-start Technique for Combinatorial Global Optimization, *Operations Research Letters*, Vol.16, pp.101-113 (1994).

- 4) Glover, F. and Laguna, M.: Tabu Search, Chapter 3, *Modern Heuristic Techniques for Combinatorial Problems*, Reeves, C.R. (Ed.), Blackwell Scientific Publications, Oxford (1993). Recently re-issued by McGraw-Hill, London (1995).
- 5) Goldberg, D.: *Genetic Algorithms in Search, Optimization, and Machine Learning*, Addison-Wesley, Reading, MA (1986).
- 6) Jones, T. and Forrest, S.: Fitness Distance Correlation as a Measure of Problem Difficulty for GAs, *6th ICGA*, pp.184-192 (1995).
- 7) Nowicki, E. and Smutnicki, C.: A fast tabu search algorithm for the permutation flow-shop problem, *E. J. Oper. Res.*, Vol.91, pp.160-175 (1996).
- 8) Rana, S. and Whitley, D.: Bit representation with a twist, *7th ICGA*, pp.188-195 (1997).
- 9) Reeves, C.R.: A genetic algorithm for flow-shop sequencing, *Computers & Operations Research*, Vol.22, pp.5-13 (1995).
- 10) Reeves, C.R.: Landscapes, operators and heuristic search, *Annals of Operations Research* (1998), to appear.
- 11) Reeves, C.R. and Höhn, C.: Are Long Path Problems Hard for Genetic Algorithms?, *4th International Conference on Parallel Problem Solving from Nature*, pp.134-153 (1996).
- 12) Syswerda, G.: Uniform crossover in genetic algorithms, *3rd ICGA*, pp.2-9 (1989).
- 13) Whitley, D.: The genitor algorithm and selection pressure: Why rank-based allocation of reproductive trials is best, *3rd ICGA*, pp.116-121 (1989).
- 14) Wolpert, D.H. and Macready, W.G.: No Free Lunch Theorems for Search, Santa Fe Institute Report, SFI-TR-95-02-110, Santa Fe Institute (1995).
- 15) 山田武士, 中野良平: 遺伝的局所探索によるジョブショップスケジューリング問題の解法, 情報処理学会論文誌, Vol.38, No.6, pp.1126-1138 (1997).

(平成 10 年 2 月 2 日受付)

(平成 10 年 5 月 8 日採録)



山田 武士 (正会員)

昭和 39 年生。昭和 63 年 3 月東京大学理学部数学科卒業。同年 NTT 入社。現在、NTT コミュニケーション科学研究所所属。主として遺伝的アルゴリズム、シミュレーテッドアニーリング等の研究に従事。平成 8 年 9 月より 1 年間、英国コヴェントリー大学客員研究員。人工知能学会、電子情報通信学会各会員。



**Colin R. Reeves** is a senior lecturer in Operational Research in the School of Mathematical and Information Sciences at Coventry University. His main research interests are in pattern recognition, in combinatorial optimization in general, and in heuristic methods in particular. He edited and co-authored the recently published book "Modern Heuristic Techniques for Combinatorial Problems".

---