適応型シミュレーテッド・アニーリングによる任意形状部品の最適配置†

高原 茂幸*・宮本 定明**

Optimal Allocation of Irregular Shapes Using Adaptive Simulated Annealing

Shigeyuki TAKAHARA* and Sadaaki MIYAMOTO**

This paper proposes a method of adaptive simulated annealing that can effectively be used for solving combinational optimization problems. This method is based on simulated annealing and performs an adaptive search but without knowledge base. It recognizes characteristics of a problem from search history and adaptively changes its search strategies. The method thus performs search in the neighborhood not randomly but by making use of the search history. This method is applied to an optimal allocation problem of irregular shapes. This problem arises in many industrial production processes, e.g., sheet metal process and cloth cutting process. The problem is known to be very difficult, since many irregular shapes must be handled. A new algorithm for locating irregular shapes is proposed, which is based on people's way of thinking in placing such shapes. Simulation examples show that this algorithm provides natural and good locations for the shapes. The adaptive simulated annealing and the other meta-heuristics, i.e., local search, simulated annealing, genetic algorithm and tabu search, are compared using simulation experiments. The result shows that the adaptive simulated annealing is better than the other meta-heuristics.

Key Words : combinatorial optimization, adaptive simulated annealing, nesting problem, CAD, meta-heuristics, NP hard

1. はじめに

組合せ最適化問題において、厳密な最適解を求めることは組合せの爆発などの問題があり非常に困難である。よって列挙的手法を用いるのではなく、メタヒューリスティックと呼ばれる探索の手法が注目され、用いられている。これらの手法としては、シミュレーテッド・アニーリング (SA) ¹⁾、タブー探索 (TS) ^{2),3)}、遺伝アルゴリズム (GA) ⁴⁾ や、もしくは SA, TS, GA をハイブリッド化した手法 ^{5),6)} などがみられる。これらはかなり効果的な手法であり、大域的探索と局所的探索を行うことによって、有効な解を導出する。これまでに、これらメタヒューリスティックの手法を比較した報告もなされており、概して SA と TS の評価が高い ^{7),8)}。この中で、SA は確率の変動を許し、効率的に探索を行う単純かつ有効な手法である。しかし、探索ステップ数が増えてくると局所探索的な要素が強くなり、探索効率の低下が起こる。また、TS や GA は様々なオプションを利用することで効率的な探索を実現する手法である。しかし問題に対するコーディングが難し

く、有効な探索を実現することが困難である。その理由の 1 つに、これらの探索は与えられた問題の性質を利用した探索とはなっておらず、どのような問題に対しても同様のアプローチしか取れない手法であることが挙げられる。そのため、SA にしても GA にしても探索ステップ数が増えると探索効率の低下が見られていた。特に GA はそのためにいかに多様性を持たせるのが重要な課題となっている。

そこで本稿では、探索の手法であるメタ戦略の中で、コーディングの比較的簡単な SA をベースにして、知識ベースを用いることなく、かつ様々な問題への適応性を持たせた、適応型 SA を提案する。この方法では、与えられた問題の性質を過去の探索から判断し、探索戦略を適応的に変えていく方法をとる。これにより、問題に依存せず、なおかつその特徴を生かした有効な探索が可能となる。その有効性を、SA, TS, GA などの他のメタ戦略手法と比較したシミュレーションによって示す。

本論文では、この適応的 SA を最適配置問題に適用する。最適配置問題とはネスティング問題とも呼ばれ、ある決められた複数の部品をプレート上に配置する際、プレートの利用率を最適にする問題である。このような問題は、カッティングを必要とする板金加工や裁断などの様々な産業工程において現れる。この問題は様々な形状の部品が対象であるため、部品数が多くなるとやはり組合せが爆発的に増える。よって解決策として、これまでにメタ戦略を用いたアプローチが取ら

† 第 37 回計測自動制御学会学術講演会で発表 (1998・7)

* 香川県工業技術センター 高松市郷東町 587-1

** 筑波大学電子情報工學系 つくば市天王台 1-1-1

* Industrial Technology Center, Kagawa Prefectural Government, Takamatsu

** Institute of Information Science and Electronics, University of Tsukuba, Tsukuba

(Received January 12, 1998)

(Revised July 3, 1998)

れている^{9)~11)}が、それらが配置の対象としている部品は矩形に限定したものが多く、現実的な問題に対して役に立つようなものではない。また、任意形状の部品配置が可能な例としてヒューリスティックな手法を用いた例^{12),13)}があるが、これはユーザーの仕様に応じて知識ベースを更新する必要があるため、問題への依存度が高くなり、問題によって得られる解の良否が分かれる。

本稿で提案する最適配置手法は、任意多角形状部品の配置順序が定められたとき、その部品の配置位置を決定するアルゴリズムであり、一般的に人が部品を配置する場合にどのように試行錯誤しながら、部品配置していくのかをもとにして考えたアルゴリズムである。基本的に人は、部品の凸部と凹部を組み合わせるように配置をしていくものと考えられる。これと同様の手順で探索を進めることで、複雑な任意形状部品の配置問題に対しても、人の配置に近い結果が得られる。このことについても、シミュレーションにより示す。

2. 適応型シミュレーテッド・アニーリング

一般的にメタ戦略で扱われる問題の解の表現形式は、

$$x = (i_1, i_2, \dots, i_k, \dots, i_n) \quad (1)$$

となる。この中の要素は文字列や数値列であるが、基本的に順序列として表現される。この順序列によって表される順序 x , y ($x, y \in \rho$) に様々な戦略を用いて解集合 ρ 内で変化させることによって、目的関数 f を最小、もしくは最大とするような最適解 p が探索されるが、要素 i_k はおのおの実問題における対象を表現している。例えば、スケジューリング問題であれば、それは行わなければならない”仕事 (work)”であり、巡回セールスマン問題であればそれは巡回しなければならない”都市 (city)”である。

通常、SAやTSなどでは探索を局所化するために近傍 $N(x)$ の定義を行う必要がある。一般的に、近傍としては順列操作である挿入近傍や交換近傍¹⁴⁾などを用い、要素はランダムに選ばれるが、ここではその近傍内操作をランダムに行うのではなく、過去の探索履歴から各要素の問題における重み付けをし、それを参照することで適応的に探索を行う手法を考える。基本的な探索の進め方はSAをベースとする。この簡単な概念図をFig.1に示す。このように、この手法では通常の状態と適応の状態の2つの状態が存在する。この状態は、探索における1つのステップに対応しており、1ステップ間は同一の状態に留まる。Fig.1ではどちらの状態にも探索解を決定するコントローラが存在するが、通常の状態では特に指標はなくランダムに探索解を決定する。しかしこのコントローラは、ある条件が満たされると、過去の探索履歴の情報が蓄積されている適応度評価テーブルを参照することで探索解を決定する適応の状態へ遷移する。1ステップ間そのような状態に留まると、また通常の状態へ戻る。通常の状態では、ある条件が満たされるまで数ステップの間留まる。また、この間に目的関数の値を参照することにより評価テーブルが更新される。

2.1 適応度評価テーブル

各要素 i_k が目的関数に与える影響度を決定するために、それまでの解探索によって得られた情報を保持する必要がある。そのために、適応度評価テーブル ET を用意する。それぞれの要素 i_k の評価は $ET[i_k]$ によって与えられる。ここでは、適応度の高い要素ほど問題の目的関数に対する影響度が高いものと考ええる。また一般的に、初期状態ではどの要素も目的関数に与える影響度は均一であるとするが、何らかの条件が与えられているときには、それをもとに初期化を行う方法もある。

2.2 テーブルの更新

SAの探索過程(通常の状態)において、コントローラによってランダムに選択された探索解の目的関数による評価をもとにしてテーブルの更新を行う。更新の方法は様々考えられるが、次の2点は有効な情報となる。

- 1) 暫定解が更新されるようにそれまでにない有効な解が得られた場合には、その操作に関与した要素の適応度を変化させる。
- 2) 現在得られている暫定解と同じ解が得られるような場合には、その操作に関与した要素の適応度を変化させる。

このように解が更新された時の情報や解の結果から、各要素の特性を判断するようにする。また、極端に悪い結果を与えるような操作に対しても、その要素の適応度を変化させることも考えられる。この手法については第5節でいくつかの異なる手法を考える。

2.3 適応的探索への動作条件

通常の状態から適応の状態へ遷移することがこの手法の重要な部分である。適応の状態ではこれまで定義した適応度評価テーブル ET をもとにして、適応的探索を行うのであるが、その動作を起こすための条件を決める必要がある。そのための条件として、主だった方法として次の2つが考えられる。

- 1) 定期的に適応的探索を行う。

Fig.1 Conceptual diagram of adaptive SA

2) ある条件が満たされれば、適応的探索を行う。

適応的探索に移るタイミングも探索の効率に影響すると考えられる。探索に進展が見られなくなった場合などに動作するような条件を決めることが考えられる。

2.4 テーブル参照による確率的探索解決定

適応的状态に遷移すると、適応度評価テーブル ET によって決定された各要素の適応度を参照することによって、コントローラが確率的に探索解を選択する。選択の方法として、次のようなものが考えられる。

1) 昇順選択 要素 i_k を

$$(ET[i_k] - ET[L]) / \sum_j (ET[i_j] - ET[L]) \quad (2)$$

の確率で選択する。ここで $ET[L]$ は最も適応度の低い要素の適応度である。適応度の高い要素が高い確率で選択される。

2) 降順選択 要素 i_k を

$$(ET[H] - ET[i_k]) / \sum_j (ET[H] - ET[i_j]) \quad (3)$$

の確率で選択する。ここで $ET[H]$ は最も適応度の高い要素の適応度である。適応度の低い要素が高い確率で選択される。

このようにして選択された要素を動かす操作を近傍内で行うことによって、探索解を決定する。また、解探索の状況に応じて調整方法を変化させることも考えられる。局所状態に陥り、探索が進まなくなった場合には、適応度の高い要素を優先的に選択し、その状態から抜け出そうとする一方、探索の最終段階では、適応度の低い要素を優先的に選択し、微調整をかけるようなことも考えられる。

2.5 適応型SAのアルゴリズム

以下に適応型SAのアルゴリズムを示す。Pascal風の記述を用いる。ここでの目的は、関数 f を最小化することである。

Step0 : Set ($T_0, T_E, L, \gamma, N, S$);

Init (ET);

$x := \text{select}(\rho)$;

$p := x$;

$t := T_0$;

Step1 : if S is False then begin

{* normal stage *}

repeat

$y := \text{select}(N(x))$;

$\Delta := f(y) - f(x)$;

if $\Delta \leq 0$ then begin

$x := y$;

if $f(y) < f(p)$ then $p := y$;

Update (ET);

end

else if prob ($e^{-\Delta/t}$) then $x := y$;

until L ;

end

else begin

{* adaptive stage *}

repeat

$y := \text{select}(N(x) \text{ referring to } ET)$;

if $f(y) < f(x)$ then $x := y$;

if $f(y) < f(p)$ then $p := y$;

until L ;

end;

$t := \gamma t$;

if $t \leq T_E$ then return p

else begin

Update (S);

end;

ここでは、テーブルの初期化が Init (ET)、テーブルの更新を Update (ET)、適応的探索の状態を S 、その状態の更新を Update (S) で表している。また、select ($N(x)$ referring to ET) で、テーブル ET を参照して探索解を決定する。また、一般的な SA のパラメータとして、凍結温度 $T_E (> 0)$ 、初期温度 $T_0 (> T_E)$ 、反復条件 L 、温度減少率 $\gamma (0 < \gamma < 1)$ とする。

3. 最適配置問題の記述

前章で示した適応的SAを最適配置問題に適用する。ここでは、そのための最適配置問題へのアプローチ方法を述べる。まず、配置する n 個の任意形状部品に対してそれぞれ整数 $1, 2, \dots, n$ を対応づける。そうすると、それらの順列の集合として、

$$\rho = \{\sigma = (i_1, i_2, \dots, i_n)\} \quad (4)$$

が考えられる。この順列 $\sigma \in \rho$ により、その順序で部品を配置していくことで配置のパターンが決定される。つまり、配置位置を決定するアルゴリズムによって解となる順列 σ の評価が行われる。このアルゴリズムによって決定される順列の評価値を $F(\sigma)$ とする。この順列を様々に変えることで、最適な配置を見つけようとするものである。よって、ここでの最適配置問題は、

$$\min_{\sigma \in \rho} F(\sigma) \quad (5)$$

と記述される。よって、この $F(\sigma)$ がこの問題の目的関数となる。一般的には、SA、GA等のメタ戦略を用いて解 σ を変化させ、(4) 式を満たす解 σ の集合 ρ 内での探索を行うことを考えるが、本論文では解 σ の探索手法として適応型SAを用いる。また配置位置決定のアルゴリズムとして、任意形状部品配置が可能なアルゴリズムを用いる。

Fig.2 Illustration for the Irregular shape allocation algorithm

ここで述べるアルゴリズムでは、任意形状部品を2次元平面に配置し、すべての部品を含む1つの矩形の面積が最小となるようにすることを目的とする。より正確に言えば、 σ に対してアルゴリズムを適用したときに、すべての部品を含む最小の矩形の面積を $F(\sigma)$ とし、 σ を変えることによって $F(\sigma)$ を最小化するのである。

4. 任意形状部品の配置

配置する任意形状部品として、ここでは任意の凸多角形状部品について考える。また、配置させる領域は、ある一定の大きさの矩形領域であるとする。ここで提案するアルゴリズムの基本的な考え方は、既に配置されている部品群の凹部分である頂点と、配置しようとする部品の凸部分である頂点とが組み合わせられるような部分を探索することによって配置位置を決定しようとするものである。このとき、評価関数として部品を配置した際に占められる矩形領域の面積を考え、この領域の面積が最小となるように部品の配置位置が決定される。その際に、同じ評価値が得られるような配置が複数存在することが考えられる。ここでは、配置の基準点を定義し、その基準点から配置した部品の最遠点までのユークリッド距離が最も近い位置に配置する。また、その基準点は配置する矩形領域の左下の頂点とする。

4.1 形状情報

部品の配置を行うにあたって、その形状の情報を定義しておく必要がある。ここでは、既に配置されている部品群の稜線の形状情報と配置しようとする部品の外形線の形状情報である。部品群の稜線及び部品の外形線の頂点列を、それぞれ

$$P_j = (A_{j,1}, A_{j,2}, \dots, A_{j,r_j})$$

$$O = (B_1, B_2, \dots, B_m) \quad (6)$$

とすると、それぞれの頂点では、

Fig.3 Layout example1

Fig.4 Layout example2

$$A_{j,k} = (p_{j,k}, \theta_{j,k})$$

$$B_l = (p_l, \theta_l) \quad (7)$$

という情報を持つ。ここで、 $p_{j,k}$, p_l は位置情報であり、 $\theta_{j,k}$, θ_l は角度情報である。これにより、部品の配置状態が把握できる。

4.2 配置位置の探索

形状情報の角度情報をもとにして、稜線の凹部と外形線の凸部の組み合わせを求める。その組み合わせの中で評価関数を最小とするような配置を決定する。このとき、組み合わせられた凹部と凸部の角度情報がそれぞれ $\alpha_{i,j}$, β_k であるとする、次の3つの場合が考えられる。

$$1) \alpha_{i,j} > \beta_k$$

$$2) \alpha_{i,j} = \beta_k$$

$$3) \alpha_{i,j} < \beta_k$$

ここで、1), 3) の場合には組み合わせられた位置で、どのように組み合わせたときに評価関数を小さくできるかについての探索も必要になる。探索手順としては、最初に、部品の干渉が起らないようにある点を基準とした位置決めを行う。そして、その点を中心として回転を行うことによって評価関数が最小となる位置の探索を行う。

これまでに提案されている方法¹⁵⁾では、最初にすべての部品の位置関係が決まっており、そこから各部品の外形線の左側及び下側が接するように位置決めを行っている。これでは、様々な配置パターンを網羅できていないとも言えない。ここで提案している手法では、稜線上を探索することによって、広い探

Table 1 Results of adaptive SA

	Data1					Data2				
	SABM	SABS	SAUM	SADM	SANM	SABM	SABS	SAUM	SADM	SANM
Average	17.150	18.258	19.475	18.293	18.449	18.198	18.785	19.087	18.446	18.841
Best value	14.533	13.375	16.258	16.226	15.078	15.733	16.676	14.536	16.730	16.185
Deviation	1.142	2.028	1.966	1.570	2.074	0.934	1.461	2.403	1.330	1.390

索空間を考えている.

4.3 配置アルゴリズム

以下にその配置アルゴリズムを示す.

- step0 配置順序 $x = (s_1, s_2, \dots, s_n)$ を決定する.
- step1 配置させたい矩形領域において形状情報として初期稜線情報 $P_1 = (A_{1,1}, A_{1,2}, A_{1,3}, A_{1,4})$ を生成する. また, 最大稜線数 $l = 1$ とする.
- step2 配置しようとする部品の外形形状情報として外形線情報 $O = (B_1, B_2, \dots, B_m)$ を生成する.
- step3 稜線の形状情報 $P_i = (A_{i,1}, A_{i,2}, \dots, A_{i,n})$ ($i = 1, \dots, l$) の中での凹部分と部品の形状情報 O の中での凸部分のすべての組み合わせを探索し, 評価関数が最小となる位置を決定する.
- step4 step3により選択された位置に部品を配置する. その配置に伴って稜線の形状情報を更新する. この場合, 中空領域が出現すれば $l = l + 1$ とし, その領域に新たな稜線の形状情報 $P_l = (A_{l,1}, A_{l,2}, \dots, A_{l,n})$ を生成する. 配置しようとする部品がまだ残っていれば step2 へ進む. なければ終了する.

この簡単な流れを Fig.2 に示す. ここでは, 凸多角形状部品のみを配置の対象として考えたが, 凹部分を含んだ多角形部品を対象とするアルゴリズムに拡張するのは容易である. また, この手法を用いて部品を配置した例を Fig.3, Fig.4 に示す. Fig.3 は任意の形状部品を配置した例であり, Fig.4 は同一形状の部品を配置した例である. これらの例からも人の思考に近い配置が実現できていることが分かる.

5. シミュレーション

最適配置問題に対して, メタ戦略として第2章で述べた適応型SAを用い, またその目的関数として第4章で述べた配置アルゴリズムを用いてシミュレーションを行い, その性能の評価を行う.

5.1 適応的SAの設定

本手法では, 適応度評価テーブルの更新方法によって, 適応的動作がうまく機能するかどうかが決まる. ここでは, その更新方法を変えることで最適解探索にどのような影響があるのかについてシミュレーションにより見る. また, 適応的探索の動作条件の違いによる影響も見. 探索に用いる近傍には交換近傍を用いる. テーブル参照による探索解決の方法としては, 昇順選択を用いる. よって, 適応度の高いものが選択される. 以下の場合について比較する.

- SABM: テーブルの更新手法として, 有効な解が得られた場合に, 交換した2つの要素の適応度を1上げ, 暫定解と同じ解が得られた場合には交換した2つの要素の適応度を1下げる. 適応的探索への動作条件として, 暫定解が4ステップ更新されなければ適応的探索に移る.
- SABS: テーブル更新はSABMと同様にし, 適応的探索の動作条件として, 暫定解が2ステップ更新されなければ適応的動作に移る.
- SAUM: テーブルの更新手法として, 有効な解が得られた場合にだけ, 交換した2つの要素の適応度を1上げる. 動作条件はSABMと同様とする.
- SADM: テーブルの更新手法として, 暫定解と同じ解が得られた場合にだけ, 交換した2つの要素の適応度を1上げる. 動作条件はSABMと同様とする.
- SANM: テーブルの更新手法として, 暫定解と同じ解が得られた場合にだけ, 交換した2つの要素の適応度を1下げる. 動作条件はSABMと同様とする.

初期条件としての情報は何もなく, 探索の過程の中で部品の適応度を決定する. よって, 初期状態ではどの部品の適応度も均一であり, 0であるとする. そして, ここで述べている暫定解と同じ解とは, 目的関数値が同じ解のことであり, 配置は異なってもかまわないものとする. また適応的探索における確率的探索解決の方法であるが, ここでは昇順選択を用いる.

5.2 適応的SAの評価

シミュレーションに用いたデータは, 部品数が20で, 違ったデータを2種類用いた. またSAのパラメータとして,

Fig.5 Comparison of meta-heuristics search techniques

Table 2 Comparison of meta-strategies

	Data1							Data2						
	LS	SA	SABM	GA1	GA2	TS	RS	LS	SA	SABM	GA1	GA2	TS	RS
Average	18.887	17.670	17.150	18.849	18.359	19.489	22.796	20.417	19.203	18.198	19.047	18.567	20.324	20.254
Best value	16.664	16.417	14.533	16.688	16.282	16.036	19.045	17.469	17.035	15.733	17.215	16.512	17.163	18.545
Deviation	1.139	0.954	1.142	1.383	1.479	1.808	1.691	1.720	1.502	0.934	1.458	1.521	2.039	1.160

	Data3							Data4						
	LS	SA	SABM	GA1	GA2	TS	RS	LS	SA	SABM	GA1	GA2	TS	RS
Average	17.825	17.491	16.743	18.464	17.940	18.845	19.774	16.773	15.443	14.956	15.696	16.301	15.956	17.573
Best value	14.683	15.526	15.415	16.854	14.783	16.920	18.576	15.124	14.223	14.144	13.486	14.550	12.920	16.067
Deviation	1.705	1.544	0.830	0.857	1.888	1.455	0.808	1.376	0.731	0.677	1.391	0.832	2.047	0.835

	Data5							Data6						
	LS	SA	SABM	GA1	GA2	TS	RS	LS	SA	SABM	GA1	GA2	TS	RS
Average	19.479	17.806	17.357	17.328	18.142	19.572	19.031	19.096	17.127	15.446	16.823	17.711	18.430	18.845
Best value	16.221	16.059	15.183	14.690	17.006	17.449	17.346	17.426	13.717	13.434	14.104	15.927	15.847	16.276
Deviation	1.838	1.014	1.193	1.455	0.879	1.650	1.195	1.498	1.553	1.635	1.745	1.292	2.005	1.260

$T_E=40$, $T_0=1000$, $\gamma=0.85$, L は100回の繰り返しとした。これらの設定で、それぞれの場合で10回のシミュレーションを行う。ここで探索を始める解の初期値はその都度ランダムに生成することとした。そして、得られた最適解の平均値と、それらの中での最良値を見る。また、解探索の効率を見るために、その標本に基づく標準偏差についても見る。得られた結果をTable 1に示す。ここに出ている数値は、配置した結果占められた矩形領域で、部品以外の廃材となる領域の矩形領域に占める割合である(Fig.3, Fig.4参照)。よって、この数値が小さいほうが良い結果である。

この結果を見ると、テーブル更新の方法としては、SABMの方法が最も良い結果が得られた。ここで扱っている部品配置問題では、面積の小さな部品は配置順序を変えても全体に与える影響はほとんどない場合が考えられる。逆に形状の複雑な部品や面積の大きな部品というのは全体に与える影響が大きいのと思われる。ここでは、解の更新があった場合には適応度を上げ、暫定解と同じ解が見つかった時には適応度を下げることでうまく問題に与える影響の大きな要素を捕らえることに成功している。SADMも比較的良好な結果が得られているが、これはあまり影響力のない要素を捕らえ、その周辺を集中的に局所的な探索が行っている結果であろう。また、SAUMはそれほど良い結果は得られなかったが、これは暫定解が更新されるということはそれほど頻繁に起こることではないため、十分な評価テーブルが作成できなかったためだと考えられる。それとは逆にSANMでは、影響が小さいであろうと思われる要素の適応度を下げるだけで十分な評価テーブルを作成できるかどうかを試みたが、結果的にはそれだけでは不十分であることも分かった。

また、適応的探索の動作条件であるが、SABSのように適応的探索を頻繁に行うと、本来のSAの良さが消されてしまい、

偶然良い解を見つけることはあっても、偏差の値を見ても分かるように全体的なパフォーマンスは低下する。本シミュレーションでは、SA全体のステップ数が20であったので4ステップ解の更新がなければ適応的探索を行うように設定したが、問題に応じて妥当な条件を見つける必要がある。

5.3 他の戦略との比較

メタ戦略として様々な手法が存在するが、それらと適応型SAとの比較を行う。ここでは、局所探索(LS)、シミュレテッド・アニーリング(SA)、遺伝アルゴリズム(GA)、タブー探索(TS)、ランダム選択(RS)との比較を行う。部品数20のデータを6種類用意し、5.2節と同様の条件でシミュレーションを行った。LSのパラメータとして、最大探索回数15とし、近傍として交換近傍を用いている。SAのパラメータは5.2節と同様である。そして、GAとして2種類のパターンを試した。その1つであるGA1のパラメータは個体数40、世代数50、交叉確率0.9、突然変異確率0.1とした。もう1つのGA2は個体数、世代数は同じで、交叉確率を0.1、突然変異確率を0.9とした。また、その交叉の操作として順序交叉(order crossover)を取り、突然変異の操作として逆位の操作を取った。そして、エリートイズムの戦略も用いている。TSについては、タブーリストのサイズを3、探索回数15とした。タブーリストの構成は、各ステップで見つけた暫定解を得るためのムーブ(交換した要素)をタブーリストに加える。よってタブーリストのサイズが3であるということは、最大6個の要素がリストに加えられることになる。また、アスピレーション基準も導入している。これによって、タブーリストの中から有効であると考えられるムーブ(要素)を2つリストから外すようにしている。RSについてはランダムに2000個の解を生成し、その中の最も良い解を選択するものとする。また、それぞれの手法で探索を始める解の初

期値であるが、ここでもその都度ランダムに生成することとした。ここで設定しているパラメータ値は、どの手法でも計算時間がほぼ同じとなるようにしている。LSだけは、解が更新されなくなると探索を終了するという定義のため、探索によって計算時間にばらつきが生じている。得られた結果をTable 2に示す。また、それぞれの手法での同じ初期解から探索を開始した場合の解探索の過程をFig.5に示す。この図では、探索の枠組みが似ているSA, LS, TSとSABMとの比較を行っており、見つけた最良解の遷移を示している。LSについては、探索が途中で終了したが、最後に得られた値を延長し表している。

Table 2において、それぞれの手法の有効性を示しているのは見つけた最良解の平均値及びその標準偏差であると考えられる。それを見ると、適応型SA (SABM) が他の手法と比較して最適解探索に最も有効であることが分かる。ほぼどのデータでも一番良い結果が得られている。10回の探索で見つけた最良値は、どのような初期値から出発するかという偶然性もあるのでこれだけで比較はできない。それでも、常に一定のレベルのものは見つけている。これにより、問題によって扱う部品の形状が全く異なっている、その問題に適応した探索が実現できていることが分かる。

また、Fig.5を見ると、SABMにおいて定常状態に陥った場合に適応的探索が有効に機能していることも分かる。よって、着実に暫定解の更新が行われている。SAであると、定常状態に長期に陥る可能性があり、そこから抜け出すのが困難であることが分かる。

また、タブー探索については、ここではそれほど良い結果が得られなかった。TSに対しては、高い評価が与えられているが、ここで全く効果が発揮できなかったのは、タブーリストとアスピレーション基準を導入するだけに止めたからだと考えられる。逆に言えば、タブー探索はそのベースとなる考え方であるタブーリストの利用というものが探索の過程においてあまり機能していないとも言える。

また、意外であったのがRSの標準偏差がどのデータの結果でも小さかったことである。しかし、見つけている最良解という点ではやはり最も悪い結果となっている。

6. おわりに

探索の手法であるメタ戦略の中でも、コーディングの比較的簡単なSAをベースにして、知識ベースを必要とせずに様々な問題への適応性を持たせた適応型SAを提案した。そして、最適配置問題に適用することにより、その有効性を示した。この手法によれば、従来のSAの手法よりもかなりの改善が得られる。またタブー探索と比べて問題に対するコーディングがしやすいというSAの特徴を生かした探索が行えるという利点がある。

また、ネスティング問題を扱う場合、部品を配置する位置選択を決定する必要があるが、これまでにあるような矩形だけの配置を考えるのではなく、任意の多角形状部品までを配置の対象としたアルゴリズムを提案した。これにより、人の思考に

近い配置が実現できることが可能となる。

参 考 文 献

- 1) S.Kirkpatrick, C.D.Gelatt and M.P.Vecchi : Optimization by Simulated Annealing, Science, 220, 671/680 (1983)
- 2) F.Glover : Tabu Search-Part I, ORSA Journal on Computing, 1, 190/206 (1989)
- 3) F.Glover : Tabu Search-Part II, ORSA Journal on Computing, 2, 4/32 (1990)
- 4) J.H.Holland : Adaptation in Natural and Artificial Systems, MIT Press (1992)
- 5) B.L.Fox : Integrating and accelerating tabu search, simulated annealing, and genetic algorithms, Annals of Operations Research, 41, 47/67 (1993)
- 6) 北野宏明 (編) : 遺伝的アルゴリズム, 第3章, 産業図書 (1993)
- 7) E.H.L.Aarts, P.J.M.van Laarhoven, J.K. Lenstra and N.L.J.Ulder : A Computational Study of Local Search Algorithms for Job Shop Scheduling, ORSA Journal on Computing, 6-2, 118/125 (1994)
- 8) 高原、宮本 : メタ戦略を用いたネスティング問題の解法, 計測自動制御学会論文集, 33-11, 1081/1086 (1997)
- 9) D.Smith : Bin Packing with Adaptive Search, Proc. of the 1st International Conference on Genetic Algorithms and Their Applications, Pittsburgh P.A., 202/207 (1985)
- 10) B.Kröger, P.Schwenderling and O.Vornberger : Parallel Genetic Packing of Rectangles, Proc. of Parallel Problem Solving from Nature 1st Workshop, Springer-Verlag, 160/164 (1991)
- 11) 白井、松本 : レイアウト設計のためのメタ戦略によるハイブリッド技法, 日本機械学会論文集 (C編), 63-611, 2337/2344 (1997)
- 12) A.Albano and G.Sapuppo : Optimal Allocation of Two-Dimensional Irregular Shapes Using Heuristic Search Methods, IEEE Trans. Syst. Man Cybern., SMC-10, 242/248 (1980)
- 13) Y.Cai, L.Liu, W.Wang and J.Sun : An Expert System for Automatic allocation of 2D Irregular Shapes, Expert Systems in Computer-Aided Design, Proc. of the IFIP WG, 407/422 (1987)
- 14) 茨木俊秀 : 組合せ最適化とスケジューリング問題 : 新解法とその動向, 計測と制御, 34-5, 340/346 (1995)
- 15) 藤田、赤木、広川 : 遺伝的アルゴリズムと極小値探索アルゴリズムとのハイブリッド化による板取り問題の一解法, 日本機械学会論文集 (C編), 59-564, 312/319 (1993)

「著 者 紹 介」

高 原 茂 幸 (正会員)

1990年, 京都大学工学部数理工学科卒業, 同年香川県入庁, 香川県工業技術センター勤務, 現在に至る。設計における知識処理の応用に関する研究に従事。人工知能学会の会員。

宮 本 定 明 (正会員)

1978年, 京都大学大学院工学研究科博士課程修了, 同年筑波大学技官, 同大学講師, 助教授, 徳島大学教授を経て, 1994年筑波大学教授。ファジィ理論, 情報検索, クラスタ分析, メタ戦略の応用などの研究に従事(工学博士)。システム制御情報学会, 日本ファジィ学会, 情報処理学会, 日本応用数学会などの会員。