

GA による 3 次元箱詰め戦略の自動チューニングと ルールベースの構築†

川上 敬^{††} 皆川 雅章^{†††} 嘉数 侑昇^{††††}

3次元の空間認識を伴う複雑な組み合わせ問題として知られる3次元箱詰め問題を自動的に解くための新しいアプローチを提案する。本論文では、3次元箱詰め戦略が自動的にチューニングされ、最適な箱詰め解が得られるような機構の実現を試みる。この環境適応型の自動チューニング機構を生物の進化システムを模倣したGA(ジェネティックアルゴリズム)の適用により実現する。この機構により、箱詰め性能は徐々に向上し、近最適な戦略が得られる。ここでチューニング対象となる箱詰め戦略は、二つのステップから構成される。①与えられた長方形空間内における配置位置の評価値を評価関数により計算し、その値に従い次の配置位置を決定する。②決定した配置位置に最も好ましい箱を評価関数により決定する。そこで近最適な戦略を得るために2本の評価関数の各重み付け係数の連結により表現されるストリングに対してジェネティックオペレータを適用する。本論文ではジェネティックオペレータとして、再生、乗り換え、突然変異を採用する。また、チューニングにより獲得された戦略を蓄積し、再利用するために、新しく3次元箱詰めルールベースを構築可能とするような方法論についても示す。そして、本方法論に基づき、3次元箱詰めシミュレータと3次元箱詰めルールベースを作成し、数値実験によりその有用性が確認された。

1. はじめに

自動化された工場において、自動箱詰めシステムの構築を考えた場合、解決しなければならない困難な問題のうち最も重要かつ困難な問題は、3次元箱詰め問題の解として導出可能な、詰め込みシーケンスの決定である。本論文で扱う3次元箱詰め問題とは、与えられた入れ物としての直方体(以下、コンテナと呼ぶ)空間内にそれぞれ寸法の異なった品物としての直方体(以下、箱と呼ぶ)群を無効空間領域ができるだけ少なくなるように、効率よく3次元的に配置する問題である。本問題は、3次元の空間認識を伴うため、複雑な組み合わせ最適化問題として知られている。そのため詰め込む箱の種類や個数の増加に伴い、詰め込み作業そのものや、失敗した場合のやり直しには、多大な時間がかかるため、高性能を備えたシステムの出現が望まれている。

従来この問題に対して、数学的なものからプロダクションルールを用いたもの²⁾まで様々なアプローチが行われてきたが、それらのアプローチは各問題設定ご

とにあらかじめ決められた理論やルールを適用するため、問題空間への依存性という点で課題が残る。そのため、完全に自動化された箱詰めシステムへの適用は困難であろう。

そこで、本論文ではこの問題空間への依存性の解消をはかることを目的とする。すなわち、3次元箱詰め問題の解探索のための戦略が、問題空間に適応するように自動的にチューニングされ、近最適な箱詰め解が得られるようなメカニズムの実現を試みる。この環境適応型の自動チューニング機構を生物の進化システムを模倣したGA(ジェネティックアルゴリズム)の適用により実現する。この機構により、箱詰め性能は徐々に向上し、近最適な解が得られるまで、適応動作が繰り返される。ここでチューニング対象となる箱詰め戦略は、箱詰め手順を導出する2本の評価関数により表現される。この評価関数は、各重み付け係数が対応した各評価項目の線形結合として定義され、この各重み付け係数が箱詰め性能に応じて環境に適応するように自動チューニングされることにより、箱詰め戦略を表現する評価関数が、各問題ごとに近最適化される。

また、チューニングにより獲得された戦略を蓄積し、再利用するために、新しく3次元箱詰めルールベースを構築可能とするような方法論についても述べる。このルールは、入力データの特徴を示す条件部分と箱詰め戦略を示す動作部分の結合により表現され、条件部分のマッチングによりその後の箱詰め作業に適したルールが選択され、適用される。そして最後に、

† Automatic Tuning of 3-D Packing Strategy and Rule-Base Construction Using GA by TAKASHI KAWAKAMI (Department of Management and Information, Hokkaido Women's Junior College), MASAOKI MINAGAWA (Faculty of Social Information, Sapporo Gakuin University) and YUKINORI KAKAZU (Department of Precision Engineering, Faculty of Engineering, Hokkaido University).

†† 北海道女子短期大学経営情報学科

††† 札幌学院大学社会情報学部

†††† 北海道大学工学部精密工学科

本方法論を検証するために、3次元箱詰めシミュレータと3次元箱詰めルールベースのプロトタイプを作成し、数値実験を行い、その有用性を検討する。

2. ジェネティックアルゴリズム

ジェネティックアルゴリズムは、生物システムを持つ組織的進化律による自然選択と自然遺伝学の機構からヒントを得た探索アルゴリズムである³⁾⁻⁵⁾。つまり、集団内に存在する個体群がその集団をとりまく環境に対してそれぞれ何らかの適応値を持ち、その値に従った適者生存とある程度ランダムに行われる情報交換により集団が進化してゆく仕組みで、この進化のために本論文では、ジェネティックオペレータとして再生、乗り換え、突然変異を採用する(図1)。ジェネティックアルゴリズムでは、各個体を数字もしくはアルファベットの列でできたストリングで表現し、それらストリング群が集まった集団が形成される。各ストリングは、環境への適応度、すなわちそのストリングを用いて行われた探索の結果に応じて確率的に淘汰され、生き残ったストリング群のみが次世代に複製される。この過程を再生(Reproduction)と呼ぶ。適応値に応じてストリング群を複製することは高い適応性能を持ったストリングは次世代においても子孫に貢献する可能性が高いことを意味している。より環境に適応したストリングは後続の世代においてもより多くの子孫を複製する。

淘汰の結果生き残ったストリング群には、さらにより良い進化のためにある程度ランダムな操作として、ストリング間の乗り換え(Cross-over)と突然変異(Mutation)を発生させ、その結果作られたストリングを次世代に与える。これは自然界においても高い適応値を持った親同士をかけあわせてできた子供は、より高い適応値を持つ可能性があるという観点から試みられる。

乗り換えオペレータの動作は、ランダムに選ばれた

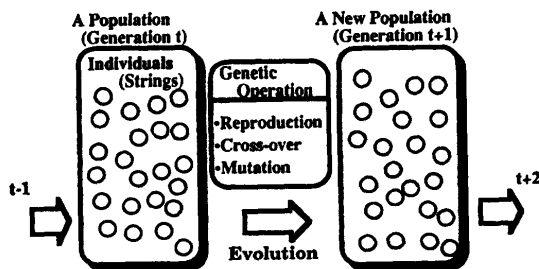


図1 ジェネティックアルゴリズムの概要
Fig. 1 An outline of the genetic algorithms.

2本のストリング間におけるランダムな位置での部分ストリングの交換として行われる。突然変異オペレータについてもやはりランダムに作用する。選択されたストリングに対して、ランダムに突然変異位置が選ばれ、その値が変化する。0と1の2値で表現されたストリングの場合は単に値が反転する。この突然変異は自然界では、ごく希に起こるもので、ジェネティックアルゴリズムにおいてもある小さな確率で起こるように設定しておく。

以上のジェネティックオペレータを各個体の適応結果に従って適用し続け、その度ごとに集団は淘汰され進化し、その世代を重ねてゆく。

3. 問題の記述

ここで本論文で扱う問題の設定を行う。一般に環境適応問題 EAP は次の4項組で表現することができる。

$$EAP = (E, A, G, R) \quad (1)$$

ここに E は環境、 A は環境に対して適応してゆく構造の集合、 G は適応を行うプラン、 R は環境内における異なった構造の適応度をはかる評価基準で、各項目をそれぞれ、扱う幾何形状をすべて直方体とした場合の3次元箱詰め問題に適用すると以下のように設定される。

① E : 3次元箱詰め問題への入力に相当する。すなわち、

$$D_B = \{D_B^i; i=1 \sim N_B\} \quad (2)$$

$$D_B^i = (W_i, L_i, H_i) \quad (3)$$

$$D_b = \{D_b^j; j=1 \sim n_b\} \quad (4)$$

$$D_b^j = (w_j, l_j, h_j) \quad (5)$$

$$N_b = \{N_b^j; j=1 \sim n_b\} \quad (6)$$

ただし、 D_B^i : i 番目のコンテナ、 (W_i, L_i, H_i) : D_B^i の寸法集合でそれぞれ W_i : 幅、 L_i : 長さ、 H_i : 高さを示す、 D_b^j : 詰め込むべき j 番目の箱、 (w_j, l_j, h_j) : D_b^j の寸法集合、 N_B : コンテナの種類数、 N_b^j : j 番目の箱の数、 n_b : 箱の種類数である。

そして、以下では問題を容易にするため $N_B = 1$ とする。また、箱詰め問題を3次元の空間幾何問題として扱うために、コンテナ内の任意の点の座標 (x, y, z) が次式を満足するようにコンテナを X, Y, Z 直行座標系内に設定する。

$$(0 \leq x \leq W). \text{ AND. } (0 \leq y \leq L). \text{ AND. } (0 \leq z \leq H) \quad (7)$$

さらに、各箱のコンテナ内への配置は、箱の各面が必ず X, Y, Z 軸のいずれか一つの軸と直行

するような姿勢をとるものとする。すなわち、一つの箱に対して6種類の可能な姿勢が存在することとなる。

②A: 箱詰め手順を制御する戦略の集合で、各戦略はストリングにより表現される。一つの戦略にはその戦略の実行結果としての箱詰め結果がただ一つ対応する。この箱詰め結果には、配置された箱の総数、箱の配置順序、箱の配置位置が記述される。

③G: Aに対して与えられる適応プランで、ここではジェネティックアルゴリズムに対応し、次に示すRを基準として、各オペレータが適用される。

④R: AにGを適用する上での目的関数で、AはRの値を大きくする方向に進化していくものとする。ここではRを箱詰め結果から導かれる容積率に対応させる。すなわち、

$$R = \left\{ \sum_{j=1}^{nb} f_j \cdot (w_j \cdot l_j \cdot h_j) \right\} / (W \cdot L \cdot H) \quad (8)$$

ここで、 f_j は配置された第 j 種の箱の個数を示す。

4. 箱詰め戦略

次に、箱詰め戦略を直方体群の空間配置として図2に示すように二つのステップに分けて設定する。すなわち、

- S1) コンテナ内に割り当てる配置位置の好ましさを評価関数を用いて定量化し、その値により次の配置位置を決定する。
- S2) 決定した配置位置に対して、未配置の箱の集合から最も好ましい箱の選択を評価関数により決定する。

この二つのステップを再帰的に実行し、問題空間の状態が終了条件を満足するまで順に詰め込んでいく。

4.1 配置位置選択の評価関数

次に箱を配置するに好ましい配置位置を決定するた

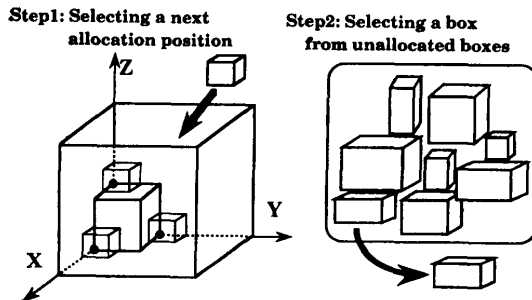


図2 箱詰め戦略

Fig. 2 The strategy of the 3-D packing.

めに、ある配置位置選択のための評価関数を設定し、式(9)で示されるような評価値が割り付けられた配置位置候補の集合 $\{APC_i\}$ の中から選択する。すなわち評価値 P_i は、次式で表現される。

$$P_i = f(APC_i) \equiv f(x_i, y_i, z_i) \quad (9)$$

本論文ではこの評価関数を例えば以下のような評価項目の線形結合で表す。

$$P_i = e_1 \cdot x_i^2 + e_2 \cdot y_i^2 + e_3 \cdot z_i^2 \quad (10)$$

ここで、 e_1, e_2, e_3 は重み付け係数、 x_i, y_i, z_i は配置位置候補 i の座標とする。これによって計算された評価値に対して式(11)を満足するような配置位置候補 P^* を選択する。

$$P^* = \min_{i \in IP} [P_i] \quad (11)$$

ここで、 IP は配置位置候補の番号の集合である。

上記の評価関数は、通常人間が箱詰め作業を行う場合のヒューリスティックな知識を表現するために設定される。すなわち、配置位置に関しては例えば以下のような知識が考えられる。

- コンテナの隅から順に詰めてゆく
- コンテナ内の下の層から積み上げてゆく

つまり式(10)により計算された評価値のうち、最も小さい値を与える配置位置候補が、次の配置位置として選択される。したがって結果としてコンテナ内に設定した座標空間内の原点に近い位置から選択されることになる。

4.2 箱選択の評価関数

決定された配置位置に対して配置させるに好ましい箱を未配置の箱の集合 $\{NAB_i\}$ から評価関数により決定する。すなわち、未配置な各箱に対して式(12)により求められる評価値のうち最大値を与える箱が選択される。

$$B = \sum_{i=0}^{ne} (a_i \cdot b_i) \quad (12)$$

ここで a_i は重み付け係数、 b_i は箱の寸法に関する評価項目、 ne は評価項目の数である。ここでは評価項目数を3とし各評価項目は以下のものを使用する。

- 1) b_1 : 箱の大きさの度合い:
$$b_1 = (w_i/W)^2 + (l_i/L)^2 + (h_i/H)^2 \quad (13)$$

- 2) b_2 : 箱の縦長さの度合い:
$$b_2 = h_i^2 / (w_i^2 + l_i^2) \quad (14)$$

- 3) b_3 : 箱の底面積の広さの度合い:
$$b_3 = (w_i \cdot l_i) / (W \cdot L) \quad (15)$$

ここで、 w_i, l_i, h_i は $\{NAB_i\}$ における第 i 種の

箱の各寸法, W, L, H はコンテナの寸法である.

この評価関数群も配置位置選択と同様, 以下のような配置する箱に関するヒューリスティックな知識を表現するために設定される.

- なるべく大きな箱から詰める
- 底面積の大きい箱から詰める
- 幅の薄い箱は後で空いている隙間に詰める
- サイズが同じ箱は並べて詰める

ただしこれらのヒューリスティックな知識は, 与えられた問題, すなわちコンテナのサイズや箱群のサイズ, 数等によって当然異なると考えられるため, 各評価項目の重み付け係数をチューニングすることを考える.

4.3 配置位置候補の追加と削除

次の箱詰めのために状態空間を変化させ新しい配置空間を生成する. つまり, 箱が一つ配置される度に配置位置候補の集合 $\{APC_i\}$ を更新する必要がある. これを以下の手続きで行う. ただし, 詰め込み開始時には座標原点を唯一の配置位置候補として設定する. したがって, 箱詰め作業において最初に配置される箱は必ず, 座標原点に配置されることになる.

- 1) 1回の配置結果に基づき新しい配置位置候補を作成し集合に追加する. 本論文では, 配置位置 (x, y, z) に寸法 (w, l, h) の箱を配置した場合 (図3), 新しい配置位置候補 NEW_P を,

$$NEW_P = \{(x+w, y, z), \\ (x, y+l, z), \\ (x, y, z+h)\} \quad (16)$$

とし, この3点を集合 $\{APC_i\}$ に追加する.

- 2) すべての配置位置候補に対して, 残存している箱が配置可能かどうかをすべての未配置な箱の種類について調べ, 配置可能な箱の種類が一つも存在しない場合, その候補点を集合 $\{APC_i\}$ から削除する.

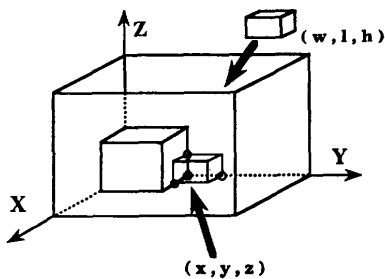


図3 配置位置候補集合の追加と削除
Fig. 3 Altering the set of allocable positions.

4.4 終了条件

箱詰め手順の終了条件として問題空間が次の二つの条件のどちらかを満たした場合に, 詰め込み作業を終了する.

- c 1) 与えられた箱群がすべてコンテナ内に配置され, 未配置の箱が存在しない場合. すなわち,

$$\{NAB_i\} = \phi \quad (17)$$

- c 2) 上記の配置位置候補の削除手続きにより配置位置候補が存在しなくなった場合. すなわち,

$$\{APC_i\} = \phi \quad (18)$$

5. ジェネティックアルゴリズムの適用

ここで, 上記の箱詰め戦略を進化させるために, その手順を制御する2本の評価関数の各重み付け係数をジェネティックアルゴリズムを用いて, 自動チューニングすることを考える. ここでは, 各々の係数に対してそれぞれ4ビット (すなわち, 10進数で0~15を表現) を用意し, それらを順に連結した型で, 1本のストリングを表現する. 1本のストリングは前述の六つの評価項目に対する重み係数を用いて,

$$S = e_1 e_2 e_3 a_1 a_2 a_3 \quad (19)$$

と表され, 各係数は,

$$e_j, a_j = \text{bit}_1 \text{ bit}_2 \text{ bit}_3 \text{ bit}_4 \quad (20)$$

$$(j=1, 2, 3, \text{bit}_i=0, 1)$$

となり, 1本のストリングは24個のビット列で表現されることになる.

このストリングのランダムな集まりとして初期集団 $A(0)$ が構成され, 世代をおうごとに集団が進化し, 世代 t における集団として $A(t)$ が作られる.

$$A(t) = \{S_i\} \quad (21)$$

$$(t=0, 1, 2, \dots, i=1, 2, 3, \dots, \text{pop_size})$$

ここで, pop_size は集団内に設定した個体数である.

本論文で用いているジェネティックオペレータは, 再生, 乗り換え, 突然変異の3種である. 各オペレータのオペレーションを以下に示す.

5.1 再生 (Reproduction) オペレータ

再生は, 前述したように各ストリングを用いて行われた箱詰め結果の容積率 R を評価値として, そこから求められる各ストリングの再生確率の値をもとに行われる.

一つの世代 t において, 各ストリングが実行されそれぞれの容積率 R_i ($i=1, 2, \dots, \text{pop_size}$) を得る. こ

の値に対応して、各ストリングの再生確率は、その世代における最悪の結果との差の 2 乗の値 r_i から計算される。

$$r_i = (R_i - R_{\text{worst}})^2 \quad (22)$$

$$R_{\text{worst}} = \min_{i \in IS} [R_i] \quad (23)$$

ここで、 IS はストリングの番号の集合である。この r_i に従い、各ストリングの再生確率は次式により求めることができる。

$$P_{\text{-repro}_i} = r_i / \left(\sum_{j=1}^{\text{pop-size}} r_j \right) \quad (24)$$

これによりストリング S_i の世代 $t+1$ に対する再生数 $N_{\text{-repro}_i}$ は、

$$N_{\text{-repro}_i} = P_{\text{-repro}_i} \cdot \text{pop-size} \quad (25)$$

となる。

5.2 乗り換え (Cross-over) オペレータ

乗り換えオペレーションは、再生オペレーションの結果生き残ったストリングに対して行われる。この生き残ったストリング群は、オペレーション用のプールの中に入れられ、次なるオペレーションを待っている。この乗り換えはプールの中からランダムに選択された 2 本のストリングに対して作用し、このストリング間のランダムに決められた交叉位置での部分ストリングの交換によって行われる。

本論文では、各世代において乗り換えが行われるストリング数は、集団の 80% に設定した。

5.3 突然変異 (Mutation) オペレータ

突然変異も乗り換えと同様にオペレーション用プールの中で行われるオペレーションである。ランダムに選択されたストリングのランダムな位置における値の変化（この場合、0, 1 値が反転）として作用する。この突然変異は遺伝的アルゴリズムのオペレーションにおいて微量ではあるが、重要な役割を果たしているといえる。再生と乗り換えが、現存の概念の探索と再結合を効果的に行っていたとしても、これらは時として集中的に作用しすぎて潜在的に有用な遺伝情報を失ってしまう。突然変異オペレータはそのような回復不能な損失からシステムを守る働きを担う。本論文において突然変異は 1,000 ビットの転送に 1 回の割合で起こるように設定した。

6. 3 次元箱詰めルールベースの構築

遺伝的アルゴリズムを用いて自動チューニングにより獲得した箱詰め戦略を、その後の箱詰め作業に有効利用するために、新しく 3 次元箱詰めルール

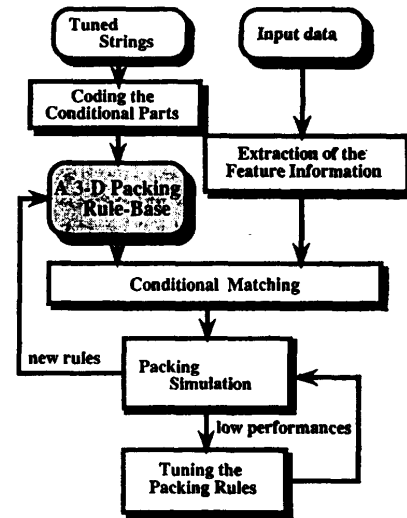


図 4 3 次元箱詰めルールベースの概念
Fig. 4 A concept of a 3-D packing rule base.

ベースの構築を考える。このルールベースは、本問題の環境である入力データから特徴情報を抽出し、その環境に対して高性能を有すると思われる箱詰め戦略を自動的に取り出し、箱詰めシミュレーションを行う機構を持っている (図 4)。ここで扱う箱詰めルールは入力データの特徴を表す条件部分と箱詰め戦略を表す動作部分から構成され、チューニングされたストリングが動作部分に当たる。また、条件部分も特徴情報のコーディングにより表現されたストリングとなる。

6.1 特徴情報の抽出

条件部分のコーディングのため、入力データの特徴情報を抽出する。ここでは、入力データの特徴を抽出するために、コンテナと詰め込む各箱との体積比 RV を考える。

$$RV_i = (W \cdot L \cdot H) / (w_i \cdot l_i \cdot h_i) \quad (26)$$

$$(i = 1, 2, \dots, n_b)$$

そして、この RV を用いた次の 5 項目の値を入力データの特徴情報値とする。

① C_1 : RV の最小値

$$C_1 = \min_{i \in I_{nb}} [RV_i] \quad (27)$$

② C_2 : RV の最大値

$$C_2 = \max_{i \in I_{nb}} [RV_i] \quad (28)$$

③ C_3 : RV の平均値

$$C_3 = \left(\sum_{i=1}^{n_b} RV_i \right) / n_b \quad (29)$$

④ C_4 : 箱群の数と RV の平均値の差

$$C_4 = n_b - C_3 \quad (30)$$

⑤ C_5 : RV の標準偏差

$$C_5 = \left\{ \frac{\sum_{i=1}^{n_b} (RV_i - C_3)^2}{n_b} \right\}^{1/2} \quad (31)$$

ここで、 Inb は箱の種類を表す番号の集合である。

次にこれらの特徴情報値が各ルールごとに条件部分にコーディングされる。このコーディングに際して、上記5項目それぞれが5ビット長の部分ストリングで表現され、5項目分結合した計25ビットでルールの条件部分が構成される。コーディングルールは、次の形で与えられ、それにより、各部分ストリング上のいずれかの位置に特徴情報がマッピングされる。

$$s_{ik} = 1: \text{ if } (k-1) \cdot rc_i \leq C_{ij} - \min C_i < k \cdot rc_i \quad (k \neq 5)$$

$$(k-1) \cdot rc_i \leq C_{ij} - \min C_i \leq k \cdot rc_i \quad (k=5)$$

$$s_{ik} = 0: \text{ otherwise}$$

$$(i=1, 2, \dots, 5, k=1, 2, \dots, 5) \quad (32)$$

ここで、 s_{ik} は第 i 項目を表す部分ストリング上の第 k ビットの値を示し、 C_{ij} ($j=1, 2, \dots, nrule$) はルールベース中の第 j ルールの第 i 特徴抽出項目の値、この $nrule$ はルール数である。また、 $\min C_i$ 、 rc_i は次式で表現される値である。

$$\min C_i = \min_{j \in Jr} [C_{ij}] \quad (33)$$

$$rc_i = (\max_{j \in Jr} [C_{ij}] - \min_{j \in Jr} [C_{ij}]) / 5 \quad (34)$$

ここで Jr はルールベース中のルールの番号の集合である。

6.2 条件マッチング

入力されたデータに対して有効なルールをルールベースから選択するために条件マッチングを行う。この時、入力データもルールの条件部分と同様の計算により、特徴情報を表すストリングにコーディングされている。マッチングは各項目ごとにビット列の突き合わせにより行い、値1を持つビット位置のずれを部分特徴距離とする。各項目ごとに得られる部分特徴距離を5項目分加えた値を入力データと各ルールとの特徴距離とす

る。その結果、特徴距離が最も小さいルールを取り出し、ルールの動作部分を箱詰め戦略として使用する。

この時、条件部分のマッチングが失敗した場合、すなわち適したルールが見つからない場合や選択したルールによる箱詰め性能が低い場合には再度チューニングが行われ、新しいルールがルールベースに追加される。

7. 実験と結果

上述の問題設定、方法論に従って箱詰めシミュレーションシステムを構築し計算機実験を行った。本実験では各世代におけるストリングの集団数 (pop-size) は50個体とした。

表1にサンプルデータによるシミュレーション結果を示し、図5にはデータ8に対するシミュレーション結果をグラフィック出力により示す。

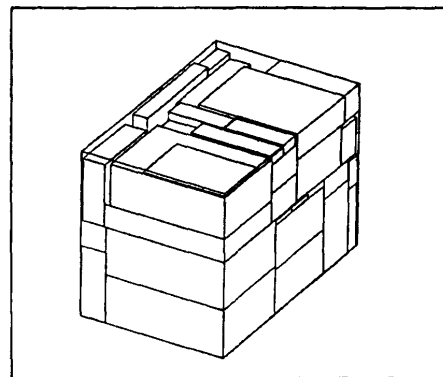


図5 箱詰め結果の図形出力 (データ8)

Fig. 5 Graphical output of the packing result (data8).

表1 箱詰めシミュレーションの結果
Table 1 The simulation results of typical input data.

	The dimensions of a container (W, L, H)	n_b	Minimum (w_i, l_i, h_i)	Maximum (w_i, l_i, h_i)	Rate of capacity (%)	The number of packed boxes
data1	(35, 40, 30)	50	(2, 3, 4)	(19, 20, 17)	92	37
data2	(35, 40, 30)	50	(6, 7, 8)	(19, 20, 17)	86	28
data3	(35, 40, 30)	50	(5, 5, 5)	(14, 15, 14)	84	45
data4	(205, 240, 120)	200	(21, 22, 23)	(39, 40, 37)	82	169
data5	(45, 50, 40)	120	(2, 3, 4)	(19, 20, 17)	94	60
data6	(50, 50, 40)	100	(5, 6, 7)	(19, 20, 17)	87	34
data7	(45, 50, 35)	100	(5, 5, 5)	(14, 15, 14)	85	92
data8	(506, 316, 363)	3,000	(68, 29, 30)	(195, 105, 260)	91	53

7.1 最良ストリングの成長

図6にサンプル中の3種類のデータを入力した場合の箱詰め性能の成長を示す。図中、横軸には世代数、縦軸には最良ストリングによって求められた結果、すなわちその世代における容積率の最大値を示す。この結果から、世代を重ねるに従って、より高い環境適応能力を持つ個体が生まれていることが分かり、ジェネティックアルゴリズムの優れた探索能力が示される。

7.2 集団全体の挙動

集団全体の挙動を見るために、各世代における最良値、最悪値、平均値を図7に示す。これにより集団全体が進化し、適応できない個体が淘汰されていることが分かる。

7.3 特徴距離と箱詰め性能の相関

図8は蓄えたルール群(ルール数30)に対して入力を与え条件マッチングおよび箱詰めシミュレーションを行った結果を示している。図中、横軸はルールと入力データとの特徴距離を示し、縦軸はそのルールによる箱詰め結果(容積率)を表している。これにより特

徴距離がある程度よりも小さい場合(特徴距離 < 7)には、箱詰め性能が良いことが分かり、これからルールの適用可能範囲を測定することができる。

上記の結果に基づき、3次元箱詰めルールベースを試作した場合の結果を図9に示す。本実験では、ルールベース中のルール数を50とし、ルールの発火条件としては、入力したデータに対して特徴距離が5以下のルールのみを抽出し、条件を満たすルールを適用した場合の性能を求めた。ここでの箱詰め結果とは、各入力データに対して抽出されたルール中で最大性能を示したルールの性能をそのデータに対する結果とした。すなわち、あるデータに対して条件を満たすルールが3種類発火した場合、その3種の中で最も高い性能を示すルールによる性能をそのデータに対する結果とする。入力データとしては100種類のデータを試行し、その箱詰め性能に従ってグラフ化した。図中、横軸は各入力データに対する箱詰め性能を示し、縦軸はその結果を示した入力データの数である。この結果から全体の試行数の内の84%が、容積率85%以上の

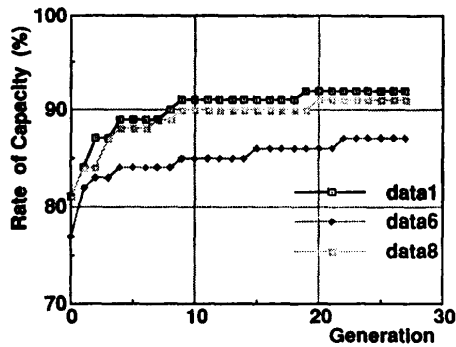


図6 箱詰め性能の成長
Fig. 6 Growth of the packing performances.

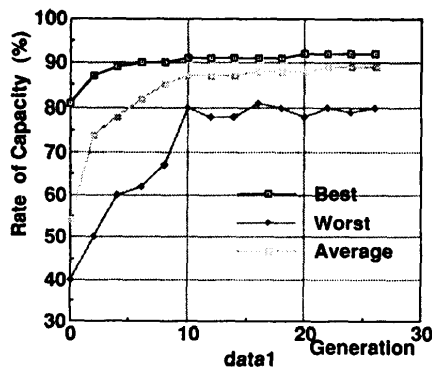


図7 集団の挙動
Fig. 7 The dynamics of the population.

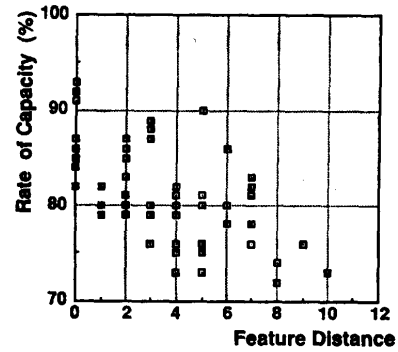


図8 特徴距離と箱詰め性能との相関関係
Fig. 8 The relationship between the feature distance and the packing performances.

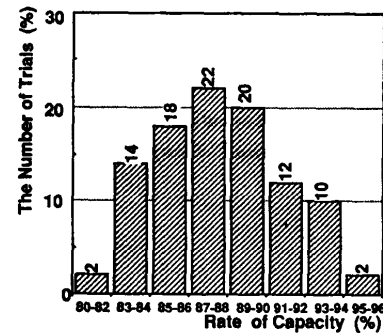


図9 3次元箱詰めルールベースの性能
Fig. 9 The performances of a 3-D packing rule-base.

結果を与えたことが分かる。

8. おわりに

以下をもって本論文の結言とする。

- 1) 3次元箱詰め問題を対象として、ジェネティックアルゴリズムを適用した解探索方法を提案した。
- 2) 箱詰め手順を制御する評価関数の重み係数をストリングで表現し、これにジェネティックオペレータを作用させることにより箱詰め戦略を問題空間向きに進化させる自動チューニングメカニズムを示した。
- 3) 上記方法論により3次元箱詰めシミュレータを作成し、計算機実験によりその有用性を示した。
- 4) 進化した戦略をルールとして蓄え、その後の箱詰め作業に適用可能な3次元箱詰めルールベースの構築可能性を示した。

謝辞 本論文の誤りを指摘し、多くの有益なコメントをいただきました査読者の方々に感謝いたします。

参考文献

- 1) 皆川雅章, 嘉数侑昇: スケジュール問題へのジェネティックアルゴリズムの適用に関する実験的考察, 北大工学部研究報告, Vol. 152, pp. 67-77 (1990).
- 2) Minagawa, M., Kanazawa, R. and Kakazu, Y.: Development of 3D Packing Simulator, *Memoirs of Faculty of Engineering, Hokkaido University*, Vol. 152, No. 4, pp. 501-512 (1989).
- 3) Goldberg, D.E.: *Genetic Algorithms in Search, Optimization & Machine Learning*, pp. 1-25, Addison-Wesley (1989).
- 4) Davis, L. (ed.): *Genetic Algorithms and Simulated Annealing*, pp. 74-88, Pitman (1987).
- 5) Holland, J.H.: *Adaptation in Natural and Artificial Systems*, pp. 20-120, University of Michigan Press (1975).
- 6) Grefenstette, J.J.: A System for Learning Control Strategies with Genetic Algorithms, *3rd International Conference on Genetic*

Algorithms, pp. 183-190, Laurence Erlbaum Associates (1987).

- 7) Whitley, D., Starkweather, T. and Fuquay, D.: Scheduling Problems and Traveling Salesmen: The Genetic Recombination, *3rd International Conference on Genetic Algorithms*, pp. 133-140, Laurence Erlbaum Associates (1987).
- 8) 小坪成一, 須貝康雄, 平田廣則: 遺伝的要素を取り入れた改良型アニーリング法によるブロック配置手法, 電子情報通信学会論文誌, Vol. J73-A, No. 1, pp. 87-94 (1990).

(平成3年6月25日受付)

(平成4年4月9日採録)



川上 敬 (正会員)

昭和38年生。昭和63年北海道大学大学院工学研究科精密工学専攻修士課程修了。現在、北海道女子短期大学経営情報学科講師。機械学習、ジェネティック・アルゴリズム等に興味を持つ。精密工学会会員。



皆川 雅章 (正会員)

昭和56年北海道大学工学部精密工学科卒業。昭和58年同大大学院修士課程修了。昭和61年博士後期課程修了。工学博士。現在、札幌学院大学社会情報学部助教授。生産システム、GAの研究に従事。精密工学会、日本機械学会、日本OR学会各会員。



嘉数 侑昇 (正会員)

昭和16年生。昭和48年北海道大学大学院工学研究科精密工学専攻修士課程修了。工学博士。北海道大学工学部精密工学科教授。CAD/CAM, 画像処理工学, ロボティクス, 知識工学, ニューラルネットワーク等の研究に従事。日本機械学会, 精密工学会, 日本ロボット学会等の会員。