

(39) 遺伝的プログラミングを用いた製品詰合せパターンの生成

A Packing Pattern Creation by using Genetic Programming

中川博之^{*}、渡辺克彦^{*}、赤木宏匡^{**}

Hiroyuki NAKAGAWA, Katsuhiko WATANABE, Hiromasa AKAGI

^{*} 鹿島情報システム部 (〒107 東京都港区元赤坂 1-2-7)

^{**} 鹿島エンジニアリング本部 (〒107 東京都港区元赤坂 1-3-6)

Delivery facilities, in a pharmaceutical industry and so on, have looked for automation methods which make delivery early, cheap and accurately. In this paper, we pay attention to the packing process where ordered products in cardboard boxes are taken, and propose a packing pattern creation method using Genetic Programming (GP) technique.

Because there are many different shapes which are mostly rectangular parallelepipeds in the products, the packing involves maximizing the density needs through trial-and-error and much creative energy. Corresponding the products to terminal nodes and relationships of their locations (for example, left-right, top-down, front-back, etc.) to function nodes, we made a simulation of the packing pattern creation by using GP. We describe a GP applicable method for the packing pattern creation and a result of its simulation.

Key Words: Genetic Programming, Packing, Factory Automation

1. はじめに

医薬品などの物流施設における製品の梱包箱（ダンボール箱）への詰合せ作業をピッキングロボットにより効率化する場合において、ロボットに与えるバランスのよい製品の詰合せパターンを遺伝的プログラミング(GP: Genetic Programming; 以下 GP)により生成する方法について述べる。詰合せパターンとは大きさの異なる様々な製品を梱包箱内のどこに配置するかを示すものである。

さて、よい詰合せパターン即ちバランスがとれダンボール箱にぎっしりと最密充填に近い配置を求める問題は離散的な組合せ最適化問題として表現することができる。これらの問題に対しては、整数計画法を用いて目的関数と制約条件を洗い出し定式化して求める方法や、知識工学手法を用いて人間の経験則をルールとして抽出して推論により配置を求める方法などが適用できる。

但し、今回検討する詰合せパターンの生成では、最終的に配置する製品が数百に及ぶなど組合せ数がかかなり大きくなることが見込まれ、かつ、よい詰合せと評価する制約条件に「開封時に取出し易い」など曖昧さがあり3次元空間内の配置と相俟って一般的な定式化や効率的なルールの抽出が難しいと予想される。

一方、遺伝的アルゴリズム (GA: Genetic Algorithms; 以下 GA) が生物進化の過程を模した離散的最適化問題に対する有効な探索アルゴリズムとして注目されてい

る。GA は確率的探索手法として機能し、準最適解が簡単な定式化により選られることから、大規模かつ複雑な問題への適用が比較的容易である。GP は遺伝子を文字列から木構造に発展させ併せて遺伝的操作も若干変更することで、プログラムの自動生成を可能とするようにGA を拡張したものである。これにより詰合せパターン生成の様な3次元物体の配置の様な複雑な構造を持つ問題にも自然に適用することができる。

本稿では、GP と製品詰合せ問題について概説し、詰合せパターン生成への GP の適用方法とその結果について報告する。

2. 遺伝的プログラミング

ここで今回問題解決に適用する GP の概要について説明する。GP は GA の一種でありその流れは次の通りである（図-1 参照）。

GP では問題の解候補を多種用意する。これを個体と呼ぶが、それぞれの個体は表現型と遺伝子型と呼ばれる二つの状態を持つ。表現型とは与えられた問題に応じた解の形であり、詰合せ問題においては実際に製品がダンボールに詰められた状態を表す幾何的なイメージに相当する。一方、遺伝子型というのは問題を GP で処理するための解の形であり、情報の木構造で表される。例えば、一つの箱に対する詰合せでは2分木を用い、分岐部分（節

点)にそこから分かれる2つの枝の関係(上下、左右、前後など)を表す記号を置き、枝の部分にはそこが葉になる場合は製品を、さらにまた枝となる場合は、そこに再び2分木を置くこととなる。この2分木の表現からなる遺伝子型は、3次元の幾何的なイメージに対応する表現型を十分に表すことができる。GPではまず、問題の解候補となる個体の表現型から遺伝子型を導き、これらを多種生成し初期集団とする(図-1(1))。一般には一定数の個体をランダムに生成し、それを初期集団とする。

初期集団が生成されると、問題ごとに定義された適応度の計算を行う(図-1(2))。詰合せ問題では遺伝子型を実際の詰合せイメージである表現型に翻訳し、その詰合せ方が適切かどうかを評価することとなる。各個体の適応度が評価されると、その値に応じて個体を選択(図-1(3))し、交叉、兄弟交換、突然変異と呼ばれる遺伝的操作が行われる(図-2参照)。交叉とは、選択された2個体の遺伝子型(木構造をしている)の一部分(枝に当る部分)をそれぞれ入れ替え、子となる個体を作る操作である(図-1(4))。兄弟交換とは、ひとつの個体の枝の順序を入れ替える操作であり、突然変異とは確率で一部の個体の遺伝子型を変える操作のことを言う(図-1(4))。交叉、兄弟交換、突然変異のいずれの操作を用いるかは乱数で選択するのが一般的であるが、これらの操作は、より適応度の高い個体の特徴がより多く子孫に反映する機構となっている。戦略によっては、適応度の良い個体をそのまま子供として保存させるような操作も加わる。

これらの一連の操作(図-1(2)～図-1(4))を1世代とし、設定した終了条件を満たすまで世代交代を繰り返す。

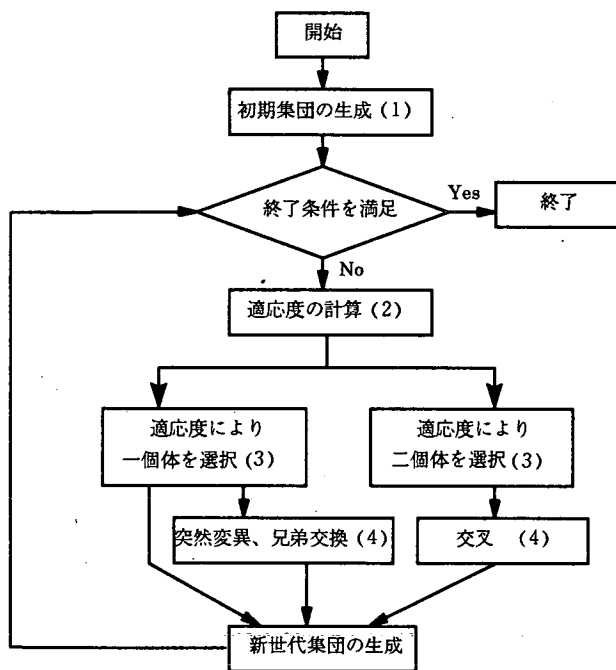


図-1 GPの処理手順

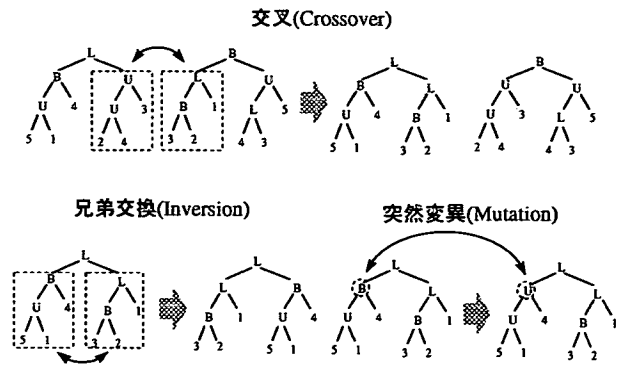


図-2 GPの遺伝的操作

返す。そして、最後まで残った適応度の最も高い個体を解とする。終了条件には世代数を設定するのが一般的である。GPでは評価関数の設定により、計算に要する時間及び解の精度が変動する。

3. 製品詰合せ問題

医薬品などの物流施設においては、発注された製品を得意先に配送する際の詰合せは手作業で行われている。あるケースでは、約400品種、総数約8,000個の製品を約450個のダンボールに詰合せる作業を発送のひとつの単位としているが、これを数人のチームで数時間を要して行っている。個々のダンボールへの詰合せでは、製品を得意先別に分別して詰め込むと共に製品が痛まないようにきれいに並べ、かつ、ダンボールを減らし詰込む時間も節約する効率のよい詰合せを心掛けている。しかし、ダンボールに形状の異なる製品を詰込む詰合せパターンは多彩であり、製品を詰合せする際に何度か詰め換えを行う試行錯誤が発生している。

ここでダンボール数を減らすことだけを目的とする製品の最密充填を求めることは必ずしも最重点課題ではない。しかし、最密充填に近いよい詰合せパターンが予め得られれば、ダンボールの入替えなどの時間を省き結果として時間も節約することが期待できる。

よい詰合せパターンを求めて作業者に提示するだけでも従来の方法に対する改善策とできるが、ピッキングロボットなどの産業ロボットと連携し電子的に得た詰合せパターンをこれに受け渡すことで詰合せの自動化を可能とするところが実用上のメリットとなる。

さて、ここで改めて詰合せ問題を整理する。

- ・ 詰合せ問題とは、形状の異なる複数の製品を定型の複数のダンボールに詰合せる問題とする。
- ・ 詰合せに際して、ダンボールを節約する、取出す側の便を考え同種の製品を近づける、などを根拠に最適化を行う。
- ・ 製品は発送の単位、その中で得意先別に括られ、この単位で問題を考える。製品数は平均して6,70個、ダンボールにして2,3箱である。

- ・ 製品がひとつのダンボールに納まらない場合は複数のダンボールに分けて溢れることなく詰合せる。
- ・ 製品はそれぞれパッケージの形で与えられ、その形状はほぼ直方体である。このサイズは品種によって様々に異なる。
- ・ 製品は天地のあるものを除いて横にしてよい。

これらの条件の下に、コンピュータを用いてダンボール内におけるそれぞれの製品の位置を表す詰合せパターンを計算することを具体的な問題とする。

4. 製品詰合せ問題への GP の適用

製品詰合せ問題に GP を適用する際、遺伝子型の設定（コーディング）、評価関数の定義及び遺伝的操作の方法を定める必要がある。提案手法ではこれらを以下のように定義する。

4.1 コーディング

まず、ダンボールに詰合される製品の品種を 1 から順に付番し、それを品種番号とする。品種番号を持つ製品はひとつまたは複数のダンボールに詰合せされることになるが、どの品種の製品をどのダンボールのどこに配置するかを GP が生成する詰合せパターンとする。つまり、ひとつの詰合せパターンがひとつの個体に対応し、遺伝子型は品種番号が末端の葉の部分（ターミナル）に位置する木構造で表される。

遺伝子型の各葉、つまりターミナルにはそれぞれ箱詰めされる製品ひとつが対応付けられる。提案手法では、製品の形状は直方体とし、天地があるものを除いては寝かせて配置することができる。従って、ターミナルは製品の品種番号の他に回転状態も管理する。ひとつの製品において 3 種類の底面があり、それぞれ 2 種類の配置が可能であることから、計 6 種類の回転状態を考慮する。天地がある製品は底面が決まっているため 2 種類の回転しか考慮しない。木の形は詰合せパターンにより様々であるが、全ての製品はどこかに配置される必要があるため、ターミナルつまり葉の数は一定でなければならない。

さて、遺伝子型の木は上述の各ターミナルが統合され、ボトムアップ的に構成されていると見ることができる。ここで、枝を統合する節点の部分（ファンクション）にはそれぞれの枝の位置関係が置かれる。提案手法では、位置関係に「箱分け（箱を分ける）」、「上下（上に置く）」、「左右（左に置く）」、「前後（奥に置く）」の 4 つの概念を用意する。（図-3 参照。ただし、図中の D(Divide) は「箱分け」を、U(Up) は「上下」、B(Back) は「前後」、L(Left) は「左右」の関係を表すファンクションである。また、数字はターミナルの製品番号を表している。）「箱分け」のファンクションは箱詰めに必要な箱数 N に応じた N 分岐の節点であるが、その他のファンクションは左枝と右枝との位置関係を定める 2 分岐の節点である。節点は葉、あるいは部分木を子としてもつ

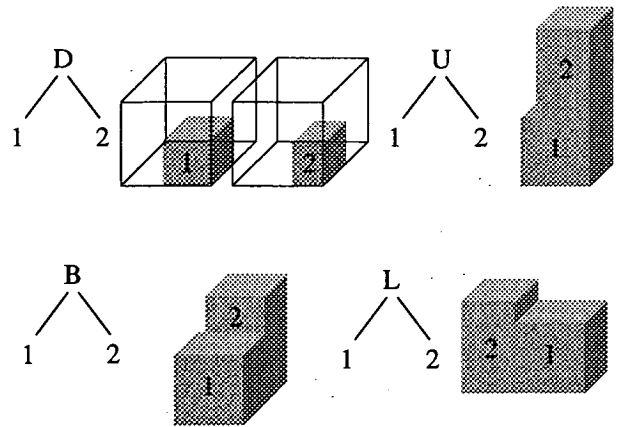


図-3 4つの配置関係

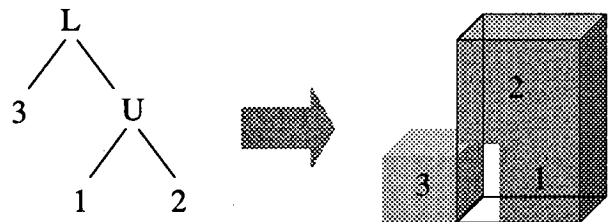


図-4 製品群に対する配置

ため、一製品、あるいは製品群の間の位置関係を表すことになる。ところで、各枝が葉、つまり一製品を表現する場合には、ファンクションはその製品を基準オブジェクトとした位置関係であると判断することができるが、部分木、つまり製品群を扱う場合には、位置関係を一意に定めるために、製品群中から位置関係の基準となるようなオブジェクトを決定する必要がある。提案手法では、処理を簡素化するため、各枝が表現する製品配置を包括する最小の直方体を基準オブジェクトとして考える（図-4 参照）。製品が直方体であることを考えると、各枝は必ず直方体の基準オブジェクトを持つことになる。

基準オブジェクトの定義に伴い、位置関係のファンクションを拡張する。「箱分け」を除く各ファンクションを左右の枝がもつ基準オブジェクトが接する面の沿わせる角により細分化する。「上下」の配置を表すファンクションは、左枝が持つ基準オブジェクトの上面四隅のいずれに右枝の基準オブジェクトを沿わせるかにより 4 種類用意する。「左右」の配置を表すファンクションは手前、奥の下側の角のどれに沿わせるかにより 2 種類、「前後」の配置を表すファンクションは左右の下側の角のどちらに沿わせるかにより 2 種類用意する。「左右」と「前後」に関しては、重力を考慮してそれぞれ下側の角に対する 2 種類のファンクションしか用意しない。

配置に関するファンクションは以上 9 種を用いることにする。従って、ダンボールや他の製品に寄りかかるような斜めの配置は考慮しない。

上述のような設定のもとでは、遺伝子型から表現型に

$$P_{arrange} = \alpha P_{float} + \beta P_{over} + \gamma P_{blank} + \delta P_{devide} \quad \dots (1)$$

$$P_{over} = \beta_1 \sum_i \{(Box_x - x_i) + (Box_y - y_i) + (Box_z - z_i)\} + \beta_2 \sum V_{overed} \quad \dots (2)$$

$$P_{blank} = \sum_i (x_i \cdot y_i \cdot z_i - \sum_j V_{ij}) \quad \dots (3)$$

$$P_{devide} = \sum_i \sum_j \frac{\sum |elem_{ij}|}{|elem_{ij}|} \quad \dots (4)$$

$$(ただし |elem_{ij}| = 0 \text{ なら } \frac{\sum |elem_{ij}|}{|elem_{ij}|} = 0)$$

(i : 箱番号, j : 製品番号)

$P_{arrange}$: 製品配置に関するペナルティ

P_{float} : 不安定な製品に関するペナルティ

P_{over} : はみ出した配置に対するペナルティ

P_{blank} : 隙間に関するペナルティ

P_{devide} : 分散した製品に関するペナルティ

Box_x, Box_y, Box_z : 箱の幅, 奥行, 高さ

x_i, y_i, z_i : 箱 i への配置(直方体)の幅, 奥行, 高さ

$\sum V_{overed}$: はみ出した製品の総体積

$\sum V_{ij}$: 箱 i に詰め込まれる製品の総体積

$|elem_{ij}|$: 箱 i に詰め込まれる製品番号 j の製品数

$\alpha, \beta, \gamma, \delta, \beta_1, \beta_2$: 各ペナルティの寄与度 (重み)

図-5 製品配置に関するペナルティ

翻訳した際に、ある製品が宙に浮いてしまうような遺伝子が出現する可能性がある。このような遺伝子には、次節で述べる評価関数により高いペナルティを与える。

4.2 評価関数

適応度とは GP における各個体がどの程度問題に適合しているかを表す指標である。詰合わせ問題の数学モデルを構成し、その指標を適応度とする。ここでは、ダンボールへの製品の納まり具合、つまり製品の配置に関するペナルティを制約として設け、そして、箱詰めに必要なダンボール数を効率を表すペナルティとして与える。この問題ではペナルティを適応度と考え、これを最小化させる。言葉からは適応度の低いものをよいとするのは違和感があるが、パッキングとしてはこの方が考え易いのでこのまま適応度をペナルティに置き換えて議論を進

$$P = W_{arrange} \cdot P_{arrange} + W_{box} \cdot P_{box} \quad \dots (5)$$

$P_{arrange}$: 製品配置に関するペナルティ

$W_{arrange}$: 製品配置に関するペナルティの寄与度

P_{box} : ダンボール数のペナルティ

W_{box} : ダンボール数のペナルティの寄与度

図-6 評価関数

める。

(1) 製品配置に関するペナルティ

製品配置に関するペナルティは以下の4つの要素に対して適用することとする (式-(1))。

まず、何もない空間の上に製品があるなど物理的に配置できない、あるいは不安定な配置に対しては、その不安定な製品数に応じてペナルティを与える。次に、生成した詰合せパターンのそれぞれ最大の幅、奥行き、高さがダンボールの内法に納まらなければ、オーバーした寸法に応じて、そしてはみ出した製品の体積の総和に応じてペナルティを与える ((式-(2))。また、製品を包括する直方体に隙間がある場合は、その隙間の体積に応じてペナルティを与える (式-(3))。さらに、同じ製品をまとめることにより効率の良い詰合わせが実現されと考えられるため、同じ製品が他の製品と混じって複数のダンボールに分散する場合はペナルティを与える (式-(4))。

(2) ダンボール数のペナルティ

箱詰めに必要なダンボール数に対してペナルティを与える。

こうして得られたペナルティの重み付き加重和を各個体のペナルティとする (式-(5))。ペナルティの重みについては、ペナルティの優先度と GP の解の収束状況を合わせて調整する。解の収束状況はある世代の全個体の平均と最もよい個体の両方について世代に対する推移を把握する (図-9 参照)。

4.3 遺伝的操作

GP 操作として選択淘汰と交叉、兄弟交換、突然変異の方法を以下の様に定める。

(1) 選択淘汰

選択淘汰を行う際に重要なのが、どの個体同士を交配させるかである。選択の主な方法として、a) 適応度比例戦略、b) 期待値戦略、c) ランク戦略、d) エリート保存戦略、e) トーナメント選択戦略などがある。今回は最も一般的

であるエリート保存戦略と適応度比例戦略を採用する。

エリート保存戦略は、ある世代の個体群の中から優秀な個体を次世代にそのままコピーし、解を早く収束させることを狙う戦略である。また、適応度比例戦略はルーレット戦略とも呼ばれ、適応度の分布に比例した確率で子孫を残す方法である。

この二つの混合戦略では、あらかじめ設定された割合だけ、優秀な個体をエリート保存戦略によってコピーし、残りをルーレット戦略で選ばれた親個体に交叉、突然変異の操作を行い、個体を進化させていく。

(2) 交叉

交叉は二つの親となる個体の遺伝子型を組み替えて子の遺伝子型を作る操作である。GP では、遺伝子型は木の構造をしているのでそれぞれの木の枝の部分を選択的に選んで交換する操作を行う。しかし、今回は扱う問題の制約上、ターミナル、つまり箱詰めされる製品の種類及び数量を遺伝的操作の前後で保存する必要がある。従って提案手法では、ターミナルが保存される範囲での交換のみを許すこととする（図-7(1) 参照）。

(3) 兄弟交換

兄弟交換は親となるひとつの個体の遺伝子型の中で枝の順序を入れ替えて子の遺伝子型を作る操作である。兄弟交換はファンクションの引数の順序を変える遺伝的操作である。

提案手法では兄弟交換により、詰め合わせ製品の上下関係、前後関係、及び左右の関係が逆転する。箱分けのファンクションに関しては兄弟交換は無意味であることから、根の部分に関してはこの操作は適用しない。

(4) 突然変異

人工的に遺伝子の突然変異を発生させる。1～5%前後の低い確率で起こすものとし、遺伝子の変更位置を乱数により定め、その内容をこれも乱数により変更する。

突然変異には一般的に、ターミナル、ファンクションに関してそれぞれラベルの付け替えのみを行うものの他に、ターミナルからファンクションへの変化（新しい部分木の生成）、ファンクションからターミナルへの変化（部分木の削除）が用いられている。

しかし、突然変異による無造作な部分木の生成、削除はターミナルの喪失、増加を招き、各製品の数量が保存されないため、提案手法ではこれらの操作の代替として部分木の付け替え（図-7(2)参照）、あるいは部分木と葉との付け替え（図-7(3)参照）を行うことにする。また、ターミナルのラベルの付け替えに関しても、各製品

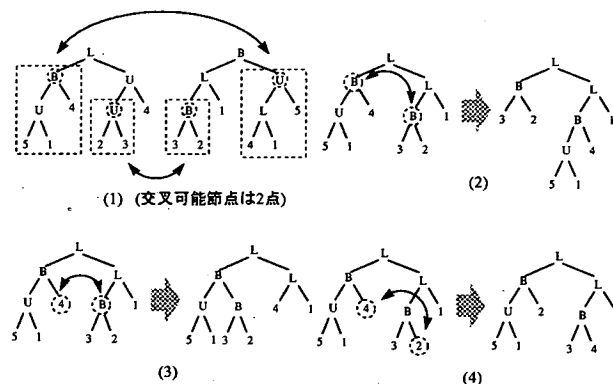


図-7 拡張された遺伝的操作

の数量を保存するために、ターミナル、つまり葉を付け替えることにする（図-7(4) 参照）。なお、ファンクションのラベルの付け替えは従来通り行う。ターミナルの回転状態も突然変異のプロセス内で確率的に変化させる。

5. 実験

前章までで提案した手法の有効性を確認するために、提案手法を実際に計算機上に構築し、詰合わせパターンの獲得を行った。本章では構築システムによる実験結果を示すとともに、本手法の有効性と問題点について考察する。

5.1 実験の概要

提案手法を Sun Sparc 上に Java 言語を用いて構築した。実験としては、まず、本手法の基本的な性能を評価するために、一つの箱への詰合わせパターンの獲得を行った。

次に、複数の箱に製品を振り分けて箱詰めする際のシステムの挙動を検証するために、製品数を増やして詰合わせパターンの獲得を行った。

実験データには、実際の製品データと箱のサイズを用いた。なお、本手法の本質的な性能を評価するため、いずれの実験にも天地のある製品は含めていない。

5.2 実験結果

一箱に箱詰めする例として、7種類 21 個の製品（充填率 58%）の箱詰めを固体数 100、世代数 300 で実行した。得られた解を図-8に、解の収束状況を図-9に示す。

複数の箱に製品を振り分けて箱詰めする例としては、11種類 67 個の製品（充填率 62%）の箱詰めを固体数 100、世代数 500 で実行した。得られた解を図-10に示す。

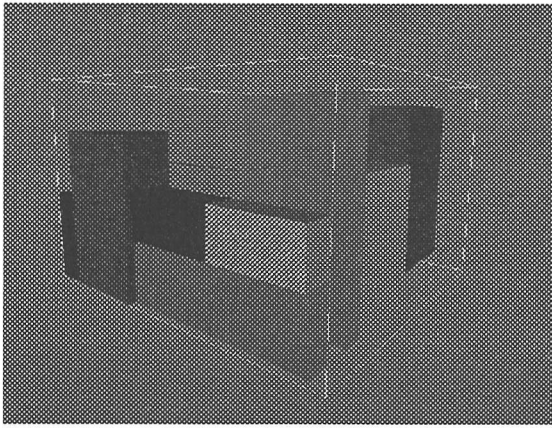


図-8 一つの箱への箱詰め

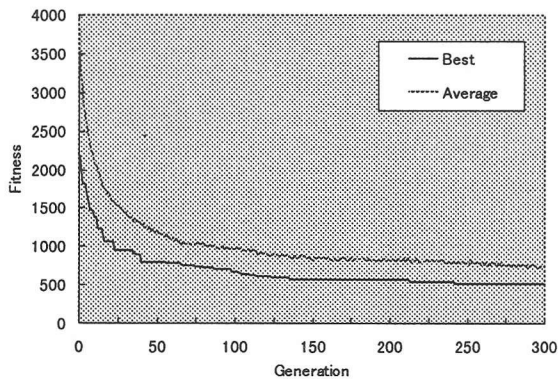


図-9 解の収束状況

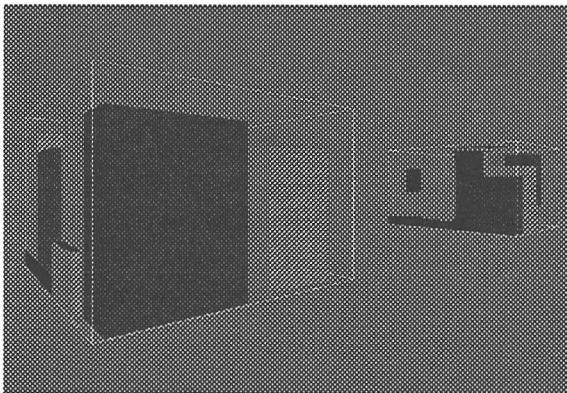


図-10 複数の箱への箱詰め

5.3 考察

ひとつの箱への箱詰めでは、製品に若干のばらつきがあるものの、箱をはみ出さない配置を獲得することができた。本稿では示していないが、2,3種の製品の詰め合わせにおいては、充填率87%の配置も可能であった。一方、複数の箱への箱詰めでは、同品種の製品の振り分けにばらつきがみられ、従って配置も若干不安定なものであった。

いずれの実験にも当てはまるが、同品種の製品が離れ

た位置に配置されており、これが一つにまとまりブロック化されればより適した解が獲得されることが考えられる。ブロック化に関する対策としては、Kozaの自動的関数化手法[1]が有効であると考えられる。これは適応的にサブルーチンを生成するGPの拡張手法であり、効率的な詰合わせパターンの保存、木構造の冗長性に対して有効な手法である。

6. おわりに

本稿では、物流施設における製品の詰合わせという現実の問題に対して、詰合わせパターン生成にGPを適用し、効率よく最密充填に近い解を求める方法を提案した。また、提案手法を実際に計算機上に実装し、その性能に関して検討した。

GPは遺伝子型が一次元配列であるGAと比較して、木構造を扱うため、表現力が豊富であり、3次元の幾何的構造を持つ詰め合わせ問題に対しても十分対応することができる。しかしその反面で、計算量が問題点となる。構築システムも現段階では、解の探索に多大な計算時間を必要としている。

今後の課題として、高速化、製品配置の簡潔化が挙げられる。高速化に関しては、並列計算と自動的関数化手法の導入を検討している。後者は製品配置の簡素化を促す製品のブロック化にも有効な手法であり、より複雑で現実的な詰合わせ問題にも適用可能であると考えられる。

参考文献

- 1) John R. Koza : Genetic Programming, MIT Press, 1992.
- 2) 伊庭斉志：遺伝的プログラミング，東京電気大学出版局，1996.
- 3) 今野浩，鈴木久敏編：整数計画法と組合せ最適化，日科技連，1982.
- 4) 本堂直浩，伊庭斉志，嘉数脩昇：拡張型GPによるロボット行動の自動生成へのアプローチ，日本機械学会論文集，63巻609号，No. 96-0425，1997.
- 5) 北野宏明 編：遺伝的アルゴリズム，産業図書，1993.
- 6) 北野宏明 編：遺伝的アルゴリズム3，産業図書，1997.