広島大学総合科学部紀要Ⅳ理系編，第26巻，117－129ページ，2000年12月

# 部屋の家具配置における遺伝的アルゴリズムの試み

黒田　哲・山縣　敬一

広島大学総合科学部

# A Trial of Genetic Algorithm to Furniture Arrangement in a Room

Tetsu KURODA and Keiichi YAMAGATA

*Faculty of Integrated Arts and Sciences, Hiroshima University*
*Higashi-Hiroshima* 739-8521, *Japan*

**Abstract** : Many layout design problems can be considered as constraint satisfaction problems. However it is not easy to obtain the optimal solution in many cases uniquely. In this paper, we tried to use genetic algorithm for furniture arrangement in a room according to user's planning intention. The orientation and position of furniture are coded into bit strings of genes. After processing of crossover breeding and random mutation, higher fitness solutions are selected. This process is executed iteratively. The key point of this research is not to find unique optimal solution, but to find some of suitably fitted solutions according to user specified constraints. For this purpose, we discussed man machine interface concretely in user's planning process. It was clarified that in the selection process of genes based on constraint fitness, elite-keeping selection is suitable to improve fitness gradually and selection by roulette has possibility to derive a different type layout.

**Keywords** : genetic algorithm, furniture arrangement, constraint satisfaction

## 1．はじめに

部屋の家具配置を人が考える場合を想定してみる。人は頭脳の中のイメージだけでもプランを作ることができる。この場合、窓の位置、扉の位置を考慮した上で、机を窓際に置くとかソファを部屋の中央に置くなどの相対的位置関係の望ましいものを考えながら、お互いの重なりがないように、プランを作る。あるいは、もう少し細部の検討をしたい場合は一定の縮尺に基づく部屋の図面と家具の型紙を用意して並べてみるやり方をする。このとき、隙間に家具を押し込むような場合は別として、ソファを部屋の中央に置きたいと言っても、厳密な意味で中央に置く必要はなく、むしろ回りの状況から、多少中央からずれても使い勝手がよければ、人はそれで良いと判断する。

一方、同じように図形を配置する問題でも、回路基板に部品を配置するような場合は、ある部品についてはその間の距離を3mm以上空けないといけない、あるいは特定の部品と部品はすぐ隣に置かなければならないなど、精密な条件が設定される。その条件のもとで、全体の基板の大きさを最

2000年10月1日受付；2000年11月1日受理

小にしたい、といった問題解決が必要になる。ここで述べたのは例示に過ぎないが、上のような家具配置では、与えられる条件が定性的であり、回路基板の設計では定量的である。このような問題をコンピュータで処理しようとするとき、コンピュータのプログラムでは、定性的な表現が難しく、一般に図形配置問題を解くときには、数理計画法の手法により空間に座標を設定し、それぞれの図形を少しずつ位置をずらしながら、あらゆる組み合わせを調べることになる。この手法は回路基板の設計のように厳密な最適解を見つけるのには必要であるが、計算時間がかかるため、家具配置問題にはもう少し別のアプローチ必要になる。これが本研究の動機である。

本研究で適用を試みた遺伝的アルゴリズムは、ソフトコンピューティングの一手法であって、高度の精度性を要求せず、生物の進化に習った適応と学習の方法論として注目されている。数理計画法との関係で遺伝的アルゴリズムを捕らえる研究も行われているが[1,2]、本研究においては、数理計画法による厳密解を探索するのではなく、上に述べた家具配置問題の特色に注目して、もう少し緩やかなユーザーにとって好ましい解、あるいはもしも解に多様性があるならばそれを無理に一つの解に絞り込まず、いろいろな解があること自体をユーザーに提示するようなシステムを試作してみたものである。文献[3,4]には、回路素子の配置問題や図形配置問題を遺伝的アルゴリズムで解く試みが議論されているが、本研究では解の捕らえ方を別の視点から見ている。以下に、家具配置問題の特徴を吟味し、それを遺伝子の中に組み込む表現法を述べ、試作したシステムについて特にユーザーインタフェースを説明し、家具配置問題と遺伝的アルゴリズムの持つ解の特性について、具体的な例を使って議論する。

## 2．家具配置問題と遺伝的アルゴリズム

ある部屋の中に家具をいかにうまく配置するかということを考えた時に、まず重要になるのがユーザーの要求による条件設定である。部屋のスペースとその中に置く家具の種類、そして、その置き方にはユーザーの好みがあるはずで、机を壁に向かって置きたいとか、機能的な面を重視して本棚の近くに机を置きたいなど、いろいろな条件があってそれらは各個人によって要求が異なる。また、複数の条件を指定する場合にそれら全てが満たされない場合は、どの条件を優先させるかもその人の要求による。従って、家具配置のプランニングに対してコンピュータを使って支援させようとするとき、まずユーザーによって異なる多様な条件を指定できなければならない。しかも、すでに述べたようにユーザーの要求は定性的な指定が多く、ユーザーの要求に対する適応度から解を見出す必要がある。

家具配置問題もそうであるが、一般に図形を２次元領域にある制約の下に適切に配置する図形配置問題は、制約充足問題として定式化される。このような問題では、数理計画法を使って家具の位置を少しずつ動かしながら制約充足解を探索すると多くの計算時間を必要とする。そこで、従来このような場合に取られてきた手段は、ヒューリスティックスを利用して近似解を求めるものである。ヒューリスティックスというのは、適用する問題の機構や性質を熟知している人や、その人の知識を用いて、その問題固有の性質を十分に知って、人によって作成されるアルゴリズムを用いる方法である。この手法の基本は経験に基づいているので、以下のような欠点を持っている。

(1) ユーザーの条件設定が多様であることを考えると、有効な手法の発見が難しい。
(2) 制約がほんの少し変化しただけでも以前の手法が有効でなくなる場合が多く、柔軟性に欠ける。

このため、制約を直接解かない遺伝的アルゴリズムのような統計的手法を使用する方法が近年注目を集めている。

遺伝的アルゴリズムとは、生物の進化過程における遺伝的な法則をモデル化したものである。自然界における生物の進化は、ある世代を形成している個体群の中で環境への適応度の高い個体が高い確率で生き残り、子孫を残すことができる。さらに、交叉や突然変異などの遺伝における現象を経て次の世代の個体群を形成していくのである。これを繰り返すことにより、より環境に適応した個体が形成されていく。

遺伝的アルゴリズムを家具配置問題へ適用する場合、以下のような特徴が考えられる。

(1) 配置アルゴリズムが不要
評価関数を与えるだけで自動的に配置が得られるため、複雑な制約に対しても配置のためにアルゴリズムを考案する必要がない。

(2) 制約の種類、強さを指定できる
評価関数を変化させるだけで使用する制約を変えることができる、また、各制約に対する評価関数の重みの値を変化させることによって得られる配置を微調整することができる。よって、人により異なる配置を得ることができる。

(3) 柔軟である
もし、矛盾した制約を与えた場合、遺伝的アルゴリズムでは何らかのそれなりに意味のある結果を得ることができる。これに対して線形制約の解決システムにおいては矛盾する制約があると単に解けないという結果が得られるだけである。

家具配置問題においては、部屋に置く家具のそれぞれについて、位置と向きを２進数で符号化し、一つの家具配置を一つの遺伝子個体で表現する。今回の応用例では、一つの家具について、上下左右の向きを表すのに２ビット、家具の部屋の位置を表すのにX座標に７ビット、Y座標に７ビットを用いるので一つの家具の配置が16ビットで表される。これを家具の数だけ並べて一つの遺伝子の個体を構成する。後に述べる応用例では７つの家具を取り上げているので、112ビットの０と１のパターンが一つの個体の染色体になる。

次に遺伝子に関する操作は、これを単なるビット列とみなしてこれに以下のような処理を施す。このとき、遺伝子操作については、この問題のコード化の意味は考慮されず、遺伝子操作を行って適応度の高い個体を作り出すことを試みる。本報告での例に則して主要な遺伝子操作を要約すると以下のようになる。

**手順１（初期化）** 乱数を使って40個体の遺伝子を生成する。これが初期世代となる。

**手順２（再生）** 各固体の適応度を評価し、適応度の高いものを残し低いものを淘汰する。このとき、残す個体の選択の仕方として、エリート保存選択とルーレット選択を採用する。前者は、適応度の最も高い個体を無条件で残す方法であり、後者はある程度以上の適応度を持つ個体の中からランダムに選んで残す方法である。

**手順３（交叉）** 個体群の中でランダムにペアを選び、染色体の一部を入れ替える。今回の研究では２点交叉法を採用した。以下のように二つの個体間でランダムに２箇所の切れ目を入れて、染色体の一部を入れ替える。

親1）０１０｜０１１｜０１　　子1）０１０００１０１

⇒

親2）０１１　００１　１０　　子2）０１１０１１１０

**手順４（突然変異）**　一定の割合で、個体の染色体を変化させる。今回の研究では部分的に染色体の一部を反転させる反転方式を採用した。

１０１０１１１１０⇒１０１００１１００
　　↑　　↑
　　ここが突然変異を起こすとする

**手順５（判定）**　今回の研究では、ユーザーとシステムとのインタラクションにより、家具配置をユーザーに図示し、ユーザーによって最終判定を行う。

以上が家具配置問題における遺伝的アルゴリズムの概要であるが、この遺伝子操作においては、染色体のビット列の意味は考えていない。従って、最適化アルゴリズムによる解法とは根本的に異なっている。単に適応度を高めるだけに操作を進めると、行き詰まりを生じるような問題でも、交叉や突然変異によって別の傾向の解が見出される可能性を持っている。これが、この手法の特徴である。

## ３．試作システム

これまで述べて来たように、本研究のシステムでは、従来の数理計画法と違ってあらゆる配置の組み合わせを調べて一つの最適解を探すことが目的ではなく、ユーザーの希望する条件の適応度により遺伝子の操作によって構成される配置をユーザーに提示し、最後はユーザー自身が判断して解を得るシステムになっている。従って、好ましい配置を探して行く過程で、ユーザーとのインタラクションが容易に行われることを重視したシステムになっている。以下の説明もこの点に視点をおいて説明する。

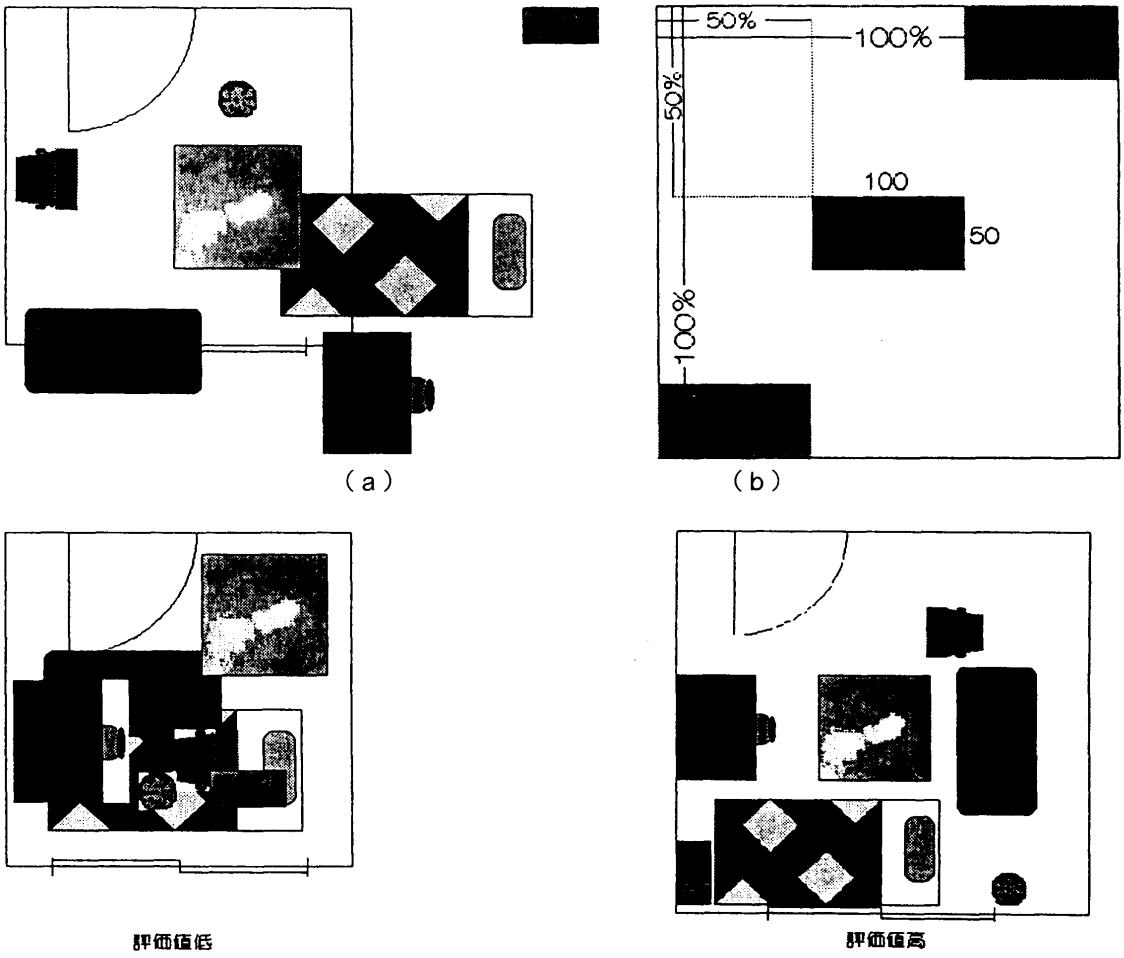
まず、家具配置をシステムがどのように表現するかを、図１に示す。ここで取り上げている例では、最大７個の家具（ベッド、ソファー、テーブル、テレビ、机、本棚、観葉植物）の配置を考える。ここで、家具の大きさ（幅と奥行き）は予め定義されているものとし、家具の１点に基準点（コーナーの点）を定めて部屋の中の位置を定める。家具の向きも基準の姿勢を定めた上で、配置の向きを２ビット符号により次のように表す。

00ー上向き、01ー下向き、10ー右向き、11ー左向き

ところで、部屋に単純に座標系をとって配置するだけでは、図１（ａ）のようなことが起こる。これは、部屋の大きさへの対応がとれておらず、部屋が小さいために外に配置されていたり、家具の基準点は部屋の中にあっても家具全体が部屋に収まらない場合で、応用上役に立たない。実は、遺伝的アルゴリズムでこのような制約条件を扱うことは避けることが望ましい。すでに述べたように、遺伝子操作では元の意味を考えることなく処理が進むので、遺伝子の個体それぞれについて制約条件を検証し、それを淘汰に生かす方法は好ましくない。

本研究のシステムでは、これを避けるため、図１（ｂ）に示すように、部屋の中に特定の家具を置ける範囲を定めた上で、相対的な位置を100%表示で表して、遺伝子に表現する。例をあげると、横300、縦300の部屋に、幅100、奥行き50の物体をそのままの向きで置くとする。すると、物体の左上のX座標が部屋の中に存在し得る範囲は０から200の間となる。同様に、Y座標に関しては０から250の間となる。配置の許容範囲を100%としてそのうちの何パーセントの場所に位置しているかということにするのである。遺伝子としては０から100の間の数字を表せればよい。その数字が50、50

だとすると、実際に配置した場合図１（ｂ）のようになる。




（ａ）　（ｂ）

（ｃ）　（ｄ）

**図１　家具配置の表現**

このようにすると遺伝子におけるデータは、部屋の大きさや家具の大きさとは独立になり、この表示で操作を行う限り、家具が部屋の外にはみ出ることもない。このような問題は、以後の遺伝子操作の処理効率に関わるので、遺伝子の表現を定めるときに大変重要になる。結果を図示するときに部屋の大きさと家具の大きさを使って計算すればよく、これは遺伝子操作から切り離された処理になる。２節で述べた位置情報の符号化はこの方法を用いたものである。

次に、図１（ｃ）は小さな部屋に家具を詰め込みすぎて配置ができない例である。この家具の重なりについては、遺伝子個体の適応度評価で低い評価値を与える。具体的には、適応度評価値に家具の重なりの面積を負の効果として計算に入れている。しかし、ここでも家具の重なりを持つ個体があってもそれをすぐに淘汰してしまうことはしない。評価値を下げた状態でもその個体は残る可能性を持っている。このことは、たとえ広い部屋でもたまたま重なりがある場合があるし、それを少し変えることで望ましい解が得られるかも知れないからである。図１（ｄ）は、家具の重なりが

なく、高い評価値を持つ例である。そして、ここから先はユーザーの好みをどこまで反映させるかが重要になる。ユーザーによって机を窓際に持ってきたい人もいれば、ベッドを窓際に持ってきたい人もいるはずである。

以上の考察から、図２の設定画面が用意されている。ここでは、部屋の大きさと配置する家具を選ぶと共に、使用する制約の欄で考えられるユーザーの嗜好を指定できるようになっている。この家具の位置関係の制約はユーザーがどれを重視するかの重みをつけて（最大10）指定することにより、遺伝子個体を評価する際に反映される。このことにより、遺伝子操作を進めるうちに、ユーザーの嗜好に沿う遺伝子が多く残り、ユーザーの嗜好に沿わない遺伝子は淘汰されることになる。なお、遺伝子操作についても突然変異率（%）と再生時の選択方法を指定できる。これについては後述するが、いろいろな家具配置を試してみようとするときに効果を発揮する。

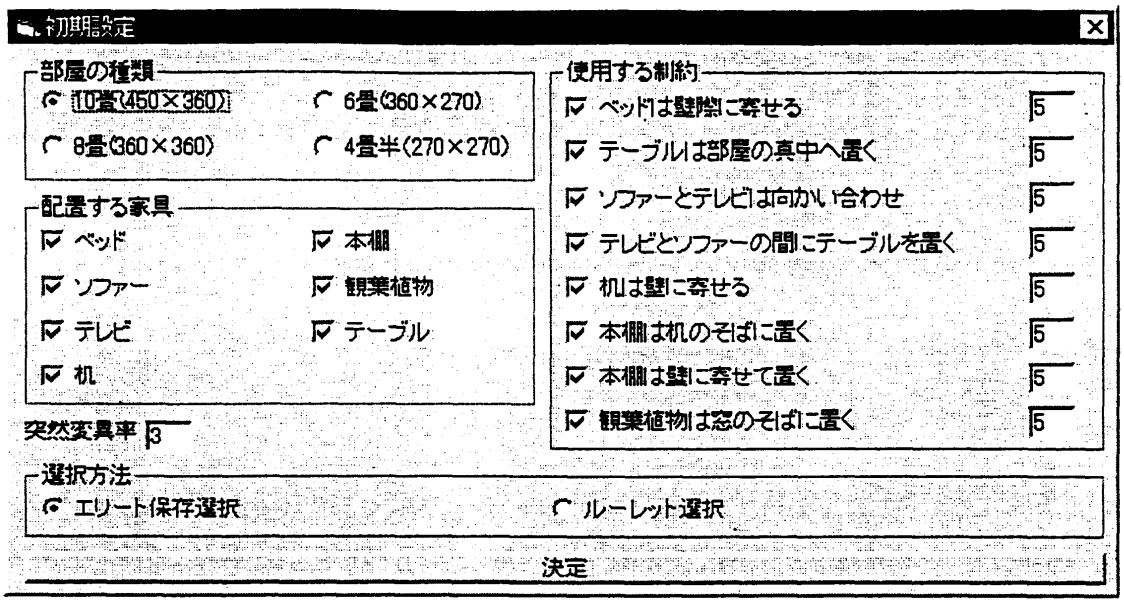



図２　ユーザーによる条件設定画面

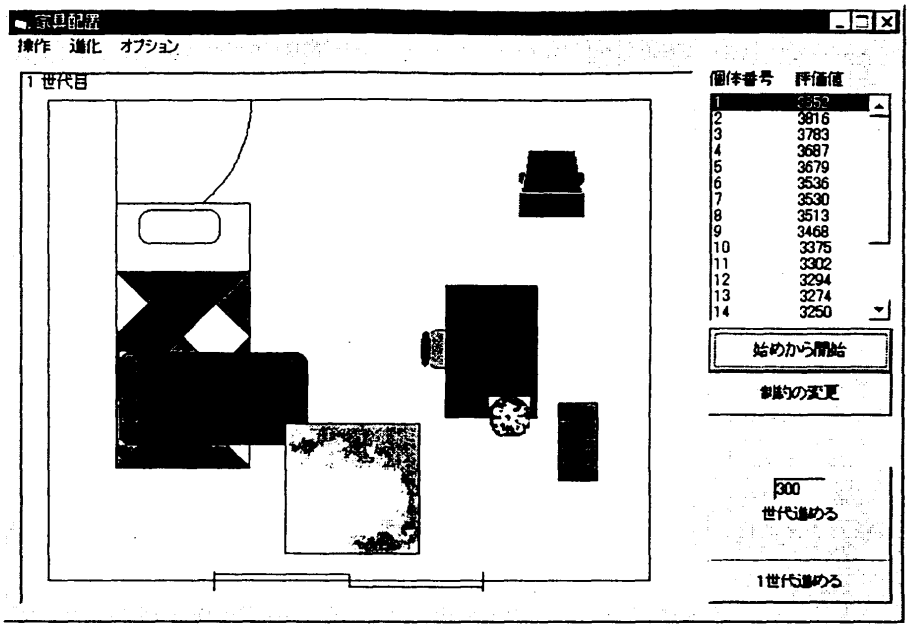
## ４．実行例と有用性の評価

実行例として、10畳の部屋にベッド、ソファー、テレビ、机、本棚、テーブル、観葉植物を配置することを考える。ユーザーの指定する制約は

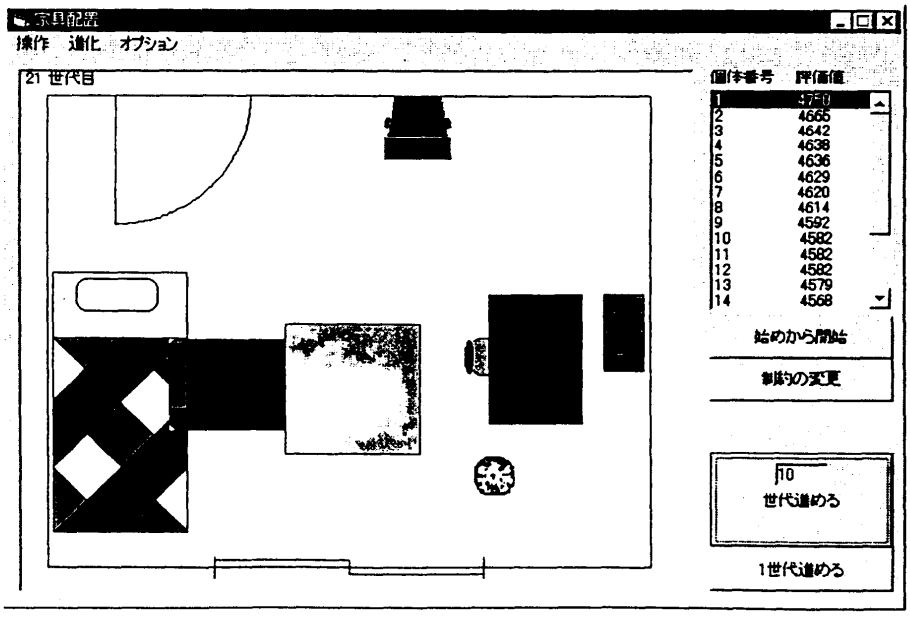
・ベッドは壁際に寄せる
・テーブルは部屋の真中に置く
・ソファーとテレビは向かい合わせ
・机は壁に寄せる
・本棚は壁に寄せて置く

であるとする（それぞれ重みは5/10）。図２の初期選定画面でこれらの条件を入れた後、決定のボタンを押すと、図３に示すようなアプリケーションのメインとなるウィンドウに切り替わる。ウィンドウの右上に表示されているのが個体のリストである。その左がリストで選択された個体の遺伝子

をデコードして配置を図示したものである。リストをクリックするとその個体の配置が表示されるようになっている。図３（ａ）が初期世代で40個の遺伝子個体が作られ、その中で適応度評価値の最大のものが図示されている。




（ａ）




（ｂ）

図３　エリート保存選択による進化

個体の評価法は、どの制約を使用するかによって変わってくる。このプログラムを実行した場合、始めにどの家具、どの制約を使用するか、それぞれの制約の重みなどを決定する。そこで選択された制約それぞれに、個体の位置関係から制約がどの程度満たされているかを検証するプログラム・モジュール（プログラムにおける関数）が存在しており、使用すると決定されている制約に関する得点に重みをかけたものを合計して評価値とする。ここで、使用すると決定されている制約があってもその制約に関する家具が選択されていなければ、その制約に関する評価値は0とする。制約に関する個体の評価式は、数式で表現できる形ではなく、プログラムに組み込まれており、そこで実行される主要な機能を挙げると、次のようになる。

・家具と家具の重なりの程度をその重なりの面積から計算（これについては後述）
・ドアの開閉部分と家具の重なりの度合いを評価
・壁際、壁に寄せるなどは、家具と壁の間の距離を部屋の大きさの相対的な位置として評価
（この場合、家具の向きも考慮）
・部屋の中央などは、家具の中心と部屋の中心との距離を評価
・間にあるかどうかは、X座標Y座標それぞれについて特定の家具が間にあるかを評価

このような検証機能を各家具についてその向きを考慮しながら得点を計算し、全体の評価値の基礎とする。なお、遺伝的アルゴリズムの性質から、例えば少しでも重なりがあれば駄目という評価の仕方はせず、制約条件を満たす度合いに応じて、得点を出して行く。

家具同士を重ならないようにするという部分の評価を例にとってもう少し詳しく述べておく。基本的には、家具と家具の重なっている面積を調べ、その量によって評価値を決定する。この場合、重なっていないほうがよいので、重なっている面積が小さいほど評価値が高くなるようにしなければならない。そのために、あらかじめ二つの家具の組み合わせについて最大の重なり面積を調べておき、その面積から実際に家具同士の重なっている部分の面積を引いた値を計算する。そして、残った面積に他の制約とのバランスを調整するための重みをかけたものを重なりに関する評価値として与える。“ドアの開閉に邪魔になる部分には家具を置かない”、という制約に関しても上と同様に、ドアの開閉の邪魔になる部分と家具の重なりに間する同様の問題と考えることができるので、同じように評価値を与えることができる。

さて、図3（a）が初期世代であり、この後、エリート保存選択を使って、再生淘汰を20世代繰り返すと、図3（b）が得られる。40の個体のうち、適応度の高いもの20個を残し、これを親として20個体の子を遺伝子操作で作り出す。このとき、エリート保存選択では、適応度評価値の高いものを必然的に残すので、評価値の最大値は世代が進むにつれて単調に増大して行く。図3（b）をみると、初期世代に比べてユーザーの指定した制約が次第に満たされ、ベッドや机が壁際に寄せられている様子がわかる。最後は、ユーザーの判断によっているので、数十世代ごとに配置を確認して行く。図4に得られた結果を示す。ほぼ、ユーザーの指定した制約が満たされ、家具の重なりもなくなっている。

本報告の冒頭でも述べたように、人手で家具の配置を考える場合、家具の型を切り抜いて紙の上に並べるやり方をする。この状況を考えると、図4の結果がシステムによって示されることは十分に実用性があると思われる。ただし、ユーザーの指定する制約の適応度によって解をさがしているので、他にもユーザーにとって好ましい解があるかも知れない。この視点から調べるために、同じ条件のもとで、再生淘汰のやり方をルーレット選択にした場合の進化の状況を図5に示す。

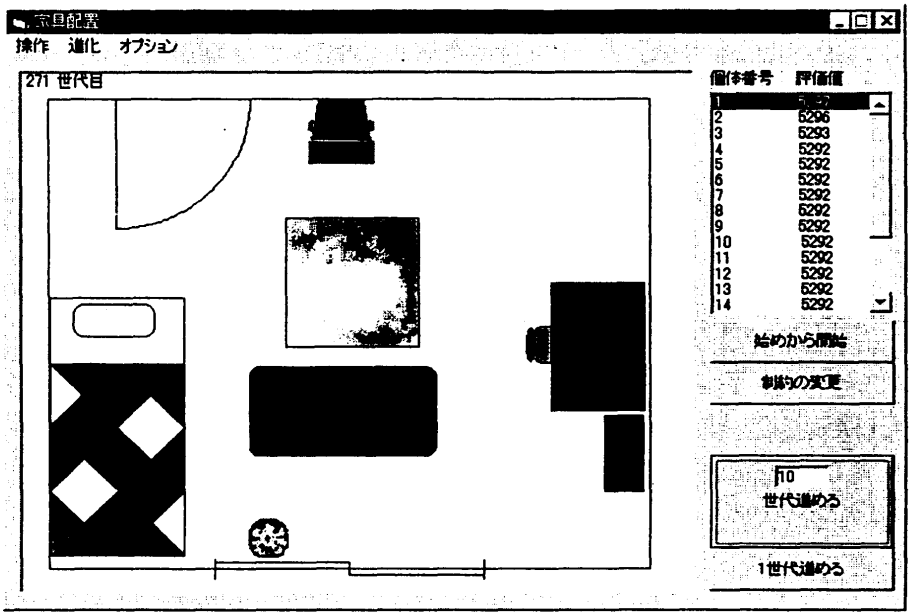
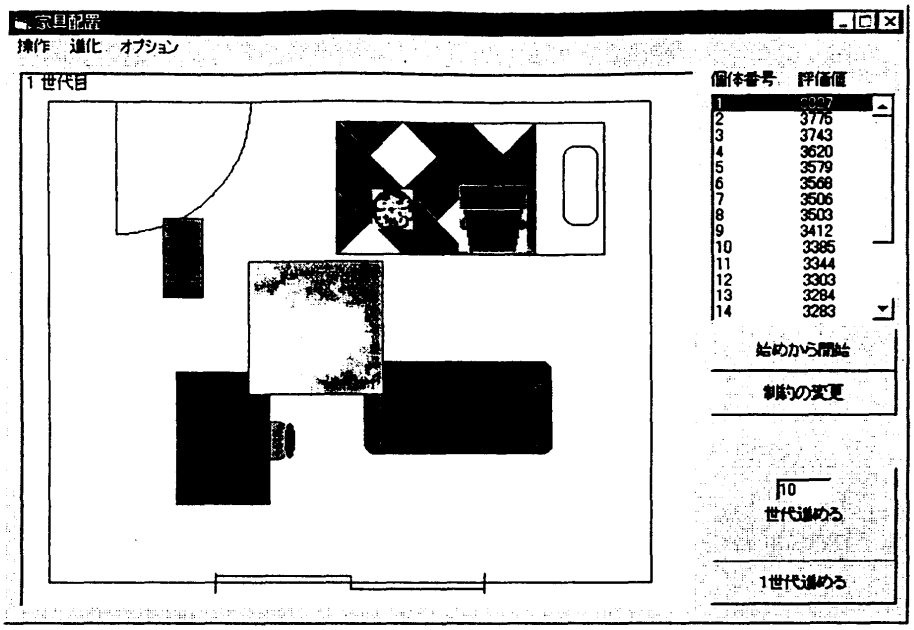
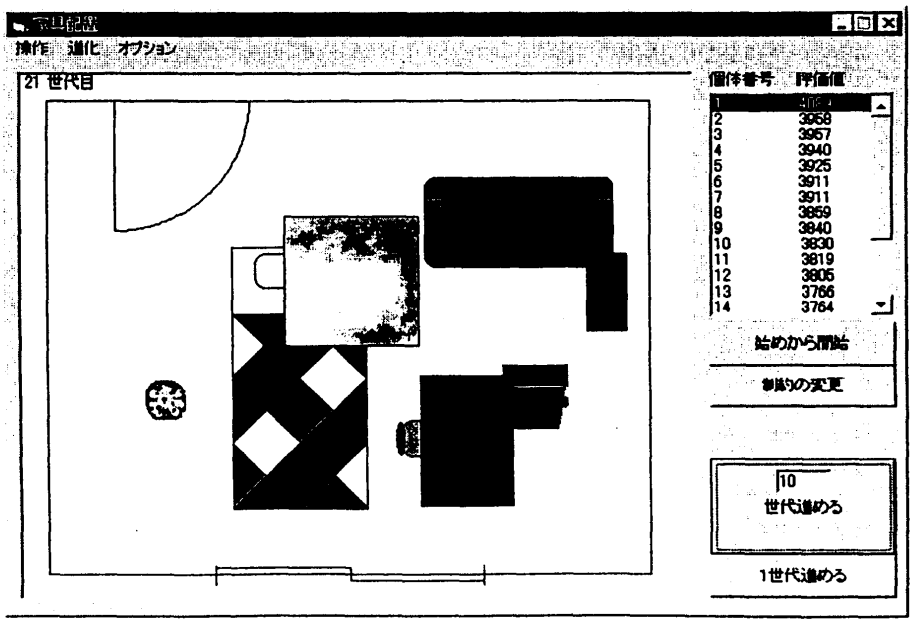



図4　エリート保存選択による最終結果

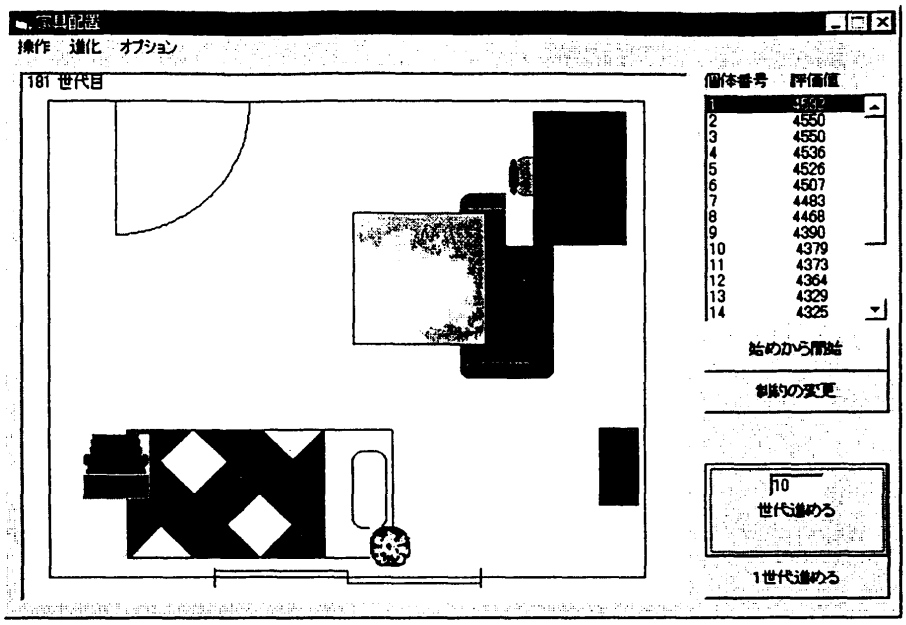
ルーレット選択では、残す遺伝子個体を選ぶときに、ある程度の適応度を持つものからランダムに選ぶ。従って、エリート選択と異なって、常に評価値の高いものが選ばれるとは限らない。このことの良い点は、多様な種類の解を探してみるのに適している。図5の（a）から（d）に世代の進化に沿って四つの場合が図示されている。これらは、制約条件が同じであるにも拘わらず、例えばベッドの位置と向きが変わって行くのが分かる。制約条件についてみると、一部は満たされていても他の条件が不満足という状況が繰り返される。従って、多様な解を導き出せる可能性を持っているが、ルーレット選択だけに頼っていると、ある所から評価値が停滞してしまって、満足の行く解に到達できない。



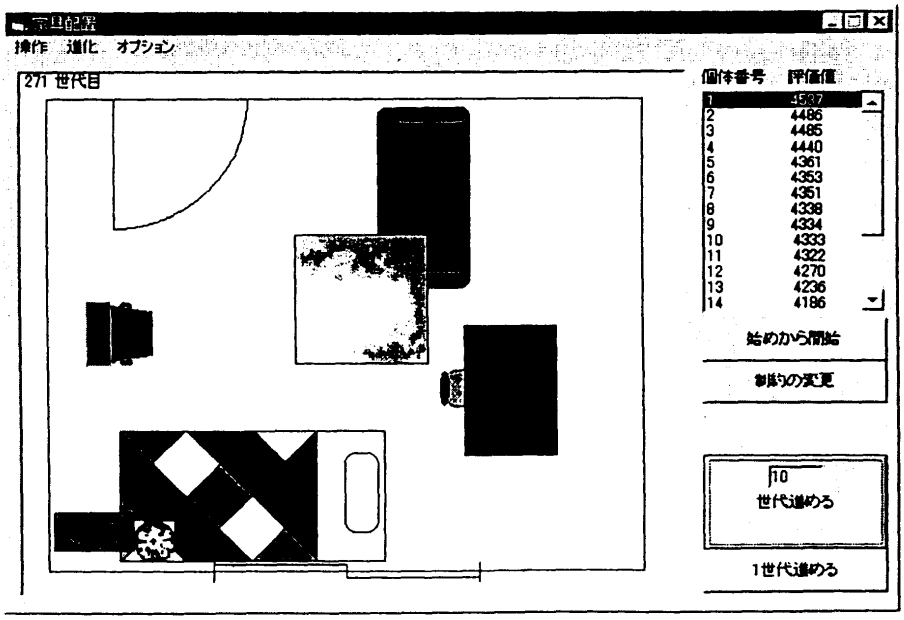

（a）


操作　進化　オプション
21 世代目
個体番号　評価値
1 3968
2 3958
3 3957
4 3940
5 3925
6 3911
7 3911
8 3859
9 3840
10 3830
11 3819
12 3805
13 3766
14 3764
始めから開始
制約の変更
10
世代進める
1世代進める

（b）


操作　進化　オプション
181 世代目
個体番号　評価値
1 4562
2 4550
3 4550
4 4536
5 4526
6 4507
7 4483
8 4468
9 4390
10 4379
11 4373
12 4364
13 4329
14 4325
始めから開始
制約の変更
10
世代進める
1世代進める

（c）




（d）

図５　ルーレット選択による進化

このことをもう少し詳しく見たのが、図６である。世代の進化と共に、エリート保存選択では、制約条件の調整が進み大きく配置が変わることはないので一つの解に到達しているのが分かる。一方ルーレット選択では、特定の配置に基づいて評価値を高める機能が弱くなり、いろいろな配置を試すように進化が進む。図６の進化の状況は例示にすぎないが、この傾向は一般的に見ることができて、初期世代の設定を変えて30例について進化を調べ、評価値の最大、最小、平均値を調べると表１のようになる。

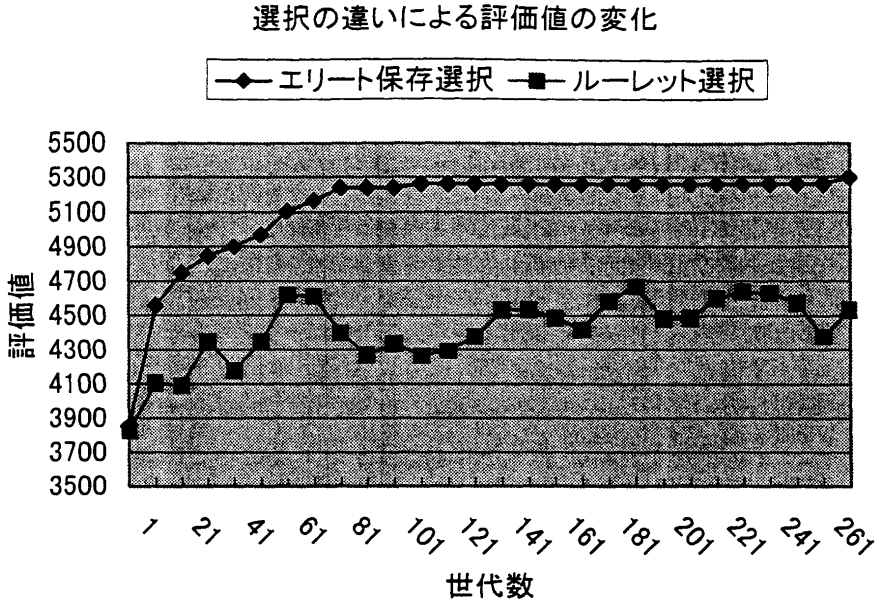



図６　進化における選択法の比較

表1 二つの選択の方法による評価値の平均

| | 最大評価値 | 最小評価値 | 平 均 |
|---|---|---|---|
| エリート保存選択 | 5358 | 5104 | 5263 |
| ルーレット選択 | 4669 | 4181 | 4451 |

それぞれの手法が特徴を持っているので、機能を使い分けるとすると、ルーレット選択でいろいろな配置を試み、適当なところでエリート選択に切り替えるのが良いことが分かる。ただし、配置によっては制約を十分満たす解に到達し得ない場合もある。選択法の切り替え自体をシステムに判断させるのは、興味深いテーマではあるが、現在の所その手段はわかっておらず、その代わりに、進化を進める途中でも制約設定画面を表示して、ユーザーの判断で、切り替えができるようになっている。また、ユーザー指定の制約も重みを途中でも変更可能であり、特に強調したい制約の重みを途中で変更することもできる。

なお、与えられた条件によっては行き詰まってしまう場合も当然起こる。部屋の中に多くの家具を詰め込みすぎて配置ができないとか、制約条件による要求が多すぎてその全ての要求を満たせない場合にこれが起こる。このような場合、制約充足問題では過制約と呼ばれているが、遺伝的アルゴリズムでは、許容解が存在しないという判断はできない。これは、数理計画法を用いる場合に比べると弱点になるが、解の探索に関するアプローチの違いなので止むを得ない。制約の要求が多すぎる場合、このシステムでは遺伝子操作で配置を提示して来るが、ある所から適応度評価値が上がらなくなって、進化が進まなくなる。家具配置の問題では、配置の図示によってユーザーとのインタラクションを進めるので、最後はユーザーの判断になる。

ここに述べた応用例でのユーザーから見た実行時間は、CPU Celeron 366MHz駆動（メモリー128MByte）で、100世代あたりの平均実行時間は2.3秒であった。実際に最後の結果を得るのには数百世代を必要とするが、同じ問題については、その計算の性質上世代間の大きな時間の差はなく、途中でインタラクションを取りながら処理を進めるので、実用上十分であると思われる。

## 5. まとめ

ここで述べた手法の有用性を定量的に評価するのは困難であるが、試作システムを使ってみることによって、以下の点が明らかになった。

(1) 制約条件を直接遺伝子に組み込むことは、一般的に言って困難であるが、今回の研究では、部屋の大きさに拘わらず部屋の中に置くという条件は遺伝子に組み込み、ユーザー指定の制約については、適応度評価に入れることでシステムが実現できた。

(2) 制約条件の一部として、ユーザーの嗜好を入力して遺伝的アルゴリズムを働かせることの有用性が示された。

(3) エリート保存選択による遺伝子操作では、制約を満たすための調整に有効であり、ルーレット選択の手法では、多様な解を導くのに有効であることが分かった。

以上を総合すると、数理計画法によって厳密な最適解を見出すのとは別の観点から、遺伝的アルゴリズムの特徴を活かして解を導けることが分かった。実際問題に適用する場合、人がプランニングを実行するときに、手がかりを提供するツールとして利用できると思われる。

なお、前節の最後に述べた行き詰まりを生じる場合、許容解が存在しない場合もあれば、許容解

があるとしてもそれが条件によっては見つけにくい場合も起こる。このような場合のユーザーとのインタラクションの取り方には改良の余地が残されている。

さらに今後の課題として、今回の試作システムを汎用性のあるものにする問題がある。配置する家具の数が増えると遺伝子のビット列を長くする必要がある。また、家具の種類が増えた場合、計算に必要なのはその奥行きと幅のサイズである。これらはデータの追加であるから、プログラムに入力することは困難ではないが、ユーザーとのインタラクションにおける図示のためにイメージが必要になる。このことを考えると、ユーザーからの要求が出ると思われる家具については、なるべくいろいろな種類を考えて、データと図またはイメージを用意しておくことが望まれる。もう一つの問題は配置条件の指定の仕方である。これを汎用的にするためには、“近くに置く”、“向かいあわせにする”、“壁に寄せる”などの表現を整理してシステムに登録し、家具の名称とこれらの表現を組み合わせて配置条件を指定する方法が考えられる。この問題は一部言葉の理解に関係するので、現在はまだ実現されていないが、今後検討して行きたい課題である。

## 参考文献

1）Toshinori Munakata: Genetic Algorithms and Evolutionary Computing in Fundamentals of the New Artificial Intelligence, Springer-Verlag, 1998.
2）坂和正敏，田中雅博：遺伝的アルゴリズム，朝倉書店，1995.
3）電気学会編：遺伝アルゴリズムとニューラルネット，コロナ社，1998.
4）増井俊之：遺伝的アルゴリズムの図形配置問題への応用，北野宏明編，遺伝的アルゴリズム第6章，産業図書，1993.