

プログラミングコンテストにおける最適配置問題のための 遺伝的アルゴリズムを用いた解法

宮本 行庸* 大西 賢亮*
田村 滋基* 渡邊 俊介*

A Solution Based on Genetic Algorithm for Floor Planning Problems of Programming Contest

Yukinobu MIYAMOTO, Kensuke OHNISHI
Shigeki TAMURA, Shunsuke WATANABE

ABSTRACT

Genetic Algorithm (GA) is one of the searching methods and has recently received much attention as a problem solver for optimization problems. It does not need learning and returns some of the semi-optimal solutions quickly; therefore, many researchers apply GA to optimization problems. The issues for using GA are how to determine the genes and evaluation functions, and they are usually tuned for each problem in an ad-hoc way. In this paper, we propose an example of adapting GA for floor planning problems provided by the 14th Programming Contest for All-Japan College of Technology. We challenged in this contest and show the result of the contest.

KEY WORDS: genetic algorithm, floor planning problem, programming contest

1. はじめに

遺伝的アルゴリズム (Genetic Algorithm: GA)^{1),2),3)} は、生物の進化プロセスを模して作られた多点探索に基づく最適化手法の一種である。GA には、解の表現形式や評価方法に制約がないため、広範囲な分野に適用可能であり、これまでに多くの研究成果をあげてきた。また、学習が不要であり、準最適解への収束が早いことが知られているため、厳密な解を必要とはしないが早期に解を求めたい問題に適している。

このような問題の例として、最適配置問題が考えられる。この問題は、定められた領域の中に制限時間内になるべく多くの色々な形の平面を敷き詰めることを目的とするものである。

本稿では、GA をこれらの問題に適用する際に、個体をひとつの解に当てはめ、優秀個体の選択を繰り返しながら非適合解を淘汰する手法を適用する。また、各世代に

おける評価方法として、敷き詰めによって生じた隙間を最小化する解をより高く評価する方法を採用している。このような世代交代を繰り返し、隙間が最小と想定される段階で探索を終了する。

本手法を実問題に適用した例として、第 14 回全国高等専門学校プログラミングコンテスト競技部門 (以下、プロコン)⁴⁾ において出題された問題をあげる。我々は本手法を実装したシステムを用いてプロコンに参加し、その結果と反省点を以下の章で述べる。

2. 最適化問題と GA

最適化問題とは、制約条件の下である目的関数を最小化 (最大化) する数理計画問題であり、一般に次のように記述される。

$$\begin{aligned} \min_x f(\mathbf{x}) \\ \text{subject to } \mathbf{x} \in F \\ F \subseteq X \end{aligned}$$

*電気情報工学科

ここで、基本空間 X およびその部分集合の解空間 (可能領域) F は、組み合わせ的・離散的なものである。

可能領域が組み合わせ集合となる最適化問題では、一般に解を数え上げるという列挙法的なアプローチとならざるを得ない。そこで、問題に関する先験情報を利用して実際に列挙・探索する範囲を絞り込むことが重要となるが、質のよい解を得るためにはできるだけ広い領域を探索することが望ましく、一方、計算時間を短縮するためには先験情報によってできるだけ探索領域を狭めることが望ましい。これらの相矛盾する二つの要求はトレードオフとなるが、GA ではこの二つをバランスさせるように設計される。したがって、GA によって常に最適解が得られるという保証はなく、その意味で、計算終了時点での解を準最適解と呼ぶことがある。

3. GA による最適配置問題の解法

最初に、本手法が対象としている、最適化問題の一種である最適配置問題の概要を示す。対象問題には、大まかに以下のような特徴がある。

- ・ 一定の領域の中に、はみ出さないように立体の展開図を敷き詰める。
- ・ 立体には色々な形のものが多種存在し、ふたのない (閉じていない) 形状も存在する。
- ・ 敷き詰めた結果、隙間の最小化を目的とする。

本手法では、対象問題における立体とその姿勢を遺伝子に、部品の配置順を遺伝子配列に、個体を解に当てはめ、優秀個体の選択を繰り返して準最適解に絞り込む手法を採用している。手順としては、以下のようなアルゴリズムとして記述される。

初期個体の生成：

1. すべての立体に立体番号を付与する。
2. 各立体のすべての展開図に番号を付与し、これらの立体と展開図との組み合わせを遺伝子とする。
3. 2. の作業をすべての立体について行う。
4. 各立体をランダムに遺伝子座に配置して、ひとつの個体を生成する。
5. 4. で生成した個体の各立体から、ランダムに一つの展開図を選択し、これを遺伝子配列とする。

6. 上記 4. - 5.の手続きを、個体数だけ繰り返す。

個体の評価：

7. 個体から配列順に遺伝子を取り出し、立体の展開図に変換する。
8. 領域の原点 (左上隅) から配置可能な場所を探索し、その展開図で立体を配置する (図 1)。

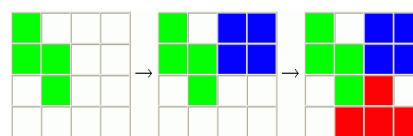


図 1 展開図の配置

9. 7. - 8.の手続きをすべての立体を敷き詰めるか、配置不能になるまで繰り返す。
10. 配置を停止した時点の占有面積を絶対評価値とし、それらの絶対評価値の大きな順に個体を整列する。
11. 絶対評価値が同じ場合、平均開散度を求める。平均開散度は以下の式で得られ、隙間が同じ数の場合は、なるべく大きな面積の隙間を持つ配置を高く評価する仕組みとなっている。

$$\text{平均開散度} = \frac{1}{n} \sum_{i=1}^n S_i$$

ただし、 n は隙間の個数、 S_i は各隙間の面積を表している。図 2 における平均開散度は $23 / 8$ となる。

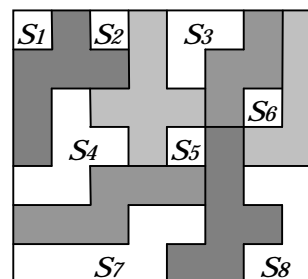


図 2 平均開散度計算の例

交叉：

12. 交叉のためのカップリングは、以下の 3 つの手法から等確率で選択する。なお、1 回の交叉での戦略はどれか 1 つとする。また、このときに

用いる評価値を内部評価値といい、絶対評価値と平均開散度の和で求められる。

(a) エリート戦略

内部評価値により整列された個体を、上から順に組にしていく。

(b) ルーレット戦略

内部評価値の総和に対する比率に比例して組を作る。

(c) ランダム戦略

一様乱数に基づいて組を作る。

13. 組となった親を元に、交叉を行う。交叉法は以下の 3 つの手法から各組ごとに等確率で選択する。

(a) 一点交叉

ある一点においてその前後で組となる遺伝子を入れ替える。

(b) 二点交叉

重複のない二点において、組となる遺伝子を入れ替える。

(c) ローテート

組となる個体の特定の遺伝子の位置が同じになるように、一方の個体の遺伝子の位置を横にずらす。

突然変異：

14. 交叉によって生成された個体のうち、突然変異を実行する個体を一定の小さな確率で選択する。
15. 14. で選択された各個体について、ランダムにそれぞれ 2 つの遺伝子を選択し、各遺伝子同士を入れ替える。

個体の再評価：

16. 交叉・突然変異が適用された新世代個体について、前述の 7. - 11. の評価法によって再評価・整列する。

淘汰：

17. 淘汰の操作は、以下の 3 つの手法から等確率で選択される。いずれの手法においても、絶対評価値または内部評価値が最大のエリート個体は淘汰されない。

(a) エリート戦略

内部評価値の低い個体を淘汰する。

(b) ルーレット戦略

内部評価値の逆数に比例して選択・淘汰する。

(c) ランダム戦略

一様乱数に基づいて淘汰する。

進化と停止条件：

18. 上記 12. - 17. の操作を、あらかじめ設定された世代数だけ繰り返す。
19. 最大評価値 (隙間の数が 0) を持つ個体が発生すれば、自動的に進化計算を即時停止する。
20. 設定世代数終了後の評価値に変化が見られない (これ以上よい解が期待できない) と人間が判断した場合には、人手によって進化計算を停止する。

以上のような手続きにより、停止時点における最大評価を持つ個体を対象問題における準最適解 (あるいは最適解) と判断し、最適配置問題の解とする。

4. システム構成

本システムは、大きく入力部、計算部、出力部の 3 つの部位より構成され、さらに計算部は展開部と敷き詰め部に分けられる。図 3 に、システム構成を示す。各部位の機能は以下のようにになっている。

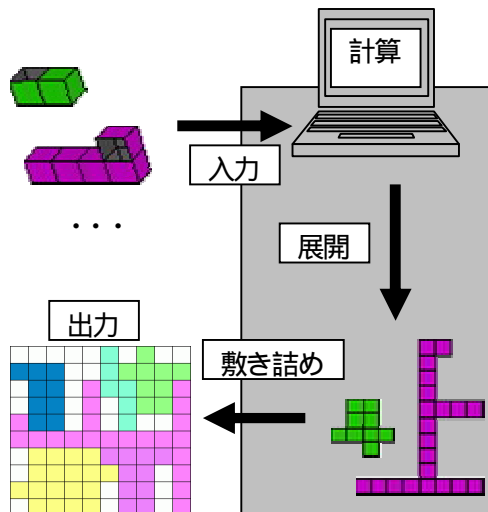


図 3 システム構成

入力部：

- 与えられた立体の形状を、人手により視覚的にコンピュータに入力していく (図 4) 。
- 入力デバイスを用いて、立体をひとつずつ与えられたマス目に入力する。
- 入力画面は 3 次元表示され、視点を変更して見やすい位置から入力できる。

- ・ 立体的に隠れて見えない部分の補助のために、高さ方向の断面も表示する。
- ・ 1つ目の立体入力完了した時点から並列に展開計算を始め、引き続き2つ目以降の立体の入力を受け付ける。

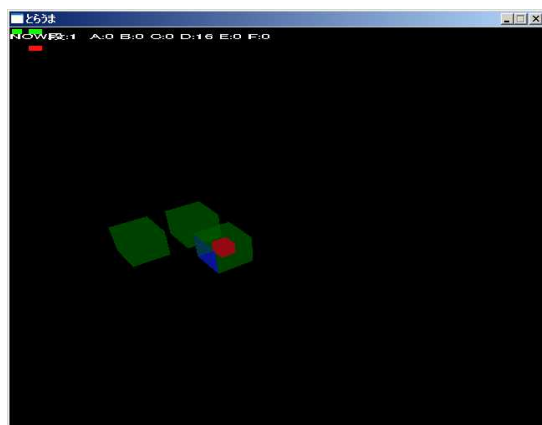


図4 入力部

計算部 (展開):

- ・ 入力完了した立体から順に、プロセスを立ち上げて展開を開始する。
- ・ 各面について辺の数を列挙し、辺ごとに共有する面をリストアップする。
- ・ ある一つの面に注目し、他の面との経路が唯一(両者以外の面を経由してもよい)になるように、ランダムに辺を切る。
- ・ 展開図が複数に分断されたり、展開した面に重なりが生じたりした場合は、辺の切断をやり直す。
- ・ 着目した面について適切な展開図が得られない場合には、他の面に着目して同様の手続きを行う。
- ・ 上記の一連の作業を、すべての立体について行う。

計算部 (敷き詰め):

- ・ 第3章に述べた手続きに従って、解候補をひとつに絞り込む。

出力部:

- ・ 領域の左上隅より順に、敷き詰める展開図を表示していく(図5)。
- ・ キー入力により、次の展開図の表示へと進む。
- ・ 各展開図は、異なった彩色で表示される。

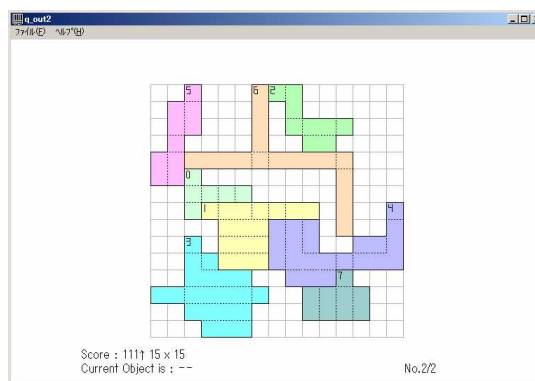


図5 出力部

5. 実戦結果

本システムを適用して、プロコンに参戦したときの問題と結果を以下に示す。問題の詳細についてはプロコンの公式サイト⁴⁾を参照されたい。

5.1 出題内容

プロコンで出題された競技の概要は以下のようになっている。

- ・ あらかじめ与えられた長方形領域に、複数の立体を展開したものを敷き詰める。
- ・ 立体は規定の大きさの正方形の組み合わせを面として構成される。
- ・ 立体にはふたのない箱のような、閉じていない立体も含まれる。
- ・ 隙間がなるべく少ない方がよく、結果が同じ隙間数であれば、競技時間の短い方がよい解と判断される。
- ・ 競技者は自身のコンピュータにより解を導き出し、その後解答用コンピュータに入力する。
- ・ 領域が隙間なく満たされないこともある。
- ・ 立体が余る場合もある。

以上の条件より、立体の形状の入力・得られた出力結果の解答用コンピュータへの入力という人手による作業時間と、コンピュータによる解の計算時間をどのような割合で配分するかが、よい競技結果への鍵となる。

5.2 競技の結果

表1に、決勝戦までの一部の問題と本校が挑戦した結果を、図6に例としてプロコン公式サイトに掲載された決勝戦の問題を示す。なお、表1中の得点は展開された立体が占める領域内の面積をマス目単位で、残り時間は競技時間10分のうち、最終解答を送信した時

点から競技終了までの時間を秒単位で表している。

競技に用いたノート型コンピュータの性能は、CPU が Pentium 4 Dual / 3.0GHz、メモリが 1GB であった。また、本競技においては 1 台のコンピュータのみを用いており、複数台のコンピュータによる通信や協調計算は行っていない。

表 1: 競技の結果

試合		1 回戦	2 回戦	決勝
問題 1 夕	領域サイズ	10×10	11×10	11×11
	領域面積	100	110	121
	出題立体数	8	9	10
	立体総面積	111	120	144
一 位 校	使用立体数	7	7	8
	得点	90	102	118
	残り時間	371	79	247
本 校 結 果	使用立体数	6	5	-
	得点	73	45	-
	残り時間	69	19	-
	順位	4	7	-

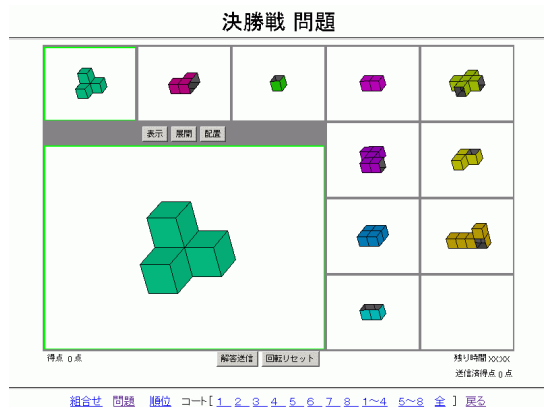


図 6 決勝戦の問題

6. 考察

表 1 および本システムを競技に適用した現場での結果から、以下のような考察が得られる。本大会においては、本校は決勝戦まで進めなかったため、以下では主な敗因について考察する。

まず、解の導出時間である。競技に要した時間は立体形状の入力作業時間が最も多く、次いで解答用コンピュータへの入力作業時間となっており、本システムによる解の計算時間は長くても数秒程度であった。しかしながら、得られた解候補は必ずしも最適に近い解とはなっていなかった。持参したノート型コンピュータの性能は充分であったと考えられるため、よりよい解を求めて引き続き計算を継続すべきであった。競技の性質上、制限時間一杯までの計算は解答入力時間を

失うリスクが伴うため、計算時間と作業時間の配分は現場での判断とならざるを得ないのが現実である。

本大会における最大の反省点は、展開する立体の切断箇所を表示する機能を実装しなかった点にある。入力された立体を展開し、展開図を敷き詰めた結果には十分により解もあったが、展開図から立体を切り開く箇所を頭の中で考えねばならず 2 回戦では結局その解を断念せざるを得なかった。表 1 における本校の 1 回戦の結果においても、計算で得られた解では 7 つの立体を使用しているが、そのうち 1 つの切断箇所を競技時間内に特定できなかったため、残る 6 つの立体で勝負する結果となってしまった。本競技において、敷き詰め最適化に重点を置き、解答表示用のインターフェイスを結果的に軽視する形となったことは、次年度以降に大いに反省すべき点が見出せたと考えられる。

7. おわりに

本稿では、GA を最適配置問題に適用する一解法について述べた。また、最適配置問題の例としてプロコンの問題を取り上げ、本システムを実装して実際にプロコンに参加した状況を述べた。競技の結果には本手法の利点は大きくは反映されなかったが、大会での反省を生かし、プロコンを通して学生にシステム開発の工程管理と運用テストの必要性を提示することができたと考えられる。

次大会以降の対策として、異なるアルゴリズムを完全に並列化し、それぞれに解を求める手法が考えられる。入力部や出力部などのインターフェイス部分のみを共通化し、計算部はそれぞれ独立したアルゴリズムを実装すれば、問題の性質によっては各手法の得意分野を生かした解が得られる可能性がより高まる。本大会においても、上位進出校にはこの手法を採用していたところもあったように見受けられた。今後は、複数のアルゴリズムを並列に実装する際に、ハードウェア面からは十分な性能と複数台による通信手段の検討、ソフトウェア面からは関数やライブラリの共通化・独立化を検討することが、大会でよい結果を得る一つの契機になると考えられる。

参考文献

- 1) 北野宏明：遺伝的アルゴリズム 4，産業図書 (2000)。
- 2) 電気学会編：遺伝アルゴリズムとニューラルネットワーク，コロナ社 (1998)。
- 3) L. Davis 編，嘉数侑昇・他 訳：遺伝アルゴリズムハンドブック，森北出版 (1994)
- 4) 全国高等専門学校プログラミングコンテスト実行委員会編：プログラミングコンテスト公式ウェブサイト，<http://www.procon.gr.jp/> (2003)。