動的遺伝操作切替を用いた遺伝的アルゴリズムによる プロジェクトスケジューリング

学生員 池 内 智 哉 (大阪大学)

正 員 一 階 良 知 (大阪大学)

非会員 荒 木 大 (大阪大学)

正 員 大 川 剛 直 (大阪大学)

正 員 薦 田 憲 久 (大阪大学)

Project Scheduling Using Genetic Algorithm with Adaptable Changing Genetic Operators
Tomoya Ikeuchi, Student Member (Osaka University), Yoshitomo Ikkai, Member (Osaka University),
Dai Araki, Non-member (Osaka University), Takenao Ohkawa, Member (Osaka University),
Norihisa Komoda, Member (Osaka University)

Genetic Algorithm(GA) has been widely used to solve planning problems. However, it requires to determine the optimal values of many genetic parameters, such as population sizes, crossover probability, mutation probability and so on. To make matter worse, the most suitable combination of parameters for one problem is not always optimal for others. Therefore, these parameters should be tuned up whenever the problem changes.

In this paper, we propose an adaptable GA mechanism which has an autonomic parameter tuning for the composition of generic operators. This mechanism raises the probability of the genetic operators which acted effectively, that is, the operator created better individuals than the other operators. And, it adjusts the combinations of genetic parameters as the suitable one for the target problem successively. We applied the adaptable GA mechanism to a project scheduling model (PSM) and evaluated with the manual tuning methods.

キーワード：計画問題, 遺伝的アルゴリズム, 遺伝操作切替, プロジェクトスケジューリングモデル

1. はじめに

遺伝的アルゴリズム (Genetic Algorithm: GA) ⁽¹⁾⁽²⁾は、アルゴリズム自体が単純であるため、比較の実装が容易である。また、個体の評価方法が適切であれば、そのアルゴリズムは幅広く問題に適用できる。また、個体の評価方法を適切に設計すれば、遺伝操作等の探索アルゴリズムは、解を探索するという目的に対しては、データに依存せずに幅広く適用できる。また、遺伝操作自身は、一点交叉など問題の構造に依存せずに一般的な手法が多く知られている。このような理由から、GAによるスケジューリング問題へのアプローチが盛んになされている ^{(3)~(5)}。

ところが、GAの探索効率、すなわち、最適解あるいは

許容解を得るまでに必要な計算時間を短縮するためには、2つの課題がある。どんな遺伝操作を使った方が効率がよいのかと、各遺伝操作の発生確率等のパラメータチューニングが難しいことである。

そこで本研究では、各遺伝操作を用いる確率を世代を重ねるごとに動的に変化させることで、遺伝操作を問題に適した確率で適用できるシステムを提案する。つまり、ある世代を生成した際に有効に作用した遺伝操作の次世代生成時における選択確率を上げる。これにより、用意された遺伝操作のうち、問題に適さない遺伝操作は確率が次第に減少し、淘汰されることになる。これより、問題の特性に対応した遺伝操作の切替が可能になり、スケジュール担当者をパラメータチューニングの煩わしさから解放することが

期待できる。

本論文では、2章に遺伝的アルゴリズムについて3章に遺伝操作切替手法について説明し、4章にプロジェクトスケジューリング問題への適用手法と適用結果を述べる。

2. 遺伝的アルゴリズム

遺伝的アルゴリズム (GA) では、ある世代を形成している個体の集合、すなわち個体群の中で、環境への適合度の高い個体ほど、高い確率で次の世代にその特徴を残していく。さらに、交叉 (crossover) や突然変異 (mutation) によって、次の世代が形成される。このような遺伝操作によって世代を重ねることで、個体はできるだけ環境に適合するように進化していく。つまり、質の良い解集合を逐次生成し、最適解を探索する。個体群に含まれている個体の数を個体群サイズと呼ぶ。

計画問題をGAによって最適解を求めようとすれば、GAのアルゴリズムを実装することは難しくないが、遺伝操作の手法や交叉確率、突然変異確率などの様々なパラメータの決定に膨大な時間を費さなければならない。しかも、計画の立案は、追加・変更などにより、度々行わなくてはならない。その都度莫大な時間をかけて感度解析を行い、もっともらしいパラメータに設定する必要がある。しかも、それらのパラメータの組合せは多様で、感度解析によって得られたパラメータは、暫定的なものであり、最適である確証はない。

3. 遺伝操作の動的切替法を用いたGA

本論文では、交叉・突然変異ともに数種類の操作を用意し、その中で、ある世代を生成した際に有効に働いた遺伝操作の確率を上げることによって遺伝操作の適用確率を動的に変化させ、自動的に最適な交叉・突然変異確率によって遺伝操作を行う動的切替法を提案する。

遺伝操作の動的切替法を用いたGAの概略を図1に示す。解を表現する遺伝子をランダムに個体群サイズの個数だけ生成し、初期解集合を生成する。また、各遺伝操作にそれぞれが選択される確率を設定し、遺伝操作初期確率とする。遺伝操作確率は世代を重ねるごとにふさわしいものに収束していくため、それぞれの遺伝操作初期確率には特に気配りをする必要はない。

遺伝操作の有効性を定義する場合、

1. 極端に良い解を導くことは期待できないが、平均的に良い解を導くことのできる遺伝操作が有効である。
2. 平均的に良い解を導くことはできないが、極端に良い解を導くことのできる遺伝操作が有効である。

という、2通りの考え方がある。1に基づいた方法をとった場合、世代を重ねるごとに確実な解の進化は期待できるが、ダイナミックな解の進化が期待できにくい。すなわち、局所最適解に収束する可能性が高い。それに対して、第2の考え方では、局所最適解に陥ることを避けるような、ダイナミックな解の進化の起こる可能性が高く、今回はこの

図1 システム構成図

Figure 1. Structure of the system.

考え方に基づいて、遺伝操作の有効性を定義する。

有効性を表す尺度を求めるために、各遺伝操作の有効度 (E: effectiveness) と暫定有効度 (TE: tentative effectiveness) を定義する。各遺伝操作によって生成された全ての個体の中での、その親との評価値の差の最大値を暫定有効度とよび、遺伝操作 i の暫定有効度を $TE(i)$ で表す。また、生成された個体の評価値が、全て親の評価値より悪い遺伝操作の暫定有効度は0と定義する。図2はこの考え方を模式的に示したものであり、図中の黒丸が子個体、その隣の数値が親との評価値の差を示している。

図2 遺伝操作暫定有効度の決定

Figure 2. Definition of tentative effectiveness.

また、既に確率の高い遺伝操作は生成する個体の絶対値が大きいため、優秀な個体を生成する確率がもともと高い。しかし、わずかな確率で生成する個体数が少ないにもかかわらず優秀な個体を生成する遺伝操作の場合は、その遺伝操作が、極めて有効であると考えられる。そこで、第 k 世代を生成する際の遺伝操作 i の選択確率を $P_k(i)$ とすると、遺伝操作 i の有効度 $E(i)$ は $TE(i)$ と $P_k(i)$ を用いて、
$$E(i) = \frac{TE(i)}{P_k(i)^2} \quad (1)$$
 と定義する。選択確率の低い遺伝操作は劇的に有効度を大きくしなければ、次世代における選択確率に影響しにくい。そのため、分母には $P_k(i)$ の2乗を用いた。これにより、上述の考えを満たす有効度を表現できる。

現在の暫定解を更新するような個体を生成することは、GAによる探索において本質的な部分である。したがって、このような解を生成する遺伝子は有効であると考えられる。そこで、暫定解より優秀な解を生成した遺伝操作には、有

効度に加えて、ボーナスポイント (BP = 一定) もプラスする。つまり、

$$P'_{k+1}(i) = \begin{cases} P_k(i) + \alpha E(i) & ; \text{ボーナスを加えない操作} \\ P_k(i) + \alpha E(i) + BP & ; \text{ボーナスを加える操作} \end{cases} \quad (2)$$

とする。ここでαはPとEのスケールを一致させるための定数である。最後に正規化を行ない、次世代生成時に用いる確率 $P_{k+1}(i)$ を計算する。

$$P_{k+1}(i) = \frac{P'_{k+1}(i)}{\sum_l P'_{k+1}(l)} \quad (3)$$

以上の考え方を模式的に表したのが図3である。この図では、遺伝操作は a, b, c の3種類であり、第k世代を生成する際の確率がそれぞれ $P_k(a), P_k(b), P_k(c)$ であった。遺伝操作bによって生成された個体が、暫定解より優れていたため、遺伝操作bには、ボーナスポイントも加算されている。

図 3 遺伝操作確率切替
Figure 3. Changing genetic operators

4. プロジェクトスケジューリングへの適用

〈4.1〉プロジェクトスケジューリングモデル

遺伝操作切替型 GA を表1~3で定義されるプロジェクトスケジューリングモデル (Project Scheduling Model: PSM) に適用した。表1はジョブの作業時間と必要とするリソース及びその数を表したものである。以下、表2ではジョブの前後関係を、表3では、リソースの容量を表している。

これらの諸条件のもとで、プロジェクトの完了時間を可能な限り早めるスケジュールを求める。

表 1 ジョブ番号、作業時間、必要リソース数

Table 1. Job data.

作業名	作業時間	R_a	R_b	R_c	R_d
A	15	0	0	1	0
B	5	2	0	1	0
C	10	1	1	0	0
D	60	0	2	1	1
E	20	0	0	1	0
F	10	1	2	0	1

R_k : リソース名

表 2 ジョブの前後関係

Table 2. Order of jobs.

先行作業番号	後続作業番号
A	B
A	C
B	D
C	E
D	F
E	F

表 3 資源制約

Table 3. Resource condition.

Resource 名	最大使用量
Resource.Ra	2
Resource.Rb	2
Resource.Rc	2
Resource.Rd	1

〈4.2〉遺伝子と再生法

ジョブの割り付け優先順位をジョブ番号の並びとして遺伝子を表現する。表1~表3のプロジェクトにおいて、図4のノード2の状況にあった場合、割り付け可能ジョブ B,C のうち、遺伝子中に早く現れているのはBであるので、Bを最早時刻に割り当てる。

図 4 遺伝子へのコーディング

Figure 4. Coding.

各個体の評価値は、実際に計画エンジンにおいてスケジューリングを行い、作成したスケジュールのプロジェクト終了時間で表す。つまり、評価値が低い個体ほど最適解に近いことになる。

再生の手法にはエリート保存選択を採用した。遺伝操作をして新しい世代を生成した際に、親世代の上位4個体は無条件でそのまま子世代に残ることとした。また、交叉する対象を選択する際にも、評価値の良い個体ほど選ばれやすいようにした。

〈4.3〉 交叉

交叉は2つの親の遺伝子を入れ換えるという操作である。ところが、前節の方法でコーディングを行うと、遺伝子はジョブの重複を許さない順列で表されている。このような遺伝子に対しは、順序交叉法が有効である。つまり、交叉部分を入れ換えるのではなく、交叉部分にあるジョブ番号を、相手の遺伝子中の相対的な順序、つまり、相手の遺伝子に現れる順序に並び替える。用意した交叉方法は以下の2通り。

- 2点交叉 (two-point crossover)

2つの交叉点で切断された中央のパーツに注目し、この部分に含まれているジョブ番号を相手の遺伝子中に現れる順に並び替える (図5)。しかし、この操作では遺伝子の両サイドについてはなかなか交叉の対象となりにくい。そこで、2つの交叉点で切断された両サイドに注目し、それぞれのパーツについて中央部を並び替えた時と同様の交叉を行う方法も採用する。中央部の交叉を CenterX、両サイドの交叉を SideX と呼ぶことにする。
- 一様交叉 (uniform crossover: UniformX)

遺伝子中のジョブ番号それぞれについて50%の確率で、マスクをかける。マスクのかからなかった番号を、相手の遺伝子に現れる順序に並び替えることで、任意個の交叉点を取ることと等しくなる (図6)。

順序交叉法を用いた場合、交叉部分は2つの遺伝子で異なっても交叉可能である。交叉区間がいかなるものであれ、その区間にある数字を相手の遺伝子に現れる順に並べ変えるのであるから、容易に交叉することができる (図7)。CenterX、SideX、UniformX について、交叉部またはマスクをかける位置が2つの遺伝子で異なる CenterX-d、SideX-d、UniformX-d も併せて用意する。

〈4.4〉 突然変異

突然変異の手法は、以下の5つを用意した。

- Inversion ランダムに選んだ2点間の順序を反転させる。
- Advance ランダムに1つ選び、それを前方に挿入する。挿入点もランダムに決める。
- Swap 交換点をランダムに2点選び、交換する。
- Recombination 遺伝子中のある区間をランダムに並び替える。並び替える区間が長すぎると新しい遺伝子をランダムに生成することと等しくなってしまうため、平均5、分散1の正規分布乱数により、その長さを決定する。
- Replacement いくつかの遺伝子座をピックアップし、それらの値をランダムに並べ替える。recombinationと同様、ピックアップする数が多すぎると、遺伝子全体の並べ替えとなってしまう。そこで、この操作についてもピックアップする個数を平均5、分散1の正規分布乱数によって決定する。

図5 2点交叉 (CenterX)
Figure 5. Crossover Center.

図6 一様交叉
Figure 6. Crossover Uniform

図7 2点交叉 (CenterX-d)
Figure 7. Crossover Center-d

〈4.5〉 シミュレーション結果

本切替手法の有用性を検証するために、実際に計算機に実装して、感度解析を人手により行った場合との比較シミュレーションを個体群サイズ200、反復世代数1500として行った。また、遺伝操作確率の初期値は、全ての遺伝操作を等確率に設定し、式(2)の α の値は、 1×10^{-5} とした。本手法を適用するプロジェクトスケジューリングモデルを作成する際に用いた仕様を表4に示す。

表4 評価問題作成仕様
Table 4. Specifications of generating problem.

ジョブ数	平均100, 分散49の正規分布
作業時間	平均100, 分散100の正規分布
総リソース数	5 (各リソース最大容量5)
後続作業数	平均3, 分散1の正規分布

シミュレーションでは、表4の仕様に基づいて作成した10題のPSMを、1) 遺伝操作切替えGA、2) 遺伝操作確率を表5のパターン1に固定したGA、3) 遺伝操作確率を表5のパターン2に固定したGAの3つの手法を用いて解くことを行った。今回これらの比較対象確率パターンは、予め行なった感度解析の結果の中で最優秀のものである。それぞれの手法で各問題を初期乱数を変えて10回シミュレーションした結果を表6に示す。なお、表中の数値は上段、中

表 5 用いた確率パターン
Table 5. Probability pattern.

遺伝操作	確率パターン No.	
	1	2
Inversion	6.0	2.5
Advance	6.0	2.5
Swap	6.0	20.0
Recombination	6.0	2.5
Replacement	6.0	2.5
CenterX-d	11.7	10.0
SideX-d	11.7	10.0
CenterX	11.7	10.0
SideX	11.7	10.0
UniformX	11.7	20.0
UniformX-d	11.7	10.0

断、下段それぞれ、10 回の最良値、平均値、最悪値である。さらに、下線の引いてある数値は、同一問題・同一評価基準での 3 手法中のもっとも良いものを表している。

なお今回のシミュレーションは、Silicon Graphics ワークステーション INDY(CPU: MIPS R4600, 100MHz, 64MB) を使用し、1 題あたり 200 個体 1500 世代の計算で 200～240 分を要した。また、この計算時間は遺伝操作切替 GA、確率固定 GA のどちらの場合にも共通するものであった。つまり、今回の手法において遺伝操作切替部ではほとんど計算時間がかかっていないと言える。

この表を見ると、動的切替え GA が多くの問題において良い結果となっている。さらに、固定パターンの確率を求める際の感度解析に多大な時間を要することを考えあわせれば、本手法を用いることにより、問題が変化してもそのたびにパラメータチューニングを行うことなく優秀な解を求めることが可能であるといえる。更に言及すれば、本手法を用いた場合、3 つの手法でそれぞれ 10 回行ったシミュレーション結果の平均値が 10 問中 7 問で、最悪値が 10 問中 8 問で遺伝操作確率を固定する手法を上回る結果が得られている。言い替えれば、本手法では極端に悪い解に収束することが極めて少ない。したがって、厳密な最適解ではなく、ある程度の優秀な解を求めることで十分であるような場合には、特に有効な手法であるといえる。

次に、本手法を用いた際に遺伝操作確率が実際に変化した様子の一例を図 8 に示す。このグラフでは、横軸に探索の世代を、縦軸には各遺伝操作の選択確率を累積したものを表現している。つまり、横軸と一番下の折れ線との間隔が Inversion が選択された確率、1 番下の折れ線とその 1 つ上の折れ線との間隔が Advance が選択された確率、といった具合である。これにより、遺伝操作動的切替が行なわれていることが確認できる。

表 6 シミュレーション結果
Table 6. Result of simulation.

問題 番号	遺伝操作確 率切替 GA	遺伝操作確率固定 GA	
		パターン 1	パターン 2
1	8100	8176	8207
	<u>8222.1</u>	8320.4	8353.4
	<u>8426</u>	8525	8536
2	6105	6112	<u>6064</u>
	<u>6261.1</u>	6293.2	6281.7
	<u>6393</u>	<u>6393</u>	6632
3	3838	3824	<u>3793</u>
	3873.6	3870.4	<u>3864.0</u>
	<u>3953</u>	<u>3910</u>	3972
4	8609	8561	<u>8528</u>
	<u>8771.6</u>	8818.6	8782.1
	<u>8912</u>	9129	9123
5	5897	<u>5886</u>	5926
	5976.5	<u>5970.4</u>	6033.5
	<u>6097</u>	6119	6161
6	5446	5537	<u>5392</u>
	<u>5531.6</u>	5598.7	5536.4
	<u>5653</u>	5668	5761
7	<u>4404</u>	4405	4410
	<u>4462.6</u>	4499.2	4496.9
	<u>4504</u>	4584	4532
8	5723	<u>5711</u>	5725
	<u>5839.9</u>	5875.5	5847.2
	<u>5982</u>	6094	6043
9	<u>7168</u>	7234	7353
	7411.4	<u>7393.0</u>	7453.8
	7557	7560	<u>7548</u>
10	<u>8706</u>	8829	8803
	<u>8889.8</u>	9011.4	8980.5
	<u>9052</u>	9195	9229

最良値
平均値
最悪値

このグラフは、次の 4 つの部分に分けて考えることができる。

第 1 期: 初期世代～50 世代過ぎ

探索の草創期に当たり、収束すべき確率の組合せを模索しながら、遺伝操作の確率が激しく変動している。

第 2 期: ～250 世代過ぎ

200 世代直前に 1 度 UniformX-d の確率が増加した以外は、激しい確率変動が見られない。ここでは、ある程度最適に近い遺伝操作確率の組合せで探索が進んでいると考えられる。

第 3 期: ～400 世代辺り

UniformX-d の確率が段階的に増加している。第 2 期

図8 遺伝操作確率の推移

Figure 8. Change of operators probability.

の組合せでは、UniformX-dの確率が少なすぎため、このような変化が見られる。さらに、400世代直前に、それまで0.03程であったAdvanceの確率が0.15強に増加している。この間は、第2期の暫定的な最適組合せから最終的に収束すべき組合せに向けての変化であると考えられる。

第4期: それ以降

この段階では第2期同様大きな確率の変動はほとんど見られない。また、グラフの様子から、最適な遺伝操作確率の組合せへの収束が行なわれているものと考えられる。

5. 結論

遺伝的アルゴリズムにおいて遺伝操作を動的に切替える機能を追加し、異なる問題に対して自動的に適した遺伝操作確率に遷移する手法を提案した。さらに、この手法をプロジェクトスケジューリングモデルに適用することにより、その有効性を確認した。

(平成8年5月31日受付, 平成8年10月30日再受付)

文献

- (1) 坂和 正敏, 田中 雅博: 遺伝的アルゴリズム, 朝倉書店 (1995)
- (2) 小林 重信: “遺伝的アルゴリズムの現状と課題,” 計測と制御, vol.32, no.1, pp.2-9 (Jan. 1993)
- (3) M.Gen, Y.Tsujimura and E.Kubota: “Solving Job-Shop Scheduling Problem,” in Proc. of 1994 IEEE Int. Conf. on Systems, Man and Cybernetics, pp.1577-1582 (Oct. 1994)
- (4) L.Tautou and H.Pierreval: “Using Evolutionary Algorithms and Simulation for The Optimization of Manufacturing Systems,” in Proc. of the IEEE Symposium on Technologies and Factory Automation, pp.509-516 (Oct. 1995)
- (5) 柳浦 睦憲, 茨木 俊秀: “遺伝的アルゴリズムと局所探索法のロバスト性,” システム制御情報学会生産スケジューリング・シンポジウム'95 講演論文集, pp.79-90 (Oct. 1995)

池内 智哉

(学生員) 1973年12月19日生まれ。96年3月大阪大学工学部情報システム工学科卒業。同年同4月大阪大学大学院工学研究科情報システム工学専攻博士前期課程入学。生産計画, 知識工学に関する研究に従事。

一階 良知

(正員) 1969年10月23日生まれ。95年3月大阪大学大学院情報システム工学専攻博士前期課程修了。現在, 同情報システム工学専攻助手。生産計画, 知識工学に関する研究に従事。94年度電気関連学会関西支部連合大会奨励賞受賞。IEEEなどの会員。

荒木 大

(非会員) 1962年11月1日生まれ。88年3月大阪大学大学院工学研究科通信工学専攻博士前期課程修了。同年(株)東芝入社。現在, 研究開発センター, システム・ソフトウェア生産技術研究所において, 計画型エキスパートシステム, 知識獲得支援技術の研究と応用に従事。96年4月大阪大学大学院工学研究科情報システム工学専攻博士後期課程入学。IEEEなどの会員。

大川 剛直

(正員) 1963年5月1日生まれ。88年3月大阪大学大学院通信工学専攻博士前期課程修了。90年同後期課程退学。現在, 同工学部情報システム工学専攻助教授。工博。知識獲得, 機械学習に関する研究に従事。93年度電気関連学会関西支部連合大会奨励賞受賞。IEEEなどの会員。

薦田 憲久

(正員) 1950年2月27日生まれ。74年3月大阪大学大学院電気工学専攻修士課程修了。同年(株)日立製作所入社。81~82年UCLA留学。91年4月大阪大学工学部情報システム工学科助教授, 現在, 同教授。工博。システム工学, 知識情報処理の研究に従事。計測自動制御学会86年度論文賞, 同学会87年度技術賞受賞。IEEEなどの会員。