

大学時間割作成への遺伝的アルゴリズムの適用

土 性 雅 史 (大学院工学研究科情報工学専攻)*¹
小 野 俊 彦 (情 報 工 学 科)

Application of Genetic Algorithms to University Timetable Problem

Masafumi DOSYOU (Communication and Computer Engineering, Graduate school of Engineering)
Toshihiko ONO (Department of Computer Science and Engineering)

Abstract

The edition of the timetable for a university is one of the difficult and time-consuming tasks, because there are a lot of constraints to consider. The method proposed here is to automatize it by genetic algorithms (GAs), which adopt a multiple gene construction. Each gene undergoes independent and parallel genetic operations, such as a crossover and mutation. Special considerations are given to a double hour class whose time is twice as much as an ordinary one. The allocation of classrooms is done by separate algorithms cooperating with GAs.

Key words: *Genetic algorithms, Timetable, Optimizing problem, Soft computing*

1. はじめに

時間割作成問題は、スケジューリング問題の一種である。一般に時間割作成といっても小学校から大学まで幅広い範囲がある。小学校は一人の教員が一つのクラスの面倒を見るため、その教員のみスケジュールによって時間割を作成することが出来る。また中学、高校は基本的に各クラスが使用する教室が決まっており、教員のスケジュールのみが問題となる。それに対して、大学の時間割は教員のスケジュール、受講者数および教科にあわせた教室の確保、複数クラス合同授業、2時間連続授業など様々な要因があり、前者に比べてその作成は非常に困難になる。

時間割は従来手作業で作成されているが、これら各種制約条件により、その作成は通常2~3ヶ月かかる。そのため時間割作成の自動化が望まれており、企業、大学などでも研究がなされている。時間割作成問題の歴史は古く、計算機が一般に普及してきた1960年代から多くの研究がなされてきた²⁾、これらには Prolog による推論機構を用いた方法⁹⁾、制約緩和手法を用いた方法⁸⁾、対話的な方法^{7,10)}などの手法や、その他にも動的計画法、アニーリング法などがあるが、一般に確立されて標準になっている方法はほとんどない。それは制約条件が非常に膨大であり、また学校特有の条件の存在などから汎用化が困難であるためである。本研究では、この問題に対して遺伝的アルゴリズム (Genetic Algorithms, 以後GA) を適用し、その有効性を検証する。

GAは、生物の遺伝情報の伝達機能を取り入れた最適化アルゴリズムで、米国ミシガン大学の John H.

* 1 現在(株)東芝に勤務
平成10年10月19日受付

Holland によって1960年代に考案された。このアルゴリズムは解候補を遺伝子として表現し、それに対し選択、交叉、突然変異を順次適用することによって解を最適化する。このアルゴリズムは遺伝子を複数個用意することによって一種の並列探索を行うことが出来、また突然変異を取り入れることで局所解 (local optimum) に陥りにくいという特徴を持っている。またオペレーターが比較的シンプルであり、解候補を遺伝子に表現できればモデル化が困難な問題にも適用できるため、様々な分野でその応用が研究されている。

本システムでは、GAの適用を以下の方針で行った。

- (1) 時間割を授業が順番に並んでいると考え、順序問題として取り扱う。その際に順序型のGAを適用する。これにより「1週間の授業を過不足無く割り当てる」と「複数の教員が同一時間に同じクラスで授業を行わない」という条件を常に満たすことが出来る。また遺伝子表現には後述する複数遺伝子構造を採用し、各学年各クラスにまたがる制約条件のチェックを容易に行えるようにする。
- (2) 実験などの2時間連続して行われる授業には2時間授業再配置アルゴリズムを導入し、この条件を完全に満たすように時間割を修正する。
- (3) 教室の確保には教室確保アルゴリズムを別に用意して確保を行う。

以下にそれぞれについて説明するとともに、二種類のシミュレーションを実施し、その性能を検証したので、その結果を報告する。

2. 時間割作成における問題点

2.1 制約条件の考慮

時間割作成における問題点としては、以下のような点が挙げられる。

まず、第一に考慮すべき制約条件が膨大であることが挙げられる。同一時間に複数の授業を担当しないなどの基本的なものからその学校特有のものを含め、数多くの条件が存在する。そのため、全ての制約条件を完全に満たすことは不可能に近く、一種の多目的最適化問題であるといえる。

そこで本システムでは、時間割を構成するために絶対必要な制約条件の優先度を高く設定し、その他の妥協可能な条件については、その程度によって優先度を設定する。各条件にはその優先度に応じた重み係数を与え、評価を行う。本システムで使用した制約条件お

よび評価関数は後述する。

2.2 2時間連続授業への考慮

実験などは通常の講義とは異なり、2コマ分を連続して行うことが多い。ここではこれを2時間授業と呼ぶ。2時間授業は適切な時間(昼休みを間に挟まない、別の日にまたがらないなど)に配置しなければならない。そこで、遺伝子の構成方法として2時間授業を一つのコマ要素として構成する方法が考えられる。しかしこの方法では、2時間授業があるクラスとないクラスで部分遺伝子長の違いが出てくる。そのため、クラスにまたがる条件の評価が非常に困難になる。そこで今回は、2時間授業を1時間授業が二つあると定義し、評価前に遺伝子内の配置を変更して2時間授業を連続かつ適切な時間に再配置するアルゴリズムを採用した。このアルゴリズムを2時間授業再配置アルゴリズムと呼ぶこととする。

まずあらかじめ遺伝子内の各要素に2時間授業を示すフラグを与えておき、これにより2時間授業の識別を行い、以下に述べる手順で、2時間授業の再配置を行う。なお一日の授業時間を最大5時間とし、2時間授業が収まるべき適切な位置を1, 2時間目と3, 4時間目と仮定する。アルゴリズムを以下に示す。

Step 1: 遺伝子内の先頭要素から2時間授業のフラグをチェックしていき、2時間授業要素を探し出す。この要素を e_1 とする。

Step 2: e_1 に対応する要素を残りの要素から探し出す。この要素を e_2 とする。

Step 3: 要素の再配置を行う。ここで、 e_1 の位置は次の3つの場合が考えられるので、それぞれの場合によって処理を変える。

- (1) e_1 が奇数時間目(1, 3時間目)にある場合は e_2 を e_1 の一つ後ろの要素と交換する。(図1)

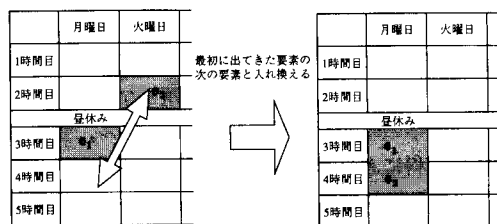


図1 2時間授業再配置アルゴリズム-1 (奇数時間目にある場合)

Fig.1 Relocation algorithms for double hour class-1

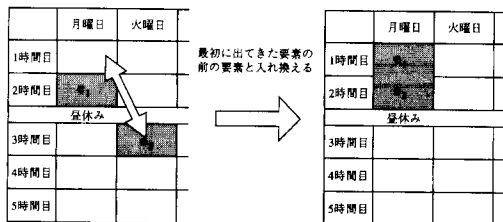


図2 2時間授業再配置アルゴリズム-2 (偶数時間目にある場合)

Fig. 2 Relocation algorithms for double hour class-2

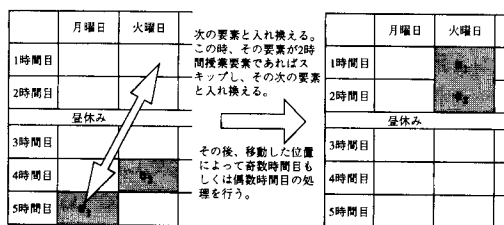


図3 2時間授業再配置アルゴリズム-3 (5時間目にある場合)

Fig. 3 Relocation algorithms for double hour class-3

(2) e_1 が偶数時間目(2, 4時間目)にある場合は e_2 を e_1 の一つ前の要素と交換する。(図2)

(3) e_1 が5時間目にある場合は e_1 と次にくる要素を交換する。この時、その要素が2時間授業要素であれば、スキップしてさらにその次の要素と入れ換える。そのあと e_1 の移動した位置によって e_2 に対して奇数時間目もしくは偶数時間目の処理を行う。(図3)

Step 4: 全ての2時間授業の再配置が終わってれば終了する。そうでなければ Step 1.に戻る。

このアルゴリズムによって、2時間授業の配置に関する条件を常に満たす時間割を作成できる。

2. 3 教室の確保についての考慮

時間割作成を行う上で教室の確保は重要である。しかし、遺伝子表現の中に教室の情報までも含むと、その探索空間は非常に広大なものとなる。そこで本システムでは別アルゴリズムを用意し、それをを用いて教室の確保を行う。このアルゴリズムを教室確保アルゴリズムと呼ぶ。

このアルゴリズムでは、まず教室テーブルを用意する(図1)。これは、全ての教室の一週間の使用状況を

表すテーブルである。教室テーブル内の要素は初期段階では全て空白である。割当の条件としては収容人数、教室種別があり、その条件を満たせば使用可能である。割当は以下の手順で行う。

Step 1. 遺伝子の要素を先頭から順に読み出す。この時、その要素が空白時間なら次の要素を読み出す。

Step 2. 読み出した要素を教室テーブルに割り当てる。この時、以下の順番で割当場所を決定する。

1. そのクラスが主に使用する教室が使用可能であればそこに割り当てる。
2. そのクラスが所属している建物内で使用可能な教室があればそこに割り当てる。
3. 学内全体で使用可能な教室があればそこに割り当てる。

Step 3. もしその要素を割り当てられなければ、割当を打ち切り、その要素をカウントしておく。

Step 4. 全ての要素の割当が終わっていれば終了する、そうでなければ Step 1.に戻る。

以上の手順の結果、確保できなかった授業数をカウントしておき評価に用いる。

表1 教室テーブル

Table 1 Classroom table

	月曜日1時間目	月曜日2時間目	月曜日3時間目
講義室121			
講義室122			
講義室123			
講義室124			

3. システム構成

本システムのフローチャートを図4に示す。各部のアルゴリズムの詳細は2節で述べた通りである。

今回、GAのワークベンチとして、コロラド州立大学のD.W. Whitleyらによって開発されたGenitorをベースに、本研究の内容にあわせて機能の追加、変更を行って使用した。

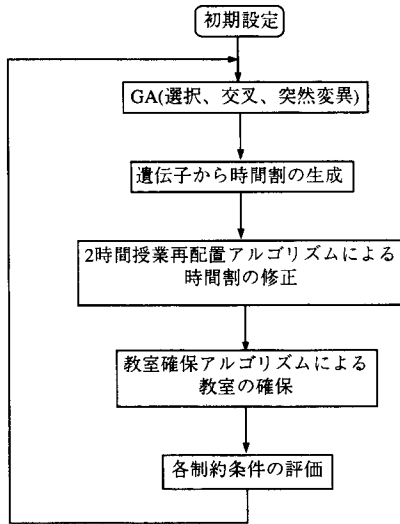


図4 本システムのフローチャート
Fig. 4 Flowchart of timetabling

4. GAの適用

4.1 遺伝子構造

今回採用した順序型GAにおいて、時間割を遺伝子に表現する方法として最も単純な方法は、全ての授業の順番を一次元に並べてコーディングする方法である(図5)。しかしこの方法では、遺伝子内の各学年組の区別が困難であり、また交叉、突然変異を適用する際に工夫が必要になる。また、各クラスにまたがる制約条件のチェックにも困難をきたすこととなる。そこで本研究では遺伝子を各学年各組ごとに分割し、図6に示すように複数遺伝子として授業番号の順序で表現する。ここで表現法には順序を直接表現するパス表現を用いる。また遺伝子長を同じにするために空白時間には空白を意味する番号を与える。

4.2 交叉

交叉の適用法を図7に示す。この図の示すように、

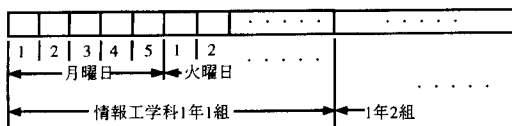


図5 単一遺伝子構造
Fig. 5 Simplex genes construction

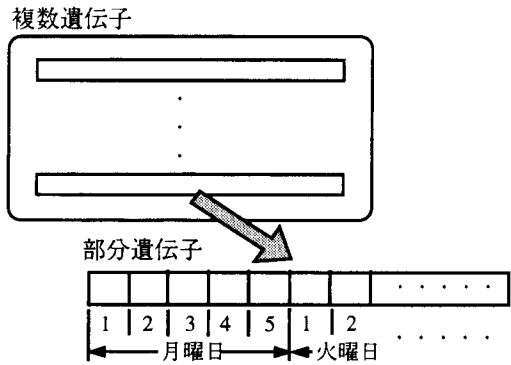


図6 複数遺伝子構造
Fig. 6 Multiple genes construction

二つの遺伝子の対応する部分遺伝子(同じ学年組)のみで交叉を行い、その他の部分は変更しない。また、交叉には順序表現用の交叉である循環交叉(CX法: Cyclic crossover)を使用する。

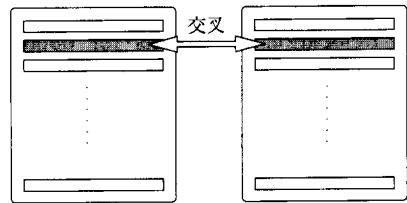


図7 交叉の適用法
Fig. 7 Crossover for multiple genes construction

4.3 突然変異

突然変異の適用法を図8に示す。まず遺伝子から対象とする部分遺伝子(学年組)を任意に選び出す。次に部分遺伝子内の任意の二つ要素の位置を交換する。この突然変異をここでは位置交換突然変異(Swap法)と呼ぶ。

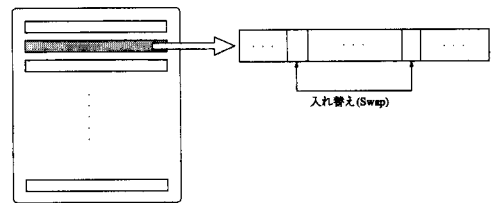


図8 突然変異の適用法
Fig. 8 Mutation for multiple genes construction

表2 制約条件
Table 2 List of restrictions

	制 約 条 件	優先度	解 決 法	評 価 値
1	1週間の授業を過不足無く割り当てる	最重要	複数遺伝子構造で考慮	—
2	同じ教員が同じ時間に複数の授業を持たない	最重要	評価関数	重なっている授業数
3	複数の教員が同じ時間に同じクラスを持たない	最重要	複数遺伝子構造で考慮	—
4	2時間連続授業の適切な時間への配置	最重要	2時間授業再配置アルゴリズム	—
5	教室の確保	最重要	教室確保アルゴリズムと評価関数	確保できなかった授業数
6	教員の研修日の確保	1	評価関数	確保できなかった授業数
7	なるべく5時間目の授業を行わない	2	評価関数	5時間目の授業数

4.4 制約条件

今回、システムに組み込んだ制約条件を表2に示す。この条件を取り入れた評価関数を定める。

制約条件は他にもあるがここでは以上の制約条件を対象とする。ここで1, 3の条件は今回採用した遺伝子構造では起こり得ない条件なので、評価関数に入れる必要はない。4の条件は前述の2時間授業再配置アルゴリズムにおいて考慮する。従って、この1, 3, 4を除いた制約条件を満足するよう評価関数に反映させる。

4.5 評価関数

評価関数は以下のように設定する。

$$Perf = \sum_i w_i x_i \tag{1}$$

ここで、 w_i は各制約条件の重み係数(優先度)、 x_i は各制約条件の評価値である。そして、この評価関数値が小さいほど優良な個体である。

重みは制約の度合いに応じて値を変える。すなわち制約の度合いが大きい条件は重みを大きくする。

5. シミュレーションおよび結果

本システムの性能を検証するために以下の二つの時間割の作成を行った。

各シミュレーションはそれぞれ以下のパラメータで行った。

- 個体数 : 300
 - 遺伝子長 : 300 (シミュレーション-1)
1300 (シミュレーション-2)
 - 交叉率 : 1.0
 - 突然変異率: 0.3
- シミュレーション1

まず本学の情報工学科を例としてシミュレーションを行った。各学年3クラスの全学年12クラスを対象に、一週間の授業数を25時間、授業数を53授業、教員数を専門教養を含み直接関係のある56人、教室数を34室とした。結果を表3に示す。

この表が示すように、全ての制約条件を完全に満たすことができた。この時の最良値の推移を図9に示す。

この図が示すように、良好な収束を見せた、また計算時間も約10分と十分に実用可能な時間で解を得ることができた。このシミュレーションで、教室の確保に関しては初期世代の段階からすでに条件を完全に満

表3 シミュレーション結果-1
Table 3 Results of simulation-1

制 約 条 件	満たせなかった条件数	
	初期状態	最終状態
同じ教員が同じ時間に複数の授業を持たない	3	0
教室の確保	0	0
教員の研修日の確保	38	0
なるべく5時間目の授業を行わない	35	0

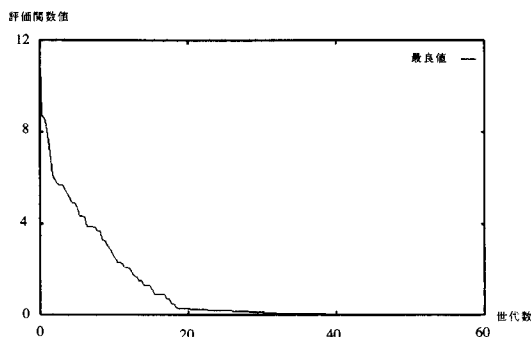


図9 最良値の推移 (シミュレーション-1)
Fig.9 Characteristics of convergence (Simulation-1)

たしており、教室の確保を別アルゴリズム化した効果が現れている。

シミュレーション2

つぎにシミュレーションを全学部に拡張して行った。2学部8学科全52クラスを対象に授業数を401授業、教員数を196人、教室数を99室にした。一週間の授業数はシミュレーション1と同じく25時間である。結果を表4に示す。

この表が示すように、優先度が高い条件を満たすことができたが、優先度を最も低く設定していた「なるべく5時間目の授業を行わない」という条件を完全に満たすことができなかった。しかしながら、この条件は妥協可能な条件であり、時間割としては実施可能であるので、満足のいく結果であると言える。この時の最良値の推移を図10に示す。

このグラフが示すようにシミュレーション2の場合でも、良好な収束を見せた。ただし、このシミュレーション2では遺伝子長が長いこともあり計算時間が約3時間かかった。この点についてはアルゴリズムの改善、高速機の適用などを考えていきたい。教室の確保に関しては、このシミュレーションでもその効果が現れており別アルゴリズム化による性能の向上を確認で

表4 シミュレーション結果-2
Table 4 Results of simulation-2

制 約 条 件	満たせなかった条件数	
	初期状態	最終状態
同じ教員が同じ時間に複数の授業を持たない	49	0
教室の確保	0	0
教員の研修日の確保	157	0
なるべく5時間目の授業を行わない	151	38

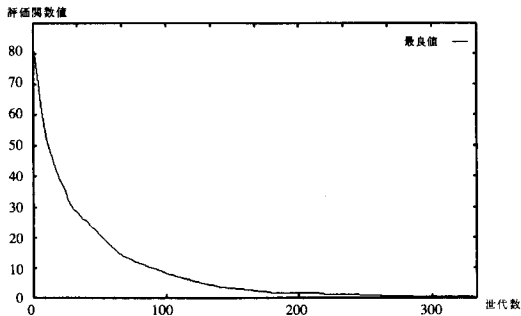


図10 最良値の推移 (シミュレーション-2)
Fig. 10 Characteristics of convergence (Simulation-2)

きた。

6. おわりに

本稿では、大学の時間割作成にGAを適用し、その際に一部の制約条件を別アルゴリズムによって満足させることによって、大幅に性能を向上させることができた。また本システムによる、時間割作成の自動化の可能性を検証することができた。

しかし規模が大きくなると相乗的に計算時間が増大するという問題があり、今後はアルゴリズムの改善などの対応策を考えていくとともに、制約条件の追加などを行っていく予定である。

参 考 文 献

- 1) D.E. Goldberg: Genetic Algorithms in Search, Optimization, and Machine Learning, Addison-Wesley Pub. Co., 412p (1989).
- 2) Edmund Burke and Peter Ross (Ed): Practice and theory of automated timetabling: First International Conference, Springer Verlag, Co., 379p (1996).
- 3) 北野宏明: 遺伝的アルゴリズム, 産業図書, 328p (1983).
- 4) 北野宏明: 遺伝的アルゴリズム 2, 産業図書, 335 p (1995).
- 5) 稲見泰生: 学校時間割作成プログラムについて, 茨城大学教育学部紀要 42号, pp.1-6 (1993).
- 6) Abid M. Elabdalla and Riyad A. Husein: An Optimal Examination Scheduler, Computer Educ. Vol. 17, No. 4, pp.267-269 (1991).
- 7) 安部佳代子, 山崎明治: 半自動対話型時間割編成支援システム, 福岡工業大学研究論集 Vol.29, No. 2, pp.351-362 (1997).
- 8) 杉田純也, 遠藤哲朗: GAを利用した時間割作成法, 電子情報通信学会技術研究報告 No.93, pp.17-24 (1994).
- 9) 青木征男, 佐塚秀人, 猪飼秀隆, 江草祐右, 加藤和則: 計算機による時間割作成の試み, 久留米工業大学知能工学研究所報告 第4号, pp.69-75 (1991).
- 10) N. Chahal and D. de Werra: An interactive system for constructing timetables on a PC, European Journal of Operational Research 40,

- pp.32-37 (1989).
- 11) 金子一哉, 吉川昌澄, 山之内徹: 大学時間割編成問題への制約緩和手法の適用と評価, 情報処理学会第55回全国大会予稿集, pp. 2 -578-579 (1997).
 - 12) 土性雅史, 小野俊彦: 遺伝的アルゴリズムの時間割作成問題への適用, 情報処理学会第55回全国大会予稿集, pp. 2 -473-474 (1997).
 - 13) 小野俊彦, 池田武史, 土性雅史, 田代順一: 遺伝的アルゴリズムによる最適問題の解法, 福岡工業大学言語情報工学研究所彙報 第 8 巻, pp.19-29 (1997).
 - 14) 土性雅史, 小野俊彦: 大学時間割作成への遺伝的アルゴリズムの適用, 人工生命とその応用シンポジウム論文集, pp.49-55 (1997).