数理モデル化と問題解決　18－3
（１９９８．３．２０）

可変パラメータを持つ共存型遺伝的アルゴリズムの時間割作成への適用

金盛 友孝　松居 辰則　岡本 敏雄

電気通信大学大学院 情報システム学研究科

学校では毎年、教師の数や教室の数を考慮しながら、カリキュラムで定められた各教科を１週間で行なうための時間割を作成している。時間割には様々な制約があるため、これらを満たしながら時間割を作成することは、教師に負荷を課すことになる。本研究では、時間割を作成する際に個人と全体に分けて評価を行うことで共存型 GA を実現する。また、全体の時間割における各教師個人の時間割の影響度合から、次世代の個体を生成する際のパラメータを可変にすること、各個体群に優先度をつけること、枯死遺伝子の概念を取り入れることで、従来の SimpleGA と比較して解の収束性が向上することを期待する。

TimeTable Generation System Using Cooperative Genetic Algorithm With Variable Parameter

Tomotaka KANAMORI　Tatsunori MATSUI　Toshio OKAMOTO

Graduate School of Information Systems,
The University of Electro–Communications

The timetable generation problem is said to be very difficult because of various constraints about teachers' schedule, classroom capacity and school curriculum and so on. In this paper, we propose the solving system of the timetable generation problem using the cooperative genetic-algorithm with variable parameter. The main theoretical features of this system are coding method of gene which represents teacher's one week schedule, using new type gene called Dead-Gene and relative merits between teachers.

1.　はじめに

今日、コンピュータは、解法のアルゴリズムが存在し、明確にそれが記述可能な問題を解く能力は人間を遥かに上回る。一方、解くべき問題がアルゴリズムとして記述困難な世界における情報処理能力においては柔軟ではない。人工知能の分野においてもシステム・知能化の対象がより高水準な経験や知識の抽出を必要とする分野へ適用されるようになると、明確に解法のアルゴリズムや知識を記述する古典的な数理手法やエキスパートシステムなどの適用が必ずしも有効でないとの指摘もなされている [1]。

遺伝的アルゴリズム (Genetic Algorithm:以下、GA と略す) は、こうした従来の手法の抱える問題点に対し、近年注目されている手法の１つである。GA は、その汎用性の高さからさまざまな分野での適用がはかられているが、研究分野として未発達なため、個々の問題への適用の試行錯誤により成果を蓄積しているのが現状である。本研究では、この GA をスケジューリング分野の１つである時間割作成に適用する。

2.　研究の目的

ＧＡで時間割のようなスケジューリング問題を解くという研究は行なわれているが、それらには制約が多くなると一般解が求められない問題点がある [2]。本研究では、時間割を作成する際に個人と全体に分けて評価を行うことで、共存型 GA を実現す

る。また、全体の時間割における各教師個人の時間割の影響度合から、次世代の個体を生成する際のパラメータを可変にすること、各個体群に優先度をつけること、枯死遺伝子の概念を取り入れることで、従来の SimpleGA と比較して解の収束性が向上することを期待する。

3. 時間割作成問題

時間割は、あらかじめ決められた各授業を、クラス、教師、教室などを考慮しながら、より多くの制約を満たすように1週間のうちの一つの時間に割りつける制約充足問題である。制約としては、絶対条件、制約緩和条件といった制約条件と、許可に関する付加条件がある (表1)。

表 1: 制約条件

制約の分類	制約の内容
絶対条件	●カリキュラムで定められた時間数 ●同じ時間に同じクラスを複数の教師が受け持つことの禁止
制約緩和条件	●1日および1週間の教師の時間数の制限 ●同じ日に同じ教科を3時間以上行なわない ●教科内容が曜日によって偏らない
付加条件	○2時間連続の教科 ○複数クラス合同の教科

●…禁止事項　○…許可事項

4. 時間割作成問題に対する遺伝的アルゴリズムの適用

以下、本研究で用いる時間割作成問題におけるGAのコード化と時間割の評価の方法、適用した遺伝的操作の方法について述べる。

4.1 コード化

学校において時間割は2種類の形で表現されている。一つは、各クラスで使用している各クラスごとの1週間の時間割。もう一つは、職員室で使われる各教師ごとの1週間の時間割である。前者の形式では、個体中の一つの遺伝子座は1時間の授業科目になる。この場合、どの教師によるどの教科の授業かという2値のデータを考慮したコード化を考えなければならない。後者の形式では、個体中の一つの遺伝子座は1時間のその教師が担当するクラスになるため、クラス情報のみで表現することが可能である。よって本研究では、後者の形式でコード化する (図1)。

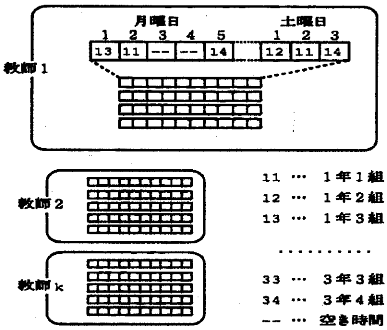

図 1: コード化のイメージ

当然、教師ごとに時間割を作成した場合、教師間で同じ時間に同じクラスを教えてしまうといった教師のブッキングが起こることが考えられる。よって、教師ごとに時間割を作成したら次に教師間での整合性を取る必要が生じる。これに関しては、4.5 で述べる。

4.2 評価関数と適応度

本節では、本手法における各教師個体の評価法に関して述べる。

4.2.1 各教師の時間割と評価関数の設定

表1であげた制約条件を考慮して、各教師の個体を評価する際に、各教師に共通した制約と担当教科によっては許可に関する制約により評価を行なう。これを基に適応度を算出する。

(a) 同じ日に同じクラスを3時間以上担当しないという制約に関する評価

1日の授業で同じ教科が3時間以上行なわれることは悪いわけではないが好ましくないと考えられる。よって各教師個体で曜日ごとに同じクラスを担当している数をチェックし、3時間以上担当してい

る場合には、その数を違反点として個体に与える。すなわち、式 (1) のようになる。

$$Frequecy(c_t) = \sum_{i=1}^{T}\sum_{j=1}^{week}(\sum_{k=1}^{day} t_{ijk} - limit) \qquad (1)$$

T: 担当クラス数

$week$: 曜日の数 (月曜日から土曜日まで)

day: 1日の時間数

$$t_{ijk} = \begin{cases} 1 & \text{クラス i を担当する} \\ 0 & \text{担当しない} \end{cases}$$

$limit$: 1日の担当時間の上限

(b) 2時間連続の教科に関する評価

美術や技術家庭科などは専門性が高いためにしばしば2時間連続で授業が行なわれることがある。このような許可が出された場合に、2時間連続でないクラスが出現したら、出現クラスの数だけ違反を与える。すなわち、式 (2) のようになる。

$$Continuity(c_t) = \sum_{i=1}^{T} i \qquad (2)$$

T: 担当クラスの数

$$i = \begin{cases} 0 & \text{連続である} \\ 1 & \text{連続でない} \end{cases}$$

4.2.2 適応度

4.2.1 で述べた2つの違反点がそれぞれ少ないほど、その教師にとって良い時間割となる。そこで式 (3) に示すように、評価値 $Frequency$ と担当教科によっては、$Continuity$ の和に着目し、その値 $Fitness$ がより小さい時間割を生成する。

$$Fitness(c_t) = Frequency(c_t) + Continuity(c_t) \qquad (3)$$

$Continuity(c_t)$ は、2時間連続教科のみ

4.3 初期集団の形成

一般に初期集団は決められた個体数 M 個だけ個体をランダムに生成する (initialization)。

本手法では、時間割という性質から、ただランダムに集団を生成するのではなく、担当するクラスを受け持つ時間数は、カリキュラムで定められた時間数と同じにするという条件で初期集団を生成する。

今後の交叉・突然変異等の操作を行う際にもこのカリキュラムで定められた時間数を守るという制約は考慮する。また個体を生成する際に、あらかじめ教師の都合が悪い時間というものを与えておく。この詳細は、4.3.1 で述べる。

4.3.1 教師の都合を考慮した枯死遺伝子

非常勤教師などが存在する場合などには、少なからず都合の悪い時間というものが存在する。そういったことを考慮するためにあらかじめ教師に都合の悪い時間を入力してもらい、その情報を個体中に枯死遺伝子として与えておく。

枯死遺伝子とは、個体中の1つもしくは複数の遺伝子座に存在し、あらゆる遺伝的操作の影響も受けない遺伝子のことである。よって、枯死遺伝子として情報が与えられた時間は、常に空き時間という情報が入ることになる。

4.3.2 担当教科による個体の重要度

各教師ごとに時間割を作成した後、教師間の整合性を保つために、教師のブッキングチェック (4.5) を行うが、その際に担当教科によって優先度をつけておく。

教科を特別教科と一般教科に分け、特別教科担当教師の個体は強個体、一般教科担当教師の個体は弱個体とする。4.5 節で述べる教師間の競合が生じた際には、特別教科担当教師である強個体の時間割を一般教科担当教師の時間割よりも優先的に扱う(表2)。

表 2: 担当教師による優先度

教師区分	特別教科担当教師	一般教科担当教師
該当教科	音・体・美・技	国・数・理・社・英
個体の種類	強個体	弱個体
競合の際	優先度が高い	強個体の影響を受ける

4.4 遺伝的操作

以下では、時間割作成という点から本手法で提案する遺伝的操作について述べる。

4.4.1 選択 (Selection)

選択には、ランク方式とエリート保存戦略を用いる。ランク方式は、評価関数が自由に設定でき、適応度のばらつきが小さい集団においても個体間の優劣を反映させた選択方式を実現する。そのため、ルーレット方式よりも早熟収束 (premature convergence) の問題が起きにくい。エリート保存戦略は、集団の

局所解への収束の危険や、良質の個体が集団から失われる危険を回避するために用いる。

4.4.2 交叉 (Crossover)

時間割作成において、SimpleGA で行われている交叉をそのまま適用すると、カリキュラムで定められた時間数を守るという絶対条件に違反し致死遺伝子[1]になってしまう。本研究では、致死遺伝子を生成させないために、松本ら [3] が提案した交叉手法を改良して適用することにする。以下に本研究の交叉に関してその操作を記述する。

1. 乱数により、親となる個体 $parentA$ の a 番目の時間割 T_a と、交叉をする長さ $length_a$ を決める。$n \leftarrow 0$ とする。これで求めた列を Δ_a とする。

2. 同様に、もう一方の親となる個体 $parentB$ にもこの操作を行い、Δ_b を取り出す。

3. $parentA$ から得られた Δ_a を $parentB$ に、$parentB$ から得られた Δ_b を $parentA$ に継承する。その際、Δ_a と最も類似している $parentB$ の部分 Δ' に Δ_a を挿入する。ただし、Δ_a と Δ' の両方ともに含まれている時間割の数が多いほど、類似しているとする。Δ_b を $parentA$ に挿入する場合も同様である。

図 2 に一連の操作を示す。

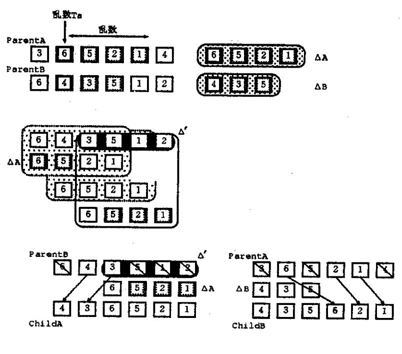

図 2: 本手法の交叉法

[1] 明らかに解になり得ない遺伝子

4.4.3 突然変異 (Mutation)

本研究においては、SimpleGA のようにランダムに突然変異を起こすと、その個体は致死遺伝子になってしまう。本研究では、個体中の無作為に抽出した 2 つの遺伝子座を交換するという手法で致死遺伝子が生成されない突然変異を行う。

4.5 教師間の整合性

教師ごとの時間割を集めて来て、全体の時間割を作成しようとする際に、おそらく教師間で授業のブッキングが起こり得る。それを解消する手法を 4.5.1 で、また次世代の個体を生成する際にその影響を各教師個体に反映するような遺伝的オペレータの変更方法を 4.5.2 で述べる。

4.5.1 ブッキングチェック

教師間での不整合を数値化するために、本研究では、他教師への不整合度というものを定義する。

教師 i の他教師への不整合度 $E_i(\mathbf{t})$: t 世代の全体の時間割を作成する際に、仮に教師 i がいなかった場合にどれくらい時間割が作りやすくなるかというのもを表す度合

この不整合度 $E_i(t)$ は、式 (4) で定義する。

$$E_i(t) = \frac{E(t) - E_{-i}(t) \times \frac{n^2}{(n-1)^2}}{E(t)} \quad (4)$$

$E_i(t)$: 全体の整合度

$E_{-i}(t)$: 教師 i を除いた整合度

式 (4) は、以下のように算出される。

不整合度の算出

全体の時間割は、式 (5) のように表現される。

$$T(t) = \begin{pmatrix} x_{11} & x_{12} & \dots & x_{1m} \\ x_{21} & x_{22} & \dots & x_{2m} \\ \dots & \dots & \dots & \dots \\ x_{n1} & x_{n2} & \dots & x_{nm} \end{pmatrix} = (x_{ij}) \quad (5)$$

x_{ij} は、教師 i の j 番目の時間割を示す

転置行列 $T^t(t)$ を作り、次のように $OT(t)$ とする。

$$OT(t) = T(t) \cdot T^t(t) = y(ij), 1 \leq i, j \leq n \quad (6)$$

$$T(t) \cdot T^t(t) = x_{ik} \cdot x_{kj} = \begin{cases} 1 & x_{ik} = x_{kj} \\ 0 & x_{ik} \neq x_{kj} \end{cases} \quad (7)$$

式 (6), 式 (7) から、$y(ij)$ は、教師 i の教師 j に対するブッキング数であるといえる。

以上より、t 世代における全体の整合性 E(t) は、以下のよう定義する。

$$E(t) = 1 - \frac{\sum_{j=1}^{m}\sum_{i=1}^{n} y_{ij} - \sum_{i=1}^{n} y_{ii}}{mn^2} \tag{8}$$

T(t) から i 行 (教師 i) を除去したものを $T_{-i}(t)$ とすると、同様に、

$$OT_{-i}(t) = y(ij), 1 \leq i, j \leq n-1 \tag{9}$$

となり、t 世代における教師 i を除去した全体の整合性 $E_{-i}(t)$ は、以下のよう定義する。

$$E_{-i}(t) = 1 - \frac{\sum_{j=1}^{m-1}\sum_{i=1}^{n-1} y_{ij} - \sum_{i=1}^{n-1} y_{ii}}{m(n-1)^2} \tag{10}$$

4.5.2 交叉率・突然変異率の変更

4.5.1 で述べた不整合度が大きいほど、全体へ悪影響を及ぼしている教師個体であるといえるので、そのような教師個体は大幅な個体の変更をする必要がある。また小さければ、特に個体を変える必要ない。これらのことから、次世代の教師個体を生成する際に、不整合度を考慮した遺伝的操作を行う。

方法は、4.5.1 で求めた不整合度 $E_i(t)$ を用いて、次世代の交叉率 (C(t+1))・突然変異率 (M(t+1)) を変更する。

t 世代の教師 i の不整合度を考慮した t+1 世代の教師 i の交叉率・突然変異率はそれぞれ以下のように更新される。

$$C_i(t+1) = E_i(t)(\beta - \alpha) + \alpha \tag{11}$$

$$M_i(t+1) = E_i(t)(\omega - \gamma) + \gamma \tag{12}$$

5. システム構成

システム構成を図 3 に示す。

まず、ユーザは、学校の情報である、学年ごとのクラスの数、各曜日に行う授業の時間数と、教師情報である担当する教科、担当するクラス、都合の悪い時間を入力する。

次に教師個人時間割作成モジュールにより、各教師の 1 週間の時間割を作成する。その後、各教師の

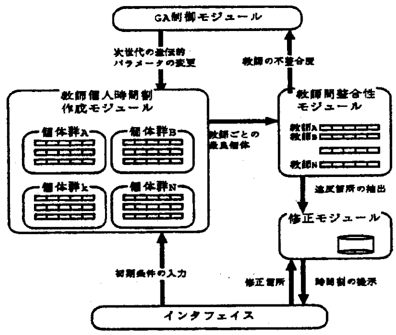

図 3: システム構成図

最良個体が教師間整合性モジュールで整合性チェックを受け、その結果が次世代の遺伝的オペレータのパラメータ値に影響を与える。この操作を繰り返し、時間割を作成していく。解が収束したら、違反がある場合にはユーザに違反箇所を含めて提示し、ユーザとの対話により最適解を生成する。

6. 評価

全体の整合度の推移を図 4 に示す。この図から、世代が進行するにつれて整合性の度合が高くなっており、解が次第に収束していることがわかる。図 5 は、

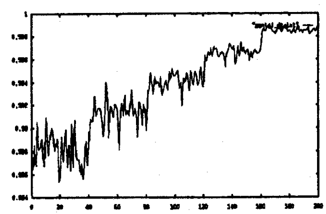

図 4: 全体の整合度の推移

特別教科担当教師と一般教科担当教師の交叉率の推移をグラフにしたものである。実線が一般教科担当教師、点線が特別教科担当教師を表しているが、明らかに特別教科担当教師の方が低い値を示している。これは、4.3.2 で述べたように、特別教科担当教師が競合の際に優先度が高く、一般教科担当教師はその影響を受けるということを実証している。枯死遺伝子が存在する場合にも、同様の結果が得られた。

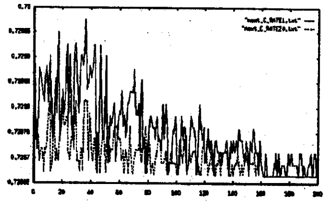

図 5: 担当教師による交叉率の推移

これらの結果はスキーマ定理を用いて説明することができる。

スキーマ定理: 染色体中の意味のあるパターンがどの程度の割合で次世代に生き残るかを示す定理 (式 (13))。

$$m(H,t{+}1) \geq m(H,t)\frac{f(H,t)}{\overline{f}(t)}\left(1-\frac{p_c\delta(H)}{l-1}-p_m o(H)\right) \tag{13}$$

$m(H,t)$: 世代 t におけるスキーマ H に属する個体数
$f(H,t)$: 世代 t におけるスキーマ H の評価値
$\overline{f}(t)$: 個体群中の全個体の平均適合度
p_c, p_m : 交叉率・突然変異率
$o(H)$: スキーマ H の order
$\delta(H)$: スキーマ H の定義長
l: コードの長さ

本研究で提案したそれぞれの手法の収束性をスキーマ定理を用いて説明する。

(a) 一般教科担当教師と特別教科担当教師の収束の違い

一般教科担当教師の交叉率、突然変異率をそれぞれ p_{gc}, p_{gm} とし、特別教科担当教師の交叉率、突然変異率をそれぞれ p_{sc}, p_{sm} とすると、

$$p_{gc} \geq p_{sc}, p_{gm} \geq p_{sm}$$

であるから、式 (13) の右辺は特別教科担当教師の方が大きくなる。よって、次世代にスキーマ H に属する個体が残る割合は、特別教科担当教師の方が高くなり、収束性が高いことが言える。

(b) 枯死遺伝子の有無による違い

枯死遺伝子が存在する個体の場合は、遺伝的操作の影響を受けない遺伝子が存在するため、通常の個体よりも $\delta(H)$, o(H) は、短い。よって、式 (13) の右辺は枯死遺伝子を持つ個体の方が大きくなるので、収束しやすい。

(c) 交叉率、突然変異率が世代ごとに変化

世代が進むにつれて、教師の全体への影響度 $E_i(t)$ は小さくなり、次世代の各教師個体を生成するための交叉率 $C_i(t+1)$・突然変異率 $M_i(t+1)$ は小さくなる。よって、13 式の右辺は世代が進行するにつれて大きくなる。つまり、次世代にスキーマ H に属する個体が残る割合の期待値は、世代が進行するごとに上がるため、収束しやすいことがいえる。

以上のことから、本研究の遺伝的操作は従来の SimpleGA に比べ、収束性が高いといえる。

7. まとめ

本稿では、解の収束性を向上する手法として、可変パラメータを用いること、個体に優先度をつけること、枯死遺伝子の概念を取り入れることを提案した。また、時間割を作成する際に、教師個人と学校全体の時間割を分離して作成することで解を 2 方面から評価することを提案し、実際に時間割作成問題に適用、その効果を検証した。

今回の実験では、この手法によって、若干ではあるが従来の SimpleGA での解法よりも解の収束性の向上が見込まれることがわかった。このことにより、本研究の手法が効率の良い時間割作成に適しているということができる。しかしながら、解の収束段階で残ってしまった制約違反を修正・解消するためのユーザとの対話機構が不十分なため、システムとしてまだ改良の余地が残されているという問題点も残った。

また、本研究で提案した手法は、時間割作成問題だけでなく、一般的なスケジューリング問題のような、多重制約性を持ち合わせた問題への適用も可能であると考えられる。

参考文献

[1] 小林 重信: 計画問題と人工知能：概観, 人工知能学会誌 Vol.7 No.6 pp.965-969 (1989).

[2] 南雲 行夫, 江尻 博尚, 高山 文雄, 川合 英俊: 大学の時間割における人工生命的手法の適用, 情報処理学会第 5 4 回 (平成 9 年前期) 全国大会論文誌, pp.319-320 (1997).

[3] 松本 剛, 岡本 敏雄: 遺伝的アルゴリズムによるユーザの意図を取り入れた教授項目系列の生成, 電子情報通信学会技術報告, ET96-110, pp.45-52 (1997).