島根大学教育学部紀要（自然科学）第36巻　71頁～77頁　平成14年12月

遺伝的アルゴリズムと焼きなまし法による学校時間割作成について

福島　誠*

Makoto FUKUSHIMA*

A Class-Teacher Timetabling using Genetic Algorithm and Simulated Annealing

あらまし

本論文では，遺伝的アルゴリズム（GA: Genetic Algorithm）と焼きなまし法（SA: Simulated Annealing）を，学校時間割アルゴリズムとして中学校時間割作成に適用し，その結果について比較検討している．GAを使用した時間割アルゴリズムは，新規に提案する手法を含めて4種類の手法，SAについては2種類の手法を採用し，それぞれアルゴリズムのオペーレータ，及びパラメータを変化させて実験している．そして，得られた結果をコスト値として評価し比較検討を行った結果，GAによる改良されたアルゴリズムの効果は認められたが，SA法を採用した場合と比較するとやや劣る結果が得られることが分かった．

ABSTRACT

This paper presents an experimental result on automated class-teacher timetablings using Genetic Algorithm (GA) and Simulated Annealing (SA). The experiments are performed by using four types of GA including a currently enhanced one and two types of SA with varying their operators, such as crossover methods in GA and permutation methods in SA. From the results, the enhanced GA using a preferential assignment of hardly constrained tuples with the tournament of building blocks is effectively improves the reproduced offsprings in a cost measure, compared to other GAs under the assumed timetabling conditions. It is, however, also found that the best result of the GA is still slightly inferior to those of the SA in reducing the timetabling costs.

［キーワード：ビルディングブロック，局所適応度，コスト，交叉, 事前割当］

［**Key words**: building block, local fitness , cost, crossover, pre-assignment］

Ⅰ．まえがき

遺伝的アルゴリズム（GA）を使用した時間割作成アルゴリズムはいくつか報告されているが[1]-[3]，単純なGAの手法では実行可能な時間割を得ることは困難であることが分かっている．そのため，これまでにも，いくつかの修正されたGA法が時間割編成のアルゴリズムとして提案されている．修正は，repair，filteringといった操作（operator）の追加や，local fitness，Building Block（以下BB）を使用した交叉（crossover）といった，交叉方法の改良などがある．また，筆者らも1日単位の時間割をBBとして使用した交叉方法を提案している[4]．しかし，これらの改良を加えても，複雑な時間割問題に対してはなかなか完全な時間割を得ることは困難であった．またこれらの修正，改良について，どのように組み合わせれば効果的なのかもよくわかっていな

*島根大学教育学部技術教育研究室

表１　時間割データ
Table 1 Timetabling data

Number of teachers	30 (3x10)
Number of subjects for each class	10
Number of grades	3
Number of classes	15 (3x5)
Number of lessons per week for a subject	3
Number of periods per day	6
Number of weekdays	5

い．これは同一条件下で，これらの手法が比較評価された報告が少ないためである．また，一方で焼きなまし法（SA）も時間割編成アルゴリズムとしてAbramson[5]によって提案されており，時間割問題によっては比較的良い結果が得られている．筆者らの実験[6]でも，このSA法を使用した結果をGAによる結果と比較検討しているが，そこではGAよりも良い結果が得られている．本論文ではGA法について１種類の新規の進化手法を追加提案し，GAとSAの実験結果の比較と評価についてさらに検討する．

Ⅱ．時間割編成

２．１　クラス－教師時間割モデル

アルゴリズム評価のための時間割問題は，文献[6]で採用した拘束性の強い時間割問題を再度使用する．その時間割データを表１に示す．表１のデータについては時間割作成の条件を簡単化するために，以下のような条件を仮定している．1）全ての教師は学年ごとに1科目の授業を担当する．2）1学年は5クラスである．3）全科目数を10として各クラスに対して１科目につき週３回の授業を行い，合計週30時間（６時間/日，５日/週）の授業が行われる．各アルゴリズムを使用した時間割作成にはコストの概念を導入し，以下のような条件を満足する時間割に到達するまでに取り除かなければならないコスト値を計算する．コスト計算のための時間割の条件は，その必要度に応じて以下のように１次と２次の２種類に分類できる．

１次条件：
(1) 同一時限内の教師の授業数は１．
(2) 同一時限内の各クラスの授業数は１．

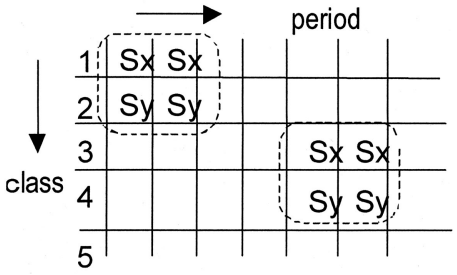

図１　同時展開と連続授業の概念．授業SxとSyは同時展開で，それぞれ連続授業である．
Fig. 1 Multiple and grouped lessons for subjects **Sx** and **Sy**, where **Sx** and **Sy** are combined as a grouped lesson and each of them is held as a multiple lesson.

(3) １日の教師の授業数を４時間以下に制限する．
(4) 同一科目の授業は連続授業を除いて各クラスに対して１日１時限以内とする．

２次条件：
(5) 教師の指定できる自由時間を１週間に３時間以上確保する．
(6) ２科目については連続授業を各学年に設定し，それぞれの科目ごとに特殊教室を使用する（図１参照）．
(7) 上記の連続授業は隣り合う２クラスの同時展開授業とする．但し，１学年のクラス数が奇数の場合は最後のクラスは同時展開の対象にはしない（図１参照）．

上記のような時間割編成条件について，それぞれコストを設定し，各条件に違反した時間割にはコスト値として１を加えることにする．但し，２次条件（6）と（7）については以下のようなコスト計算を適用した．

コスト６=（割り当ての必要な連続授業数）－（割り当てに成功した連続授業数）
コスト７=（割り当ての必要な同時展開授業数）－（割り当てに成功した同時展開授業数）

時間割アルゴリズムによって作成された各時間割のコスト値は，条件（1）～（7）に関連したコスト値の合計で与えられる．従って，時間割の完成とはその合計コスト値がゼロになった場合である．

２．２　時間割編成のためのGA

時間割編成アルゴリズムの評価用として使用するGAは，GA1～GA4の４種類である．これらのGAに共通して，以下のような条件を仮定した．１）tupleを１個の

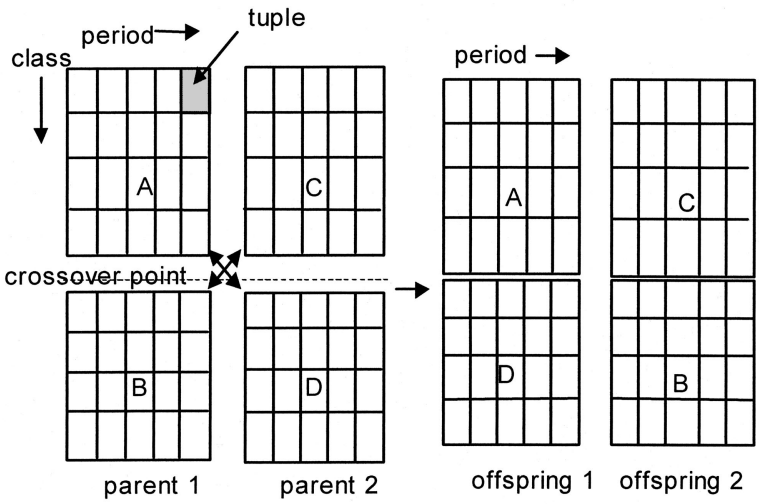

図２　Si-Eng による交叉（GA1）
Fig. 2 Crossover proposed by Si-Eng for GA1.

遺伝子とする（tupleは教師名，授業クラス名等の情報を属性として持つコマのこと），2）１時間割を１個体とする．３）GAの選択（selection）はルーレット方式で行う．この場合，各個体の適応度Fは，

F＝集団中の個体の最大コスト値－各個体のコスト

とする．

GA1：Si-Eng[2]の方法により，クラス時間割の各クラスにそのクラス属性に従ってtupleを配置する（図２参照）．従って，クラスの衝突（clash）はないで，コスト２の値は常にゼロである．但し，特殊教室を使用する場合の授業の衝突は存在する．GA操作における交叉は図２に示すように同一のクラスを含む時間割の部分で行う．同一教師の授業数の超過または減少を修正するために，交叉後はfiltering[2]を行う．連続・同時展開授業の事前割当（pre-assignment）は採用しない．

GA2：Colorni[3]らの方法に従って，時間割は教師時間割とする．tupleはその教師属性に従って配置するので，教師の衝突はなく，コスト１はゼロである．２つの時間割をランダムに選択してそれぞれ「親(parent)1」,「親２」として交叉の対象とする．「親１」の時間割を教師ごとにlocal fitness[3]を計算して良い順に並べ，local fitnessの基準値を基に２つのBBに分割する（図３参照）．これにより「親１」はAとBのBB,「親２」はCとDのBBに分割される．ここで，CはAに含まれていない教師の時間割より構成されているとする．Local fitnessの値は，各教師の１週間の時間割に，上記の時間割条件（3）から（7）を適用して計算できるコストの合計値とする．これらのBBは交叉により子孫（offspring）に受け継がれ，交叉時の１日単位及び１週間単位のtupleの突然変異と，交叉後のfiltering も行う．但し，local search[3]は採用しない．

GA3：Abramson とAbera[1]，及びGA2のColorniらの手法を基にして，筆者らが既に提案したBBのトーナメントとlocal fitnessを導入した手法である．時間割はクラス時間割で，BBは１週間単位ではなく，図４に示すように１日単位である．従って，クラス時間割は５個のBBを含んでいる．交叉は，２つの時間割の同じ日のBBのlocal fitnessを比較し，fitnessの高い（コスト値の低い）適しているBBを子孫の２つの時間割にコピーして残す．但し，子孫の時間割が，親集団の最大コスト値よりも大きくなっても，その子孫は廃棄（discard）しない．

GA4：今回新たに提案するGA3を修正した手法である．初期世代の交叉では2. 1の時間割条件の（6），(7)の条件のみをGA3により改善し，合計コストを一定の値（ここでは５以下）にしておく．そして，その後の世代では全てのコストについて同様にGA3を適用する方法である．即ち，困難と思われる条件を優先的に処理する方法で，事前割当の変形手法といえる．

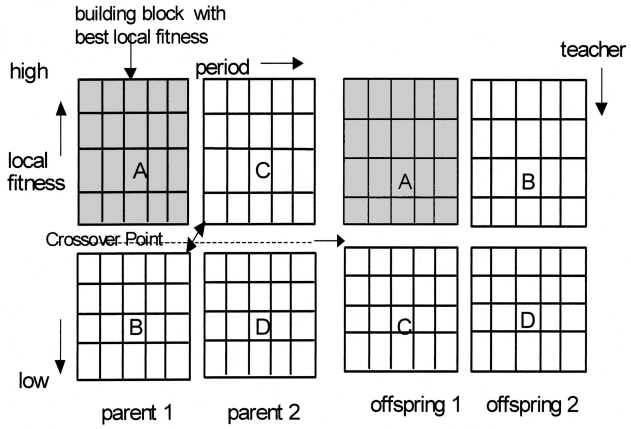

図３　Colorni らによるGA2の交叉．親1の教師時間割の行方向（period方向）はlocal fitnessの値に従って上から順に並び替えている．ここで，親2の時間割のブロックCは，親1のブロックAには含まれない教師の時間割，ブロックAは最良のlocal fitnessを有する教師時間割でそれぞれ構成されている．

Fig. 3 Crossover proposed by Colorni et al. used in GA2. The rows of the teachers' timetable (parent 1)are sorted in order of the local fitness. The block C of parent 2 consists of the rows excluded in the building block A, which consists of the rows of the best local fitness.

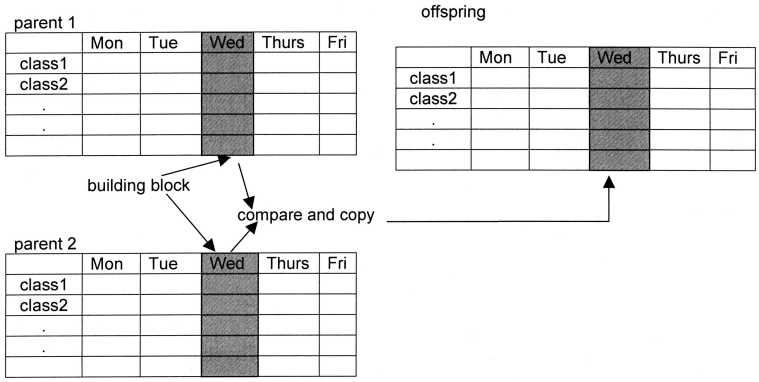

図４　GA3とGA4で採用したbuilding blockのトーナメントによる交叉．ここで，building blockは全クラスの1日のtupleより構成されている．交叉ではlocal fitnessを比較して，良いほうのbuilding blockを子孫にコピーする．

Fig. 4 Crossover by the tournament of the building blocks for GA3 and GA4, where the building block consists of the specific tuples assigned to the same day for all classes. After comparing both building blocks using their local fitness, one building block with a better local fitness is copied to the offsping.

GA1，GA2におけるfiltering は，時間割の同じクラスあるいは同じ教師でのtupleの入れ替え（swap）を，コスト１とコスト２についてより低い値を発見するまで続けている．GA1～4での集団のサイズは全て100とする．GAで採用した操作（operator）とその他のパラメータの値を表２に示す．

２．３　SAによる時間割編成

ここでは，以下のようなSA1，SA2を採用した．SA1はAbramson[5]の提案した方法をそのまま採用し，SA2はSA1の手法を修正したものである．

SA1：最初に全てのtupleをクラス時間割にランダムに割り当て，最初の時間割コストを計算する．そして，任意の２個のtupleを選択して入れ替え（swap）を行う．tupleの入れ替え後の時間割コストを計算し，入れ替え前のコストと比較して，コスト値が改善（低下）していればそのまま入れ替えを採用するが，悪化していれば再度入れ替えを行い，これを繰り返す．但し後述するように，入れ替え後のコストが悪化してもある確率でその

表２　GAの操作（operator）とパラメータ
Table 2 Operators and parameters of GAs

(a) 操作
(a) Operators

Operators \ GAs	GA1	GA2	GA3 &GA4
discard worse offspring			
repair for lost & duplication tuples			○
filtering	○	○	
Building block		○	○
Local fitness		○	○
tournament of building block			○
day mutation		○	
one-point mutation	○	○	○

(b) パラメータ
(b) Parameters

Parameters \ GAs	GA1	GA2	GA3 & GA4
one-point mutation rate	0.25	0.3	1.0
day mutation rate	_	0.01	_
crossover rate	0.9	0.8	1.0
filtering rate	0.9	1.0	_

結果を採用する．

SA2：最初のtupleの割り当てはランダムではなく，tupleのクラス情報に従って割り当てる．また，その後のtupleの選択と入れ替えも同じクラス内で行う．即ち，特殊教室を使用する場合を除いて，クラスの衝突（コスト２）をゼロに設定する．その他の操作及びパラメータはSA1と同じである．

上記の，SA1とSA2において，焼きなましの温度の更新は，更新の反復回数をnとして，$T_{n+1}=T_n \cdot R$ で行う．ここで，Rは冷却率（cooling rate）でR=0.99，0.90と仮定し，反復回数nはtupleの入れ替えを受け入れた回数に設定する．また，コストが悪化するtupleの受け入れ確率 $P(\Delta C)$ は，$P(\Delta C)=exp(-\Delta C/T)$ で与えられる．ここで，ΔCはコストの変化を示す．Abramsonの提案した，*maxsucessswaps*と*maxswaps*の値はそれぞれ1と1000である．即ち，成功したtupleの入れ替え回数ごとにnを1増加させ，入れ替えが失敗の場合は1000回を上限として，成功する（コストが低下する，または確率Pの範囲で受け入れ可能になる）まで反復する．また，温度の初期値T_0は10^{10}とする．

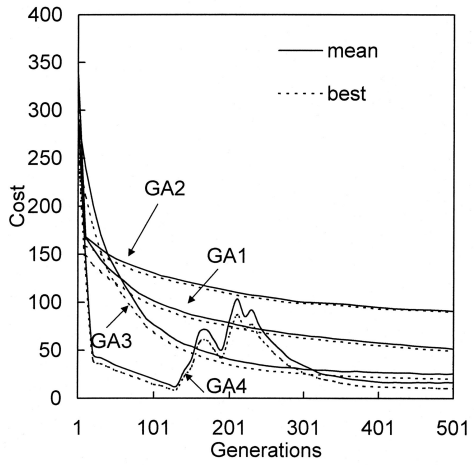

図５　各世代に対するGA1, GA2, GA3, GA4の合計コスト値．図中の‘mean’と‘best’は，集団の平均コストと最良個体のコストを，10回の試行について平均した値を示す．

Fig. 5 Total cost versus generation for GA1, GA2, GA3, and GA4, where 'mean' and 'best' indicate the cost values of the average and the best individuals in the population, respectively. The cost values are averaged over 10 runs.

Ⅲ．結果と検討

３．１　GAを使用した結果

図５はGAの世代を１から501まで増加させた場合の，GA1～4についてのコスト合計値の世代変化を示している．ここでのコストの値は，各世代における集団の平均値（mean）と最良値（best）を，10回の試行に対して平均した値を示す．

コストは世代が進行するにつれて改善されているが，その改善の程度は，GA4→GA3→GA1→GA2の順に悪化している．この理由として考えられることは，1）GA1とGA2はfilteringの効果のみである．2）GA2，GA3，GA4はさらにBBの効果が付加されている．3）GA3とGA4は１日単位のBBのトーナメント方式を採用している．4）GA4はさらに特定のコストを優先して減らしていることなどによると考えられる．従って，筆者らの提案したGA3，GA4がGA1，GA2よりも効果的であることがここでも示されている．また，特にGA4のように，特定の時間割条件を優先して解決する手法が有効なことは，既に報告した事前割当を採用した実験結果でも証明されている[6]．しかし，このGA4でも与えられた時間

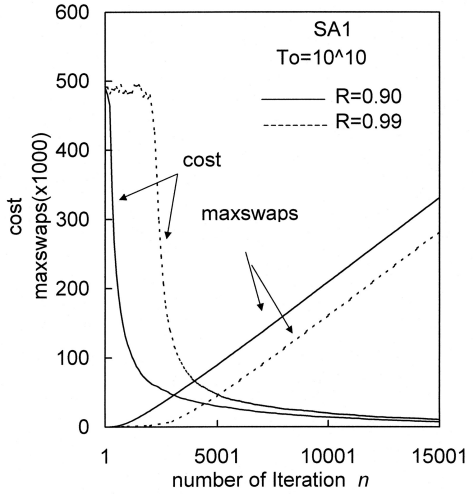

図６　SA1でR＝0.9と0.99の場合のコストと*maxswaps*の合計を繰り返し回数nに対して表示している．ここで，コストと*maxswaps*の値は10回の試行の平均である．

Fig. 6 Total of the costs and the *maxswaps* as a function of the number of iterations n when using SA1 with R=0.9 and 0.99. The values of the costs and the *maxswaps* are averaged over 10 runs.

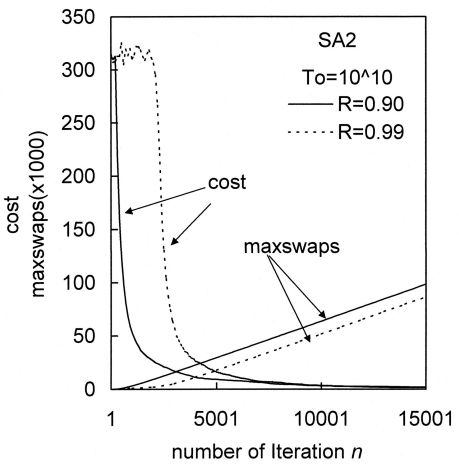

図７　SA2でR＝0.9と0.99の場合のコストと*maxswaps*の合計を繰り返し回数nに対して表示している．ここで，コストと*maxswaps*の値は10回の試行の平均である．

Fig. 7 Total of the costs and the *maxswaps* as a function of the number of iterations n when using SA2 with R=0.9 and 0.99. The values of the costs and the *maxswaps* are averaged over 10 runs.

割条件を全て解決することはできなかった。このことは，文献[6]の事前割当を採用した場合のGA3よりも劣る結果であるといえる．

３．２　SAによる時間割編成の結果

図６，７は反復回数nに対するコストの変化を，10回の試行で平均したSAによる時間割作成の結果を示している．ここで，図６はSA1，図７はSA2による結果を示しており，それぞれ文献[4]，[6]の結果の再確認のためにグラフ表示として示す. 図にはコストを評価した回数，即ち*maxswaps*の合計値の変化も同時に示している．これらの図よりSA2はSA1よりも良好な結果を示すことが分かる．しかし，SA2でも評価した回数以内ではコスト値の平均は完全にゼロにはならないことは既に文献[6]でも確認している．図５との比較から，SA2はGA4よりもやや良好な結果を示すことが分かる．

３．３　検討

今回新たに提案した，GA4による時間割作成の結果では，設定した時間割問題を完全に解くことはできなかった．既に報告している事前割当を採用したGA3による結果と単純に比較すると，このGA4の結果は改善されているとは言い難い．しかし，事前割当による方法では事前割当専用のアルゴリズムを別途用意する必要があること，また，拘束力の強い時間割問題の場合，事前割当自体が失敗する可能性もあることなど，事前割当のアルゴリズムに問題が生じる可能性もある．GA4は事前割当をGAによって行う方法を採用していると考えると，事前割当において拘束性の強いtupleを完全に割り当てられない場合に，さらにGA4を適用することも考えられる．GA4の改良については，例えばBBのコストを改善するlocal searchなどのtuple操作を追加する方法[7][8]なども検討すれば，より有効な時間割アルゴリズムになる可能性もある．SAによる方法でも事前割当が有効なことは既に報告しているので，拘束性の強い時間割問題ではできるだけ拘束条件を低下させてから，GA4，SA2のようなアルゴリズムを適用することが，より現実的な時間割問題の解決法のひとつであると考えられる．

Ⅳ．結　論

GAとSAによる時間割作成アルゴリズムによる時間割問題解決についての実験結果を報告した．結論として言えることは，今回提案したようなGAのアルゴリズムの改善でもある程度の効果はあるが，GAで事前割当を採用した場合と比較するとやや劣る結果であった. 従って，事前割当も含めたGAアルゴリズムの修正，および他の

アルゴリズムとの組み合わせ等の改良についてさらに検討することが必要である．SAについては，事前割当が完全に行えれば，SAアルゴリズム自体の改良の余地は少ないと考えられる．

文　献

[1] D. Abramson and J. Abela , 'A parallel genetic algorithm for solving the school timetabling problem', 15th Australian Computer Science Conference, pp. 1-11, 1992.

[2] Si-Eng Ling, 'Integrating genetic algorithms with a Prolog assignment problems as a hybrid solution for a polytechnic timetable problem', Parallel Problem Solving from Nature 2, pp. 321-329, 1992.

[3] A. Colorni, M. Dorigo and V. Maniezzo, 'A genetic algorithm to solve the timetable problem', Technical Report, 90-060 revised, Politecnico di Milano, 1992.

[4] 福島，田中，‘学校時間割問題における遺伝的アルゴリズムの適用について’，電子情報通信学会論文誌，Vol. J81-D-I，6，pp. 883-885, 1998.

[5] D. Abramson, 'Constructing school timetables using simulated annealing', Sequential and parallel algorithms, Management Science, Vol. 37, 1, pp. 98-113, 1991.

[6] 福島，'中学校時間割編成アルゴリズムの比較と編成システムの試作'，日本産業技術教育学会会誌，Vol. 42，1，pp. 1-11，2000.

[7] D. Corne and P. Ross,' Peckish initialisation strategies for evolutionary timetabling', Springer-Verlag Lecture Notes in Computer Science, Vol. 1153, pp. 227-240, 1996.

[8] E. K. Burke, J. P. Newall and R. F. Wear, 'A memetic algorithm for university exam timetabling', The Practice and Theory of Automated Timetabling, Springer-Verlag Lecture Notes in Computer Science, Vol. 1153, pp. 241-250, 1996.