

遺伝的アルゴリズムによる時間割作成システム

川 場 隆

Timetable Scheduling System using Genetic Algorithms

Takashi Kawaba

1. はじめに

本システムは、著者が1996年に提案したアイデア^[1]を実働システムとして構築したものである。本システムでは、時間割作成とは、各教員が選好する場所（時間割上のコマ）の奪い合いと捕らえ、教員の個別な満足度を統合して全体の最適化を図ることを第一の目標にしている。ただ、実用システムとして考えたとき、同じ教員が同一時間帯に講義を持たないとか、教室の確保が可能であるなど、複数の目的を最適化する方向で議論することも可能であるが^[2]、よく考えると、それらはシステムの目的ではなく制約条件にすぎない。実用システムで本当に大切なのは、単に制約条件を満たすということではなくそれに係る人間の満足度である。もちろん、時間割として実施可能であるために満足しなければならない条件はいくつも存在し、それらをすべて満たさねばならないことはいうまでもない。しかし、今回のシステム構築の経験からそれらが現システムの拡張によって可能であることは明らかになった。それについては、次の機会に詳細を論じるつもりである。

ところで、現実の利用では、全教員の選好データを収集しなければならないし、そうする以上は、作成結果をすばやくフィードバックしなくてはならないであろう。そのような場合に、システムがネットワーク上で利用できれば都合がよい。データ収集のフロントエンドから時間割作成の実行までを一貫してネット上で行い、結果を早く広くフィードバックできるからである。そこで、学内 LAN やインターネット上での稼動を前提し、本システムはプログラム言語 JAVA^[3]を使って作成した。JAVA は稼動するコンピューターの種類を選ばないばかりか、インターネット上での利用に都合のいい機能を持っている。現在はプロトタイプの段階なので、スタンドアローンのシステムとして作成しているが、たとえば、これをインターネットの WEB (WWWサービスクライアント) 上で稼動するように変更するのは極めて容易であるし、また、作成した時間割から WEB ページを生成するサブシステムも容易に構成できる。

さて、本論の目的は、このような特質を持つプロトタイプシステムについて、そのシステム構成と実現されたアルゴリズムを詳細に示すことである。今後、プロトタイプから実用シ

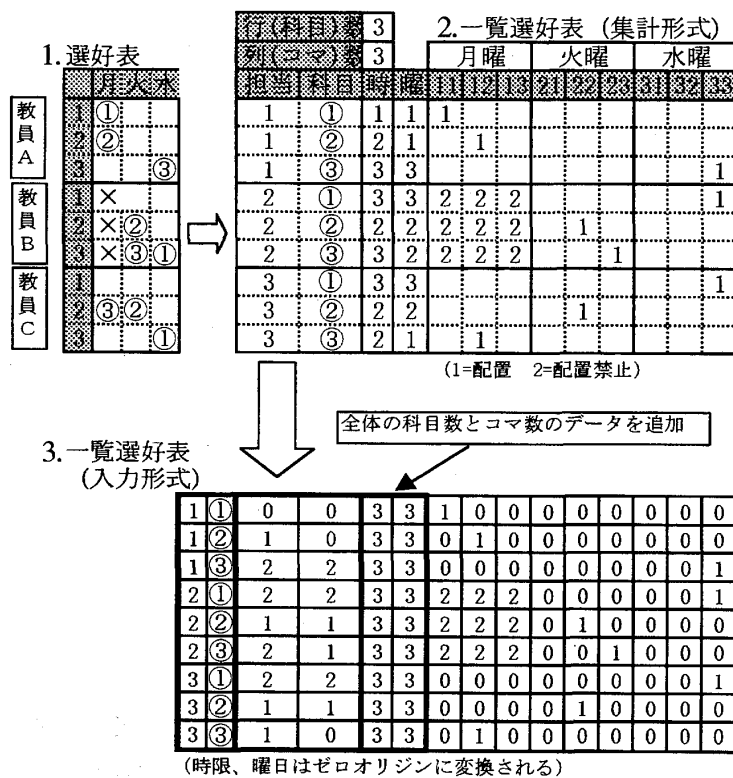
システムへ拡張する際、これらの知見は重要な指針となるであろう。

2. システムの構成

2. 1. 選好データ

選好データとは、各教員が時間割についての自分の選好（好み）を記述したもので、どの科目をどの曜日のどの時限で開講したいかを時間割上に示したものをいう。また、それには、どの曜日のどの時限は開講できない（配置禁止）かも×印で記入できる。図-1に示すのは簡略のため曜日数3、時限数3からなる場合の例である。各教員はそれぞれ3つの担当科目を持ち、教員Bは、月曜日がすべて開講できないという設定になっている。

この選好表を整列したものが「2. 一覧選好表」で、ここでは、担当者コード、科目コード、時限番号、曜日番号をデータのヘッダとして左側にまとめ、残る右半分が時間割上での配置状態になっている。記号は、1が当該科目を配置したコマ位置を示し、2は配置禁止としたコマ位置である。



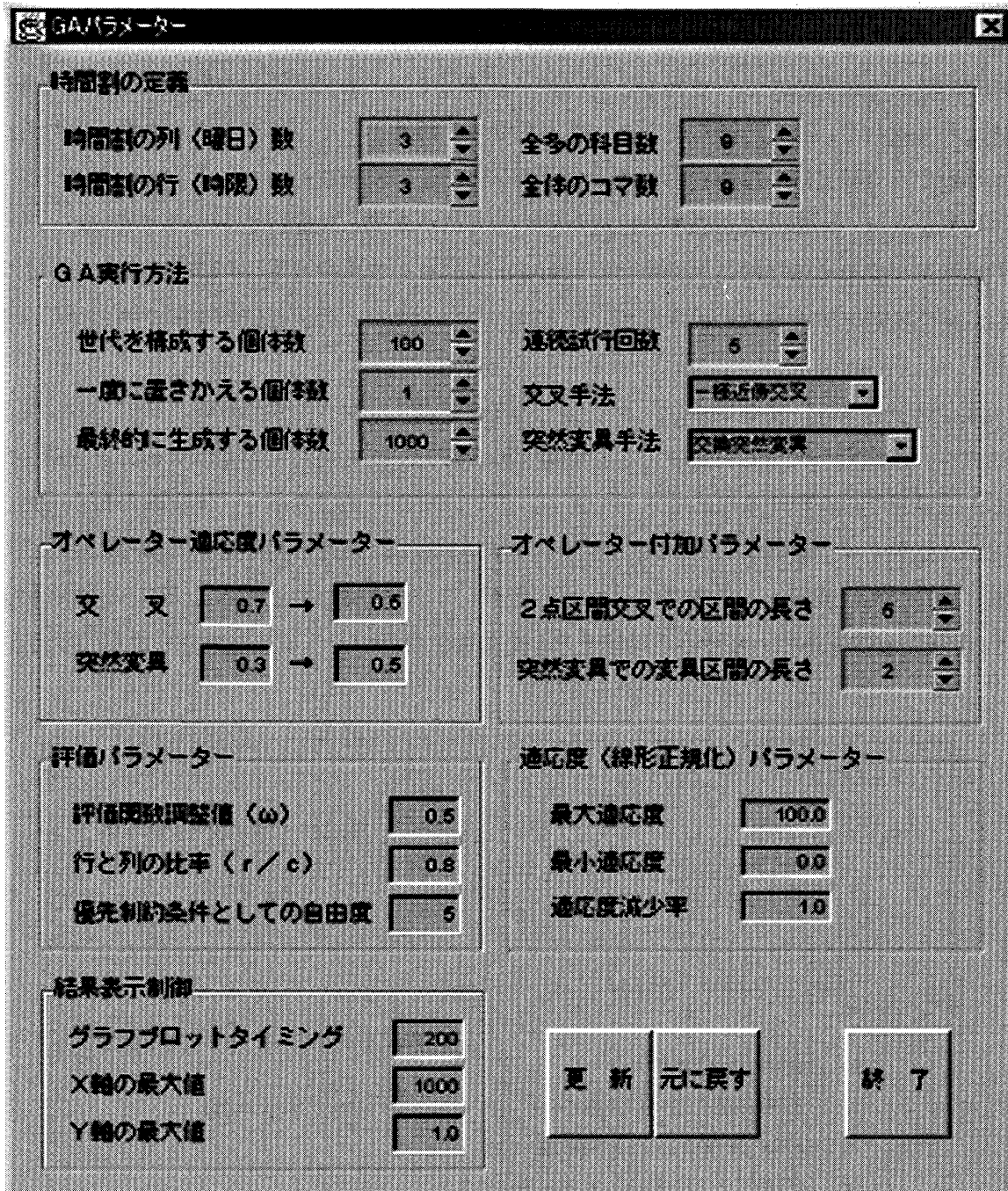
《図-1 選好データの形式》

最終的にシステムへの入力データとして使われるのは、「3. 一覧選好表 (入力形式)」である。これは1行が1件のデータとなるが、たとえば、最初の行は、「教員コード=1、科

目番号=①、月曜日 (= 0) の1時限目 (= 0) に配置を希望する」というデータである。

2. 2. 実行制御パラメーター

実行制御パラメーターは図-2のプログラム実行例に示すように7種類23個にのぼる。大きく分けると時間割のサイズや GA (Genetic Algorithms) の実行方式など全体の枠組みを指



《図-2 制御パラメータ入力画面》

定する基礎パラメーター、評価のための重みパラメーターや適応度を正規化するためのパラメーターなど反復実行しながらチューニングしなければならない動的パラメーター、そして結果をグラフ表示するための表示パラメーターに大別される。

基礎パラメーターでは、時間割の定義を行う他に、世代数や世代交代の方法を細かく指定できる。1996年の論文では、交差手法として一様近傍交差を、また突然変異として交換突然変異のみを提案していたが、本システムでは、可能性を探るため一様近傍交差を二点交差に拡張した二点区間近傍交差と、一定区間の遺伝子を混ぜ合わせるスクランブル突然変異を新たに付け加えた。二点区間近傍交差は、遺伝子全体に交差が一様に及ぶのではなく、確率的に定められる区間に対して交差オペレーターを適用する。遺伝形質の継続性では一様近傍交差よりも優れていると考えられるが、逆に多様性を犠牲にしてしまう欠点もある。どちらが本問題により効果的かは今後の数値実験で確かめる予定である。また、同様に、スクランブル突然変異は、ある区間を確率的に取り出してその中の遺伝子配列をかき混ぜるため、二つの遺伝子を交換する交換突然変異よりも大きな変異を生み出す可能性がある。どちらがよいか、また交差と突然変異の組み合わせはどれがよいかなど、今後の数値実験で明らかになる。

本システムでは、交差オペレーターと突然変異オペレーターを独立させ、それぞれにオペレーター適応度を付与している。これは選ばれた親に対してどちらのオペレーターを適用するかを確率的に決定するためのものである。このような方式はオペレータ主導の再生方式と言われているが、さらに、これらのオペレータ適応度は、GAの実行途中で動的に変化させる。すなわち、初期は、交差のウェイトを大きくして多様性をまとめるようにし、逆に終盤では、早い収束を避けて多様性を維持するために突然変異のウェイトを大きくするのである。

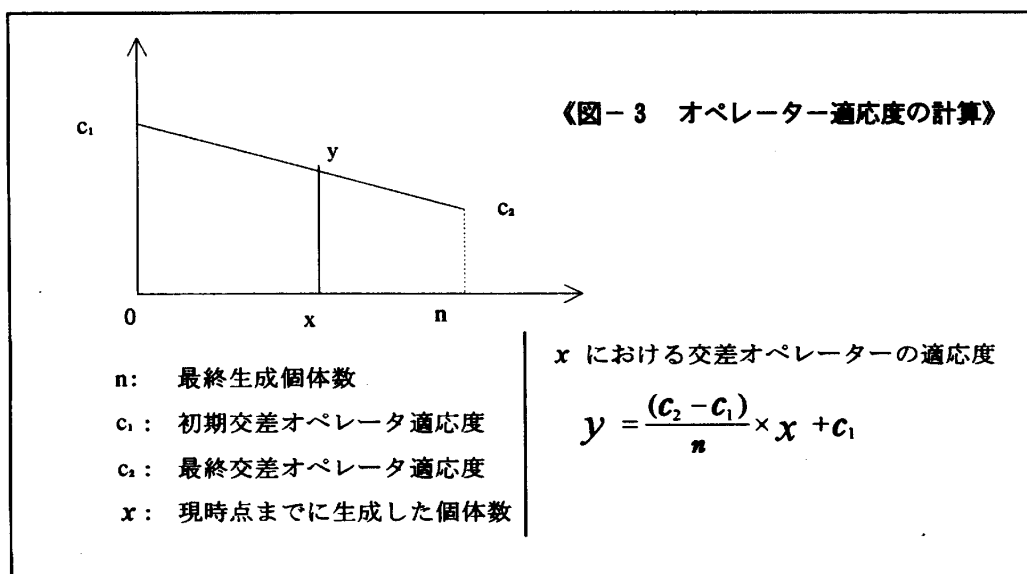


図-2のオペレーター適応度パラメーターは、このために設けられたもので初期のオペレーター適応度と最終のオペレーター適応度を指定することにより、途中のオペレーター適応度を図-3で示すように線形に変化させる。

また、本システムでは個体の評価値をそのまま適応度とするのではなく正規化することとした。というのも、評価値＝適応度としてしまうと、評価値の変動が小さい場合、親としての選択圧力が平準化してしまうからである。また、最大評価値と最小評価値に大きな差がある場合には、最大評価値の個体に急速に収束する可能性もある。そこで、各個体を評価値の順序に整列し、その順位を使って適応度を正規化する。負体的には、

$$\text{適応度} = \text{定数} - \text{順位} \times \text{刻み}$$

という形であるが、これは評価値の順位をもとに定率で適応度を増減するもので、評価値の変動が小さい場合でも選択競争を激化させることができるし、超個体 (super individual) がある場合でもそれほど急速に集団を支配できない。これは、線形正規化と呼ばれる手法で、一般に、刻み幅すなわち減少率を大きくすると競争が激化し、小さくすると緩和される。本システムでは線形正規化パラメーターにより、最大、最小の適応度を指定し、順位に基づく定率の増減幅を任意に指定できるようにした。

最後に、本システムでは、Syswerda (1989) が定常状態再生 (steady-state reproduction) と名付けた手順^[4]により世代交代のサイクルを実行する。その骨子は以下のようなものである。

1. 再生によって重複しない n 個の子を作る
2. 集団の n 個の個体を削除して場所を空ける
3. 子を評価して集団に加える

n は 1～2 で使うのが普通であるが、本システムでは n の値を「一度に置き換える個体数」でパラメータとして指定できる。また、子の評価にあたっては、最初に同じ個体がすでに集団内に存在しないかどうかチェックし、存在する場合はこれを棄却して新たな子を生成しなおす。

2. 3. 実行制御クラス

次に、本システムで GA がどのように制御されるか示すため、作成した主たる実行制御クラスを列挙し説明を加えておく。

①TSP クラス

GA を同一条件で複数回実行して平均値を算出し、結果をグラフ表示する。グラフは、横軸が世代数、縦軸が基準化した評価値の折れ線グラフである。複数回の実行における平均値

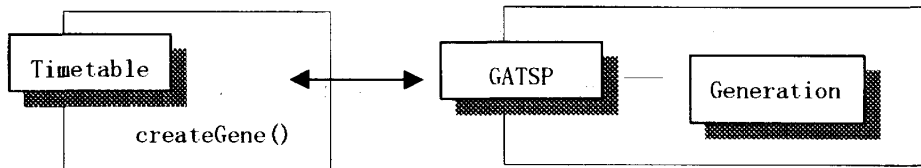
を表示するので、これにより設定したパラメーターの効果や GA システムの有効性を確認することができる。

②GATSP クラス

GA を実行して結果を返す。TSP はこの GATSP クラスを何度も呼び出すことにより GA プロセスを複数回実行する。GATSP はたとえば以下のようなベクトルを返すが、これは、500個の個体を生成する GA プロセスを1回実行し、100世代ごとの適応度をベクトルで返す場合のものである。

生成済み個体数	100	200	300	400	500
適 応 度	0.65	0.68	0.72	0.80	0.85

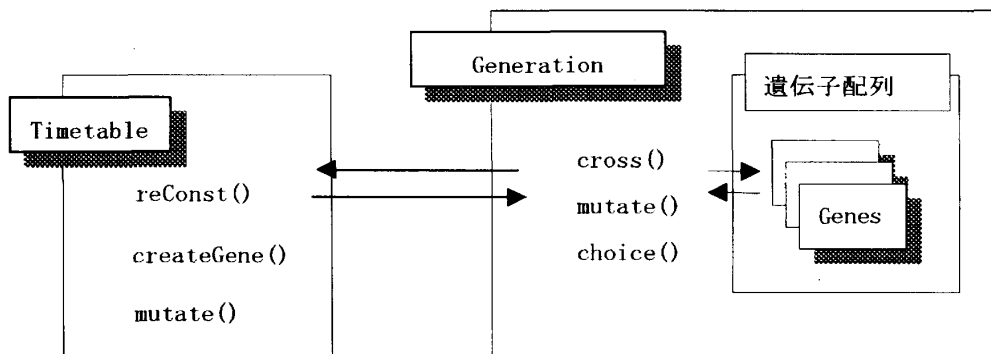
GA 実行の本体のクラスがこの GATSP であるが、GATSP は個体を生成するエンジンとして Timetable クラスを使う。Timetable クラスはそのメソッドである createGene() 関数が呼び出されるたびに新しい染色体（個体のこと）をひとつ生成して返す。染色体はすでに評価値が付与された状態で返される。



GATSP は1世代分の染色体プールを内部に持っている。染色体プールは、染色体オブジェクト Genes の配列を内部に保持する Generation クラスとして作成されており、ここに染色体をプールしていく。

③Generation クラス

染色体プールすなわち「世代」を保持するクラスである。ルーレット方式の親選択、交差、突然変異などを実行するメソッドをもっているが、それらの実行エンジンは Timetable クラスである。例えば、このクラス内で交差は以下のように実行される。



2. 生成途中

	担当	科目	選好	空き	空き 割当	1限目			2限目			3限目		
						月	火	水	月	火	水	月	火	水
						0	9	9	6	9	9	6	9	0
						0	-1	-1	-1	-1	-1	-1	-1	0
教員 A	1	①	(0 / 0)	0	0	1	1	1	1	1	1	1	1	1
	1	②	(1 / 0)	9	-1	1	0	0	0	0	0	0	0	1
	1	③	(2 / 2)	9	-1	1	0	0	0	0	0	0	0	1
教員 B	2	①	(2 / 2)	0	8	2	1	1	2	1	1	2	1	1
	2	②	(1 / 1)	6	-1	2	0	0	2	0	0	2	0	1
	2	③	(2 / 1)	6	-1	2	0	0	2	0	0	2	0	1
教員 C	3	①	(2 / 2)	9	-1	1	0	0	0	0	0	0	0	1
	3	②	(1 / 1)	9	-1	1	0	0	0	0	0	0	0	1
	3	③	(1 / 0)	9	-1	1	0	0	0	0	0	0	0	1

3. 生成完了

	担当	科目	選好	空き	空き 割当	1限目			2限目			3限目		
						月	火	水	月	火	水	月	火	水
						0	0	0	0	0	0	0	0	0
						0	1	4	5	2	7	5	3	
教員 A	1	①	(0 / 0)	0	0	1	1	1	1	1	1	1	1	1
	1	②	(1 / 0)	0	1	1	1	1	1	1	1	1	1	1
	1	③	(2 / 2)	0	5	1	1	1	1	1	1	1	1	1
教員 B	2	①	(2 / 2)	0	8	2	1	1	2	1	1	2	1	1
	2	②	(1 / 1)	0	2	2	1	1	2	1	1	2	1	1
	2	③	(2 / 1)	0	4	2	1	1	2	1	1	2	1	1
教員 C	3	①	(2 / 2)	0	7	1	1	1	1	1	1	1	1	1
	3	②	(1 / 1)	0	6	1	1	3	1	1	1	1	1	1
	3	③	(1 / 0)	0	3	1	1	1	1	1	1	1	1	1

《図-4 Timetable の状態遷移》

この例は、染色体がランダム生成（初期生成）される場合の遷移を表したものである。表中の太字の数字は時間割として割当が済んだ部分を表している。ここで「割当」欄を横に見ると、特定の時限（コマ）にどの科目を割り付けたかが分かる。数値は割り付けた科目の相対行番号（ゼロオリジン）である。初期状態ではすべて未配置（-1）状態であるが、染色体の生成過程で徐々に埋まってくる様子が図の2.、3.で灰色の網掛けによって示してある。

3.の状態における網掛けした部分は、完成した染色体として遺伝子型を扱うクラスに参照される。また、逆に、交差や突然変異などのオペレーター操作では、受け取った染色体はこの表に展開されて、表現型に復元される。この表データは、1996年の論文では「配置参照リスト」と呼んでいたものに相当する。

3. 2. バックトラック機構

バックトラックは経路探索の代表的な手法である。解へ向かって経路を進んでいくとき、判定条件からこれまで進んできた経路が誤りであるとわかった場合に、分岐点まで経路を後戻りして、そこから別の方向へ分岐をたどることで探索を再開する。バックトラックはしなやかに越したことはないが、先読みなどの手法で経路を絞り込む方法では計算のオーバーヘッドが大きすぎる場合は、バックトラックが有効である。

本論の場合、配置参照テーブルのコマに科目を割り当てていく過程で、科目間の競合関係からどうしても割当を継続できなくなる場合（配置不能）がある。配置不能にならないように、特定の科目を割り当ててはいけないコマ位置を事前に発見し、配置不能のマークをつけて絞込みをしておくことは可能である。しかし、そのためには、すべての空いているコマ位置を網羅的に調べなければならず、相当のオーバーヘッドが見込まれるのである。

そこで、事前の絞込みをやめ、バックトラックを採用した。このために探索経路を記憶するスタックを設けてすべての探索経過を記録している。ここで、バックトラックが起こる、すなわち、配置不能と判断するのは、以下のいずれかが成り立つ場合である。

- ①まだ割当を決定していない科目で、配置できる空きコマがひとつもないものがある。
- ②まだ割当を決定していない科目の数が空いているコマの数よりも多い。

図-5.1はそのようなケースを説明するためのものである。図-5.1では、割付も終盤に来て二つの科目が残るだけとなっている。この状態では、上記の条件①②のどちらにも該当しないので、配置処理は続行される。

先に割り当てる科目は確率的に決定されるので、例えばそれが教員Cの科目番号②番の科目であったとしよう。この科目の可能な割付位置は **A** か **B** のいずれかであるが、ここでは **A** が選ばれたとする。次に教員Bの科目番号②番の科目が残っているので配置を継続しようとするが、その際、継続条件の照合が行われ上記の①②両方が成り立っていることが分かる。すなわちこれは配置不能である。

					1 限目			2 限目			3 限目			
					月	火	水	月	火	水	月	火	水	
空き					0	0	2	0	0	0	1	0	0	
割当					0	1	-1	3	5	2	-1	6	3	
教員 A	1	①	(0 / 0)	0	0	1	1	1	1	1	1	1	1	
	1	②	(1 / 0)	0	1	1	1	1	1	1	1	1	1	
	1	③	(2 / 2)	0	5	1	1	1	1	配置禁止	1	1	1	
教員 B	2	①	(2 / 2)	0	8	2	1	1	2	1	2	1	1	
	2	②	(1 / 1)	1	-1	2	1	0	2	1	1	2	1	1
	2	③	(2 / 1)	0	4	2	1	1	2	1	1	2	1	1
教員 C	3	①	(2 / 2)	0	7	1	1	1	1	1	1	1	1	
	3	②	(1 / 1)	2	-1	1	A	0	1	1	B	0	1	1
	3	③	(1 / 0)	0	3	1	1	1	1	1	1	1	1	



《図-5.1 バックトラック以前》

					1 限目			2 限目			3 限目		
					月	火	水	月	火	水	月	火	水
空き					0	0	0	0	0	0	1	0	0
担当	科目	選好	空き	割当	0	1	7	5	5	2	-1	0	0
教員 1	①	(0 / 0)	0	0	1	1	1	1	1	1	1	1	1
	②	(1 / 0)	0	1	1	1	1	1	1	1	1	1	1
教員 A	③	(2 / 2)	0	5	1	1	1	1	1	1	1	1	1
教員 2	①	(2 / 2)	0	8	2	1	1	2	1	1	2	1	1
	②	(1 / 1)	1	-1	2	1	(1)	2	1	1	(2)	1	1
教員 B	②	(2 / 1)	0	4	2	1	1	2	1	1	2	1	1
	③	(2 / 2)	0	7	1	(A)	1	1	1	1	(B)	1	1
教員 3	①	(2 / 2)	0	7	1	(A)	(1)	1	1	1	(1)	1	1
	②	(1 / 1)	0	2	1	1	1	1	1	1	1	1	1
教員 C	③	(1 / 0)	0	3	1	1	1	1	1	1	1	1	1

《図-5.1 (続き) バックトラック以前》

そこで継続を中止し、バックトラックが起動される。図-5.2がその様子を示している。バックトラックでは、直前の経路選択 (A) がスタックから取り出され、これに基づいて選択の取り消し処理が行われる。スタックに保存してある情報は選択操作の種別 (配置処理、配置不許可処理) と処理対象とした行・列番号である。取り消し処理では、(A) の位置の配置マーク (=1) を取り消し、代わりに配置不許可マーク (=3) を書き込む。

●バックトラック処理

					1 限目			2 限目			3 限目		
					月	火	水	月	火	水	月	火	水
空き					0	0	(1)	0	0	0	1	0	0
担当	科目	選好	空き	割当	0	1	-1	5	5	2	-1	0	0
教員 1	①	(0 / 0)	0	0	1	1	1	1	1	1	1	1	1
	②	(1 / 0)	0	1	1	1	1	1	1	1	1	1	1
教員 A	③	(2 / 2)	0	5	1	1	1	1	1	1	1	1	1
教員 2	①	(2 / 2)	0	8	2	1	1	2	1	1	2	1	1
	②	(1 / 1)	1	-1	2	1	(0)	2	1	1	(2)	1	1
教員 B	②	(2 / 1)	0	4	2	1	1	2	1	1	2	1	1
	③	(2 / 2)	0	7	1	(A)	1	1	1	1	(B)	1	1
教員 3	①	(2 / 2)	0	7	1	(A)	(3)	1	1	1	(0)	1	1
	②	(1 / 1)	2	-1	1	1	1	1	1	1	1	1	1
教員 C	③	(1 / 0)	0	3	1	1	1	1	1	1	1	1	1

《図-5.2 バックトラック》

これで、教員Cの科目番号②番の科目は、(A) を選択できなくなる。(A) 以外から配置すべきコマを探し直すのだが、この例の場合 (B) 以外にないのでそれが選ばれることとなる。結果的に、この選択により教員Bの科目番号②番の科目も配置不能とはならず、すべての配置作業が完了する。

3. 3. ヒューリスティクスの導入

さて、バックトラックは便利な機構であるが、起動しないに越したことはない。バックトラックは計算のオーバーヘッドには違いないからである。そこで本論では、バックトラック

を起こりにくくするため、時間割作成の経験則を導入することとした。それは、「自由度の低い科目は先に配置する」というものである。

例えば、どうしても月曜日以外は開講できないなどといった事情により、時間割上で選択できるコマ位置が極端に少ない科目の場合、人間の手で作成する時は、それらを先に割り付けてしまうのが常である。これは時間割作成上の代表的な経験則で、こうすることによりクリティカルな割付を避け、より簡単に作成できることが知られているのである。

そこで、本システムでは「割付可能なコマが n 個以上ない科目は優先的に割り付ける」というアルゴリズムを組み込んだ。 n は「優先条件としての自由度」パラメーターとして実行時に GUI の入力画面から指定できる。これによって、時間割に割り付けを行うすべての処理に先だって、該当する科目がピックアップされ優先的に処理される。

3. 4. 交差における最適化

染色体を大量に生成して初期世代を作る処理では、ランダム生成を行う。ランダム生成とは、どの科目から先に配置するか、(ヒューリスティクスによる優先グループを除いて) その順序をランダムに決定する生成処理のことである。

これに対して、交差では、遺伝子の半分は親Aの配置をそのまま引き継ぎ、残り半分を親Bの情報に基づいて配置する。ただし、親Bと全く同じ配置は不可能であるから、親Bに似せるということしかできない。したがって、同じ科目が親Bではどの位置に配置されているかを調べ、空いている位置のうち、そこに最も近い位置に配置する(近傍交差)。

ところが、この近傍配置自体が「どう配置すれば最適化された近似になるのか」という問題を含んでいる。適正に処理するためには、近似度を配置全体から定義し、その最適化問題を解くことが必要であるようにも思える。しかし、GAの本質に立ち返って考えると、その最適化自体も、選択と淘汰の過程の中で最適解に収束していくものと考えられる。そこで、本システムでは交差処理においても、どの科目から配置するかをランダムに決定し、決定された科目から計算される近似度(=距離)のみを指標にして割付を行うこととした。

4. まとめと展望

1996年にアイデアをまとめてから、はや3年が過ぎようとしている。この間、時折中断しながらも続けてきたシステム化の作業は、ようやくプロトタイプを完成するところまでこぎつけた。システムのインプリメントの中で、当初のアイデアを変更しなければならなくなったのは、探索の絞り込みをやめてバックトラックに頼らねばならなくなった部分である。バックトラックは基本的には網羅的探索の一種であるが、ヒューリスティクスを導入することで、効率よく機能する(つまり、頻繁には起動されなくなる)ことが期待される。

本システムの特徴は、単に物理的な組み合わせの可能性を求めるのではなく、教員の側か

らみた満足度を最適化する点にある。今後は、まず、プロトタイプシステムによる数値実験を繰り返して、動的パラメーターのチューニングを行う計画である。GAでは、周知のようにこれらのパラメーターの適正値は試行錯誤によって割り出すしかないのである。

次に、実用システムへ向かっての拡張に着手する。基本的に致死遺伝子が生じないアルゴリズムを開発していく方針で、例えば、学部学科をまたぐ輻輳カリキュラムに対しては、現在の Timetable クラスをネットワーク構造に配置する拡張が有力であると考えている。その際、染色体もネットワーク状に配置されたものとなり、交差や突然変異はネットワーク上を他の染色体へ伝播していくことになる。

また、教室など設備の利用制約については、すでの実現している染色体の重複チェックと同じレベルの検証過程として扱うこともできる。それは、不適であれば捨てて生成し直すという方法である。しかし、これは致死遺伝子として生成してしまうのと変わらないので、やはり配置を決定する時点で検証を行うようにアルゴリズムを一部手直しすることが必要となろう。具体的には、設備資源が利用不能であれば、バックトラックして配置不許可とする処理を追加することとなろう。ここにも何らかのヒューリスティクスを導入できればよいと考えている。

参 考 文 献

- [1] 川場隆：遺伝的アルゴリズムの時間割作成問題への適用，活水論文集第39集，pp53-62，1996
- [2] 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，人口生命とその応用シンポジウム論文集，pp49-55，1997
- [3] Campione, M, Walrath, K: The Java™ Tutorial, Second Edition, Addison Wesley, 1998
- [4] Syswerda, G.: "A Study of Reproduction in Generation and Steady-State Genetic Algorithms," Rawlins, G.(ED.), *foundation of Genetic Algorithms*, Morgan Kaufman, 1991