特集 遺伝的アルゴリズム

解説

GA のスケジューリング問題への応用

西川禕一*

1. はじめに

遺伝的アルゴリズム（GA）の1つの典型的な応用は最適化への応用，特に組合せ最適化問題への応用である．組合せ最適化問題では，離散変数の有限個の組合せの中から最適解を見出すことになるので，理論的には有限の手数で厳密な解を得ることができるが，問題の規模が大きいと途方もない計算の手数，したがって途方もない計算時間を要することになり，厳密解を求めることは実際上不可能である．ちなみに，変数の数を n として，手数が e^n のオーダになる問題は NP 完全問題と呼ばれるが，組合せ最適化問題の多くは NP 完全である．

したがって，現実的な大規模組合せ最適化問題に対しては，厳密解を求めるための典型的な方法である分枝限定法などは，事実上使えないといってよい．別のいい方をすれば，最適に近く実用的には十分良い準最適解を求める，系統的な解法が必要とされる．従来からも問題の特徴に基づくさまざまのヒューリスティクスが用いられているが，GA は良い近似解を求めるための比較的一般的な枠組を与える，1つの有力なツールといえる[1),2)]．

ここでは，組合せ最適化問題の典型例であり，実際上も重要なジョブショップ型スケジューリング問題（Jobshop Scheduling Problem, JSP）を対象として，GA の適用について留意すべきポイントを指摘しながら解説してみよう．

2. スケジューリング問題

物を作り上げる工程，すなわち生産や建設のプロジェクトを完成する1つの工程においては，いく種類かの機械を使って，工程の単位となるいくつかの仕事（job）を順序よく進めていかなければならない．なんらかの意味で最も適切に仕事の順序を決定する問題を，一般にスケジューリング問題という[3)]．

つぎのようなスケジューリング問題を考えてみよう．1つの仕事には複数個の作業が含まれていて，各作業を処理できる機械は特定のものに限られている．逆に，各機械で処理できる作業はいく種類かある．またいくつかの仕事の間には，先行関係（ある仕事を始めるには，これだけの仕事が終っていなければならないという関係）が指定されている．

このような条件のもとで，最も早くプロジェクトを完了するにはどのように仕事を進めればよいか．記号を使って問題を記述すれば，つぎのようになる．

2.1 問題の記述

n 個の仕事 J_j $(j=1,\cdots,n)$ を m 台の機械 M_i $(i=1,\cdots,m)$ で処理するプロジェクトがある．仕事 J_j は n_j 個の作業 O_u $(u=N_{j-1}+1,\cdots,N_j$; $N_0=0$, $N_j=\sum_{k=1}^{j} n_k)$ からなっていて，これら n_j 個の作業 O_u は u の増加順に処理されなければならない．また，各作業 O_u は機械 M_{μ_u} (μ_u は作業 O_u が処理される機械の番号）上で分割なしに所要の時間 p_u で処理されるものとする．あらかじめ指定された仕事間の先行関係を満たし，プロジェクトの完了時間 $C_{\max}$ が最小となるように，各機械上で各作業が処理開始される時刻を決定せよ．

機械と作業の対応関係および仕事の先行関係を満たすだけでよいなら解はいくつもあるので，その中でプロジェクトが最も早く完成できるような最適解を求めよ，というのが問題である．上記のようなスケジューリング問題を，特にジョブショップ型スケジューリング問題と呼んでいる．

なおスケジューリング問題において，複数台ある機械の機能がすべて同じ場合を並列機械型，機械の機能が異なる場合をショップ型という．さらにショップ型

* 京都大学工学部 京都市左京区吉田本町
キーワード：スケジューリング問題 (scheduling problem)，ジョブショップ問題 (jobshop problem)，選択グラフ (disjunctive graph)，近傍モデル (neighborhood model)，並列計算 (parallel computation).

のうち，すべての仕事で使用機械の順序が同じものをフローショップ型，仕事ごとに機械の順序が異なって指定されるものをジョブショップ型，どの仕事でも機械の順序が自由なものをオープンショップ型と呼ぶ．

2.2　選択グラフによる問題の表現

数理的にいえば，問題を最も明確に表現する方法は数理計画問題の形に定式化することである．実際に，上記の問題は混合整数計画問題（MIP）として定式化することができる[3]．決定変数は，同じ機械で処理される作業間の順序を表わす変数であり，これらが決まると各作業の開始時刻は一意に決定される．しかし先にも述べたように，問題の規模が大きいとき，このMIP を厳密に解くのは事実上不可能である．

そこで GA を適用する準備として，ここでは問題を選択グラフ (disjunctive graph) で表わしてみよう．グラフによる表現は，問題の直感的理解のためにもきわめて有用である[5]．まず作業を節点 (node) に対応させ，あらかじめ定まっている作業間の先行関係を節点間の有向弧 (arc) で表わす．このとき，プロジェクトの開始節点0と終了節点＊をダミー節点として付け加えておく．また，MIP の決定変数に相当する同一機械で処理される2作業間の先行関係は，対応する2節点間の選択弧 (disjunctive arc) と呼ばれる有向弧の対で表わしておく．さらに，すべての有向弧には，それらの始点が表わす作業の処理時間を長さとして記入する．一例として**表1**のプロジェクトを選択グラフで表現したものが**図1**である．図中の破線が選択弧対を表わす．

選択グラフで表わした問題では，各選択弧対から一方の弧を除去すること（選択弧対の解消）で決定変数

表1　プロジェクトの例

仕事	作　業 （機械，処理時間）		先行仕事
J_1	$O_1(M_1, 5)$	$O_2(M_2, 4)$	—
J_2	$O_3(M_1, 3)$	$O_4(M_2, 4)$	—
J_3	$O_5(M_3, 2)$	—	J_1, J_2

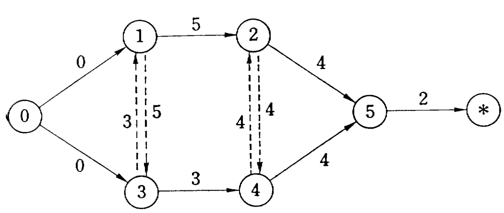


図1　選択グラフによる問題の表現
節点0および＊は，それぞれプロジェクトの開始および終了を表わすダミー節点．

が定まり，すべての選択弧対が解消されたとき，1つのスケジュールが与えられる．ただし解消する際，グラフに閉路（サイクル）が生じないように注意しなければならない．閉路がないとき，節点0から＊までのクリティカル・パスの長さが C_{max} を与える．

2.3　スケジューリング問題に対する GA の特徴

スケジューリング問題に対してはこれまで種々の解法が提案されている[4]．それらは，厳密解法と短時間で近似解を求める近似解法に大別される．

まず厳密解法としては，MIP に定式化して解く方法や分枝限定法を直接適用する方法などがあるが，いずれにしても広い解空間（可能領域）をくまなく探索することを基本とするので，膨大な計算時間を要する．

それに対して，近似解法では問題に関する先験情報を利用して探索空間を絞り込むことを基本とするが，質の良い解を得るためにはできるだけ広い空間を探索することが望ましく，一方，時間を短縮するためには先験情報によってできるだけ探索空間を狭めることが望ましい．これらの相矛盾する2つの要求は，トレードオフとして考えなければならない．前者を重視するのがランダム探索法（モンテカルロ法）であり，後者に重きを置くのが逐次改善法やリスト・スケジューリング法である．

GA では，問題の解候補を 0/1 記号列からなる個体（遺伝子列＝染色体）に対応させる．いいかえれば，実際の解候補が表現形 (phenotype) であり，0/1記号列の個体が遺伝子形 (genotype) である．そして個体に対して選択 (selection)，交叉 (crossover) や突然変異 (mutation) などの演算を繰返し適用することによって，逐次適応度の高い個体（すなわち質の良い解候補）を生成していくアルゴリズムであるから，GA は逐次改善法の範疇に属する解法とみることもできる．ただし，単一の個体を扱うのでなく個体の集合 (population) を対象としてその適応度を高めていくこと，記号列によって表わされる個体全体のみでなくその部分列 (building block) も評価され，個体の適応度を高めるに役立つ部分列が選択によって同時並列的に増えていくこと (intrinsic parallelism) などが大きな特徴である．これらの特徴が，先述の二要求をそれぞれ適度に満たす優れた性質を与えることになる[1),2)]．

3.　遺伝的アルゴリズムの構成

3.1　選択グラフに基づく方法

GA を具体的な問題に適用しようとするとき，最も大切なポイントは問題をどのように表現し，どのよう

に GA の遺伝子形に翻訳するかという点，すなわち問題をどのように GA にコーディングするかという点である．応用結果の成否はもっぱらこの点の適否にかかっているといっても過言ではない．ここでは，まず筆者らが提案した選択グラフに基づいて個体の遺伝子形を定める方法について述べよう[5]．

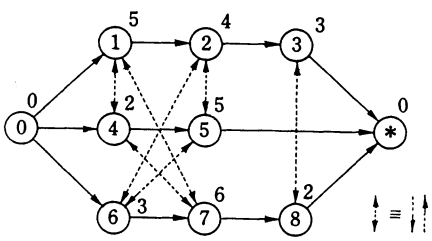
（1） 個体の遺伝子形

選択グラフにおける選択弧対の解消方向（どちら向きの選択弧を除去するか）を 0 と 1 で表わし，この 0/1 記号を選択弧対の数だけ並べた記号列を個体の遺伝子形とする．たとえば，**図 2(a)**のグラフに対しては

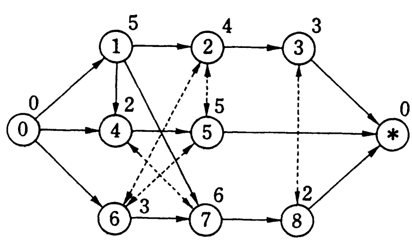
$$\begin{matrix} 0 & 0 & 1 & 0 & 1 & 0 & 1 \\ d_{14} & d_{17} & d_{47} & d_{25} & d_{26} & d_{56} & d_{38} \end{matrix}$$

の形である．ただし，d_{ij} は節点 i, j 間の選択弧対を表わし，0 は番号の大きな節点から小さな節点への選択弧を除去すること，1 はその逆と約束しておく．

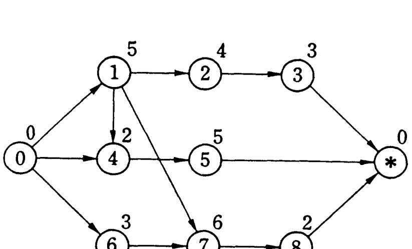
（2） 表現形への変換（デコーディング）規則



(a) 選択グラフ



(b) 現在の選択弧対解消状況を示すグラフ



(c) 未解消の選択弧対が除去されたグラフ

図 2 C_{max} を求めるクリティカル・パス計算
簡明のため，各弧の始点にその長さが記入されている．

実行可能なスケジュールを表わすグラフにおいては，閉路は一切含まれてはならない．上記の遺伝子形で，各選択弧対を独立に解消（独立に 0 と 1 を決定）すれば，きわめて高い確率で閉路が生じる．特に GA の場合，初期個体集合において閉路がないような個体ばかりを選んだとしても，交叉や突然変異演算を施せばたちまち閉路を含む個体が発生するであろう．

そこで，各記号の値を選択弧対の優先解消方向と解釈し，遺伝子形から表現形（実際のスケジュール）への変換（デコーディング）規則をつぎのように定める．

0° すべての選択弧対を未解消にしておく．

1° 現在までの選択弧対の解消状況を示すグラフ（例として図 2(**b**)）に対して，未解消の選択弧対をすべて除去したグラフ（図 2(**c**)）でクリティカル・パス計算を実行し，各節点（作業）の最早開始時刻を求める．その結果，1つのスケジュールが定まる．

2° 上で求まったスケジュールでコンフリクト，すなわち同一機械上で 2 つ以上の作業の処理時間が重なる状況が発生していなければ，終了節点＊の最早開始時刻として C_{max} を求める．コンフリクトが発生していれば，最も早い時刻にコンフリクトが生じている 2 節点間の選択弧対を，記号の値を参照して解消し，1° へ戻る．

たとえば，図 2 の例題で，遺伝子形が

0 0 1 0 1 0 1

のときのクリティカル・パス計算は，**表 2** のように行われる．また，最終的な選択弧対の解消状況を**図 3** に

表 2 クリティカル・パス計算の例

ステップ	解消されるべき選択弧対	節点の最早開始時刻	コンフリクト
1	—	0 5 9 0 2 0 3 9 12	(1, 4)
2	$d_{14}(1\to4)$	0 5 9 5 7 0 3 9 12	(1, 7)
3	$d_{17}(1\to7)$	0 5 9 5 7 0 5 11 13	(4, 7)
4	$d_{47}(7\to4)$	0 5 9 11 13 0 5 11 18	(3, 8)
5	$d_{38}(8\to3)$	0 5 13 11 13 0 5 11 18	—

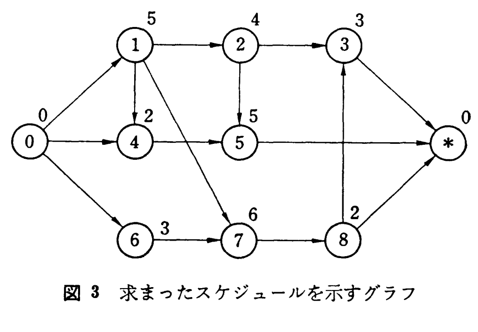


図 3 求まったスケジュールを示すグラフ

示す．

（3）　適応度の計算

GA では適応度の計算が重要である．ここではつぎの式によって $C_{\max}$ に対応した適応度 $f(C_{\max})$ を求める．

$$f(C_{\max})=\frac{C^{u}_{\max}-C_{\max}}{C^{u}_{\max}-C^{l}_{\max}}\times 100 \qquad (1)$$

ただし，$C^{u}_{\max}$ および $C^{l}_{\max}$ はそれぞれ $C_{\max}$ の上限値および下限値であり，たとえば前者には全作業の処理時間の和，後者には選択弧対をすべて除去したグラフのクリティカル・パスの長さを用いる．

（4）　遺伝子形における冗長性

上に示した遺伝子形では，表現形へ変換する際に参照されない記号がいくつか含まれている（どの記号が参照されないかは，クリティカル・パス計算を実行してみないとわからない）．図2の例では，3，5および6番目の記号（それぞれ d_{47}, d_{26} および d_{56} に対応）が参照されない．このように，遺伝子形には冗長性が含まれている．つまり遺伝子形としては異なるものが同じ表現形（スケジュール）となることを許している．これは(2)の変換規則からもわかるように，いかなる遺伝子形からも閉路をもたない実行可能なスケジュールを生成させるための1つの工夫である[5]．

（5）　計算例と考察

比較的簡単な，仕事間に先行関係のない8仕事の例題を示す．なお，遺伝演算子としては選択，交叉および突然変異の標準的な3種のみを用いたが，その内容はつぎのようなものである．ただし，ここでいう個体とはもちろん遺伝子形のことである．

選択：ある世代の個体集合中の各個体 x についてその適応度 $f(x)$ を求め，また集合の適応度の平均値 $\bar{f}$ を計算する．そして，個体 x の子 (offspring) の数の期待値を $f(x)/\bar{f}$ とする．

交叉：個体集合内の2個体をランダムに選んでペアリングし，各ペアについて交叉点 (cross point) をランダムに2カ所選び，確率 P_c で交叉する．

突然変異：各個体について1ビットをランダムに選び出し，確率 P_m で記号を反転する．

なお，初期個体集合は一様乱数を用いて生成した．

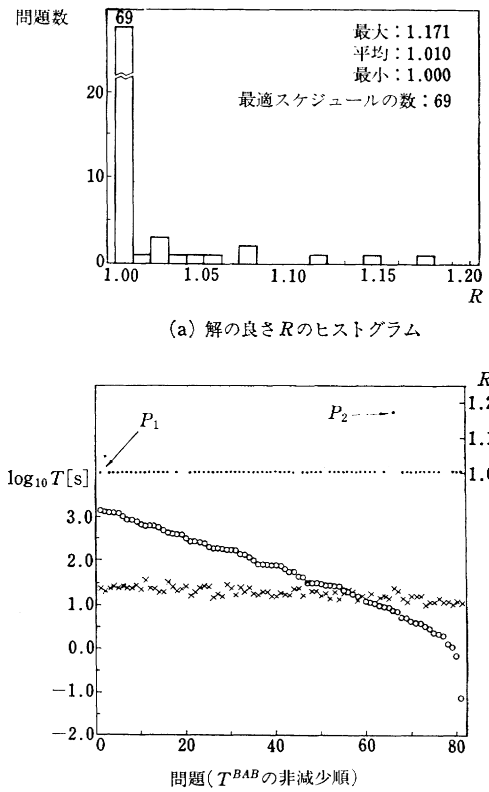
さて，

各仕事に含まれる作業数：2～4

作業を処理する機械数：M_1～M_6 の6台

各作業の処理時間：1～12

として一様乱数によって決定した100個の問題に対して GA を適用し，また厳密解法である分枝限定法 (BAB) も適用して比較した．GA では，個体集合サイズを 20，計算世代数を 50，交叉確率 P_c を，0.6，変異確率 P_m を 0.005 と設定した．



(a) 解の良さ R のヒストグラム

問題（T^{BAB} の非減少順）

×および○は，それぞれGAおよびBABによる結果

(b) 問題ごとの計算時間と R

図 4　GA と BAB による結果の比較

図 4に，GA によるスケジュールの良さ R：

$$R=\frac{C^{\mathrm{GA}}_{\max}}{C^{\mathrm{BAB}}_{\max}}\quad (=\text{最小値に対する比}) \qquad (2)$$

および GA と BAB との計算時間の比較を示す．ただし100個の問題のうち，BAB で最適スケジュールが求まらなかった（生成ノード数 200,000 個で打切り） 19 個の問題については，結果を省いてある．同図より，つぎのことがわかる．各問題の規模はほぼ同じであるが，BAB では計算時間のばらつきがかなり大きい．対比的に，GA では計算時間はほぼ一定であり，またほとんどの問題に対して最適解ないしきわめて良い準最適解が得られている．

ここに掲げた結果はほんの一例にすぎないが，GA の特徴をよく見てとることができる．

（6）　その他の方法

筆者らによる上記の方法のほかもう1つ，山田・中野による方法を紹介しておこう[6]．ジョブショップ型

問題の解の最もオーソドックスな図式表現法として，ガント・チャート (Gantt chart) によるものがある．この表現に基づく山田らの方法では，仕事 i の第 j 作業 O_{ij} の完了時刻を x_{ij} と表わし，x_{ij} を (i, j) 成分とする行列 X によって解の遺伝子形を与える．そのとき，X が表わすスケジュールの総所要時間 S_X は

$$S_X=\max_{i,j}\{x_{ij}\} \qquad (3)$$

で与えられる．

選択演算は遺伝子形に関係しないので標準的なものを用いるが，交叉と変異の演算には上記の遺伝子形に適した独自のものを使用する．まず，土台になっている Giffer と Thompson のアクティブ・スケジュール生成法 (GT 法) について簡単に述べよう．

スケジュールの各段階において，開始時刻の決まっていない作業のうち先行関係などによって定まる処理開始可能な作業の集合 C（カットと呼ぶ）を求める．また，最早開始スケジュールを作成し，そこでの作業 O_{ij} の最早完了時刻を求めて e_{ij} とする．GT 法では，すべての作業のスケジュールが確定するまで，以下の 1°～3° を繰返す．

1° $e_{i^*j^*}=\min_{O_{ij}\in C}\{e_{ij}\}$ となる作業 $O_{i^*j^*}$ を求める．$O_{i^*j^*}$ と同じ機械で処理される集合 C に属する作業 O_{ij} のうちで，最早開始スケジュールにおいて $O_{i^*j^*}$ と処理時間が重なる作業の集合 G（コンフリクト集合と呼ぶ）を求める．

2° 集合 G から1つの作業 $O_{i_sj_s}$ を選び，その開始時刻を $e_{i_sj_s}$ に定める（スケジュールする）．

3° カット C およびスケジュールされていない作業 O_{ij} の最早完了時刻 e_{ij} を修正する．

さて，2個の親個体 (mom と dad と呼ぶ) から1個の子個体 (kid) を生成する交叉演算は，つぎのように実行される．なお，変異は交叉の 1° に含まれる形で実現される．

1° GT 法の 1° を実行し，カット C, 最早完了時刻 e_{ij} およびコンフリクト集合 G を求める．

2° G から1個の作業 $O_{i_sj_s}$ をつぎの手順で選び出す．

(a) 乱数 $\varepsilon\in[0, 1)$ を発生させ，ε の値をあらかじめ定めておいた突然変異確率 M と比較して，$\varepsilon<M$ ならば集合 G からランダムに $O_{i_sj_s}$ を選ぶ．

(b) $\varepsilon\geqq M$ ならば，mom と dad のいずれかを等確率（確率 1/2）で選ぶ．mom が選ばれたときは，G の中から mom が表わすスケジュールで完了時刻が最も早い作業を選び出し，それを $O_{i_sj_s}$ とする．dad が選ばれたときも同様である．

(c) $O_{i_sj_s}$ の開始時刻を $e_{i_sj_s}$ に定める（スケジュールする）．

3° カット C およびスケジュールされていない作業 O_{ij} の最早完了時刻 e_{ij} を修正する．

この手順 1°～3° をすべての作業がスケジュールされるまで繰返すことによって，新しい個体としての kid が誕生する．さらにもう一度同様の手順を繰返して2個の子 kid_1 と kid_2 を生成する．以上で，ひととおりの交叉演算が完了するわけである．

さらに，良い親個体が消滅するのを防ぐために，親 mom, dad と子 kid_1, kid_2 の4個体の中から実際に次世代に残す2個体を，つぎのようにして選ぶ．

(1) 4個体のうち，最大の適応度をもつものをまず mom' として残す（エリート戦略）．

(2) (1)で子のうちの1つが mom' となった場合には，もう一方の子を dad' として残す．(1)で親の1つが mom' となった場合には，適応度の大きな方の子を dad' として残す（いずれにしても両親がそのまま残ることはない）．

山田らは例題として Muth と Thompson のベンチマーク問題（6機械6仕事，10 機械10 仕事，5機械 20 仕事）および乱数によって作成した 20 機械 20 仕事問題に適用し，問題が大規模になるほど GA で最適解が得られる割合は減少するものの，全般的に良い結果が得られることを確認している[6]．

4. 初期収束と近傍モデル

GA における難点の1つとして，初期収束の問題がある．これは，選択演算のために個体集合の中で適応度の比較的高い個体が急速に支配的になり，集合中の多様性が早い世代で失われて，それ以後は適応度の向上が遅々として進まなくなる現象である．初期収束を緩和し集合の多様性を適切に保持する1つの手立てとして，筆者らは近傍モデルの概念を導入した[7),8)]．

このモデルでは，各個体の位置を定めてその近傍を

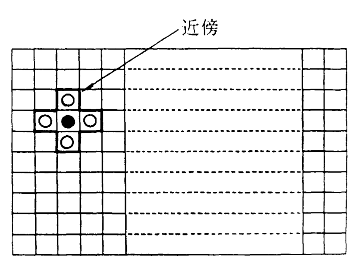


図 5 近傍モデルにおける個体配置と近傍

定義し，選択は近傍ごとに並列に実行される．具体的には，**図5**に示すように各個体を2次元の閉じた平面（トーラス）上のます目に配置し，各個体からのハミング距離が1以内にあるます目の集合をその近傍と定義する．この近傍モデルでは，集合中に比較的適応度の高い個体が存在しても，選択演算の局所化によってその影響は近傍間の重なりを通してしか波及しないので，急速に集合の多様性が失われることはない．

またこのモデルを用いれば，選択演算を並列に実行すること，したがってアルゴリズム全体を並列計算機上にインプリメントすることが容易になる．実際，筆者らは MIMD (Multi-Instruction Multi-Data) 型の計算機トランスピュータを使用し，アルゴリズムの高速化を実現した[7),8)]．

なお，Mührenbein や Schleuter らも近傍ごとに遺伝演算を並列的に実行する方法を提案しているが[9),10)]，筆者らの方法とは個体の配列法，近傍の定義および演算子の適用法が異なる．また最近，Davidor は ECO-GA と称する手法を提案しているが，これは現実の世界（環境）において生物の交流・交配と選択は近傍においてのみ発生するという見方に立つもので，筆者らの方法と通じる面が多い．いずれにしても，集合における多様性の維持，近傍モデル，そして計算の並列化などは，興味ある今後の研究課題であろう．

5. おわりに

本稿では，組合せ最適化問題の1つの典型としてジョブショップ型スケジューリング問題（JSP）を取り上げ，それに対する GA の適用過程と結果の例について，筆者らの方法を中心に解説した[5),7),8)]．JSP は，いわゆる GA easy な問題ではない．それゆえ，JSP の解法としていつでも GA が最適であると主張するつもりはない．ただ，GA easy でなくてもコーディングの方法などに工夫を加えれば，結構有用なツールになりうるところが興味深い．その意味で GA のポテンシャルを示す例であり，今後さまざまの分野でさまざまの問題に挑戦してみるきっかけとなれば幸いである．

逆に言うと，標準的あるいは万能ともいえる GA の手法が確立されていて，だれでもいつでもそれを機械的に適用すれば問題が解けるわけではない．GA はいまだそれほど成熟してはいないのである．だから，GA のポテンシャルを掘り起こし，その利点・欠点を見きわめるには今後とも多くの興味ある研究課題が残されている．

謝　辞

GA に関する良き共同研究者である玉置　久，喜多一（京大工学部）の両氏に感謝の意を表する．

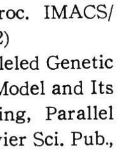
（1992年11月4日受付）

参　考　文　献

1) J. H. Holland: Adaptation in Natural and Artificial Systems, University of Michigan Press (1975)
2) D. E. Goldberg: Genetic Algorithms in Search, Optimization and Machine Learning, Addison-Wesley (1989)
3) 鍋島：スケジューリング理論，森北出版 (1974)
4) E. L. Lawler, J. K. Lenstra and A. H. G. Rinnooy Kan: Recent Development in Deterministic Sequencing and Scheduling: A Survey in Deterministic and Stochastic Scheduling, D. Reidel Pub. Co., 35/73 (1982)
5) 西川，玉置：ジョブショップ型スケジューリング問題に対する遺伝アルゴリズムの一構成法，計測自動制御学会論文集，**27**-5, 593/599 (1991)
6) T. Yamada and R. Nakano: A Genetic Algorithm Applicable to Large-Scale Job-Shop Problems, Parallel Problem Solving from Nature, 2, Elsevier Sci. Pub., 281/290 (1992)
7) H. Tamaki and Y. Nishikawa: Maintenance of Diversity in a Genetic Alogorithm and an Application to the Jobshop Scheduling, Proc. IMACS/SICE Int. Symp. on RM², 869/874 (1992)
8) H. Tamaki and Y. Nishikawa: A Paralleled Genetic Algorithm based on a Neighborhood Model and Its Application to the Jobshop Scheduling, Parallel Problem Solving from Nature, 2, Elsevier Sci. Pub., 573/582 (1992)
9) H. Mührenbein: Parallel Genetic Algorithms, Population Genetics and Combinatorial Optimization, Proc. 3rd Int. Conf. on Genetic Algorithms, Morgan Kaufmann, 416/421 (1989)
10) M. Gorges-Schleuter: ASPARAGOS: An Asynchronous Parallel Genetic Optimization Strategy, op. cit., 422/427 (1989)
11) Y. Davidor: A Naturally Occuring Niche and Species Phenomenon: The Model and First Results, Working Paper (1992)

［著　者　紹　介］

西川禕一君（正会員）



昭和8年3月18日生．昭和35年京都大学大学院電気工学専攻博士課程修了．京都大学工学部助手，助教授を経て，47年教授となり電気工学教室・計測制御工学講座を担任して現在に至る．UCLA 客員助教授，IIASA 研究員などを歴任．システム制御・最適化，自律分散システム，ニューラルネットワーク，エネルギーシステムなどの研究に従事．電気学会，電子情報通信学会などの会員．SICE 自律分散システム部会主査．