実践講座

スケジューリング問題の新解法

# スケジューリング問題に対する遺伝アルゴリズム

三宮　信夫

## 1. はじめに

この実践講座の第1回目で，ほとんどのスケジューリング問題は基本的なものでさえ解くのが困難（NP困難）であることが示された．一方，現実のシステムでは，例えばバッファや搬送車を介していくつかのサブシステムが結合しており，複雑な制約条件のある大規模な問題を扱わなければならない．そのような問題では，最適性は犠牲にしても実行可能解を容易に得ることが必要で，そのために本講座に後から登場するAIやシミュレーション技法が用いられる．

最適性を追求するならば，厳密には数えあげる手段しかなく，第2回目に登場した分枝限定法が唯一の方法であろう．しかしこの方法は，限定操作をいかに設定するかが重要なポイントであり，この設定が問題に大きく依存するという難点がある．そこで，メタヒューリスティックスと呼ばれるいくつかの近似アルゴリズムが，今回の遺伝アルゴリズム（Genetic Algorithm, 以下GAと略記する）を含めて本講座に登場することになる．近似アルゴリズムは，第1回目で指摘されているように，理論的な観点から何が解明されたのかと問われれば，現状では未だ確定的な回答はできない．しかし，近似アルゴリズムにはそれ以外に，問題への適用の容易さや適用可能性が問題のタイプにあまり依存しないといった別の観点からの評価も必要であろう．

GAは，Hollandらによって提唱され，自然界のシステムの適応過程を説明し，生物進化のメカニズムを模擬する人工モデルである[1,2]．これが提唱後20年近く経過した最近になって注目をあび，多方面に応用されるようになった[8]．本稿では，GAを最適化法としてスケジューリング問題へ適用する方法を述べる．

さんのみや　のぶお　京都工芸繊維大学 工芸学部
〒606 京都市左京区松ヶ崎御所海道町

## 2. GAの概要

GAでは生物の進化と遺伝のメカニズムに基づいて最適化アルゴリズムが構成されるので，用語もそれに由来したものが用いられる．

生物界の各個体 (individual) は，固有の染色体 (chromosome) をもち，染色体は遺伝子 (gene) の配列で構成されている．そこで，GAでは解くべき問題の解を個体に対応させて，次式のような記号列で表す．

$$A_1\ A_2\ \cdots\ A_i\ \cdots\ A_n \tag{1}$$

ここに，記号$A_i$は遺伝子に対応し，遺伝子が置かれている位置を遺伝子座 (locus) と呼ぶ．また，各遺伝子が取り得る値を対立遺伝子 (allele) と呼ぶ．

(1)のような記号列の表現を遺伝子型 (genotype) と呼ぶ．これに対して，その遺伝子によって定まる個体の性質を表現型 (phenotype) と呼ぶ．最適化問題では，解自身がこれにあたる．ただし，遺伝子型と表現型の間の変換は必ずしも1対1対応である必要はなく，同一の問題でも多くの個体表現あるいは変換の仕方が考えられる．

自然界における生物の進化過程では，ある世代 (generation) を形成している個体の集合，すなわち個体群 (population) を考え，その個体群の中で環境への適応度 (fitness) の高い個体が次世代に多く生き残るように淘汰 (selection または reproduction) される．また，交叉 (crossover) や突然変異 (mutation) が生じて，新しい個体が形成される．これを最適化問題を解く繰り返し過程に対応させると，次のようになる．すなわち，問題の解の候補を複数個選んでおき，第$t$回目の繰り返し計算における解集合を次式のように構成する．

$$X(t) = \{x_1(t),\ x_2(t),\ \cdots,\ x_M(t)\} \qquad (2)$$

ここに，$x_i(t)(i = 1,2,\cdots,M)$ は (1) の形に表現されており，$M$は個体群のサイズを表す．適応度は，その値が大きい方が優れており，正の数値で表される．具体的には，目的関数値を変換して適応度を定義する．

$M$個の個体の集合である個体群 $X(t)$ は，淘汰，交叉および突然変異という操作を受けて，次世代の個体群 $X(t+1)$ を生み出す．この操作を遺伝演算子 (genetic operator) と呼ぶ．淘汰は，個体群の中から不必要と思われる個体を除去する操作である．交叉は，2つの個体から有益な性質を取り出して，新しい個体を作り出す操作である．また突然変異は，1つの個体を変更して新しい個体を作る操作である．前回の分枝限定法に対応させれば，下界値テストや優越関係のような部分問題を終端させる操作が淘汰であり，一方部分問題に分割する手続きが交叉や突然変異に当たる．ただし，これらの対応は厳密な意味でなく，あいまい性，不完全性を含んでいるので，遺伝演算子はそれだけ問題固有の性質に強く影響を受けずに設定できる．

さらに，従来の最適化アルゴリズムでは，探索点は常に1個であったが，GAでは複数個の探索点を同時に扱う．その結果，探索点を大域的に分布させると，解の多様性が保持され局所解に落ち込むことが少なくなると考えられる．また，交叉という演算子の効果により，単に複数個の探索点を並列に求めるのとは異なり，ランダム法に比べて効率的に最適解を求めることが期待される．

GAの全体の手順は次の4つのステップからなる．

Step 1. 世代を $t=0$ とする．$M$個の個体をランダムに生成して，初期個体群 $X(0)$ を設定する．

Step 2. 各個体の適応度を計算し，適応度に依存した一定のルールで個体の淘汰を行う．すなわち，適応度の低いいくつかの個体は死滅し，その個数だけ適応度の高い個体が増殖する．

Step 3. 一定の確率で交叉や突然変異を行い，新しい個体（子）を生成する．子は，その生成に関与した古い個体（親）と置き換わる．

Step 4. $t < t^*$(最終世代) ならば，$t = t+1$ としてStep 2へ戻る．$t = t^*$ならば，そのときに得られている適応度最大の個体を問題の準最適解とする．

上に述べたアルゴリズムでは，個体群のサイズ $M$ は世代を通じて常に一定に保たれている．しかし，$M$ の値を可変にすることもできる．

最適化アルゴリズムの構成としては，生物の進化過程から何かヒントが得られればよく，忠実に生物の過程を反映させる必要はない．むしろ逆に，最適化法としては生物の過程では考えない効率性が問われるので，GAでは問題固有の性質を取り込んでアルゴリズムに工夫をこらすことが多い．したがって，GAの具体的な形には，全体の手順から遺伝演算子などの細部の構成に至るまで，実に多くの提案がある．

## 3. スケジューリング問題への適用

スケジューリング問題は一般に，資源（機械，人，搬送車など）を用いてジョブをどのような順序で処理させるかを決定する問題である．したがって，その解の基本空間は順序の集合で与えられることが多い．

具体的に考えるために，ある生産システムにおける製品投入順序問題を取り上げよう．いま，$N$種類の製品 $B_i(i = 1,2,\cdots,N)$ があり，各製品 $B_i$は $Q_i$個からなり，合計

$$Q = \sum_{i=1}^{N} Q_i \qquad (3)$$

の製品の処理を行わなければならないとする．また，問題は目的関数最小化の問題として定義されているとする．もしもすべての $i$ に対して $Q_i = 1$ ならば，この問題は巡回セールスマン問題 (Traveling Salesman Problem, TSP) とほぼ同様に扱うことができる．

### 3.1 個体表現

この問題に対する最も基本的な個体表現法は，製品投入順序列をそのまま個体表現とすることである．このとき (1) は，製品番号 $\{1,2,\cdots,N\}$ の重複を許す順列で，$n = Q$ である．GAでは個体表現が与えられたとき，問題の目的関数値が算出でき，かつ問題の制約条件を満たしているか否かの判定ができれば，計算は実行できる．

その場合，生産システムや問題の要求から生ずる制約条件をいかに扱うかが重要である．まず製品の個数に関する制約から，1つの個体に対立遺伝子 $i$ が $Q_i$ 個ずつ現われなければならないが，これは新しい個体を生成する交叉および突然変異規則を工夫することにより実現できる．しかし，問題の要求として納期制約のような時間的制約が付加されると，(1) の個体表現は実行不可能な解に対応する場合が生ずる．その場合，その個体は致死遺伝子をもつという．

致死遺伝子の出現を抑えるためには，遺伝子型から表現型に変換（デコーディング）するときに実行可能解を得るような工夫を試みるか，あるいは目的関数にペナルティ項を付加して評価を行う方法が通常採用される．GAでは個体群を扱うので，個体群中致死遺伝子をもつ個体が少しぐらい含まれていても，それは淘汰されて全体に悪影響を及ぼさない．しかし，制約の厳しい問題では，致死遺伝子が増えてアルゴリズムの効率が低下する．

### 3.2 淘汰規則および適応度の定義

適応度の高い個体が次世代に子孫を多く残すという考えが最適化に用いられるのは，良い解の近くにはもっと良い解が存在するのではないかと想定されるからである．その淘汰の規則としてよく用いられるのは，ルーレット方式と呼ばれるものである．これは，各個体をその適応度に比例する確率でランダムに選択して生き残らせるものである．しかしこのままでは，せっかく作られた良い個体が選ばれないことがあるので，適応度最大の個体を無条件で次世代に残すようにする（エリート保存方式）か，ランダムでなく確定的にルーレット方式を用いる場合がある．

淘汰の要点は，良い個体をどれだけ残すかを決めることであるが，それは適応度の定義の仕方にも依存する．一例として，個体$x_i$の適応度$F_i$を次式のように定める場合を考えよう[10,3]．

$$F_i = U \min_k f_k + \max_k f_k - f_i \qquad (4)$$

ここに，$f_i$は個体$x_i$に対する目的関数値である．

(4)において右辺第1項は，どの個体に対しても適応度を一定量だけかさ上げする効果を与える．このとき，パラメータ$U(>0)$の値を大きくするほど，適応度の小さい個体を生き残らせるようにすることができる．これより，$U$の値によって淘汰速度を調節することができる．$\min f_k$は最適な目的関数値の推定値と考えられるので，この値と$U$の積を第1項に用いることにより，$U$の値の問題依存性が少なくなる．

### 3.3 交叉規則

交叉は，GAにおいて最も重要な役割を果たす演算子である．すなわち，個体群よりランダムにペアを作り，定められた確率$P_c$で各ペア（親）から新しい個体（子）を生成する．個体表現に制約がない場合には，1点交叉，多点交叉，一様交叉などの比較的簡単な方法を採用することができる．しかし，TSPや製品投入順序問題では，個体に含まれる対立遺伝子の個数が定まっているので，この制約を破る交叉規則は望ましくない．そのために，従来主としてTSPを対象として，その制約を満足する交叉則が提案されていた[11]．ここでは，それらを製品投入順序問題に適用可能な形に拡張して紹介する[6,7]．

なお，以下では2人の親から2人の子を生成する方法において，子1を生成する手順を述べる．子2に対しては，1，2の立場を入れ換えればよい．

#### 1) Order crossover (OX)

OXは遺伝子の順序に注目して，各遺伝子の前後関係が保存されることを重視したものである．まず記号列に切れ目をランダムに二つ選ぶ．左の切れ目の左側にある親1の記号列を，親2の記号列に左から現われる順番に左から並びかえる．つぎに2つの切れ目の間は親1をコピーする．最後に右の切れ目の右側にある親1の記号列を，親2の記号列に右から現われる順番に右から並びかえる．以下にOXの例を示す．

2 1 2 | 3 1 | 3 2 1　　1 2 2 3 1 1 2 3
1 3 1 | 2 2 | 1 2 3　⟹　1 3 1 2 2 3 2 1

#### 2) Cycle crossover (CX)

CXは遺伝子座が保存されることを重視したものである．まず親1の一番左にある遺伝子座の記号をその位置に入れる．つぎに，この遺伝子座の親2の記号を，その記号が存在する親1の遺伝子座を左から探しその位置に入れる．以下同様に行って初めに選ばれた記号に戻ってくると，親からコピーされなかった遺伝子座のうち最も左にある遺伝子座にその位置の親2の記号を入れる．そして上記の手順を1，2の立場を逆にして行う．この時点でまだ子1に全ての記号がコピーされていない場合，さらに上記の手順を繰り返す．以下にCXの例を示す．

2 1 2 3 1 3 2 1　　2 1 1 3 2 3 2 1
1 3 1 2 2 1 2 3　⟹　1 3 2 2 1 1 2 3

#### 3) One point crossover (1X)

1Xは遺伝子の順序と遺伝子座の両方を保存するが，切れ目の位置に制約が加わる．まず記号列に切れ目をランダムに一つ選ぶ．切れ目の左側は親1をコピーし，右側は親2をコピーする．なお，この方法では交換する部分列において，各記号毎の個数が全て一致し

ていないと実行不可能な個体が生じる．したがって，この場合の切れ目は各記号毎の個数が全て一致するところからランダムに一つ選ぶものとする．以下に1Xの例を示す．

$$\begin{array}{l} 2\,1\,2\,3\,1 \,|\, 3\,2\,1 \\ 1\,3\,1\,2\,2 \,|\, 1\,2\,3 \end{array} \Longrightarrow \begin{array}{l} 2\,1\,2\,3\,1\,1\,2\,3 \\ 1\,3\,1\,2\,2\,3\,2\,1 \end{array}$$

4) Partially mapped crossover (PMX)

PMXは1Xで切れ目の位置に制約が加わらないようにしたものであるが，それだけ遺伝子の順序や遺伝子座が軽視されている．まず記号列に切れ目をランダムに二つ選ぶ．切れ目の間におけるある遺伝子座の親1の記号を$i$，親2の記号を$j$とする．いま，親1と同じ記号$i$が存在する親2の遺伝子座を探す．そのような親2の遺伝子座は$Q_i$個存在するが，そのうちの一つを適当に選んで記号$j$を入れる．また，二つの切れ目の間で$j$が入っていた遺伝子座に$i$を入れる．この操作によって記号$i$と$j$が置き換えられる．以上の操作を二つの切れ目の間のすべての遺伝子座に対して行った後でまだ空いている遺伝子座があれば，そこに親2をそのままコピーする．これより子1を得る．以下にPMXの例を示す．

$$\begin{array}{l} 2\,1\,2 \,|\, 3\,1 \,|\, 3\,2\,1 \\ 1\,3\,1 \,|\, 2\,2 \,|\, 1\,2\,3 \end{array} \Longrightarrow \begin{array}{l} 1\,2\,1\,3\,1\,2\,2\,3 \\ 2\,1\,3\,2\,2\,3\,1\,1 \end{array}$$

5) Edge recombination (ER)

ERは遺伝子の（特に隣同士の）順序をOXより重視したものであるが，遺伝子座はOXより軽視されている．まず，各記号の右隣がどの記号になっているかを調べる．それを表にしたものをEdge tableと呼ぶ．つぎに，親1の一番左の記号$i$を子1にコピーする．記号$i$の右隣にはEdge tableの$i$の欄からランダムに一つの記号$j$を選んでそれを入れ，Edge tableの$i$の欄から一つだけ記号$j$を消去する．以上の操作を繰り返して子1をつくるが，$Q_i$個の記号$i$がすべて子1にコピーされた場合は，記号$i$に関するデータをEdge tableからすべて消去する．また，途中でEdge tableに何も残ってなくて右隣に記号を入れられなくなった場合は，親1をそのまま子1とする．以下にERの例を示す．なお，Edge tableは，{1：2 3 3 2 2，2：1 3 1 2 1 3，3：1 2 1}である．

$$\begin{array}{l} 2\,1\,2\,3\,1\,3\,2\,1 \\ 1\,3\,1\,2\,2\,1\,2\,3 \end{array} \Longrightarrow \begin{array}{l} 2\,1\,2\,1\,3\,1\,2\,3 \\ 1\,2\,1\,2\,2\,3\,1\,3 \end{array}$$

6) Position based crossover (PBX)

PBXは遺伝子座を保存するものであるが，CXほど重視せずにそのかわり順序もある程度保存する．各遺伝子座毎に確率0.5で親1をその位置にコピーする．残りの遺伝子座については，記号列の真ん中より左側は親1の記号列を親2の記号列に左から現われる順番に左から並び変える．また，真ん中より右側は親1の記号列を親2の記号列に右から現われる順番に右から並び変える．以下にPBXの例を示す．

$$\begin{array}{l} 2\,\underline{1}\,2\,3\,\underline{1}\,\underline{3}\,2\,1 \\ 1\,3\,1\,2\,2\,1\,2\,3 \end{array} \Longrightarrow \begin{array}{l} 3\,\underline{1}\,2\,\underline{2}\,\underline{1}\,\underline{3}\,1\,2 \\ 2\,3\,1\,1\,2\,1\,3\,2 \end{array}$$

### 3.4 突然変異規則

交叉では，2個体の親に依存する限られた範囲の子孫しか生じない．突然変異はこれを補う役割を演じ，個体群の多様性を維持する働きをする．

具体的には，個体群の各個体に対して定められた確率$P_m$で，ランダムに選ばれた2個の遺伝子座の遺伝子を交換する．突然変異としてはこの他にも，個体表現の一部分が同じ個体の他の部分に位置を変えるもの（転座）や個体上で部分的に遺伝子の配列順序を逆転させるもの（逆位あるいは反転）がある．

### 3.5 パラメータの決定

個体表現，適応度関数および遺伝演算子を決定するとGAの枠組ができ上がるが，その中にはいくつかのパラメータが含まれている．これまでの説明から，次のようなものが挙げられる．

$M$ ： 個体群のサイズ
$t^*$ ： 最終世代
$U$ ： 淘汰速度調節パラメータ
$P_c$ ： 交叉確率
$P_m$ ： 突然変異確率

これらの値は，アルゴリズムの収束状況および個体群の多様性を考慮して試行錯誤的に決められる．

## 4. 変形フローショップ問題

第2回目の講座で3機械フローショップ問題を取り上げたので，ここでは，その問題を複雑にした変形フローショップ問題[4,5]を考える．なお，ジョブショップ問題については，すでに解説記事[9,12]があるのでここでは触れない．

図1に示されるフローショップ型工程で，$N$種類の製品$B_i$が機械$M_1$，$M_2$，…，$M_L$の順に加工される問題を考える．製品$B_i$の生産量は$Q_i$で，納期は$d_i$であるとする．総生産量は(3)の$Q$である．

製品は加工前製品置き場$Y_1$→機械$M_1$→機械$M_2$→…→機械$M_L$→加工後製品置き場$Y_2$の順に流れていき，この製品の運搬は$Y_1$から出発する1台の搬送車が行っている．搬送車は一定速度で巡回しているが，各機械に到着したときまだ製品が加工中ならば，加工完了までその場で待機するものとする．ただし，搬送車の第$n$周目$(1 \leq n \leq L)$は，機械$M_n$へ製品を運んだ後は空のまま動くものとする．

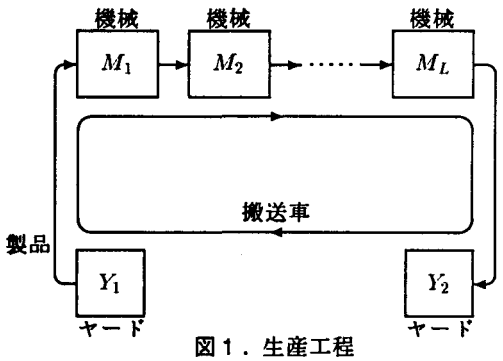

図1．生産工程

この問題では，搬送車の運行により後工程の影響が前工程にも及ぶことになるので，その点で通常のフローショップ問題とは異なる特徴をもつ．また，この問題の目的関数としてつぎの4種類を考える．

① 生産比率一定化
② 総生産時間最小化
③ 納期ずれ時間最小化
④ 納期遅れ時間最小化

計算例として，製品の種類$N$，工程数$L$および生産量$Q$を種々に設定した各事例において，機械による製品の生産時間や搬送車の移動時間をランダムに与えて50例ずつ例題を作成した．

いま，簡単のため納期制約を無視して，①と②の目的関数を相加的に両方考慮した問題を考える[4,5]．このとき，6種類の交叉規則を用いた場合のGAによる計算結果を表1に示す[6]．同表において，値は50例の平均目的関数値であり，括弧内の数値は各$Q$でその交叉規則が何番目に良かったかを示している．なお，遺伝パラメータは，$M = 40$, $t^* = 50$, $U = 0.01$, $P_c = 1.0$および$P_m = 0.01$とした．

これより，以下のことが確認できた．

(a) 少しの例外を除いて，ほぼOX，CX，1X，ER，PMX，PBXの順に良い．

(b) $Q = 10$では染色体長が短いので$Q \geq 20$の順位と少し異なる．$Q \geq 20$の順位はほぼ一定である．

(c) $L$の値はほとんど順位には影響を及ぼさない．

以上より，この問題に最も良い交叉規則はOXであるが，それではなぜこの交叉規則が良かったのであろうか．以下でその理由をいくつか検討してみる．まず，本問題は製品投入順序問題であるから「順序」がポイントとなり，個体表現も「順序」を重視したものとなっている．一方，OXも親の遺伝子の「順序(前後関係)」を保存するものとなっており，この点でOXは問題と個体表現の両方に適合していると考えられる．

表1．目的関数値の比較($N = 6$)

(a) $L = 2$

| $Q$ | 10 | 20 | 30 | 40 |
|---|---|---|---|---|
| OX | 166.5(1) | 328.9(1) | 516.6(1) | 708.9(1) |
| CX | 167.0(3) | 343.8(2) | 548.7(2) | 790.0(2) |
| 1X | 169.6(4) | 354.2(3) | 578.4(3) | 840.6(3) |
| PMX | 166.9(2) | 397.2(4) | 708.8(5) | 1053.3(4) |
| ER | 187.9(6) | 437.7(5) | 699.6(4) | 1080.8(5) |
| PBX | 184.7(5) | 494.5(6) | 1019.7(6) | 1980.4(6) |

(b) $L = 4$

| $Q$ | 10 | 20 | 30 | 40 |
|---|---|---|---|---|
| OX | 361.8(1) | 690.1(1) | 1017.8(1) | 1334.1(1) |
| CX | 361.8(1) | 691.6(2) | 1039.3(2) | 1391.4(2) |
| 1X | 364.5(3) | 721.9(3) | 1123.5(3) | 1494.0(3) |
| PMX | 365.5(4) | 801.6(5) | 1338.5(5) | 1943.8(5) |
| ER | 369.6(5) | 786.9(4) | 1312.2(4) | 1902.9(4) |
| PBX | 386.8(6) | 939.7(6) | 1772.7(6) | 2492.8(6) |

またOXは，親の遺伝子の「位置(遺伝子座)」については二つの切れ目の間は必ず保存するが，切れ目の外側については必ずしも保存するとは限らない．しかし，実際に計算してみるとOXの「位置」の保存率は80％程度あり，保存率が100％であるCXや1Xには及ばないもののかなり保存率は高いといえる[4]．

さらに，本問題では投入順序の最初と最後付近に良い部分列を組めば，比較的良い解が得られやすいという特徴がある．すなわち，良い投入順序列の真ん中付近を並び変えてできた投入順序列群の平均目的関数値をX，また良い投入順序列の両端部付近を並び変えてできた投入順序列群の平均目的関数値をYとする

と，それらは表2のようになった[5,6]．ここで，$Q=10$では良い投入順序列として列挙法で求めた最適投入順序列を選び，並び変える部分列は6個の製品とした．また，$Q \geq 20$では良い投入順序列としてGAで求めた投入順序列を選び，並び変える部分列は8個の製品とした．同表より，投入順序列の両端部を固定したXの方がよいことがわかる．OXは染色体の両端部からコピーしていくので両端部は親の形質をそのまま受け継ぎやすく，これが良い結果を得た原因の一つと考えられる．

表2．部分列が目的関数値に及ぼす影響
($N=6$, $L=2$)

| $Q$ | 10 | 20 | 30 | 40 |
|---|---|---|---|---|
| X | 205.3 | 384.4 | 532.7 | 691.9 |
| Y | 233.8 | 572.1 | 880.7 | 1213.0 |

つぎに，目的関数を変更した場合の結果[7]を要約すると，交叉規則としては目的関数の形によらずOX, CX, ERが良くPBXは良くなかった．このように，問題の主要な性質（例えばフローショップであること）をうまく取り入れてGAを構成するならば，問題の条件を変更しても，そのGAを用いて比較的良い解が得られることがわかる．

最後に，この問題に納期制約を陽に考慮した場合には，3.1で述べたように，ⓐデコーディングの改良かⓑペナルティ関数の導入の2つの方法がある．解の精度と計算時間の点からいえば，ⓑの方が良かった．一般的に，ⓐはアイデア次第であり，問題に対応して良いアイデアが導入されれば一定の改良が期待できる．一方，ⓑはどのような問題にも適用でき，計算時間も多くならずに済むが，ペナルティパラメータを調節しなければならないという難点がある．

## 5. おわりに

GAをスケジューリング問題に適用する試みは多いが，実際にどれだけ使われたかについてまだあまり報告されていない．実用化への努力は，まずGA向きの問題に対して向けられるべきであろう．

一方，GAを計算時間や精度の点のみで他の最適化手法と比較することは適当でない．GAは解集合に操作を加えるという点をむしろいかして，それぞれのスケジューリング問題固有の性質や構造的特徴をGAの解過程から見出し，そこから新しい解法を生み出すことができれば良いと思われる．あたかも地球上の種々の生物がいかに栄え，滅んでいったかを解明するように．

最後に，GAの共同研究者である京都工芸繊維大学大学院生飯間等君に感謝の意を表する．

## 参考文献

[1] Goldberg, D. E.: Genetic Algorithms in Search, Optimization and Machine Learning, Addison-Wesley (1989).

[2] Holland, J. H.: Adaptation in Natural and Artificial Systems, University of Michigan Press (1975).

[3] 飯間，三宮：遺伝アルゴリズムによる製品投入順序問題の解法，計測自動制御学会論文集，Vol.28, No.11, pp.1337-1334 (1992).

[4] 飯間，三宮：遺伝アルゴリズムを用いた変形フローショップ問題の解法，システム制御情報学会論文誌，Vol.6, No.10, pp.437-445 (1993).

[5] 飯間，三宮：変形フローショップスケジュール問題における遺伝アルゴリズムの個体の考察，電気学会論文誌C, Vol.113, No.10, pp.879-885 (1993).

[6] 飯間，三宮：遺伝アルゴリズムを用いた変形フローショップ型加工工程のスケジューリング，第3回FAN Symposium講演論文集，pp.215-220 (1993).

[7] 飯間，三宮：スケジューリングの条件変更に対する遺伝アルゴリズムの挙動，平成6年電気学会電子・情報・システム部門大会講演論文集，pp.197-200 (1993).

[8] 北野編：遺伝的アルゴリズム，産業図書 (1993).

[9] 西川：GAのスケジューリング問題への応用，計測と制御，Vol.32, No.1, pp.46-51 (1993)

[10] 三宮:遺伝アルゴリズムによる最適化問題の解法，第36回システム制御情報学会研究発表講演会講演論文集，pp.9-18 (1992).

[11] Starkweather, T., McDaniel, S., Mathias, K., Whitley, D. and Whitley, C.: A Comparison of Genetic Sequencing Operators, Proc. 4th Intern. Conf. on Genetic Algorithms, pp.69-76 (1991).

[12] 山田，中野:遺伝アルゴリズムとスケジューリング問題，システム/制御/情報，Vol.37, No.8, pp.484-489 (1993).