



HOKKAIDO UNIVERSITY

Title	スケジューリング問題へのジェネティックアルゴリズムの適用に関する実験的考察
Author(s)	皆川, 雅章; 嘉数, 侑昇
Citation	北海道大學工学部研究報告, 152, 67-77
Issue Date	1990-09-26
Doc URL	https://hdl.handle.net/2115/42250
Type	departmental bulletin paper
File Information	152_67-78.pdf



スケジューリング問題へのジェネティックアルゴリズムの 適用に関する実験的考察

皆川 雅章* 嘉数 侑昇*

(平成 2 年 6 月 30 日受理)

An Experimental Study on the Application of Genetic Algorithms to a Scheduling Problem

Masaaki MINAGAWA and Yukinori KAKAZU

(Received June 30, 1990)

Abstract

This paper intends to explore a new scheduling algorithm for the Job-Shop Scheduling problem and to provide some analyses to obtained experimental results. To guide scheduling solution search effectively and to obtain near-optimal solutions, we adopt Genetic Algorithms(GAs) which mimic natural evolution process.

Evaluation functions for the search are weighted using bit strings and these strings are manipulated by genetic operators such as reproduction, crossover and mutation. So as to follow the change of input job characteristics and keep obtaining better scheduling performance, the weights are automatically tuned by the algorithm. Based on the experimental results, some useful conclusions are drawm.

1. 緒 言

スケジューリング問題は作業群を要求資源の時間軸上に割り当てる問題として捉えられ、作業間での要求資源の競合解消を含む、困難な組み合わせ最適化問題の 1 つである。本報告ではこのような問題の中で、機械工場におけるジョブフロアスケジューリング問題の 1 つであるジョブショップスケジューリング問題を対象としてその解決のためのアプローチ法及び実験結果を示す。この問題は、作業遂行のための機械を要求資源とし、もう 1 つの代表的なジョブフロアスケジューリング問題であるフローショップスケジューリング問題とは異なり、工場内での作業の流れ方を特に仮定していない。即ち、ライン編成的なフローショップの場合に比べて解空間内の組み合わせ数が増加し、問題の困難さが増す。

従来この問題に対して、OR 的アプローチ、AI 的アプローチが行なわれてきたが、前者の場合には機械台数が 2 台までの場合に関してのみ最適解が得られているが、それ以上の場合については最適解を得ることが困難であることが知られている。また、後者の場合には、現実的な解をもとめるためにヒューリスティックな手法によるアプローチが行なわれ、対象問題を限定した場

* 精密工学科 精密機器学第一講座

合においては有効であるが、適用する手法の問題空間への依存性の解消という課題が残る。

本報告では、後者のアプローチの問題点の解決が図られている。即ち、スケジューリング問題の解探索のための評価関数を評価項目の線形結合の形式で表現し、結合の重み付け係数を対象問題に対して自動的にチューニングを行ない、問題に応じて評価関数を変更するメカニズムの実現を試みる。即ち、入力情報の特性の変化、スケジュールの評価基準の変更に対しても適応可能な探索手法を探る。この多目的型スケジューリングのメカニズムの実現にジェネティックアルゴリズムを適用する。

ジェネティックアルゴリズムは自然界の進化の過程を模倣した解探索手法で、近最適な解を探索する。対象問題の解あるいは戦略は個体と呼ばれるストリングに数字、アルファベット等を用いてコーディングされ、さらにそこから集団と呼ばれるストリング群が構成される。各ストリングはそれを用いて行なわれた探索の結果の評価値の良さに応じて淘汰される。この過程は再生(reproduction)と呼ばれ、良い評価値のストリングは存続し、悪い値のストリングは消滅する。淘汰の結果生き残ったストリング群には、さらに、進化のために乗り換え(crossover)、突然変異(mutation)を発生させ、さらに良い性質をつくりだすことが試みられる。乗り換えは2つのストリング間におけるランダムな位置での部分ストリングの交換、突然変異は1つのストリング中のランダムな位置での値の変化である。例えば、0と1の2値でコーディングされたストリング

$$S_1 = 0010100$$

$$S_2 = 1100011$$

$$S_3 = 0000111$$

の S_1 と S_2 において、第3の区間(ビット3と4との間)で乗り換えが発生すると(乗り換え位置を/で示す)、

$$S_1' = 001 / 0011$$

$$S_2' = 110 / 0100$$

となる。また、 S_3 の第2の位置に突然変異(この場合には値の反転)が発生すると

$$S_3' = 0100111$$

となる。本報告においては評価関数の重み付けの値はこのような2値ベクトルによって表現され(10進数の重み付け値の2進数表現)、これに対して再生、乗り換え、突然変異のジェネティックオペレーション群を集団が均一になるまで適用し続ける。均一な集団では全てのストリングが同一になる。これらの一連のオペレーションが実行されるごとに、集団は淘汰されて進化し、その世代(generation)が移り変わる。

本文中の例では、スケジューリングのための評価関数の評価項目に作業時間、残作業時間、残作業数、割当待ち回数、そして納期余裕の5つを用い、各々の重み付けに対して4ビットの2値ベクトルを割当て、長さ20(4ビット×5項目)のストリングを構成している。また、実験においては最大完了時刻をスケジュールの評価基準として採用している。

2. 対象問題

本報告において対象とするのは、機械の時間軸上に作業を割り当てるジョブショップスケジューリング問題である。この問題は一般に次のように記述される。

$$n/m/G/F_{\max}$$

ここに n は仕事数、 m は機械の台数、 G は仕事の到着過程が一般または任意であること、 $F_{\max}$

はスケジュール評価の判定基準で、最後の仕事が出来ただけ早く完了すべきであることを表している。

さらに、対象問題は単純ジョブショップ過程を構成していると仮定する。作業は最小の遂行単位で、仕事は技術的順序に従って構成される作業群から構成され、同一仕事内の作業間に分岐、合流は存在しないものとする。

3. スケジューリングの方法

3.1 評価関数

ここではスケジューリングの方法を以下のように設定する。即ち、機械に割当てられる作業の好ましさを評価関数を用いて定量化し逐次、作業を機械に割り当てる。ここではこの評価関数を、以下のように評価項目の線形結合として表すものとする。

$$F = \sum_{i=1}^{N_{ev}} a_i \cdot f_i$$

ここに、 a_i は重み付け係数、 f_i は評価項目、 N_{ev} は評価項目数である。以下の例では、評価項目数を 5 とし、各評価項目には次のものを使用する

- 1) 作業時間：作業時間は個々の作業を完了するために必要な時間である。ここでは仕事 i の作業 j の作業時間を p_{ij} と表すことにする。また、全仕事数を $NJOB$ 、仕事 i が持つ作業の総数を $NTASK_i$ とする。
- 2) 残作業時間：各仕事における割付未完了の作業の作業時間の和を指す。仕事 i において完了済みの作業の個数を $NCOMP_i$ とすると、残作業時間 $RTIME_i$ は

$$RTIME_i = \sum_{j=NCOMP_i+1}^{NTASK_i} p_{ij} \quad (NCOMP_i < NTASK_i)$$

- 3) 残作業数：各仕事における未完了作業の作業個数。残作業数 $RNUM_i$ は

$$RNUM_i = NTASK_i - NCOMP_i$$

- 4) 待ち回数：各仕事が機械の割当てを待った回数（他の仕事が優先された回数）。
- 5) 納期余裕：各仕事の納期までの余裕時間。仕事 i において作業 $(j-1)$ まで完了しているならば、その時点から見た納期余裕 E_i は

$$E_i = \max(0 - L_i)$$

$$L_i = C_{i,j-1} - d_i$$

d_i : 仕事 i の納期

$C_{i,j-1}$: 仕事 i の作業 $(j-1)$ の完了時刻

3.2 割当決定手続き

未割付作業群に対して仕事単位での評価値を計算し、それに基づいて逐次、割当決定を行なう。ここで、作業群の集合 $TSET$ を

$$TSET = \{ TASK_{ij} \}$$

$TASK_{ij}$: 仕事 i の作業 j

$$(i = 1, 2, \dots, NJOB, j = 1, 2, \dots, NTASK_i)$$

と表し、手続きは以下の通り

```

while TSET  $\neq$   $\phi$  do
  FSET :=  $\phi$ 
  for i=1 to NJOB
    if NCOMPi < NTASKi then
      Fi :=  $\sum_j a_j \cdot f_j$ 
      FSET := FSET  $\cup$  Fi
  ;
  I := selectmax (FSET)
  assign (TASKI,NCOMPI+1)
  TSET = TSET - TASKI,NCOMPI+1
;

```

ここで、selectmax は最大の F を持つ仕事番号 I を選択する関数、assign は選ばれた作業を当該機械に割り当てる関数である。

4. ジェネティックアルゴリズム

上記のスケジューリング方法において述べた評価関数の線形結合の重み付けを決定するためにジェネティックアルゴリズムを適用する。ここでは各係数が 0 から 15 の範囲の値を取るものとして、次のように係数群を 2 値コーディング（即ち、2 進 4 桁表現）して一本のストリングで表現する。第 j 評価項目の係数 a_j は、

$$a_j = s_1 \ s_2 \ s_3 \ s_4 \ (s_k = 1/0)$$

よって 5 個の評価項目の係数をまとめて表現すると

$$\begin{aligned} S &= a_1 \ a_2 \ a_3 \ a_4 \ a_5 \\ &= s_1 s_2 \cdots s_{20} \end{aligned}$$

このストリングを用いて世代 T における集団 A (T) を構成する。

$$A(T) = \{S_i\}$$

$$T = 0, 1, 2, \dots, \quad i = 1, 2, \dots, \text{PSIZE}$$

ここで PSIZE は集団が持つストリング数である。

本報告で用いているジェネティックオペレータは再生、乗り換え、そして突然変異の 3 つである。全体の処理流れを図 1 に示し、各オペレータの詳細を以下に記す。

4.1 再生オペレータ

再生は各ストリングの評価値 V_i をもとにストリング S_i の存在確率 P_i を計算した値をもとに行なわれる。

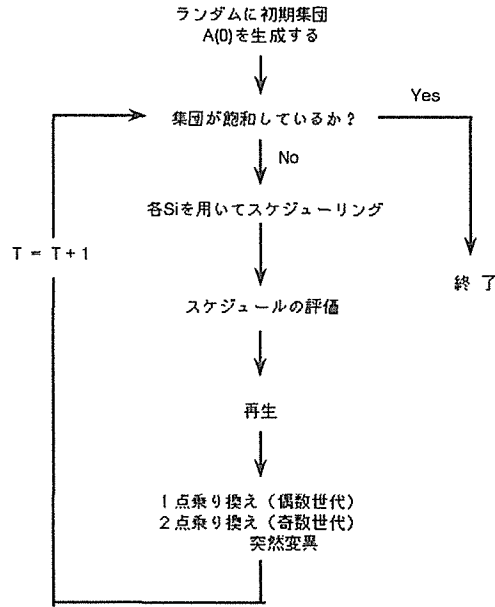


図1 ジェネティックオペレータ適用の流れ

各ストリングに対応した重み付けを持つ評価関数（1世代中に PSIZE 個ある）を用いて行なったスケジューリングの結果は最大完了時刻を用いて評価する。ここでは V_i を

$$V_i = (C_{\max} - C_i)^2$$

$$C_{\max} = \max_i [C_i]$$

C_i : ストリング S_i によって計算された最大完了時刻

とし、完了時刻の小ささを2乗の重み付けで評価している。 V_i は各 S_i によって計算した最大完了時刻が小さいほど大きな値となり、存在確率を大きくする。 P_i の計算式は次の通りである。つまり、集団中において各ストリングが持つ評価値の割合から存在確率が決まる。

$$P_i = \frac{V_i}{\sum_{j=1}^{PSIZE} V_j}$$

これより、次世代におけるストリング S_i の生存数 $NSUR_i$ は、

$$NSUR_i = PSIZE \times P_i$$

となる。この値に応じて各ストリングは選択され、交配プールと呼ばれる作業領域に格納され、プール中のストリング群を対象に乗り換え、突然変異が行なわれる。

4.2 乗り換え

乗り換えでは、部分ストリングの交換方法が異なる2種類のオペレーションが導入されている。即ち1点乗り換えと、2点乗り換えの2つで、前者は偶数世代、後者は奇数世代で交互に適用さ

れている。このような乗り換えを行なうことにより、解探索の幅が広がる可能性を大きくする。また、両乗り換えともに、各ストリングに対する乗り換えオペレーションは1世代に1度だけに限定している。

1) 1点乗り換え

以下の手続きにより、ストリング中の1箇所で切られた部分ストリングの交換が行なわれる。

```

str1 = rnd (1, PSIZE)
while str1 = str2 do
    str2 = rnd (1, PSIZE)
;
site = rnd (1, STLEN-1)
exc (str1, str2, site, STLEN-1)

```

rnd (*a*, *b*) : 区間 [*a*, *b*] で乱数を発生する関数

exc (*a*, *b*, *c*, *d*) : ストリング *a*, *b* 間で位置 *c*, *d* 間の部分ストリングを交換する関数。

STLEN : ストリング長

2) 2点乗り換え

上記の1点乗り換え手続きの後半を以下の手続きで置き換えて、2つの位置で切った部分ストリングの交換を行なう

```

site1 = rnd (1, STLEN-1)
while site1 = site2 do
    site2 = rnd (1, STLEN-1)
;
exc (str1, str2, site1, site2)

```

4.3 突然変異

以下の手続きにより、ストリング中の値を変化させて突然変異を行なわせる。

```

str = rnd (1, PSIZE)
site = rnd (1, STLEN)
invval (str, site)

```

invval (*a*, *b*) : 第 *a* ストリングの第 *b* 番目の位置の値を反転する関数

突然変異を導入することにより、集団が均一化してきたときに新たな探索領域を試すことが出来る。

5. 実 験

上述の問題設定及び方法論に基づいて行なった実験の結果を以下に示す。集団中のストリング数 PSIZE は 40 である。1 世代の間に乗り換えが行なわれるストリング数は全体の 80%，突然変異は 1 世代あたり約 0.1% である。

5.1 最大完了時刻の最小値の変化

3種類のデータ（図2：以下データ1，2，3とする）の各々に対して世代ごとの最大完了時刻の最小値の推移を図3に示す。いずれのデータの場合にも

仕事数(n) = 15

機械数(m) = 15

作業数 = 141

で、作業時間の分布のみが異なっている。図2中、横軸は作業時間、縦軸はその作業時間を持つ作業の個数である。図3における値は、各世代において、集団を代表する最も良いストリングによって得た値であることを意味する。図中、横軸は世代数(Generation)、縦軸は世代ごとの最大完了時刻の最小値である。集団の初期値は4種類の乱数(図中 RN - 1000,...,RN-10000：数字はコンパイラの乱数系列指定パラメータ)を発生させて設定している。

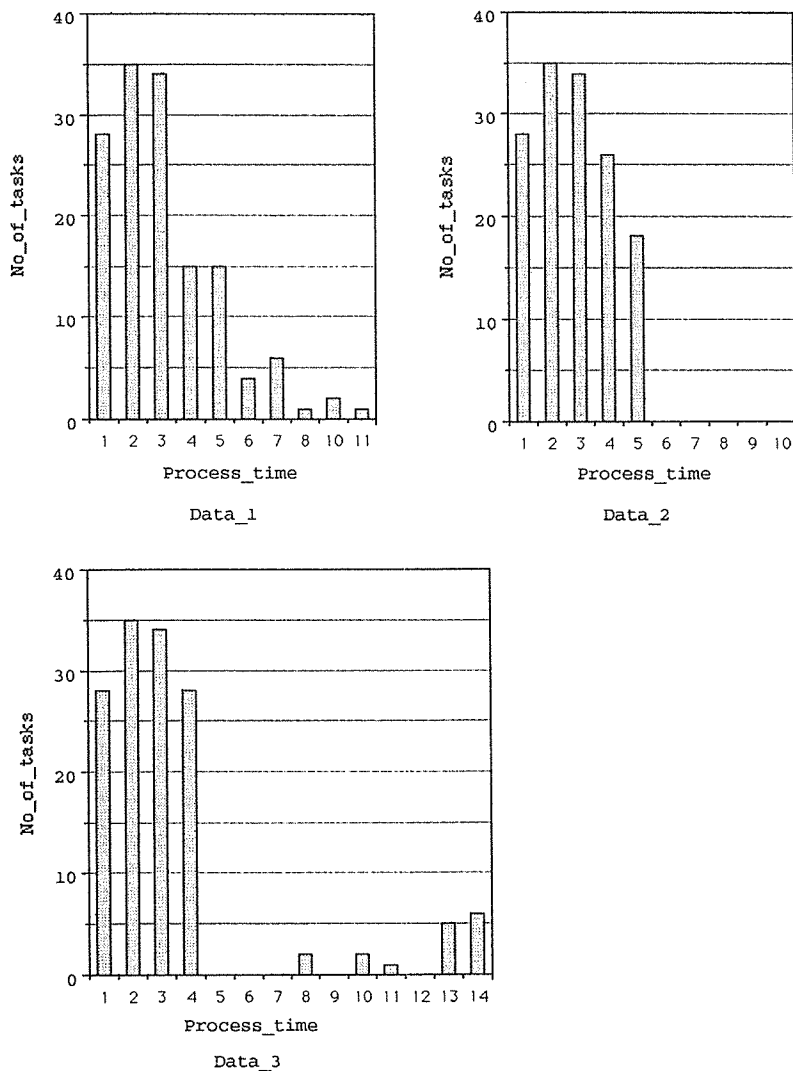


図2 入力データ（作業時間分布）

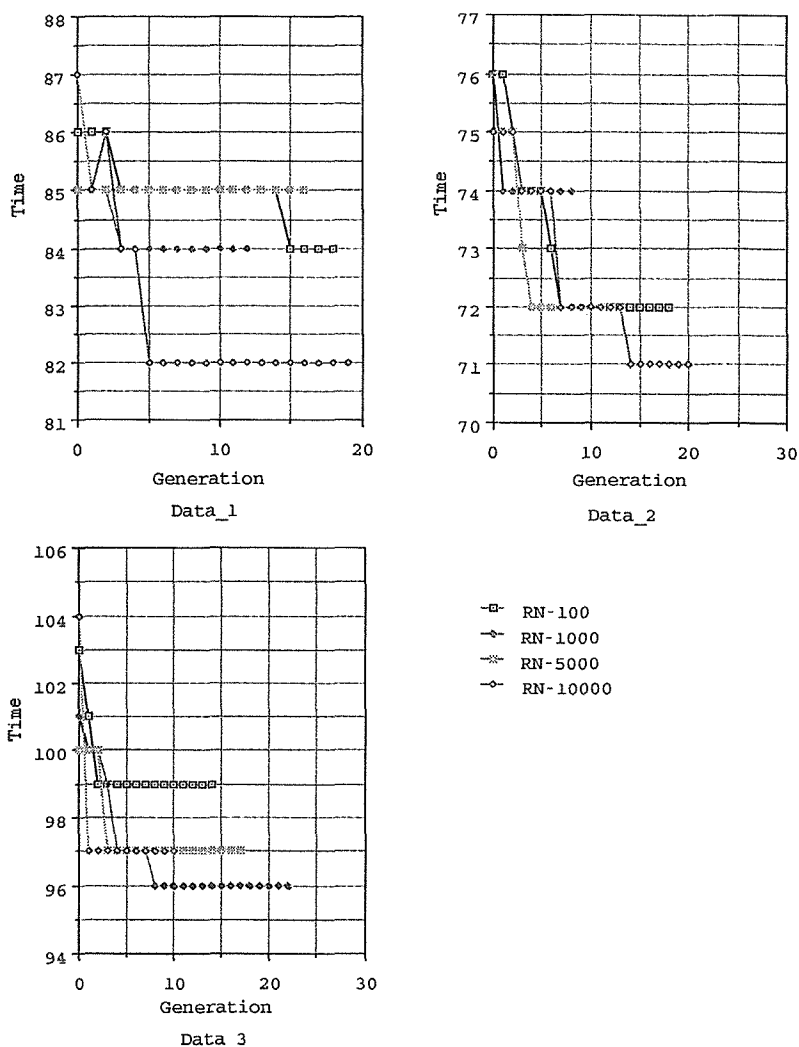


図3 最大完了時刻の最小値の推移

5.2 集団の収束

最良値の変化のみでなく、集団全体としての値の変化、即ち収束性を見るために、データ1、2に対する各世代における最大完了時刻の最小値（最良値）、最大値（最悪値）、そして平均値の推移を図4に示す。図中では各々、Best, Worst, Averageで示されている。

5.3 最良ストリング個数の変化

図4の場合に、再生によって各世代にどれだけの最良ストリングが残るのかを見るために、その個数の推移を図5に示す。St_の後に付けられた数字は最良値を示す。図中、縦軸は当該最良値を持つストリング数である。

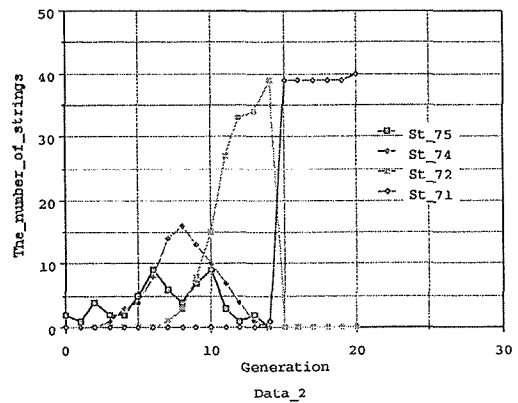
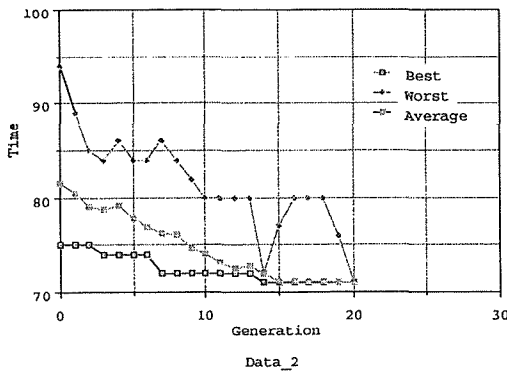
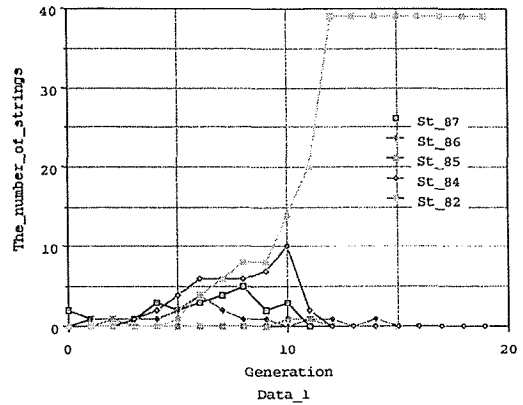
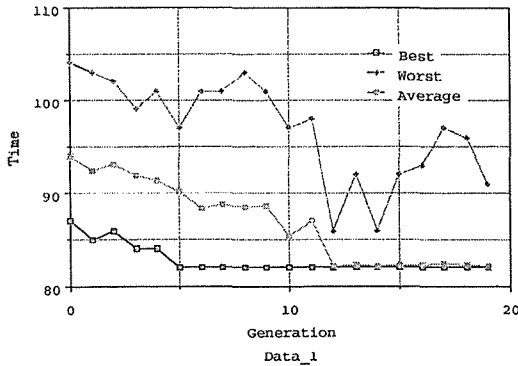


図4 最大完了時刻の最小値、最大値、平均値の推移

図5 最良ストリングの個数分布の推移

5.4 ランダムな方法との比較

GAによる方法の性能を見るために、ここではランダムにストリングを発生させ続けた場合と、本方法による場合との結果の比較を行ない、その結果を図6に示す。ランダムなストリング発生回数は3,600回(30ストリング/世代×30世代×4回)で、その際の最良値の発生個数の分布を同図上段に示している。下段の図は、ランダムに900回発生させた場合と、GAによる場合との値の推移の比較を示したものである。

5.5 最良ストリング群の利用

前述の3種類のデータに対するスケジューリングの結果得た最良ストリング群を初期集団中に入れ込んで他のデータ(データ4)に対して実行した場合の例を図7に示す。このデータにおいては

仕事数(n)=27

機械数(m)=15

作業数=279

である。図中、Best_stringで示されているのが最良ストリング群を用いた場合である。

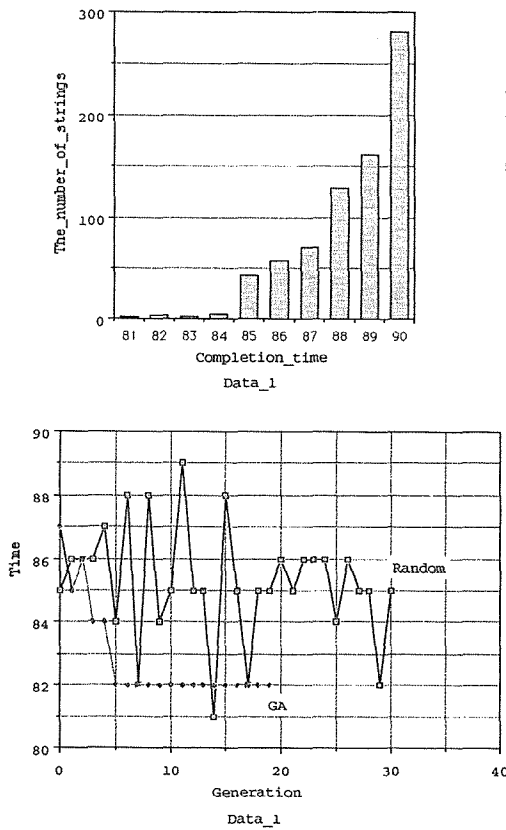


図6 ランダムなストリング発生方法との比較

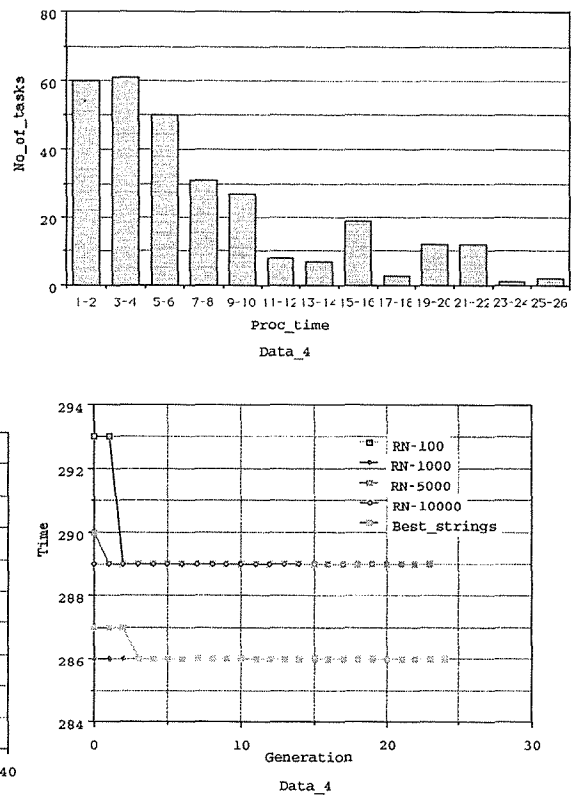


図7 最良ストリング群の初期集団への導入結果

6. 考 察

上記の実験結果をもとに、本アプローチ方法によるスケジューリングに関して以下のことが推測される。

- 1) スケジューリングの解はストリング集団の飽和に従って収束する(図3, 4)。悪い解の挙動は収束に影響していない。集団の再生時に性能の悪いストリングが消失することが1つの要因であると考えられる。
- 2) 解の収束のしかたは、初期集団の発生方法(乱数の選択)に依存する(図3)。常にうまく収束し、良い解に達するようにするにはヒューリスティックに初期集団を発生させることが必要であろう。
- 3) ある世代における最良ストリングは、さらに良いストリングの出現(発生)に従って消失する(図5)。これも1)の場合と同様に再生メカニズムによるものであると考えられる。
- 4) ランダムなストリング発生と比較して、少ない試行回数で近最適な解が得られる可能性がある(図6)。ただし、収束性を考慮しなければ、ランダムな方法では偶然に良い解が得られることもある。
- 5) 最良ストリング群を初期集団に入れ込むことにより、比較的速やかに良い解に収束する可能性がある(図7)。これは2)で述べたヒューリスティックな初期集団生成の1つとして考える

ことが出来る。

- 6) 本報告で提案した方法論は問題設定に関して特に強い前提条件を置いていないので、評価項目の種類、個数、結合方法、解の評価基準を変更した場合についても、同様な性質が導き出されることが推測出来る。勿論、フローショップ問題への適用に際しては、本問題設定の特別な場合として扱うことが出来る。

7. 結 言

- 1) ジョブショップスケジューリング問題を対象として、多目的型スケジューリングのメカニズムを実現するためにジェネティックアルゴリズムを適用する方法論を提案した。
- 2) 解探索の評価関数の重み付け値をジェネティックオペレーションによって操作することにより、評価関数を問題向きにチューニングする方法を示した。
- 3) 提案した方法論に基づいて計算機実験を行ない、上記メカニズムの挙動を示し、考察を加え、いくつかの有用な性質を実験により示した。即ち
 - (1) 収束性 (安定性)
 - (2) 探索能力
 - (3) 汎用性

である。

- 4) 以上によりジェネティックアルゴリズムのスケジューリング問題への適用可能性を示した。

参 考 文 献

- 1) De Jong, K. A. "Adaptive System Design : A Genetic Approach," IEEE Transactions on Systems, Man, and Cybernetics, Vol SMC-10, No.9, September 1980.
- 2) Davis, L. "Job shop scheduling with genetic algorithms", Proceedings of an International Conference on Genetic Algorithms and Their Applications, J. J. Grefenstette (ed.) 1985.
- 3) Goldberg, D. E. "Genetic Algorithms in Search, Optimization & Machine Learning", Addison-Wesley 1989.
- 4) 関根智明 (監訳), "スケジューリングの理論", 日刊工業新聞社, 1971.
- 5) Booker, L. "Improving Search in Genetic Algorithms", Genetic Algorithms and Simulated Annealing, Davis, L.(ed.), Pitman, 1987.