

二つの「交換型突然変異」の発想の必然性

The necessity of the idea of two swap-type mutations

樋口 光明*

Abstract

In the Bulletin of Niigata University of International and Information Studies [No.4], I proposed the character-preserving mutation and announced that genetic manipulation was effective in the scheduling of a certain synthetic resin factory.

The character-preserving mutation was applied to a Job Shop scheduling problem which was a sample problem for the measurement of efficiency. And, it was confirmed to be statistically significant, as reported in JCKBSE-2002.

This research object prohibits including the same elements in one schedule. Therefore, an individual phenotype has to be created as a case of genetic manipulation by a genetic algorithm for the order expression. However, if crossover and mutation are done using the order expression, a good gene sequence in the previous generation may not leave the prototype.

Many reports have indicated that it is not suitable to apply a genetic algorithm to a scheduling problem. However, other reports issued at almost the same time as mine proposed the same idea that I proposed, using a different operational method.

One of the purposes of the mutation is that the characteristics of the previous generation are not inherited by the next generation. I became interested in the idea of the character-preserving mutation coming out of other researcher, and I examined the mutation in the scheduling problem and the differences between the two experimental techniques.

*HIGUCHI, Mitsuaki [情報システム学科]

1. はじめに

筆者は、紀要第4号で、「形質遺伝を重視した突然変異」の提案をし、その遺伝子操作が実在の合成樹脂工場のスケジューリング作成で効果があることを発表した⁽¹⁾。またそれをもとに、効率測定のためのサンプル問題であるJob shop scheduling（以下JSSと表記する）に適用し、統計的に有意であることを確認し、JCKBSE-2002で報告した⁽²⁾。

この研究対象は重複選択を許さないという大きな性質を持っている。そのため、一般には個体表現型を順序表現にして遺伝子操作を行っている。しかし、順序表現で「交叉」及び「突然変異」を行うと、前の世代の良い遺伝子が原形を留めなくなるので、「交叉」や「突然変異」が必ずしも優秀な個体を生み出すとは限らないのではないかと考えられていて、事実スケジューリング問題にGAは不向きだとする論文も多い⁽³⁾など。

ところが、筆者の発表とほぼ同じ頃、同じ発想で別の操作法を用いた形質を遺伝させる突然変異の論文を発見した⁽⁴⁾。

突然変異とは、そもそも親の形質を引き継がないために起す遺伝子操作なのに、形質遺伝を重視した突然変異という発想が他からも出たということに興味を持ち、スケジューリング問題における突然変異について考察し、併せて2つの手法の違いについても実験を伴った考察をした。

2. ここで提案する「交換型突然変異」とは

本論文で提案する「交換型突然変異」では、JSSにGAを適用するときに、Genotypeのまま突然変異をさせ、しかも致死遺伝子にならないように、以下のような操作（交換型突然変異）を考案した。

この操作について、簡単な例を用いて説明する。

例えば、遺伝子長が10でgenotypeがA C E G I B D F H Jの染色体の2番目の遺伝子座がCからHに変異したとする。

これを同じ要素が現れないように突然変異させるために順序表現にすると、

1 2 3 4 5 1 1 1 1 1から1 7 3 4 5 1 1 1 1 1に変異したことになる。これを再びgenotypeに戻すと、A H D F I B C E G Jとなる。これでは、変異前の形質は“CEG”が残るに過ぎない。望ましい形質がより長く存在するとき、こうした突然変異は良い結果を生まないことになる。

一方Genotypeで直接突然変異をすると、変異後は A H E G I B D F H J となり、Hが2度現れて致死遺伝子になる。そこで、変異前には H だった9番目の遺伝子座を、変異した2番目の遺伝子座が変異前に持っていた状態 C に変える。すると染色体は A H E G I B D F C J となり、生存可能になるばかりではなく、変異前の遺伝子並びのうち、“E G I B D F” が変異前の形質を残している。

A <u>C</u> E G I B D F H J	⇒	A <u>H</u> E G I B D F <u>H</u> J	⇒	A H E G I B D F C <u>J</u>
↑ 突然変異		致死遺伝子		強制的に生存可能遺伝子にする ↑

(図1) 交換型突然変異の仕組み

この方法の長所は、突然変異前の遺伝形質を残すだけでなく、サブツアー交叉⁽⁶⁾のように部分集合が一致しないと交叉出来ないのに比べ、必ず変異できるところにある。

3. 関連論文の中の「交換型突然変異」の仕組み

この手法は、単一機械でのスケジューリングにダイナミックプログラミングを適用するもので使われていた⁽⁶⁾。

突然変異をGenotypeのまま行うのは同じである。ただ突然変異する遺伝子座が決まれば、それが何に変異するかは隣(次)の遺伝子座により決まる。つまり次の遺伝子と交換するのである。

この操作について、簡単な例を用いて説明する。

例えば、遺伝子長が10でgenotype が A C E G I B D F H J の染色体の2番目の遺伝子座が変異したとする。すると、2番目と3番目が交換して、新しい遺伝子 A E C G I B D F H J が誕生する。勿論致死遺伝子になることはない。

しかも前の世代の形質は“G I B D F H J”と遺伝する。むしろこちらの方こそ「交換型」と言えそうで、筆者の提案した突然変異は、「置き換え型」とでも言った方がよさそうである。ただし、本研究では、筆者の手法を「交換型」といい、この方法を「隣接交換型」と呼ぶ。

4. 本研究で取り上げたJSS問題

筆者の提案した「交換型」は、JCKBSE2000で、JSS問題へ適用することが意見として出された。それを受けて、JSSに対して実験をした結果、良い結果が得られた。それを整理すると以下ようになる。

遺伝子は長い方がよりはっきりと差が見えてくる。そこで、筆者は、2つの機械による2個の加工手順から構成され、加工経路が同じのN個の仕事がある場合に、総作業時間が最小になるスケジュールを作成することを試みた。(Nジョブ、2マシン、2手順、J-n-2-2)

最初は文献⁽⁶⁾より、J-6-2-2(表1)より試みた。これは、ジョンソンの方法により、最適解が求まる。総処理時間は94である。これは6!通りの組み合わせしかないので、すぐ解は求まり、突然変異の方法による差はなかった。そこで、(表1)を3個つなげた作業(J-18-2-2)、及び4個つなげた作業(J-24-2-2)について実験を試みた。

以上については、既にJCKBSE2002で発表したもので、同じ条件の実験を「隣接交換型」についても試みた。

5. 形質を遺伝する突然変異の意味

実験については後述するが、両方の手法とも、使用しない場合に比べて有意差があったので、ここでは、その考察をする。

全ての個体に交換型突然変異を適用すると、従来の順序表現による突然変異よりも悪い結果になる。これについては、次の様な原因が考えられる。

適応度の低い個体についても遺伝子形質を残すということは、良い遺伝子の発生を妨げることになってしまう。

そこで、残したい遺伝子形質を存続させるために、適応度の良い個体のみ形質を残す試みをした。具体的には、理論上の最高の適応度を目標に、目標値に5%又は10%程度接近したもののについてのみ交換型突然変異を適用した。

また、従来型の突然変異でも、交換型突然変異でも、最終的には満足する結果が得られる。その場合、最終の適応度を比較するだけでは優劣の判断がしにくい。そこで、適応度に基準を設け、その基準値に到達するまでの世代数を比較すると、かなり差が見える。本実験の場合、最適解が分かっていたので、それを基準値とした。

今回の研究についても、3つの手法（「順序表現」「交換型」「隣接交換型」）とも、最適解が得られた。従って、適応度の比較では差が出ないので、最適解に到達するまでの世代数の比較を行うことによって、有意差を検定した。

また、交叉だけだと進化の行き止まりを迎える。一つの世代がまるまる同一個体になった時、同じ個体同士交叉しても同じ個体が生まれることになるので、当然である。

逆に、突然変異の場合は、これだけでも今までに存在しなかった個体が発生する。ただし、交叉を行わなかった場合のデメリットは数量的には把握していないが、今度の実験では、交叉の影響を排除して、突然変異の効果のみを見るため、あえて交叉はしなかった。

6. 実験の実施

対象とするジョブは、表1の3倍長さ及び4倍のものとした。つまり、遺伝子長18及び24で、遺伝子並びがそのまま作業順序となる。

1世代の個体の数を10及び100とした。

交叉は行わず、突然変異だけで進化をさせた。

突然変異の確率は1遺伝子座につき9分の1、つまり1個体で平均2回のチャンスを作り、交叉のない分進化を速めるようにした。

乱数の初期値を変えて、100回の結果を比べた。

実験は3通り、「順序表現による突然変異」「交換型突然変異」「隣接交換型突然変異」である。どの時点で交換型を適用するかについては、J-6-2-2の解である時間単位94の3倍程度を目処に、適応度280前後を分岐点として、これより少ない適応度の遺伝子については形質を残し、それ以上のものについては従来通りの順序表現による突然変異を行った。

実験の結果の一部を（表2）に示す。これは乱数の初期値を1から10まで変えた、最初の10実験である。同じ乱数の初期値を使うと、第0世代が同じになり、一般に解への到達世代も似た傾向にある。そこで、本研究の提案の有効性を調べるために、各実験毎に、到着世代数のどちらが少なく済んだかの勝敗を有意差検定に使うことにした。

7. 実験結果と考察

結果として、「交換型突然変異」の実験では、「1世代の個体数10」、「J-18-2-2」、「形質を残す基準点を270」の実験結果が顕著な差が出たので、以下はこれについて述べる。

結果として表3を得た。これは、交換型突然変異が非常に有効に働いていることを意味する。

正解への到達世代で見ると、前者は100ケースで平均の差が8.7、標準偏差が18.7である。初期個体の発生の仕方により到達が大きく変わるので、同一個体で出発したときの到達世代を比較して個別に評価した結果は、59勝30敗11分となった。これは、両者に差がないという仮説、すなわち勝率50%と仮定すれば、引き分けを無視すれば、 $N(44.5, 4.722)$ に従うので、59勝する可能性は0.1%であるつまり仮説は棄却される。

「隣接型突然変異」については「1世代の個体数10」、「J-18-2-2」、「形質を残す基準点を265」の実験が一番顕著な結果となった。こちらも、42勝21敗37分となり、やはりこの手法も使った方がうまくいくことが分かった。こちらは、同じく順序表現による突然変異と勝率に差がないという仮説は、 $N(31.5, 3.972)$ に従う。

勝率5割と仮定すれば63戦中42勝の可能性は、こちらも0.4%になり、仮説は棄却できる。

8. 結論

前項で述べた通り、無条件に「交換型突然変異」を行なえば順序表現による突然変異より全部良くなるわけではない。目標適応度（理論値）に対して105%又は110%の結果を得た個体についてのみ形質を残した方が良い結果が出るのが判明した。

すなわち、今回は単純に適応度だけでどちらの突然変異を選ぶかを決めた。しかし、例えば残したい遺伝子並びを部分的に持った個体については、適応度が悪くても形質を残すなどの工夫をすることで、更に効果的な遺伝子操作が出来るのではないかと考えられる。

理論上の最適値が261の本実験の場合、部分最適解が含まれる確率は表4の通りである。この場合の部分最適解とは、最後に表1の製品2を二つ続けて作ることである。製品2は、機械2を使わずに製品が完成するので、その部分はもうそれ以上良い組み合わせが存在しない。遺伝子を組替えることで、折角最適解に近づいたものを破壊するのは得策でない場合がある。そこを見極めるのは難しいが、少なくとも、親の形質を突破するために起す突然変異で、その趣旨に反して形質遺伝を重視する突然変異の必然性は確認された。

なお、実験のためのプログラムと、全実験の結果については、以下のWEBに掲載する。

<http://www.nuis.ac.jp/hig/kiyou3.html>

<http://www.nuis.ac.jp/hig/kiyou4.html>

表1 J-6-2-2のオリジナル問題

製品名	1	2	3	4	5	6	合計
機械1の所要時間	20	8	24	7	15	13	87
機械2の所要時間	15	0	18	13	20	14	80

表2 実験結果の一部・最適値到達世代数（1世代10個体）・「交換型」の場合、適応度270以下で選択／「隣接交換型」の場合、適応度265以下で選択

乱数の初期値	順序表現	交換型	隣接交換型
1	85	23	13
2	3	14	11
3	15	42	15
4	14	5	14
5	35	27	22
6	45	12	20
7	44	14	13
8	82	11	38
9	6	12	6
10	26	9	26

表3 J-18-2-2における各手法比較（1世代10個体、適応度270/265以下に適用）

突然変異の手法	順序表現の適用	交換型突然変異	隣接交換型突然変異	(参考)交換型全面利用
解到達平均世代	29.5	20.8	24.1	1831.6
標準偏差	25.6	18.7	20.2	5254
対順序表現勝敗	—	59勝30敗11分	42勝21敗37分	17勝81敗2分

表4 J-18-2-2 1000000の個体における部分最適解含有率

ケース	適応度265以下	適応度270以下	適応度275以下	275を越えるもの
部分最適解含有率 100個体10000世代	100%	29.8%	9.9%	1.4%
部分最適解含有率 10000個体100世代	100%	22.1%	7.7%	0.9%

参考文献

- (1) 樋口光明：「形質遺伝を重視した突然変異の提案とその有効性」新潟国際情報大学情報文
化学部紀要 第4号pp123-135
- (2) M.Higuchi,M.Nagata: 「」 Knowledge-Based Software Engineering, Proceeding of the Fifth Joint
Conference on Knowledge-Based Software Engineering, IOS press, Ohmusha, Amsterdam,2002
pp307-312
- (3) 三木・広安・水田・吉田：「並列分散遺伝的アルゴリズムを用いた巡回セールスマン問題
の解法」 http://mikilab.doshisha.ac.jp/dia/research/person/taka/doc/ipsj_61.pdf
- (4) Yagiura: "A dynamic programming method for single machine scheduling." European J. Operational
Research, vol.76,pp72
- (5) 山村雅幸、小野貴久、小林重信：「形質遺伝を重視した遺伝的アルゴリズムによる巡回セ
ールスマン問題の解法」人工知能学会誌、Vol.7,No6,pp.117-127,1992
- (6) 黒田充、田部勉、圓川隆夫、中村甚一郎：「生産管理」朝倉書店、1989