

組合せ最適化問題における内挿/外挿的な領域への 遺伝的多段階探索の有効性

花田 良子[†] 廣安 知之^{††} 三木 光 範^{††}

dMSXF は JSP における有効な交叉の 1 つである MSXF を改良した交叉であり, TSP において非常に良好な解探索性能を示している. MSXF, dMSXF はいずれも, 順序付け問題において問題固有の近傍構造および距離を導入し, 両親の形質遺伝を目的とした多段階交叉である. これらは内挿的な交叉であるため両親の距離が近すぎる場合には有効に働かない. そのため MSXF ではその相補的な操作として外挿的な領域への多段階探索である MSMF を併用することで, その性能を向上させている. 一方, dMSXF は多段階探索交叉を決定的に行う手法であり, MSMF の仕組みについては採用されていない. 本論文では dMSXF の相補的な操作である外挿領域への決定的な多段階探索 dMSMF を提案する. 異なる構造を持つ 2 種の代表的な組合せ最適化問題 TSP, JSP において dMSXF+dMSMF の有効性を示すことで, 内挿領域および外挿領域への決定的な多段階探索を組み合わせることによって有効な探索が実現できることを示す.

Effectiveness of Genetic Multi-step Search in Interpolation and Extrapolation Domain for Combinatorial Optimization Problems

YOSHIKO HANADA,[†] TOMOYUKI HIROYASU^{††} and MITSUNORI MIKI^{††}

The dMSXF is an improved crossover method of MSXF which is one of promising methods of JSP, and it shows high availability in TSP. Both of these crossover methods introduce a neighborhood structure and distance in each permutation problem and perform multi-step searches in the interpolation domain focusing on inheritance of parents' characteristic. They cannot work effectively when parents stand close each other since they search in interpolation domain. Therefore in the case of the MSXF, the MSMF, which is the multi-step search in the extrapolation domain, is combined as the complementary search to improve their search performance. On the other hand, the dMSXF just performs deterministic multi-step search and the mechanism of the MSMF is not applied. In this paper, we introduce a deterministic MSMF mechanism as complementary multi-step extrapolation search. We apply dMSXF+dMSMF to TSP and JSP, which have structural difference between their landscapes. Through the experiments it was shown that the deterministic multi-step search in interpolation/extrapolation domain performed effectively in combinatorial problems.

1. はじめに

近年, 進化的戦略を用いた確率的最適化手法が注目を集めており, その代表的なものに遺伝的アルゴリズム (Genetic Algorithms: GAs¹⁾) がある. GA は生物の進化を模倣した確率的な最適化アルゴリズムであり, ヒューリスティック法とランダム探索法を組み合わせた進化戦略手法の 1 つである. GA は様々な問題に適用可能であり, 巡回セールスマン問題 (Traveling

Salesman Problem: TSP) をはじめとする種々の組合せ最適化問題への適用についても多くの研究がなされてきた.

組合せ最適化問題の中でも特に, 順序付け問題を GA で解く際には, 両親の形質をバランス良く受け継ぐよう, 問題固有の構造, 性質を考慮した交叉の設計が重要となる. TSP においては, エッジあるいは部分ツアーの継承を考慮した交叉²⁾⁻⁴⁾, ジョブショップスケジューリング問題 (Job-shop Scheduling Problem: JSP) においては, 部分仕事列の継承を考慮した交叉⁵⁾⁻⁷⁾ など, 問題ごとに様々な手法が提案されてきた. また, 交叉のほか世代交代モデルについても議論がなされている^{8),9)}.

順序付け問題では GA と近傍探索法を組み合わせ

[†] 同志社大学大学院/日本学術振興会特別研究員
Graduate School of Engineering, Doshisha University/JSPS Research Fellow

^{††} 同志社大学工学部
Department of Engineering, Doshisha University

ることで効果的な探索が可能である^{10)~12)}ことから、JSPにおいて、近傍探索のメカニズムを利用した多段階探索交叉 MSXF (Multi-step Crossover Fusion)¹³⁾が提案されている。MSXFは問題固有の近傍構造および距離の概念を導入し、一方の親から他方へと距離が小さくなる方向に近傍を生成し解遷移するといったことを多段階に行う。そして、探索の過程で生成された多様な子個体群の中から両親の良い部分をうまく受け継いだ子個体が次世代に残る。MSXFはJSPを対象とした大規模なベンチマークにおいて非常に高い解探索性能を示している。しかしながら、MSXFは近傍個体への解遷移の受理判定の際に、解探索性能に大きく影響を与える温度パラメータを有している。JSPにおいてMSXFをGAに適用する際には、2種の異なるスケジュール¹³⁾からなる2つの母集団を有すること、また、温度パラメータなど設定が困難なパラメータが多いことなどから、実装することが大変困難な手法であると考えられる。そこで、近傍および距離を定義するだけで容易に交叉を構成できるdMSXF (deterministic MSXF)¹⁴⁾が提案されている。dMSXFはMSXFのアイデアを基にした交叉であり、近傍探索では、個体間の距離のみを用いた決定的な方法によって解を遷移させる。また、近傍探索にともなうパラメータはステップ数および近傍個体数のみであり、これらは定義した構造によっては容易に推測できるといった特徴を持つ。dMSXFを単独で利用するだけで、TSPを対象とした大規模なベンチマークにおいて優れた性能を示している¹⁴⁾。

一方、両親の間に子個体を生成することだけではなく、両親の形質をベースに他の形質を獲得するため、両親の外側に探索を行うことを目的としたアプローチがある¹⁵⁾。Sakumaらは前者のような交叉を内挿領域での交叉、後者のような交叉を外挿領域での交叉と定義している。JSPにおいて、内挿性の交叉に加え、外挿的な領域への多段階交叉EDX¹⁵⁾を併用することにより、解探索性能が向上することが示されている。また、GAのオペレーションだけでなく、散布探索法 (scatter search)^{16),17)}の解の組合せ操作としてしばしば採用されるパス再結合法 (Path Relinking: PR)^{16),17)}においても、これまでに得られた解群の間を集中して探索する操作に加え、それと異なる空間を積極的に探索する外挿結合 (extrapolated relinking)が議論されている。先に述べたMSXFは両親の距離が近すぎる場合には有効に働かないことから、GAにMSXFを適用する際には、その相補的な操作として両

親から遠ざかっていく多段階突然変異MSMF (Multi-step Mutation Fusion)を併用することで高い性能を示している。MSXFの原論文では内挿、外挿という概念については述べられていないが、MSXFは内挿領域での多段階探索、MSMFは外挿領域における多段階探索であると考えられる。これらのことから、順序付け問題において、両親の間および両親の外、すなわち内挿領域および外挿領域での多段階探索を組み合わせることによって効果的な探索が可能になると考えられる。

dMSXFは近傍および距離を定義するだけで容易に交叉を構成でき、かつ性能が高く、順序付け問題において非常に有効な交叉である。dMSXFはMSXFと同様に、内挿領域での探索であり、MSMFの仕組みである外側への多段階探索を考案すれば、さらに良い手法になると考えられる。本論文ではdMSXFの相補的な操作である外挿領域への決定的な多段階突然変異dMSMF (deterministic Multi-step Mutation Fusion)を提案する。dMSMFは、MSMFと同様に両親の距離が近すぎる場合には遠ざかる方向へ多段階に近傍探索を行い、解遷移はdMSXFと同様に個体間の距離のみを用いた決定的な方法によって行う。まずdMSXFが高い性能を示した大谷構造であるTSPにおいて、dMSMFを組み合わせることによって探索性能が向上することを示す。次に、異なる構造を有するクラスの問題としてJSPにおいて、dMSXFおよびdMSMFの有効性を示すことで、組合せ最適化問題において、内挿領域および外挿領域への決定的な多段階探索を組み合わせることによって有効な探索が実現できることを示す。

2. dMSXFと提案手法dMSMF

2.1 dMSXFとは

Ikedaらによって提案されたdMSXF¹⁴⁾は、多段階探索交叉MSXF¹³⁾を改良した新たな手法である。いずれも、親 p_1 から親 p_2 に向けて局所探索を行うことで、両親の形質の受け継ぎ方が多様な子個体群を生成する。dMSXFのアルゴリズムを以下に示す。以降、個体 s_1, s_2 の距離を $d(s_1, s_2)$ と表現する。親 p_1, p_2 から生成される子個体群を $C(p_1, p_2)$ と表す。

【dMSXFのアルゴリズム】

- Step 0 p_1, p_2 を両親、その子個体群 $C(p_1, p_2) = \phi$ とする。
 Step 1 探索初期点 $x_1 = p_1, k = 1$ とし、 x_1 を $C(p_1, p_2)$ の要素として加える。
 Step 2 ステップ k における探索点 x_k の近傍個体を μ

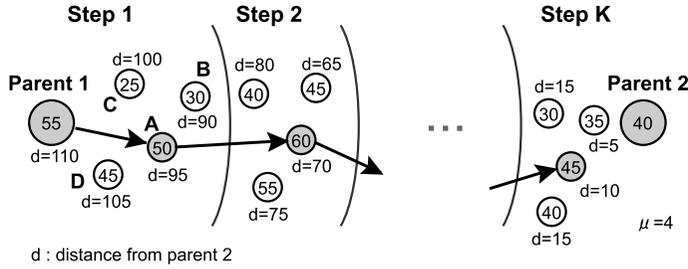


図 1 dMSXF における探索の概念図: MSXF では個体 B, A, C, D の順に選ばれやすいのに対して, dMSXF では必ず最良の個体 A が選ばれる

Fig. 1 Aspects of dMSXF: dMSXF selects the best candidate A for a transition target, while MSXF selects the target from candidates {A, B, C, D} at random, but with a bias in favor of a candidate with a smaller distance from parent 2.

個生成し, その集合を $N(x_k)$ とする. ただし, $N(x_k)$ のすべての近傍解 y_i ($0 < i < \mu$) は必ず $d(y_i, p_2) < d(x_k, p_2)$ を満たさなければならない.

Step 3 $N(x_k)$ の中で最も良い解 y を選択する. $x_{k+1} = y$ とし, x_{k+1} を $C(p_1, p_2)$ の要素として加える.

Step 4 $k = k + 1$ とし, $k = k_{max}$ あるいは x_k が p_2 に等しくなれば終了. そうでなければ, Step 2 にもどる.

図 1 にその探索の様子を示す. この操作では $k_{max} \times \mu$ 個の個体が生成される. MSXF では生成された近傍個体の中で親 p_2 に近い個体ほど選択されやすく, 選択された個体への解遷移の受理判定が温度パラメータに依存していた. 一方, MSXF では Step 2 において, 暫定解 x_k から生成する近傍の解 y_i を解 x_k よりも p_2 に近い個体に制限し, x_{k+1} が x_k よりも劣っていたとしても必ず探索を進めることで解遷移を決定的に行う.

2.2 提案手法 dMSMF

dMSXF の基となった MSXF は両親の持つ形質を子に受け継がせることを目的とした内挿領域での多段階探索であり, 両親間の距離が近すぎる場合には有効に働かない. そのため, 相補的な操作として両親から遠ざかっていく多段階探索 MSMF (Multi-step Mutation Fusion)¹³⁾ を併用している. 内挿/外挿的な領域への交叉の概念図を図 2 に示す. 図中, グレーの領域が内挿的な領域, 斜線の領域が外挿的な領域を示している.

dMSXF は内挿領域での多段階探索を決定的に行う手法であるが, 外挿的な領域への探索である MSMF の仕組みを採用していない. 本論文では dMSXF の相補的な操作である外挿領域への決定的な多段階突然変異 dMSMF (deterministic Multi-step Mutation Fusion) を提案する. 以降, 内挿的多段階交叉である dMSXF と外挿的多段階突然変異 dMSMF をそれぞれ

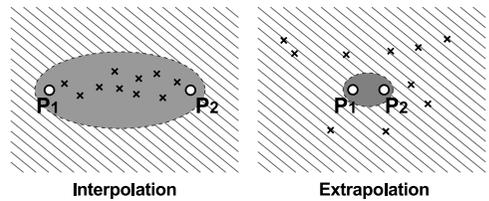


図 2 内挿/外挿的な領域への交叉の概念図

Fig. 2 Concept of the crossover in Interpolation and extrapolation domain.

れ, dMSXF(内挿), dMSMF(外挿) と表記する.

dMSXF(内挿) が親 p_1 から親 p_2 に近づいていく方向への近傍探索であるのに対して, dMSMF(外挿) は親 p_1, p_2 から遠ざかっていく方向, すなわち外挿的な領域へ多段階に探索を進めていく. dMSMF(外挿) は dMSXF(内挿) と同様に個体間の距離のみを用いた決定的な方法によって行う. dMSMF(外挿) のアルゴリズムを以下に示す.

【dMSMF のアルゴリズム】

Step 0 p_1, p_2 を両親, その子個体群 $C(p_1, p_2) = \phi$ とする.

Step 1 探索初期点 $x_1 = p_1, l = 1$ とする.

Step 2 ステップ l における探索点 x_l の近傍個体を λ 個生成し, その集合を $N(x_l)$ とする. ただし, $N(x_l)$ のすべての近傍解 y_i ($0 < i < \lambda$) は必ず $d(y_i, p_1) > d(x_l, p_1)$ かつ $d(y_i, p_2) > d(x_l, p_2)$ を満たさなければならない.

Step 3 $N(x_l)$ の中で最も良い解 y を選択する. $x_{l+1} = y$ とし, x_{l+1} を $C(p_1, p_2)$ の要素として加える.

Step 4 $l = l + 1$ とし, $l = l_{max}$ となれば終了. そうでなければ, Step 2 にもどる.

図 3 にその探索の様子を示す. この操作では $l_{max} \times \lambda$ 個の個体が生成される. MSMF では生成された近傍個体の中で親 p_1, p_2 から遠いほど選択されやすく, その解遷移の受理判定に温度パラメータを用いていた.

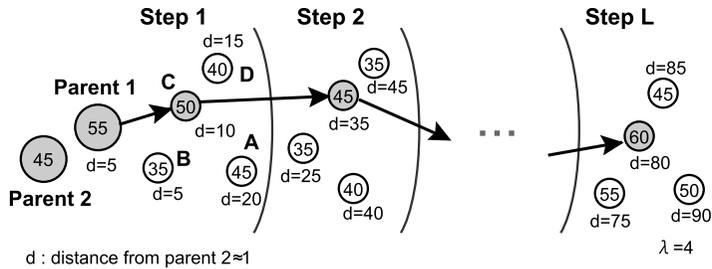


図 3 dMSMF における探索の概念図: MSMF では個体 A, D, C, B の順に選ばれやすいのに対して, dMSMF(外挿) では必ず最良の個体 C が選ばれる

Fig. 3 Aspects of dMSMF: dMSMF selects the best candidate C, while MSMF selects the target from the candidates at random, but with a bias in favor of a candidate with a larger distance from the parents.

一方, dMSMF(外挿) では, Step 2 において暫定解 x_l から生成する近傍の解 y_i を解 x_l よりも p_1, p_2 から遠い個体に制限し, x_{l+1} が x_l よりも劣っている場合でも, 必ず探索を進めることで解遷移を決定的に行う. dMSMF(外挿) を個々の問題に適用する際には, dMSXF(内挿) と同様に近傍および距離を定義しなければならない. また, Step 2 の近傍個体の生成方法を設計する必要がある.

2.3 節で述べる世代交代モデルにより, $C(p_1, p_2)$ の最良解が p_1 と置き換わる. このとき, dMSXF(内挿) では $C(p_1, p_2)$ に $x_1 (= p_1)$ が含まれているのに対して, dMSMF(外挿) ではこれを含まない. そのため, dMSMF(外挿) を適用した場合は必ず親 p_1 が他の解と置き換わる. これは, 母集団の偏りを強制的に緩和することを目的としているためである. dMSMF(外挿) は問題の構造に応じてその有効性は異なると考えられる. 本研究では, 大谷構造を持つといわれる TSP, および大域的多峰性を持つといわれる⁹⁾JSP について, 3, 5 章でその有効性を検証する.

2.3 dMSXF, dMSMF を用いた GA の解法

dMSXF(内挿), dMSMF(外挿) を GA に適用する際の処理手順を次に示す. これは, dMSXF(内挿) および EAX⁴⁾ を TSP に適用したときに良好な結果を示している世代交代モデル^{14),18)} をもとにしている.

- 【dMSXF, dMSMF を用いた GA の解法】
- Step 1 N_{pop} 個のランダムな個体 $x_1, x_2, \dots, x_{N_{pop}}$ で構成される母集団を生成する.
- Step 2 個体につけられたインデックスをランダムに付け直す.
- Step 3 N_{pop} の個体のペア $(x_i, x_{i+1}) (1 \leq i \leq N_{pop})$ を選択する. ただし $x_{N_{pop}+1} = x_1$.
- Step 4 それぞれのペア (x_i, x_{i+1}) について, x_i, x_{i+1} 間の距離が所与の値 d_m よりも小さいなら dMSMF, そうでない場合は dMSXF を適用する.
- Step 5 それぞれのペア (x_i, x_{i+1}) から生成された子個体 $C(x_i, x_{i+1})$ の最良解を選択し, x_i と置き換える.

Step 6 所与の終了条件(例: 世代数, 総評価回数)を満たした場合は終了. そうでない場合は Step 2 にもどる.

本研究では, 順序付け問題を対象としており, 局所探索と親和性が高い. そのため, Step 1 において, TSP を解く場合には初期解に 2-opt 法を適用するといったように, ランダムに生成した初期解に対してヒューリスティックな方法を適用することでより効率的な探索を図る.

3. TSP における内挿/外挿での多段階探索の有効性

TSP に dMSXF(内挿) を適用するにあたり, 池田らは距離を異なるエッジの数とし, 近傍には, 交叉 EAX⁴⁾ のプロセスで生成される AB サイクルに基づいた近傍を採用している. 親 p_1 から親 p_2 に近づいていく方向への探索において, 各ステップでの暫定解 x_k の近傍個体群を生成する際には, x_k と p_2 についての AB サイクルを 1 つだけ x_k に適用することで, x_k よりも多く p_2 のエッジを継承する個体群が生成される. この操作を多段階に行うことで, 両親の形質の受け継ぎ方が多様な子個体群の中から両親の良好な部分をうまく受け継いだ子個体が次世代へ残る. 本研究では, その dMSXF(内挿) の設計をもとに, TSP における dMSMF(外挿) を考案する.

3.1 TSP における dMSMF の設計

dMSXF(内挿) は母集団が保持する要素を組み合わせることにより解探索を進めていく. 大谷構造を持つといわれる TSP では初期母集団の持つ要素をうまく組み合わせていくこと, すなわち内挿領域を探索することで質の良い解が得られる. このことは EAX や dMSXF(内挿) が TSP において非常に高い探索能力を有しているということから明らかである. 表 1 に示すように, dMSXF(内挿) では初期母集団が保持す

表 1 dMSXF(内挿)における母集団サイズの影響
Table 1 Influence of the population size on dMSXF.

Instance	$N_{pop} = 50$	$N_{pop} = 100$	$N_{pop} = 200$	$N_{pop} = 300$
rat575	1	10	19	27
rat783	0	18	29	30
pr1002	2	15	29	30

30 試行中最適解を得た回数

る要素が多いほど、最適解を得る確率は高くなる。この結果は母集団サイズ N_{pop} を 50, 100, 200 および 300, $k_{max} = 5$, $\mu = 8$, 100 世代で計算を打ち切ったときの最適解を得た回数である。なお、初期母集団生成時には、ランダムに生成した初期解に対して 2-opt 法を適用している。

一方、表 1 から分かるように、例題に対して母集団サイズが小さい場合には初期個体群が保持する要素が少なく、内挿領域における探索だけでは良好な解が得られない。このことから、dMSMF(外挿)に必要なとされる役割は初期母集団にない、あるいは探索の途中で失った要素を補うことであると考えられる。

外挿領域に解を生成する方法を考える際には、操作を施したときの距離の変化、すなわち変異のスケールを考慮しなければならない。たとえば、2-change¹⁹⁾のように、dMSXF(内挿)の 1 ステップの操作に対して変異のスケールが小さすぎるような操作であると内挿領域から超えることが困難であり、不適切であると考えられる。変異スケールが大きな操作については λ -change²⁰⁾ など様々なものが考えられるが、本研究の目的は外挿領域への多段階探索の有効性を示すことであり、ここでは単純に次の方法で検証する。

【TSP における dMSMF】

Step 1 ランダムに 1 個体生成し、局所探索を適用する。これを p_3 とする。

Step 2 p_1 から p_3 に対して dMSXF(内挿)を l_{max} ステップ適用する。ただし、暫定解 x_1 における近傍個体生成の際には必ず、dMSMF(外挿)の Step 2 のように、近傍解 y_i は $d(y_i, p_1) > d(x_1, p_1)$ かつ $d(y_i, p_2) > d(x_1, p_2)$ を選ぶ。

上記の過程で得られた $x_2 \sim x_{l_{max}}$ が $C(p_1, p_2)$ となる。 p_1, p_2 の外挿領域における探索を実現するための解 p_3 は、dMSMF(外挿)を適用する際に毎回新しく生成する。なお、本実験では p_3 を得るための局所探索法として 2-opt 法を用いる。

これにより、問題例に対して母集団サイズが小さいときにも、母集団に不足しているエッジを dMSMF(外挿)で補うことで、母集団サイズの設定のいかんによ

らず、dMSXF(内挿)のみを用いるのに対して同等あるいはより良好な結果が得られると考えられる。

3.2 TSP における dMSXF+dMSMF の性能

TSP における dMSXF+dMSMF の解探索性能を検証する。対象問題例として、GA による有力な解法^{14),18)} で用いられている都市サイズ 500~4,000 程度の例題をベンチマーク集 TSPLIB²¹⁾ から選んだ。dMSXF(内挿)に必要なパラメータはステップ数 k_{max} と近傍個体数 μ であり、dMSMF(外挿)に必要なパラメータもステップ数 l_{max} と近傍個体数 λ である。TSP において考案した dMSMF(外挿)は、実質的に dMSXF(内挿)の操作が行われるため、 $k_{max} = l_{max}$ とする。同様に近傍個体数についても $\mu = \lambda$ とする。

本実験では dMSXF(内挿)の原論文で用いられている $k_{max} = 5$, $\mu = 8$ と設定する。dMSMF(外挿)は、例題 vm1748 以下の都市サイズの例題については、親 2 個体でエッジの相違数が (都市数) $\times 0.05$ 、それよりも大きなサイズの例題では (都市数) $\times 0.02$ よりも小さい場合に適用する。母集団サイズは例題 pcb3038, f3795 については 200、他の例題については 100 とする。計算終了世代は例題 pr2392 については 200 世代、例題 pcb3038, f3795 については 300 世代、それ以外の例題は 100 世代とする。

表 2 に最適解を得た回数 (#opt)、および最適解との誤差平均 (%err) を示す。これらは 30 試行の結果である。外挿領域への多段階探索の有効性を示すため、dMSXF(内挿)に突然変異的操作を併用したときの性能 (表中, +Mutation) と比較している。ここで用いた突然変異的操作とは、3.1 節に示した dMSMF(外挿)の 1 ステップ分を適用することである。ただし、近傍個体を生成するプロセスにおいて内挿/外挿領域のいかんにかかわらず個体を生成する。なお、生成個体数は $l_{max} * \lambda$ であり、その中のベストを p_1 と置き換える操作を行っている。通常 TSP における突然変異としては 2-change などの操作が適用されているが、それらは dMSXF(内挿)の操作に対して変異のスケールが小さすぎ、無意味な操作となる可能性が高い。そのため、本実験では上記の方法を用いた。

表 2 から、dMSXF(内挿)に対して突然変異的操作 (+Mutation) を併用することで良好な結果を得ていることが分かる。このことから dMSXF(内挿)に外挿的な要因を加えることで、より効果的な探索が行われることが分かる。また、dMSMF(外挿)を適用したときの解探索性能は、内挿/外挿領域によらずランダムに解を生成する突然変異的操作 (+Mutation) と比較して非常に高い。これらのことから、外挿領域への探

表 2 TSP における dMSXF+dMSMF の性能比較
Table 2 Performance of dMSXF+dMSMF on benchmarks of TSP.

Instance	dMSXF		+Mutation		+dMSMF	
	#opt	err(%)	#opt	err(%)	#opt	err(%)
pr439	26 (3.5×10^4)	0.002	30 (3.7×10^4)	0.0	30 (3.8×10^4)	0.0
att532	7 (0.8×10^5)	0.034	11 (1.5×10^5)	0.027	13 (1.8×10^5)	0.023
rat575	10 (0.8×10^5)	0.015	17 (1.2×10^5)	0.009	23 (1.6×10^5)	0.004
rat783	18 (8.8×10^4)	0.012	25 (9.2×10^4)	0.008	28 (9.7×10^4)	0.005
pr1002	15 (1.2×10^5)	0.019	23 (1.6×10^5)	0.012	25 (1.9×10^5)	0.006
pcb1173	11 (1.4×10^5)	0.007	12 (1.9×10^5)	0.005	19 (2.2×10^5)	0.004
vm1748	2 (1.8×10^5)	0.054	7 (3.1×10^5)	0.047	10 (4.0×10^5)	0.046
pr2392	14 (2.2×10^5)	0.010	16 (2.7×10^5)	0.008	24 (3.0×10^5)	0.002
pcb3038	1 (7.6×10^5)	0.007	3 (8.3×10^5)	0.006	4 (9.8×10^5)	0.006
f3795	14 (1.8×10^6)	0.022	16 (1.9×10^6)	0.017	18 (1.9×10^6)	0.017

30 試行中最適解を得た回数および誤差平均

表 3 TSP における dMSXF+dMSMF の性能比較
Table 3 Performance of dMSXF+dMSMF on benchmarks of TSP.

Instance	$N_{pop} = 50$		$N_{pop} = 100$		$N_{pop} = 200$		$N_{pop} = 300$	
	dMSXF	+dMSMF	dMSXF	+dMSMF	dMSXF	+dMSMF	dMSXF	+dMSMF
rat575	1	9	10	23	19	30	27	30
rat783	0	22	18	28	29	30	30	30
pr1002	2	14	15	25	29	30	30	30

30 試行中最適解を得た回数

索を明確に設計することが非常に重要であることが分かる。最適解を得た評価計算回数を比較すると、外挿性の強いものほど多くの評価を必要としている。これより、大谷構造を持つ TSP においては、外挿的な探索を併用することで、高い探索性能を得られる一方で、収束は遅れることが確認できる。

表 3 に母集団サイズ N_{pop} を 50, 100, 200 および 300 としたときの dMSXF(内挿) と dMSXF+dMSMF を比較した結果を示す。パラメータは前節と同様である。

表 3 より、問題例に対して個体数を十分にとったときには、dMSXF+dMSMF は dMSXF(内挿) と同等の結果を得ており、dMSMF(外挿) を組み合わせることで性能が低下しないことが分かる。一方、個体数が少ない場合には、dMSXF(内挿) を比較して、非常に有効に働いていることが分かる。このことから、大谷構造を持つ TSP において、母集団サイズの設定のいかんにかかわらず、dMSMF(外挿) を併用することは有効であるといえる。

4. JSP における内挿での多段階探索の有効性

前章では、大谷構造を持つ TSP において dMSXF+dMSMF が高い探索性能を有することを示した。本研究では、異なる構造を有するクラスの問題である JSP を対象として、dMSXF(内挿)、および

dMSMF(外挿) を併用することの有効性を示す。JSP は有力な局所解を多く含んだ複雑なランドスケープを有するといわれており、dMSMF(外挿) を適用することで、より効果的な探索が可能になると考えられる。本章では、まず dMSXF(内挿) の解探索性能を検証する。そして、次章において dMSXF+dMSMF の有効性を示す。

4.1 近傍と個体間の距離

dMSXF(内挿) および dMSMF(外挿) を JSP に適用するには、問題固有の特性を表現した適切な個体間の距離、近傍の定義が必要となる。また、2.1 節および 2.2 節で示した処理手順における Step 2 の近傍個体の生成方法を設計しなければならない。

本研究での探索対象スケジュールはアクティブスケジュールとし、近傍個体には MSXF や EDX で用いられているアクティブ CB 近傍¹³⁾ を用いる。アクティブ CB 近傍はクリティカルブロック上の作業の移動に基づく近傍で、生成されたスケジュールがアクティブ・スケジュールとなるよう GT 法²²⁾ に基づく修正操作が加えられたものである。

JSP における個体間の距離としては様々なものが提案されているが、本研究では、アクティブ CB 近傍と親和性が高いと考えられる I_2 距離¹⁵⁾ を用いる。 I_2 距離は、機械上での各作業の絶対的な位置に基づいた距離である(付録 A.1 参照)。本研究で考案する

dMSXF(内挿)では親 p_1 から親 p_2 に向けて I_2 距離が小さくなる方向に解遷移が行われる。

4.2 dMSXF(内挿)の設計

dMSXF(内挿)の処理手順における Step 2 での近傍個体群 $N(x_k)$ の生成方法について述べる。 x_k の近傍個体 $y_i \in N(x_k)$ は、 $d(y_i, p_2) < d(x_k, p_2)$ を満たさなければならない。これを満足させるため、 x_k と p_2 に対して、次に示す手法で得られる移植個体 x'_k を生成し、その個体についてのアクティブ CB 近傍を生成する。移植個体 x'_k および $\mu - 1$ 個のアクティブ CB 近傍が x_k の近傍となる。なお、移植個体の生成手続きは inter machine JOX⁷⁾ において投入位置を保存する仕事を 1 つに限った操作とほぼ同じである。

-
- 【dMSXF における近傍個体の生成手順】**
- Step 1** p_2 において x'_k へ投入順序を反映させる仕事を 1 つ選ぶ。
- Step 2** Step 1 で決めた仕事について、 p_2 から x'_k へ投入位置を保存するようにコピーする。
- Step 3** Step 2 でコピーしなかった仕事について、各機械で x_k から x'_k へ順序を保存するように左から右へあいている投入位置へコピーする。このようにして得られた解が移植個体 x'_k である。
- Step 4** 移植個体 x'_k に対して、 $\mu - 1$ 個のアクティブ CB 近傍を生成する。
-

dMSXF(内挿)における 1 ステップでは、上記の方法で得られた μ 個の近傍個体群の中から最良の個体を選ぶ。上記手順の Step 1 における仕事の選択には、1) ランダム、2) I_{2i} 距離が最大の仕事 i 、3) I_{2i} 距離が大きいほど選ばれやすくなるよう、 I_{2i} 距離の大きい順に仕事をソートし、番号 k に反比例する確率で選択、といった方法など様々なものが考えられる。本研究では 3) を採用している。近傍生成のもととなる移植個体を生成することで、移植する仕事 i についての I_{2i} 距離において、 $d(y_i, p_2) < d(x_k, p_2)$ はほぼ満たされる。ただし、JSP は機械間の依存が非常に強い問題であるため、 I_2 距離については $d(y_i, p_2) < d(x_k, p_2)$ が満たされるとは限らない。なお、この操作で得られるスケジュールはアクティブスケジュールであるとは限らないので、制約を外れた場合には付録 A.2 に示す GT 法によるアクティブスケジュールへの修正操作を適用する。

この方法はパス再結合法(Path Relinking: PR)^{16),17)} で用いられる局所探索の過程において、解候補の生成に交叉 inter machine JOX の限定的な利用、および解候補の改善法としてアクティブ CB 近傍を採用したものととらえることができる。PR は、散布探索法

(scatter search)^{16),17)} における解の組合せ操作としてしばしば採用される手法であり、集団から選択された 2 つの解に対して、一方の解から他方の解へ接近するような局所探索を行う方法である。Aiex らは JSP に PR を適用する際に、ある機械上で 2 つの作業をスワップさせることで p_2 への接近を実現している²³⁾。しかしながら、JSP においては各仕事を処理する機械の順序(技術的順序)が定められており、機械間の依存が非常に強いいため、1 機械上での操作では実行可能解の効率的な生成が困難である。本研究では、技術的順序による制約を考慮した inter machine JOX に基づく解生成を行うことで、健全性、形質遺伝性に優れた遷移が可能となると考えられる。

4.3 JSP における dMSXF(内挿)の性能

JSP における dMSXF(内挿)の解探索性能を検証する。問題例として、JSP のベンチマーク集²⁴⁾ より、代表的な小例題である ft10、abz5 (10 仕事 10 機械) および ft20 (20 仕事 5 機械) を採用する。ft10 は GA で最適解を得やすい例題であり、abz5 は UV 構造⁹⁾ を持つといわれる GA で最適解を得るのが困難な例題である。ft20 は上記の 2 例題と総作業数と同じであるが、機械数に対する仕事数の割合が高く、2 例題と比較して解空間が大きな例題である。これらの異なる特徴を有する例題を用い、他の手法と比較することで dMSXF(内挿)の有効性を示す。

表 4 に最大ステップ数 k_{max} を 4, 6, 8 および 10 としたときの最適解を得た回数、および最適解を得るのに必要とした平均評価計算回数を示す。これらは母集団サイズを 100、計算終了世代は 200 世代としたときの 30 試行の結果である。比較的小規模な例題であるため、近傍個体数 $\mu = 5$ としている。なお、個体の評価には LR 法¹³⁾ を採用している。また、評価個体について μ 個の任意のアクティブ CB 近傍を生成し、その中で最も良い解と置き換えることで評価値の改善を図っている。比較手法として、形質遺伝を重視した世代交代モデル CCM⁸⁾ に準ずるモデルを用いる。CCM については交叉に inter machine JOX、突然変異に job based shift change⁷⁾ を用いる。母集団サイズを 100 とし、交叉では 1 ペアに対し 20 子個体生成している。

表 4 から、内挿性の交叉の中でも性能の高い inter machine JOX と比較して、dMSXF(内挿)は高い確率で最適解を得ており、 k_{max} が大きいほど解探索性能が向上していることが分かる。なお、パラメータ μ については、これらのような比較的小きな例題に対しては、その設定によらず表 4 に示すように高い確率で

表 4 JSP における dMSX の性能比較

Table 4 Performance of dMSXF on benchmarks of JSP.

Instance	CCM	dMSXF			
		$k_{max} = 4$	$k_{max} = 6$	$k_{max} = 8$	$k_{max} = 10$
ft10	30 (1.4×10^5)	28 (1.2×10^5)	30 (1.1×10^5)	30 (1.0×10^5)	30 (0.9×10^5)
ft20	12 (1.6×10^5)	10 (3.1×10^5)	16 (4.5×10^5)	19 (4.9×10^5)	24 (5.8×10^5)
abz5	1 (1.8×10^5)	19 (1.5×10^5)	21 (1.2×10^5)	23 (0.9×10^5)	27 (1.0×10^5)

30 試行中最適解を得た回数 (最適解を得るのに必要とした平均評価計算回数)

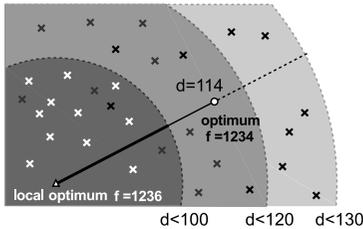


図 4 局所解側に偏らせた初期母集団の生成：初期母集団はいずれも局所解 1236 から job-based shift change を数回適用して得た解から構成される

Fig. 4 Generation of biased population: Each initial solution generated with a few applications of mutation from the local optimum ($f=1236$).

最適解を得ていることが予備実験より分かっている。

dMSXF(内挿)は大谷構造を有する TSP において非常に良好な結果を得ている。それに加え、大域的多峰性⁹⁾といわれる JSP においても良好な結果を得ており、これらの結果は順序付け問題において、近傍および距離を定義し、多段階の近傍探索を行うことで有効な探索が可能になるという dMSXF(内挿)の主張を裏付けるものと考えられる。

4.4 外挿領域への探索の必要性

dMSXF(内挿)は比較的簡単な例題において他の内挿性の交叉と比べ、高い割合で最適解を得られることが分かった。しかしながら、JSP は複雑なランドスケープを有しており、規模の大きな例題になると局所解に陥る可能性は高い。そして、局所解の谷に探索が進んでしまうと他の谷にある最適解を発見することは困難である。

図 4 のように、abz5 においてある有力な局所解側に初期個体を偏らせたときの収束傾向を検証する。ここで用いている局所解は評価値 $f=1236$ の解であり、最適解との距離は $d=114$ である。これを局所解 1236 と表現する。表 5 は、最適解との距離 $d=114$ に対して、初期解を局所解 1236 から距離 100, 120, 130 未満の領域に生成した場合に得られた解を比較したものである。これらは 50 試行の結果である。なお、これらは 50 試行の結果であり、パラメータは 4.3 節と同

表 5 局所解側に偏らせた初期母集団の収束の傾向

Table 5 Convergence tendency of GA.

distance	局所解 1236	最適解	他の局所解
$d < 100$	48	1	1
$d < 120$	38	6	6
$d < 130$	29	12	9

それぞれの解に収束した試行数 (50 試行中)

様の値を用いている。

表 5 に示すように、最適解を含まない領域である距離 100 未満の母集団はほぼすべて局所解 1236 に陥っている。一方、局所解に偏っているものの、最適解を含む領域に初期化した母集団は最適解、あるいは他の局所解を発見している。局所解 1236 と最適解はそれほど離れていないにもかかわらず、dMSXF(内挿)が内挿性の交叉であるため、局所解側の谷に母集団が固まってしまうと最適解を発見することが困難であることが分かる。

5. JSP における内挿/外挿での多段階探索の有効性

5.1 JSP における dMSMF(外挿)の設計

有力な局所解を多く含んだ複雑なランドスケープを有するといわれる JSP においては、TSP のように初期母集団にない要素を補うアプローチでは最適解を得ることは困難であると考えられる。そのため、JSP における dMSMF(外挿)の役割を、MSMF のように山を越えるような探索を行うことを目的として設計する。

個体間の距離および近傍についてはすでに 4.1 節に示した。ここでは、dMSMF(外挿)の処理手順における Step 2 での近傍個体群 $N(x_l)$ の生成方法について述べる。JSP における dMSXF(内挿)では、各ステップにおける暫定解 x_k に対して、まず移植個体を生成し、その近傍個体を生成することによって、強制的に p_1 から p_2 に向けて I_2 距離が小さくなる方向に解遷移を行っていた。それに対し、dMSMF(外挿)では各ステップの暫定解 x_l に対して、次のように強制的に I_2 距離が大きくなるような変異個体 x'_l を生成し、それをもとに近傍個体を生成する。変異個体 x'_l および

表 6 JSP における dMSXF+dMSMF の性能比較
Table 6 Performance of dMSXF+dMSMF on benchmarks of JSP.

Instance	dMSXF		+Mutation		+dMSMF	
	#opt	err(%)	#opt	err(%)	#opt	err(%)
ft10	29 (1.4×10^5)	0.022	29 (1.2×10^5)	0.025	30 (1.2×10^5)	0.0
ft20	24 (5.8×10^5)	0.19	25 (4.8×10^5)	0.11	30 (5.3×10^5)	0.0
abz5	19 (1.4×10^5)	0.16	22 (2.1×10^5)	0.12	30 (1.8×10^5)	0.0

30 試行中最適解を得た回数 (最適解を得るのに必要とした平均評価計算回数) と誤差平均

$\lambda - 1$ 個のアクティブ CB 近傍が x_l の近傍個体となる.

【変異個体の生成】

- Step 1 x_l において投入順序を変化させる仕事を 1 つ選ぶ.
- Step 2 探索点 x_l に対して, Step 1 で決めた仕事に属する作業全体を左あるいは右にシフトした変異個体 x'_l を生成する.
- Step 3 変異個体 x'_l をもとにアクティブ CB 近傍を $\lambda - 1$ 個生成する.

Step 2 の操作は小野らによって考案された突然変異 job-based shift change⁷⁾ と同等の操作である. なお, この操作で得られるスケジュールはアクティブスケジュールであるとは限らないので, 付録 A.2 に示す GT 法によるアクティブスケジュールへの修正操作を適用する. dMSMF(外挿) では, ステップ l における暫定解 x_l に job-based shift change を適用していくことで x_l から I_2 距離が大きくなる方向に探索が進むこととなる. p_1 と p_2 が非常に近い場合には, この操作を適用することで, 2 個体から遠ざかる方向に探索が進み, 母集団の偏りが緩和されることで, 解探索性能の向上を図ることができると考えられる.

5.2 dMSXF+dMSMF の性能の検証

5.2.1 小規模な問題例での性能

ft10, ft20 および abz5 を対象として dMSMF(外挿) を組み合わせることの有効性を示す. ft10, abz5 については $k_{max} = l_{max} = 5$, ft20 については $k_{max} = l_{max} = 10$ とする. 近傍個体数 $\mu = \lambda = 5$ とする. dMSMF(外挿) を適用する条件は親 2 個体の I_2 距離が (機械数) × (仕事数) × 0.1 よりも小さい場合, あるいは評価値が同じ場合に適用する. 母集団サイズを 100 とし, 計算終了世代は 200 世代とする. なお, 個体の評価には LR 法¹³⁾ を採用する. また, 評価個体について μ 個の任意のアクティブ CB 近傍を生成し, その中で最も良い解と置き換えることで評価値の改善を図っている.

表 6 に最適解を得た回数 (#opt), および最適解との誤差平均 (%err) を示す. これらは 30 試行の結果である. 外挿領域への多段階探索の有効性を示すため,

表 7 局所解側に偏らせた初期母集団の収束の傾向

Table 7 Convergence tendency of GA.

	局所解 1236	最適解	他の局所解
dMSXF のみ	48	1	1
dMSXF+dMSMF	21	28	1

それぞれの解に収束した試行数 (50 試行中)

dMSXF(内挿) に突然変異 job-based shift change⁷⁾ を併用したときの性能 (表中, +Mutation) と比較している. job-based shift change では, dMSMF(外挿) で生成する個体数と同数 ($=l_{max} * \lambda$) の突然変異個体を生成し, p_1 と置き換える操作を行っている. 表 6 から, TSP と同様に, 内挿領域での探索に重点を置く dMSXF(内挿) に対して, 突然変異あるいは dMSMF(外挿) を組み合わせることで性能が向上していることが分かる. また, 突然変異と比較して, dMSMF(外挿) を組み合わせることでより高い確率で最適解を得ており, 外挿領域における多段階探索が有効に働いていることが分かる. なお, 最適解を得るのに必要とした評価計算回数を比較すると, 傾向は例題によって異なり, 小規模な 3 例題においては外挿性の要因を加えることによる収束への影響は見られない.

5.2.2 dMSMF の探索の効果

4.4 節において, dMSXF(内挿) のみを用いた場合には, 局所解の谷に探索が進んでしまうと他の谷にある最適解を発見することは困難であることを述べた. 表 7 に abz5 を用いて, dMSXF+dMSMF において初期母集団を有力な局所解側に初期個体を偏らせたときの収束の傾向を示す. 4.4 節と同様に母集団サイズ 100, $k_{max} = l_{max} = 10$, 近傍個体数 $\mu = 5$ とし, 200 世代を打ち切りとしている. 表 7 は有力な局所解 1236 と最適解との距離 114 に対して, 最適解を含まない領域である距離 100 未満の領域に初期個体を生成した 50 試行の結果である.

表 7 から, dMSMF(外挿) を用いることによって, 局所解側の谷に母集団が固まってしまった場合でも, 最適解を発見することが可能であることが分かる. このことから, 複雑なランドスケープを持つクラスの間

表 8 10tough 問題における dMSXF+dMSMF の性能と他手法との比較
Table 8 Performance of dMSXF+dMSMF on 10 tough problems.

Ins.	dMSXF+dMSXF			dMSXF のみ			MSXF	JOX
	#opt	avg	wst	#opt	avg	wst	+ MSMF	+ EDX
abz7	658	665.3	668	664	666.6	669	678	670
abz8	668	670.4	675	670	672.1	676	686	683
abz9	679	685.9	689	686	687	689	697	686
la21	1047	1051.6	1053	1052	1052.4	1055	9/30	1/10
la24	5/10 (3.5×10^6)	936.5	938	1/10 (2.1×10^6)	939.2	941	4/30	4/10
la25	7/10 (7.1×10^6)	978.1	984	1/10 (4.9×10^6)	980.8	984	9/30	4/10
la27	5/10 (2.4×10^7)	1237.8	1242	1/10 (3.6×10^7)	1242.6	1250	1/30	1236
la29	1154	1164	1167	1163	1166.6	1168	1166	1167
la38	10/10 (9.9×10^6)	1196	1196	2/10 (4.3×10^6)	1200.7	1206	21/30	1/10
la40	1224	1227	1234	1225	1230	1240	1224	1224

10 試行中, 最適解を得た回数 (平均評価計算回数) or 最良値, 平均, 最悪値

題においても, 外挿領域への探索を行うことが重要であることが分かる.

5.2.3 10tough 問題における性能比較

比較的大規模な例題として 10tough 問題における dMSXF+dMSMF の性能を示す. 母集団サイズを 400 とし, 終了条件は 200 世代にわたって最良解の評価値が変わらないか, あるいは総評価計算回数が 5.0×10^7 となった時点とする. ステップ数 $k_{max} = 20$, $l_{max} = 10$ とし, 近傍個体数 $\mu = \lambda = 20$ とする. dMSMF(外挿) は前節と同様に, 親 2 個体の I_2 距離が (機械数) \times (仕事数) $\times 0.1$ よりも小さい場合, あるいは評価値が同じ場合に適用する. 探索の効率化のため, 初期母集団生成時において, ランダムに生成された初期解 x_i ($1 \leq i \leq N_{pop}$) それぞれに対し, dMSMF(外挿) に基づく局所探索を適用する. これは, 初期解 x_i を dMSMF(外挿) における親 p_1 ととらえ, dMSMF(外挿) の処理手順の Step 2 における親 p_2 に関する距離の制約を除いた処理である.

表 8 に dMSXF(内挿) と dMSXF+dMSMF を比較した結果を示す. これらは最適解を得た回数 (#opt) あるいは最良解, 10 試行平均 (avg), および最悪値 (wst) である. 比較として, 近傍構造を考慮した有力な交叉である MSXF+MSMF¹³⁾, JOX+EDX¹⁵⁾ の結果を参照している. なお, 終了条件の総評価計算回数 5.0×10^7 は EDX で用いられているものと同等にしているが, 最適解を得るのに必要とした評価計算回数についてはいずれも記述がないため, 最終的に得られた解精度のみの比較にとどめている.

表 8 から, dMSXF(内挿) のみを適用した場合でも良好な結果が得られているが, dMSMF(外挿) を適用することによって性能が大幅に向上していることが分かる. 最適解を得た例題に着目すると, dMSXF(内挿) と比較して dMSXF+dMSMF はおおむね, 最

解を得るのに多くの評価計算を必要としている. なお, 10 例題すべてにおいて, dMSXF+dMSMF は dMSXF(内挿) と比較して, 最終的に得られた解 (局所解を含む) を得るのにかかる計算回数は多く, 外挿的な探索を併用することで早熟収束を緩和できることが分かっている.

本手法は他の手法と比較して, 解の精度の面から, 考案している手法が多数の例題で良好な結果を得ている. また, Aiex らによる PR の解法²³⁾ と比較して非常に高い性能を示すことを確認している. MSXF および EDX は SA による解法²⁵⁾ など他の有力な近似解法と比較して, 解探索性能が良いとされている手法であり, 本研究で考案した JSP における dMSXF(内挿) および dMSMF(外挿) が有効な探索手法であることが分かる.

これらの結果から, 複雑なランドスケープを持つクラスの問題においても, 近傍および距離を定義し, 内挿および外挿領域において多段階の近傍探索を行うことで効果的な探索が可能であることが分かった.

6. おわりに

dMSXF は問題固有の近傍構造および距離を定義するだけで容易に構成できる交叉であり, その近傍探索のプロセスでは, 温度パラメータを用いるのではなく, 個体間の距離のみを用いた決定的な方法によって解を遷移させる. dMSXF は両親の持つ形質を子に受け継がせることに重点を置く内挿領域での探索であり, 他の形質を獲得するための外挿領域での多段階探索を考案すれば, さらに良い手法になると考え, dMSXF の相補的な操作である外挿領域への決定的な多段階探索 dMSMF (deterministic Multi-step Mutation Fusion) を提案した. 本論文では, まず dMSXF が高い性能を示した大谷構造である TSP において, dMSMF

を組み合わせることによって探索性能が向上することを示した。次に、dMSXF+dMSMF を TSP と異なる構造を有するクラスの問題である JSP において、その有効性を示した。これらの実験を通して、組合せ最適化問題に対して、内挿領域における決定的な多段階探索交叉が有効に働くこと、また、外挿領域への決定的な多段階探索を組み合わせることにより有効な探索が実現できることが分かった。

参 考 文 献

- 1) Goldberg, E.D.: *Genetic Algorithms in Search Optimization and Machine Learning*, Addison-Wesley (1989).
- 2) 前川, 玉置, 喜多, 西川: 遺伝アルゴリズムによる巡回セールスマン問題の一解法, 計測自動制御学会論文集, Vol.31, No.5, pp.598-605 (1995).
- 3) 山村, 小野, 小林: 形質の遺伝を重視した遺伝的アルゴリズムに基づく巡回セールスマン問題の解法, 人工知能学会論文誌, Vol.7, No.6, pp.117-127 (1992).
- 4) 永田, 小林: 巡回セールスマン問題に対する交叉: 枝組み立て交叉の提案と評価, 人工知能学会論文誌, Vol.14, No.5, pp.848-859 (1999).
- 5) 小野, 佐藤, 小林: サブシーケンス交換交叉と GT 法に基づくジョブショップスケジューリング問題の進化的解法, 電気学会論文誌 C, Vol.117-C, No.7, pp.888-895 (1997).
- 6) Shi, G. and Sannomiya, N.: A New Encoding Scheme for Solving Job Shop Problem by Genetic Algorithm, *Proc. 35th IEEE Conf. on Decision and Control*, pp.4395-4400 (1996).
- 7) 小野, 小林: Inter-machine JOX に基づく JSP の進化的解法, 人工知能学会誌, Vol.13, No.5, pp.780-790 (1998).
- 8) Ono, I. and Kobayashi, S.: A Genetic Algorithm Taking Account of Characteristics Preservation for Job Shop Scheduling Problems, *Proc. International Conference on Intelligent Autonomous Systems 5*, pp.711-718 (1998).
- 9) 池田, 小林: 生得分離モデルを用いた GA と JSP への適用, 人工知能学会誌, Vol.13, No.5, pp.530-538 (2002).
- 10) Ulder, N.L.J., et al.: Genetic Local Search Algorithms for the Traveling Salesman Problems, *1st PPSN*, pp.109-116 (1991).
- 11) Yagiura, M. and Ibaraki, T.: *Genetic and Local Search Algorithms as Robust and Simple Optimization Tools*, Kluwer Academic Publishers, MA, USA (1996).
- 12) Freisleben, B. and Merz, P.: New Genetic Local Search Operators for the Traveling Salesman Problem, *PPSN IV*, pp.890-899 (1996).
- 13) 山田, 中野: 遺伝的局所探索によるジョブショップスケジューリング問題の解法, 情報処理学会論文誌, Vol.38, No.6, pp.1126-1138 (1997).
- 14) Ikeda, K. and Kobayashi, S.: Deterministic Multi-step Crossover Fusion: A Handy Crossover for GAs, *PPSN VII*, pp.162-171 (2002).
- 15) Sakuma, J. and Kobayashi, S.: Extrapolation-Directed Crossover for Job-shop Scheduling Problems: Complementary Combination with JOX, *GECCO 2000*, pp.973-980 (2000).
- 16) Glover, F.: Genetic algorithms and scatter search: unsuspected potentials, *Statistics and Computing 4*, pp.131-140 (1994).
- 17) Marti, R., et al.: Principles of Scatter Search, *European Journal of Operational Research*, Vol.169, No.2, pp.359-372 (2006).
- 18) 永田: 局所的多様性の損失を考慮した評価関数を用いた GA の TSP への適用, 人工知能学会論文誌, Vol.21, No.2, pp.195-204 (2006).
- 19) Croes, G.A.: A method for solving traveling-salesman problems, *Operations Research*, Vol.6, pp.791-812 (1958).
- 20) Papadimitriou, C.H. and Steiglitz, K.: *Combinatorial Optimization: Algorithms and Complexity*, Dover Publications (1998).
- 21) TSPLIB. <http://www.iwr.uni-heidelberg.de/groups/comopt/software/TSPLIB95/>
- 22) Giffler, B. and Thompson, G.: Algorithms for Solving Production Scheduling Problems, *Operations Research*, Vol.8, pp.487-503 (1960).
- 23) Aiex, R.M.S.B. and Resende, M.G.C.: Parallel GRASP with path-relinking for job shop scheduling, *Parallel Computing*, Vol.29, pp.393-430 (2003).
- 24) CS410/510SS Project Job Shop Scheduling. <http://www.cs.pdx.edu/bart/cs510ss/project/jobshop/>
- 25) Aarts, E., van Laarhoven, P., Lenstra, J. and Ulder, N.: A Computational Study of Local Search Algorithms for Job-shop Scheduling, *ORSA J. on Comput*, Vol.6, No.2, pp.118-125 (1994).

付 録

A.1 I_2 距離

JSP における個体間の距離として、機械上での各作業の絶対的な位置に基づいた距離である I_2 距離¹⁵⁾を採用する。スケジュール s_a, s_b の仕事 i についての I_{2i} 距離 $I_{2i}(s_a, s_b)$ 、および s_a, s_b の I_2 距離 $I_2(s_a, s_b)$ は式 (1), (2) のように定義される¹⁵⁾。式中、

M を機械数, N を仕事数とし, 仕事 p に属する作業で機械 q で処理されるものを $o(p, q)$, 仕事 i に属する作業の集合を $J_i (= \{o(i, k) | k = 1, \dots, M\})$ としている. 作業 o に対して, $L(o)$ は o が実行される順番を示す.

$$I_{2i}(s_a, s_b) = \sum_{k=1}^M |L(o_a(i, k)) - L(o_b(i, k))| \quad (1)$$

$$I_2(s_a, s_b) = \sum_{i=1}^N I_{2i}(s_a, s_b) \quad (2)$$

A.2 JSP における個体の修正操作

dMSXF(内挿) および dMSMF(外挿) の Step 2 の操作において生成される移植個体, 変異個体はアクティブスケジュールであるとは限らない. より少ない修正でアクティブスケジュールに戻すため, 次のような GT 法に基づいた操作を行っている.

【アクティブスケジュールへの修正】

Step 1 技術的順序上, 次に処理可能な作業全体を G とし, G の中で最早完了時刻が最も小さい作業を O^* とする. O^* を処理する機械を M^* とする.

Step 2 機械 M^* 上で, すでに $(i-1)$ 個の作業が処理されているとすると, コンフリクト集合 $C[M^*, i] = \{O \in G | EC(O^*)\}$ を求める. ただし, 作業 O の最早開始時刻を $ES(O)$, 最早終了時刻を $EC(O)$ とする.

Step 3 M^* で, 現スケジュール上で i 番目に処理すべきだとされている作業を g_{M^*i} とする. $g_{M^*i} \in C[M^*, i]$ ならば g_{M^*i} を, そうでなければ, 現スケジュール上で $g_{M^*i} \in C[M^*, i]$ の作業中, 最も早く処理すべきだとされている作業を O とする.

Step 4 O を最早開始時刻 $ES(O)$ に等しくなるよう処理する.

(平成 18 年 1 月 5 日受付)

(平成 18 年 7 月 4 日採録)



花田 良子 (正会員)

2004 年同志社大学大学院工学研究科修士課程修了. 現在, 同志社大学大学院工学研究科博士課程在学中. 日本学術振興会特別研究員. 進化的計算, 最適設計, 並列処理等の研究

に従事.



廣安 知之 (正会員)

1997 年早稲田大学大学院理工学研究科後期博士課程修了. 早稲田大学理工学部助手を経て, 1998 年同志社大学工学部助手. 2003 年より同大学工学部知識工学科助教授. 進化的計算, 最適設計, 並列処理等の研究に従事. IEEE, 電子情報通信学会, 計測自動制御学会, 日本機械学会, 超並列計算研究会, 日本計算工学会各会員.



三木 光範 (正会員)

1950 年生. 1978 年大阪市立大学大学院工学研究科博士課程修了, 工学博士. 大阪市立工業研究所研究員, 金沢工業大学助教授を経て, 1987 年大阪府立大学工学部航空宇宙工学科助教授, 1994 年同志社大学工学部教授. 進化的計算手法とその並列化, および知的なシステムの設計に関する研究に従事. 著書は『工学問題を解決する適応化・知能化・最適化法』(技法堂出版)等多数. IEEE, 米国航空宇宙学会, 人工知能学会, 日本機械学会, 計算工学会, 日本航空宇宙学会等各会員. 通産省産業技術審議会委員等歴任. 超並列計算研究会代表.