Genetic Multi-Step Search in Interpolation and Extrapolation Domain

[Extended Abstract]

Yoshiko Hanada
Graduate School of Knowledge Engineering and Computer Sciences, Doshisha University, Kyoto, Japan
hanada@mikilab.doshisha.ac.jp

Tomoyuki Hiroyasu
Department of Knowledge Engineering and Computer Sciences, Doshisha University, Kyoto, Japan
tomo@is.doshisha.ac.jp

Miki Mitsunori
Department of Knowledge Engineering and Computer Sciences, Doshisha University, Kyoto, Japan
mmiki@mail.doshisha.ac.jp

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

WOODSTOCK '97 El Paso, Texas USA
Copyright 200X ACM X-XXXX-XX-X/XX/XX ...$5.00.

ABSTRACT

The dMSXF is an improved crossover method of MSXF which is a promising method 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’ characteristics. 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 supplementary search to improve its search performance. On the other hand, the search mechanism for acquisition of characteristics, such as MSMF, is not applied to dMSXF. 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.

Categories and Subject Descriptors

[Genetic Algorithms]

Keywords

genetic algorithm, combinatorial optimization, local search

1. INTRODUCTION

Genetic Algorithms (GA) are among the most effective approximation algorithms for optimization problems. GAs are applicable to a wide range of problems and have found many applications in combinatorial problems, such as the Traveling Salesman problem (TSP) and various scheduling problems.

A GA actualizes an effectual search using genetic operators for inheritance and acquisition of characteristics. These two classes of search, focusing on inheritance or acquisition, are called, respectively, the interpolation search and the extrapolation search by introducing a distance measure between solutions definitions of which are given in the next section [1]. In the general framework of GA, crossover plays a role in the former exploiting parents’ characteristics, while the latter corresponds to mutations that explore outside the distribution of the population.

When we apply GA to a particular problem, especially for permutation problems, it is important to design a crossover method with emphasis on the heredity of favorable characteristics of parents. Various types of crossover have been proposed in consideration of problem-specific structures and features [2, 3, 4, 5, 6]. Deterministic Multi-step Crossover Fusion (dMSXF) [7] is a promising interpolation-directed crossover method based on neighborhood search. dMSXF is an improved version of Multi-step Crossover Fusion (MSMF) [8], which introduces a problem-specific neighborhood structure and a distance measure, and performs a multi-step neighborhood search between parents by a deterministic rule. This method can generate a wide variety of offspring between parents; however, a search mechanism for exploring the external domain is required because it does not work effectively when parents’ characteristics are extremely similar to each other.

In this paper, we propose deterministic Multi-step Mutation Fusion (dMSMF) as a complementary search of dMSXF for exploring the extrapolation domain. Our method, dMSMF, performs a multi-step neighborhood search and it starts from the neighborhood of the parents and advances its search in the direction separate from them when parents are close to each other. Unlike a random mutation to be applied to perturb the population, it generates offspring in an efficient manner to gradually increase acquisition of characteristics that do not appear in the parents by multi-step search. We examined the effectiveness of our method in two problems, TSP and JSP that is among the most difficult scheduling problem. dMSXF has been reported to show good search ability in TSP [7]. Here, we first show the effectiveness of incorporation of dMSMF into dMSXF on TSP, and then dMSXF and dMSXF+dMSMF were applied to JSP. These two problems are different with regard to the
landscape; TSP has a big valley structure [9], while JSP is considered to have a complex multimodal landscape [10]. These experiments on two typical problems demonstrated the effectiveness of the deterministic multi-step search qualitatively in interpolation and extrapolation domains on combinatorial problems.

2. DETERMINISTIC MULTI-STEP CROSSOVER FUSION

2.1 Deterministic Multi-step Crossover Fusion
dMSXF and its original method MSXF that are multi-step searches based on a neighborhood search are proposed since incorporation of neighborhood searches into GAs is essential to adjust structural details of solutions in combinatorial problems [11]. Both these crossovers implement multi-step neighborhood searches from a parent \( p_1 \) in the direction approaching the other parent \( p_2 \). The previous method MSXF shows good search ability on JSP; however, it requires Metropolis criterion consisting of the temperature parameter \( T \) and a distance measure and searches in a deterministic manner. dMSXF [7] is the improved crossover, MSXF, moreover it is difficult to set due to dependence on parameter \( T \) since incorporation of neighborhood searches into GAs is considered to have a complex multimodal landscape [10].

2.2 Aspects of dMSXF from the perspective of extrapolation/interpolation domain

The entire solution space can be divided into two domains - the interpolation domain and the extrapolation domain - under the definition of a distance measure in discrete space [1]. Once these domains are defined, we can comprehend where dMSXF searches. Given a distance measure \( d \), the interpolation domain \( D_n \) and the extrapolation domain \( D_{ex} \) are defined as follows, where \( S \) denotes the entire solution space, and their aspects are illustrated in Fig. 2. In addition, this discussion can be seen not only in GAs but in Path-relinking (PR) [12], which is often adopted as an operator of Scatter Search [13, 14].

\[
D_n = \{ s \in S | d(s, p_1) \leq d(p_1, p_2) \} \\
D_{ex} = \{ s \in S | d(s, p_1) > d(p_1, p_2) \}
\]

At step 2 of the procedure of dMSXF, every neighborhood candidate \( y_i \) (1 \( \leq i \leq \mu \)) generated from \( x_k \) must be closer to \( p_2 \) than \( x_k \). In addition, dMSXF necessarily moves its transition toward \( p_2 \) even if all solutions in \( N(x_k) \) are inferior to the current solution \( x_k \). dMSXF requires two parameters, \( k_{max} \) and \( \mu \). \( k_{max} \) is the number of steps of neighborhood search and \( \mu \) is the number of generated solutions at each step of the neighborhood search. In the procedure of dMSXF, at most \( k_{max} \times \mu \) solutions would be generated, and \( C(p_1, p_2) \) is comprised of the best neighbor solutions, i.e., \( \{ x_1, x_2, \ldots, x_{k_{max}} \} \). The generation-alternation model of \( p_1, p_2 \) and \( C(p_1, p_2) \) used is described in section 3.2.

![Figure 2: Interpolation and extrapolation domains](image)

**Figure 2:** Interpolation and extrapolation domains

dMSXF searches inside \( D_n \) in Fig. 2, and to search globally, an exploration mechanism for \( D_{ex} \) is required to enhance its efficacy, especially for complicated problems. One choice as an extrapolation search is a random mutation method to be adopted to perturb the population by generating offspring randomly. However, it cannot work due to the broad distribution of offspring against the population in the later stages of the search. On the other hand, our method proposed in the next section was designed based on a stochastic local search on a well-defined extrapolation domain, which targets the acquisition of lost or lacking characteristics in the population.

3. PROPOSAL OF DETERMINISTIC MULTI-STEP MUTATION FUSION

3.1 Deterministic Multi-step Mutation Fusion

In this study, we propose a complementary search method to dMSXF, deterministic Multi-step Mutation Fusion (dMSMF). dMSMF is defined in a problem-independent manner based on a neighborhood search. In contrast to the search with approaching direction of dMSXF, dMSMF advances the search in the direction that separates from the parents’ neighborhood using the deterministic rule as follows:
Procedure of dMSMF
0. Let \( p_1, p_2 \) be parents and set their offspring \( C(p_1, p_2) = \phi \).
1. \( l = 1 \). Set the initial search point \( x_1 = p_1 \).
2. /Step 1/ Prepare \( N(x_1) \) composed of \( \lambda \) neighbors generated from the current solution \( x_1 \). \( \forall y_i \in N(x_1) \) must satisfy both \( d(y_i, p_1) > d(x_1, p_1) \) and \( d(y_i, p_2) > d(x_1, p_2) \).
3. Select the best solution \( y \) from \( N(x_1) \). Let the next search point \( x_{i+1} = y \) and add \( x_{i+1} \) into \( C(p_1, p_2) \).
4. Set \( l = l + 1 \) and go to 2. until \( l = l_{max} \).

At the step 2 of dMSMF, every neighborhood candidate \( y_i \) (\( 1 \leq i \leq \lambda \)) generated from \( x_1 \) is restricted to satisfy both \( d(y_i, p_1) > d(x_1, p_1) \) and \( d(y_i, p_2) > d(x_1, p_2) \). Even if all solutions in \( N(x_1) \) are inferior to the current solution \( x_1 \), the transition to a solution in \( N(x_1) \) is necessarily accepted.

The search aspect is illustrated in Fig. 3. In this procedure, at most \( l_{max} \) \( \lambda \) solutions would be generated.

![Figure 3: Aspects of dMSMF: dMSMF selects the best candidate C](image)

Flow of GA
1. Generate the initial population composed of \( N_{pop} \) random solutions, individuals, \( \{x_1, x_2, \ldots, x_{N_{pop}}\} \).
2. Reset indexes \{1, 2, \ldots, \( N_{pop} \)\} to each individual randomly.
3. Select \( N_{pop} \) pairs of parents \( (x_i, x_{i+1}) \) (1 \( \leq i \leq N_{pop} \)) where \( x_{N_{pop}+1} = x_1 \).
4. For each pair \( (x_i, x_{i+1}) \), if \( d(x_i, x_{i+1}) \) is smaller than predefined value \( d_m \), apply Inter-MSX, otherwise apply Extra-MSM to it.
5. For each pair \( (x_i, x_{i+1}) \), select the best individual \( c \) from offspring \( C(x_i, x_{i+1}) \) generated by parents \( (x_i, x_{i+1}) \) and replace the parent \( x_i \) with \( c \).
6. Go to 2 until some terminal criterion is satisfied, e.g., generations and/or the number of evaluations.

The effectiveness of Extra-MSM is supposed to depend on features of the problem, such as aspects of the landscape. Here, we examined the effectiveness of incorporation of Extra-MSM into Inter-MSX in TSP and JSP: the former is one of the problems that satisfy the big valley hypothesis [9], while the landscape of the latter is globally multimodal and conforms to the UV structure hypothesis [10]. It is thought to be easy for GA to find the global optimum in problems conforming to the big valley hypothesis. On the other hand, problems corresponding to the UV structure hypothesis have a number of influential local optimal solutions by which populations of GA tend to be trapped. The application and experiments of both methods for TSP are described in sections 4 and 5, and those for JSP are shown in sections 6 and 7.

### 4. APPLICATION OF INTER-MSX AND EXTRA-MSM FOR TSP

In this section, we describe how to apply Inter-MSX and Extra-MSM to TSP.

#### 4.1 Neighborhood and Distance

Inter-MSX has already been applied to TSP, in which the distance measure is defined as the number of different edges between parents, and the neighborhood structure based on the AB-cycle generated during the procedure of EAX is adopted [7]. EAX is a state-of-the-art crossover specialized for TSP, and its element, AB-cycle, can be considered as a building block that adequately perceives the characteristics of TSP. The AB-cycle is defined as a closed loop on the set of edges composed of tours of both \( p_1 \) and \( p_2 \), which can be generated by alternately tracing the edges of \( p_1 \) and \( p_2 \).

Inter-MSX has been shown to perform very well in TSP using the above definitions. Therefore, we developed Extra-MSM based on the design of Inter-MSX.

#### 4.2 Inter-MSX in TSP

We describe step 2 of previously reported Inter-MSX [7]. Inter-MSX generates neighborhood candidates \( N(x_k) \) of the transitional solution \( x_k \) as follows, and an example of this procedure is shown in Fig. 4.

Formation of neighborhood candidates of \( x_k \)
0. Let \( p_2 \) be one of parents and set the neighborhood candidates \( N(x_k) = \phi \).
1. Pick one of AB-cycles between \( x_k \) and \( p_2 \).
2. Generate an intermediate individual \( x'_k \) by applying the AB-cycle to \( x_k \) in the XOR manner, i.e., by removing edges of \( x_k \) included in the AB-cycles from \( x_k \), and adding edges of \( p_2 \) in the AB-cycles to \( x_k \). Here, \( x'_k \) is consisting of subtours and it is not a complete solution yet.
3. Modify \( x'_k \) to a valid tour by merging its sub-tours, and add it into \( N(x_k) \) as a neighbor of \( x_k \).

4. Go to 1 until \( |N(x_k)|=\mu \) is satisfied.

The generation method of \( x'_k \) is equivalent to the procedure of EAX between \( x_k \) and \( p_2 \) with one of the AB-cycles. A neighborhood \( x'_k \) that possesses more edges of \( p_2 \) than \( x_k \), i.e., \( d(x_k, p_2) < d(x_k, p_2) \), is necessarily generated by applying one AB-cycle.

![Figure 4: An example of generating a neighborhood:](image)

The design of the complementary search method, ExtraxMSM, is introduced in the next section. For TSP, which is thought to have a big valley structure, the global optimal solution can be obtained easily by continuing combining characteristics observed in the population from the initial generation. Hence, Inter-MSX, which is an interpolation-directed search, has a greater possibility of finding the global optimum in accordance with increases in the population size. In contrast, this is difficult for a small population against instances of benchmarks due to lack of favorable characteristics in the initial population. Thus, Extra-MSM should be designed to cover lacking edges and lost edges in the population.

### 4.3 Design of Extra-MSM in TSP

When we design a method for generation of neighbor solutions in the extrapolation domain of the parents, "a variational scale" should be considered. Here, we express the distance of solutions \( s_1 \) and \( s_2 \) as a variational scale, where \( s_2 \) denotes a solution generated by applying neighborhood search. For example, the variational scale by one step in Inter-MSX is \( d(p_1, p_2)/k_{\text{max}} \). Operations, such as \( 2\text{-change} \), of which the variational scale is too small\(^1 \) against the neighborhood search of Inter-MSX, are not appropriate as one step of Extra-MSM because it is difficult to generate solutions outside the interpolation domain. Here, we adopted a simple method to maintain the variational scale. In this procedure, the variational scale by one step in Extra-MSM is approximately \( d(p_1, p_2)/l_{\text{max}} \).

#### Design of Extra-MSM

0. Let \( p_1, p_2 \) be parents.
1. Prepare a random tour, solution, as a new individual and let this solution be \( p_3 \). Apply \( 2\text{-opt} \) method to \( p_3 \).
2. Apply \( l_{\text{max}} \) steps of the neighborhood search of Inter-MSX from \( p_1 \) to \( p_3 \). Here, neighbor candidates \( N(x_1) \) of the transitional solution \( x_1 \) consist of solutions satisfying both \( d(y_1, p_1) > d(x_1, p_1) \) and \( d(y_2, p_2) > d(x_2, p_2) \). Offspring between \( p_1 \) and \( p_2 \), \( C(p_1, p_2) \), consist of \( x_2, x_3, \ldots, x_{l_{\text{max}}} \) obtained by the above procedure. For each pair of parents for Extra-MSM, \( p_1 \) is newly generated to search in the extrapolation domain.

#### 5. SEARCH PERFORMANCE OF INTER-MSX+EXTRA-MSM IN TSP

The effects of incorporation of Extra-MSM into Inter-MSX in TSP were examined using the medium-scale benchmarks from TSPLIB\(^2 \). To confirm the superior ability in the search in the well-defined extrapolation domain, we compared Extra-MSM with a mutation method denoted by Inter-MSX+Extra-MSM and Inter-MSX+Mutation, respectively. Both Inter-MSX and Extra-MSM require the number of steps in the neighborhood search, \( k_{\text{max}} \) and \( l_{\text{max}} \), and the number of neighbor candidates, \( \mu \) and \( \lambda \), as parameters. Here, as Extra-MSM substantially implements Inter-MSX outside the parents, we set \( k_{\text{max}}=l_{\text{max}} \) and \( \mu = \lambda \). In addition, we used \( k_{\text{max}}=5 \) and \( \mu=8 \), as recommended previously [7]. For each pair, \( p_1 \) and \( p_2 \), for reproduction of offspring, Extra-MSX was applied instead of Inter-MSX when the distance between the parents was smaller than \( N_{\text{city}}^\ast \) where \( N_{\text{city}} \) denotes the total number of cities of the instance. We set \( a=0.05 \) for instances, in which \( N_{\text{city}} \) was below \( vn1748 \), and \( a=0.02 \) was used for other instances.

For Inter-MSX+Mutation, we adopted Extra-MSM with \( l_{\text{max}}=1 \) as a mutation method with the exception that neighborhood candidates were generated regardless of interpolation/extrapolation domain. This is because normal mutation methods, such as \( 2\text{-change} \), are anticipated to be unproductive operations due to the diminutive variational scale. The mutation generated \( l_{\text{max}} \ast \) offspring, and replaced the parent \( p_1 \) with the best solution of the offspring.

Table 1 shows the number of trials that obtained the optimum (\#opt), the average number of evaluations to acquire the optimum (\#eval), and the average error (%) from 30 trials. Here, we set the population size to 200 for \( pb3038 \) and \( fl3795 \), and 100 for others. In addition, these three methods were terminated after 200 generations of GA for \( pr2392 \), 300 generations for \( fl3795 \), and 100 generations for other instances.

As shown in Table 1, both Extra-MSM and the mutation method showed a high possibility of finding the optimum solution, which indicates that incorporation of extrapolation factors leads to improvement of search performance in TSP. Moreover, Extra-MSM enhanced the performance of Inter-MSX compared with the mutation method, which generates offspring randomly. In comparison of \#eval, to obtain the optimum, the method more highly focusing on the extrapolation domain requires more evaluations. These observations indicate that the search performance improves, while the convergence speed is reduced by the extrapolation search.

---

\(^{1}\)The variational scale of \( 2\text{-change} \) is 2.

\(^{2}\)TSPLIB: http://www.iwr.uni-heidelberg.de/groups/comopt/software/TSPLIB95/
Next, we discuss the influence of population size, $N_{\text{pop}}$, to highlight the impact of Extra-MSM. Table 2 shows a comparison between Inter-MSX and Inter-MSX+Extra-MSM under $N_{\text{pop}}=50, 100, 200,$ and 300.

As shown in Table 2, when $N_{\text{pop}}$ is set sufficient for the scale of the instances, Inter-MSX+Extra-MSM performs the search equally well as Inter-MSX. Moreover, Inter-MSX+Extra-MSM shows superior performance to Inter-MSX when $N_{\text{pop}}$ is small.

From these results, we conclude that combination of interpolation/extrapolation multi-step search is effective in TSP, which has a big valley structure.

### 6. APPLICATION OF INTER-MSX AND EXTRA-MSM FOR JSP

In this section, we discuss the effectiveness of both Inter-MSX and Inter-MSX+Extra-MSM in JSP, which is a globally multimodal problem for which there would be a strong requirement for extrapolation searches.

#### 6.1 Neighborhood and Distance

We cover the *active schedule* as the search space and adopt the *active CB neighborhood* [8] that has been used in MSXF and EDX [1]. The active CB neighborhood is composed of the solutions generated by shifting an operation inside a *critical block*, which are parts of the *critical path*, to either the head or the end of the block on a solution. In addition, these solutions of the active CB neighborhood are corrected to be active schedules using the *GT algorithm* proposed elsewhere [15]. We then adopt the $I_2$ distance [1] based on the absolute positions of operations belonging in each machine due to its high affinity with the active CB neighborhood.

With $M$ machines and $N$ jobs, $I_2$ distance on job $i$ of the schedule $s_a$ and $s_b$, $I_2(s_a, s_b)$, and $I_2$ distance of these schedules, $I_2(s_a, s_b)$ are defined as equations (1) and (2), respectively. In these equations, $o(p, q)$ denotes the operation to be processed by the machine $q$ and belonging to the job $p$. The set of operations belonging to job $i$ is represented by $J_i=\{o(i, k)|k=1, \cdots, M\}$. $L(o)$ denotes the absolute position of operation $o$. An example for this distance metric can be found in [1].

$$I_2(s_a, s_b) = \Sigma_{k=1}^{M} |L(o_a(i, k)) - L(o_b(i, k))| \quad (1)$$

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

#### 6.2 Design of Inter-MSX

Here, a generation method of neighborhood solutions at step 2 in the procedure described in the section 2.1 is designed. At step 2 of Inter-MSX, every neighborhood candidate $y_i$ ($1 \leq i \leq \mu$) generated from $x_k$ is restricted to satisfy $d(y_i, p_2) < d(x_k, p_2)$. To satisfy the condition, Inter-MSX generates the intermediate solution $x'_k$ and active CB neighbors of $x'_k$ as follows, and an example of this procedure is shown in Fig. 5.

**Formation of neighborhood candidates of $x_k$**

1. Select a job $J_i$ randomly, but with a bias in favor of $J_i$ with a large $I_2$ distance.
2. Copy the operations belonging to $J_i$ chosen at step 1 from parent $p_2$ into the intermediate solution $x'_k$, preserving their loci.
3. Copy the operations of all jobs except for $J_i$, from $x_k$ into $x'_k$, preserving their orders.
4. Generate ($\mu$-1) active CB neighborhoods of $x'_k$.

These offspring, $x'_k$ and neighborhoods of $x'_k$, construct $N(x_k)$. This procedure generates the intermediate solution

---

**Table 1: Performance of Inter-MSX + Extra-MSM on benchmarks of TSP**

<table>
<thead>
<tr>
<th>Instance</th>
<th>Inter-MSX</th>
<th>Inter-MSX+Mutation</th>
<th>Inter-MSX+Extra-MSM</th>
</tr>
</thead>
<tbody>
<tr>
<td>pr439</td>
<td>26 (3.5x10^5)</td>
<td>30 (3.7x10^5)</td>
<td>30 (3.8x10^5)</td>
</tr>
<tr>
<td>att532</td>
<td>7 (0.8x10^5)</td>
<td>11 (1.5x10^5)</td>
<td>13 (1.8x10^5)</td>
</tr>
<tr>
<td>rat575</td>
<td>10 (0.8x10^5)</td>
<td>17 (1.2x10^5)</td>
<td>19 (1.6x10^5)</td>
</tr>
<tr>
<td>rat783</td>
<td>18 (8.8x10^4)</td>
<td>25 (9.2x10^4)</td>
<td>28 (9.7x10^4)</td>
</tr>
<tr>
<td>pr1002</td>
<td>15 (1.2x10^5)</td>
<td>23 (1.6x10^5)</td>
<td>25 (1.9x10^5)</td>
</tr>
<tr>
<td>pch1173</td>
<td>11 (1.4x10^5)</td>
<td>12 (1.9x10^5)</td>
<td>19 (2.2x10^5)</td>
</tr>
<tr>
<td>vm1748</td>
<td>2 (1.8x10^6)</td>
<td>7 (3.1x10^6)</td>
<td>10 (4.0x10^6)</td>
</tr>
<tr>
<td>pr2392</td>
<td>14 (2.2x10^5)</td>
<td>16 (2.7x10^5)</td>
<td>24 (3.0x10^5)</td>
</tr>
<tr>
<td>pch3038</td>
<td>1 (7.6x10^5)</td>
<td>3 (8.3x10^5)</td>
<td>4 (9.8x10^5)</td>
</tr>
<tr>
<td>ltd795</td>
<td>14 (1.8x10^6)</td>
<td>16 (1.9x10^6)</td>
<td>18 (1.9x10^6)</td>
</tr>
</tbody>
</table>

The number of trials out of 30 that reached the optimum, average number of evaluations needed, and average error

---

**Table 2: Influence of population size in Inter-MSX + Extra-MSM**

<table>
<thead>
<tr>
<th>Instance</th>
<th>$N_{\text{pop}}$ = 50</th>
<th>$N_{\text{pop}}$ = 100</th>
<th>$N_{\text{pop}}$ = 200</th>
<th>$N_{\text{pop}}$ = 300</th>
</tr>
</thead>
<tbody>
<tr>
<td>rat575</td>
<td>Int-MSX: 1</td>
<td>Int-MSX: 9</td>
<td>Int-MSX: 23</td>
<td>Int-MSX: 30</td>
</tr>
<tr>
<td>rat783</td>
<td>Int-MSX: 0</td>
<td>Int-MSX: 22</td>
<td>Int-MSX: 28</td>
<td>Int-MSX: 30</td>
</tr>
<tr>
<td>pr1002</td>
<td>Int-MSX: 2</td>
<td>Int-MSX: 14</td>
<td>Int-MSX: 25</td>
<td>Int-MSX: 30</td>
</tr>
</tbody>
</table>

The number of trials out of 30 that reached the optimum
Figure 5: An example of generating neighborhoods: (a) $x_k$ and (b) parent 2 are given, intermediate solution $x'_k$ (c) is created from $x_k$ by inheriting $J_t$ of parent 2, and a CB neighborhood (d) is generated from $x'_k$.

$x'_k$ in consideration of technological sequence of machines to be processed on each job, since the swap between two operations on a certain machine, which is adopted in PR [5], is difficult to generate feasible solutions due to strong dependency among machines.

At each step $k$ ($1 \leq k \leq k_{max}$), Inter-MSX selects the best solution $y$ among $x'_k$ and $(\mu-1)$ neighborhoods of $x'_k$ and moves its transition from $x_k$ to $y$.

6.3 Design of Extra-MSM

We designed Extra-MSM for multimodal problems, the function of which should search across valleys of local optima.

Here, formation of neighborhood candidates of $x_t$ at step 2 of Extra-MSM mentioned in the section 2.1 is described. Every neighborhood candidate $y_i$ ($1 \leq i \leq \lambda$) generated from $x_t$ must satisfy both $d(y_i, p_1) > d(x_t, p_1)$ and $d(y_i, p_2) > d(x_t, p_2)$. Extra-MSM first generates the mutated solution $x'_t$ to advance its search in the direction to obtain candidates that have larger $L_2$ distance. It then generates $(\lambda-1)$ active CB neighbors of $x'_t$. $N(x_t)$ composed of $x'_t$ and neighborhoods of $x'_t$ is generated as follows:

**Formation of neighborhood candidates of $x_t$**

1. Set the mutated solution $x'_t = x_t$ and select a job $J_{t_i}$ randomly.
2. On the mutated solution $x'_t$, shift all operations belonging to $J_{t_i}$ leftward or rightward randomly.
3. Generate $(\lambda-1)$ active CB neighborhoods of $x'_t$.

At each step $l$ ($1 \leq l \leq l_{max}$) in Extra-MSM, the best solution $y$ among $x'_t$ and $(\lambda-1)$ neighborhoods of $x'_t$ is selected as the next solution $x_{t+1}$.

7. SEARCH PERFORMANCE OF INTER-MSX+EXTRA-MSM IN JSP

7.1 Performance of Inter-MSX in JSP

We examined the performance of Inter-MSX on JSP using the benchmarks ft10, ft20, and ab25. ft10 and ab25 have 10 jobs and 10 machines and ft20 has 20 jobs and 5 machines. The population size was 100 and the termination was set to 200 generations. Here, we set $\mu=5$ and $k_{max}=4$, 6, 8, 10. In the experiments described in this paper, the $LR$ method [8] was used for evaluation of individuals. In addition, we generated $\mu$ active CB neighbor solutions of the individuals on evaluation and replaced them with the best solution of the neighborhoods to improve the performance.

Table 3 shows the number of trials that obtained the optimum and the average number of evaluations to acquire the optimum. These results are from 30 trials. To compare Inter-MSX with another interpolation-directed crossover, we show the results of inter machine JOX [5], which is one of promising crossovers on JSP. In the comparative method, the job-based shift change [5] was applied as the mutation, and CCM [5] was adopted for the generation alternation model focusing on inheritance of parents’ characteristics. The population size was the same as that of Inter-MSX and each pair at the crossover generated 20 offspring.

Table 3 shows the superiority of Inter-MSX to JOX in terms of successful trials and its performance becomes good in accordance with increases in $k_{max}$. Inter-MSX performed well without relying on setting of the parameter $\mu$ in preparative experiments. Inter-MSX was shown to be effective on TSP, which has a big valley structure. In addition, it showed good ability on JSP, which is one of the problems with a global multimodal landscape. The above results confirm the assertion in [7] that the multi-step search with definition of both a distance measure and a neighborhood structure enables efficient searches in combinatorial problems.

7.2 Performance of Inter-MSX+Extra-MSM

7.2.1 Efficacy of extrapolation multi-step search

We examined the search performance of incorporating Extra-MSM with Inter-MSX. To highlight the effectiveness of the multi-step search in the extrapolation domain, we compared Inter-MSX+Extra-MSM with Inter-MSX+Mutation.

The benchmarks ft10, ft20, and ab25 are applied for this examination. We set $k_{max}=l_{max}=5$ for ft10 and ab25, and $k_{max}=l_{max}=10$ for ft20. The values of the parameters of GA were the same as in section 7.1. To each pair, $p_1$ and $p_2$, for reproduction, Extra-MSM was applied instead of Inter-MSX in two cases as follows: 1) $L_2$ distance between parents is smaller than $N_{op} \ast 0.1$ where $N_{op}$ denotes the total number of operations of the intended instance, 2) The fitnesses of $p_1$ and $p_2$ in the neighborhood are lower than the fitnesses of $x$. Half of the pairs were selected for the crossover.

Table 3: Performance of Inter-MSX on benchmarks of JSP

<table>
<thead>
<tr>
<th>Instance</th>
<th>JOX</th>
<th>Inter-MSX</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>$k_{max}=4$</td>
<td>$k_{max}=6$</td>
</tr>
<tr>
<td>ft10</td>
<td>28 (1.2x10^4)</td>
<td>30 (1.1x10^4)</td>
</tr>
<tr>
<td>ft20</td>
<td>12 (1.6x10^4)</td>
<td>10 (3.1x10^4)</td>
</tr>
<tr>
<td>ab25</td>
<td>1 (1.8x10^5)</td>
<td>19 (1.5x10^5)</td>
</tr>
</tbody>
</table>

The number of trials out of 30 that reached the optimum, average number of evaluations needed, and average error.
Table 4: Performance of Inter-MSX+Extra-MSM on benchmarks of JSP

<table>
<thead>
<tr>
<th>Instance</th>
<th>Inter-MSX</th>
<th>Inter-MSX+Mutation</th>
<th>Inter-MSX+Extra-MSM</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>#opt</td>
<td>err(%)</td>
<td>#opt</td>
</tr>
<tr>
<td>ft10</td>
<td>29 (1.4x10^4)</td>
<td>0.022</td>
<td>29 (1.2x10^2)</td>
</tr>
<tr>
<td>ft20</td>
<td>24 (5.8x10^2)</td>
<td>0.19</td>
<td>25 (4.8x10^2)</td>
</tr>
<tr>
<td>abz5</td>
<td>19 (1.4x10^3)</td>
<td>0.16</td>
<td>22 (2.1x10^3)</td>
</tr>
</tbody>
</table>

The number of trials out of 30 that reached the optimum, average number of evaluations needed, and average error

Table 5: Convergence tendency of GA

<table>
<thead>
<tr>
<th>Instance</th>
<th>l_{opt}(1236)</th>
<th>g_{opt}</th>
<th>another l_{opt}</th>
</tr>
</thead>
<tbody>
<tr>
<td>d &lt; 100  (Inter-MSM)</td>
<td>48</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>d &lt; 120  (Inter-MSM)</td>
<td>38</td>
<td>6</td>
<td>6</td>
</tr>
<tr>
<td>d &lt; 130  (Inter-MSM)</td>
<td>29</td>
<td>12</td>
<td>9</td>
</tr>
<tr>
<td>d &lt; 100  (Inter-MSX+Extra-MSM)</td>
<td>21</td>
<td>28</td>
<td>1</td>
</tr>
</tbody>
</table>

The number of trials out of 50 that reached each solution

As shown in Table 5, Inter-MSX with the population restricted to the domain $d < 100$ that does not include the global optimal solution is difficult to find. The populations initialized in the domain $d < 120$ and $d < 130$ covering the global optimal solution but with convergence to $l_{opt}(1236)$ can find the optimal solution or another local optimum several times. It is quite difficult for interpolation-directed Inter-MSX to obtain $g_{opt}$ although $l_{opt}(1236)$ is not some distance from $g_{opt}$ once the population begins to converge toward a local optimum. In contrast, it is highly possible to find $g_{opt}$ by incorporating Extra-MSM even if the initial population does not cover $g_{opt}$. These results indicate that the extrapolation-directed search is also essential for problems with complex landscapes.

Figure 6: Generation of biased population: Each initial solution generated with a few applications of mutation from $l_{opt}(1236)$

7.2.2 Analysis of Extra-MSM

Inter-MSX obtains the optimum with higher probability at small instances than other interpolation-directed crossover algorithms. However, JSP has a complex landscape consisting of a number of influential local optimal solutions and GA has the possibility of lapsing into a local optimum on large instances. It is difficult to find the global optimum standing in another valley once GA with an interpolation-directed crossover progresses its search into the valley of a local optimum.

We use the instance abz5 supposed to possess serious UV structure and the behavior of GA with initial populations biased toward an influential local optimum as shown in Fig. 6 is elucidated. Here, we use the local optimum of which fitness is 1236 and the distance from the global optimum is $d = 114$. It denotes $l_{opt}(1236)$ and the global optimum denotes $g_{opt}$.

Table 5 shows the convergence tendency of GA with Inter-MSX using initial populations composed of individuals satisfying, respectively, the distance $d < 100$, $d < 120$, and $d < 130$ from $g_{opt}$. It also shows the results of Inter-MSX+Extra-MSM with the initial population restricted to $d < 100$. These results are from 50 runs. The values of the parameters of GA are the same as in sections 7.1.

Table 6 compares Inter-MSX+Extra-MSM and Inter-MSX. These results are the best fitness in 10 trials or the number of trials finding the optimum, and average fitness and worst fitness. To compare our method with other multi-step search methods, we draw the results of MSXF+MSMF [8] and JOX+EDX [1] that are crossovers consisting of interpolation and/or extrapolation multi-step searches. The number of evaluations used, $5.0x10^5$, is the termination criterion of JOX+EDX. We confined the comparison to the best fitness because other indicators of performance, such as the number of evaluations to acquire the optimum, were not described previously [1] and [8].

From Table 6, we can see that Inter-MSX performs well.
in terms of accuracy of solutions. Moreover, application of Extra-MSM considerably improves search performance and shows superiority to both MSXF and EDX, both of which perform more effectively than other approximation algorithms, such as PR [12], SA [16] and TS [17].

8. CONCLUSIONS

The deterministic Multi-step Crossover Fusion (dMSXF), denoted here as Inter-MSX, is a promising crossover method that can be constructed by introducing a problem-specific neighborhood structure and a distance measure. This method performs a neighborhood search using the deterministic rule composed of only a distance measure in a problem-independent manner. However, Inter-MSX does not work effectively when parents are close to each other as it searches in the interpolation domain. Inter-MSX+Extra-MSM considerably improves search performance in terms of accuracy of solutions. Moreover, application of Extra-MSM considerably improves search performance and shows superiority to both MSXF and EDX, both of which perform more effectively than other approximation algorithms, such as PR [12], SA [16] and TS [17].

9. REFERENCES


### Table 6: Performance of Inter-MSX+Extra-MSM on 10 tough problems

<table>
<thead>
<tr>
<th>Problem</th>
<th>best_{opt}</th>
<th>10 trials</th>
<th>avg</th>
<th>wst</th>
<th>best_{opt}</th>
<th>10 trials</th>
<th>avg</th>
<th>wst</th>
</tr>
</thead>
<tbody>
<tr>
<td>abz7</td>
<td>658</td>
<td>655.3</td>
<td>658</td>
<td>656</td>
<td>660</td>
<td>666.6</td>
<td>667</td>
<td>669</td>
</tr>
<tr>
<td>abz8</td>
<td>666</td>
<td>670.4</td>
<td>670</td>
<td>670</td>
<td>672.1</td>
<td>676.7</td>
<td>687</td>
<td>689</td>
</tr>
<tr>
<td>abz9</td>
<td>679</td>
<td>685.9</td>
<td>686</td>
<td>686</td>
<td>687.2</td>
<td>690.8</td>
<td>697</td>
<td>697</td>
</tr>
<tr>
<td>la21</td>
<td>*1046</td>
<td>1051.6</td>
<td>1052</td>
<td>1052</td>
<td>1052.4</td>
<td>1052.4</td>
<td>1055</td>
<td>1055</td>
</tr>
<tr>
<td>la24</td>
<td>*935</td>
<td>935.6</td>
<td>935</td>
<td>935</td>
<td>933.2</td>
<td>940.8</td>
<td>941</td>
<td>941</td>
</tr>
<tr>
<td>la27</td>
<td>*977</td>
<td>977.1</td>
<td>977</td>
<td>977</td>
<td>980.8</td>
<td>984.1</td>
<td>984</td>
<td>984</td>
</tr>
<tr>
<td>la29</td>
<td>*1235</td>
<td>1235.8</td>
<td>1235</td>
<td>1235</td>
<td>1235.12</td>
<td>1242.6</td>
<td>1242</td>
<td>1250</td>
</tr>
<tr>
<td>la38</td>
<td>*1196</td>
<td>1196.1</td>
<td>1196</td>
<td>1196</td>
<td>1196.12</td>
<td>1200.7</td>
<td>1200</td>
<td>1206</td>
</tr>
<tr>
<td>la40</td>
<td>*1222</td>
<td>1222.2</td>
<td>1222</td>
<td>1222</td>
<td>1225.12</td>
<td>1230.1</td>
<td>1230</td>
<td>1234</td>
</tr>
</tbody>
</table>

The best fitness, number of trials out of 10 that reached the optimum, average, and worst fitness.