The Thirteenth International Symposium on Artificial Life and Robotics 2008(AROB 13th '08),
B-Con Plaza, Beppu, Oita, Japan, January 31-February 2, 2008

Effects of Non-Geometric Binary Crossover on Multiobjective 0/1 Knapsack Problems

N. Tsukamoto, Y. Nojima, and H. Ishibuchi

Department of Computer Science and Intelligent Systems,
Graduate School of Engineering, Osaka Prefecture University
1-1 Gakuen-cho, Naka-ku, Sakai, Osaka 599-8531, Japan
(Tel : 81-72-254-9198; Fax : 81-72-254-9915)
({nori@ci., nojima@, hisaoi@}cs.osakafu-u.ac.jp)

Abstract: Standard binary crossover operators (e.g., one-point, two-point and uniform) tend to decrease the diversity of solutions while they improve the convergence to the Pareto front. This is because standard binary crossover operators, which are called geometric crossover, always generate an offspring in the line segment between its parents under the Hamming distance in the genotype space. In our former study, we have already proposed a non-geometric binary crossover operator to generate an offspring outside the line segment between its parents. In this paper, we examine the effect of our crossover operator on the performance of multiobjective genetic algorithms through computational experiments on various multiobjective knapsack problems. Experimental results show that our crossover operator improves the search ability of multiobjective genetic algorithms for a wide range of test problems.

Keywords: Non-geometric crossover operators, evolutionary multiobjective optimization (EMO) algorithms, multi objective 0/1 knapsack problems.

I. Introduction

Evolutionary multiobjective optimization (EMO) algorithms have two goals. One is to maintain the diversity of solutions and the other is to improve the convergence of solutions to the Pareto front. One approach to these goals is to modify genetic operators (e.g., crossover, mutation).

Recently the concept of geometric crossover was proposed to analyze crossover operators in terms of the distances between an offspring and its parents [1]. A crossover operator is referred to as being a geometric crossover operator when the following equation holds between an offspring C and its two parents P1 and P2:

$$D(\mathrm{C}, \mathrm{P1}) + D(\mathrm{C}, \mathrm{P2}) = D(\mathrm{P1}, \mathrm{P2}), \qquad (1)$$

where $D(\mathrm{A}, \mathrm{B})$ denotes the distance between A and B in the genotype space. Standard binary crossover operators are geometric crossover because Eq. (1) always holds for the Hamming distance. That is, such a crossover operator always generates an offspring in the line segment between its two parents under the Hamming distance in the genotype space. On the other hand, many crossover operators for real number strings such as simulated binary crossover (SBX [2]) are non-geometric crossover. Such a crossover operator can generate an offspring C satisfying the following equation:

$$D(\mathrm{C}, \mathrm{P1}) + D(\mathrm{C}, \mathrm{P2}) > D(\mathrm{P1}, \mathrm{P2}), \qquad (2)$$

where the Euclidean distance is used to measure the distance between real number strings. In real number strings, non-geometric crossover operators have been used to maintain the diversity of solutions.

In our former study [3], we have already proposed a non-geometric binary crossover operator to increase the diversity of solutions. Our crossover operator generates an offspring satisfying (2) in the Hamming distance in the genotype space. That is, our crossover operator generates an offspring outside the line segment between its parents. Preliminary results [3] showed our crossover drastically improves the diversity of solutions without severe deterioration of convergence.

In this paper, we further examine the effect of our crossover operator on the performance of multiobjective genetic algorithms through computational experiments on various multiobjective knapsack problems, where we vary the size of the search space and the feasibility ratio. Experimental results show that our crossover operator improves the search ability of multiobjective genetic algorithms for a wide range of test problems.

II. Non-Geometric Binary Crossover

In this section, we explain our crossover operator [3]. Let **x** and **y** be two binary strings of length n. We denote them as $\mathbf{x} = x_1 x_2 \ldots x_n$ and $\mathbf{y} = y_1 y_2 \ldots y_n$. Our crossover operator generates an offspring $\mathbf{z} = z_1 z_2 \ldots z_n$ satisfying eq. (2) where the distance between two binary strings $\mathbf{x}$ and $\mathbf{y}$ is measured by the Hamming distance. The basic idea is to generate an offspring from one parent in the opposite side of the other parent.

First one parent is chosen as a primary parent (say $\mathbf{x}$). We choose better one from two parents as a primary parent. The other parent is used as a secondary parent (say $\mathbf{y}$). Then an offspring $\mathbf{z}$ is generated from the primary parent $\mathbf{x}$ and the secondary parent $\mathbf{y}$ as follows:

1. When $x_i = y_i$: $z_i = x_i$ with a probability $(1 - P_{BF})$,
   $z_i = (1 - x_i)$ with a probability P_{BF}.
2. When $x_i \neq y_i$: $z_i = x_i$.

In our crossover operator, the standard bit-flip operator is applied to x_i of the primary parent with a prespecified probability P_{BF} when $x_i = y_i$. On the other hand, when $x_i \neq y_i$, x_i is always inherited to the offspring.

Our crossover operator is illustrated in Fig. 1 where Parent 1 is used as a primary parent. Since the values are the same between the two parents in the first 10 loci in Fig. 1, the standard bit-flip operator is applied to each of the first 10 values of Parent 1 with a prespecified probability. On the other hand, the last 10 values of Parent 1 are inherited to the offspring with no modification since the values in the last 10 loci are different between the two parents.

```
                 *  *  *   *
Parent 1 (x)    0000000000 1111111111
Offspring (z)   0100101001 1111111111
Parent 2 (y)    0000000000 0000000000
```

Fig. 1. Example of our non-geometric binary crossover operator. In this figure, Parent 1 and Parent 2 are used as primary and secondary parents, respectively.

III. Multiobjective 0/1 Knapsack Problems

In this paper, we use four multiobjective 0/1 knapsack problems with two objectives (i.e., knapsacks) and 100, 250, 500 and 750 items in Zitzler & Thiele [4]. The k-objective n-item problem is denoted as the k-n problem (e.g., 2-100 problem). A multiobjective 0/1 knapsack problem with k knapsacks (i.e., k objectives) and n items can be written in a generic form as follows:

$$\text{Maximize} \quad \mathbf{f}(\mathbf{x}) = (f_1(\mathbf{x}), f_2(\mathbf{x}), \ldots, f_k(\mathbf{x})), \tag{3}$$

$$\text{subject to} \quad \sum_{j=1}^{n} w_{ij} x_j \le c_i, \quad i = 1, 2, \ldots, k, \tag{4}$$

where

$$f_i(\mathbf{x}) = \sum_{j=1}^{n} p_{ij} x_j, \quad i = 1, 2, \ldots, k, \tag{5}$$

$$c_i = \phi \cdot \sum_{j=1}^{n} w_{ij} x_j, \quad i = 1, 2, \ldots, k. \tag{6}$$

In this formulation, $\mathbf{x}$ is an n-dimensional binary vector (i.e., $(x_1, x_2, \ldots, x_n) = \{0, 1\}^n$), p_{ij} is the profit of item j according to knapsack i, w_{ij} is the weight of item j according to knapsack i, c_i is the capacity of knapsack i, and ϕ is the feasibility ratio. In this paper, we use the feasibility ratio $\phi = \{0.25, 0.5, 0.75\}$. Each solution $\mathbf{x}$ is handled by a binary string of the length n. When an infeasible solution is generated, we use a maximum-ratio greedy repair to transform the infeasible solution into the feasible one (for details, see [4]). We implement this repair scheme in the Lamarckian manner.

IV. Computational Experiments

We incorporated our crossover operator into NSGA-II [5]. NSGA-II is a well-known and frequently-used EMO algorithm. As test problems, we used four knapsack problems of 2-100 (i.e., two-objective 100-item), 2-250, 2-500, and 2-750 problems in Zitzler & Thiele [4]. Each solution for an n-item problem is handled by a binary string of length n. Thus the size of the search space is 2^n. We used the following parameter specifications:

Population size: 200 (i.e. $\mu = \lambda = 200$),
Crossover probability P_X: 0.8,
Mutation probability P_M: $1/n$,
Stopping condition: 2000 generations.

As a standard binary crossover operator, we used the uniform crossover operator. Our crossover and uniform crossover operators were used with the probabilities $P \cdot P_X$ and $(1-P) \cdot P_X$, respectively. We examined 11 values of P: P = 0.0, 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1.0. When P is 0.0, only the uniform crossover operator was used. On the other hand, only our crossover operator was used in the case of P = 1.0. We also examined 11 values of P_{BF}: $P_{BF} = P_M$, $2 \times P_M$, $4 \times P_M$, $5 \times P_M$, $6 \times P_M$, $7 \times P_M$, $8 \times P_M$, $10 \times P_M$, $20 \times P_M$, $40 \times P_M$, $50 \times P_M$. Average performance was calculated over 30 runs.

In order to measure both the diversity and the convergence, we used the hypervolume that calculates the volume of the dominated region by a non-dominated solution set in the objective space.

For the choice of the primary parent, two parents were compared using Pareto ranking and a crowding distance in the same way as in the fitness evaluation in NSGA-II.

1. Comparison using Obtained Solutions

To examine the effect of our crossover operator, we illustrate all solutions in some generations in a single run of the original NSGA-II and the NSGA-II with our crossover operator in Fig. 2 and Fig. 3 for the test problem with n = 500 and ϕ = 0.5. Our crossover operator was implemented using P = 0.5 and P_{BF} = $4 \times P_M$ (i.e., P_{BF} = 0.008). From the comparison between Fig. 2 and Fig. 3, we can see a large positive effect of our crossover operator on the diversity of solutions in NSGA-II.

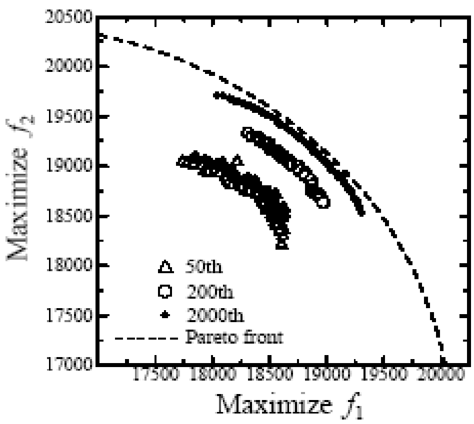

Fig. 2. All solutions in some generations of NSGA-II.

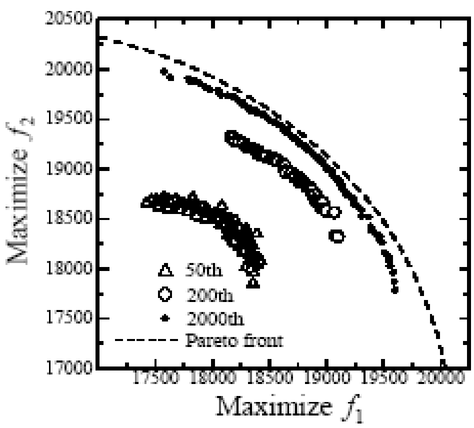

Fig. 3. Results by NSGA-II with our crossover.

2. Varying the Size of the Search Space

We examine the effect of our crossover operator using the test problems with a different number of items n (i.e., different specifications of n). The feasibility ratio ϕ was fixed. In this case, the size of the search space is given by 2^n.

Figs. 4-6 show the hypervolume for the test problems with n = {100, 500, 750} items and the feasibility ratio ϕ = 0.5, respectively. From these figures, we can see that the performance of NSGA-II (i.e., P = 0.0) was clearly improved by our crossover operator. The number of the combinations (i.e., P and P_{BF}) obtaining better hypervolume than NSGA-II was slightly increased as the number of items (i.e., n) was increased. Only when P_{BF} was large (e.g., $P_{BF} = 20 \times P_M$, $40 \times P_M$, $50 \times P_M$), the performance of our crossover operator was severely degraded.

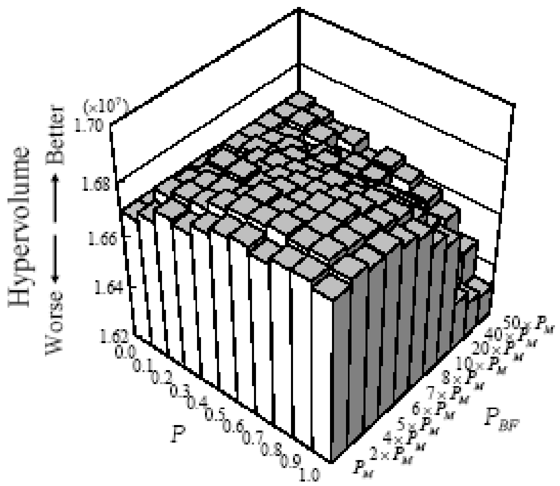

Fig. 4. Hypervolume for 2-100 problem with ϕ = 0.5.

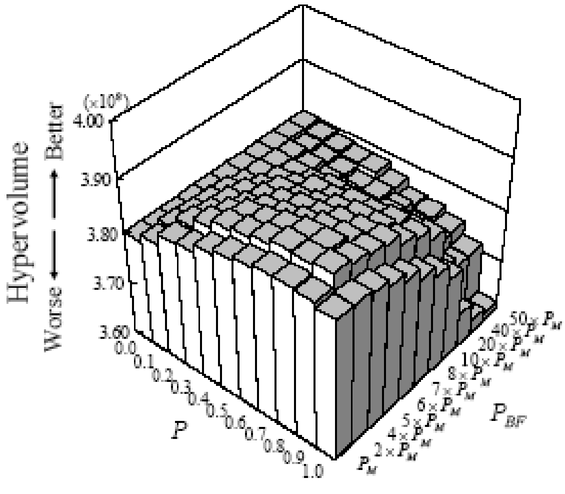

Fig. 5. Hypervolume for 2-500 problem with ϕ = 0.5.

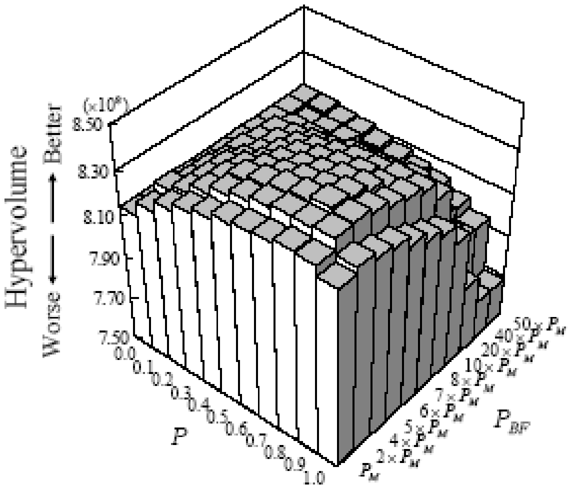

Fig. 6. Hypervolume for 2-750 problem with $\phi = 0.5$.

3. Varying the Feasibility Ratio ϕ

We also examine the effect of our crossover operator using the test problems with different values of the feasibility ratio ϕ. Figs. 7 and 8 show the hypervolume for the 2-500 problems with the feasibility ratio $\phi = \{0.25, 0.75\}$, respectively.

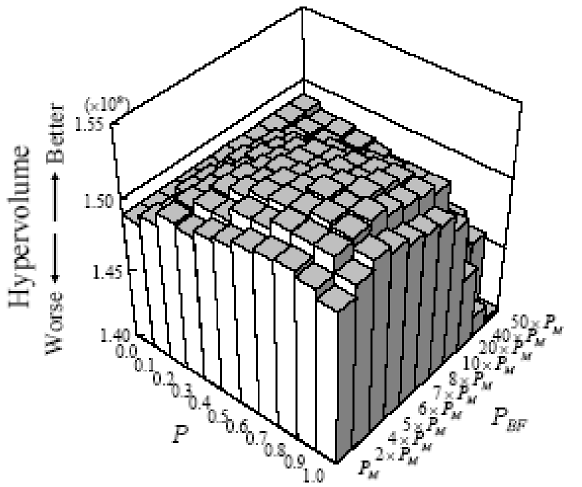

Fig. 7. Hypervolume for 2-500 problem with $\phi = 0.25$.

From Fig. 5, Fig. 7, and Fig. 8, we can observe that when the feasibility ratio ϕ is small (e.g., $\phi = \{0.25, 0.5\}$), our crossover operator improves the search ability except for the case of large P_{BF} (e.g., $P_{BF} = 40\times P_M$, $50\times P_M$). On the other hand, when the feasibility ratio ϕ is small (e.g., $\phi = 0.75$), only the small P_{BF} (e.g., $P_{BF} = P_M$, $2\times P_M$) improves the hypervolume. We can say that our crossover operator with appropriate parameter specifications improved the search ability.

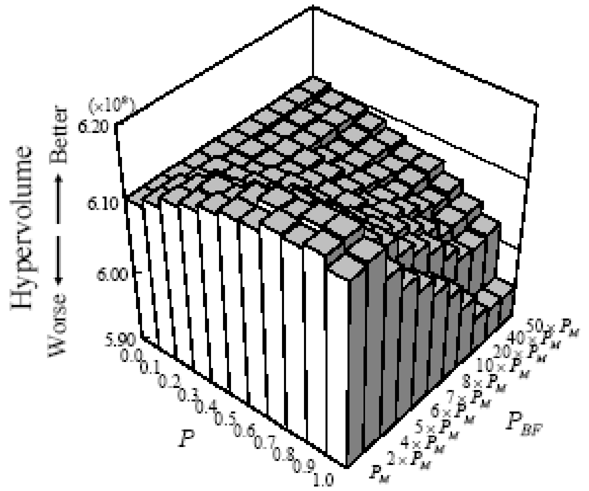

Fig. 8. Hypervolume for 2-500 problem with $\phi = 0.75$.

V. Conclusion

In this paper, we examined the effect of our non-geometric binary crossover operator using various multiobjective knapsack problems. In computational experiments, we varied the size of the search space and the feasibility ratio. Experimental results showed that our crossover operator with appropriate parameter specifications improved the search ability of multiobjective genetic algorithms.

This work was partially supported by Japan Society for the Promotion of Science (JSPS) through Grand-in-Aid for Science Research (B): KAKENHI (17300075).

References

[1] A. Moraglio, P. Poli (2006), Product geometric crossover. *LNCS 4193: PPSN 2006*, Springer, Berlin, 1018-1027.

[2] K. Deb, R. B. Agrawal (1995), Simulated binary crossover for continuous search space. *Complex Systems*, 9, 2, 431-454

[3] H. Ishibuchi, Y, Nojima, N, Tsukamoto and K. Ohara (2007), Effects of the use of non-geometric binary crossover on evolutionary multiobjective optimization. *Proc. of GECCO 2007*, 829-834.

[4] E. Zitzler, L. Thiele (1999), Multiobjective evolutionary algorithms: A comparative case study and the strength pareto approach, *IEEE Trans. on Evolutionary Computation*, 3, 4, 257-271.

[5] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan (2002), A fast and elitist multiobjective genetic algorithm: NSGA-II, *IEEE Trans. on Evolutionary Computation*, 6, 2, 182-197.