2016 International Symposium on Nonlinear Theory and Its Applications,
NOLTA2016, Yugawara, Japan, November 27th-30th, 2016

# A proposal of a new approach for strengthening the search ability of EMO algorithm -SPLASH-

Hibiki Samonji† and Shinya Watanabe†

†The Graduate School of Engineering, Muroran Institute of Technology
27–1 Mizumoto, Muroran, Hokkaido 050-8585 Japan
Email: 15043026@mmm.muroran-it.ac.jp, sin@csse.muroran-it.ac.jp

**Abstract**—In this paper, a new local search method using search history in evolutionary multi-criterion optimization (EMO) is proposed. This approach was designed by two opposite mechanisms (escaping from local optima and convergence search) and assume to incorporate these into an usual EMO algorithm for strengthening its search ability. The main feature of this approach is to perform a high efficient search by changing these mechanisms according to the search condition. If the search situation seems to be stagnated, escape mechanism would be applied for shifting search point from this stagnated condition point to another points and breaking this stagnation. On the other hand, if it observes no sign of the improvement of solutions after repeating escape mechanism in a certain number of times, this approach judges that the solutions are near global optima and convergence mechanism is applied to improve their qualities by intensive local search. In this paper, this approach is called "a escaping from local optima and convergence mechanisms based on search history - SPLASH -".

Experimental results show that the effectiveness of the proposed mechanisms was verified through investigating the influence by the presence or absence of the proposed each mechanism.

## 1. Introduction

In recent years, evolutionary multi-criterion optimization (EMO) is one of the most active research areas and various kinds of multi-objective evolutionary algorithm (MOEA) was proposed. Additionally, the combination MOEA and local search (LS) approach has been studied and presented its high search performance. These approaches are called multi-objective memetic algorithm (MOMA) and were widely proposed as typified by multi-objective genetic local search (MOGLS)[1].

In general, these approaches use a neighborhood search or a kind of gradient search and usually need high comptational costs. Also, there has been some approaches for escaping from local optima in single-objective optimization[2, 3], but very few studies in multi-objective optimization.

Therefore, we proposed a new local search method using a search history in EMO, named SPLASH[1].

SPLASH is consists of escaping from local optima machanism and convergence search mechanism. Escape mechanism estimates unexplored regions using whole search history information and tries to escape from local optima. On the other hand, convergence mechanism estimates prospective regions using a part of search history information and tries to improve a quality of solution.

The computational cost of each mechanism are same as that of genetic operation. thus, if we apply SPLASH instead of genetic operation, the computational cost in each generation is no difference to that of normal case (not using our mechanisms).

## 2. SPLASH

Our approach, named SPLASH, is consists of two mechanisms: escape machanism and convergence mechanism. These mechanisms using search history information would be expected to strengthen the search ability of MOEA. In this section, we explain how to store the search infromation into a search history, then details of these mechansisms of SPLASH.

### 2.1. How to Store Search History

Since the amount of memory in a computer simulation must be limited, every search history information cannot be stored directly. Therefore, our approach uses the discretization of the search history information when to store these information into memory. This discretization is designed by reference to the concept of long term memory[4].

The concept of this storing approach is described in Figure 1. In Figure 1, feasible region in each variables is [0,1] and the memory of search history is discretized into three parts; [0,0.33), [0.33,0.67) and [0.67,1.0].

In Figure 1, the memory of search history is reprenseted by a matrix and row of this matrix means the variable value and column presents the discriteized range of each variable value.

### 2.2. Escape Mechanism

This mechanism tries to break through a search stagnating condition in design space using search history information. In order to detect a stagnating condition, this mechanism uses a stagnation parameter $k_i$ $(i = 1, \dots, N)$. Each

[1]SPLASH is abbreviated name of "a escaping from local optima and convergence mechanisms based on search history".

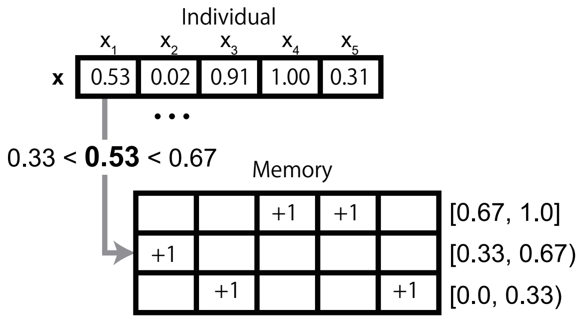


Figure 1: Concept of storing an individual to memory

individual (each sub-problem in MOEA/D) has own this parameter and increments this parameter by one ($k_i$=$k_i$+1) if the individual $\boldsymbol{x}^i$ is not update after a selection operator. On the other hand, when the individual $\boldsymbol{x}^i$ is updated, the parameter $k_i$ is assigned to zero ($k_i$=0). And then, if $k_i$ is over the pre-defined threshhold $K$ ($k_i > K$) at the beginning of next generation, escape mechanism would be performed instead of normal genetic operators for creating new solutions (such as crossover and mutation).

The flow of this mechanism is as follows. In here, $\boldsymbol{M}$ represents the memory matrix presented in 1. This matrix $\boldsymbol{M}$ is used in escape mechanism and $m_{ij}(i = 1, \ldots, D$ and $j = 1, \ldots, n)$ is a element of $\boldsymbol{M}$. Also, $D$ is the number of rows.

We defined memory $M$ each sub problems and count neighborhood solutions.

**Step1. Selecting update range**

Select candidate solutions for the seeds of new solution using *rand*, where *rand* is uniformly random number from $[0, 1]$. In here, we used two different ways of choosing candidate depending the value of *rand* like below equation.

$$\boldsymbol{P} = \begin{cases} \boldsymbol{B(i)} & \text{if } rand < \delta \\ \{1, \ldots, N\} & \text{otherwise} \end{cases}$$

The above function returns the suffix numbers of candidate solutions and vector $\boldsymbol{P}$ stores the suffix information of candidate solutions. $\boldsymbol{B(i)}$ is the function that returns around the value of input "i". On the other hand, $\{1, \ldots, N\}$ return a random integer number from 1 to $N$. This selection of vector $\boldsymbol{P}$ has a role of defining the range of updating for sub-problem in MOEA/D.

**Step2. Selecting variables for changing**

Randomly select some variables values in $\boldsymbol{x}$.

**Step3. Inverting the value of memory $\boldsymbol{M}$**

In order to invert the value of memory $\boldsymbol{M}$, each element of $\boldsymbol{M}$ is replaced by $\boldsymbol{M} = m_j^{worst} - m_{ij}(i = 1, \ldots, D$ and $j = 1, \ldots, n)$. In here, $m_j^{worst}$ means the biggest value in $j$ rows in $\boldsymbol{M}$.

**Step4. Deciding the value of variant**

Changing the value of the selected variables in Step2. In order to decide the changing velue, it needs to decide the discretized range of the selected variables. Here a roulette selection approach for each rows in $\boldsymbol{M}$ is applied to select the discretized range. Through this roulette selection, the changing velue $\boldsymbol{y}$ is randomly generated within this discretized range.

**Step5. Generate new solution**

A new solution $\boldsymbol{x}'$ is generated by replacing the values of the selecting variables of $\boldsymbol{x}$ with $\boldsymbol{y}$ generated by Step4.

**step6. Update**

Our updating follows the solution updating mechanism of MOEA/D-DE[5]. The steps of updating solutions in MOEA/D-DE are follows. In the following step, counter parameter $c$ is used for calculating the number of updating.

**1)** If $c = n_r$ or $\boldsymbol{P}$ is empty, finish. Otherwise, ramdomly pick an index $p$ from $\boldsymbol{P}$.

**2)** If $g(\boldsymbol{x}'|\lambda^p, z) \leq g(\boldsymbol{x}^p|\lambda^p, z)$, then $\boldsymbol{x}^p = \boldsymbol{x}'$ and $c = c + 1$.

**3)** Delete $p$ from $\boldsymbol{P}$ and go to 1).

## 2.3. Convergence Mechanism

This mechanism performs intensive local search around prospective areas in design space using search history information. Basic concept of this mechanism is the same as that of escape mechanism, but the role of these mechanisms are opposite. And there are two different points in terms of memory index. One point is that memory index consists of the information of near the current solution. That is, the memory used in this mechanism is same as that of escape mechanism, but focus on a part of this memory. And the second one is that the element value of memory matrix is not inverted like Step3 of escape mechanism.

The procedure of this mechanism is almost same as that of escape mechanism, but the above-mentioned search memory is quite different.

## 3. Experimental Results

In these experiments, firstly we conpared two algorithms (original MOEA/D-DE[5] and MOEA/D-DE with SPLASH) to investigate the influence of SPLASH using WFG test suites. Secondly, we investigated the influence of each mechanisms in SPLASH through analyzing a transition of the execution ratio between two mechanisms of SPLASH and genetic operation.

### 3.1. Benchmark Problems

As benchmark problems, we used WFG test suites[6]. WFG consists of nine benchmark instances, WFG1-WFG9. The details of WFG are shown in [6]. In these experiments, we set the number of variables $n = 20$, the number of position parameters $k = 2(M - 1)$ and the number of distance parameters $l = n - k$, where $M$ is number of objectives. However, only WFG3 has a degenerate Pareto flont in the case of over 3 objectives. Since there is a part of non-degenerate Pareto flont in WFG3, we used modified WFG3 proposed in [7].

### 3.2. Parameters

In these experiments, stopping criterion was 100000 function evaluations and average hypervolume value of 50 runs is used as a measure of obtained solutions. The setting of other parameter were as follows.

**MOEA/D-DE**
decomposition parameter $H = 199, 6, 4 (m = 2, 5, 8)$
population size $N = 200, 210, 330 (m = 2, 5, 8)$
neighborhood size $T = N/10$
probability of mating/update in neighborhood $\delta = 0.9$
the maximum of individuals to update $n_r = 2$
scalarizing function Tchbycheff
**genetic operators**
crossover rate $CR = 1.0$
scaling parameter in DE operator $F = 0.5$
mutation rate $MR = 1/n$
distribution index in polynomial mutation $\eta = 20$
**SPLASH**
the number of discretization of memory $D = 25$
the number of rows in convergence mechanism $D^{conv} = 5$
stagnation count to apply escape mechanism $K = 5$

### 3.3. Metrics

The search performance of algorithms was evaluated by Hypervolume(HV)[8]. HV calculates the $m$-dimensional volume that obtained solutions dominate in objective space. High HV value shows good solutions in convergence, diversity and uniformity. In these experiments, we set reference point $\boldsymbol{r} = \{3, 5, 7, 11, 13, 15, 17, 19\}$.

### 3.4. Search Perfomance in WFG

The results are shown in Table 1 to 3. Our approach performed higher HV values than those of MOEA/D-DE in WFG4 and WFG9, but HV of WFG4 was more higher than that of WFG9. The main reasons could be thought that WFG4 and WFG9 have multi-modality and WFG9 has a parameter dependence. WFG7 and WFG8 also have parameter dependence. Therefore, our approach was not so good in the case of 2 objectives. WFG2 and WFG6 are unimodal problem and the results of these problems showed similar to those of WFG7 and WFG8. Additionally, our approach was good results in the case of many-objective problem.

As a consequence, our approach was efficient for multimodal problems and many-objective problems. On the other hand, our approach was not good for unimodal problems and parameter depending problems.

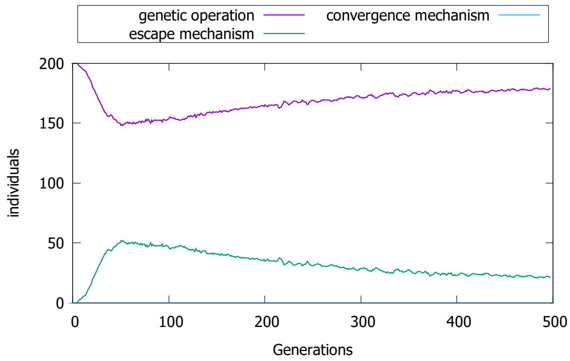


Figure 2: A transition of the execution ratio in 500 generations (MOEA/D-DE with only escape mechanism)

### 3.5. Process of Search

Table 4 shows HV and stagnation count in WFG4. In this table, SPLASH indicates MOEA/D-DE with SPLASH, escape means MOEA/D-DE with only escape mecanism and convergence means MOEA/D-DE with only convergence mecanism.

Stagnation count is the number of individuals that are not updated in each generations and the maximum value is 100000. The highest HV value is performed by proposed and lowest is performed by original MOEA/D-DE and the highest value of stagnation count is also performed by original MOEA/D-DE.

A transition of the execution ratio are shown in Figure 2 to 4. In these figures, vertical axis represents each individual and horizontal axis is generations. Figure 2 shows escape mechanism is effective in middle part of the search and Figure 3 shows convergence mechanism is conducted in last part of the search.

## 4. Conclusions

In this paper, a new local search method using search history in EMO, SPLASH was proposed. Experimental results showed that the effectiveness of the proposed mechanisms was verified through investigating the influence with or without the proposed mechanisms. Escape mechanism is effective in middle part of the search and convergence mechanism is in last part of the search. In future work, we will research about dynamic parameter settings and how to store search history can be effective in problem having parameter dependence.

## References


[1] H. Ishibuchi and T. Murata. Multi-objective genetic local search algorithm. In *Evolutionary Computation, 1996., Proceedings of IEEE International Conference on*, pp. 119–124, May 1996.

[2] Laura Barbulescu, Jean paul Watson, and L. Darrell Whitley. Dynamic representations and escaping lo-

Table 1: HV of 2 objectives ($\times 10^1$)

| | MOEA/D-DE | SPLASH |
|---|---|---|
| WFG1 | 0.570 | **0.589** |
| WFG2 | **1.143** | 1.141 |
| WFG3 | **1.095** | 1.093 |
| WFG4 | 0.844 | **0.864** |
| WFG5 | **0.815** | 0.814 |
| WFG6 | **0.812** | 0.801 |
| WFG7 | **0.868** | **0.868** |
| WFG8 | **0.813** | 0.808 |
| WFG9 | 0.832 | **0.838** |

Table 2: HV of 5 objectives ($\times 10^4$)

| | MOEA/D-DE | SPLASH |
|---|---|---|
| WFG1 | 0.400 | **0.432** |
| WFG2 | 1.001 | **1.026** |
| WFG3 | 0.677 | **0.679** |
| WFG4 | 0.653 | **0.788** |
| WFG5 | 0.651 | **0.732** |
| WFG6 | 0.697 | **0.714** |
| WFG7 | 0.687 | **0.783** |
| WFG8 | 0.525 | **0.654** |
| WFG9 | 0.555 | **0.576** |

Table 3: HV of 8 objectives ($\times 10^7$)

| | MOEA/D-DE | SPLASH |
|---|---|---|
| WFG1 | 1.570 | **2.000** |
| WFG2 | 3.387 | **3.420** |
| WFG3 | 2.084 | **2.088** |
| WFG4 | 1.586 | **1.801** |
| WFG5 | 1.609 | **1.797** |
| WFG6 | 1.914 | **2.187** |
| WFG7 | 1.755 | **1.941** |
| WFG8 | 1.262 | **1.328** |
| WFG9 | 1.464 | **1.537** |

Table 4: Result of WFG4

| | HV ($\times 10^1$) | | | | stagnation count | | | |
|---|---|---|---|---|---|---|---|---|
| | MOEA/D-DE | SPLASH | escape | convergence | MOEA/D-DE | SPLASH | escape | convergence |
| WFG4 | 0.844 | **0.864** | 0.852 | 0.861 | 91114 | 87262 | 90893 | **87237** |

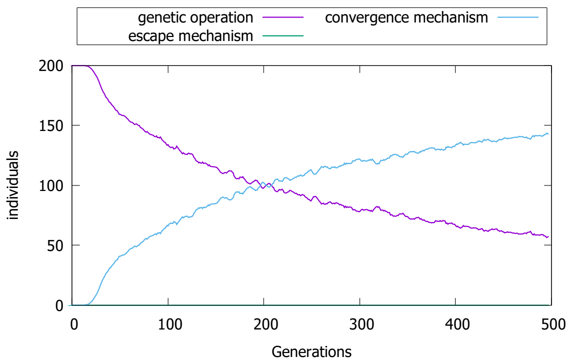


Figure 3: A transition of the execution ratio in 500 generations (MOEA/D-DE with only convergence mechanism)

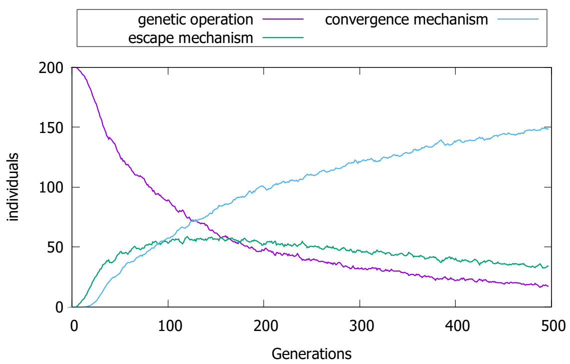


Figure 4: A transition of the execution ratio in 500 generations (MOEA/D-DE with SPLASH)

cal optima: Improving genetic algorithms and local search. In *In AAAI/IAAI*, pp. 879–884, 2000.

[3] Alexandre Belloni, Tengyuan Liang, Hariharan Narayanan, and Alexander Rakhlin. Escaping the local minima via simulated annealing: Optimization of approximately convex functions. *CoRR*, Vol. abs/1501.07242, , 2015.

[4] D.M. Jaeggi, G.T.Parks, T.Kipouros, and P.J.Clarkson. The development of a multi-objective Tabu Search algorithm for continuous optimisation problems. *European Journal of Operational Research 185*, pp. 1192–1212, 2008.

[5] H. Li and Q. Zhang. Multiobjective Optimization Problems with Complicated Pareto Sets, MOEA/D and NSGA-II. *IEEE Trans on Evolutionary Computation, vol. 12, no 2*, pp. 284–302, 2009.

[6] Simon Huband, Luigi Barone, Lyndon While, and Phil Hingston. A Scalable Multi-objective Test Problem Toolkit. *Evolutionary Multi-Criterion Optimization*, pp. 280–295, 2005.

[7] H. Ishibuchi, H. Masuda, and Y. Nojima. Pareto fronts of many-objective degenerate test problems. *IEEE Transactions on Evolutionary Computation*, Vol. PP, No. 99, pp. 1–1, 2015.

[8] E. Zitzler and L. Thiele. Multiobjective Optimization Using Evolutionary Algorithms - A Comparative Case Study. *Parallel Problem Solving from Nature - PPSN-V*, pp. 292–301, 1998.