Hybridization of Genetic Algorithm and Local Search in Multiobjective Function Optimization: Recommendation of GA then LS

Ken Harada
Tokyo Institute of Technology
4259 Nagatsuta-cho, Midori-ku
Yokohama-shi, Kanagawa, Japan
ken@fe.dis.titech.ac.jp

Kokolo Ikeda
Kyoto University
Nihon-matsu-cho, Yoshida, Sakyou-ku
Kyoto-shi, Kyoto, Japan
kokolo@media.kyoto-u.ac.jp

Shigenobu Kobayashi
Tokyo Institute of Technology
4259 Nagatsuta-cho, Midori-ku
Yokohama-shi, Kanagawa, Japan
kobayasi@dis.titech.ac.jp

ABSTRACT

Hybridization with local search (LS) is known to enhance the performance of genetic algorithms (GA) in single objective optimization and have also been studied in the multiobjective combinatorial optimization literature. In most such studies, LS is applied to the solutions of each generation of GA, which is the scheme called "GA with LS" herein. Another scheme, in which LS is applied to the solutions obtained with GA, has also been studied, which is called "GA then LS" herein. It seems there is no consensus in the literature as to which scheme is better, let alone the reasoning for it. The situation in the multiobjective function optimization literature is even more unclear since the number of such studies in the field has been small.

This paper, assuming that objective functions are differentiable, reveals the reasons why GA is not suitable for obtaining solutions of high precision, thereby justifying hybridization of GA and LS. It also suggests that the hybridization scheme which maximally exploits both GA and LS is GA then LS. Experiments conducted on many benchmark problems verified our claims.

Categories and Subject Descriptors

I.2 [**Artificial Intelligence**]: Problem Solving, Control Methods, and Search

General Terms

Algorithms, Performance, Experimentation

Keywords

Multi-objective optimization, Genetic algorithms, Local search, Hybridization

1. INTRODUCTION

The problem of simultaneously optimizing multiple conflicting objectives is called multiobjective optimization. The objective of multiobjective optimization is to find a set of Pareto-optimal solutions, which are the solutions to which no other feasible solutions are superior in all objective functions.

Genetic algorithms (GA) have been shown to be effective in solving multiobjective optimization problems and have been studied extensively [3]. GA maintains a set of solutions and converges it progressively toward Pareto-optimal solutions, exploiting the information about the objective function landscapes that the set of solutions carry with them.

Hybridization with local search (LS) is known to enhance the performance of GA in single objective optimization. Assuming that the same is true for multiobjective optimization, the hybridization has also been applied to multiobjective optimization [13]. In most of such studies, LS is applied to the solutions of each generation of GA. There have been a relatively small number of such studies for multiobjective function optimization whose variables are real-valued. Bosman et al. [1] propose an LS method called Combined Objectives Repeated Line-search (CORL) for multiobjective function optimization and conduct experiments comparing GA and GA with LS on the ZDT benchmark problems [18, 3]. They conclude that it seems LS contributes marginally to enhancing the performance of GA. However, we believe that it is too early to dismiss the effectiveness of hybridization because the benchmark problems on which the experiments were conducted are rather peculiar with respect to the landscapes of the objective functions, shapes of their Pareto-optimal solutions, and their constraints.

Assuming that objective functions are differentiable, this paper presents the reasons why GA is not suitable for obtaining solutions of high precision in multiobjective function optimization, which justifies the hybridization of GA and LS. It then suggests that the hybridization scheme which maximally exploits the advantages of both GA and LS is GA then LS, which is the hybridization scheme in which LS is applied to solutions obtained with GA. These claims are verified through experiments on various benchmark problems.

Section 2 reviews the basics of multiobjective optimization and surveys the studies of hybridization. Section 3 examines the behaviors of GA that is idiosyncratic to multiobjective function optimization and explains the reasons why it is hard for GA to find solutions of high precision, and the hybridization scheme which maximally exploits the advantages of both GA and LS is GA then LS. Section 4 presents the experimental results confirming these claims, and Section 5 concludes this paper.

2. MULTIOBJECTIVE OPTIMIZATION AND EXISTING APPROACHES

2.1 Multiobjective Optimization

Let the dimensions of the real-valued variable space and the objective space be N and M, respectively. Denote a solution by $\boldsymbol{x} = (x_1, x_2, \ldots, x_N)^T \in \mathbb{R}^N$, the vector of objective functions by $\boldsymbol{f} = (f_1, f_2, \ldots, f_M)^T$, the feasible region by $S \subset \mathbb{R}^N$, and the image of $\boldsymbol{x}$ in the objective space by $\boldsymbol{f}(\boldsymbol{x}) \in \mathbb{R}^M$. *Multiobjective function optimization problems* can be formulated as:

$$\text{Minimize } f_i(\boldsymbol{x}) \ (i = 1, 2, \ldots, M), \text{ subject to } \boldsymbol{x} \in S.$$

f_i are assumed to be differentiable herein.

If the following holds for some solutions $\boldsymbol{x_1}, \boldsymbol{x_2} \in S$, $\boldsymbol{x_1}$ is said to be *superior* to $\boldsymbol{x_2}$, which is denoted by $\boldsymbol{x_1} \succ \boldsymbol{x_2}$:

$$\begin{aligned} &\forall i \in \{1, \ldots, M\},\ f_i(\boldsymbol{x}_1) \leq f_i(\boldsymbol{x}_2) \\ &\wedge \exists i \in \{1, \ldots, M\},\ f_i(\boldsymbol{x}_1) < f_i(\boldsymbol{x}_2). \end{aligned}$$

If there is no feasible solution $\boldsymbol{x}'$ such that $\boldsymbol{x}' \succ \boldsymbol{x}$, the solution $\boldsymbol{x}$ is called a *Pareto-optimal solution*. There are often multiple Pareto-optimal solutions. If there is no solution $\boldsymbol{x}'$ such that $\boldsymbol{x}' \succ \boldsymbol{x}$ in the feasible ε-vicinity of a solution $\boldsymbol{x}$, $\boldsymbol{x}$ is called a *local Pareto-optimal solution*. Local Pareto-optimal solutions henceforth denote those that are *not* Pareto-optimal.

The objective of multiobjective optimization is to find a set of solutions that approximates Pareto-optimal solutions. Hence, the performances of multiobjective optimization methods are evaluated based on how well the solutions obtained with them approximate Pareto-optimal solutions. Specifically, they are evaluated with respect to *proximity* and *diversity* of the solutions obtained with them, often in the objective space [3, 13]

Proximity consists of two components: *Overcoming of local Pareto-optimal solutions*, which means that solutions are not trapped around local Pareto-optimal solutions, and *high precision*, which means that the distances between solutions and Pareto-optimal solutions are small.

Diversity consists of two components [3, 13]: *Extent*, which means that solutions are extended toward the periphery of Pareto-optimal solutions, and *distribution*, which means that solutions are distributed uniformly across Pareto-optimal solutions.

2.2 Directions Defined by the Gradients of Objective Functions

Denote by $\nabla f_i(\boldsymbol{x})$ $(i = 1, 2, \ldots, M)$ the gradients of objective functions at a solution $\boldsymbol{x}$. *Descent directions* [9] are defined as directions that satisfy Eq. (1).

$$\boldsymbol{d} \cdot (-\bar{\nabla} f_i(\boldsymbol{x})) \geq 0 \ (i = 1, 2, \ldots, M), \tag{1}$$

where $\bar{\nabla} f_i(\boldsymbol{x}) = \nabla f_i(\boldsymbol{x}) / ||\nabla f_i(\boldsymbol{x})||$. There are often multiple descent directions.

Pareto descent directions are defined as descent directions to which no other descent directions are superior in improving all objective functions [9]. A descent direction $\boldsymbol{d}$ is a Pareto descent direction iff $\boldsymbol{d}$ can be expressed as a convex combination of the steepest descent directions of objective functions. There are often multiple Pareto descent directions.

Ascent directions are defined as directions in which solutions can be moved to worsen all objective functions and are exactly the opposite to descent directions.

2.3 Existing Approaches

2.3.1 Genetic Algorithm

GA maintains a set of solutions and efficiently converges it progressively toward Pareto-optimal solutions [3]. GA consists mainly of the following two components.

Crossover operator: A crossover operator generates new offspring solutions from parent solutions that are expected to be superior to those of the current generation. Older crossover operators represent variables with bitstrings, which limits the achievable precision of solutions. Modern crossover operators such as SBX [5], UNDX [16], SPX [10], and PCX [4] represent variables explicitly as real-valued, and they have proven to be effective in enhancing precision.

Selection operator: A selection operator consists of mating selection and survival selection [19]. Mating selection determines which solutions to participate in crossover, and survival selection determines which solutions are to be discarded and which solutions are to be carried over to the next generation.
Survival selection is considered to be particularly important in multiobjective optimization. The components that are considered to be crucial in designing high performance survival selection such as NSGA-II [6] and SPEA2 [19] are *ranking* and *niching*. Ranking ranks solutions according to how good the solutions are with respect to other solutions and assign fitness to these solutions based on their ranks. Since solutions around local Pareto-optimal solutions are inferior to those around Pareto-optimal solutions, ranking can remove the former. Hence, ranking plays a crucial role in overcoming local Pareto-optimal solutions.
When it is necessary to determine which solutions are to be discarded among those with the same fitness, niching culls those in the area in the objective space or variable space where solutions are concentrated. An appropriate niching is expected to bring about solutions that are distributed uniformly over the whole Pareto-optimal solutions. Therefore, niching plays an important role in obtaining diverse solutions.

Based on the above-mentioned design of GA, it is reasonable to assume that GA allows for overcoming of local Pareto-optimal solutions and obtaining diverse solutions. It is often assumed, in addition, that GA with an appropriate crossover operator and selection operator finds solutions of high precision [2]. However, it is not immediately apparent whether that is the case.

In fact, it can be inferred that it is *not* the case from some recent papers. For one thing, the failure of Evolution Strategies (ES) in obtaining solutions of high precision demonstrated in [14] can reasonably be extended to GA. In addition, defining *deterioration* as the situation in which some of the solutions of a generation are inferior to some solutions of the previous generations, Laumanns et al. [15] have confirmed that deterioration occurs when NSGA-II is applied to multiobjective combinatorial optimization problems, which translates to the failure of GA in obtaining solutions of high precision.

2.3.2 *Hybridization of GA and LS*

Hybridization with LS is known to enhance the performance of GA in single objective optimization [1], and it has been applied to multiobjective optimization, chiefly to multiobjective combinatorial optimization [13]. In most such studies, LS is applied to the solutions of each generation of GA, which is called "GA with LS" herein. The effectiveness of GA with LS has been demonstrated in many papers such as [11].

There is another scheme in which LS is applied to solutions obtained with GA, which is called "GA then LS" herein. The number of studies of GA then LS is very limited, and they give conflicting results: Talbi et al. [17] conclude that LS contributes marginally to enhancing the performance of GA, and Goel et al. [8] conclude that GA then LS performs better than GA with LS. However, despite the conclusion of [8], its experimental results show that some solutions obtained with GA with LS are actually better than those obtained with GA then LS. Hence neither scheme can be declared better than the other.

For multiobjective function optimization, there are much fewer studies of hybridization to date [1]. For one thing, Bosman et al. [1] propose an LS method called Combined-Objectives Repeated Linesearch (CORL) and compare the performances of GA and GA with LS using CORL. However, they came to a rather disappointing conclusion that LS contributes little to enhancing the performance of GA and it may even obstruct GA.

In summary, although the effectiveness of hybridization of GA and LS has been empirically demonstrated, there needs to be more studies as to which hybridization scheme is better and the reasoning behind it, particularly for multiobjective function optimization.

3. THE BEHAVIORS OF GA AND LS

3.1 The Behaviors of GA

As described in Subsection 2.3.1, GA has the abilities to overcome local Pareto-optimal solutions and to obtain diverse solutions, by design. Therefore, we henceforth focus on assessing GA's ability to obtain solutions of high precision. In order to investigate it, it is henceforth assumed that domination relation among offspring solutions can be ignored, and the increase or decrease of objective functions can be estimated with the gradients of respective objective functions.

In assessing GA's ability to obtain solutions of high precision, it is essential to examine how GA operates during the later phase of its search. During the later phase of the search, it is likely that most solutions are of the best rank and the number of these solutions are likely to exceed the population size [3]. Therefore, during that phase, ranking basically discards offspring solutions in the ascent directions of any of the parent solutions, and niching culls solutions in concetrated areas among those which survived ranking. In the following, the reasons why these ranking and niching

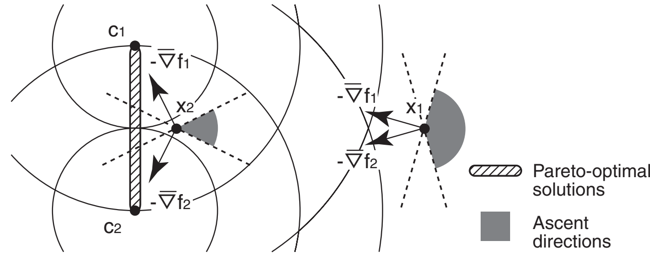


Figure 1: Paret descent directions, descent directions, and ascent directions at solutions far from and near the Pareto-optimal solutions: ***Ascent directios are a little less than half of all directions for solutions far from the Pareto-optimal solutions, such as x_1, and limited for those near the Pareto-optimal solutions, such as x_2.***

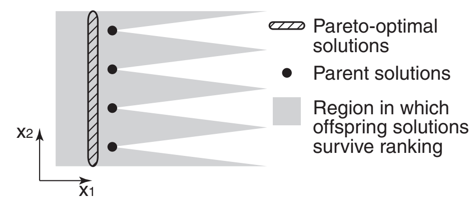


Figure 2: Region in the variable space in which offspring solutions survive ranking in the case parent solutions are near Pareto-optimal solutions: ***As parent solutions approach the Pareto-optimal solutions, more offspring solutions farther from the Pareto-optimal solutions survive ranking.***

can prevent solutions from approaching Pareto-optimal solutions are discussed.

The effects of ranking in the later phase of GA's search: Since ranking basically discards offspring solutions in the ascent directions of parent solutions, it is crucial to examine how ascent directions of a solution change as it approaches the Pareto-optimal solutions.

Consider the solutions $\boldsymbol{x}_1$ and $\boldsymbol{x}_2$ of a 2-variable-2-objective problem shown in Fig. 1. For solutions far from the Pareto-optimal solutions such as $\boldsymbol{x}_1$, ascent directions are a little less than half of all directions [2]. On the other hand, for solutions near the Pareto-optimal solutions such as $\boldsymbol{x}_2$, ascent directions are limited [2]. The ascent directions decreases as the solution approaches the Pareto-optimal solutions.

During the later phase of GA's search, the parent solutions are generally near the Pareto-optimal solutions, as depicted in Fig. 2. Since the ascent directions of these parent solutions are limited, ranking discards offspring solutions in the limited ascent directions. The region in which offspring solutions survive ranking when parent solutions are near the Pareto-optimal solutions are shown in Fig. 2. The figure shows that not only the offspring solutions nearer to the Pareto-optimal solutions than the parent solutions survive ranking, but also those farther from the Pareto-optimal solutions do. Note that the offspring solutions nearer to the Pareto-optimal solutions and parent solutions are concentrated around the Pareto-optimal solutions while those farther from the Pareto-optimal solutions are sparse[1], and the same is expected to be true in the objective space.

The effects of niching in the later phase of GA's search: Consider applying niching to solutions which survived ranking in the later phase of GA's search. Since parent solutions and some offspring solutions are concentrated around Pareto-optimal solutions, they are more likely to be removed in niching. On the other hand, since offspring solutions far from Pareto-optimal solutions are sparse, they are less likely to removed in niching. Therefore, although solultions which survived ranking may be close to the Pareto-optimal solutions on average, niching can push them back away from the Pareto-optimal solutions on average.

Therefore, solutions after survival selection, on average, do not move closer to Pareto-optimal solutions during the later phase of GA's search, which means that it is difficult to obtain solutions of high precision with GA.

3.2 The Behaviors of LS

Suppose that solutions have already overcome local Pareto-optimal solutions with the help of some global optimization method such as GA. Solutions can be moved steadily closer to Pareto-optimal solutions by LS methods that move them in descent directions. Therefore, LS is suitable for obtaining solutions of high precision. However, LS itself is unable to overcome local Pareto-optimal solutions or bring about diverse solutions.

When choosing an LS method for hybridization with GA, it is desirable to choose one that moves solutions in feasible Pareto descent directions or descent directions, as appropriate, thereby efficiently improving all objective functions simultaneously. A recently proposed LS method, Pareto Descent Method (PDM) [9], meets these requirements and is one appropriate choice for hybridization. An overview of PDM is given in the appendix.

3.3 The Behaviors of Hybridized GA

Considerations given in Subsection 3.1 and 3.2 imply that GA and LS are complementary to each other in multiobjective function optimization, which translates to the justification for hybridization of GA and LS. As mentioned earlier, there are two hybridization schemes: "GA with LS" which applies LS to solutions of each generation of GA and "GA then LS" which applies LS to solutions obtained with GA. The expected behaviors of these hybridization schemes are described below.

GA with LS: LS can be applied to solutions either before or after survival selection. Suppose applying LS to solutions before survival selection. It is computationally prohibitive to apply LS to all solutions since the number of solutions before survival selection is large. Therefore, LS has to be applied to some portion of the solutions, resulting in the situation described in Subsection 3.1 in which some solutions are concentrated around Pareto-optimal solutions while others are away from them. Hence, the effects of LS are cancelled by the following survival selection. For this reason, LS is henceforth assumed to be applied after survival selection.
In the earlier phase of search of GA with LS, LS can accelerate the search of GA by driving solutions closer to (local) Pareto-optimal solutions. At the same time, LS may trap solutions around local Pareto-optimal solutions, which may not be resolved by ranking when complex local Pareto-optimal solutions exist.
Suppose that local Pareto-optimal solutions have been overcome and solutions are near Pareto-optimal solutions in the later phase. If the crossover operator generates solutions that are farther from the Pareto optimal solutions than the parent solutions, the situation described in Subsection 3.1 occurs and hence refining of the solutions stagnates.

GA then LS: It has been demonstrated in many earlier studies that GA overcomes local Pareto-optimal solutions and finds diverse solutions, which carries over directly to the first phase of GA then LS. In the second phase, LS refines the solutions obtained with GA thereby achieving high precision. There might be a concern as to whether LS may negatively influence the diversity of solutions that have already been achieved with GA. Suppose choosing an LS method that moves solutions in Pareto descent directions for hybridization. Since Pareto descent directions are directions to which no other directions are superior in improving all objective functions, the solutions are expected to be moved almost perpendicularly to the curves or surfaces that Pareto-optimal solutions constitute. Therefore, when such an LS method is employed, LS does not negatively influence the diversity.

Recommendation of GA then LS: Although GA with LS is expected to outperform GA on problems without local Pareto-optimal solutions, it may suffer from the conflict between LS and survival selection, which undermines the precision of solutions. In addition, LS may inadvertently drive solutions toward local Pareto-optimal solutions. By contrast, GA with LS can overcome local Pareto-optimal solutions with GA and refine solutions with LS, without suffering from the conflict between LS and survival selection. Hence, the hybridization scheme which maximally exploits the advantages of both GA and LS is GA then LS.

4. EXPERIMENTS

4.1 Aims and Experiment Setups

Subsection 3.1 and 3.2 argued that GA is not suitable for obtaining solutions of high precision, and Subsection 3.3 described the expected behaviors of GA with LS and GA then LS. This section verifies these claims through 2 experiments.

Experiment 1 compares GA and LS on benchmark problems without local Pareto-optimal solutions and verifies that GA is not suitable for obtaining solutions of high precision compared to LS. Experiment 2 compares GA, GA with LS, and GA then LS on various benchmark problems and verifies that GA then LS performs the best overall.

4.1.1 Benchmark Problems

Experiments are conducted on the following benchmark problems.
BNH [3] 2-variable-2-objective. Has no local Pareto-optimal solutions. The Pareto-optimal solutions form a kinked line segment in the variable space and part of it is on the feasible region boundary.
SPH-(3,30) [14, 19] 30-variable-3-objective. Has no local Pareto-optimal solutions. The Pareto-optimal solutions are

[1] When Pareto-optimal solutions are on feasible region boundaries, it is not that ascent directions are limited. Hence, the situation described in the text is less likely to occur.

on a plane in the variable space and convex in the objective space.

MED1(0.5) [9] 30-variable-3-objective. Has no local Pareto-optimal solutions. The Pareto-optimal solutions are on a plane in the variable space and non-convex in the objective space.

POL [3] 2-variable-2-objective. Has local Pareto-optimal solutions. The Pareto-optimal solutions form two lines in the variable space with part of it on the feasible region boundary.

FON30 [3] 30-variable-2-objective. Has no local Pareto-optimal solutions. The Pareto-optimal solutions form a straight line segment in the variable space and is non-convex in the objective space.

KUR, KUR30 [3] 3- and 30-variable-2-objective, respectively. Has local Pareto-optimal solutions. The Pareto-optimal solutions form complex line segments, most of which are generally parallel to some axes of the variables.

4.1.2 Performance Metrics

Two commonly used performance metrics, generational distance (GD) and D1R, are used and these metrics are averaged over 10 trials.

GD is defined as the mean of the distances from each solution to its nearest Pareto-optimal solution in the normalized objective space [3]. The smaller GD is, the nearer the solutions are to the Pareto-optimal solutions on average. Hence, GD measures proximity.

D1R is defined as the mean of the distances from each Pareto-optimal solution to its nearest solution in the normalized objective space [12]. The smaller D1R is, the better Pareto-optimal solutions are approximated. Hence, D1R measures both proximity and diversity.

Note that the Pareto-optimal solutions are necessary for evaluating GD and D1R. It is customary to run GA with a larger population size and with more generations and assume the solutions obtained with the GA are Pareto-optimal. The same strategy is employed for the experiments in this paper.

4.1.3 Experiment Setups

For each trial, the same initial solutions are used for all methods. The population size of GA is 100. 50 pairs of parent solutions are chosen for mating in each generation. 20 offspring solutions are generated for each pair since preliminary experiments revealed that increasing the number generally improved precision.

Crossover operator: Preliminary experiments comparing crossover operators SBX, UNDX, SPX, and PCX combined with modified SPEA2, which will be explained shortly, showed that the best performing crossover operator is problem-specific. However, since UNDX exhibited relatively good performance for many problems, UNDX with its parameter values as suggested in [16] is used in the experiments. Since UNDX, SPX, and PCX have similar design principles, the tendency exhibited in the results of experiments using UNDX is expected to be observed when using SPX or PCX as well.

Selection operator: NSGA-II [6] and SPEA2 [19] are known to exhibit good performances. However, it has been demonstrated in [7] that SPEA2 finds more uniformly distributed solutions than NSGA-II does. However, the original SPEA2 requires substantial computation and memory space, and it often runs out of memory. Therefore, modified SPEA2, which approximates crowdedness around a solution with the Euclidean distance between the solution and the other solution nearest to it in the normalized objective space, is used in the experiments. Since NSGA-II, SPEA2, and modified SPEA2 all have the ranking and niching mechanisms described in Subsection 2.3.1, the tendency exhibited in the results of experiments using modified SPEA2 are expected to be observed when using NSGA-II or SPEA2 as well.

Local search: Pareto descent method (PDM) is used. Gradients are approximated by forward difference with the difference of 10^{-4}. In order for PDM to sufficiently approximate the complete convex cone of feasible descent directions, 40 combination weights are randomly drawn for the direction calculation. For line search, golden section method is used, with the basic search segment length of 10^{-2}, the maximum number of extension of the segment of 20, and the number of iteration of 20. A solution is assumed to be on a feasible region boundary if the distance between them is less than $10^{-2} \times \tau^{20}$, where τ is the golden ratio. In GA with LS, one iteration of LS is applied to each solution after survival selection. In GA then LS, GA is overtaken by LS when half of the total number of function evaluations is spent on GA.

4.2 Experiment 1: Results and Discussion

For problems without local Pareto-optimal solutions, overcoming of local Pareto-optimal solutions is unnecessary. Therefore, comparing GA and LS on such problems reveals which is suitable for obtaining solutions of high precision. Experiment 1 verifies that GA is not suitable for obtaining solutions of high precision compared to LS by comparing GA and LS on BNH, SPH-(3,30), and MED1(0.5).

The transitions of GD and D1R as GA and LS are applied to these benchmark problems are shown in Fig. 3 and 4, respectively. LS performed better than GA regarding GD on all benchmark problems. On the other hand, LS performed no better than GA regarding D1R. Therefore, it has been confirmed that good proximity, which in this case is high precision, is better achieved by LS than GA while good diversity is better achieved by GA. Hence, GA and LS are complementary to each other in multiobjective function optimization, which justifies hybridization of GA and LS.

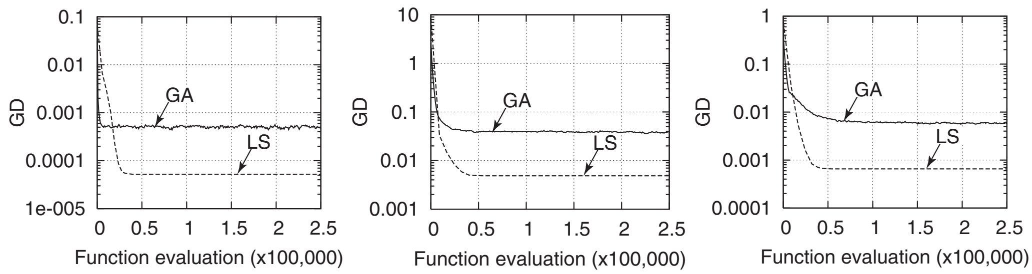
4.3 Experiment 2: Results and Discussion

Experiment 2 verifies that GA then LS outperforms GA and GA with LS by comparing them on various benchmark problems.

The transitions of GD when GA, GA with LS, and GA then LS are applied to the benchmark problems are shown in Fig. 5. The transitions of D1R are omitted since all methods exhibited similar behaviors regarding D1R. The results regarding GD can be classified in the following three cases:

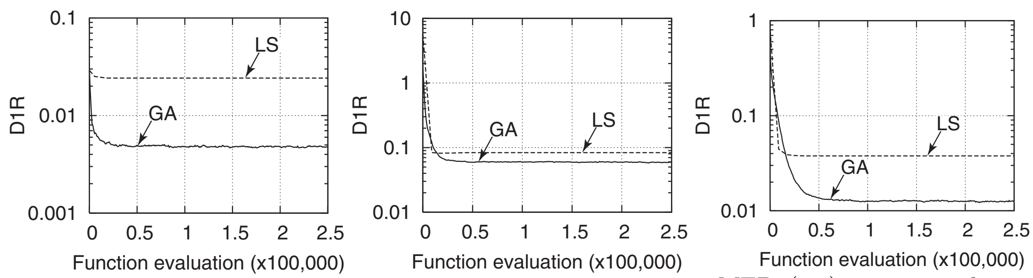
GA then LS $\succ$ GA with LS $\succ$ GA: This relationship was observed on BNH and SPH-(3,30). GA with LS performed worse than GA then LS because refinement by LS is hindered by survival selection as explained in Subsection 3.3.

GA then LS $\sim$ GA with LS $\succ$ GA: This relationship was observed on FON30 and KUR. GA then LS and GA with LS performed equally well on FON30 since its Pareto-optimal



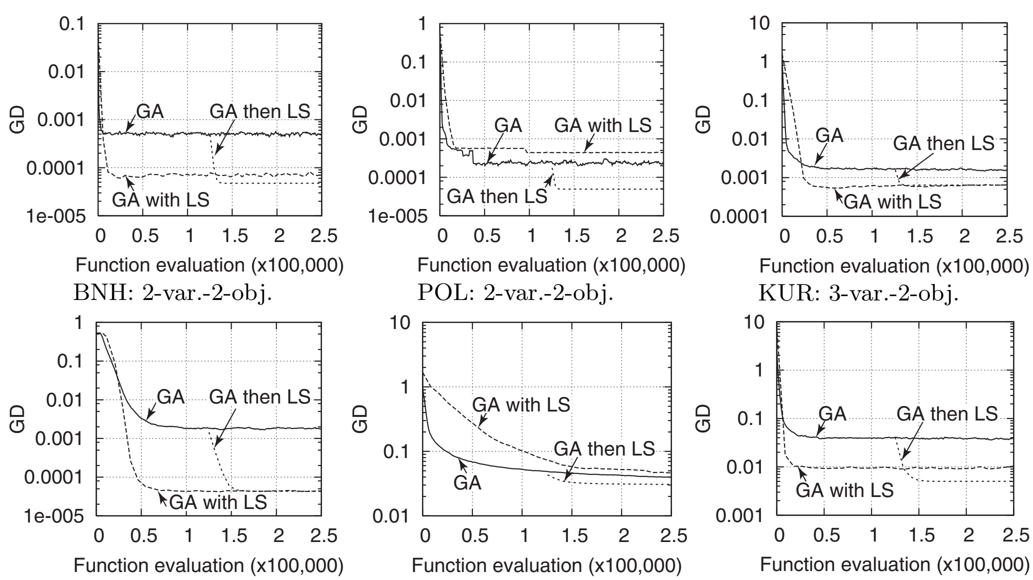
BNH: 2-var.-2-obj. SPH-(3,30): 30-var.-3-obj. MED1(0.5): 30-var.-3-obj.

Figure 3: Transitions of GD when GA and local search are applied to benchmark problems without local Pareto-optimal solutions



BNH: 2-var.-2-obj. SPH-(3,30): 30-var.-3-obj. MED1(0.5): 30-var.-3-obj.

Figure 4: Transitions of D1R when GA and local search are applied to benchmark problems without local Pareto-optimal solutions



BNH: 2-var.-2-obj. POL: 2-var.-2-obj. KUR: 3-var.-2-obj.

FON30: 30-var.-2-obj. KUR30: 30-var.-2-obj. SPH-(3,30): 30-var.-3-obj.

Figure 5: Transition of GD when GA, GA with LS, and GA then LS are applied to various benchmark problems

solutions form a straight line segment, and crossover operators such as UNDX, SPX, and PCX are unlikely to generate offspring solutions that are farther from the Pareto-optimal solutions than parent solutions. Although KUR has local Pareto-optimal solutions, GA with LS and GA then LS performed equally well since ranking removed solutions which LS trapped around local Pareto-optimal solutions, thereby overcoming local Pareto-optimal solutions, as described in Subsection 3.3.

GA then LS $\succ$ GA $\succ$ GA with LS: This relationship was observed on KUR30 and POL. GD of GA with LS is *worse* than that of GA since KUR30 and POL have rather complex local Pareto-optimal solutions, and ranking failed to remove solutions trapped around local Pareto-optimal solutions. GA then LS performed best overall.

4.4 Additional Discussion

On the whole : Experiments have confirmed that GA is not suitable for obtaining solutions of high precision. They have also confirmed that, although GA with LS often outperforms GA, it may perform worse than GA on problems with complex local Pareto-optimal solutions. It was also confirmed that GA then LS performs best overall by further improving GD after GA is taken over, and diversity achieved by GA is not worsened by the following LS. Hence, it has been verified that the hybridization scheme which maximally exploits the advantages of both GA and LS is GA then LS.

The timing with which GA is taken over by LS in GA then LS: In order to save the number of function evaluations spent on GA, it is probably the best to switch from GA to LS just after GA overcomes local Pareto-optimal solutions. Since there is generally no way of knowing when local Pareto-optimal solutions are overcome, it seems the when to switch from GA to LS has to be specified by the user. However, the above experimental results showed that GA and LS converged GD with similar speeds. Hence, switching from GA to LS after half of the total function evaluations are spent on GA is one reasonable choice.

5. CONCLUSION

Although hybridization of GA and LS has been studied extensively in the past, the reasoning behind the hybridization and which hybridization scheme to be employed have been unclear. Assuming the differentiability of objective functions, this paper described the reasons why GA is not suitable for obtaining solutions of high precision for multiobjective function optimization, which translates to the necessity of such hybridization of GA and LS. This paper also argued that GA with LS may not be able to overcome local Pareto-optimal solutions on problems with complex local Pareto-optimal solutions and that GA then LS is the hybridization scheme which maximally exploits the advantages of both GA and LS. These claims have been verified through experiments conducted on various benchmark problems.

We reckon that it is crucial to use effective constraint-handling mechanisms in order to further improve the performance of GA and its hybridization with LS. In particular, for problems on which almost all the solutions GA generates violate constraints, such constraint-handling mechanisms are indispensable. Although some effective constraint-handling mechanisms have already been proposed, we are planning to investigate what are the intrinsic requirements for constraint-handling mechanisms to be effective, which may help us design even more effective constraint-handling methods or improve existing methods.

6. ADDITIONAL AUTHORS

Additional authors: Jun Sakuma (Tokyo Institute of Technology, email: jun@fe.dis.titech.ac.jp) and Isao Ono (Tokyo Institute of Technology, email: isao@dis.titech.ac.jp).

7. REFERENCES

[1] P. A. N. Bosman and E. D. de Jong. Exploiting gradient information in numerical multi-objective evolutionary optimization. In H.-G. Beyer, U.-M. O'Reilly, D. V. Arnold, W. Banzhaf, C. Blum, E. W. Bonabeau, E. Cantú-Paz, D. Dasgupta, K. Deb, J. A. Foster, E. D. de Jong, H. Lipson, X. Llora, S. Mancoridis, M. Pelikan, G. R. Raidl, T. Soule, A. M. Tyrrell, J.-P. Watson, and E. Zitzler, editors, *Proceedings of the Genetic and Evolutionary Computation Conference - GECCO-2005*, pages 755–762, New York, NY, USA, 2005. ACM Press.

[2] M. Brown and R. E. Smith. Effective use of directional information in multi-objective evolutionary computation. In E. Cantú-Paz et al., editors, *Proceedings of the Genetic and Evolutionary Computation Conference - GECCO-2003*, volume 2723, pages 778–789. Springer. Lecture Notes in Computer Science, jul 2003.

[3] K. Deb. *Multi-Objective Optimization Using Evolutionary Algorithms*. John Wiley & Sons, Chichester, 2001.

[4] K. Deb, A. Anand, and D. Joshi. A computationally efficient evolutionary algorithm for real-parameter optimization. *Evolutionary Computation Journal*, 10(4):371–395, 2002.

[5] K. Deb and H.-G. Beyer. Self-adaptive genetic algorithms with simulated binary crossover. *Evolutionary Computation Journal*, 9(2):197–221, 2001.

[6] K. Deb, A. Pratab, S. Agrawal, and T. Meyarivan. A fast elitist non-dominated sorting genetic algorithm for multi-mbjective optimization: NSGA-II. *IEEE Transaction on Evolutionary Computation*, 6(2):181–197, 2002.

[7] K. Deb, L. Thiele, M. Laumanns, and E. Zitzler. Scalable multi-objective optimization test problems. In *Proceedings of the Congress on Evolutionary Computation (CEC-2002)*, pages 825–830, 2002.

[8] T. Goel and K. Deb. Hybrid methods for multi-objective evolutionary algorithms. In *Proceedings of the Fourth Asia-Pacific Conference on Simulated Evolution and Learning (SEAL'02)*, pages 188–192, 2002.

[9] K. Harada, J. Sakuma, I. Ono, and S. Kobayashi. Local search for multiobjective optimization: Pareto descent method (in Japanese). In *Proceedings of the 18th Symposium on Decentralized Autonomous Systems of The Society of Instrument and Control Engineers*, pages 345–350, 2006.

[10] T. Higuchi, S. Tsutsui, and M. Yamamura. Theoretical analysis of simplex crossover for real-coded genetic algorithms. In *Proceedings the*

Parallel Problem Solving from Nature (PPSN 2000), pages 365–374, 2000.

[11] H. Ishibuchi and K. Narukawa. Some issues on the implementation of local search in evolutionary multiobjective optimization. In *Proceedings of Genetic and Evolutionary Computation Conference LNCS*, pages 1246–1258, 2004.

[12] J. D. Knowles and D. W. Corne. On metrics for comparing non-dominated sets. In *Proceedings of the 2002 Congress on Evolutionary Computation Conference (CEC02)*, pages 711–716. IEEE Press, 2002.

[13] J. D. Knowles and D. W. Corne. Memetic algorithms for multiobjective optimization: issues, methods and prospects. In N. Krasnogor, J. E. Smith, and W. E. Hart, editors, *Recent Advances in Memetic Algorithms*, pages 313–352. Springer, 2004.

[14] M. Laumanns, G. Rudolph, and H.-P. Schwefel. Mutation control and convergence in evolutionary multi-objective optimization. In Matousek and P. Osmera, editors, *Proceedings of the 7th International Mendel Conference on Soft Computing (MENDEL 2001)*, pages 24–29, 2001.

[15] M. Laumanns, L. Thiele, K. Deb, and E. Zitzler. Combining convergence and diversity in evolutionary multi-objective optimization. *Evolutionary Computation*, 10(3):263–282, 2002.

[16] I. Ono and S. Kobayashi. A real-coded genetic algorithm for function iptimization using unimodal normal distribution crossover. In *7th International Conference on Genetic Algorithms (ICGA7)*, pages 246–253, 1997.

[17] E.-G. Talbi, M. Rahoual, M. H. Mabed, and C. Dhaenens. A hybrid evolutionary approach for multicriteria optimization problems: Application to the flow shop. In *Evolutionary Multi-Criterion Optimization - EMO 2001*, pages 416–428, 2001.

[18] E. Zitzler, K. Deb, and L. Thiele. Comparison of multiobjective evolutionary algorithms: Empirical results. *Evolutionary Computation*, 8(2):173–195, 2000.

[19] E. Zitzler, M. Laumanns, and L. Thiele. SPEA2: Improving the strength pareto evolutionary algorithm for multiobjective optimization. In K. Giannakoglou et al., editors, *EUROGEN 2001, Evolutionary Methods for Design, Optimization and Control with Applications to Industrial Problems*, pages 12–21, 2001.

APPENDIX

A. AN OVERVIEW OF PARETO DESCENT METHOD

Pareto Descent Method (PDM) finds feasible Pareto descent directions or descent directions as appropriate to efficiently improve all objective functions simultaneously. PDM finds these directions by solving linear programming problems. Therefore, it is computationally inexpensive. Another advantage of PDM is that it provides a test of (local) Pareto-optimality.

The algorithm flowchart of PDM is shown in Fig. 6, in which it is assumed that a solution can exist either inside

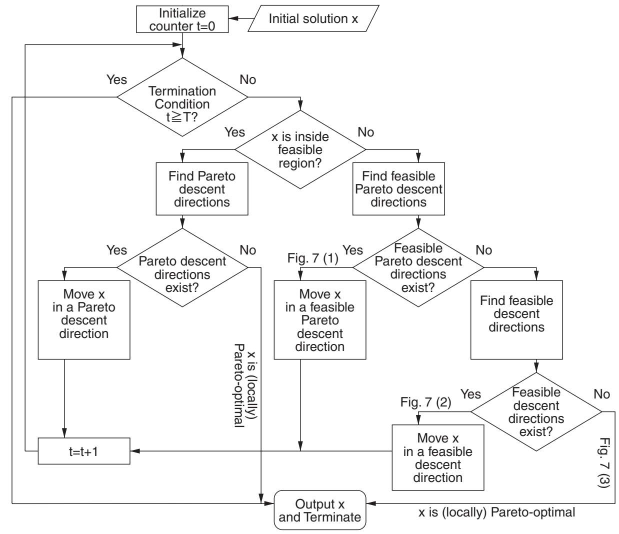


Figure 6: Algorithm flowchart of PDM

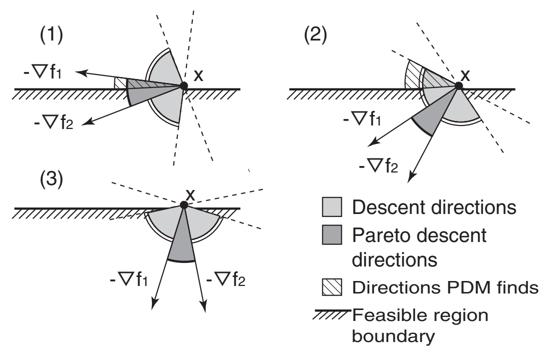


Figure 7: Directions that PDM finds: ***PDM finds appropriate directions in accordance with the feasibility of Pareto descent directions and descent directions***

the feasible region or on the boundary of it. The algorithm comes to an end when the number of times direction calculation and line search are conducted reaches T. Possible relationships among Pareto descent directions, descent directions, and a feasible region boundary for a solution on the boundary are shown in Fig. 7. PDM identifies these cases and appropriately handles each case. When an appropriate direction is found, PDM moves the solution in thus found direction until just before any of the objective functions deteriorate or any of the constraints are violated.