Learning Variable Importance to Guide Recombination

Miyako Sagawa*, Hernán Aguirre*, Fabio Daolio*, Arnaud Liefooghe†
Bilel Derbel†, Sébastien Verel‡ and Kiyoshi Tanaka*
*Faculty of Engineering, Shinshu University, Nagano, Japan
Email: 15st204e@shinshu-u.ac.jp
†Univ. Lille, CNRS, UMR 9189 – CRIStAL / Inria Lille-Nord Europe, France
‡Univ. Littoral Côte d'Opale, LISIC, France

***Abstract*—In evolutionary multi-objective optimization, variation operators are crucially important to produce improving solutions, hence leading the search towards the most promising regions of the solution space. In this paper, we propose to use a machine learning modeling technique, namely random forest, in order to estimate, at each iteration in the course of the search process, the importance of decision variables with respect to convergence to the Pareto front. Accordingly, we are able to propose an adaptive mechanism guiding the recombination step with the aim of stressing the convergence of the so-obtained offspring. By conducting an experimental analysis using some of the WFG and DTLZ benchmark test problems, we are able to elicit the behavior of the proposed approach, and to demonstrate the benefits of incorporating machine learning techniques in order to design new efficient adaptive variation mechanisms.**

I. Introduction

When dealing with a multi-objective optimization problem (MOP), evolutionary multi-objective optimization (EMO) algorithms seek an approximation of the set of optimal trade-offs between the objectives, known as the set of Pareto optimal solutions. One of the main challenges in EMO is to shorten the time required to identify such Pareto optimal solutions, especially for large-scale and real-time MOPs. When the time required for evaluating the objectives, and the time required for the search process is limited, the discovery of Pareto optimal solutions as soon as possible is substantially beneficial. The time to find Pareto optimal solution is closely related with the search performance of EMO algorithms. This performance is mainly determined by the effectiveness of EMO operators, such as selection, recombination (crossover) and mutation. Recombination and mutation operators appear to be particularly important, since their purpose is to produce new candidate solutions for the next iteration. If it is possible to generate an improving solution by means of recombination and mutation, the search process can proceed. Otherwise the solution search stagnates. Therefore, it is important to increase the efficiency of generating improving solutions.

The objective functions defining a MOP may be characterized by a large number of decision variables. Among those variables, some could be more relevant to convergence than others. This becomes even more relevant in many-objective optimization, where different subsets of variables may influence convergence towards different objective subspaces. The acquisition of such a knowledge is particularly valuable to the decision maker. In design optimization, for instance, besides producing optimal solutions, the decision maker is interested in understanding the correlation between feasible changes in the decision space and trade-offs in the objective space. This allows her/him to know how to build alternative designs. In real-world applications, this kind of analysis is often done offline to allow for a better problem understanding.

In this paper, we focus on bi-objective MOPs, and we propose an original way to *learn online* what variables favor Pareto improvements. We then bias variation operators accordingly in order to find Pareto optimal solutions as soon as possible, and hence improve the algorithm convergence. More precisely, we use random forest, a machine learning algorithm, in order to perform a regression of the Pareto rankings, in terms of non-dominated sorting, over decision variables at each iteration. From fitting the statistical regression model, we obtain estimates of the variable importance. We later use this knowledge to select the variables that will undergo variation. Although both recombination and mutation are important variation operators, in this work we focus on recombination. Thus, in the proposed approach, the EMO algorithm focuses on searching the variable space by crossing more often the variables that affect convergence to the Pareto front. In [1] the authors proposed an algorithm based on decision variable analyses including control property analysis and variable linkage analysis. They applied interdependent analysis between pairs of variables, divide the problem into low dimensional sub-problems, and solved them independently. In this paper, we identify variables baced on the variable importance towards improving pareto ranking.

We use AεSεH [2], [3] as a baseline algorithm and compare its performance against versions that include the proposed informed recombination approach by biasing the variable selection from the conventional SBX recombination operator. We experiment the corresponding algorithm with numerical functions from the WFG benchmark suite [4] and DTLZ benchmark suite [5]. In these MOPs, variables are divided in two subsets. One is related to convergence, whereas the other one is related to diversity of solutions in objective space. Although this sharp separation of variables does not necessarily hold in real-world applications, where the interac-

tion of several variables usually affect both convergence and diversity, these kind of problems are an appropriate benchmark to investigate the effectiveness of models that learn the relative importance of variables for promoting convergence.

The remainder of the paper is organized as follows. In Section II, we present the proposed approach for computing variable importance within the EMO search process. In Section III, we give the experimental setting of our analysis. In Section IV, we provide the experimental results of the proposed approach on some WFG benchmark functions. The last section concludes the paper and gives further insights for future research.

II. Learning Variable Importance

A. Overall Concept

We intend to learn which variables affect convergence of solutions to the Pareto front in order to recombine them more often, and eventually find Pareto optimal solutions faster. In this paper, we use the *Pareto ranking* induced by non-dominated sorting [6] as the score to represent how good solutions are with respect to convergence. Thereby, the score shows the convergence improvement direction. We fit a statistical regression model to predict this score from the decision variables of the solutions contained in the current population. From the regression model, we extract variable importance and aim to perform an effective solution search by biasing recombination towards variables with larger importance. Fig. 1 illustrates the overall concept.

B. Variable Importance

As anticipated, we will extract variable importance from a statistical model that is learnt whilst the EMO search process evolves. Since the functional form of the relationship between variable values in the decision space and Pareto rankings in the objective space is *a priori* unknown, we need a flexible model that makes as few assumptions as possible. Moreover, since we want to model such a relationship online, we need our model not only to be flexible, but also computationally efficient. For these reasons, we settled on a random forest regression.

Random forest [7] is an ensemble method based on classification and regression trees [8]. Each tree is fit to a bootstrap sample of the training dataset and, additionally, the tree fitting algorithm picks for each split the most discriminative variable among a subset of m randomly-selected candidates. These two sources of randomness have the effect of reducing the correlation among individual trees, which is paid for by a small increase in their bias, in order to achieve a greater reduction in variance when aggregating trees predictions at the forest level. Indeed, bootstrapping and aggregating predictors, i.e. *bagging* [9], is especially effective with trees because of their low bias but high variance.

In this paper, we apply random forest to predict the Pareto ranking of solutions from the current population using the corresponding decision variables as inputs. With bagging, each tree has its training set drawn with replacement, which means that it will only contain a subset of solutions. These left-out observations, called *out-of bag* (OOB), can be used as a test set to estimate the accuracy of that particular tree. The OOB cases are submitted to the tree and a mean squared error (MSE) between true ranks and predicted ranks is recorded. Hence, if we want to gauge the relative importance of a variable, we can randomly permute its values in the OOB data and recalculate the OOB error: the greater the increase in MSE, the more important the considered variable. The average of such an increase in MSE over all trees in the forest constitutes the raw importance score for that particular variable [7] [10]. This measure is often called *mean decrease in accuracy* (MDA) or *permutation importance*.

TABLE I
Problems' Features.

	Separability	Modality	Geometry
WFG1	S	U	convex, mixed
WFG3	NS	U	linear
DTLZ3	S	M	concave

III. Experimental Setting

A. Benchmark Problems

We use WFG1, WFG3 [4] and DTLZ3 [5] multi-objective test problems, which are scalable in the number of objectives M and number of variables n. In WFG problems the first n_p variables determine the *position* of the solution within a front and the next n_d variables the *distance* of the solution to the optimal Pareto front. The number of position- and distance-related variables n_p and n_d can be set freely, such that $n = n_p + n_d$. In WFG1 the objective functions are separable unimodal and the optimal Pareto front has a convex and mixed geometry. On the other hand, in WFG3 the objective function are non-separable unimodal and the optimal Pareto front has a linear and degenerate geometry. DTLZ problems have $M-1$ position-related variables and $n-(M-1)$ distance-related variables. DTLZ3 objective functions are separable multi-modal and the optimal Pareto front has a concave geometry. Table I shows the features of each problem.

In WFG problems, a solution is Pareto optimal if all distance-related variables satisfy the following property: $x_i = 2i \times 0.35$, $i \in \{n_p+1, \ldots, n\}$. In DTLZ problems, a solution is Pareto optimal if all distance-related variables satisfy the following property: $x_i = 0.5$, $i \in \{n_p+1, \ldots, n\}$.

B. Algorithms

1) Baseline Algorithm (`orig`): We use the Adaptive ε-Sampling and ε-Hood (AεSεH) approach [2], [3] as a baseline EMO algorithm. AεSεH follows the main steps of a population-based elitist evolutionary algorithm, i.e. parent selection, offspring creation and survival selection. Two important features of AεSεH are the εHood method used to select parents for recombination and the ε-Sampling method used for survival selection. In this work, SBX crossover [11] is used as recombination operator, and is applied with a rate

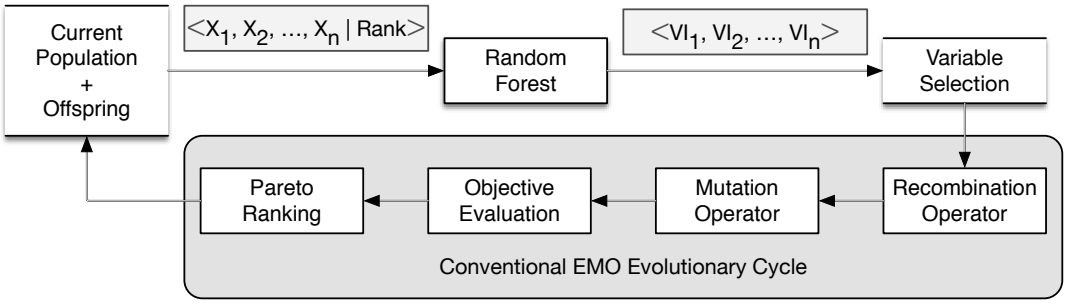

Fig. 1. Overall concept of the EMO search process with recombination based on variable importance.

P_c per individual and P_{cv} per variable. AεSεH has been shown to perform similar or better than NSGA-II for different MOPs [3], and it is known to be one of the state-of-the-art algorithms for many-objective optimization [2]. We call this baseline algorithm **`org`** for short.

2) Recombination applied to Convergence-Related Variables(`ideal`): Let us assume that convergence-related variables are known. We modify the baseline algorithm in order to take advantage of this information, and apply recombination to the variables that determine the distance to the Pareto front. This corresponds to a *cheating* algorithm having a perfect knowledge of the important variables allowing to get closer to the Pareto front; it is then expected to show the best search ability in terms of convergence. This approach, denoted **`ideal`**, will allow us to appreciate the convergence that can be achieved with a given recombination operator that perfectly learns variable importance.

Let n be the number of variables of the MOP under consideration, n_d the number of distance-related variables, n_p the number of position-related variables, and P_{cv} the probability of crossover per variable. In `ideal` there are two ways of selecting variables for recombination. When $P_{cv} \times n \leq \ n_d$, $P_{cv} \times n$ variables are selected randomly from the subset of distance-related variables. On the other hand, when $P_{cv} \times n > \ n_d$, all distance-related variables are selected for crossover and the remaining $P_{cv} \times n - n_d$ are selected at random from the subset of n_p position-related variables.

3) Recombination based on Variable Importance(`rf`): This algorithm modifies the baseline AεSεH algorithm to include the proposed method. After solutions have been evaluated in the course of the EMO search process, we obtain the estimated importance of each variable from the random forest statistical model as described in Section II-B, and we preferably select for recombination those with high importance. Selection of variables for recombination can either be determinist or probabilistic. The deterministic approach sorts the variables in the order of importance and chooses the $P_{cv} \times n$ most important ones. The probabilistic approach chooses variables based on a probability that depends on the variable importance, given by:

$$P_{cv}^{(i)} = P_{cv} \frac{VI_i}{\sum_{j=1}^{n} VI_j}$$

where $P_{cv}^{(i)}$ is the crossover probability of the i-th variable, P_{cv} is the overall crossover probability per variable, VI_i is the estimated importance of the i-th variable, and n is the total number of variables. We call this algorithm **`rf`** for short. In this work we report results for the deterministic approach.

C. Experimental Setup

We set the number of objectives to $M = 2$ in all problems, and the total number of variables to $n = 10$. We set the number of distance-related variables to $n_d = \{6, 4, 2\}$ in WFG1 in order to study the effects of recombination when the ratio between the number of distance- and position-related variables vary, while keeping the total number of variables constant. We refer to these problem instances as A, B and C, respectively. For WFG3, we set the total number of variables to $n = 10$ with $n_d = 6$ and $n_p = 4$. For DTLZ3, we set the total number of variables to $n = 10$ with $n_d = 9$ and $n_p = 1$.

The number of generations is set to 2000, and the population size is set to 100 individuals. We use SBX crossover and polynomial mutation operators with conventional parameter settings from the EMO literature. In particular, the distribution exponents are set to $\eta_c = 15$ and $\eta_m = 20$, respectively. The crossover probability per individual is set to $P_c = 1.0$ and the crossover probability for each variable is $P_{cv} = 0.5$. The mutation probability is set to $P_m = 1/n$. We report results collected from 30 independent runs.

We use an archive population that keeps the non-dominated solutions found through the generations in order to evaluate the searching ability of the algorithms. We calculate Generational Distance (GD) to evaluate convergence of the population and Inverted Generational Distance(IGD) to evaluate diversity of the population. For the calculation of GD and IGD, we have derived a reference set of 100,000 solutions for each instance in each problem.

IV. Experimental Results and Discussion

We applied the algorithms described in Section III-B to three instances of WFG1. Table II reports the average Generational Distance (GD) and Inverted Generational Distance

TABLE II

COMPARISON OF ALGORITHMS (ORG, RF, AND IDEAL) ON WFG1 INSTANCES WITH RESPECT TO GENERATIONAL DISTANCE (GD) AND INVERTED GENERATIONAL DISTANCE (IGD). THE FIRST VALUE IS THE AVERAGE INDICATOR-VALUE, THE SECOND VALUE IS THE STANDARD DEVIATION. ANY STATISTICAL DIFFERENCE BETWEEN ORG AND RF IS SHOWN IN **BOLD**.

WFG1 ($\times 10^{-1}$)

	A ($n_p = 4, n_d = 6$)		B ($n_p = 6, n_d = 4$)		C ($n_p = 8, n_d = 2$)	
	GD	IGD	GD	IGD	GD	IGD
org	$2.13_{(1.27)}$	$2.13_{(1.28)}$	$3.43_{(2.08)}$	$3.43_{(2.08)}$	$6.66_{(2.42)}$	$6.69_{(2.54)}$
rf	$\mathbf{0.56}_{(0.66)}$	$\mathbf{0.57}_{(0.65)}$	$\mathbf{2.37}_{(1.50)}$	$\mathbf{2.32}_{(1.52)}$	$\mathbf{4.01}_{(2.49)}$	$\mathbf{3.86}_{(2.68)}$
ideal	$0.64_{(0.94)}$	$0.77_{(0.91)}$	$1.13_{(1.41)}$	$1.06_{(1.43)}$	$3.15_{(3.22)}$	$2.86_{(3.49)}$

TABLE III

COMPARISON OF ALGORITHMS (ORG, RF, AND IDEAL) ON WFG3 AND DTLZ3 WITH RESPECT TO GENERATIONAL DISTANCE (GD) AND INVERTED GENERATIONAL DISTANCE (IGD). THE FIRST VALUE IS THE AVERAGE INDICATOR-VALUE, THE SECOND VALUE IS THE STANDARD DEVIATION. ANY STATISTICAL DIFFERENCE BETWEEN ORG AND RF IS SHOWN IN **BOLD**.

	WFG3 ($\times 10^{-3}$) ($n_p = 4, n_d = 6$)		**DTLZ3** ($\times 10^{-4}$) ($n_p = 1, n_d = 9$)	
	GD	IGD	GD	IGD
org	$7.00_{(0.89)}$	$0.94_{(0.52)}$	$2.46_{(1.68)}$	$2.78_{(1.55)}$
rf	$\mathbf{5.17}_{(0.60)}$	$\mathbf{0.47}_{(0.35)}$	$\mathbf{0.86}_{(0.65)}$	$2.28_{(0.68)}$
ideal	$4.25_{(0.36)}$	$0.39_{(0.37)}$	$0.83_{(0.81)}$	$2.57_{(0.65)}$

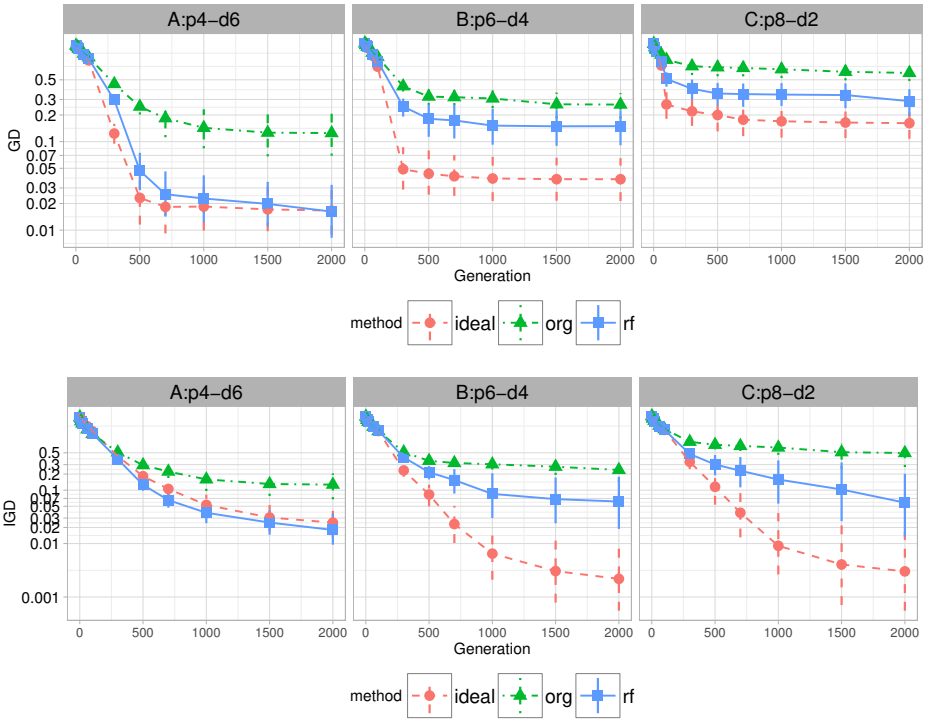

Fig. 2. GD and IGD in WFG1

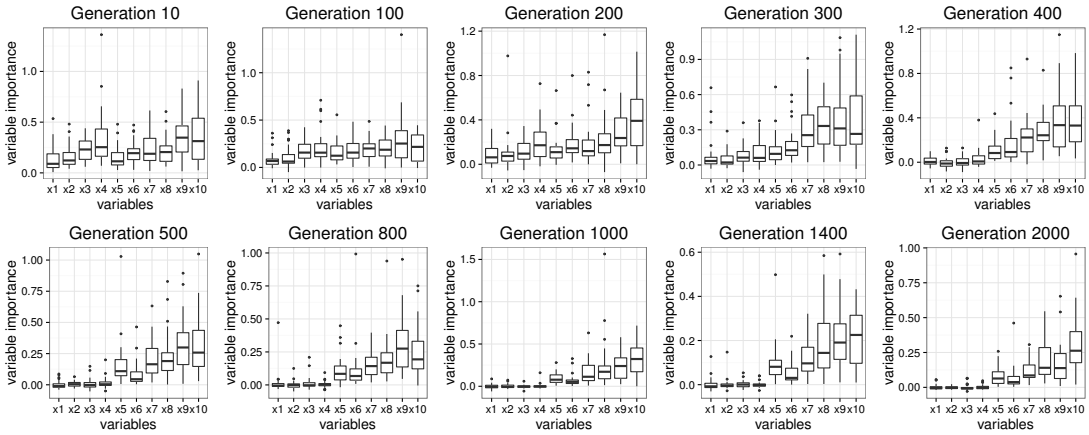

Fig. 3. VI in WFG1 instance A

(IGD) values obtained by all algorithms for all instances of WFG1, computed at the last generation of the 30 runs. Standard deviations are shown in parenthesis. Results in bold show if there is any significant difference between `org` and `rf`, based on a Mann-Whitney non-parametric statistical test with a p-value of 0.05. From this table, it can be seen that `rf` is never outperformed by `org` on any case. More importantly, it is able to provide statistically better GD- and IGD-values than `org` in all instances. It is also interesting to notice that `ideal` achieves the best indicator-values in most of the instances, then validating the idea of designing variable importance-aware recombination mechanisms to improve the search ability.

Fig. 2 shows the GD (on top) and the IGD (at the bottom) values obtained by the algorithms over the generations on all WFG1 instances, that result from the combination of number of distance- and position-related variables described above. The algorithms are shown in red, green and blue lines and are labeled `org`, `rf`, and `ideal`, respectively. Looking at Fig. 2, it can be seen that `ideal` achieves the best GD-values through generations of all instances except for the values at 2000 generation of instance A. This is expected since `ideal` represents an algorithm with a perfect model for variables related to convergence. We can see that the method `rf`, which learns online what variables are important to improve convergence and emphasizes their recombination, obtains significantly better GD-values than the baseline algorithm `org`. The GD-values of method `rf` decrease substantially during the initial stage of evolution and it can be said that the proposed method has a large impact for population convergence especially in this initial stage. This helps to increase convergence speed of population during solution search. It also can be seen that `rf` has better IGD-values than `org` in all instances. The IGD-values of method `rf` decrease more gradually than the GD-values. This is because position-related variables are not given much chance to undergo crossover in `rf`. Decreasing IGD-values is caused by both convergence and diversity of population. In instance A, we can see that `ideal` has worse IGD-values than `rf`. This is because instance A has 6 distance-related variables and `ideal` always applies crossover to $P_{cv} \times n = 0.5 \times 10 = 5$ of those variables. So, `ideal` never applies crossover to position-related variables.

Fig. 3 shows the boxplot of variable importance for each variable by `rf` at given generation steps. The number of generations is showed on the top of each boxplot graph. It seems that there is no difference at the beginning of the search, but after 400 generations, distance-related variables $x_5 \sim x_{10}$ have larger variable importance than position-related variables $x_1 \sim x_4$. That is, the regression model in random forest is able to correctly distinguish between distance- and position-related variables. In method `rf`, the variables that have larger variable importance get more chance to recombine. So we can find solutions which have better convergence by giving high crossover chance to distance-related variables.

In addition to WFG1, we applied the same algorithms to two instances of WFG3 and DTLZ3 that have different function features from WFG1. Table III reports the average GD- and IGD-values obtained by all algorithms of WFG3 and DTLZ3, computed at the last generation of the 30 runs. We can see significantly better GD- and IGD-values of `rf` than those of `org` except for IGD in DTLZ3. Fig. 4 and Fig. 5 show the GD- (left) and the IGD- (right) values obtained by the algorithms over the generations on WFG3 and DTLZ3. Fig. 6 and Fig. 7 show the boxplot of variable importance for each variable by `rf` at some points of generations on WFG3 and DTLZ3. Note that we have 4 position-related variables and 6 distance-related variables in WFG3, and 1 position-related variable and 9 distance-related variables in DTLZ3. Looking at Fig. 4 and Fig. 6, the method `rf` has better GD- and IGD-values than `org` through all generations, and there is a difference in between position- and distance-related variables from the beginning of the generation unlike WFG1. From these results, it appears that the proposed method is able to identify the distance-related variables. In this case, `ideal` has better IGD-values than `rf`. The estimate of distance-related variables on WFG3 is better than that on WFG1 so the distance-related variables get higher chance to undergo crossover on WFG3

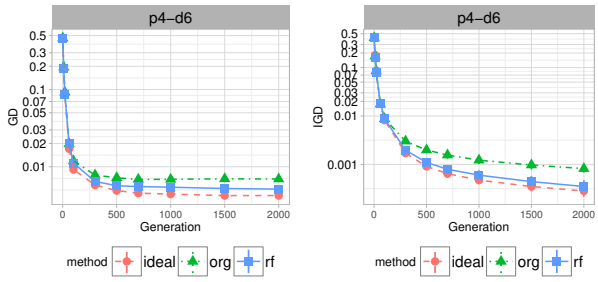

Fig. 4. GD and IGD in WFG3.

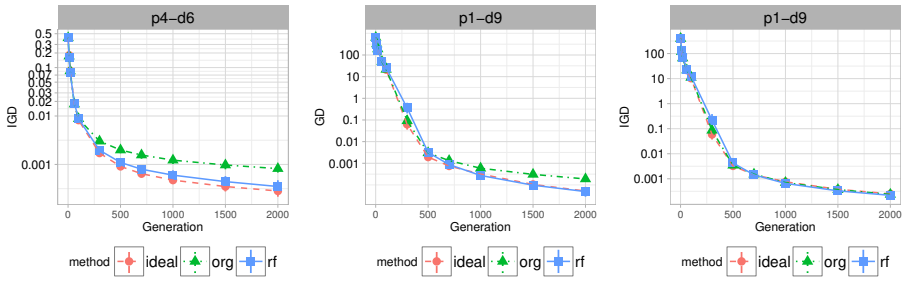

Fig. 5. GD and IGD in DTLZ3.

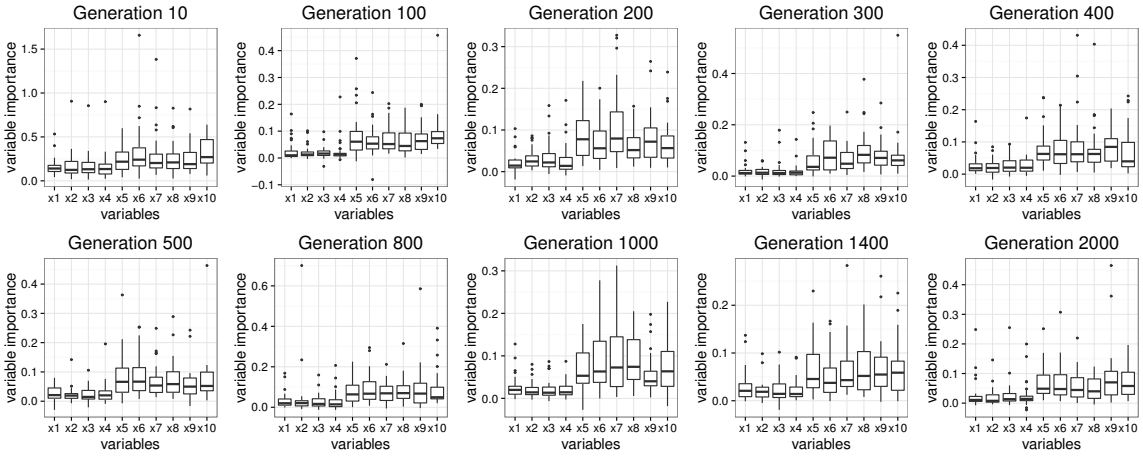

Fig. 6. VI in WFG3

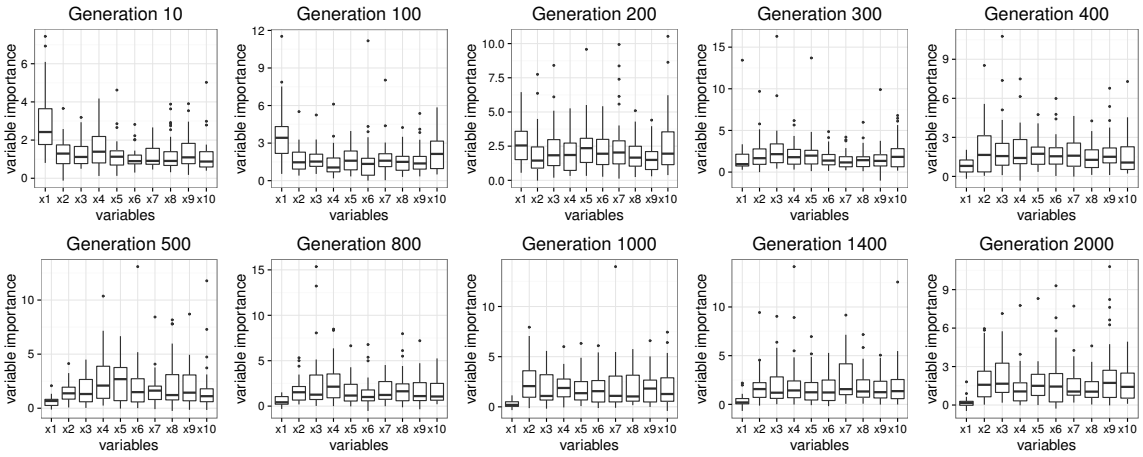

Fig. 7. VI in DTLZ3

than on WFG1.

Looking at Fig. 5 and Fig. 7, it can be seen that there is not much difference in IGD and there are small differences in GD. The method `rf` does not have better GD-values at generation 300 but it improves over `org` after 500 generations. In the beginning, the variable importances of distance-related variables are lower than those of position-related variables. So crossover is applied to position-related variables frequently. This might be why `rf` obtains worse GD-values before 500 generation.

Looking at Fig. 4 and Fig. 5, we cannot see large differences in GD and IGD metric values between algorithms as with WFG1. This might be due to a limit in the variation operators ability to keep generating improving solutions, and it depends on the features of test problems. We attribute this limit to the solution search ability of the SBX crossover operator, which we use in all the algorithms. If SBX was more effective on the WFG3 and DTLZ3 problems, we would expect to see a larger advantage in terms of GD-values for the `ideal` method, and consequently a larger advantage of the `rf` method over the `org` method. Even so, the proposed `rf` approach requires half the number of generations to obtain GD-values comparable to those of the `org` method.

V. Conclusions

In this work, we investigated how to learn, online, which variables favor Pareto improvements, and how to guide variation operators accordingly. This machine learning-based approach uses random forest to perform a regression of the Pareto ranking over decision variables in order to estimate variable importance at each iteration. We compared the convergence ability of a baseline algorithm (AεSεH), a version enhanced with the proposed method, as well as an ideal version with a perfect knowledge of the variables that are important for convergence on WFG1, WFG3 and DTLZ3 test problems. We showed that the machine learning-enhanced algorithm achieves a significantly better convergence using GD and IGD metric. An overall clear statistical difference in performance in favor of the proposed method can be seen. We verified that the regression model is able to distinguish correctly between distance- and position-related variables throughout the generations based on the estimated variable importance.

However, we understood that there is a limit to the searching ability of SBX operator in terms of convergence. We cannot exceed the searching ability of the ideal algorithm using SBX in this paper even though we attempt to learn important variables for convergence.

In the future, we plan to explore other machine learning alternatives to compute variable importance. It will also be interesting to extend the proposed method in order to guide mutation in addition to recombination, and then study the effect of variable importance-aware variation on problems with three and more objectives. Moreover we need to apply the proposed method to other variation operators because we notice that SBX has limits in its ability to get close to the true POS.

References

[1] R. C. Xingyi Zhang, Ye Tian and Y. Jin, "A decision variable clustering-based evolutionary algorithm for large-scale many-objective optimization," *IEEE Transactions on Evolutionary Computation*, 2016.

[2] H. Aguirre, A. Oyama, and K. Tanaka, "Adaptive ε-sampling and ε-hood for evolutionary many-objective optimization," in *Evolutionary Multi-Criterion Optimization*, ser. Lecture Notes in Computer Science, vol. 7811, 2013, pp. 322–336.

[3] H. Aguirre, Y. Yazawa, A. Oyama, and K. Tanaka, "Extending AεEεH from many-objective to multi-objective optimization," in *Conference on Simulated Evolution and Learning*, ser. Lecture Notes in Computer Science, vol. 8886, 2014, pp. 239–250.

[4] S. Huband, P. Hingston, L. Barone, and R. While, "A review of multi-objective test problems and a scalable test problem toolkit," *IEEE Transactions on Evolutionary Computation*, vol. 10, no. 5, pp. 477–506, 2007.

[5] K. Deb, L. Thiele, M. Laumanns, and E. Zitzler, "Scalable test problems for evolutionary multi-objective optimization," pp. 105–145, 2005.

[6] D. E. Goldberg, *Genetic Algorithms in Search, Optimization and Machine Learning*. Addison-Wesley, 1989.

[7] L. Breiman, "Random forests," *Machine learning*, vol. 45, no. 1, pp. 5–32, 2001.

[8] L. Breiman, J. Friedman, C. J. Stone, and R. A. Olshen, *Classification and regression trees*. CRC press, 1984.

[9] L. Breiman, "Bagging predictors," *Machine learning*, vol. 24, no. 2, pp. 123–140, 1996.

[10] ——, "Manual on setting up, using, and understanding random forests v3. 1," *Statistics Department University of California Berkeley, CA, USA*, 2002.

[11] K. Deb and R. B. Agrawal, "Simulated binary crossover for continuous search space," *Complex Systems*, vol. 9, pp. 115–148, 1995.