Embedding a Repair Operator in Evolutionary Single and Multi-Objective Algorithms - An Exploitation-Exploration Perspective*

Kalyanmoy Deb[1], Sukrit Mittal[2], Dhish Kumar Saxena[2], and Erik D. Goodman[1]

[1] BEACON Center, Michigan State University, East Lansing, MI USA
{kdeb,goodman}@egr.msu.edu
[2] Indian Institute of Technology Roorkee, India
{smittal1,dhish.saxena}@me.iitr.ac.in

COIN Report Number 2020023

Abstract. Evolutionary algorithms (EAs) are population-based search and optimization methods whose efficacy strongly depends on a fine balance between *exploitation* caused mainly by its selection operators and *exploration* introduced mainly by its variation (crossover and mutation) operators. An attempt to improve an EA's performance by simply adding a new and apparently promising operator may turn out to be counter-productive, as it may trigger an imbalance in the exploitation-exploration trade-off. This necessitates a proper understanding of mechanisms to restore the balance while accommodating a new operator. In this paper, we introduce a new *repair* operator based on an AI-based mapping between past and current good population members to improve an EA's convergence properties. This operator is applied to problems with different characteristics, including single-objective (with single and multiple global optima) and multi-objective problems. The focus in this paper is to highlight the importance of restoring the exploitation-exploration balance when a new operator is introduced. We show how different combinations of problems and EA characteristics pose different opportunities for restoration of this balance, enabling the *repair*-based EAs/EMOs to outperform the original EAs/EMOs in most cases.

Keywords: Evolutionary algorithms · Exploration · Exploitation · Convergence · Repair operator · *Innovization.*

1 Introduction

Evolutionary algorithms (EAs) are population-based search and optimization algorithms that start with a set of random solutions to achieve a single best solution or multiple equally good (non-dominated) solutions. Different EAs use various operators, like selection, crossover, mutation, etc., to create offspring

* This research work has been supported by Government of India under SPARC project P66.

solutions in each generation. While deploying these operators, two key characteristics should be considered, namely, *exploration* and *exploitation*. Exploitation refers to the extent of emphasis provided to the current population-best (or above-average population members) in mating- or survival-selection. In contrast, exploration refers to the extent of search introduced by the variation operators, such as crossover and mutation operators, in creating an offspring population. Any working EA maintains a balance between these two by using a combination of evolutionary operators; however, failure to do so may cause poor or premature convergence. This issue is discussed further in Section 2.

Even though learning during the search process is not new [1], recent studies [7, 8, 13, 14] have focused on *online innovization*, which attempts to extract variable patterns from certain good solutions and utilize those through an additional *repair* operator (apart from crossover and mutation) in subsequent generations. This operator has been discussed in Section 3. Here, the repair operator's disruption to the otherwise existing exploitation-exploration balance and the necessary changes to restore this balance have also been discussed. Subsequently, this repair operator is applied in single-objective, multi-modal, and multi-objective problems. Sections 4, 5 and 6 demonstrate the application and efficacy of this operator from the exploitation-exploration perspective and discuss some ways to restore the balance that existed before introducing this new *repair* operator. Finally, Section 7 presents the conclusions of this study.

2 Exploitation-Exploration Perspective

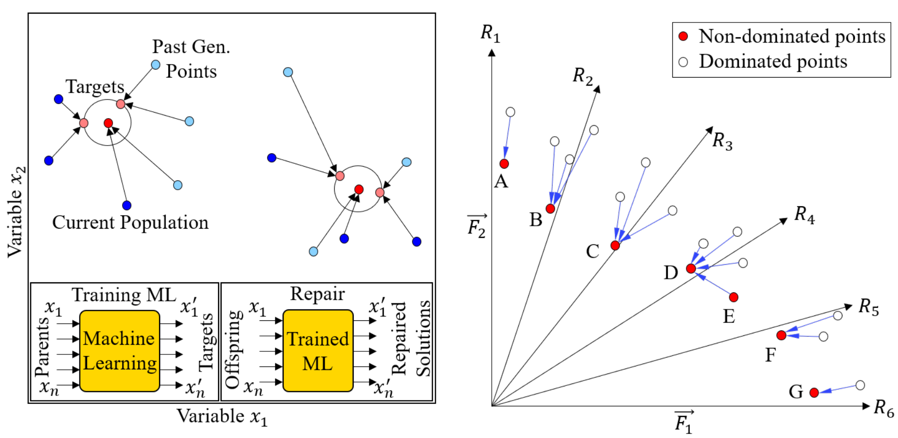
In a canonical EA, the exploitation is established in a population by the mating selection and the survival selection operators, and exploration refers to the extent of *search* done by introducing variation using crossover and mutation operators. Goldberg and Deb [9] showed that if the exploitation-exploration balance is not established, the resulting Genetic Algorithm (GA) fails to solve even a simple linear and unconstrained "onemax" problem.

While it is crucial to make a suitable balance between exploitation and exploration within an EA's population by selecting and parameterizing its operators, their quantification is not very clear. Goldberg and Deb [9] used the selection pressure (number of copies of the population-best solution in mating pool) as an indicator of exploitation and the crossover probability (p_c) for exploration in an EA (mutation was not used). The higher the value of these parameters, the greater is the extent of the corresponding issue. Future research can devise more comprehensive indicators for exploitation and exploration in a population, considering every operation implemented in an EA. But here, we discuss how the balance is affected by introducing a new (and notably, promising) operator.

The majority of EA applications use tournament selection for creating a mating pool and real-parameter crossover and mutation operators for creating an offspring population. The best half of a combined parent and offspring population is often used as the survival selection operator. In some applications, the top κ% of parent population members are included in the next population to preserve elites. EA application then requires fine-tuning of the associated hyper-parameters (such as tournament size, crossover and mutation probabilities, and κ) to develop an efficient EA for a specific problem. When an EA with standard operators is used with their optimized hyper-parameters, an important question arises: if a novel operator is to be introduced—how shall the imbalance between exploitation and exploration caused by the new operator be restored to make the overall algorithm efficient again? We address these issues in the context of introducing a new repair operator, first for single-objective optimization to search for single and multiple solutions, and then for multi-objective optimization.

3 An 'Innovized' Repair Operator

Since the beginning, EAs have been treated as Markov chains, in which the new population at generation $t+1$ is created entirely from the population at generation t, with some exceptions such as particle swarm optimization (PSO) [12]. But a recent focus in the optimization literature, particularly in population-based methods, has been on *learning* a mapping of current solutions with relevant solutions from the past generations and then utilizing the learned mapping to *repair* newly created offspring in the hope of improving them further. Recent advances in machine learning (ML) methods allow us to consider such a repair operator as a new additional operator for creating better offspring solutions. Learning from past generations for knowledge acquisition is commonly called *innovization* in the context of evolutionary multi-objective optimization (EMO) [5].



(a) $M = 1$ (b) $M \geq 2$

Fig. 1: The mapping process for ML training in the *Innovized Repair* Operator.

Figure 1a illustrates the principle of the *innovized* repair (IR) operator in the context of a generic multi-modal optimization problem, seeking to find multiple optimal solutions. The red points are population-best solutions with high fitness. They and their neighboring high-fitness solutions are target points. Past and current population members are mapped to one of the target points based on distance in variable space. In the training phase, an ML method is used to learn the mapping between a past or current population member with a single target solution in the variable space having a high fitness value. Once the mapping is learned, an offspring can be repaired using the learned (or trained) ML model to create a new and hopefully better solution.

Offspring created using crossover and mutation operators applied on the current parent population are expected to possess good properties of parents, but the success of this process depends on the crossover and mutation operators used. However, if the mapping of past populations to the current-best target points is captured, a not-so-good offspring can be repaired before it is evaluated in the hope of making the solution better. This is the basic motivation for the proposed IR operator. A comparison with the other learning-based methods like *surrogate-assisted* EAs, and crossovers like that of *Differential Evolution*, has been drawn in [14], and an extensive performance analysis of the IR operator is presented in [15]. The scope of this paper is limited to discussing the disruption of exploration-exploitation balance in the context of the IR operator and restoring that balance.

4 Search for a Single Optimal Solution

With inclusion of the IR operator for a single-objective optimization task (number of objectives, $M = 1$), the resulting EA has an additional exploration operator, as it is involved in creating a new solution. However, instead of extending the exploration to unexplored regions of the search space, it attempts to bring offspring close to already found good target solutions. Thus, in some sense, although it is used as an additional exploration operator, it helps to exploit the current-best regions of the search space, thereby increasing the extent of exploitation of the original algorithm. This creates an imbalance in the previously-balanced exploitation-exploration battle. To make the resulting EA efficient, either the exploitation must be reduced or the exploration must be increased.

Algorithm 1 Generation t of EA with IR operator

Require: Parent population P_t, Archive A_t.
1: $A_t \leftarrow A_T \bigcup P_t \backslash P_{t-t_{past}}$ `% Archive Update`
2: $P_{mating} \leftarrow$ Tournament Selection on P_t
3: $Q_t \leftarrow$ Crossover and mutation on P_{mating}
4: $T \leftarrow$ Variable vector(s) of target solution(s) identified from P_t
5: $X \leftarrow$ All variable vectors in A_t
6: $D \leftarrow$ Map the solutions in X to T
7: $D \leftarrow$ Dynamic normalization of D `%using equation 1`
8: $Model \leftarrow$ Train the ANN using D
9: $Q_t \leftarrow$ Repair randomly selected 50% offsprings Q_t using $Model$
10: Evaluate Q_t
11: $P_{t+1} \leftarrow$ Survival Selection on $P_t \bigcup Q_t$
12: **return** Next Parent Population P_{t+1}

Algorithm 1 shows the implementation of a learning-based *innovized* repair operator for a single- or multi-objective EA. The operator is executed in two steps, i.e., the learning step and the repair step. The training dataset D is created by mapping all solutions in the archive A_t (generated from t_{past} previous generations) to the target solution(s) T. D is then normalized and used to train an ANN, which serves as the prediction model for the repair step. In an n-variable problem, the k^{th} variable is normalized using the bounds given in Equation 1, where $[x_k^l, x_k^u]$ are the variable bounds specified in the problem definition and $[x_k^{l,t}, x_k^{u,t}]$ are the minimum and maximum values of the k^{th} variable in the archive A_t [13]:

$$x_k^{\min} = 0.5(x_k^{l,t} + x_k^l), \quad x_k^{\max} = 0.5(x_k^{u,t} + x_k^u). \tag{1}$$

The ANN used here has a pre-determined architecture with two hidden layers of 30 neurons each. The input and output layers have neurons equal to the number of decision variables n in the optimization problem. The loss function used is *Mean Squared Error* (MSE), activation function is *sigmoid* and the optimizer is *adam*[3]. The training is terminated if the MSE is stagnant for 50 epochs. The maximum epoch limit is set to 2,500. For the repair step, each offspring is first normalized using the already acquired bounds from Equation 1, repaired using the trained ANN and then denormalized to the original scale. In this process, it is possible that some variable may go out of the bounds $[x_k^l, x_k^u]$ as defined in the problem. To address this, another repair is done using an Inverse Parabolic Spread Distribution [16], individually, for the variables lying out of bounds.

There are different ways in which one ($N_T = 1$) or multiple ($N_T > 1$) target solutions can be selected. We start with the obvious choice of one target solution—i.e., the best fitness solution as discussed in the Subsection 4.1. Learning to map all existing solutions to a single target solution would emphasize the exploitation aspect of this IR operator. In order to reduce that exploitation, multiple diverse target solutions can be selected from P_t. Even though some targets may have worse fitness values, selecting a target that is dissimilar to the best solution can help. The selection of multiple target solutions and how it practically differs from using IR with a single target is discussed in Subsection 4.2.

4.1 Repairing a Single-Target Solution

Here, all solutions in A_t are mapped to the current best-fitness solution T. For this study, four 20-variable single-objective test problems are chosen: Ackley, Different Powers Function (DPF) [6], Rastrigin, and Zakharov.

A comparison is done between an EA with and one without the IR operator. The EA with repair is referred to as EA-IR. The EA (here, GA) is used with population size $N = 100$, binary tournament selection, SBX crossover ($\eta_c = 3$, $p_c = 0.9$), polynomial mutation ($\eta_m = 20$, $p_m = 1/n$), a survival selection operator where the best half of the $P_t \bigcup Q_t$ are selected and $t_{past} = 5$ [13]. For each test instance, 31 independently seeded runs are done and the median of the best fitness F-value at each generation is shown in Figure 2. The p-value is calculated using the Wilcoxon Ranksum Test. It is evident from the figure

[3] The parameters are taken from https://keras.io/api/optimizers/adam/

that EA-IR applied in this way performs worse in all cases than the EA without repair. The IR operator, focusing on a single solution, increases the exploitation factor, thereby disrupting the original exploration-exploitation balance of the EA.

Table 1 shows various EA operators and their effects on the exploration or exploration aspects. Thus, when a new operator to be implemented within an EA is expected to disrupt the balance between exploration and exploitation, one of the operators can be modified as depicted in the table to try to restore the balance.

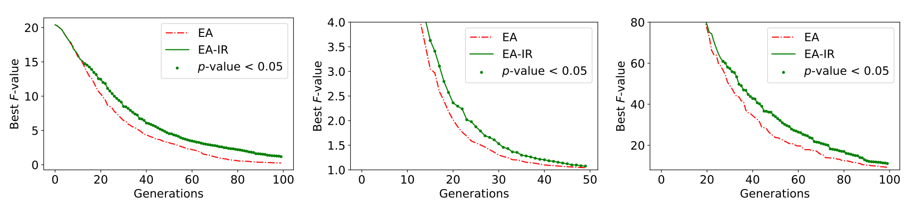
Table 1: Effect of operators and their parameters on exploration-exploitation aspects.

Operator	Modification	Effect
Initialization	Biased $\uparrow$	Exploration $\downarrow$
Crossover	$\eta_c \downarrow$ or $p_c \uparrow$	Exploration $\uparrow$
Mutation	$\eta_m \downarrow$ or $p_m \uparrow$	Exploration $\uparrow$
IR	$t_{freq} \uparrow$ or $N_T \uparrow$	Exploitation $\downarrow$
Mating Selection	Tournament size $\uparrow$	Exploitation $\uparrow$
Survival Selection	Niche-preservation	Exploitation $\downarrow$
Elites	% Population preserved $\uparrow$	Exploitation $\uparrow$
Child Acceptance	Steady-state	Exploitation $\uparrow$

4.2 Repairing to Multiple Target Solutions

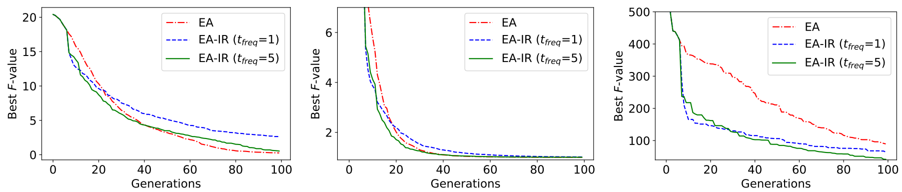
The increase in exploitation (discussed in Subsection 4.1) can be reduced if a distributed set of N_T neighboring solutions to the current best-fitness solution are used as targets. This requires the following two considerations:

1. A clearing strategy is used to choose N_T good and diverse solutions. First, the population is sorted from best to worst according to fitness. Then, the first solution on the list is chosen as the first target solution. Thereafter, all members within ϵ Euclidean distance in the variable space from the chosen solution are cleared and the next un-cleared solution is chosen as the second target solution. Then, all subsequent solutions within ϵ Euclidean distance from the two chosen target solutions are cleared and the next solution in the list is accepted as the next target solution. This is continued until N_T target solutions are picked. Thus, the target solutions are guaranteed to have a minimum Euclidean distance of ϵ from each other.
2. For the training phase, every past solution is associated with the nearest target solution in the variable space. Since only N_T targets are chosen from P_t, the other $(N - N_T)$ members remain part of the archive A_t. To avoid mapping of an archive solution to a target having a worse fitness value, the



(a) Ackley (b) DPF (c) Rastrigin

Fig. 2: Median fitness plots with EA using IR with a single target.



(a) Ackley (b) DPF (c) Zakharov

Fig. 3: Median fitness plots with EA using IR with multiple diverse targets.

algorithm ensures that the target must have a better fitness than the associated archive solution. Otherwise, the pairing is excluded from the training dataset D.

Keeping all other parameters similar to the EA-IR used in Subsection 4.1, the experiments are repeated using $N_T = 5$. The parameter ϵ is adaptively fixed at every generation by evaluating the average Euclidean distance of the k^{th} (=5 used here) nearest neighbors of members of P_t. Besides selecting multiple targets for learning, another way to reduce the exploitation of the IR operator is to use it after every t_{freq} (> 1) generations rather than $t_{freq} = 1$ (see Table 1). In this subsection, we compare $t_{freq} = 1$ and 5. The median best-fitness value is plotted in Figure 3, suggesting that $t_{freq} = 5$ performs better than $t_{freq} = 1$.

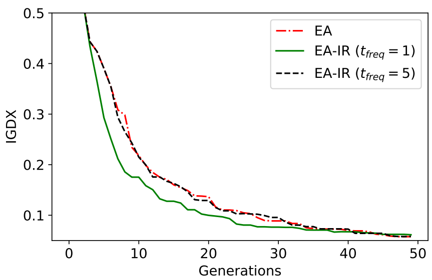
There is a significant improvement in fitness values on the Zakharov problem, while for the remaining problems, the fitness improves in the beginning, but degrades slightly in the later generations. This can be explained as follows. Ensuring a diverse set of target solutions (by using ϵ) improves the search in initial generations by increasing the exploration to make a balance with the increased exploitation caused in the IR operator. However, the ϵ gets smaller with generation number, as population members approach the optimal solution. With a reduced diversity in target solutions, the exploration is reduced, thereby disrupting the exploitation-exploration balance. However, the exploitation effect of the IR operator gets reduced when t_{freq} is increased to 5, restoring the balance, and the performance of EA-IR gets better.

As discussed in Table 1, a niche-preserving operation can also reduce the effect of increased exploitation due to the IR operator. This motivates us to introduce the IR operator in a multi-modal EA to find multiple optimal solutions with increased efficiency. The implementation and results are discussed in the next section.

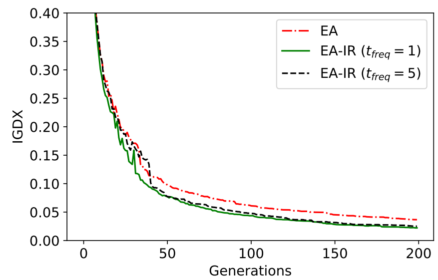
5 Search for Multiple Global Optima

For finding multiple optimal solutions simultaneously in a single run, EAs with niche-preserving methods were proposed [17, 10]. We introduce the IR operator for finding better offspring solutions, but also simultaneously using a diverse set of target solutions to find multiple optima, as shown in Figure 1a.

Here, clearing [17] is used as the niche-preservation method for survival selection in EA. Since it already has a parameter ϵ that defines a niching radius in the variable space, the same ϵ is used to identify the target solutions for ANN



(a) Himmelblau Problem ($n = 2$)



(b) MMP Problem ($n = 15$)

Fig. 4: Median IGDX comparison for multi-modal problems with clearing-baed multi-modal EA (with and without the IR operator).

training dataset D. The rest of the IR operator is the same as used in Subsection 4.1 for selecting multiple target solutions. For the experiments, we use the Himmelblau function with $n = 2$ and four optimal solutions and the MMP problem [4] with $n = 15$ having eight optimal solutions. The EA is used with binary tournament selection, SBX crossover ($\eta_c = 3$, $p_c = 0.9$), and polynomial mutation ($\eta_m = 10$, $p_m = 1/n$). The niching parameter $\epsilon = 0.1$ and 0.2, and population size $N = 40$ and 100 are used for the Himmelblau and MMP problems, respectively. Two values of target solutions N_T (5 and 10) are tried for the IR operator. Since there are multiple optimal solutions, the best fitness value is not an appropriate measure. Hence, IGDX (IGD computed in the variable space with known $\mathbf{x}^*$ set) is used as the performance indicator, as shown in Figure 4.

Since in this problem, multiple optima in the variable space are naturally expected to be distant from each other, and a training of the ANN is likely to learn mapping of solutions to multiple distant target solutions simultaneously, an increase in exploration caused by artificially spreading solutions near diverse targets gets somewhat balanced by the increase in exploitation caused by convergence of solutions to each target by the IR operator. The figure shows that using $t_{freq} = 1$ (applying IR at every generation) performs better than either the original niching EA, or the IR-based EA with $t_{freq} = 5$. Pushing offspring solutions near multiple target solution makes a good balance of exploitation and exploration, and applying it at every generation provides maximum advantage. If the frequency of applying the IR operator is reduced (i.e., t_{freq} is increased to 5), the fullest advantage is lost and the performance of the IR-based EA drops.

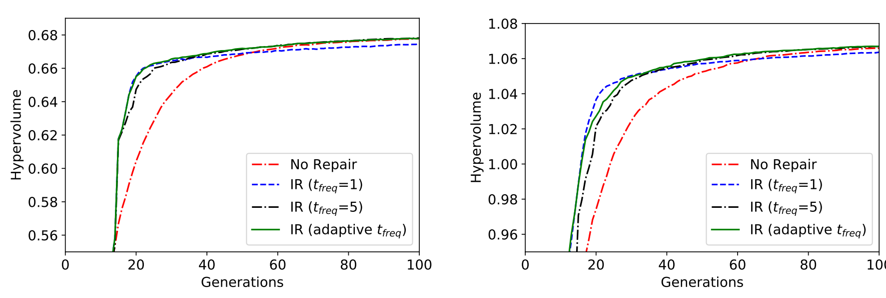
Following the above, it is clear that if implemented well from an exploitation-exploration perspective, the IR operator should demonstrate a performance enhancement in multi-objective optimization as well, where the goal is to find a well-distributed set of Pareto-optimal (PO) solutions. This is presented next.

6 Search for Multi-Objective Solutions

With proper implementation, the increased focus on multiple non-dominated but distant variable space solutions may prove to be beneficial in an evolutionary multi-objective optimization (EMO) procedure. A preliminary study [13] motivated us to launch this extensive study. In EMO, all non-dominated solutions at generation t are equally good and, hence, can serve as the target solutions. Their count N_T may range from 1 to N. The archive solutions are mapped to the non-dominated solutions in P_t that act as the target solutions T. The main question that now remains is how an archive solution can be mapped to a respective target solution, particularly in the context of multiple objectives.

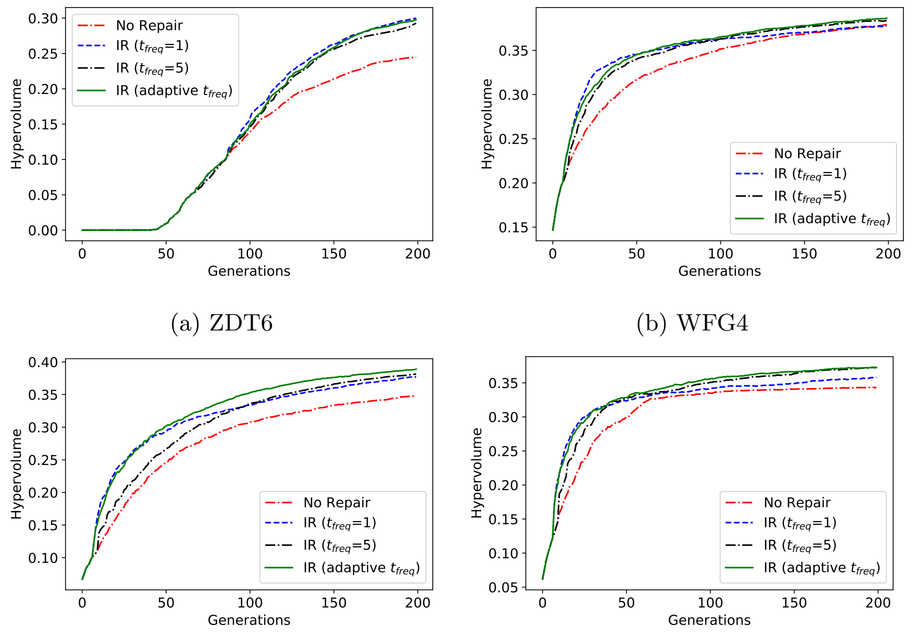
Figure 1b shows the schematic of associating archive solutions in A_t to the target solution set T. The idea is motivated from the goal of achieving better solutions, improving orthogonally to the PO front. As in Figure 1b, N equidistant reference vectors are generated (R1–R6) and the non-dominated solutions (A–G) are assigned to the reference vectors using the Achievement Scalarizing Function (ASF) [18]. As evident, even though E is a non-dominated solution, it is not selected as a target since choosing D for R4 could provide a more diverse target set. The remaining solutions in A_t are then attached to their nearest reference vectors using ASF, and consequently are associated to the target solutions assigned to those particular reference vectors.

This modified IR operator for the multi-objective framework is used with NSGA-II and NSGA-III, and its performance is evaluated on a number of test functions from ZDT [19] and WFG [11] test suites. The ZDT problems are modified to shift the optima to $x_k = 0.5$ for $k = 2, \ldots, 30$, in order to create non-boundary optimal solutions. NSGA-II uses a population size $N = 100$, SBX crossover ($\eta_c = 10$, $p_c = 0.9$) and polynomial mutation ($\eta_m = 20$, $p_m = 1/n$). N reference-vectors are generated using the Das and Dennis method [2], for mapping in the training dataset. To make the learning from some non-dominated solutions effective, we wait to apply the IR operator until at least 50% of P_t is non-dominated. For NSGA-III, the perpendicular distance matrix (PDM) [3] is used for reference-vector mapping instead of ASF (as was used for NSGA-II). This is to ensure that the targets of the survival selection and the IR operator are aligned. We use $N = 105$ for three-objective WFG problems.



(a) ZDT1. (b) ZDT3.

Fig. 5: Median HV plots for two-objective ZDT problems using NSGA-II (with and without the IR operator).



(a) ZDT6 (b) WFG4

(c) WFG6 (d) WFG9

Fig. 6: Median HV plots for ZDT6 ($M = 2$) and WFG ($M = 3$) problems. Three-objective problems are solved using NSGA-III (with and without the IR operator).

The results on two-objective ZDT problems and three-objective WFG problems are shown in Figures 5 and 6, respectively. We explore the effect with $t_{freq} = 1$ and 5. The plots represent the generation-wise median hypervolumes (HV) from 31 independent runs. With the exception of ZDT6, all other plots reveal that IR with $t_{freq} = 1$ (more exploitation) performs better in earlier generations, while $t_{freq} = 5$ (less exploitation) works well in later generations. This is consistent with our previous argument in Subsection 4.2 about the aggressive use the IR operator in the beginning when a natural diversity exists in the population. This suggests that adjusting the exploitation of the IR operator with generations by self-adapting t_{freq} parameter can achieve a better overall performance (by maintaining the exploration-exploitation balance from beginning to end).

Towards this, t_{freq} is initialized at two. Then, the count of total survived offspring (SO) is recorded at every generation. At any generation t when the offspring are repaired using the IR operator, the count of total surviving offspring SO_t and the count of offspring that survived in the previous generation SO_{t-1} are compared. If $SO_t \geq SO_{t-1}$, then t_{freq} is reduced by one, else t_{freq} is increased by one.

Figures 5 and 6 show the results for the IR operator with adaptive t_{freq}, which works better than fixing $t_{freq} = 1$ or 5 for all generations. It performs better by having higher exploitation at earlier generations and reducing it in later generations, but importantly, by striking a balance with the exploration effect caused by diversity preservation among ND solutions and by crossover and mutation operators. This study also shows the importance of balancing the exploitation-exploration issue adaptively, if possible, within an algorithm.

7 Conclusions

In this paper, a learning-based innovized repair (IR) operator for single-objective (including multi-modal) and multi-objective optimization problems has been introduced, and analyzed from the exploitation-exploration perspective. The repair operator learns the mapping of past not-so-good (dominated) solutions to relevant current good (non-dominated) solutions and uses the learned mapping to repair created offspring solutions in the hope of pushing them close to the better solutions. While this seems to be a promising operator for any EA or EMO algorithm, its standalone addition to a standard single-objective EA has not shown to improve the EA's performance. It has been argued that the introduced repair operator puts an over-emphasis on the exploitation aspect and disrupts its balance with exploration. Other adjustments (mainly to reduce the extent of exploitation, in this case) were needed to restore the balance so that the resulting modified algorithm starts to work better than the original algorithm.

Next, we have executed insightful moderations of the extent of exploitation (through mechanisms such as the use of multiple targets, niching, adaptive frequency of repair, etc.), in an attempt to restore the balance in multi-modal problems. The modified EA has been shown to produce performance comparable to or better than its original version, highlighting that the crux of a good population-based optimization algorithm design is intricately related to maintaining a balance between exploitation and exploration aspects of its operators.

In multi-objective problems, EMO algorithms distribute the effect of exploitation to a number of well-distributed multiple optimal solutions in the search space. While the increase in exploitation by the introduction of the innovized repair operator is not as severe as in unimodal problems, its aggressive use in the beginning and less-frequent use at the end of a run has been found to better maintain the exploitation-exploration balance. This concept has us led to develop a self-adaptive EMO procedure in which the frequency of use of the IR operator is adapted based on the success of repaired offspring solutions, resulting in better performance than the original EMO algorithm.

While the modified EA and EMO algorithms with the IR operator embedded have been found to perform better than their original versions, the main message of this paper remains the demonstration of the importance of maintaining a balance of two important factors in a population-based optimization algorithm: exploitation caused by convergence-enhancing operators and exploration caused by diversity-enhancing operators. By keeping the balance in mind, EA and EMO researchers will have a better understanding of the working of the algorithms, will be better able to modify and introduce new operators for performance enhancement, and importantly, will be able to apply them with confidence.

References

1. Battiti, R., Brunato, M.: Reactive search optimization: Learning while optimizing. In: Gendreau, M., Potvin, J.Y. (eds.) Handbook of Metaheuristics, pp. 543–571. Springer US, Boston, MA (2010)
2. Das, I., Dennis, J.: Normal-boundary intersection: A new method for generating the Pareto surface in nonlinear multicriteria optimization problems. SIAM Journal of Optimization **8**(3), 631–657 (1998)
3. Deb, K., Jain, H.: An evolutionary many-objective optimization algorithm using reference-point based non-dominated sorting approach, Part I: Solving problems with box constraints. IEEE Transactions on Evolutionary Computation **18**(4), 577–601 (2014)
4. Deb, K., Saha, A.: Multimodal optimization using a bi-objective evolutionary algorithms. Evolutionary Computation Journal **20**(1), 27–62 (2012)
5. Deb, K., Srinivasan, A.: Innovization: Innovating design principles through optimization. In: Proceedings of the Genetic and Evolutionary Computation Conference (GECCO-2006). pp. 1629–1636. New York: ACM (2006)
6. Finck, S., Hansen, N., Ros, R., Auger, A.: Real-parameter black-box optimization benchmarking 2010: Presentation of the noiseless functions. Tech. Rep. 2009/20, Research Center PPE (2010)
7. Gaur, A., Deb, K.: Effect of size and order of variables in rules for multi-objective repair-based innovization procedure. In: Proceedings of Congress on Evolutionary Computation (CEC-2017) Conference. Piscatway, NJ: IEEE Press (2017)
8. Ghosh, A., Goodman, E., Deb, K., Averill, R., Diaz, A.: A large-scale bi-objective optimization of solid rocket motors using innovization. In: 2020 IEEE Congress on Evolutionary Computation (CEC). pp. 1–8 (2020)
9. Goldberg, D.E., Deb, K.: A comparison of selection schemes used in genetic algorithms. In: Foundations of Genetic Algorithms 1 (FOGA-1). pp. 69–93 (1991)
10. Goldberg, D.E., Richardson, J.: Genetic algorithms with sharing for multimodal function optimization. In: Proceedings of the First International Conference on Genetic Algorithms and Their Applications. pp. 41–49 (1987)
11. Huband, S., Hingston, P., Barone, L., While, L.: A review of multiobjective test problems and a scalable test problem toolkit. IEEE Transactions on Evolutionary Computation **10**(5), 477–506 (2006)
12. Kennedy, J., Eberhart, R.: Particle swarm optimization. In: Proceedings of IEEE International Conference on Neural Networks. pp. 1942–1948. IEEE Press (1995)
13. Mittal, S., Saxena, D.K., Deb, K.: Learning-based multi-objective optimization through ANN-assisted online innovization. In: Proceedings of the 2020 Genetic and Evolutionary Computation Conference Companion. pp. 171–172. GECCO 20, Association for Computing Machinery, New York, NY, USA (2020)
14. Mittal, S., Saxena, D.K., Deb, K., Goodman, E.: An ANN-assisted repair operator for evolutionary multiobjective optimization. Tech. Rep. COIN Report No. 2020005, Michigan State University, East Lansing, USA (2020), https://www.egr.msu.edu/∼kdeb/papers/c2020005.pdf
15. Mittal, S., Saxena, D.K., Deb, K., Goodman, E.: Enhanced innovized repair operator for evolutionary multi- and many-objective optimization. Tech. Rep. COIN Report No. 2020020, Michigan State University, East Lansing, USA (2020), https://www.egr.msu.edu/∼kdeb/papers/c2020020.pdf

16. Padhye, N., Deb, K., Mittal, P.: Boundary handling approaches in particle swarm optimization. In: Bansal, J., Singh, P., Deep, K., Pant, M., Nagar, A. (eds.) Proceedings of Seventh International Conference on Bio-Inspired Computing: Theories and Applications (BIC-TA 2012). Advances in Intelligent Systems and Computing. vol. 201, pp. 287–298. Springer, India (2013)
17. Pétrowski, A.: A clearing procedure as a niching method for genetic algorithms. In: IEEE 3rd International Conference on Evolutionary Computation (ICEC'96). pp. 798–803 (1996)
18. Wierzbicki, A.P.: The use of reference objectives in multiobjective optimization. In: Fandel, G., Gal, T. (eds.) Multiple Criteria Decision Making Theory and Applications, pp. 468–486. Berlin: Springer-Verlag (1980)
19. Zitzler, E., Deb, K., Thiele, L.: Comparison of multiobjective evolutionary algorithms: Empirical results. Evolutionary Computation **8**(2), 173–195 (2000)