Adaptive Use of Innovization Principles for a Faster Convergence of Evolutionary Multi-Objective Optimization Algorithms

Abhinav Gaur*
Electrical & Computer Engineering Department
Michigan State University, East Lansing, MI
gaurabhi@msu.edu

Kalyanmoy Deb
Electrical & Computer Engineering Department
Michigan State University, East Lansing, MI
kdeb@egr.msu.edu

ABSTRACT

"Innovization" is a task of learning common principles that exist among some or all of the Pareto-optimal solutions of a multi-objective optimization problem. Except a few earlier studies, most innovization related studies were performed on the final non-dominated solutions found by an EMO algorithm. Since the innovization principles are properties of good and near-optimal solutions, an early identification of them can help improve the evolving population to converge quicker to the Pareto-optimal set. This paper advocates the discovery of innovized principles through machine learning methods during an evolutionary multi-objective optimization run and then using these principles to repair the population adaptively to achieve a faster convergence. Implementing this idea with linear regression as the learning tool and applying it in a test problem with power-law rules existing among Pareto-optimal solutions yields encouraging results. The results show not only an improvement in convergence rate but also in the diversity of non-dominated solutions.

CCS Concepts

•**Applied computing → Multi-criterion optimization and decision-making;** *Decision analysis;*

Keywords

Multi-objective Optimization; Innovization; Convergence

1. INTRODUCTION

All multi-objective optimization (MOO) problems possess a unique property of having not one, but a set of "equally good" or Pareto-optimal (PO) solutions. The idea of learning from the PO solutions and deciphering design principles that are unique to the optimization problem was first presented in [4]. The works [1, 6] show that innovization is an effective tool to discover new design principles in MOO problems.

However, the works [2, 7] point to a growing trend of coupling the ideas of EMO with that of innovization in a way, that makes innovization an integral part of the optimization process itself and not just a post optimality analysis tool. We refer to this new method as '*evolutionary multi-objective optimization with innovization*' or EMO/I method in this paper and develop a prototype of such an EMO/I algorithm.

2. METHODOLOGY

Figure 1 shows the proposed EMO/I algorithm. We test this algorithm on MOO problems whose PO solutions adhere to some power-law rule of the form shown in (1). Each of the M rules $\Lambda_i,\ i \in \{1, \ldots, M\}$ in (1) is composed of two basis functions depicted by λ's here. These basis functions are some function of the design variables and are suggested by the user. The parameters of each rule, i.e. b_i and c_i, are learned on the fly during the optimization run using linear regression with log-linear modeling. The user also specifies the minimum acceptable significance of some rule in terms of learning error, say ρ, before repairs are made to design variables of population individuals based on the rule learned during optimization.

In Figure 1, the blocks that differentiate the algorithm from a regular EMO are labeled with numbers 1 to 5 in gray circles. In block-1, user specifies the rule information for all rules, Λ_i, and minimum significance parameter, ρ. Learning should not begin very early in the optimization as initially the solutions are far away from the optima. This is shown by decision $C2$ in block-2. Example of $C2$ are, "Have certain minimum generations passed?" or "Are all solutions are in front-1". If the population has reached near optima, the *Machine Learning* block (block-3) engages and returns the parameter estimates and the residual error for each rule. Block-4 finds out if the desired rules have converged better than the minimum expected significance level. If so, then algorithm uses the estimated parameters b_i and c_i for each rule and makes a variable repair based on the rule.

$$\begin{aligned}&\Lambda_i \equiv \lambda_{i_1}(\mathbf{x}) \cdot \lambda_{i_2}(\mathbf{x})^{b_i} = c_i \ \forall\, i, \text{ where,}\\ &i \in \{1, \cdots, M\} \text{ and } b_i, c_i \in \mathbb{R}.\end{aligned} \tag{1}$$

We tested our prototype EMO/I algorithm on a modified ZDT1 problem [8]. The problem and its PO solutions are

*http://www.coin-laboratory.com

GECCO '16 July 20-24, 2016, Denver, CO, USA

ACM ISBN 978-1-4503-4323-7/16/07.
DOI: http://dx.doi.org/10.1145/2908961.2909019

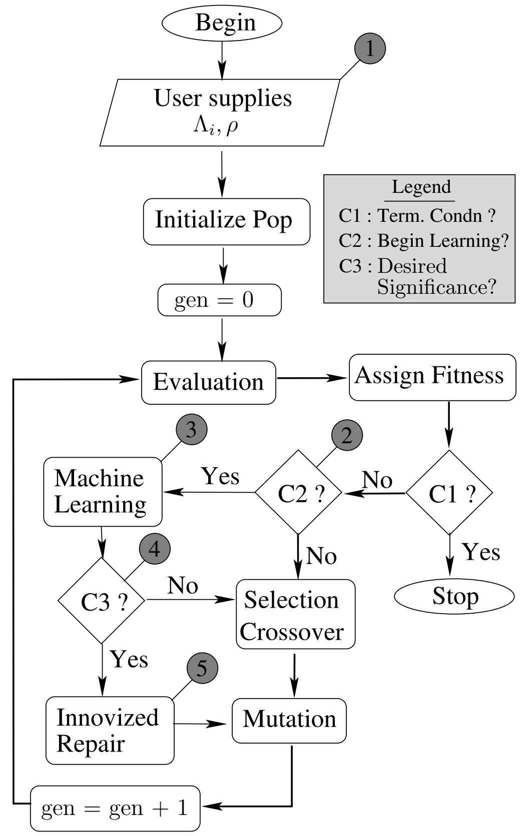

Figure 1: The proposed EMO/I algorithm.

given in (2).

$$
\begin{aligned}
&\text{Minimize } f_1(\mathbf{x}) = x_1,\ f_2(\mathbf{x}) = g(\mathbf{x})\ h(f_1(\mathbf{x}), g(\mathbf{x})),\\
&\quad \text{Where } g(\mathbf{x}) = 1 + 9\left(x_2\sqrt{x_3} - 0.5\right)^2,\ h(f_1, g) = 1 - \sqrt{\frac{f_1}{g}},\\
&\qquad\qquad x_i \in [0,1],\ \forall\, i \in \{1,2,3\},\\
&\text{PO solns } 0.0 \le x_1^* \le 1.0,\ x_2^*\sqrt{x_3^*} - 0.5 = 0.
\end{aligned} \tag{2}
$$

3. RESULTS

Here we present the results for the problem discussed in Section 2. The two algorithms under comparison here are (a) NSGA-II [3] and (b) NSGA-II with innovization (EMO/I). The two algorithms are compared on convergence rate, using median *generational distance* (GD), and diversity, using median *spread* over 100 runs. Figures 2b and 2a show the results and clearly EMO/I outperforms vanilla NSGA-II on both counts.

4. CONCLUSIONS

We are able to show that EMO algorithms can take advantage of innovization task on the fly to improve convergence rate of optimization. In future, we will make this method more adaptive in terms of the kind of design rules it can handle as well as including objective functions and constraints as basis functions in the EMO/I. Of course there is this concern about how much extra computation is spent on

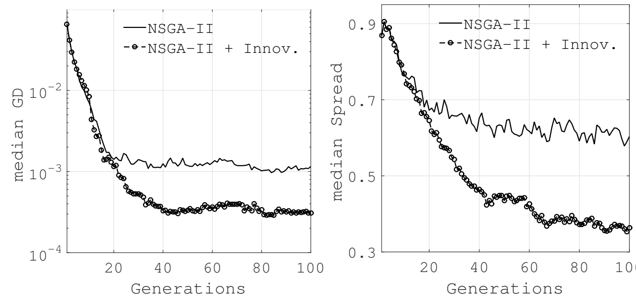

Figure 2: Results for ZDT1m problem.

learning. In current results, a population size of 8 was used as we tried to simulate a scenario where objective function evaluation is very expensive. In other problems as well, we are confident of taking advantage of vast body of efficient problem specific Machine Learning algorithms and couple it with MOO under the innovization framework to justify the extra computational effort. See the full report on this work in [5].

5. ACKNOWLEDGMENTS

The authors would like to thank the BEACON, an NSF center for the study of evolution in action at MSU, for their support towards this research.

6. REFERENCES

[1] K. Deb, S. Bandaru, D. Greiner, A. Gaspar-Cunha, and C. C. Tutum. An integrated approach to automated innovization for discovering useful design principles: Case studies from engineering. *Applied Soft Computing*, 15:42–56, 2014.

[2] K. Deb and R. Datta. Hybrid evolutionary multi-objective optimization and analysis of machining operations. *Engineering Optimization*, 44(6):685–706, 2012.

[3] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan. A fast and elitist multiobjective genetic algorithm: Nsga-ii. *Evolutionary Computation, IEEE Transactions on*, 6(2):182–197, 2002.

[4] K. Deb and A. Srinivasan. Innovization: Innovating design principles through optimization. In *Proceedings of the 8th annual conference on Genetic and evolutionary computation*, pages 1629–1636. ACM, 2006.

[5] A. Gaur and K. Deb. Adaptive Use of Innovization Principles for a Faster Convergence. http://www.egr.msu.edu/˜kdeb/papers/c2016003.pdf. [COIN Technical Report 2016003].

[6] A. Ng, K. Deb, and C. Dudas. Simulation-based innovization for production systems improvement: An industrial case study. In *Proceedings of The International 3rd Swedish Production Symposium, SPS'09, GoÌĹteborg, Sweden, 2-3 December 2009*, pages 278–286. The Swedish Production Academy, 2009.

[7] A. H. Ng, C. Dudas, H. Boström, and K. Deb. Interleaving innovization with evolutionary multi-objective optimization in production system simulation for faster convergence. In *Learning and Intelligent Optimization*, pages 1–18. Springer, 2013.

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