k-Pareto Optimality for Many-Objective Genetic Optimization

Jean Ruppert
jean.ruppert@mathcomp.lu
Mathematics and Computing S.à.r.l.
Luxembourg city, Luxembourg,
http://www.mathcomp.lu/

Marharyta Aleksandrova
Thomas Engel
first.larst@uni.lu
University of Luxembourg
Esch-sur-Alzette, Luxembourg

ABSTRACT

The Pareto dominance-based evolutionary many-objective optimization methods are known to suffer from the deterioration of searchability. We propose to redefine the calculation of Pareto-dominance. Instead of assigning solutions to non-dominated fronts, we rank them according to the number of dominating solutions or the probability of being dominated. Through the experimental results from a 0/1 knapsack problem, we demonstrate the advantages of this probabilistic approach: 1) it allows to increase the hypervolume for both the multi- and many-objective optimization problems; 2) in the case of many-objective optimization, it results in better fitted solutions as compared to *NSGA-II* and *NSGA-III*.

CCS CONCEPTS

• **Computing methodologies** → **Search methodologies**.

KEYWORDS

Multi-objective and many-objective optimization, Selection

ACM Reference Format:
Jean Ruppert, Marharyta Aleksandrova, and Thomas Engel. 2021. k-Pareto Optimality for Many-Objective Genetic Optimization. In *2021 Genetic and Evolutionary Computation Conference Companion (GECCO '21 Companion), July 10–14, 2021, Lille, France.* ACM, New York, NY, USA, 2 pages. https://doi.org/10.1145/3449726.3462732

1 PROBLEM FORMULATION

Optimization problems with 2 or 3 objectives are known as *multi-objective*; in case of higher dimensionality, they are referred to as *many-objective* [5]. Several algorithms exist to effectively solve multi-objective optimization problems, for example, *NSGA-II* [2]. Also, the non-dominated sorting procedure is very effective in this case [9]. However, Pareto dominance-based many-objective optimization evolutionary algorithms face various difficulties. One of them is the deterioration of the searchability due to the lack of selection pressure [7]. Indeed, when the number of objectives increases, the number of incomparable solutions grows exponentially. Several alternative approaches were proposed to overcome this problem. Among them, there are the relaxed dominance-based approaches [6], the indicator-based approaches [8], and the reduction of the number of objectives via scalarization [3]. However, all these approaches are associated with some disadvantages. In particular, the indicator-based approaches require calculating the value of the relative indicator function; scalarization-based methods require either running several single-objective optimizations during one run or running individual single-objective optimization over many runs [3]; finally, as stated in [6], diversity maintenance can become more difficult in relaxed dominance-based approaches.

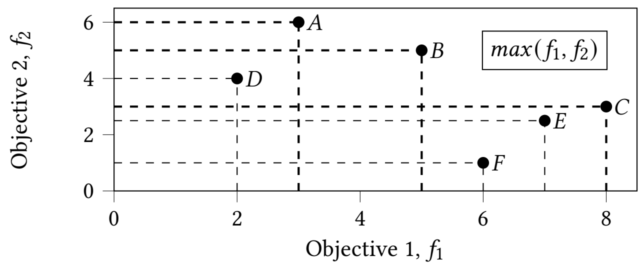

Figure 1: Illustration of sorting

2 k-PARETO OPTIMALITY

To illustrate the Pareto dominance based sorting, let us consider a 2-objective maximization problem with a set of solutions presented in Fig. 1. First, we identify all non-dominated solutions and associate them with the first front. In our example there are 3 non-dominated solutions: A, B and C. Next, we remove the solutions of the newly identified front from the consideration, and repeat the process. As a result, we obtain the second front made up of 2 solutions: D and E, and the third front with only one solution F. Instead of sorting by Pareto dominance, we propose to sort the solutions by the number of other solutions that dominate the current one. We refer to this value as *k-Pareto optimality*, where k stands for the number of dominating solutions. Examining Fig. 1, we infer that, for this case, the following three fronts are formed: front 1 with solutions A, B, C and $k = 0$; front 2 with solution E and $k = 1$; and front 3 with solutions D, F and $k = 2$. Note that this sorting procedure placed solution D to the third front, but it was in the second front according to non-dominated sorting by Pareto dominance. The calculation of k can also be performed using a probabilistic measure. Indeed, instead of calculating the number of solutions that dominate the current solution, we can estimate the probability of this solution being dominated. Assuming independence between the objectives, we can approximate this value as the product of probabilities of the solution being dominated according to every objective independently.

3 EXPERIMENTAL RESULTS

We evaluate the proposed sorting procedure, by using it in *NSGA-II* instead of Pareto dominance-based sorting. The resulting algorithms are referred to as *PO-count* and *PO-prob* depending on

GECCO '21 Companion, July 10–14, 2021, Lille, France

ACM ISBN 978-1-4503-8351-6/21/07.
https://doi.org/10.1145/3449726.3462732

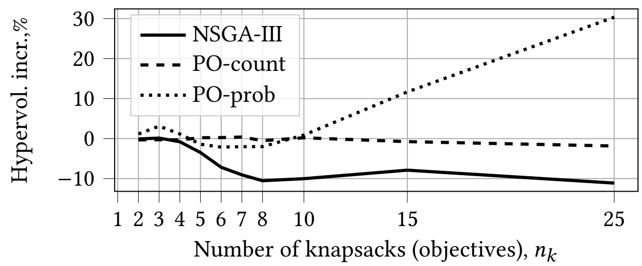

Figure 2: Increase in hypervolume compared to NSGA-II.

whether counting or probabilistic k-Pareto optimality is adopted. These algorithms are compared with implementations of *NSGA-II* and *NSGA-III* [1] from the *deap* python library[1]. For the experimental evaluation we use the 0/1 knapsack problem with independent objectives as defined in [10]. The number of knapsacks or objectives is varied within the range $n_k \in \{2-8, 10, 15, 25\}$ and the number of items is set to 250. We adopt binary tournament selection with replacement and uniform crossover with mutation probability 0.01. We set the population size to 250 and the number of generations to 500. To measure the performance, we use two metrics: *hypervolume* with the origin of coordinates as a reference point, and the *fraction of solutions dominated* by other algorithms. The results presented below are the average among 30 independent runs.

We choose *NSGA-II* as the baseline, and present the relative changes in the **hypervolume** indicator for the rest of the algorithms in Fig. 2 (increase: positive number, decrease: negative number). We notice that despite having been developed for the many-objective optimization, *NSGA-III* almost always results in lower values of hypervolume, even for a large number of knapsacks. This confirms a similar observation from [4], and supports our choice of *NSGA-II* as a baseline for implementation and comparison instead of *NSGA-III*. Further, we see that the value of relative increase for *PO-count* is always very close to 0. It means that *PO-count* yields a population covering the same hypervolume as *NSGA-II*. Contrarily, *PO-prob* improves the hypervolume, as compared to *NSGA-II*. This difference is visible for small n_k, +3% for $n_k = 3$, and is especially prominent for large n_k, +30.35% for $n_k = 25$. For n_k between 5 and 8, *PO-prob* results in lower values of hypervolume than *NSGA-II*. However, the relative decrease does not exceed −2.13%. Also, within this range *PO-count* performs slightly better than other algorithms.

We calculate the percentage of **dominated solutions** as follows. For a given pair of algorithms *algorithm1* and *algorithm2* we calculate how many solutions of *algorithm2* (*dominated algorithm*) are dominated by solutions of *algorithm1* (*dominating algorithm*). After that, we average the obtained results among all *dominating algorithms* to get an average fraction of dominated solutions, denoted as θ. We present the corresponding results in Fig. 3. We notice the following tendencies. *NSGA-II* and *PO-count* behave very similarly. For $n_k = 2$, the value of θ for these algorithms is around 25%. After that, it starts increasing and reaches its peak of approximately 45% for $n_k = 7$. Finally, it gradually decreases to 20% for $n_k = 25$. *NSGA-III* starts at a similar level and reaches its peak of approximately 30% for $n_k = 3$. After that, it decreases below 10% for $n_k = 7$ and stays relatively close to 0 for the large numbers of knapsacks. These

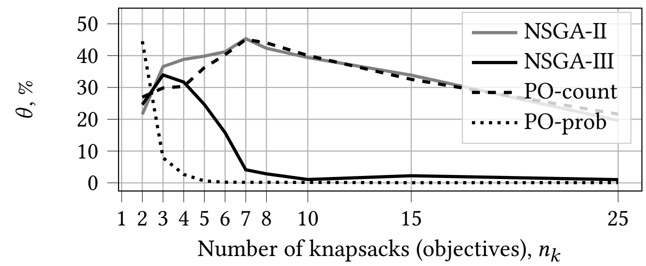

Figure 3: Average percentage of solutions dominated by other algorithms, θ.

results demonstrate the superiority of *NSGA-III* over *NSGA-II* in case of many-objective optimization. *PO-prob* starts at around 45%. However, for $n_k = 3$ it is already below 10% and for $n_k = 5$ it is almost 0. This shows that the solutions produced by this algorithm are rarely dominated. Thereby, *PO-prob* is an effective approach for many-objective optimization problems.

4 CONCLUSIONS

We proposed a novel sorting procedure based on k-Pareto optimality, where k stands for the number of solutions dominating the current solution. In the case of independent objectives, this sorting procedure can be approximated as a multiplication of probabilities of the current solution being dominated for the single objectives. We incorporate both proposed sorting procedures into *NSGA-II* and compared them experimentally with *NSGA-II* and *NSGA-III* using the 0/1 knapsack problem with various numbers of objectives. We showed that the probabilistic k-Pareto optimality allows an increased hypervolume for both multi- (up to +3%) and many-objective optimization problems (up to +30%). Additionally, for a large number of objectives, it results in better fitted solutions.

REFERENCES

[1] Kalyanmoy Deb and Himanshu Jain. 2013. An evolutionary many-objective optimization algorithm using reference-point-based nondominated sorting approach, Part I: solving problems with box constraints. *IEEE Transactions on Evolutionary Computation* 18, 4 (2013), 577–601.
[2] Kalyanmoy Deb, Amrit Pratap, Sameer Agarwal, and TAMT Meyarivan. 2002. A fast and elitist multiobjective genetic algorithm: NSGA-II. *IEEE Transactions on Evolutionary Computation* 6, 2 (2002), 182–197.
[3] Evan J Hughes. 2005. Evolutionary many-objective optimisation: many once or one many?. In *2005 IEEE Congress on Evolutionary Computation*, Vol. 1. IEEE, 222–227.
[4] Hisao Ishibuchi, Ryo Imada, Yu Setoguchi, and Yusuke Nojima. 2016. Performance comparison of NSGA-II and NSGA-III on various many-objective test problems. In *2016 IEEE Congress on Evolutionary Computation (CEC)*. IEEE, 3045–3052.
[5] Hisao Ishibuchi, Noritaka Tsukamoto, and Yusuke Nojima. 2008. Evolutionary many-objective optimization: a short review. In *2008 IEEE Congress on Evolutionary Computation*. IEEE, 2419–2426.
[6] Bingdong Li, Jinlong Li, Ke Tang, and Xin Yao. 2015. Many-objective evolutionary algorithms: a survey. *ACM Computing Surveys (CSUR)* 48, 1 (2015), 1–35.
[7] Vikas Palakonda, Samira Ghorbanpour, and Rammohan Mallipeddi. 2018. Pareto dominance-based MOEA with multiple ranking methods for many-objective optimization. In *2018 IEEE Symposium Series on Computational Intelligence (SSCI)*. IEEE, 958–964.
[8] Tobias Wagner, Nicola Beume, and Boris Naujoks. 2007. Pareto-, aggregation-, and indicator-based methods in many-objective optimization. In *International Conference on Evolutionary Multi-criterion Optimization*. Springer, 742–756.
[9] Xingyi Zhang, Ye Tian, Ran Cheng, and Yaochu Jin. 2014. An efficient approach to non-dominated sorting for evolutionary multi-objective optimization. *IEEE Transactions on Evolutionary Computation* 19, 2 (2014), 201–213.
[10] Eckart Zitzler and Lothar Thiele. 1999. Multiobjective evolutionary algorithms: a comparative case study and the strength Pareto approach. *IEEE Transactions on Evolutionary Computation* 3, 4 (1999), 257–271.

[1]https://deap.readthedocs.io/en/master/