Lightweight Interpolation-Based Surrogate Modelling for Multi-objective Continuous Optimisation

Alexandru-Ciprian Zăvoianu[1,2(✉)], Benjamin Lacroix[1,2], and John McCall[1,2]

[1] School of Computing, Robert Gordon University, Aberdeen, Scotland, UK
{c.zavoianu,b.m.e.lacroix,j.mccall}@rgu.ac.uk
[2] National Subsea Centre, Aberdeen, Scotland, UK

Abstract. We propose two surrogate-based strategies for increasing the convergence speed of multi-objective evolutionary algorithms (MOEAs) by stimulating the creation of high-quality individuals early in the run. Both offspring generation strategies are designed to leverage the fitness approximation capabilities of light-weight interpolation-based models constructed using an inverse distance weighting function. Our results indicate that for the two solvers we tested with, NSGA-II and DECMO2++, the application of the proposed strategies delivers a substantial improvement of early convergence speed across a test set consisting of 31 well-known benchmark problems.

Keywords: Surrogate modelling · Multi-objective continuous optimisation · Evolutionary algorithms · Run-time convergence analysis

1 Introduction and Motivation

A multi-objective optimisation problem (MOOP) over a multi-dimensional space (i.e., $x \in V^d \subset \mathbb{R}^d$) can be defined as:

$$\text{minimize } F(x) = (f_1(x), \ldots, f_m(x))^T, \tag{1}$$

with the understanding that the $m \in \{2, 3\}$ real-valued objectives of $F(x)$ need to be minimized simultaneously. The general solution of a MOOP is given by a Pareto optimal set (PS) that collects all solutions $x^* \in V^d$ that are not fully dominated – i.e., $\nexists y \in V^d : f_i(y) \leq f_i(x^*), \forall i \in \{1, \ldots, m\}$ and $F(y) \neq F(x^*)$. The true Pareto front (PF) is the objective space projection of the PS.

Multi-objective evolutionary algorithms (MOEAs) have emerged as very popular MOOP solvers due to their ability to discover high-quality PS approximations called Pareto non-dominated sets (PNs) after single optimisation runs [1,19]. The successful application of MOEAs to increasingly complex industrial MOOPs ranging from product design [10] to calibration [6] and quality assurance [16] has also helped to highlight that when the process of evaluating

R. Moreno-Díaz et al. (Eds.): EUROCAST 2022, LNCS 13789, pp. 53–60, 2022.
https://doi.org/10.1007/978-3-031-25312-6_6

$F(x)$ is computationally-intensive[1], the effectiveness of the solver can be severely impacted as far fewer candidate solutions/individuals $x \in V^d$ can be evaluated during the optimisation run. One of the most promising approaches for alleviating the effect of expensive $F(x)$ formulations is to replace the original fitness functions with an easy to evaluate surrogate formulation [9,11]. However, the task of constructing accurate (non-linear) surrogate models is non-trivial and can itself be computationally intensive, especially when performed on-the-fly (i.e., during the optimisation process) [14].

The present work aims to contribute to on-the-fly surrogate construction by exploring the lightweight modelling[2] capabilities of the recently introduced multi-objective interpolated continuous optimisation problem (MO-ICOP) formulation [15] based on inverse distance weighting.

2 Research Focus and Approach

2.1 Lightweight Interpolation-Based Surrogate Model

Let us denote with $e(x, y)$ the Euclidean distance between two individuals x and y and with $X = \{x_1, ..., x_N\}$ the set containing all the N individuals evaluated by a MOEA till a given stage of its execution. Using Shepard's inverse distance weighting function [12], we can estimate the fitness of any new solution candidate $y \in V^d$ across each individual objective f_i from Eq. 1 as:

$$g_{X,f_i,k}(y) = \begin{cases} \frac{\sum_{j=1}^{N} \frac{f_i(x_j)}{e(y,x_j)^k}}{\sum_{j=1}^{N} \frac{1}{e(y,x_j)^k}} & \text{if } e(y, x_j) \neq 0 \text{ for all } j \\ f_i(x_j), & \text{if } e(y, x_j) = 0 \text{ for some } j \end{cases}, \forall 1 \leq i \leq n \quad (2)$$

where k is a positive real number called the power parameter. The final lightweight multi-objective surrogate for $F(x)$ is obtained by simply aggregating the individual interpolation-based models:

$$g_{X,F,k}(y) = (g_{X,f_1,k}(y), ..., g_{X,f_m,k}(y))^T, \forall 2 \leq m \leq 3. \quad (3)$$

We propose two offspring generation strategies that leverage the lightweight interpolation-based surrogate model from Eq. 3 to discover high-quality individuals during the MOEA evolutionary cycle and thus reduce the number of fitness evaluations necessary for producing high-quality PN approximations. This will translate directly into a reduction of the prohibitive run-times observed when applying MOEAs on MOOPs with computationally-intensive fitness evaluations.

It is noteworthy that across all the optimisation runs we carried out, both subsequently described strategies were mostly beneficial in the early and middle stages of convergence. Therefore, our recommendation is to only apply them during the first $gTh\%$ generations, where gTh is a control parameter.

[1] As it is based on numerical simulation(s) or even human-in-the-loop experiment(s).
[2] Modelling that requires virtually no training.

2.2 Offspring Generation Strategy 1: Pre-emptive Evaluation (PE)

The first strategy aims to improve any MOEA-specific approach of generating a new individual (i.e., **EvolveNextOffspring**) by using a (very fast) surrogate-based pre-emptive evaluation of fitness to stimulate the creation of offspring that have high quality - i.e., a very high likelihood of improving the current PN stored by the MOEA. If a new $offspring$ does not pass the high-quality test, it will generally be disregarded. However, if the number of failed consecutive attempts to generate a high-quality offspring exceeds a certain threshold (i.e., $|solverPop| \cdot \frac{gTh}{100}$), a default acceptance criterion is triggered. In order to pass the high-quality test (lines 9 to 24 in Algorithm 1), when comparing with at least one member of the parent population, the new offspring must simultaneously:

- be better by at least $minImprTh\%$ on at least one objective;
- not be worse by more than $simTh\%$ on any objective;

Algorithm 1. The pre-emptive evaluation (PE) strategy

```
1: function PreEvalOffspring(solverPop, G_{X,F,k}, simTh, minImprTh, gTh)
2:     passed ← false ∧ rejections ← 0
3:     while ¬passed and rejections < |solverPop| · gTh/100 do
4:         rejections ← rejections + 1
5:         offspring ← EvolveNextOffspring(solverPop)
6:         for i = 0 to m do
7:             offspring.obj(i) ← g_{X,f_i,k}(offspring)
8:         end for
9:         for all p in solverPop do
10:            simObj ← 0
11:            domObj ← 0
12:            for i = 0 to m do
13:                if p.obj(i) ·(1 + simTh) > offspring.obj(i) then
14:                    simObj ← simObj + 1
15:                end if
16:                if p.obj(i) ·(1 − minImprTh) > offspring.obj(i) then
17:                    domObj ← domObj + 1
18:                end if
19:            end for
20:            if domObj ≥ 1 and simObj = m then
21:                passed ← true
22:                rejections ← rejections − 1
23:                break
24:            end if
25:        end for
26:    end while
27:    return ⟨offspring, passed⟩
28: end function
```

The result of calling the **PreEvalOffspring** function is an ordered pair containing a newly generated $offspring$ and a Boolean flag indicating if it has *passed* the high-quality test (or, conversely, has been accepted by default). The Boolean data is used to dynamically adjust the evolutionary pressure exerted by the pre-emptive evalution strategy. Thus, whenever observing more than gTh default accepts / generation, the required objective improvement threshold for subsequent offspring is reduced using the formula:

$$minImprTh = minImprTh \cdot \left(1 - \frac{gTh}{100}\right) \quad (4)$$

We highlight that successive reductions of $minImprTh$ using Eq. 4 can rapidly lead to a situation where $minImprTh < simTh$. This signals that the evolutionary search is at a stage where the lightweight interpolation-based surrogates cannot easily identify offspring that are likely to bring major improvements to the current PN stored by the MOEA. As such, we opted to reduce our application of both surrogate strategies during the first $gTh\%$ generations to instances where $minImprTh \geq simTh$.

2.3 Offspring Generation Strategy 2: Speculative Exploration (SE)

Algorithm 2 describes our second approach for providing any (generational) MOEA with a simple means of creating high-quality offspring.

Here, the idea is to construct a (surrogate) multi-objective interpolated continuous optimisation problem ($MO - ICOP$) that mirrors the definition of the original *problem* to be solved (line 3 in Algorithm 2). The only difference is that the surrogate MO-ICOP uses the model from Eq. 3 as its fitness function (instead of the original from Eq. 1). An internal solver is then applied on the surrogate MO-ICOP and an elite subset of individuals, randomly extracted from the final population of the internal solver, will form the final result of the speculative exploration of the search space using the surrogate models. Finally, the $|solverPop| \cdot \frac{gTh}{100}$ surrogate-based elites returned by the **SpecExploreOffspring** function will be subsequently treated like regular offspring inside the evolutionary cycle of the main MOEA.

Algorithm 2. The speculative exploration (SE) strategy

1: **function** **SpecExploreOffspring**($solverPop$, gTh, $problem$, $G_{X,F,k}$)
2: $eliteSetSize \leftarrow |solverPop| \cdot \frac{gTh}{100}$
3: $MO - ICOP \leftarrow$ **CreateSurrogateProblem**($problem, G_{X,F,k}$)
4: $offspringPop \leftarrow$ **InternalSolver**($MO - ICOP$)
5: $offspringPop \leftarrow$ **RandomFilter**($offspringPop, eliteSetSize$)
6: **return** $offspringPop$
7: **end function**

2.4 Tentative Parameterisation of Proposed Strategies

In order to control the two proposed surrogate-based offspring creation strategies, one needs to parameterise three thresholds (gTh, $simTh$, $minImprTh$) and k – the power parameter from Eq. 2.

gTh can be seen as a general control parameter for surrogate usage and current experiments indicate that the setting $gTh = 10$ produces good results for all tested MOEAs across a wide range of problems.

Given that an ideal characteristic of any PN is to provide a well-spread approximation of the true PF, a reasonable setting for the similarity threshold is $simTh = \frac{1}{|solverPop|}$. In order to increase the chances that the PE strategy identifies high-quality offspring despite the high uncertainty associated with light-weight surrogate estimations (especially in the first few generations), the value of the minimal improvement threshold should be a multiple of $simTh$. We recommend the setting $minImprTh = gTh \cdot simTh + \epsilon$ as it generally yields competitive results and reduces the parameterisation overload.

Higher values of k produce interpolation models with wider attraction basins around the seed points (i.e., X in Eq. 2). Our experiments indicate that surrogate fitness landscapes obtained with the setting $k \geq 10$ work better than those generated by $1 \leq k < 10$ across both offspring generation strategies. All the results from Sect. 4 were obtained with the setting $k = 20$.

3 Exprimental Setup

We've integrated and tested the PE and SE strategies with two solvers. The first one is the well-known NSGA-II [2] - a multi-objective evolutionary algorithm that relies on a highly Pareto elitist two-tier selection for survival (i.e., filtering) operator to obtain the population of generation $t+1$ from the population of generation t and the offspring generated at generation t. The first filtering criterion aims to retain Pareto non-dominated individuals whilst the second one (used for tie-breaking) aims to avoid overcrowding in objective space.

The second solver we experimented with is DECMO2++ [17]. It's main characteristic is the ability to converge fast across a wide range of MOOPs when using a fixed parameterisation. This is achieved by integrating and pivoting between three different multi-objective evolutionary paradigms (Pareto elitism, differential evolution [13], and decomposition [5,18]) via coevolved sub-populations. It is noteworthy to mention that the PE and SE strategies were independently integrated in each of the three sub-populations.

Across all[3] experiments, we fixed $|solverPop| = 200$ and used the standard / literature recommended parameterisation for both MOEAs. As a stopping criterion, we fixed the total number of fitness evaluations (nfe) per run to 50.000.

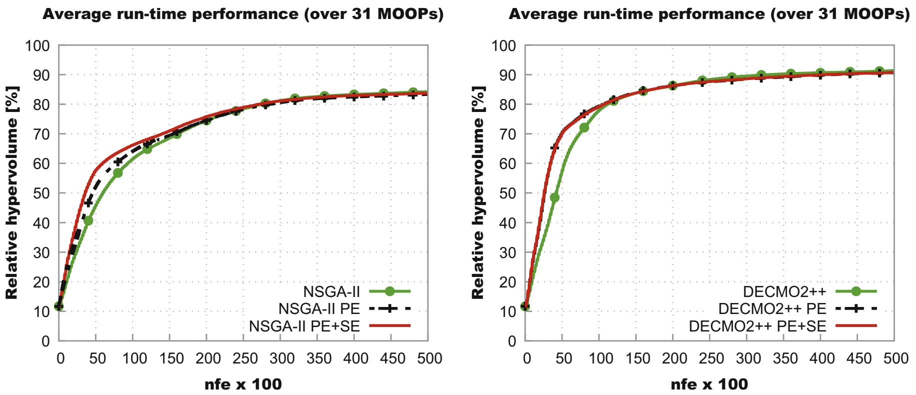
[3] Given that results from [15] indicate that NSGA-II has a competitive advantage on MO-ICOP instances where $k \geq 6$, NSGA-II was also used as the internal solver of the speculative exploration strategy (i.e., line 4 in Algorithm 2).

We performed experiments on a comprehensive test set containing 31 benchmark problems: all[4] the DTLZ[3], LZ09 [8], WFG[4] and ZDT[20] problems plus KSW10 – Kursawe's function [7] with 10 variables and 2 objectives. We carried out 50 independent optimisation runs for each MOEA-MOOP combination.

As a performance measure, we use the normalised hyervolume [21] to track the quality of the PN stored by the solvers at each stage of the optimisation. The normalised hypervolume indicates the size of a PN-dominated objective space relative to the size of the objective space dominated by the PF.

4 Results - Comparative Performance

In Fig. 1 we plot the average convergence performance of the two baseline MOEAs and their surrogate-enhanced versions across the benchmark test set.



(a) NSGA-II (b) DECMO2++

Fig. 1. Comparative convergence performance of the surrogate-enhanced and standard versions of NSGA-II and DECMO2++.

The results indicate that the usage of the pre-emptive evaluation (PE) and the speculative exploration (SE) strategies for creating (up to) the first 5000 offspring is successful in increasing the converge speed of both NSGA-II and DECMO2++. While in the case of NSGA-II, the two strategies that use lightweight interpolation-based surrogates have a compounding effect, only the pre-emptive evaluation strategy benefits DECMO2++. Furthermore, the increased selection pressure at the start of the optimisation runs prompted by the use of these strategies seems to only have a minor impact on middle and late stage MOEA performance.

The achieved improvements in terms of early convergence are noteworthy for both solvers. For example, in the case of NSGA-II, a benchmark-wide average

[4] Except ZDT5 which is not real-valued.

relative hypervolume of 60% can be reached with ≈ 50% fewer fitness evaluations when using both the SE and PE strategies. In the case of DECMO2, when using the PE strategy, a benchmark-wide average relative hypervolume of 70% can be reached with ≈ 33% fewer fitness evaluations.

5 Conclusions and Future Work

The present work shows how surrogate models can be easily derived from well-known interpolation functions and subsequently used inside two complementary offspring generation strategies to substantially improve the early convergence speed of MOEAs. The main advantage of the proposed interpolation-based approach is that it can be easily deployed in MOEA application scenarios where on-the-fly surrogate modelling is required.

Future work will revolve around extending the testing to more solvers, experimenting with different interpolation functions, and limiting the parameterisation requirements of the surrogate-based offspring generation strategies.

Acknowledgments. This work has been supported by the COMET-K2 "Center for Symbiotic Mechatronics" of the Linz Center of Mechatronics (LCM) funded by the Austrian federal government and the federal state of Upper Austria.

References

1. Coello Coello, C.A., Lamont, G.B.: Applications of multi-objective evolutionary algorithms. World Scientific (2004)
2. Deb, K., Pratap, A., Agarwal, S., Meyarivan, T.: A fast and elitist multiobjective genetic algorithm: NSGA-II. IEEE Trans. Evol. Comput. **6**(2), 182–197 (2002)
3. Deb, K., Thiele, L., Laumanns, M., Zitzler, E.: Scalable multi-objective optimization test problems. In: IEEE Congress on Evolutionary Computation (CEC 2002), pp. 825–830. IEEE Press (2002)
4. Huband, S., Barone, L., While, L., Hingston, P.: A scalable multi-objective test problem toolkit. In: Coello Coello, C.A., Hernández Aguirre, A., Zitzler, E. (eds.) EMO 2005. LNCS, vol. 3410, pp. 280–295. Springer, Heidelberg (2005). https://doi.org/10.1007/978-3-540-31880-4_20
5. Jaszkiewicz, A.: On the performance of multiple-objective genetic local search on the 0/1 knapsack problem - A comparative experiment. IEEE Trans. Evol. Comput. **6**(4), 402–412 (2002)
6. Kaji, H.: Automotive engine calibration with experiment-based evolutionary multi-objective optimization. Ph.D. thesis, Kyoto University (2008)
7. Kursawe, F.: A variant of evolution strategies for vector optimization. In: Schwefel, H.-P., Männer, R. (eds.) PPSN 1990. LNCS, vol. 496, pp. 193–197. Springer, Heidelberg (1991). https://doi.org/10.1007/BFb0029752
8. Li, H., Zhang, Q.: Multiobjective optimization problems with complicated Pareto sets, MOEA/D and NSGA-II. IEEE Trans. Evol. Comput. **13**(2), 284–302 (2009)
9. Loshchilov, I., Schoenauer, M., Sebag, M.: A mono surrogate for multiobjective optimization. In: Proceedings of the 12th annual Conference on Genetic and Evolutionary Computation (GECCO), pp. 471–478. ACM (2010)

10. Oyama, A., Kohira, T., Kemmotsu, H., Tatsukawa, T., Watanabe, T.: Simultaneous structure design optimization of multiple car models using the K computer. In: IEEE Symposium Series on Computational Intelligence (2017)
11. Pilát, M., Neruda, R.: Hypervolume-based local search in multi-objective evolutionary optimization. In: Proceedings of the 2014 Annual Conference on Genetic and Evolutionary Computation, pp. 637–644. ACM (2014)
12. Shepard, D.: A two-dimensional interpolation function for irregularly-spaced data. In: Proceedings of the 1968 23rd ACM National Conference, pp. 517–524 (1968)
13. Storn, R., Price, K.V.: Differential evolution - a simple and effcient heuristic for global optimization over continuous spaces. J. Global Optim. **11**(4), 341–359 (1997)
14. Zăvoianu, A.C., Bramerdorfer, G., Lughofer, E., Silber, S., Amrhein, W., Klement, E.P.: Hybridization of multi-objective evolutionary algorithms and artificial neural networks for optimizing the performance of electrical drives. Eng. Appl. Artif. Intell. **26**(8), 1781–1794 (2013)
15. Zăvoianu, A.-C., Lacroix, B., McCall, J.: Comparative run-time performance of evolutionary algorithms on multi-objective interpolated continuous optimisation problems. In: Bäck, T., et al. (eds.) PPSN 2020. LNCS, vol. 12269, pp. 287–300. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-58112-1_20
16. Zăvoianu, A.C., Lughofer, E., Pollak, R., Eitzinger, C., Radauer, T.: A soft-computing framework for automated optimization of multiple product quality criteria with application to micro-fluidic chip production. Appl. Soft Comput. **98**, 106827 (2021)
17. Zăvoianu, A.C., Saminger-Platz, S., Lughofer, E., Amrhein, W.: Two enhancements for improving the convergence speed of a robust multi-objective coevolutionary algorithm. In: Proceedings of the Genetic and Evolutionary Computation Conference, pp. 793–800. ACM (2018)
18. Zhang, Q., Li, H.: MOEA/D: A multi-objective evolutionary algorithm based on decomposition. IEEE Trans. Evol. Comput. **11**(6), 712–731 (2007)
19. Zhou, A., Qu, B.Y., Li, H., Zhao, S.Z., Suganthan, P.N., Zhang, Q.: Multiobjective evolutionary algorithms: A survey of the state of the art. Swarm Evol. Comput. **1**(1), 32–49 (2011)
20. Zitzler, E., Deb, K., Thiele, L.: Comparison of multiobjective evolutionary algorithms: Empirical results. Evol. Comput. **8**(2), 173–195 (2000)
21. Zitzler, E.: Evolutionary Algorithms for Multiobjective Optimization: Methods and Applications. Ph.D. thesis, Swiss Federal Institute of Technology (1999)