



HAL
open science

Approximating Pareto Set Topology by Cubic Interpolation on Bi-objective Problems

Yuri Marca, Hernan Aguirre, Saúl Zapotecas Martinez, Arnaud Liefoghe, Bilel Derbel, Sébastien Verel, Kiyoshi Tanaka

► **To cite this version:**

Yuri Marca, Hernan Aguirre, Saúl Zapotecas Martinez, Arnaud Liefoghe, Bilel Derbel, et al.. Approximating Pareto Set Topology by Cubic Interpolation on Bi-objective Problems. EMO 2019 - International Conference on Evolutionary Multi-Criterion Optimization, Mar 2019, East Lansing, Michigan, United States. pp.386-398, 10.1007/978-3-030-12598-1_31 . hal-02064548

HAL Id: hal-02064548

<https://hal.science/hal-02064548>

Submitted on 14 Sep 2021

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Approximating Pareto Set Topology by Cubic Interpolation on Bi-objective Problems

Yuri Marca^{1,5}, Hernán Aguirre^{1,5}, Saúl Zapotecas Martínez^{2,5},
Arnaud Liefooghe^{3,5}, Bilel Derbel^{3,5}, Sébastien Verel^{4,5}, and
Kiyoshi Tanaka^{1,5}

¹ Shinshu University, Faculty of Engineering, Japan
yurimarca@gmail.com, {ahernan, ktanaka}@shinshu-u.ac.jp

² Universidad Autónoma Metropolitana, Mexico
szapotecas@correo.cua.uam.mx

³ Université Lille 1, CRISAL – Inria Lille-Nord Europe, France
{arnaud.liefooghe,bilel.derbel}@univ-lille1.fr

⁴ Université du Littoral Cote d’Opale, LISIC, France
verel@univ-littoral.fr

⁵ International Associated Laboratory LIA-MODO

Abstract. Difficult Pareto set topology refers to multi-objective problems with geometries of the Pareto set such that neighboring optimal solutions in objective space differ in several or all variables in decision space. These problems can present a tough challenge for evolutionary multi-objective algorithms to find a good approximation of the optimal Pareto set well-distributed in decision and objective space. One important challenge optimizing these problems is to keep or restore diversity in decision space. In this work, we propose a method that learns a model of the topology of the solutions in the population by performing parametric spline interpolations for all variables in decision space. We use Catmull-Rom parametric curves as they allow us to deal with any dimension in decision space. The proposed method is appropriated for bi-objective problems since their optimal set is a one-dimensional curve according to the Karush-Kuhn-Tucker condition. Here, the proposed method is used to promote restarts from solutions generated by the model. We study the effectiveness of the proposed method coupled to NSGA-II and two variations of MOEA/D on problems with difficult Pareto set topology. These algorithms approach very differently the Pareto set. We argue and discuss their behavior and its implications for model building.

Keywords: Evolutionary algorithm · Multi-objective optimization · Interpolation · Difficult Pareto set topology.

1 Introduction

Multi-objective Evolutionary Algorithms (MOEA) are metaheuristic methods based on natural evolution principles that have attracted a lot of attention due to their good performance to deal with multi-objective optimization problems (MOP) [3]. Indeed, with the development of different MOEAs, many methodologies to improve their performance have been proposed [1, 3–5, 7, 17, 18].

Despite successful results obtained by MOEAs, studies have shown that their performance can deteriorate significantly when facing problems with difficult Pareto set (PS) topology [8]. Okabe et al. [11] observed that the PS topology of most artificial test problems, such as DTLZ [6], have an oversimplified geometry, arguing that we should not expect such simplification on real world problems. Since then, new test problems have been developed with some challenging PS topologies [7, 11], along with new approaches to solve these problems. Special sessions and competitions dedicated to solve problems with difficult PS topology [16] have served to promote research in this area and to improve some multi-objective algorithms. Nonetheless, efficiency and scalability remain an open question for improved algorithms, such as enhanced versions of decomposition-based algorithms. For other classes of algorithms, such as those based on Pareto dominance, performance in terms of convergence and diversity in both decision and objective space is still poor. Overall, besides final results, there is still not a clear understanding of how various classes of algorithms work on these classes of difficult PS topology problems.

On the other hand, learning and model assisted optimization is gaining attention to enhance evolutionary search, where models are built to capture properties of the landscape, learn dependencies between variables, identify variables for recombination, and so on. The models are in turn used to guide the evolutionary algorithm aiming to improve the overall efficiency and effectiveness of the search. Some recent works have tried to incorporate learning models for better solutions [9, 10, 12] when optimizing problems with difficult PS topology. These models try to learn certain regions in decision space where good solutions are more likely to be found, restricting and guiding the evolutionary search.

From this standpoint, in this paper we present a method that learns a model of the topology of the solutions in the population by performing parametric spline interpolations for all variables in decision space, aiming to assist multi-objective evolutionary algorithms on bi-objective problems with difficult PS topology. To build the model, we use Catmull-Rom parametric curves as they allow us to deal with any dimension in decision space. The proposed method is appropriated for bi-objective problems since their optimal set is a one-dimensional curve according to the Karush-Kuhn-Tucker condition. The model allows to identify and query regions in decision space that are under represented in the population of the evolutionary algorithm. That is, based on these polynomial interpolations, the model can be used to generate new candidate solutions well distributed in decision space aiming to guide the search towards approximations of the Pareto set with better distribution in decision and objective space.

In this work, the proposed method is used to promote restarts from solutions generated by the model. We study the effectiveness of the proposed method embedded in three algorithms: NSGA-II [5], MOEA/D [7], and MOEA/D-DRA [15]. These algorithms are good representatives of Pareto-dominance and decomposition based approaches to multi-objective optimization. These algorithms also show quite different behavior approaching the Pareto optimal solutions. While MOEA/D have a fast approach to the optimal for some of its weights, NSGA-

II slowly moves a better distributed set of solutions towards the optimal front. Since models to guide the evolutionary search are mostly built from the solution in the population, it is important to understand how the behavior of the algorithms affect the quality of the model. We test the performance of the modified algorithms using problems with difficult PS topology proposed in [16]. Simulation results presented in this work clarifies the correlation between the way an algorithm approaches the Pareto optimal set and the quality of the model, showing that the proposed method can help evolutionary algorithms to find better distributed solutions depending on algorithm’s evolutionary behavior.

The rest of the paper is organized as follows. Firstly, Section 2 elucidates the meaning of a MOP with difficult PS topology. Section 3 describes the learning model to enhance result of evolutionary algorithms on such problems. In Section 4, we present the experimental design of our comparative study, and Section 5 presents a discussion on the results obtained. Finally, Section 6 presents our conclusion and future work.

2 Difficult PS Topology

Solving a MOP consists in maximizing or minimizing simultaneously m objective functions subject to constrains and bounds of a set of n decision variables. Often, there is no single solution to these problems, instead a set of optimal solutions that captures the trade-offs between solutions are demanded. This set is called Pareto set (PS) in decision space, and Pareto front (PF) in objective space.

Pareto set topology refers to the geometry created by optimal solutions of a multi-objective problem in the decision space. According to the Karush-Kuhn-Tucker condition, it can be induced under certain assumptions that the PS of a continuous MOP defines a piecewise continuous $(m-1)$ -dimensional manifold in the decision space [7]. In such case, the PS would be a piecewise continuous one-dimensional curve in \mathbb{R}^n for bi-objective optimization problems, a two-dimensional curve for three-objective problems, and so on. Considering this property of continuous MOPs, Okabe et al. [11] observed that PS topologies were oversimplified for most artificial problems, arguing that we should not expect such simplification on real-world problems. For example, in DTLZ2 [6] the optimal solutions lies on the interval between 0 and 1 for the variables related to diversity, and 0.5 for all other variables related to convergence. Therefore, it might be simple to find a well-spread set of solutions in objective space on this problem since after finding one optimal point, changing only one variable would create another optimal solution. On the other hand, a more difficult case would demand a change in multiple decision variables. Such curves have received distinct denominations on different studies such as complicated PS shapes [7] and difficult PS topology [14]. In this paper, we refer to it as difficult PS topology.

Evolutionary algorithms are a powerful tool to find good solutions in multi-objective problems, but they lack proper distribution of solutions sometimes, particularly in problems with difficult topology [7, 8]. Figure 2 illustrates how MOEAs can guide solutions towards optimum values by mixing them with evolutionary operators, but evolution gets stuck at some point before finding a

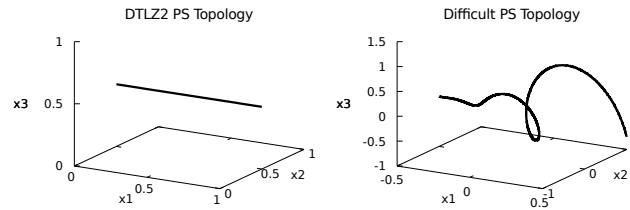


Fig. 1: DTLZ2 PS topology (left) compared to difficult PS topology (right).

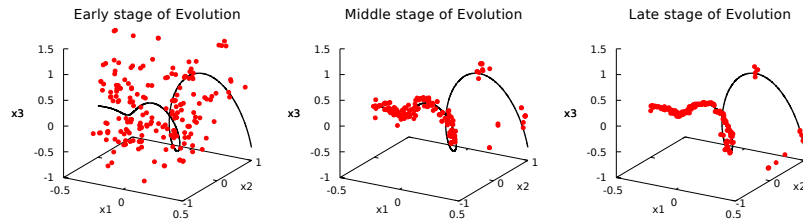


Fig. 2: Example of solutions found by NSGA-II (in red) and the PS (in black).

good representation of the PS. In this case, the decision maker would have fewer options to choose from in decision space, and most likely some trade-offs in objective space will be missing.

3 Interpolation of PS Topology

In this work, we propose a model that builds polynomial interpolations of the decision variables from the data contained in the population and use these polynomials to generate new candidate solutions to update the population. If the interpolation is close to the true PS topology, we can distribute solutions across decision space hoping to produce non-dominated solutions in objective space and give more options to the decision maker in both spaces. In the following, we describe the proposed method in detail.

For bi-objective optimization problems, decision space topology is a one-dimensional curve presented in hyper-dimensional space as illustrated in Figure 2. Thus, we use parametric Catmull-Rom curves [2] to perform the interpolation of decision space. Catmull-Rom is a family of cubic interpolating splines formulated such that the tangent at each point is calculated using the previous and next point on the spline. Usually, these curves assume a uniform parameter spacing, but Euclidean distance can also be used as the parametrization space [2]. These curves are smooth polynomial representations passing through all control points with local support, so that each point only affects a small neighborhood on the curve. Let $\mathbf{P}_i \in \mathbb{R}^n$ be the control points of a Catmull-Rom curve, $i = 1, 2, \dots, n_{cp}$, and t_i its associated parametric value. A Catmull-Rom curve is composed of $n_{cp} - 1$ polynomial segments between consecutive control points.

Let $\mathbf{Q}_{i,i+1}$ be the polynomial interpolation between control points \mathbf{P}_i and \mathbf{P}_{i+1} , associated to parameters t_i and t_{i+1} . The polynomial segment $\mathbf{Q}_{i,i+1}$ is influenced by both adjacent control points \mathbf{P}_{i-1} and \mathbf{P}_{i+2} . Note that for extreme segments $\mathbf{Q}_{1,2}$ and $\mathbf{Q}_{n_{cp}-1,n_{cp}}$, there is no \mathbf{P}_0 and $\mathbf{P}_{n_{cp}+1}$, so we define them as $\mathbf{P}_1 - 0.5(\mathbf{P}_2 - \mathbf{P}_1)$ and $\mathbf{P}_{n_{cp}} - 0.5(\mathbf{P}_{n_{cp}} - \mathbf{P}_{n_{cp}-1})$, respectively.

The $\mathbf{Q}_{i,i+1}$ segment is defined by:

$$\mathbf{Q}_{i,i+1} = \frac{t_{i+1} - t}{t_{i+1} - t_i} \mathbf{L}_{012} + \frac{t - t_i}{t_{i+1} - t_i} \mathbf{L}_{123} \quad (1)$$

where:

$$\begin{aligned} \mathbf{L}_{012} &= \frac{t_{i+1} - t}{t_{i+1} - t_{i-1}} \mathbf{L}_{01} + \frac{t - t_{i-1}}{t_{i+1} - t_{i-1}} \mathbf{L}_{12}, & \mathbf{L}_{123} &= \frac{t_{i+2} - t}{t_{i+2} - t_i} \mathbf{L}_{12} + \frac{t - t_i}{t_{i+2} - t_i} \mathbf{L}_{23}, \\ \mathbf{L}_{01} &= \frac{t_i - t}{t_i - t_{i-1}} \mathbf{P}_{i-1} + \frac{t - t_{i-1}}{t_i - t_{i-1}} \mathbf{P}_i, & \mathbf{L}_{12} &= \frac{t_{i+1} - t}{t_{i+1} - t_i} \mathbf{P}_i + \frac{t - t_i}{t_{i+1} - t_i} \mathbf{P}_{i+1}, \\ \mathbf{L}_{23} &= \frac{t_{i+2} - t}{t_{i+2} - t_{i+1}} \mathbf{P}_{i+1} + \frac{t - t_{i+1}}{t_{i+2} - t_{i+1}} \mathbf{P}_{i+2} \end{aligned}$$

For Catmull-Rom curves, it is common to define the parametrization from its geometric embedding in Euclidean space. Therefore, we can define t_{i+1} as the Euclidean distance between consecutive control points by:

$$t_{i+1} = |\mathbf{P}_{i+1} - \mathbf{P}_i|^\alpha + t_i \quad (2)$$

Here, centripetal parametrization ($\alpha = 0.5$) has been chosen since it guarantees no intersections within curve segments [13].

Figure 3 illustrates the parametrization of decision space, where red dots are examples of control points. In this case, control points are in fact solutions found during the evolutionary process. Therefore, it is possible to describe the PS topology with Catmull-Rom method if solutions are good control points, i.e. well converged in some regions. However, this method requires a proper ordering of solutions in decision space.

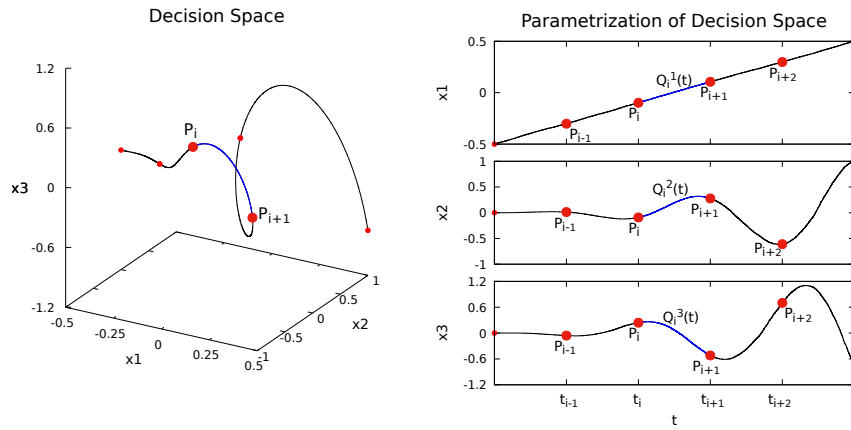


Fig. 3: Illustration of a hypothetical PS and its Catmull-Rom parametric curves.

The clustering method k-means is used to sample control points from the population of the evolutionary algorithm. Rather than using all solutions, it is reasonable to select few of them since it can be redundant to do interpolation between too close points. By applying k-means in objective space, solutions can be clustered in groups, from which we can get their centroids as control points for the interpolation. Figure 4 presents an illustration of solutions being divided in objective space, and their respective values in decision space.

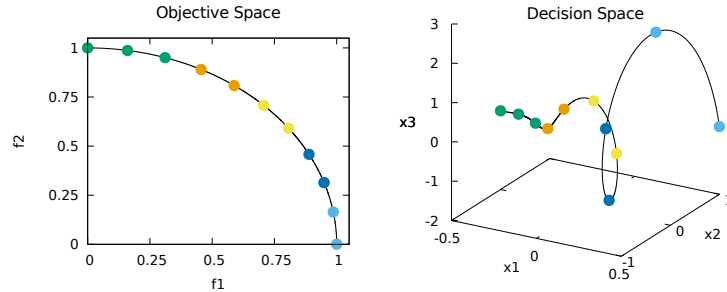


Fig. 4: k-means can distinguish solutions in different groups.

There are several ways in which the proposed method can be used within the evolutionary algorithm. One is to use it as a restart mechanism, where the solutions generated from the interpolation polynomials replace all solutions in the population. Another way is to allow competition between solutions in the population with those generated using the polynomials. To test our proposed method, in this work we use it to perform restarts during the evolutionary search.

The pseudocode of the proposed method is as follows:

Step 1. Sample and ordering

- 1.1. Apply k-means to distribute solutions in n_{cp} different clusters in objective space.
- 1.2. Order clusters according to one of the objective values. Step 2 will follow this order to perform interpolation.
- 1.3. Compute the centroids of the clusters, i.e. the average value of each variable among all solutions in the clusters. These n_{cp} centroids are used as control points in decision space $\mathbf{P}_1, \mathbf{P}_2, \dots, \mathbf{P}_{n_{cp}}$.

Step 2. Interpolation and generation of new solutions

- 2.1. Create Catmull-Rom spline using the control points and order defined in step 1. We obtain one polynomial per variable per segment, i.e. $x_k = \mathbf{Q}_{i,i+1}^k(t), k = 1 \dots, n$.
- 2.2. Repeat steps 2.3 to 2.5 for all interpolation segments $i = 1, \dots, (n_{cp} - 1)$
- 2.3. Calculate the fraction d_i of the Euclidean distance between two consecutive control points $(\mathbf{P}_i, \mathbf{P}_{i+1})$ and the sum of distances of all consecutive control points.

$$d_i = \frac{dist(\mathbf{P}_i, \mathbf{P}_{i+1})}{\sum_{k=1}^{n_{cp}-1} dist(\mathbf{P}_k, \mathbf{P}_{k+1})} \quad (3)$$

- 2.4. Define the number of solutions \bar{N}_i to be generated in the i th interval as $\bar{N}_i = \text{round}(d_i \times N)$, where N is the population size.
- 2.5. Generate new solutions based on Catmull-Rom spline. Here, calculate \bar{N}_i values of the parameter $t \in [t_i, t_{i+1}]$ by

$$t^{(j)} = t_i + \frac{t_{i+1} - t_i}{\bar{N}_i} j, \quad j = 0, 1, \dots, \bar{N}_i \quad (4)$$

and use the corresponding polynomials for all variables to generate \bar{N}_i solutions, one for each value $t^{(j)}$. Thus, solution $\mathbf{x}_j = (x_1, \dots, x_k, \dots, x_n)$ is generated from parameter value $t^{(j)}$ estimating the value of variables by $x_k = \mathbf{Q}_{i,i+1}^k(t^{(j)})$, $k = 1 \dots, n$.

Step 3. Verification

- 3.1. Discard solutions off boundaries.
- 3.2. Check if the number of new solutions is equal to population size. If there are fewer solutions, we randomly include individuals from the current population. In case we have more solutions, we randomly throw away individuals so we can have same population size in the restart.

4 Experimental Setup

In total, five bi-objective unimodal CEC09 competition problems [16] were used, namely UF1, UF2, UF3, UF4, and UF7, setting number of variables $\mathbf{n} = 30$.

To test the proposed method with different evolutionary methodologies following the CEC09 competition parameters setting, we implement our model in three algorithms: NSGA-II [5], MOEA/D [7], and an improved version of MOEA/D to solve CEC 2009 competition problems denominated MOEA/D-DRA [15]. Differential Evolution (DE) crossover operator and polynomial mutation were used, since it produces better results than SBX operator [7]. Crossover rate is $\mathbf{pc} = 1.0$, and mutation rate per variable is $\mathbf{pm} = 1/\mathbf{n}$. DE operator parameter is set to $\mathbf{F} = 0.5$, and the distribution exponent of polynomial mutation is set to $\eta_m = 20$. All algorithms perform a total number of function evaluations equals to **300000** with population size $\mathbf{N} = 600$. For MOEA/D, Tchebycheff approach and neighborhood size of $\mathbf{T} = 60$ were used in both versions. Here, we tested different numbers of control points $n_{cp} = \mathbf{150}, \mathbf{300}, \mathbf{500}$, and **restarts** = **2, 5, 10, 20**. Restarts are equally spaced in generations. We run each algorithm 30 times using the same set of seeds. Finally, IGD [3] metric was used to compare results.

5 Experimental Results and Discussion

To compare the original evolutionary algorithm against the version coupled with the model, Table 1 presents IGD results for all tested problems. To illustrate, we present results for the model with 10 restarts and 300 samples of control points to build the interpolations which overall produce good results. For each

		NSGA-II		MOEA/D		MOEA/D-DRA	
		Original	Model	Original	Model	Original	Model
UF1	Average	0.018806	0.012445	0.001479	0.001171	0.001427	0.001117
	SD	0.001136	0.003395	0.025630	0.020608	0.000324	0.000078
	p-value	3.561984e-13		0.958135		0.000141	
UF2	Average	0.015440	0.009106	0.005799	0.007126	0.003703	0.003718
	SD	0.001315	0.001615	0.020539	0.003534	0.001798	0.001606
	p-value	4.992806e-25		0.120524		0.601648	
UF3	Average	0.145545	0.082388	0.009209	0.007392	0.006490	0.005320
	SD	0.011775	0.007450	0.011240	0.006558	0.008079	0.005899
	p-value	1.340732e-33		0.215363		0.342249	
UF4	Average	0.039056	0.038930	0.064486	0.064192	0.059472	0.060421
	SD	0.001765	0.001742	0.000028	0.000015	0.004593	0.004190
	p-value	0.548976		7.984198e-36		0.730446	
UF7	Average	0.008453	0.007527	0.003074	0.003415	0.001409	0.001250
	SD	0.001826	0.002527	0.159385	0.139367	0.000314	0.000165
	p-value	0.001213		0.600833		0.019249	

Table 1: Results of IGD for 300 control points and 10 restarts.

problem, we present the average and standard deviation values of IGD among all 30 runs, together with the p-value of t-tests on the IGD sets obtained with the original algorithm and its improved version. A p-value smaller than 0.05 indicates with 95% confidence that the averages are statistically different. Statistically better IGD averages are shown in bold, i.e. smaller average IGD value and p-value smaller than 0.05. According to these results, the proposed method could find approximations with better IGD for some problems. Besides UF4, IGD values improved for all problems when using NSGA-II. In case of the decomposition algorithm, the model could improve IGD results for problem UF4 when using MOEA/D, and problems UF1 and UF7 when using its improved version MOEA/D-DRA. Note that the model did not deteriorate results in any case.

Since the proposed model is built based on solutions found by the algorithms, whether or not the model can improve results by performing interpolation would depend on the distribution of algorithm’s population. In other words, if the population is a good representation of the PS topology, the model can be effective by creating solutions from the interpolation. In this case, a good representation would include solutions in the inflection points of the curve in decision space, where the topology changes concavity. Thus, by looking at where solutions are placed through generations, we can understand whether the interpolation would properly approximate the PS topology.

Figure 5 and 6 present solutions found by NSGA-II and MOEA/D in distinct generations, presented in different colors. While NSGA-II’s population steadily approaches the optimal front with large coverage as the evolution proceeds, MOEA/D converges solutions very fast in some regions of objective space at first, and distribute solutions on those regions’ neighborhood. These different

approaches have an effect on decision space, where the Pareto-dominance based algorithm seems to produce solutions better suitable to the proposed model. As NSGA-II's solutions are placed on the inflection points of the PS topology since early generations, it offers to the learning model a better representation of the topology. In contrast, the decomposition algorithm finds at first solutions in fewer regions far from the inflection points, misleading the learning model. For example, the interpolation on solutions from generation 50 by MOEA/D would produce something close to a straight line in decision space.

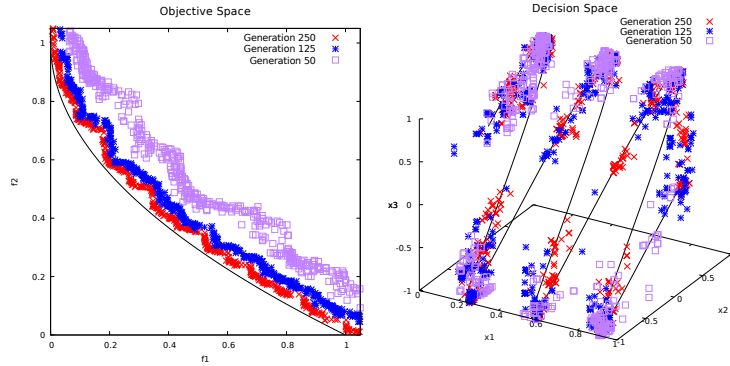


Fig. 5: Behavior of solutions found by NSGA-II (without model) on UF1.

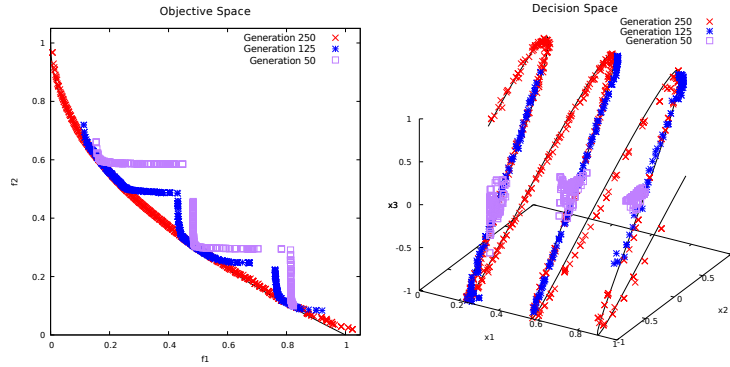


Fig. 6: Behavior of solutions found by MOEA/D (without model) on UF1.

Figure 7 presents IGD values by NSGA-II and MOEA/D-DRA on problem UF1 with different sampling size and number of restarts. Note that IGD steadily improves as the number of restarts increases. Also, note that a sample of control points half of the population size (300) gives overall good results.

Results for NSGA-II have improved significantly according to IGD metric. Figure 8 and 9 illustrates for UF1 and UF3 problems all non-dominated solutions found by the original algorithm and the one coupled with the model in their best IGD run. Both figures show that distribution of solutions have improved in both

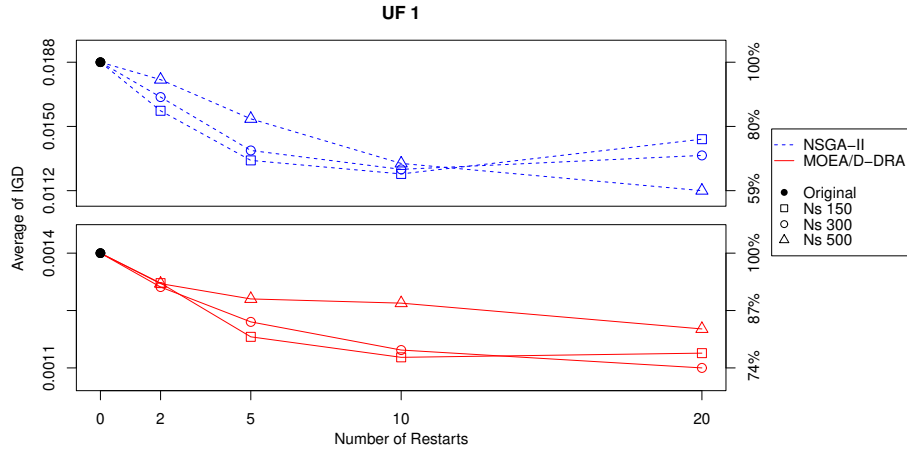


Fig. 7: Average of IGD for 30 runs including all non-dominated solutions found during the search.

spaces when using the proposed method. Note that for UF3 the improvement in convergence is more significant. These figures illustrate well the main idea behind our learning method, where interpolation takes advantage that NSGA-II can provide a set of control points to build a good interpolation to generate new candidate solutions and improve quality of results.

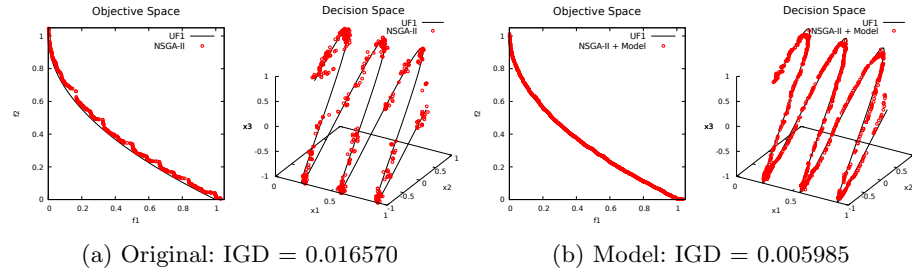


Fig. 8: Solutions found by NSGA-II from the best IGD run on problem UF1.

In contrast to previous problems, Figure 10 shows that restarts could not improve results for problem UF4. In this case, both NSGA-II and MOEA/D fail to converge solutions close to the optimal Pareto set, so when our method tries to perform interpolation based on EA’s population, it fails to represent PS topology. Therefore, this problem shows that the proposed method with restarts may not improve algorithm’s performance when the population is far from the Pareto set.

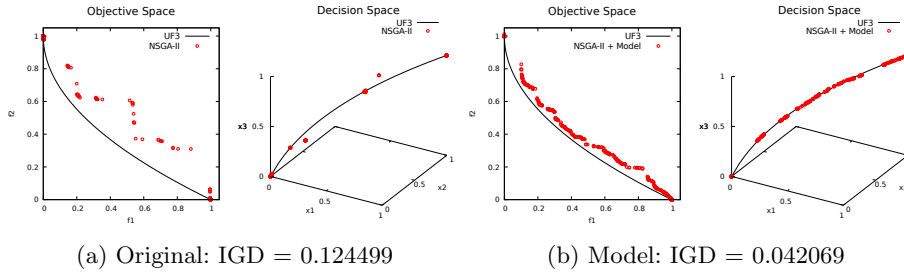


Fig. 9: Solutions found by NSGA-II from the best IGD run on problem UF3.

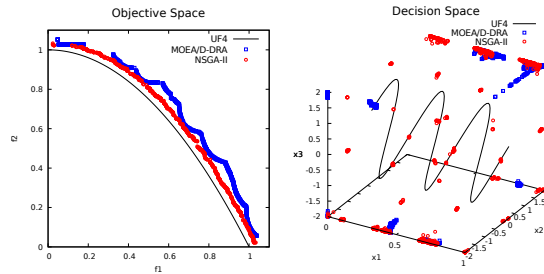


Fig. 10: Solutions found by MOEA/D-DRA (blue) and NSGA-II (red) on UF4.

6 Conclusion

In this paper, we presented a method that learns a model of the solutions' topology in the population by performing parametric spline interpolation for all variables in decision space. Here, Catmull-Rom parametric curves were used to perform interpolation, which allow us to deal with any dimension in decision space, but limited to bi-objective problems. We coupled the model with NSGA-II and two version of MOEA/D to perform restarts from solutions generated by the model. We showed that the proposed model could improve distribution and convergence of solutions for most problems in the case of NSGA-II, and for some problems in the case of MOEA/D-DRA. Also, we showed that the effectiveness of the interpolation depends on the behavior of the algorithm.

In the future, we would like to study other methods to perform interpolation to solve problems with more objectives. Also, we want to study other ways to couple this model with evolutionary algorithms. For instance, allowing competition of solutions created by the model with the current population, instead of totally replacing the population. Another aspect that we would like to investigate is the scalability of the model in decision space.

References

1. Aguirre, H., Yazawa, Y., Oyama, A., Tanaka, K.: Extending aeseh from many-objective to multi-objective optimization. In: Proceedings of 10th International

- Conference on Simulated Evolution and Learning. pp. 239–250 (Dec 2014)
2. Catmull, E., Rom, R.: A class of local interpolating splines. In: Barnhill, R.E., Riesenfeld, R.F. (eds.) *Computer Aided Geometric Design*, pp. 317 – 326. Academic Press (1974)
 3. Coello, C.A.C., Lamont, G.B., Veldhuizen, D.A.V.: *Evolutionary Algorithms for Solving Multi-Objective Problems (Genetic and Evolutionary Computation)*. Springer-Verlag New York, Inc., Secaucus, NJ, USA (2006)
 4. Corne, D.W., Knowles, J.D., Oates, M.J.: The pareto envelope-based selection algorithm for multiobjective optimization. In: *Parallel Problem Solving from Nature PPSN VI*. pp. 839–848. Springer Berlin Heidelberg, Berlin, Heidelberg (2000)
 5. Deb, K., Pratap, A., Agarwal, S., Meyarivan, T.: A fast and elitist multiobjective genetic algorithm: Nsga-ii. *IEEE Transactions on Evolutionary Computation* **6**(2), 182–197 (Apr 2002)
 6. Deb, K., Thiele, L., Laumanns, M., Zitzler, E.: Scalable multi-objective optimization test problems. In: *Congress on Evolutionary Computation (CEC 2002)*. pp. 825–830. IEEE Press (2002)
 7. Li, H., Zhang, Q.: Multiobjective optimization problems with complicated pareto sets, moea/d and nsga-ii. *IEEE Transactions on Evolutionary Computation* **13**(2), 284–302 (April 2009)
 8. Marca, Y., Aguirre, H., Zapotecas, S., Liefoghe, A., Derbel, B., Verel, S., Tanaka, K.: Pareto dominance-based moeas on problems with difficult pareto set topologies. In: *Proceedings of the Genetic and Evolutionary Computation Conference Companion*. pp. 189–190. GECCO '18, ACM, New York, NY, USA (2018)
 9. Mo, L., Dai, G., Zhu, J.: The rm-meda based on elitist strategy. In: *Advances in Computation and Intelligence*. pp. 229–239. Springer Berlin Heidelberg, Berlin, Heidelberg (2010)
 10. Morgan, D., Waldock, A., Corne, D.: Mopc/d: A new probability collectives algorithm for multiobjective optimisation. In: *2013 IEEE Symposium on Computational Intelligence in Multi-Criteria Decision-Making (MCDM)*. pp. 17–24 (2013)
 11. Okabe, T., Jin, Y., Olhofer, M., Sendhoff, B.: On test functions for evolutionary multi-objective optimization. In: *Parallel Problem Solving from Nature - PPSN VIII*. pp. 792–802. Springer Berlin Heidelberg (2004)
 12. Schütze, O., Mostaghim, S., Dellnitz, M., Teich, J.: Covering pareto sets by multilevel evolutionary subdivision techniques. In: *Evolutionary Multi-Criterion Optimization*. pp. 118–132. Springer Berlin Heidelberg, Berlin, Heidelberg (2003)
 13. Yuksel, C., Schaefer, S., Keyser, J.: Parameterization and applications of catmull-rom curves. *Comput. Aided Des.* **43**(7), 747–755 (Jul 2011)
 14. Zapotecas-Martinez, S., Coello, C.A.C., Aguirre, H.E., Tanaka, K.: A review of features and limitations of existing scalable multi-objective test suites. *IEEE Transactions on Evolutionary Computation* (2018)
 15. Zhang, Q., Liu, W., Li, H.: The performance of a new version of moea/d on cec09 unconstrained mop test instances. In: *2009 IEEE Congress on Evolutionary Computation*. pp. 203–208 (May 2009). <https://doi.org/10.1109/CEC.2009.4982949>
 16. Zhang, Q., Zhou, A., Zhao, S., Suganthan, P.N., Liu, W., , Tiwari, S.: Multiobjective optimization test instances for the cec 2009 special session and competition. Tech. rep., University of Essex and Nanyang Technological University (2008)
 17. Zitzler, E., Künzli, S.: Indicator-based selection in multiobjective search. In: *Parallel Problem Solving from Nature - PPSN VIII*. pp. 832–842. Springer Berlin Heidelberg (2004)
 18. Zitzler, E., Laumanns, M., Thiele, L.: SPEA2: Improving the strength pareto evolutionary algorithm. Tech. Rep. TIK-Report 103, ETH Zurich, Switzerland (2001)