Author(s): Mazumdar, Atanu; Chugh, Tinkle; Hakanen, Jussi; Miettinen, Kaisa

Title: An Interactive Framework for Offline Data-Driven Multiobjective Optimization

Year: 2020

Version: Accepted version (Final draft)

Please cite the original version:

Mazumdar, A., Chugh, T., Hakanen, J., & Miettinen, K. (2020). An Interactive Framework for Offline Data-Driven Multiobjective Optimization. In B. Filipic, E. Minisci, & M. Vasilei (Eds.), BIOMA 2020 : 9th International Conference on Bioinspired Optimization Methods and Their Applications, Proceedings (pp. 97-109). Springer. Lecture Notes in Computer Science, 12438. https://doi.org/10.1007/978-3-030-63710-1_8

An Interactive Framework for Offline Data-Driven Multiobjective Optimization

Atanu Mazumdar[1[0000−0002−7760−3171]], Tinkle Chugh[2[0000−0001−5123−8148]], Jussi Hakanen[1[0000−0001−9579−8657]], and Kaisa Miettinen[1[0000−0003−1013−4689]]

[1] University of Jyvaskyla, Faculty of Information Technology, P.O. Box 35 (Agora), FI-40014 University of Jyvaskyla, Finland
atanu.a.mazumdar@student.jyu.fi,{jussi.hakanen,kaisa.miettinen}@jyu.fi
[2] Department of Computer Science, University of Exeter, UK
t.chugh@exeter.ac.uk

Abstract. We propose a framework for solving offline data-driven multiobjective optimization problems in an interactive manner. No new data becomes available when solving offline problems. We fit surrogate models to the data to enable optimization, which introduces uncertainty. The framework incorporates preference information from a decision maker in two aspects to direct the solution process. Firstly, the decision maker can guide the optimization by providing preferences for objectives. Secondly, the framework features a novel technique for the decision maker to also express preferences related to maximum acceptable uncertainty in the solutions as preferred ranges of uncertainty. In this way, the decision maker can understand what uncertainty in solutions means and utilize this information for better decision making. We aim at keeping the cognitive load on the decision maker low and propose an interactive visualization that enables the decision maker to make decisions based on uncertainty. The interactive framework utilizes decomposition-based multiobjective evolutionary algorithms and can be extended to handle different types of preferences for objectives. Finally, we demonstrate the framework by solving a practical optimization problem with ten objectives.

Keywords: Decision support · Decision making · Decomposition-based MOEA · Metamodelling · Surrogate · Kriging · Gaussian processes.

1 Introduction

Sometimes while solving data-driven multiobjective optimization problems (or MOPs) additional data can not be acquired during the solution process. Instead, we may have pre-collected data of the phenomenon of interest that was obtained beforehand, e.g. by conducting physical experiments. This type of optimization problems are termed as *offline* data-driven MOPs [3, 8, 17]. For formulating the optimization problem, we can build surrogate models using the given data to approximate the behaviour of the phenomenon. Optimization can then be performed utilizing these surrogates as objective functions e.g. by a multiobjective evolutionary algorithm (MOEA). However, approximation error in the surrogates' prediction can not be avoided. Certain surrogate models such as Kriging also provide information about the uncertainty (e.g. as standard deviation) in predictions. This uncertainty information can be utilized in the optimization process to improve the quality of the solutions [11].

Previous works on offline multiobjective optimization such as [3, 8, 11, 17] approximate the entire Pareto front. This makes decision making a difficult task as the decision maker (DM) has to choose from a large set of solutions. Interactive multiobjective optimization approaches allow the DM to find solutions in an interesting region of the Pareto front and learn about the problem and the feasibility of one's preferences and adjust the latter. They also provide limited amount of information at a time thereby reducing the cognitive load (see [13] for more information). There have been many developments in interactive MOEAs [14] and decomposition based MOEAs have become quite popular because of their capability of solving MOPs with a large number of objectives [2, 4, 20]. Hence, interactive approaches such as [7, 10, 21] have been proposed for decomposition-based MOEAs. However, as far as we know, addressing DM's preferences while solving offline MOPs in decomposition-based MOEAs has not been considered.

Utilizing the uncertainty information in interactive optimization may be quite valuable to the DM for a better understanding of the solutions and better decision making while solving offline MOPs. The major challenge in utilizing uncertainty in an interactive optimization process is conveying this extra information to the DM as (s)he may not be familiar with it.

In this paper, we propose a framework for solving offline data-driven MOPs interactively using decomposition-based MOEAs. It enables the DM to understand and make decisions based on the uncertainties present in the approximated solutions. The framework does not increase the cognitive load of the DM significantly while providing preference information for uncertainties along with the preferences for objectives.

2 Background

We consider the underlying MOP that has to be solved of the following form:

$$\begin{aligned} &\text{minimize } \{f_1(\mathbf{x}), \dots, f_K(\mathbf{x})\}, \\ &\text{subject to } \mathbf{x} \in S, \end{aligned} \tag{1}$$

where $K \geq 2$ is the number of objectives and S is the feasible region in the decision space $\mathbb{R}^n$. For a feasible decision vector $\mathbf{x}$, the corresponding objective vector $\mathbf{f}(\mathbf{x})$ comprises of the underlying objective (function) values $(f_1(\mathbf{x}), \dots, f_K(\mathbf{x}))$.

A solution $\mathbf{x}^1 \in S$ dominates another solution $\mathbf{x}^2 \in S$ if $f_k(\mathbf{x}^1) \leq f_k(\mathbf{x}^2)$ for all $k = 1, \dots, K$ and $f_k(\mathbf{x}^1) < f_k(\mathbf{x}^2)$ for at least one $k = 1, \dots, K$. If a solution of an MOP is not dominated by any other feasible solutions, it is called nondominated. Solving an MOP using an MOEA typically produces solutions that are nondominated within the set of solutions it has found. The solutions of Eq. (1) that are nondominated in S are also called Pareto optimal solutions. Next, we discuss a generic approach to solve an offline data-driven MOP.

2.1 Generic Approach for Offline Data-Driven Multiobjective Optimization

A generic way for offline data-driven optimization using an MOEA described in [8, 18] is shown in Fig. 1. The solution process can be divided into three parts: a) data collection, b) formulating the MOP and building surrogate models, and c) running an MOEA. The first step involves performing experiments to acquire the data and pre-processing it if necessary. Next, surrogate models are built to approximate the behaviour of the underlying objective functions using the provided data. The prediction vector of the fitted surrogate models can be represented as $\hat{\mathbf{f}}(\mathbf{x}) = (\hat{f}_1(\mathbf{x}), \ldots, \hat{f}_K(\mathbf{x}))$, where $\hat{f}_k$ is the surrogate's prediction for f_k. Surrogate models such as Kriging also provide the uncertainty in the model's prediction generally in the form of standard deviation. The predicted uncertainty vector is represented as $\hat{\boldsymbol{\sigma}}(\mathbf{x}) = (\hat{\sigma}_1(\mathbf{x}), \ldots, \hat{\sigma}_K(\mathbf{x}))$, where $\hat{\sigma}_k$ is the uncertainty in prediction for the k^{th} objective function. In the third step, an MOEA is run to solve the optimization problem with the surrogates as objective functions.

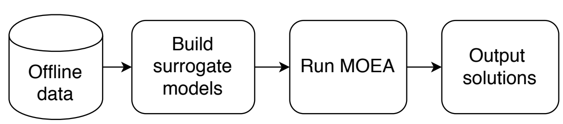

Fig. 1. A generic approach for offline data-driven multiobjective optimization.

Next, we briefly discuss an interactive approach for decomposition-based MOEAs which is a building block of the framework proposed in this paper.

2.2 Interactive Decomposition-Based MOEA

Decomposition-based MOEAs use reference (or weight) vectors to decompose the objective space into a number of sub-spaces. In general, they solve several simpler sub-problems that represent an aggregate of the objective functions by using a scalarizing function. Some examples of the scalarizing functions used are Chebyshev [20], penalty based boundary intersection distance (PBI) [20] and angle penalized distance (APD) [2]. The solutions obtained by solving these sub-problems jointly represent the approximated Pareto front of the MOP in the objective space.

Interactive decomposition-based MOEAs find solutions only in certain regions of the Pareto front. These approaches utilize preference information from the DM in the form of, e.g. a reference point, weights and preferred ranges for objectives. For more details, see, e.g. [14, 19]. In this paper, we adopt the interactive approach proposed in [7] for decomposition-based MOEAs and briefly describe its main ideas as follows.

Converting Preference Information to Reference Vectors: One of the ways to incorporate preference information into decomposition-based MOEAs is by adapting the reference vectors to follow the DM's preferences [14]. We here demonstrate how to utilize a reference point which consists of the DM's desired value for each objective. However, the framework proposed later in this paper is not limited to only this type of preference information.

Consider a set of uniformly distributed reference vectors $V = \{\mathbf{v}^i \in \mathbb{R}^k | i = 1, \dots, m\}$, where m is the total number of reference vectors, and $\bar{\mathbf{z}} \in \mathbb{R}^k$ is a single reference point provided by the DM. Each reference vector can be adapted as follows [2, 7]:

$$\overline{\mathbf{v}^i} = \frac{r \cdot \mathbf{v}^i + (1 - r) \cdot \mathbf{v}^c}{\| r \cdot \mathbf{v}^i + (1 - r) \cdot \mathbf{v}^c \|}, \tag{2}$$

where $\mathbf{v}^c = \bar{\mathbf{z}} / \|\bar{\mathbf{z}}\|$ and $r \in (0, 1)$. The central vector $\mathbf{v}^c$ is the projection of $\bar{\mathbf{z}}$ on a unit hypersphere and the spread of the adapted reference vectors is determined by the parameter r. The adapted reference vectors are close to $\mathbf{v}^c$ if r is close to zero and if r is close to one, the reference vectors are not changed much.

3 The Proposed Framework

As mentioned, since no new data is available in offline data-driven optimization, the approximation accuracy of the surrogate models determines the quality of solutions. In reality, the surrogate models' approximation involve uncertainty. As mentioned, Kriging surrogates [6] also provide an estimate of the uncertainty in its prediction. A solution with a higher uncertainty indicates that the objective values predicted by the surrogates have a lower probability of being close to the values of the underlying objective function. In other words, the uncertainty predicted by the surrogate models can represent the accuracy of the solutions when evaluated using the underlying objective functions. In [11], utilizing the predicted uncertainties from the surrogates as additional objective(s) produced solutions with a better hypervolume and accuracy in root mean squared error (RMSE) compared to the generic approach. This was because the approach simultaneously minimized the objective functions and their respective uncertainties. The solutions generated represented the trade-off between objective values and uncertainties. However, this results in an increase in both computational and cognitive load with a large number of objectives. Overall, it is desirable for the DM to get solutions that have a low uncertainty in order to achieve better accuracy.

As explained before, interactive approaches are quite advantageous as the DM can guide the optimization process through preferences for *objectives* and also learn about the problem. To incorporate preferences for *uncertainties* while solving an MOP interactively, the DM should first understand what uncertainty really means in regards to the MOP. Giving the DM an opportunity to provide preferences for uncertainties is desirable but may increase cognitive load.

The proposed framework aims at solving offline data-driven MOPs interactively by considering preferences for both objectives and uncertainties. The framework is based on a decomposition-based MOEA and preference information for objectives in the form of reference points. The first and primary challenge faced is the DM's understanding of uncertainty, specifically the uncertainty in the surrogates' approximation. Secondly, the cognitive load should not drastically increase when the DM wants to provide preferences for uncertainties along with the preferences for objectives. The proposed framework tackles both of the challenges and aims at providing an improved decision support for the DM during the solution process. Next, we discuss two steps which are the primary building blocks of the proposed framework.

3.1 Pre-Filtering Solutions Following DM's Preferences

Generally, in offline data-driven MOPs, there exists a trade-off between the quality of solutions (e.g. hypervolume) and the accuracy of the solutions (e.g. RMSE) [11]. To have a diverse range of uncertainty and objective values, we first store the solutions from all the generations of an MOEA in an archive. This allows us to filter and make decisions from a pool of solutions having various objective and uncertainty values. However, only the solutions representing the DM's preferences for objectives are interesting to him/her. Hence, the archive needs some amount of pre-filtering before we can present it to the DM. We have to further filter these solutions such that only the solutions that simultaneously achieve the best objective values and the lowest uncertainties are shown to the DM. Hence, we propose a two-stage pre-filtering approach as follows.

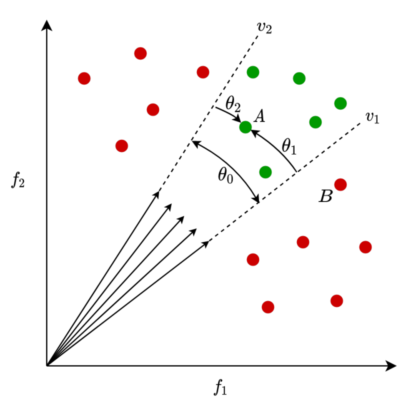

Fig. 2. Pre-filtering solutions: Green dots are kept and red dots are rejected.

The first stage is to find solutions in the archive that follow the DM's preferences for objectives, i.e., reference points. As described in Section 2.2, at first, the uniformly distributed set of reference vectors are adapted using Eq. (2) that reflects the DM's preferences for objectives. Next, we find the adapted reference vectors that have the highest component in one of the objectives and call them *edge vectors*. Initially, the set of reference vectors are uniformly distributed and have just one vector at each axis (objective). As the adaptation in Eq. (2) is linear; we find just one reference at every extreme or edge. Thus, the total number of edge vectors is K. The multidimensional volume enclosed by the edge vectors is termed as the *hypercone*. A solution is accepted by the first stage pre-filter if it lies inside the hypercone. Fig. 2 shows the idea of the pre-filtering for a bi-objective minimization problem. The edge vectors are v_1 and v_2 and the angle between the edge vectors is θ_0. The angle between solution A and the edge vectors v_1 and v_2 is θ_1 and θ_2, respectively. A solution is accepted for the next pre-filtering stage if both θ_1 and θ_2 are smaller than θ_0. In the figure, the solutions in green (e.g. A) are accepted by this pre-filtering stage, and the solutions shown in red (e.g. B) are rejected. The rejected solutions do not follow the preferences and hence are not of interest to the DM. In general, with K objectives, the angle θ_0 between any two edge vectors is the same. This is because the set of uniformly distributed reference vectors is adapted by using a linear transformation. Hence, a solution is inside the hypercone if $\theta_k^i < \theta_0$ for all $k = 1, \ldots, K$, where θ_k^i is the angle between the kth edge vector and the ith solution.

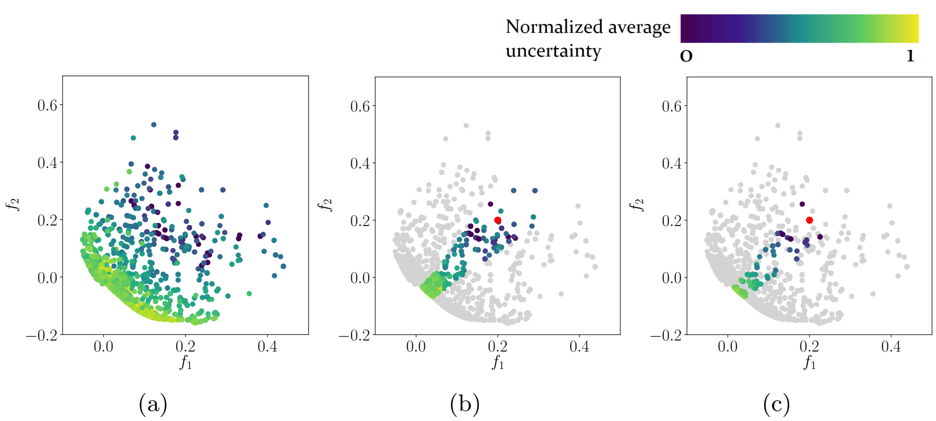

Fig. 3. The sub-figures show the solutions in different pre-filtering stages while solving a bi-objective minimization problem. The grey solutions are the ones filtered out at each stage. The red point denotes the reference point provided by the DM.

The archive contains objective vectors and their respective uncertainties from all the generations. However, only the solutions with the smallest uncertainties and objective values are interesting for the DM. Hence, we propose a second pre-filtering stage that performs nondominated sorting on the solutions filtered by the first stage and include uncertainties as additional components in the vectors while sorting (as done in [11]). Considering uncertainty while performing nondominated sorting finds the solutions representing the trade-off between objective values and uncertainty.

These two stages are applied sequentially in the pre-filtering stage of our proposed framework. The functioning of the pre-filtering stage can be understood from Fig. 3, which shows solutions in the archive for a bi-objective minimization problem. The colour code represents the normalized average of the uncertainty vector for the solutions. Sub-figure (a) shows all the solutions in the archive before the pre-filtering. Sub-figure (b) shows the solutions after the first stage pre-filtering. It can be observed that only the solutions following the preferences for objectives (here the reference point in red) are filtered. Sub-figure (c) shows the solutions obtained after the second stage pre-filtering. The solutions after the pre-filtering stage follow the DM's preferences for objectives and represent the trade-off between objective values and uncertainties in the solutions. The grey solutions are the ones that are rejected at each pre-filtering stage.

3.2 DM's Understanding of Uncertainty

As discussed before, knowledge of uncertainty is an essential aspect while solving offline optimization problems. However, while solving real-life problems, the DM is not always familiar with uncertainty in the solutions. Depending on the problem, the DM can be assumed to have an idea of permissible tolerances in objective values. For example, in the welded beam problem [5], cost and end deflection are minimized. Considering just the DM's preference regarding cost, (s)he has an idea of the highest permissible cost. Here, the permissible deviation in the objective value is referred to as one-sided *tolerance* of the DM [9]. In other words, one-sided tolerance information can be considered as a cutoff over the probable variation in the objective values. In our case, the variation in objective values is available in the form of uncertainty in the surrogates. Preferred one-sided tolerances are *preferences for uncertainties* provided by the DM and represent the maximum permissible variation in the solutions when they are evaluated by the underlying objectives. In this paper, we refer to one-sided tolerance as tolerance for simplicity.

For the proposed framework (and later in the tests), we consider indifference tolerances. They are provided as a percentage for every objective and represent the 95% tolerance interval [9]. Let us consider the indifference tolerance provided by the DM for the k^{th} objective function as $\tau_k\%$, where $k = 1, \ldots, K$. The distribution of the predicted objective value is Gaussian while using Kriging surrogates and the predicted standard deviation of the k^{th} objectives' surrogate is $\hat{\sigma}_k(\mathbf{x})$. Thus, *cutoff tolerance functions* can be formulated such that the solutions do not violate the DM's preferences for uncertainties and thus are of interest to the DM. The k^{th} cutoff tolerance function is:

$$g_k(\mathbf{x}) = 1.96\hat{\sigma}_k(\mathbf{x}) - \tau_k \cdot \hat{f}_k(\mathbf{x})/100 \leq 0, \tag{3}$$

where $\mathbf{x}$ is the decision vector and $k = 1, \ldots, K$. A solution is interesting to the DM if the objective value of the k^{th} objective function does not exceed $1.96\hat{\sigma}_k(\mathbf{x})$ or 95% confidence interval of the Gaussian distribution. Thus, the DM can change the preferences for uncertainties and visualize the solutions that do not violate the cutoff tolerance functions in Eq. (3). However, it has to be noted that the cutoff tolerance function can be modified depending on the prediction distribution of the surrogate.

3.3 Steps of the Framework

Fig. 4 shows the simplified structure of the proposed interactive offline data-driven MOEA framework. The framework can be broadly divided into five steps:

1. Building surrogate models and initializing the MOEA.
2. Running the MOEA and storing the solutions in an archive.
3. Applying two-stage pre-filtering on the archive.
4. Interactively visualizing the solutions based on the preferences for uncertainties provided by the DM.
5. Asking for preference information for objectives from the DM and adapting the reference vectors.

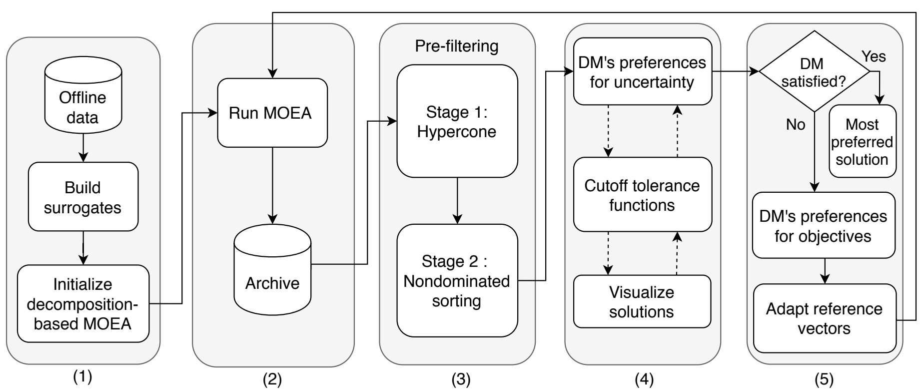

Fig. 4. The proposed framework for interactive offline data-driven multiobjective optimization.

Step 1: We formulate the MOP by utilizing the provided data. The expertise of the DM may be required in this. We build Kriging surrogate models for every objective function using the data (as in the generic approach in Section 2). Next, we initialize a decomposition-based MOEA and generate a uniformly distributed set of reference vectors and create the initial population.
Step 2: We run an MOEA for a fixed number of generations. The objective values and uncertainties for the individuals from every generation are stored in an archive that serves as a database for Step 3.

Step 3: At the end of Step 2, we have an archive containing objective vectors and uncertainties of different individuals. We apply the pre-filtering techniques as in Section 3.1. Note that for the first iteration, we do not have any preferences for objectives, and the reference vectors (that includes the edge vectors) are not adapted. Hence, the hypercone constitutes the entire objective space and the first pre-filtering stage accepts all the solutions.
Step 4: The DM provides preferences for uncertainties (indifference tolerances) $\tau_k\%$ and the pre-filtered solutions from Step 3 qualifying the cutoff tolerance functions in Eq. (3) are shown.

The DM can provide preferences for uncertainties as many times (s)he wishes thereby enabling him/her to view different solutions within the provided tolerances. For a better understanding of uncertainties while visualizing, solutions can be colour coded. This can be done by the normalized average of the uncertainty vector (in percentage) or by the maximum uncertainty of a solution for any of the objective functions. The DM may skip this step entirely if solution uncertainties are not interesting. As this step consists of just filtering solutions obtained after Step 3, it can be repeated with a very low computational cost.
Step 5: In this step, the DM can stop the optimization process if (s)he has found a satisfactory solution. Otherwise, (s)he is asked for new preference information. We adapt the reference vectors according to Eq. (2) so that solutions follow the preferences for objectives. After adapting the reference vectors, we go to Step 2.

The interaction process is split into Steps 4 and 5, where the DM provides preferences for uncertainties and objectives, respectively. Due to this, the cognitive load on the DM does not increase significantly. The DM can provide different preferences for uncertainties and view the corresponding solutions and repeat this as long as one wishes. The proposed way of providing preferences for uncertainties does not modify the selection process of the MOEA. Hence, the solution process is not affected.

4 Numerical Results

Assessing and comparing the performance of interactive approaches is still a research challenge. Hence, we demonstrate and discuss the advantages of the proposed framework by solving the general aviation aircraft (GAA) [15, 16] design problem. Due to space limitations, further analysis on benchmark problems is available at `http://www.mit.jyu.fi/optgroup/extramaterial.html` as additional material.

The GAA problem refers to designing an aircraft for recreational pilots to business executives. We solved the problem as in [15] with 27 decision variables, ten objectives and one constraint. As we are dealing with offline optimization problems, we generated data using the implementation [1]. We used Latin hypercube sampling [12] to generate 1000 samples for decision variables and evaluated them using the GAA functions to obtain the offline data. To approximate the underlying objective functions, we used Kriging with a radial basis function kernel as our surrogate models. We used RVEA as the MOEA with standard parameter settings as in [2] and executed it for 100 generations in each iteration with standard crossover and mutation parameters. The spread parameter r was set as 0.2. However, it can be increased if the DM's wants a more diverse set of solutions. As our framework does not support constraint handling, we considered the constraint violation as an additional objective function for the demonstration.

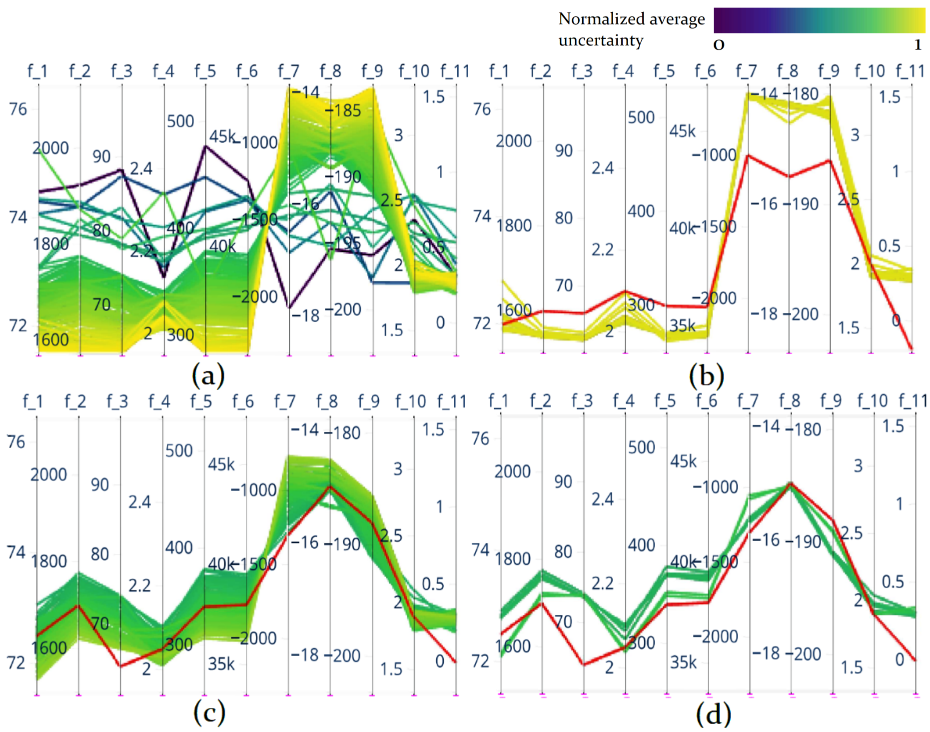

Fig. 5. The solutions obtained for two iterations of the interactive framework (all objectives are minimized). (a): solutions in the archive after the first iteration. (b) & (c): solutions after pre-filtering in the first and second iteration respectively with different reference points (red line). (d): solutions after DM provides preferences for uncertainties.

Fig. 5 shows solutions produced by the framework for two iterations. The colour coding represents the normalized average of the uncertainty vector for the solutions (blue is lowest and yellow is highest). Sub-figure (a) shows the solutions in the archive at the first iteration when there are no preferences for objectives available. In sub-figure (b) the DM provides the reference point (in red) and gets the pre-filtered solutions. It can be observed that the solutions produced follow the DM's preferences for objectives. However, (s)he chooses to skip the step of providing preferences for uncertainties as none of the solutions has a low uncertainty (as represented by the colour). In the next iteration, the DM changes the preferences for objectives. The solutions after pre-filtering, as shown in sub-figure (c) not only follow the DM's preferences for objectives but also have a lower uncertainty. We now provide hypothetical tolerances to demonstrate the framework's ability to consider preferences for uncertainties. In sub-figure (d) only a few solutions that are within the preferred uncertainty of the DM are shown. Finally, one of the solutions that matches the preferences for objectives and uncertainties may be chosen by the DM. (S)he may choose to reset the cutoff tolerances again to view a different set of solution to make decisions. Alternatively, if the DM is not satisfied with any of the solutions, (s)he may choose to change the preferences for objectives and continue the optimization.

If the DM is unaware of the uncertainties in the solutions, (s)he may be deprived of valuable knowledge regarding the acceptability of the solutions. In certain situations such as Fig. 5 (b), judging the goodness of a solution based on the objective values alone may be misleading. By observing the uncertainties, the DM avoids making a worse decision and can modify preferences for objectives. The DM may choose to provide preferences for uncertainties and see solutions within different tolerances with a low computational cost. As the DM can see the solutions pre-filtered from the archive that have various uncertainties, (s)he has a wide range of solutions to make decisions if so desired.

5 Conclusions

In this paper, we proposed a framework for interactively solving offline data-driven MOPs. It enabled the DM to understand and provide preferences for uncertainties during an interaction. By using preferences for objectives, the DM can guide the solution process. The solutions generated follow the DM's preferences for objectives and have a variety of uncertainties. By preferences for uncertainties, the DM can control which solutions (s)he can see. The two-step interaction proposed in the framework does not significantly increase the cognitive load on the DM. We also demonstrated it by solving the GAA problem that proved its capability in solving many-objective problems. The visualization in the framework enabled the DM to provide preferences for uncertainties interactively. However, more work should be done in the field of reference vectors adaptation and development of comparison metrics for interactive approaches. We also need to perform tests with different types of preferences for objectives. Furthermore, the framework is not designed to handle constraints. Handling constraints for offline data-driven problems deserves further attention.

Acknowledgements

This research was partly supported by the Academy of Finland (grant number 311877) and is related to the thematic research area DEMO (Decision Analytics utilizing Causal Models and Multiobjective Optimization, `http://www.jyu.fi/demo`) of the University of Jyväskylä. This work was partially supported by the Natural Environment Research Council [NE/P017436/1].

References

1. Castro-Gama, M.: mariocastrogama/GAA-Problem-MATLAB (2017), `https://github.com/mariocastrogama/GAA-problem-MATLAB`, accessed April 21, 2020
2. Cheng, R., Jin, Y., Olhofer, M., Sendhoff, B.: A reference vector guided evolutionary algorithm for many-objective optimization. IEEE Transactions on Evolutionary Computation **20**(5), 773–791 (2016)
3. Chugh, T., Chakraborti, N., Sindhya, K., Jin, Y.: A data-driven surrogate-assisted evolutionary algorithm applied to a many-objective blast furnace optimization problem. Materials and Manufacturing Processes **32**(10), 1172–1178 (2017)
4. Deb, K., Jain, H.: 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), 577–601 (2014)
5. Deb, K.: Evolutionary Algorithms for Multi-Criterion Optimization in Engineering Design. In: Miettinen, K., Neittaanmäki, P., Mäkelä, M.M., Périaux, J. (eds.) Evolutionary Algorithms in Engineering and Computer Science. Wiley (1999)
6. Forrester, A., Sobester, A., Keane, A.: Engineering Design via Surrogate Modelling. John Wiley & Sons (2008)
7. Hakanen, J., Chugh, T., Sindhya, K., Jin, Y., Miettinen, K.: Connections of reference vectors and different types of preference information in interactive multiobjective evolutionary algorithms. In: Proceedings of the 2016 IEEE Symposium Series on Computational Intelligence (SSCI). pp. 1–8 (2016)
8. Jin, Y., Wang, H., Chugh, T., Guo, D., Miettinen, K.: Data-driven evolutionary optimization: An overview and case studies. IEEE Transactions on Evolutionary Computation **23**(3), 442–458 (2019)
9. Krishnamoorthy, K., Mathew, T.: Statistical Tolerance Regions: Theory, Applications, and Computation. John Wiley and Sons (2009)
10. Li, K., Chen, R., Min, G., Yao, X.: Integration of preferences in decomposition multiobjective optimization. IEEE Transactions on Cybernetics **48**(12), 3359–3370 (2018)
11. Mazumdar, A., Chugh, T., Miettinen, K., López-Ibáñez, M.: On dealing with uncertainties from Kriging models in offline data-driven evolutionary multiobjective optimization. In: Deb, K., Goodman, E., Coello Coello, C.A., Klamroth, K., Miettinen, K., Mostaghim, S., Reed, P. (eds.) Evolutionary Multi-Criterion Optimization, 10th International Conference, Proceedings. pp. 463–474. Springer (2019)
12. McKay, M.D., Beckman, R.J., Conover, W.J.: A comparison of three methods for selecting values of input variables in the analysis of output from a computer code. Technometrics **21**(2), 239–245 (1979)
13. Miettinen, K.: Nonlinear Multiobjective Optimization. Kluwer Academic Publishers (1999)
14. Purshouse, R.C., Deb, K., Mansor, M.M., Mostaghim, S., Wang, R.: A review of hybrid evolutionary multiple criteria decision making methods. In: Proceedings of the IEEE Congress on Evolutionary Computation (CEC). pp. 1147–1154 (2014)
15. Shah, R.A., Reed, P.M., Simpson, T.W.: Many-Objective Evolutionary Optimisation and Visual Analytics for Product Family Design. In: Wang, L., Ng, A.H.C., D.K. (eds.) Multi-objective Evolutionary Optimisation for Product Design and Manufacturing. pp. 137–159. Springer (2011)
16. Simpson, T.W., Allen, J.K., Chen, W., Mistree, F.: Conceptual design of a family of products through the use of the robust concept exploration method. In: 6th

AIAA/NASA/ISSMO Symposium on Multidisciplinary Analysis and Optimization. pp. 1535–1545 (1996)
17. Wang, H., Jin, Y.: A random forest-assisted evolutionary algorithm for data-driven constrained multiobjective combinatorial optimization of trauma systems. IEEE Transactions on Cybernetics **50**(2), 536–549 (2020)
18. Wang, H., Jin, Y., Sun, C., Doherty, J.: Offline data-driven evolutionary optimization using selective surrogate ensembles. IEEE Transactions on Evolutionary Computation **23**(2), 203–216 (2019)
19. Wang, H., Olhofer, M., Jin, Y.: A mini-review on preference modeling and articulation in multi-objective optimization: current status and challenges. Complex & Intelligent Systems **3**(4), 233–245 (2017)
20. Zhang, Q., Li, H.: MOEA/D: A multiobjective evolutionary algorithm based on decomposition. IEEE Transactions on Evolutionary Computation **11**(6), 712–731 (2007)
21. Zheng, J., Yu, G., Zhu, Q., Li, X., Zou, J.: On decomposition methods in interactive user-preference based optimization. Applied Soft Computing **52**, 952 – 973 (2017)