Surrogate-based Optimisation for a Hospital Simulation Scenario Using Pairwise Classifiers

Pablo S. Naharro*
Lurtis Rules
Universidad Politécnica de Madrid
Madrid, Spain
p.sanchez@lurtis.com

Antonio LaTorre
Center for Computational Simulation,
Universidad Politécnica de Madrid
Madrid, Spain
a.latorre@upm.es

José-María Peña
Lurtis Ltd.
Oxford, UK
jm.penya@lurtis.com

ABSTRACT

This paper presents a surrogate-based approach that uses a relatively simple population-based optimisation algorithm, a basic Differential Evolution algorithm (DE), and experiments with two complementary approaches to construct a surrogate. This surrogate-based optimisation uses a predictive model in-line and decides whether to calculate a candidate individual (using the simulation model) or discard it as part of the optimisation process. The complementary approaches for the design of the surrogate are (1) a traditional regression-based surrogate that approximates the surface of the fitness landscape using a supervised continuous machine learning algorithm (XGBoost), and (2) a pairwise approach that models the surrogate as a binary classification problem for a machine learning algorithm (in this experiment we proposed a Decision Tree binary classifier). Although there is no statistical difference in the performance of both surrogate approaches, the surface/regression one obtains a slightly better performance when the execution is limited to 200 fitness evaluations. In contrast, the pairwise/classification approach obtains the lowest value and a lower mean in executions with 750 fitness evaluations.

KEYWORDS

differential evolution, surrogate-based optimization, xgboost, decision trees, pairwise surrogates

ACM Reference Format:
Pablo S. Naharro, Antonio LaTorre, and José-María Peña. 2021. Surrogate-based Optimisation for a Hospital Simulation Scenario Using Pairwise Classifiers. 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/10.1145/3449726.3463283

1 INTRODUCTION

One of the tools that have shown promising effectiveness in optimising complex science and engineering problems is heuristic optimisation [3]. This tool performs a trial-error iterative process to infer information about the optimisation surface, which usually implies testing several candidate solutions (these problems may require executing one or several simulations). The computational cost of performing simulations is highly variable, and in some cases, this can turn into a handicap when there is a limit in resources (time or CPU). A strategy to get the most while not exceeding this limit consists of training models that approximate the fitness function that defines the problem. These models are typically referred to, in the literature, as surrogate models.

There are two ways to train these surrogates: (1) Using (all/most) of the budget to model the surface before the optimisation, and optimising using only the surrogate as the objective function. (2) Using the model during the optimisation to screen solutions and decide whether they are promising or not previous to be calculated; then, new solutions allow the surrogate to be retrained and updated [5].

2 ALGORITHM DESCRIPTION

The proposed optimisation algorithm uses DE as the base metaheuristic combined with a surrogate model, which the algorithm updates during the optimisation process. DE uses a population size of 8 and a *rand/1/exp* as mutation strategy. Both mutation and recombination factors are set to 0.5

Surrogate models to screen solutions require creating a training dataset previous to using the model (warm-up period). When the optimisation algorithm completes this warm-up period, it trains the first surrogate model. Later, at the end of every generation of the DE, the algorithm retrains the model. For this study, the warm-up period is set to 16 simulations.

In traditional surrogate-based approaches, the model is trained to predict the fitness value. We propose to reformulate the problem to use a surrogate model that predicts the comparison of two candidates instead [6]. In particular, for a canonical DE, the algorithm compares every newly generated candidate to a given challenged solution in order to perform the replacement.

One of the features of this approach is the increase in the size of the training dataset. Meanwhile, the surface approach generates a dataset $O(n)$, pairwise approach follows $O(n^2)$ by creating one input per pair of points. This strategy increases the information available for modelling in the early stages of the optimisation (for every new solution *n* calculated, the training dataset increases by *n-1* new data points). However, in further phases, training the model may become comparable to the simulation cost and might be necessary to limit the quadratic nature of this approach. A parameter called *trail size*, which limits the number of previous samples to match, is introduced when using the pairwise approach to prevent this situation. This limit is configured to the last 75 simulations performed.

The benefits of both approaches rely on its ability to discard samples. Nonetheless, when a surrogate algorithm drifts towards

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

ACM ISBN 978-1-4503-8351-6/21/07.
https://doi.org/10.1145/3449726.3463283

discarding all solutions, the whole optimisation process gets stuck into an undesired situation. This situation may become a potential infinite loop, in which neither new samples are calculated nor the model is updated (due to the lack of new data). To prevent this situation, there is a probability of 0.01 of executing the simulation even when the algorithm discarded it. This way, the surrogate algorithm may be updated and move away from this extreme behaviour.

3 EXPERIMENTS

The addressed problem consists of optimising the output of the BabSimHospital simulator [1], which models the number of ICU beds depending on 29 input parameters.

Several configurations have been analysed in order to choose the best performing among them. Each configuration has been run 5 times and the output parameter was the lowest fitness value reached. The proposed configurations are the following:

- DE with no surrogates as the baseline algorithm.
- DE + XGBoost (XGBR) [4] the continuous regressor.
- DE + Decision tree (DTC) [2] as the pairwise classifier.

Initial experiments use the Docker version of the problem, and the most promising one was tested against the online problem. All of them are executed for both 200 and 750 simulations as the budget limit. Later, an statistical analysis proposed in [7] has been performed to evaluate the different configurations. Firstly, Kruskal-Wallis test and pairwise Conover test are performed to identify differences across groups. Secondly, a ranking is created by assigning the best place to those that are not statistically worse to any configuration. Finally, a statistical analysis is performed repeatedly until all configurations are classified, excluding in the following iterations the ones already ranked.

4 RESULTS AND DISCUSSION

Results of the statistical analysis are shown in table 1. None of the configurations is statistically different from others, but surrogates improve in mean and minimum of the best-obtained value in both budgets. Although DE + XGBR obtains the best average for 200 evaluations, DE + DTC achieves the minimum value in both short and long runs. Also, the DTC version keeps progressing when more budget allowed in contrast with DE and DE + XGBR, which keeps in the discarding loop (triggering an early stop). If this non-improvement point is reached earlier when running an alternative version of the problem, it will mean that some of the budget will be wasted. Therefore, we have chosen DE + DTC as the candidate algorithm configuration. In general, the more aggressive a surrogate is, the more samples will save, and the better the result would be. However, the more aggressive the algorithm, the probability of discarding a significant sample is more likely. This drawback of DE + DTC can be seen in Figure 1 as a greater variance. The use of surrogates, by design, filters created versus evaluated candidates. Table 2 shows the average amount of calculated samples (same as budget), candidates proposed by DE, and the ratio among these two values. High surrogate aggressiveness can also lead to an algorithm only driven by the probability strategy, which is the behaviour exhibited by DE + XGBR, whose ratio tends to 0.01 (assigned probability).

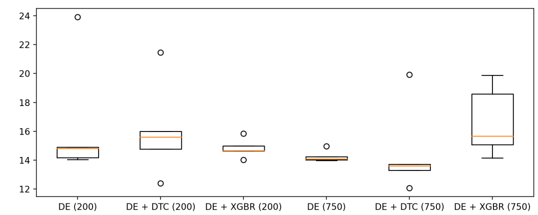

Figure 1: Boxplot of the optimum reached running the configurations against the offline version of the problem.

Table 1: Results of the experiments

Experiment	Min (200)	Mean (200)	Rank (200)	Min (750)	Mean (750)	Rank (750)
DE	14.05	16.37	1	13.98	14.26	1
DE + DTC	12.41	16.05	1	12.07	14.52	1
DE + XGBR	14.02	14.83	1	14.15	16.67	1

Table 2: Summary of created and calculated candidates

Experiment (budget)	Calculated	Created (avg)	Ratio
DE (200)	200	200	1
DE + DTC (200)	200	614.4	0.326
DE + XGBR (200)	200	6827.2	0.029

5 CONCLUSION

The pairwise surrogate approach can potentially increase the performance of the optimisation algorithm in scenarios with a lower number of available FEs by (1) modelling the decision boundary instead of the whole surface, and (2) augmenting the data available for training. DE + DTC reached 8.173219 on the online version.

6 ACKNOWLEDGMENTS

PSN and ALT would like to thank the Industrial PhD Programme of *Comunidad de Madrid* (grant IND2019/TIC-17140). PSN and JMP thank EIT Digital for their funding and support. ALT and JMP thank the Spanish Ministry of Science (grant TIN2017-83132-C2-2-R).

REFERENCES

[1] Thomas Bartz-Beielstein, Frederik Rehbach, Olaf Mersmann, and Eva Bartz. 2020. Hospital capacity planning using discrete event simulation under special consideration of the COVID-19 pandemic. arXiv:2012.07188
[2] Leo Breiman, Jerome Friedman, Charles J Stone, and Richard A Olshen. 1984. *Classification and regression trees.* CRC press, Monterey, CA.
[3] Kalyanmoy Deb. 2012. *Optimization for engineering design: Algorithms and examples.* PHI Learning Pvt. Ltd., New Delhi, India.
[4] Jerome H Friedman. 2001. Greedy function approximation: a gradient boosting machine. *Annals of statistics* 29 (2001), 1189–1232.
[5] Zhong-Hua Han and Ke-Shi Zhang. 2012. Surrogate-based optimization. In *Real-world applications of genetic algorithms.* IntechOpen, Rijeka, Chapter 17, 343–362.
[6] Pablo S. Naharro, Antonio LaTorre, and Jose-Maria Peña. 2021. Comparative study of regression vs. pairwise models for surrogate-based heuristic optimisation. *Under Review* (2021), 1–76.
[7] Frederik Rehbach, Martin Zaefferer, Joerg Stork, and Thomas Bartz-Beielstein. 2018. Comparison of Parallel Surrogate-Assisted Optimization Approaches. In *Proceedings of GECCO '18* (Kyoto, Japan). ACM, New York, NY, USA, 1348–1355.