

Solving Multi-Objective Problems by Means of Single Objective Solver

Todor Balabanov

Bulgarian Academy of Sciences

Institute of Information and Communication Technologies at

Acad. Georgi Bonchev Str., block 2, office 514, 1113 Sofia, Bulgaria

E-mail: todor.balabanov@iict.bas.bg

Abstract: The main difference between multi-objective optimization and single-objective optimization is that the first one has more than one target function and they need to optimize simultaneously. The solution of single-objective problems provides an optimal value in contrast to the multi-objective problems where a set of Pareto-optimal values can be found. That is why, one of most common approach for solving multi-objective problems is realize by transforming the multi objectives into a single objective. The simplest transformation is through an additive model, where a weighting factor is given on each objective. This research proposes a calculation procedure by which the LibreOffice Calc NLP Solver is used to generate solutions in the Pareto subset for multi-objective problems.

Keywords: multi-objective optimization, weighted sum model, LibreOffice Calc NLP Solver.

1. Introduction

Multi-objective optimization is part of general optimization. It has great practical importance since a lot of real-world problems suites for modeling by multiple conflicting objectives (Naka & Guliashki, 2021; Arora, 2017, Abraham & Jain, 2005). Some of the hard applications of multiple objective problems deals with job shop scheduling (Guliashki et al., 2019), staff scheduling (Borissova, 2015), transportation problems (Sudha & Ganesan, 2021), energy efficiency (Guliashki & Marinova, 2021), etc. A recent review of the developed multi-criteria decision-making models and their applications is proposed in (Borissova, 2021). The most popular approach to solving such problems is primarily focused on transforming

multiple objectives into a single objective including also the problems of group decision-making (Borissova & Dimitrova, 2021, Korsemov & Borissova, 2018).

Many practical optimization problems are admittedly defined as nonlinear problems having multiple conflicting objectives (Zitzler, et al., 2004). Because of the absence of proper solution techniques, these problems are mathematically transformed into a single-objective problem, and then well-known approaches for solving single-objective problems can be used. The complexity comes from the fact that multi-objective problems lead to a set of trade-off optimal solutions (Pareto-optimal set (Zitzler & Thiele, 1998)). In contrast, single-objective problems, in general cases, lead to a single optimum solution. The goal in multi-objective optimization is to find as many Pareto-optimal solutions as possible. The availability of many Pareto-optimal solutions is important because each taken two by two constitutes a trade-off in the objectives. The decision-maker would have more options to select which objectives to compromise (Deb, 2014).

In multi-objective optimization, there are two or more functions for which the optimal value is sought, for the same input vector. If the input vector is denoted by x , then its components can be denoted by $\{x_1, \dots, x_n\}$. The presence of multiple functions for simultaneous optimization $\{f_1, \dots, f_m\}$ (Emmerich & Deutz, 2018) leads to the appearance of two optimization spaces – the search space of variables and the search space of objectives (Deb et al., 2005). For the variables, lower and upper bounds may be imposed as possible values. In addition to the finite constraints, it is possible to specify linear and nonlinear equations/inequalities to be met. If only one objective or only one constraint has a nonlinear nature, then the problem becomes nonlinear.

The most used scalarization (Miettinen & Makela, 2002) formula is the additive model. In this model, each objective has a weight associated with it (Marler & Arora, 2010). The weighted objectives are generalized as follows:

$$f = w_1f_1(x) + \dots + w_mf_m(x) \quad (1)$$

It should be mention that all of the objective functions need to be normalized to get a dimensionless value of the function (1). These normalizations can be done by using the different schemes as shown in (Marler & Arora, 2004).

The weighted sum model is applied in the present study. In the general case, the minimum value of the function (Eq. 1) is seeking. A minimum value is sought also for each of the objectives. If a maximum is sought for any of the objectives, its weight participates as a negative value.

Due to importance of multi-objective optimization for solving different real-life problems, it is of great importance to use on well-known software tools like MS Excel or LibreOffice. Such tools are already developed for determination of optimal scheduling when processing dependent details in MS Excel (Borissova et al, 2018; Borissova, 2008) and most recent tool in LibreOffice is related to incremental approximation of time series (Velichkova et al., 2021).

All of these determine the necessity of development a tool to solve multi-objective problems by the weighted sum model. In this regard, the current study proposes a single-objective NLP Solver of LibreOffice Calc to be used for multi-objective problems. The rest of this paper is organized as follows: Section 1 introduces the problem with searching for Pareto optimal solutions; Section 2 describes the way in which a single-objective solver can be used for multi-objective problems; Section 3 reveals some practical experiments and related results; and Section 4 concludes with some suggestions for further work.

2. LibreOffice Calc NLP Solver

LibreOffice is free open source software that offers applications for the needs of small offices. The software consists mainly of the following modules: Writer (text processing), Calc (spreadsheets), Impress (presentations), Draw (vector graphics), Math (mathematical formulas), and Base (databases). In the spreadsheet module, it is possible to activate linear and nonlinear solvers (*Tools->Solver*) for optimization problems. Nonlinear optimization problems are of greater interest because they differ in complexity from linear problems. The NLP Solver submodule in LibreOffice Calc is implemented by a hybrid combination of Differential Evolution and Particle Swarm Optimization.

The NLP Solver itself is written in Java for the current stable release. The development team plans to rewrite this submodule in C++, but this is a medium to a long-term task. At this stage, the NLP Solver in LibreOffice Calc is designed to seek solutions to single-objective tasks only. This means that the user is offered only a single solution. It is like this even when the task is multimodal and it has more than one global optimum. The heuristic nature of both algorithms in the solver determines that the proposed solutions do not guarantee global optimality. Usually, solutions are close to the global optimum or one of the global optimums when there is more than one.

If someone wants the NLP solver to be used for multi-objective tasks, it is not impossible to add the necessary software functionality in the seventh Java code of the solver. Although possible, such work is too complex and would take a long time, even for top software professionals. The presence of scripting languages, such as Python, in LibreOffice, allows a much easier approach to solving multi-objective tasks with a single-objective solver. The first step in performing this calculation is to assign a weighting factor for each objective, as shown in Eq. 1. The second step is related to the fact that the solver provides only a single solution. In order to calculate solutions, close to the Pareto front, the solver must be started multiple times. Different weighting coefficients of the objectives are selected for each start of the solver. The difference in the coefficients symbolizes the possibility for the decision-maker to determine the different importance of the individual objectives. Having an initial set of solutions, the decision-maker can

manually change the coefficients of the objectives and manually start the solver to generate new solutions close to the Pareto front.

3. Searching Pareto Points for Binh and Korn Function

The following Binh and Korn function (Maghawry, et al., 2021) is used as a benchmark:

$$\min f_1(x, y) = 4x^2 + 4y^2 \tag{2}$$

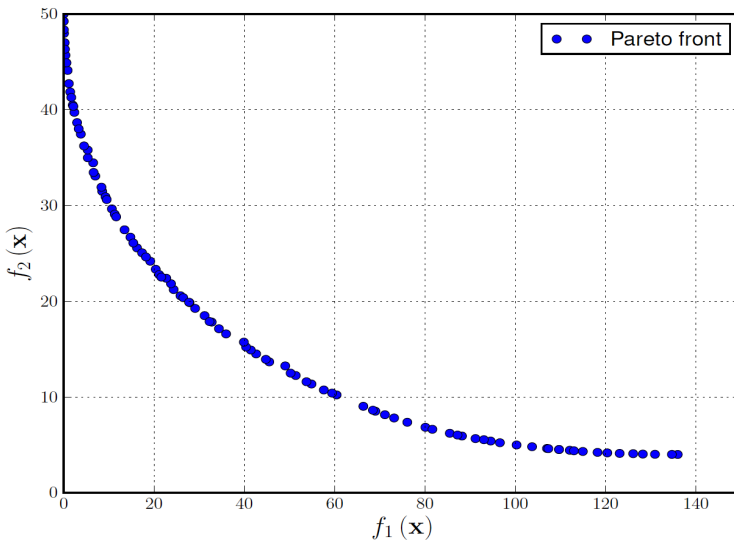
$$\min f_2(x, y) = (x - 5)^2 + (y - 5)^2 \tag{3}$$

$$g_1(x, y) = (x - 5)^2 + y^2 \leq 25 \tag{4}$$

$$g_2(x, y) = (x - 8)^2 + (y + 3)^2 \geq 7.7 \tag{5}$$

$$0 \leq x \leq 5, 0 \leq y \leq 3 \tag{6}$$

The Pareto front of the Binh and Korn function based on the formulation (Eq. 2-6) is shown in Fig. 1.



https://upload.wikimedia.org/wikipedia/commons/7/70/Binh_and_Korn_function.pdf

Fig. 1. Binh and Korn function

It is obvious from the chart that coefficients for f_1 should be much lower in order for some of the Pareto points to be generated. Experiments are done on Intel Core i5 2,3 GHz, Single CPU with 2 cores, 8GB RAM, macOS High Sierra 10.13.6, Java SE 11.0.2, LibreOffice 7.0.6.2. Python script and optimization model are publicly available (Balabanov, 2021).

Before starting the macro, it is important to set the optimization model in the solver dialog box (Fig. 2).

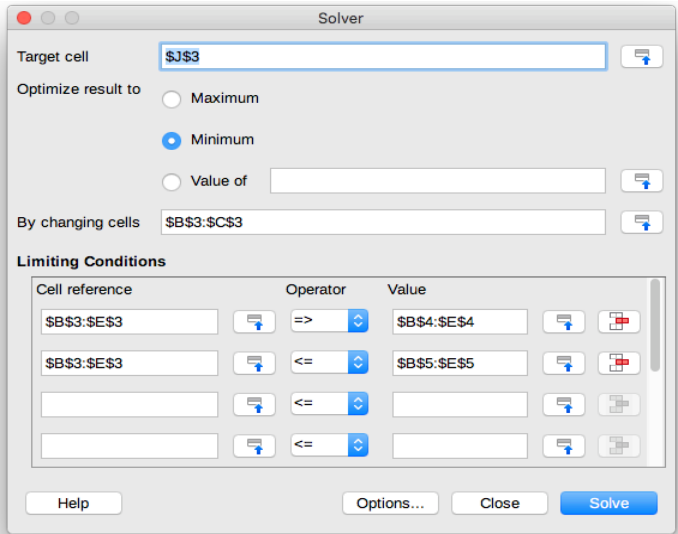


Fig. 2. Optimization model setup

It is possible to set the parameters of the model through the solver's API. It has some complexity and is therefore omitted in this study. When generating 100 points, the boundaries of the emerging Pareto front are clearly visible (Fig. 3).

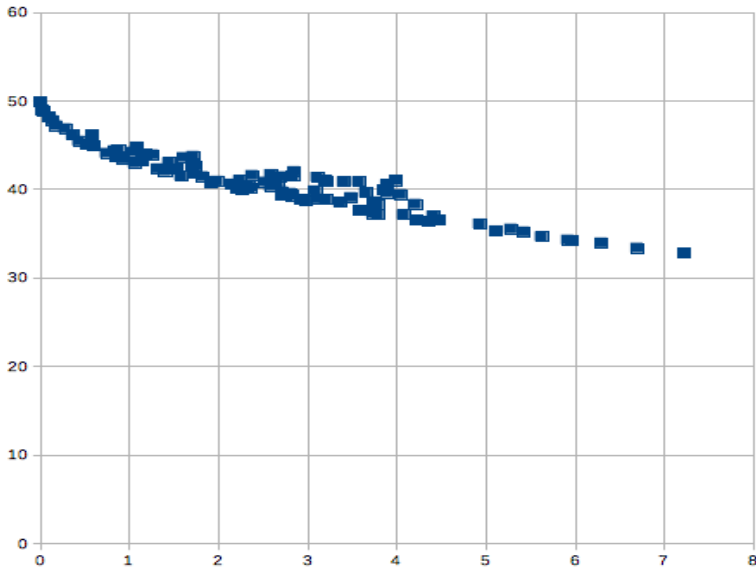


Fig. 3. Set of near Pareto optimal solutions

The points are not located exactly on the Pareto front. This is because heuristic optimization gives close to optimal but not guaranteed optimal solutions. The other very obvious thing is that there is a lack of the right side of the

theoretical Pareto front. The lack of part of the front is due to the fact that points in this area can be obtained at too large values for one coefficient and too small values for the other coefficient. The most important part of multi-objective optimization is the way in which scaling coefficients are selected. With the exception of randomly generated coefficients, a list of predefined values can be used. Very large or very small values can be included in such a list. The experiments are performed with Binh and Korn function, but any other benchmark or real function can be taken.

4. Conclusion

This research examines the capabilities of a single-objective solver to be applied to multi-objective problems. By giving random weights for the objectives, various solutions are provided close to the Pareto front. The stochastic nature of the solver does not allow the proposed solutions to be on the front itself, and they are positioned close to it. From a mathematical point of view, this is not acceptable, but in real practice, even decent solutions are preferable in the absence of any solutions.

The application of random coefficients does not always give a good distribution of the points around the Pareto front. It would be interesting to apply some strategy for a wider study of the possible values for the coefficients. Although it is possible to change the coefficients manually, some form of automation would significantly speed up the process of finding an acceptable solution.

Acknowledgment

This research is funded by Velbazhd Software LLC and it is partially supported by the Bulgarian Ministry of Education and Science (contract D01–205/23.11.2018) under the National Scientific Program „Information and Communication Technologies for a Single Digital Market in Science, Education and Security (ICTinSES)“, approved by DCM # 577/17.08.2018.

References

1. Abraham, A., Jain, L.: Evolutionary Multiobjective Optimization. Evolutionary Multiobjective Optimization. Advanced Information and Knowledge Processing, Springer, 1–6 (2005), https://doi.org/10.1007/1-84628-137-7_1.
2. Arora, J. S.: Chapter 18 – Multi-objective Optimum Design Concepts and Methods. Editor(s): J. S. Arora, Introduction to Optimum Design (Fourth Edition), Academic Press, 771–794 (2017), <https://doi.org/10.1016/B978-0-12-800806-5.00018-4>.

3. Balabanov, T.: Binh and Korn Function Optimization with LibreOffice Solver. (2021). <https://github.com/TodorBalabanov/Binh-and-Korn-Function-Optimization-with-LibreOffice-Solver>, last accessed 2021/08/09.
4. Borissova, D.: Optimal scheduling for dependent details processing using MS Excel Solver. *Cybernetics and Information Technologies* 8(2), 102–111 (2008).
5. Borissova, D., Dimitrova, Z.: An integrated group decision-making approach considering uncertainty conditions. In: Proc. of 24th International Conference on Business Information Systems (BIS'2021) pp. 307–316 (2021), <https://doi.org/10.52825/bis.v1i.52>.
6. Borissova, D., Korsemov, D., Mustakerov, I.: Multi-attribute group decision making considering difference in experts knowledge: an Excel application. In: Proc. of 12th International Management Conference on Management Perspectives in the Digital Era, pp. 387–395 (2018).
7. Borissova, D.: An optimal staffing and scheduling approach in open shop environment. *Comptes rendus de l'Academie bulgare des Sciences* 68(10), 1295–1300 (2015).
8. Borissova, D.: An overview of multi-criteria decision making models and Software Systems. In: Atanassov K.T. (ed.) *Research in Computer Science in the Bulgarian Academy of Sciences. Studies in Computational Intelligence*, vol. 934. pp. 305–323 (2021), https://doi.org/10.1007/978-3-030-72284-5_15.
9. Deb, K., Thiele, L., Laumanns, M., Zitzler, E.: Scalable test problems for evolutionary multiobjective optimization. *Evolutionary Multiobjective Optimization, Advanced Information and Knowledge Processing*. Springer, London, pp. 105–145 (2005), https://doi.org/10.1007/1-84628-137-7_6.
10. Deb, K.: *Multi-objective optimization. Search Methodologies*, Springer, Boston, MA, 403–449 (2014), https://doi.org/10.1007/978-1-4614-6940-7_15.
11. Emmerich, M., Deutz, A.: A tutorial on multiobjective optimization: fundamentals and evolutionary methods. *Natural Computing* 17, 585–609 (2018), <https://doi.org/10.1007/s11047-018-9685-y>.
12. Guliashki, V., Marinova, G., Groumpos, P.: Multi-objective optimization approach for energy efficiency in microgrids. *IFAC-PapersOnLine* 52(25), 477–482 (2019), <https://doi.org/10.1016/j.ifacol.2019.12.587>.
13. Guliashki, V., Marinova, G.: Optimization approach for improvement of energy efficiency of buildings in a microgrid. *IEICE Information and Communication Technology Forum* (2021), https://doi.org/10.34385/proc.64.ICTF2020_paper_5.
14. Korsemov, D., Borissova, D.: Modifications of simple additive weighting and weighted product models for group decision making. *Advanced Modeling and Optimization* 20(1), 101–112, (2018).
15. Maghawry, A., Hodhod, R., Omar, Y., Kholief, M.: An approach for optimizing multi-objective problems using hybrid genetic algorithms. *Soft Computing* 25, 389–405 (2021), <https://doi.org/10.1007/s00500-020-05149-3>.

16. Marler, R.T., Arora, J.S.: The weighted sum method for multi-objective optimization: new insights. *Structural and Multidisciplinary Optimization* 41, 853–862 (2010), <https://doi.org/10.1007/s00158-009-0460-7>.
17. Marler, R.T., Arora, J.S.: Survey of multi-objective optimization methods for engineering. *Structural and Multidisciplinary Optimization* 26, 369–395 (2004), <https://doi.org/10.1007/s00158-003-0368-6>.
18. Miettinen, K., Makela, M.: On scalarizing functions in multiobjective optimization. *OR Spectrum* 24(2), 193–213 (2002), <https://doi.org/10.1007/s00291-001-0092-9>.
19. Naka, E., Guliashki, V.: Optimization techniques in data management: A survey. In: 7th International Conference on Computing and Data Engineering, pp. 8–13 (2021), <https://doi.org/10.1145/3456172.3456214>.
20. Sudha, G., Ganesan, K.: A solution approach to time variant multi-objective interval transportation problems in material aspects. *Materials Today: Proceedings*, (2021), <https://doi.org/10.1016/j.matpr.2021.01.175>.
21. Velichkova, V., Tomov, P., Balabanov, T.: Incremental sinusoidal approximation of time series with LibreOffice Calc Solver. *Problems of Engineering Cybernetics and Robotics* 75, 43–50 (2021), <https://doi.org/10.7546/PECR.75.21.05>.
22. Zitzler, E., Laumanns, M., Bleuler, S.: A tutorial on evolutionary multiobjective optimization. In: Gandibleux X., Sevaux M., Sorensen K., T'kindt V. (eds) *Metaheuristics for Multiobjective Optimisation. Lecture Notes in Economics and Mathematical Systems*, vol. 535, pp. 3–37 (2004), https://doi.org/10.1007/978-3-642-17144-4_1.
23. Zitzler, E., Thiele, L.: Multiobjective optimization using evolutionary algorithms – A comparative case study. In: Eiben, A.E., Back, T., Schoenauer, M., Schwefel, HP. (eds) *Parallel Problem Solving from Nature — PPSN V. PPSN 1998. Lecture Notes in Computer Science*, vol. 1498, pp. 292–301 (1998), <https://doi.org/10.1007/BFb0056872>.