**Multi-Objective Optimization in Process Simulation using Stochastic Algorithms**

Fabian Zapf1,\*, Thomas Wallek1

*1 Institute of Chemical Engineering and Environmental Technology, Graz University of Technology, Inffeldgasse 25C, 8010 Graz, Austria*

*\*Corresponding author: fabian.zapf@tugraz.at*

**Highlights**

* Multi-objective optimization
* Process simulation
* Scalarization methods
* Genetic algorithm

**1. Introduction**

The use of a variety of commercial process simulators is state of the art, offering the possibility for calculating mass transfer phenomena, thermodynamics and reaction kinetics of single processes, individual production plants and entire production sites. The last step in process simulation is the optimization of the design and/or operating parameters of the plant in view of economy, operational safety, environmental impacts and social aspects. As these aspects are often conflicting each other, this frequently implies a multi-objective optimization (MOO) problem, also called Pareto optimization. MOO problems always include two or more objectives, common examples for that are investment costs vs. operational costs and profitability vs. environmental aspects.

Popular process simulators, like Aspen Plus, Aspen HYSYS, CHEMCAD and gProms, include optimization modules, which are capable of performing only single objective optimization (SOO). To overcome this drawback, introducing MOO in process simulators is commonly realized by coupling the process simulator with an external application which runs the optimization algorithm. In general, there exist two groups of optimization algorithms: deterministic and stochastic algorithms.

Deterministic optimization algorithms require gradient information of the optimization problem, which makes them very fast compared to stochastic algorithms. Gradient information in process simulators is only available when they are running in equation orientated simulation mode via the equation set object, defined by the CAPE-OPEN standard. [1]

Stochastic optimization algorithms, on the other hand, require only information about the dependent variables (objectives or output of the system) and the independent variables (process parameters to be optimized). Therefore, they are suitable for optimizing black-box functions, such as process simulators running in sequential modular mode. Their main disadvantage is the relatively slow convergence, due to many evaluations of the objective functions in the course of evaluation runs of the process simulator.

In this work, several approaches for applying MOO in process simulation are compared among their effort for realization and their suitability for optimization in process engineering. They are implemented in the framework of Wolfram Mathematica, which is coupled with KBC PetroSIM as a process simulator via a COM Interface.

**2. Methods**

Scalarization methods, which transform a MOO problem (vector) into a SOO problem (scalar) are very popular due to the fact that they are easy to implement and the broad availability of built-in SOO algorithms in nearly every mathematical program. [2] Therefore, as a first step, popular scalarization methods, such as the weighting method and *ε*-constraint method are implemented and their resulting SOO problems are solved by the built-in global SOO algorithms, i.e., simulated annealing, differential evolution and Nelder-Mead. [3]

As a second step, a genetic algorithm is implemented. Genetic algorithms are population based and a group of them is able to handle MOO problems directly. One of the most popular algorithms is the Non-dominated Sorting Genetic Algorithm II (NSGA-II). This standard-algorithm is implemented together with several adaptions for handling integer variables. Additionally, five different jumping-gene operators as well as the hyper-volume of the Pareto front as a convergence measure are implemented as extensions to NSGA-II. [2,4]

**3. Application**

To compare these different implementations and variations for MOO, they are applied to the example of a steam reformer located in a European refinery, optimizing the process in view of specific hydrogen production (kg H2/kg natural gas) and HPS (high pressure steam) production. Using this example, strengths and weaknesses of the aforementioned approaches are analyzed and recommendations for their application are given.

**3. Conclusion**

The presented work demonstrates the workflow and efforts to enable MOO in process simulators with the example of optimizing a steam reformer in view of process flexibility. Beyond the typical application of MOO for economic optimization, it gives the possibility of evaluating the increasing demand of process flexibility and load-shift capability of plants and processes.

**References**

1. C.A.M. Lopez, D. Telen, P. Nimmegeers, L. Cabianca, F. Logist, J. V. Impe, A process simulator interface for multiobjective optimization of chemical processes, Computers and Chemical Engineering, Vol 109, 2018, pp. 119-137
2. G.P. Rangajah, Introduction, Multi-Objective Optimization, Advances in process systems engineering, Vol 5, 2016, pp 1-27
3. R.T. Marler, J.S. Arora, Survey of multi-objective optimization methods for engineering, Structural and Multidisciplinary Optimization, Vol 26, 2004, pp 369-395
4. M. Ramteke, S.K. Gupta, Multi-Objective Genetic Algorithm and Simulated Annealing with the Jumping Gene Adaptations, Multi-Objective Optimization, Advances in process systems engineering, Vol 5, 2016, pp 93-133