

Multi-objective optimization of dynamic systems combining genetic algorithms and Modelica: Application to adsorption air-conditioning systems

Uwe Bau¹ Daniel Neitzke¹ Franz Lanzerath¹ André Bardow¹

¹Institute of Technical Thermodynamics, RWTH Aachen University, Germany,
andre.bardow@lth.rwth-aachen.de

Abstract

The Modelica language enables the fast and convenient development of physical simulation models. These models are often used for simulation studies. The re-use of simulation models for optimizations requires model-adaptions, additional tools or libraries. In this paper, we present a framework to connect Modelica models developed in Dymola to MATLAB's optimization toolbox. As optimization algorithm, we use a multi-objective genetic algorithm. The optimization procedure is tested for an adsorption air-conditioning design. Compared to a full factorial design, the optimization procedure produces better solutions using less evaluations.

Keywords: multi-objective, optimization, Pareto-resolution, dynamic systems, full factorial design, genetic algorithms, MATLAB, gamultiobj, NSGA-II, adsorption air-conditioning systems

1 Introduction

Modelica is an acausal, object-oriented and equation-based programming language with a high number of physical model libraries available. These language features allow for fast and convenient development of physical models. These models are used to enhance system understanding and performance by simulation and parameter variations. To go beyond parameter variations, optimization functionality is desired, e.g. for product design, parameter estimation or optimal control. To reduce development costs and effort, it would be desirable to employ simulation models directly for optimization.

The currently available options to transform a simulation model into an optimization model differ widely, starting from commercial Modelica libraries to freely available external tools. Several optimization tools are available specifically for Modelica:

For Dymola, the commercial library Design/Optimization exists, which includes different optimization algorithms, such as Sequential Quadratic Programming or Genetic Algorithm (Pfeiffer, 2012).

The open-source project JModelica provides Optimica, an extension of the Modelica language. Optimica allows for high-level formulation of optimization problems using modelica models (Lind et al., 2012; Dietl et al., 2014).

OMOptim is an optimization framework based on the open source OpenModelica platform. OMOptim includes meta-heuristic optimization algorithms such as genetic algorithms and is under further development (Thieriot et al., 2011).

Alternatively, Modelica models can be connected to external optimization tools:

GenOpt is such an open source optimization framework, which can be used for any simulation tool allowing input file modification and output reading. GenOpt includes several optimization algorithms, such as Generalized Pattern Search or Particle Swarm Optimization (Wetter, 2000).

Modelon provides a commercial Functional Mockup Interface (FMI) for MATLAB's widely used optimization toolbox (Henningsson et al., 2014). This FMI allows Modelica models to be optimized with MATLAB.

FMI has also been used to connect Modelica models to connect via FMI to the optimization code MUSCOD-II for gradient-based dynamic optimization (Gräber et al., 2011; Leineweber et al., 2003).

Which optimization approach suits best, depends on the resources and experience of the developer. In this work, we present a convenient approach connecting MATLAB's optimization toolbox for multi-objective optimization tasks to existing Modelica models. Here we used Modelica models developed within Dymola using our LTT Adsorption Energy Systems Library (Bau et al., 2014). We coded the MATLAB-Dymola interface using Modelica script files (.mos) embedded within the MATLAB-code.

In this paper, we present the developed interface for a design case study of an adsorption air conditioning system for battery-driven busses. As optimization algorithm, we use a multi-objective genetic algorithm, which is robust and has few requirements regarding model char-

acteristics. The problem has 8 design parameters, which are optimized regarding 2 objective functions.

In Section 2, we describe the optimization algorithm used and the MATLAB/Dymola interface. Section 3 contains the non-linear dynamic process model used for the design task. In Section 4, we discuss the obtained optimization results and compare the genetic optimization algorithm with a full factorial design. Finally, we summarize our findings in Section 5.

2 Optimization procedure

This paper studies the optimization of conflicting key performance indicators. Therefore, we define the resulting multi-objective optimization problem in Section 2.1. In Section 2.2, we present the applied optimization algorithm followed by the framework to interact with Modelica in Section 2.3. The handling of infeasible solutions and unsatisfied path-constraints is discussed in Section 2.4.

2.1 Multi-objective optimization problem

A designer often seeks to optimize several conflicting key performance indicators of a product by varying independent design parameters. Each parameter has usually a certain feasible range. Furthermore, feasible solutions often need to satisfy additional constraints (e.g. physical or operational limits).

The n_O relevant key performance indicators are here called objectives O with

$$O_i = f_i(x, z, p), \quad i = 1, \dots, n_O \quad (1)$$

as a function of differential states $x(\tau)$, algebraic states $z(\tau)$ and independent design parameters p . τ represents time with

$$\tau \in [0, T] \quad (2)$$

The n_p design parameters or decision variables p are restricted to a feasible decision or solution space $S \in \mathbb{R}^{n_p}$ within the constraints or bounds $[LB, UB]$. Furthermore, the differential and algebraic states x and z are required to satisfy so called path-constraints $LB^* \leq x, z \leq UB^*$ at any given moment in time τ .

The problem can be mathematically defined as:

$$\begin{aligned} & \min_{x, z, p} O & (3) \\ \text{s.t.} & \dot{x} = f(x, z, p) & (4) \\ & 0 = g(x, z, p) \\ & LB \leq p \leq UB \\ & LB^* \leq x, z \leq UB^* \quad \forall \tau \\ & x(\tau = 0) = x_0 \\ & p \in S, S \in \mathbb{R}^{n_p} \end{aligned}$$

The designer is concerned about the global optimum. When two solutions are compared regarding their objective values, the solution $O_{\text{dom}} = f(p_{\text{dom}})$ is said to dominate solution $O_{\text{inf}} = f(p_{\text{inf}})$, if none of the objectives $O_{i,\text{dom}}$ is worse than $O_{i,\text{inf}}$ and at least one objective $O_{j,\text{dom}}$ is strictly better than $O_{j,\text{inf}}$.

The set of solutions that are not dominated by any other solution in the entire feasible solution space S are called Pareto-optimal.

2.2 Optimization algorithm

The dynamic system studied in this paper consists of complex, non-linear and spatial discrete differential equations. The differentiation of the objectives O with respect to the design parameters p can be complicated and time consuming, especially for dynamic and complex systems. Deterministic gradient-based optimization algorithms also require good initial guesses. The relatively high number of 8 design parameters and possible combinations complicates this approach.

In this paper, we therefore use a genetic algorithm, belonging to the group of derivative-free or heuristic algorithms. Other possible heuristic algorithms are reviewed in Konak et al. (2006) or Jones et al. (2002).

A genetic algorithm tries to imitate the evolutionary process of nature. It interprets solutions as individuals and design parameters as their genes, which are passed, mixed and mutated from parents to a new generation of children. Ultimately, only the fittest individuals with most optimal objective values “survive” and are chosen as new parents.

In case of incomplete or imperfect information on the behavior of the objective functions or limited computation capacity, stochastic meta-heuristics like genetic algorithms offer a satisfying approach to analyze the complex correlation of the design parameter’s influence on the objective values. Especially in case of a highly non-convex shaped solution space S , gradient-based algorithms are susceptible to converge to local optima (Blum and Roli, 2003). In contrast, heuristic approaches with random initialization and spontaneous mutation of genes allows to move the population away from local optima (Tomoiagă et al., 2013).

According to Audet and Vicente (2008), a genetic algorithm generally has a tendency to rapidly converge to generally “good” solutions in early iterations and predominantly “smooth” the distribution of Pareto-solutions in later iterations. Therefore, a genetic algorithm is best suited to successfully address an optimization problem for cases in which three major aspects apply: First, the differentiation of the objective functions within the bounds of the domain is expensive and similar approaches to approximate the derivatives (e.g. by finite-difference) are prohibited. Second, the objective functions are not excessively non-smooth (e.g. event-based binary factorization). And third, finding a “good” lo-

cal optima fast is significantly more important than definitely converging to the global optimum.

Several different software packages or libraries contain heuristic optimization methods, such as the global optimization toolbox for MATLAB. We use MATLAB's *gamultobj*-function, which is based on a variant of the Non-dominated Sorting Genetic Algorithm-II (NSGA-II). The NSGA-II not only favors elite, but also diverse solutions and therefore covers a wide range of the solution space S (Deb, 2001). The *gamultobj*-function itself offers a set of variable algorithm options to customize key randomization properties as well as termination conditions regarding computation time.

Genetic algorithms generate a fixed number of n new parameter sets to be evaluated at each generation. These parameter sets can be split for parallel processing on multi-core computers, which can reduce computation significantly. However, since both algorithms applied in this paper (genetic and full-factorial) can be processed parallel, this topic is not discussed any further.

2.3 Optimization framework

Our implementation employs current versions of both MATLAB (R2014a) and Modelica (2014 v3.2). The model of the presented case study in this paper is based on Modelica models from our LTT Adsorption Energy Systems Library (Bau et al., 2014).

A framework to connect *gamultiobj* to existing Modelica models in Dymola is implemented, so that no further commercial optimization program is required. *gamultiobj* treats the Modelica model as a black-box. *gamultiobj* sets parameters as input values for the Modelica model, which produce objectives as output values.

A flow chart of the framework's procedure is displayed in Figure 1 and can be described as followed. The procedure first calls

$$p = \text{gamultiobj}(f_{\text{fitness}}, n_{\text{vars}}, A, b, \dots, A_{\text{eq}}, b_{\text{eq}}, LB, UB, \text{options}) \quad (5)$$

which randomly selects a set of parameters p as initial population (1 \rightarrow 2). The (in-)equality constraints and parameter bounds satisfy the following equations:

$$A \cdot p \leq b \quad (6)$$

$$A_{\text{eq}} \cdot p = b_{\text{eq}} \quad (7)$$

$$LB \leq p \leq UB \quad (8)$$

However, we only use Equation 8 and set $A = b = A_{\text{eq}} = b_{\text{eq}} = []$, the empty matrix. In each iteration, *gamultiobj* calls a fitness-function

$$O = f_{\text{fitness}}(p) \quad (9)$$

The fitness function usually contains the models equations describing the correlation between the objective O

and the parameters p . Here, we call a Modelica script file (.mos), which executes the Dymola command *simulateExtendedModel()* while passing the parameter set p (2 \rightarrow 3). The objective values are stored in a MATLAB data file (.mat) and evaluated in f_{fitness} . Finally, the resulting objective values of each solution are returned as output variables of f_{fitness} to *gamultiobj* (3 \rightarrow 4). The genetic algorithm then processes the fitness and diversity of the solution and selects the best (elite) and most diverse solutions as the parent individuals of the next generation (4). Due to random recombination, mutation and migration of individuals (5), a new set of parameters p is generated and the iteration starts over until a termination criteria applies (6 \rightarrow 7, 2). For example, one criteria can be the exceeding of the maximum generation number.

In order to apply this procedure for any given optimization problem for an existing Modelica model, the user only has to define the design parameters and their corresponding bounds, as well as the objective functions. Neither the model, nor the algorithm needs to be altered to suit a tailor-made optimization. Even the modification of the algorithm options in *gamultiobj* is optional.

One such option is the population size, which is constant for each generation. This can lead to the loss of Pareto-solutions over generations. Therefore, we store every Pareto-set of each generation in a separate file to keep all elite solution.

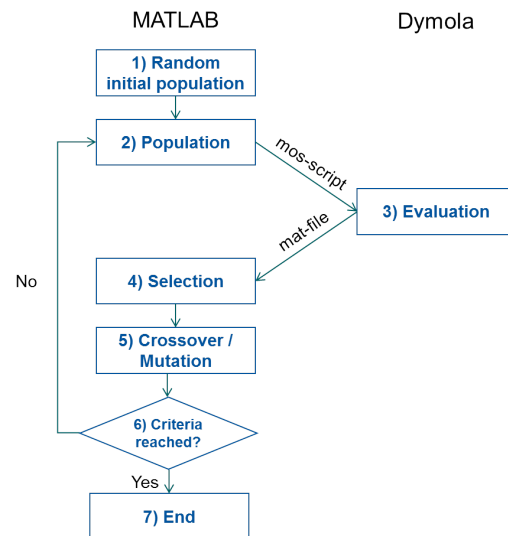


Figure 1. Procedure of optimization with genetic algorithm using MATLAB's global optimization toolbox and Dymola for evaluation (system simulation and objective calculation).

2.4 Handling infeasible solutions and unsatisfied path-constraints

The parameter constraints $LB \leq p \leq UB$ are satisfied by *gamultiobj* by choosing only feasible parameter values p for a simulation. However, the satisfaction of path-constraints and the feasibility of the simulation (e.g. due

to unsolvable stiff differential equations) depend on the choice of parameters and can only be checked after each simulation. Therefore, we implement an evaluation stage in MATLAB, which adds a penalty function PF to the objective function O :

$$O^* = O + PF \quad (10)$$

The penalty function PF is a piecewise function

$$PF = \begin{cases} 0, & LB^* \leq x, z \leq UB^* \\ \infty, & \text{else} \end{cases} \quad (11)$$

Infeasible solutions and unsatisfied path-constraints lead to positive infinite objective values and are therefore unfavored by the minimization problem. Alternative means of implementing penalty functions are possible (e.g. continuous penalty function or penalty function as separate objective value), but are not discussed in this paper.

3 Case Study: Adsorption-storage air-conditioning system

An adsorption air-conditioning system for electrical busses serves as representative case study of a dynamic and complex system in this paper. For electric busses, the air-conditioning takes up to 50% of the battery capacity when using a battery-driven compression chiller (Bottiglione et al., 2014). The adsorption air-conditioning (AC) system aims at providing cooling and heating. Since thermal energy is stored as latent and adsorption enthalpy, operating the AC system requires only a small of the battery capacity for ventilation. In the following Sections 3.1 - 3.4, the cooling mode is presented. For additional information see also (Bau et al., 2015).

3.1 Concept

The presented adsorption air-conditioning concept is based on the Pennington cycle (Pennington, 1955). During a bus ride, the adsorption AC system provides cooling and dehumidification (see Figure 2):

- Bus air (green arrows) flows through the desiccant module, which contains the sorbent material. In the desiccant module, moisture is adsorbed and hot and dry air leaves the module (4) → (5).
- The hot and dry air is cooled by a heat exchanger (5) → (6). As coolant stream, ambient air (blue arrows) is used, which is cooled by evaporative cooling (1) → (2) before it enters the heat exchanger and takes heat from the hot and dry bus air (2) → (3).
- The bus air (6) is finally cooled down to the desired temperature by evaporative cooling (6) → (7).

Within this study, the AC system is optimized assuming a dry adsorption module at beginning of bus ride.

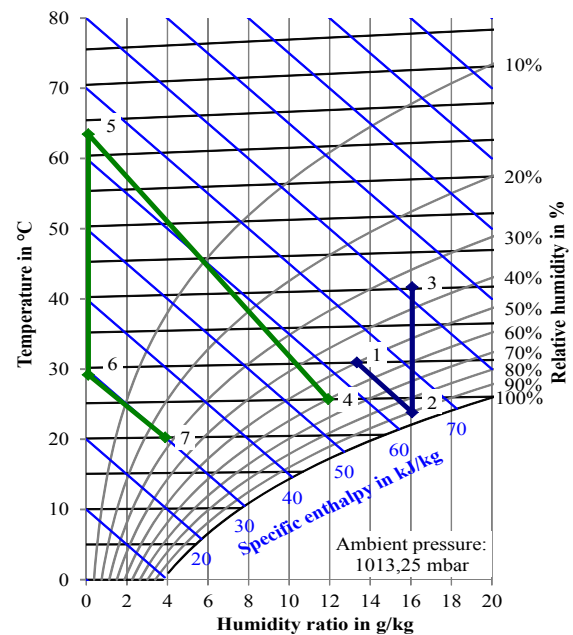
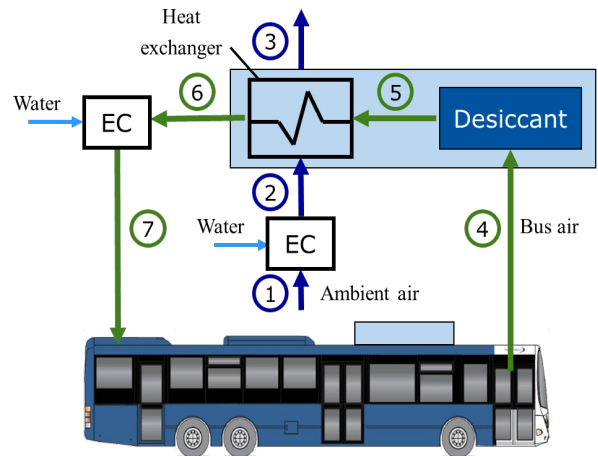


Figure 2. Cooling mode of adsorption storage: Process scheme (top) and Mollier diagram (left).

3.2 Model

The main parts of the model are the desiccant module and the heat exchanger (see Figure 3). Within these models, the air volume, the heat exchanger plates and the adsorbent are coupled by heat and mass transfer models. The Modelica model is built by using the LTT Adsorption Energy Systems library (Bau et al., 2014) and the TLK TIL library (Gräber et al., 2010). For further information regarding the heat and mass transfer correlations used, see Bau et al. (2014).

3.3 Key performance indicators (objectives)

The system performance can be quantified by two performance indicators: Specific cooling power (SCP) and coefficient of performance (COP). The specific cooling power (SCP) measures the power density by the ratio of

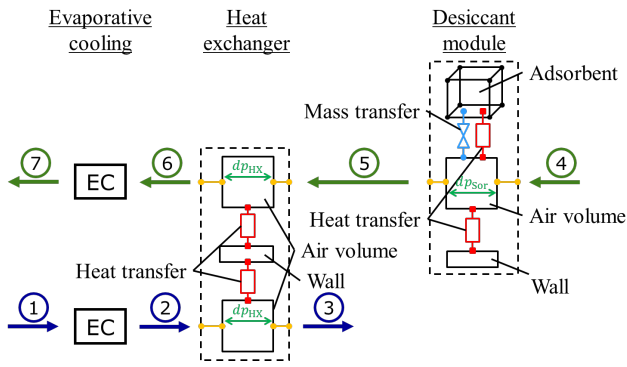


Figure 3. Structure of the dynamic model: Evaporative cooling (left), heat exchanger (middle) and desiccant module (right). The air flows correspond to Figure 2.

the average cooling power during a bus ride $\bar{Q}_{cooling}$ to the mass of the system m

$$SCP = \frac{\bar{Q}_{cooling}}{m} \quad (12)$$

The mobile coefficient of performance (COP_{mobile}) measures the efficiency by the ratio of the average cooling power $\bar{Q}_{cooling}$ to the average electric power consumption $\bar{P}_{battery}$ during the busride:

$$COP_{mobile} = \frac{\bar{Q}_{cooling}}{\bar{P}_{battery}} \quad (13)$$

$\bar{P}_{battery}$ is required only for ventilation in the adsorption-based AC system.

3.4 Design parameters

As design case, we choose a hot summer day in Germany with ambient temperature $T_{ambient} = 30^\circ\text{C}$ and humidity $\phi_{ambient} = 50\%$.

The system is optimized for $n_p = 8$ design parameters p with lower and upper bounds: 5 heat exchanger design parameters, 2 adsorber parameters and the ratio between bus and ambient air mass flow rates. All design parameters are listed in Table 1.

4 Results

In order to properly assess the benefit of a genetic optimization algorithm, we first introduce the results of a simple full factorial design as benchmark. In Section 4.2, we evaluate the development of the Pareto solutions produced by *gamultiobj()* for increasing generation numbers regarding convergence and diversity. Finally, we discuss the case study results.

Table 1. Design parameters p of adsorption air-conditioning system

Design parameter	LB	UB	Unit
$length_{HX}$	0.1	2	m
$height_{HX}$	0.1	2	m
$width_{HX}$	0.1	2	m
$duct\ height_{bus}$	0.001	0.006	m
$duct\ height_{ambient}$	0.001	0.006	m
$diameter_{sorber}$	0.0005	0.008	m
$height_{adsorber}$	0.4	0.08	m
$\frac{\dot{m}_{ambient}}{\dot{m}_{bus}}$	0.5	5	-

4.1 Full factorial design

A full factorial design evaluates all possible combinations of each n_p factors or parameters and its k assigned levels or discrete states. If we assign to each parameter the same number of discrete states, the number of required experiments or simulations n_{sim} amounts to

$$n_{sim} = k^{n_p} \quad (14)$$

In our case study, we chose to assign to each of our $n_p = 8$ parameters its upper and lower bounds a and b and the arithmetic average as possible discrete states:

$$p = \left[LB, \frac{LB+UB}{2}, UB \right] \quad (15)$$

Therefore, $3^8 = 6561$ simulations in total are required. Since a single simulation of our model takes about 8 s of computation time, a full factorial design amounts to 14.5 h CPU-time in total. The results of all feasible solutions with positive values for both objectives are displayed in Figure 4 with the final Pareto solutions highlighted in red.

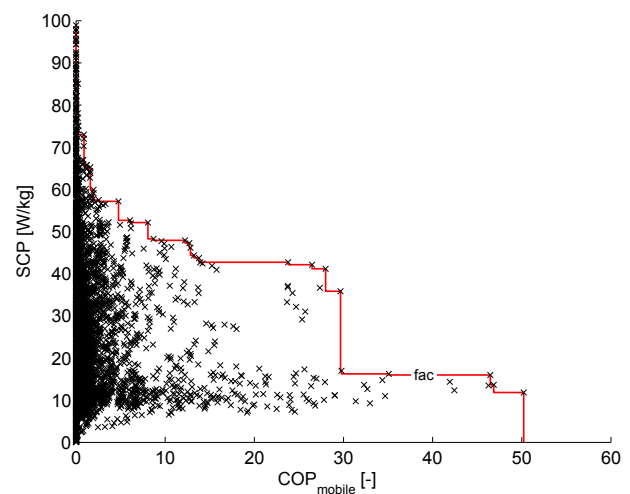


Figure 4. All data points of full factorial design (6561 simulations) and the obtained Pareto front (red line).

From the 6561 simulations, we receive 29 Pareto solutions within the maximum range of $COP_{mobile,max} = 50.2$ and $SCP_{max} = 98.9 \text{ W kg}^{-1}$. The Pareto solutions are not equidistantly distributed on either dimension. This results in occasional spacious gaps between the objective values of adjacent Pareto-solutions. If the designer aims to reach a certain benchmark or minimum limit for one objective, those gaps might force the designer to accept significant losses on the other objective.

4.2 Genetic algorithm

The Pareto solutions for selected generations of the genetic algorithm are displayed in Figure 5 in comparison to the Pareto solution of the full factorial design. Each generation consists of 164 simulations and therefore represents the equivalent computation time of approximately 2.5% of the complete full factorial design with 6561 simulations.

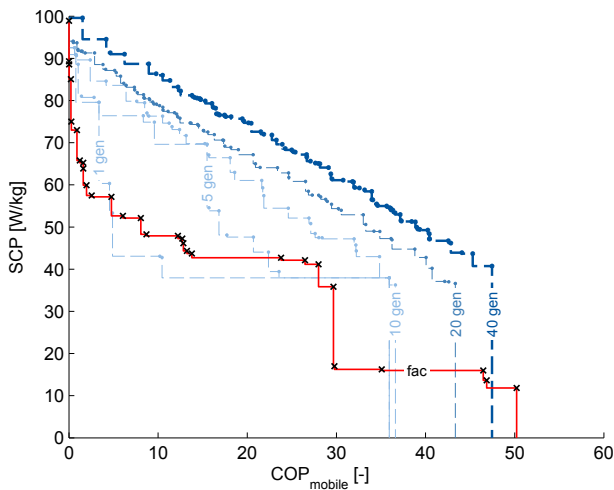


Figure 5. Development of Pareto solutions by the genetic algorithm (in shades of blue) compared to Pareto solutions by a full factorial design with the same number of simulations: $3^8 \approx 164 \cdot 40 = 6560$.

After 40 generations, the genetic algorithm produced 80 Pareto solutions within the maximum range of $COP_{mobile,max} = 48.0$ and $SCP_{max} = 99.6 \text{ W kg}^{-1}$. The range of Pareto solutions is comparable for the full factorial design and the genetic algorithm, however, the genetic algorithm produces more than twice as many solutions, which are more evenly distributed. Also, the maximum range of every objective increases for each generation, which supports the fact that the genetic algorithm favors diverse solutions and minimizes the risk of running into local optima. Thus, the designer has a higher resolution of solutions to choose from.

As predicted in Section 2.2, the genetic algorithm converges to relatively good solutions in early stages. The Pareto solutions of the genetic algorithm outperform all but 3 of those of the factorial design after 5 generations already. This corresponds to 12.5% of the respective

CPU-time of all factorial design simulation runs. Despite the fact that the initial guess is selected randomly, the algorithm converges quickly and can therefore be considered robust to initial conditions. To study its robustness with respect to the initial guess, the genetic algorithm was started several times using different initial populations leading to slightly different evolution paths and final Pareto-solutions respectively.

The genetic algorithm produces better Pareto solutions than the factorial design with generally higher objective values. The evolution of the Pareto-front over each generation implicates that the range of the solution space S can continue to expand. If the designer is interested in more “extreme” solutions, we suggest to continue the algorithm for additional generations.

In summary, the genetic algorithm can be favored over the factorial design regarding robustness, convergence, Pareto solution value quality and quantity. By implication, the necessary computation time can be significantly reduced to obtain comparable results.

4.3 Case Study Results

To interpret the optimization results, a benchmark is needed. As benchmark, we regard a conventional compression chiller for busses with a fixed specific cooling power $SCP = 100$ and a fixed coefficient of performance $COP_{mobile} = 1.9$ (Spheros GmbH, 2015). Compared to this conventional system, the optimization results of the adsorption air-conditioning system show a trade-off between SCP and COP_{mobile} . This trade-off enables the designer to choose solutions with high COP_{mobile} values and still reasonable SCP values. For example, the design point $COP_{mobile} \approx 20$ and $SCP \approx 80 \text{ W kg}^{-1}$ allows the designer to reduce the needed battery capacity by a factor of 10 while only losing 20% of specific cooling power. Which point is taken as optimal depends on the specific application; in particular, weight limitations preference for high SCP , whether battery costs preference for high COP .

The “smoothing” of the genetic algorithm in later iterations (see Section 2.2) has two consequences: First, adjacent Pareto solutions are more similar. Second, the number of produced Pareto solutions within the feasible solution range S is higher than with a full factorial design. These facts strongly favor the genetic algorithm over the full factorial design when analyzing parameter-objective correlations $O_j = f(p_i)$. One such correlation is displayed in Figure 6. In Figure 6 we analyze the correlation of the heat exchanger mass flow ratio as the design parameter $p_i = \dot{m}_{ambient}/\dot{m}_{bus}$ and the mobile coefficient of performance as the objective $O_j = COP_{mobile}$.

The Pareto solutions of the genetic algorithm (black curve) have a much higher resolution than those of the full factorial design (red), which enables the designer to identify correlations more easily and adjust the system’s performance more precisely.

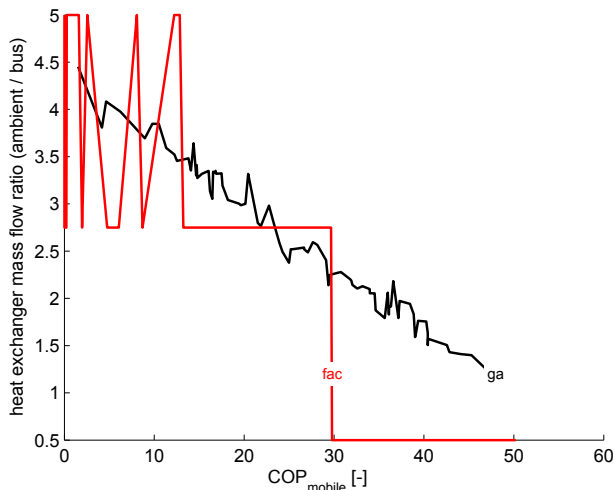


Figure 6. Change of the proposed heat exchanger mass flow ratio $\dot{m}_{\text{ambient}}/\dot{m}_{\text{bus}}$ for Pareto solutions of the genetic algorithm with ascending $\text{COP}_{\text{mobile}}$ values.

The correlation $\text{COP}_{\text{mobile}} = f(\dot{m}_{\text{ambient}}/\dot{m}_{\text{bus}})$ can be physically explained as followed: A higher ambient mass flow rate (higher $\dot{m}_{\text{ambient}}/\dot{m}_{\text{bus}}$) increases the amount of transferred heat, resulting in higher average cooling power \bar{Q}_{cooling} (higher SCP). As a trade-off, the effort for ventilation increases as well and raises the average electric power consumption \bar{P}_{battery} . The latter aspect dominates. In consequence, the mobile coefficient of performance $\text{COP}_{\text{mobile}} = \bar{Q}_{\text{cooling}}/\bar{P}_{\text{battery}}$ decreases for an increasing heat exchanger mass flow ratio $\dot{m}_{\text{ambient}}/\dot{m}_{\text{bus}}$. Other design parameters p_i can be analyzed in the same way.

5 Summary

In this paper, a procedure to connect Modelica models developed in Dymola to MATLAB's optimization toolbox is presented. The non-linear, dynamic process model is evaluated as a black box and the simulation results are loaded in MATLAB using Modelica script files. The framework is illustrated using a genetic algorithm to optimize the design of an adsorption air-conditioning system. The obtained optimization results are compared to a full factorial design: The optimization procedure outperforms the full factorial design regarding simulation time, solution diversity and objective values.

6 Acknowledgment

This work is funded by the Excellence Initiative of the German federal and state governments.

References

- Charles Audet and Luís Nunes Vicente. Derivative-Free Optimization: Theory and Practice, 2008.
- Uwe Bau, Franz Lanzerath, Manuel Gräber, Heike Schreiber, Niklas Thielen, and André Bardow. Adsorption energy systems library - Modeling adsorption based chillers, heat pumps, thermal storages and desiccant systems. In *10th International Modelica Conference, Lund, Sweden*, Linköping Electronic Conference Proceedings, pages 875–883. Linköping University Electronic Press, 2014.
- Uwe Bau, Heike Schreiber, Franz Lanzerath, and André Bardow. Adsorption-based air-conditioning for battery-driven electric busses. In *24th IIR International Congress of Refrigeration*, 2015.
- Christian Blum and Andrea Roli. Metaheuristics in Combinatorial Optimization: Overview and Conceptual Comparison. *ACM Computing Surveys*, 35(3), 2003.
- Francesco Bottiglione, Tommaso Contursi, Angelo Gentile, and Giacomo Mantriota. The fuel economy of hybrid buses: The role of ancillaries in real urban driving. *Energies*, 7(7):4202–4220, 2014. ISSN 1996-1073. doi:10.3390/en7074202.
- Kalyanmoy Deb. *Multi-Objective Optimization using Evolutionary Algorithms*. John Wiley & Sons, Ltd, Chichester, 2001. ISBN 0-471-87339-X.
- Karin Dietl, Stephanie Gallardo Yances, Anna Johnsson, Johan Åkesson, Kilian Link, and Stéphane Velut. Industrial application of optimization with Modelica and Optimica using intelligent Python scripting. In *10th International Modelica Conference, Lund, Sweden*, Linköping Electronic Conference Proceedings, pages 777–786. Linköping University Electronic Press, 2014. doi:10.3384/ECP14096777.
- Manuel Gräber, Kai Kosowski, Christoph Richter, and Wilhelm Tegethoff. Modelling of heat pumps with an object-oriented model library for thermodynamic systems. *Mathematical and Computer Modelling of Dynamical Systems*, 16(3):195–209, 2010. ISSN 1387-3954. doi:10.1080/13873954.2010.506799.
- Manuel Gräber, Christian Kirches, Hans Georg Bock, Johannes P. Schlöder, Wilhelm Tegethoff, and Jürgen Köhler. Determining the optimum cyclic operation of adsorption chillers by a direct method for periodic optimal control. *International Journal of Refrigeration*, 34(4):902–913, 2011. ISSN 0140-7007. doi:10.1016/j.ijrefrig.2010.12.021.
- Maria Henningsson, Johan Åkesson, and Hubertus Tummescheit. An FMI-Based Tool for Robust Design of Dynamical Systems. In *10th International Modelica Conference, Lund, Sweden*, Linköping Electronic Conference Proceedings, pages 35–42. Linköping University Electronic Press, 2014. doi:10.3384/ECP1409635.
- Dylan Francis Jones, Seyed Keyvan Mirrazavi, and Mehrdad Tamiz. Multi-objective meta-heuristics: An overview of the current state-of-the-art. *European Journal of Operational Research*, 137(1):1–9, 2002. ISSN 03772217. doi:10.1016/S0377-2217(01)00123-0.

Abdullah Konak, David W. Coit, and Alice E. Smith. Multi-objective optimization using genetic algorithms: A tutorial. *Reliability Engineering & System Safety*, 91(9):992–1007, 2006. ISSN 09518320. doi:10.1016/j.res.2005.11.018.

Daniel B. Leineweber, Irene Bauer, Hans Georg Bock, and Johannes P. Schlöder. An efficient multiple shooting based reduced SQP strategy for large-scale dynamic process optimization (Parts I and II). *Computers & Chemical Engineering*, 27(2):157–174, 2003.

Alexandra Lind, Elin Sällberg, Stephanie Velut, Stephanie Gallardo Yances, Johan Åkesson, and Kilian Link. Start-up Optimization of a Combined Cycle Power Plant. In *9th International Modelica Conference, Munich, Germany*, Linköping Electronic Conference Proceedings, pages 619–630. Linköping University Electronic Press, 2012. doi:10.3384/ecp12076619.

Neal A. Pennington. Humidity changer for air-conditioning: US Patent 2,700,537, 1955. URL <http://www.google.com/patents/US2700537>.

Andreas Pfeiffer. Optimization Library for Interactive Multi-Criteria Optimization Tasks. In *9th International Modelica Conference, Munich, Germany*, Linköping Electronic Conference Proceedings, pages 669–680. Linköping University Electronic Press, 2012. doi:10.3384/ecp12076669.

Spheros GmbH. REVO-E Technical Specifications, 2015. URL <http://www.spheros.de/Produkte/Klimaanlagen/Busse-ueber-12m/REVO-E.html>.

Hubert Thieriot, Maroun Nemer, Mohsen Torabzadeh-Tari, Peter Fritzson, Rajiv Singh, and John Kocherry. Towards Design Optimization with OpenModelica Emphasizing Parameter Optimization with Genetic Algorithms. In *8th International Modelica Conference, Dresden, Germany*, Linköping Electronic Conference Proceedings. Linköping University Electronic Press, 2011.

Bogdan Tomoiagă, Mircea Chindriș, Andreas Sumper, Antoni Sudria-Andreu, and Roberto Villafafila-Robles. Pareto Optimal Reconfiguration of Power Distribution Systems Using a Genetic Algorithm Based on NSGA-II. *Energies*, 6(3):1439–1455, 2013. ISSN 1996-1073. doi:10.3390/en6031439.

Michael Wetter. Design Optimization with GenOpt. *Building Energy Simulation*, (21):19–28, 2000.