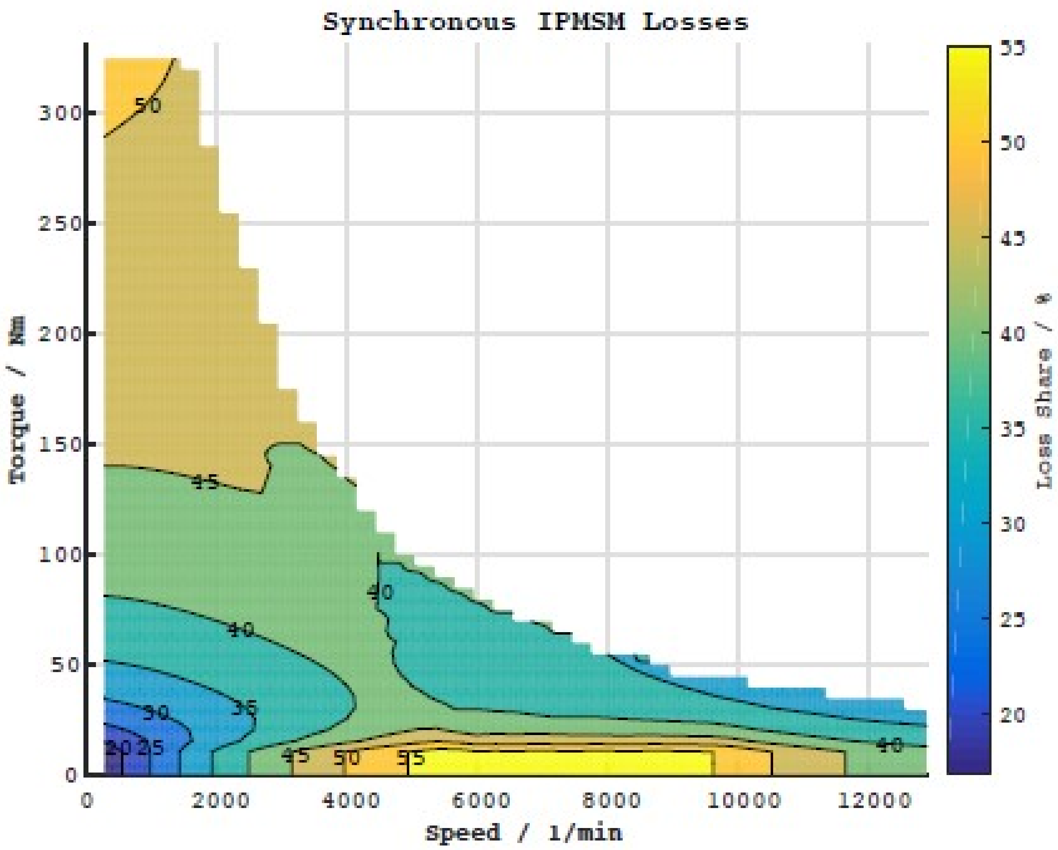
Karlsruhe Institute of Technology

Metaheuristics for online drive train efficiency optimization in electric vehicles

Tilman Apitzsch, Christian Klöffer, Patrick Jochem, Martin Doppelbauer, and Wolf Fichtner

No. 17 | December 2016

WORKING PAPER SERIES IN PRODUCTION AND ENERGY



Metaheuristics for online drive train efficiency optimization in electric vehicles

Tilman Apitzsch, Christian Klöffer, Patrick Jochem*,
Martin Doppelbauer, and Wolf Fichtner

Abstract:

Utilization of electric vehicles provides a solution to several challenges in today's individual mobility. However, ensuring maximum efficient operation of electric vehicles is required in order to overcome their greatest weakness: the limited range. One promising technology is the incorporation of a DC/DC converter into the electric drivetrain. This enables the dynamic optimization of the intermediate voltage level subject to the current driving demand (operating point) of the drivetrain. Moreover, the overall drivetrain efficiency depends on the setup of several drivetrain components' electric parameters. . In order to solve this complex problem for different drivetrain parameter setups subject to the current driving demand dynamically, metaheuristics might provide convincing results. In order to compare the performance of metaheuristics for this task, we adjust and compare the performance of different basic metaheuristics (i.e. Monte-Carlo, Evolutionary Algorithms, Simulated Annealing and Particle Swarm Optimization). The results are statistically analyzed and based on a developed simulation model of an electric drivetrain. By applying the best-performing metaheuristic, the efficiency of the drivetrain could be improved by up to 30% compared to an electric vehicle without the DC/DC- converter. The difference between computing times vary between 30 minutes (for the Exhaustive Search Algorithm) to about 0.2 seconds (Particle Swarm) per operating point. It is shown, that the Particle Swarm Optimization as well as the Evolutionary Algorithm procedures are the best-performing methods on this optimization problem. All in all, the results support the idea that online efficiency optimization in electric vehicles is possible with regard to computing time and success probability.

METAHEURISTICS FOR ONLINE DRIVETRAIN EFFICIENCY OPTIMIZATION IN ELECTRIC VEHICLES

Tilman Apitzsch[a], Christian Klöffer[b], Patrick Jochem[c,*],
Martin Doppelbauer[b], Wolf Fichtner[c]

[a] Industrial Engineering Double Master Program, Karlsruhe Institute of Technology (KIT), Karlsruhe, Germany

[b] Institute of Electrical Engineering (ETI), Hybrid Electric Vehicles, Karlsruhe Institute of Technology (KIT), Karlsruhe, Germany

[c] Institute for Industrial Production (IIP), Energy Economics, Karlsruhe Institute of Technology (KIT), Karlsruhe, Germany

* Corresponding author. Tel.: +49 721 608 44590; E-mail address: jochem@kit.edu

Abstract

Utilization of electric vehicles provides a solution to several challenges in today's individual mobility. However, ensuring maximum efficient operation of electric vehicles is required in order to overcome their greatest weakness: the limited range. Even though the overall efficiency is already high, incorporating DC/DC converter into the electric drivetrain improves the efficiency level further. This inclusion enables the dynamic optimization of the intermediate voltage level subject to the current driving demand (operating point) of the drivetrain. Moreover, the overall drivetrain efficiency depends on the setup of other drivetrain components' electric parameters. Solving this complex problem for different drivetrain parameter setups subject to the current driving demand needs considerable computing time for conventional solvers and cannot be delivered in real-time. Therefore, basic metaheuristics are identified and applied in order to assure the optimization process during driving. In order to compare the performance of metaheuristics for this task, we adjust and compare the performance of different basic metaheuristics (i.e. Monte-Carlo, Evolutionary Algorithms, Simulated Annealing and Particle Swarm Optimization). The results are statistically analyzed and based on a developed simulation model of an electric drivetrain. By applying the best-performing metaheuristic, the efficiency of the drivetrain could be improved by up to 30% compared to an electric vehicle without the DC/DC- converter. The difference between computing times vary between 30 minutes (for the Exhaustive Search Algorithm) to about 0.2 seconds (Particle Swarm) per operating point. It is shown, that the Particle Swarm Optimization as well as the Evolutionary Algorithm procedures are the best-performing methods on this optimization problem. All in all, the results support the idea that online efficiency optimization in electric vehicles is possible with regard to computing time and success probability.

Highlights

- Improving efficiency of an electric drivetrain by incorporating a DC/DC converter
- Setup of a detailed mathematical simulation model of the electric drivetrain
- Adjustment and application of different metaheuristics to this optimization problem
- Performance comparison and identification of the best-performing algorithm
- Significant reduction in computing time to about 0.2 seconds per operation point

Keywords: Electric drivetrain, Metaheuristics, Simulation model, Electric vehicle, Energy efficiency

1 Introduction

Alternative fuels and powertrains for passenger cars are currently in the focus because rising mobility demand, limited and unequally distributed oil resources, fluctuating oil prices, increasing emissions by mobility and challenging policy targets for the transport sector are highly on the political agenda [1]. Whereas alternative fuels solve only some of these challenges, electric vehicles, together with an electricity generation by renewable energy resources such as solar or wind, might cope with all of the mentioned challenges [2] (besides the rising mobility demand).

This is already taken into account by politics in launching several policy instruments for increasing the market share of electric vehicles and to compensate for their higher purchase prices. However, the breakthrough of electric vehicles in the market is still missing and in most countries the market share of electric vehicles is still below 1% [3]. In Europe, for example, there is the Regulation 443/2009 and 333/2014, which limits the average CO_2 emissions of each company's new passenger car fleet to 130 g CO_2/km by 2015 and to 95 g CO_2/km by 2021. For 2030, the target might be set to about 70 g CO_2/km, which can be translated in less than 3 liters fuel per 100 km (or more than 78 miles per gallon) [4]. Assuming a stable car demand, it is doubtful that current combustion engine driven cars can achieve this target – an (at least partial) electrification of the drivetrain seems to be necessary.

Even though the energy efficiency of electric vehicles is already substantially superior ($\eta_{EV} = 60 - 80\%$) compared to conventional vehicles ($\eta_{ICEV} = 15 - 20\%$), a further improvement seems desirable as it increases the limited vehicle range [5]. The efficiency of the electric vehicle depends on the well-to-tank efficiency (i.e. electricity generation, transport, distribution, and charging) as well as the efficiency within the car (tank-to-wheel efficiency). The latter depends on the vehicle drivetrain efficiency (i.e., energy losses from battery to final torque delivered to the wheels) which again depends on a set of drivetrain parameters for different driving demands. The drivetrain parameters are manifold – especially for hybrid electric vehicles (HEV) with internal combustion engines [6], or battery electric vehicles

(BEV) with additional supercapacitors [7]. In our contribution, the focus is on BEV, but an additional DC/DC converter is added to the drivetrain in order to influence the system efficiency by altering the voltage level of the high voltage system (**Fig. 1**). This directly influences the operation and losses of the electric motor and inverter [8, 9]. Besides the possibility to influence the voltage of the high voltage system, the DC/DC converter produces additional losses and causes higher costs of the whole electric drivetrain. The main idea of the proposed approach is to compensate or even undercut the additional losses of the DC/DC converter with reduced losses of the inverter and the electric motor. Furthermore, additional costs can be avoided if the reduction of the energy consumption leads to a smaller battery.

The relationship between efficiency of the drivetrain and the voltage level, is however not static and depends various drivetrain parameters. There is already some literature on this static relationship [10-12], however, a comprehensive analysis of dynamic drivetrain parameter setups and an overview of different metaheuristics to solve this problem is not given.

In the following, we analyze the applicability of different simple metaheuristics (i.e. Monte-Carlo, Evolutionary Algorithms, Simulated Annealing and Particle Swarm Optimization) for providing an online optimization for operating points of different vehicle speed patterns of a BEV with a DC/DC converter. We are focusing on the complex correlation between voltage level in the high voltage system of the vehicle and the efficiency of the drivetrain based on a mathematical model of the electric drivetrain. The different metaheuristics are adapted and partly extended to the considered problem and exemplarily applied for the Highway Fuel Economy Test Cycle (HWFET) and New European Driving Cycle (NEDC). For final application the metaheuristics should be further developed. Results show the main general advantages of the metaheuristics and are given for the success probability of the result (reliability) and computing time (applicability for online optimization within the vehicle). Consequently, the objective of this paper is to identify the best-performing metaheuristic type for our problem.

The structure of the paper is the following. After a literature review we specify our problem in section 3 by outlining the mechanical vehicle, the electric drivetrain, and the final optimization model. In section 4 we introduce the applied metaheuristics before the experimental settings and the corresponding performances for each metaheuristics are given in section 5. A critical assessment and final conclusions complete our contribution in section 6.

2 Literature Review

As a basis for drivetrain efficiency optimization it is necessary to accurately set up a mathematical model of drivetrain components and their respective power losses. There are several publications that describe such models for electric vehicles [e.g. 13-16]. The major aim of such electric drivetrain power loss models is the ability to simulate the behavior of an electric vehicle in operation. Based on these simulation models it is possible to analyze the drivetrain power flow. Past research deals mainly with (plug-in) HEV [e.g. 12-14, 18-20, 23-26, 29] or electric busses [8,27,28]. Some studies focused on minimizing total operation costs or optimal component sizing [e.g. 8, 17, 18, 27,28] rather than on increasing the efficiency of the electrical drivetrain during operation.

From an methodological point of view the literature on optimizing the energy management strategies of HEV might be classified in two mayor groups: optimization based and rule based [9,21,30,31]. While the first group leads to an academically interesting global optimum, the second should result in “near-optimal” solutions with significantly fewer computing time in order to allow real-time applications. This property makes it interesting for the automotive industry and is already widely applied in many studies on HEV [30].

We identified five scientific contributions, which are closely related to our research focus. Firstly, Xi et al. [9] are also applying a real-time capable optimization of a DC/DC converter in an electric drivetrain. However, they do not compare different metaheuristics, but focusing on the basic interrelations of speed and efficiency of in-wheel motors combined with a DC-DC converter. Their analyzed optimizing strategy is suitable for real-time applications, too. Secondly, Serrao et al. [29] compare three methodologies for optimizing the energy management (i.e. Dynamic programming (DP), Pontryagin’s minimum principle (PMP), and equivalent consumption minimization strategy (ECMS)) – however for an HEV drivetrain. The corresponding gains in fuel economy of these three different approaches are calculated for three different driving cycles. As a result, the differences between these three optimizing approaches are rather marginal (i.e. 0-2%). Thirdly, Tenner et al. [11] use a technical optimization algorithm based on the Monte Carlo approach and which they test in six different driving cycles. During inner-city, their approach leady to an efficiency gain of up to 45% compared to the usual operation strategy. On high speeds, the DC/DC converter might even lead to disadvantages. Fourthly, Travão et al. [16] analyze a BEV with two batteries and split their energy management algorithm, which is based on simulated annealing, in two parts: strategic and action planning. They approve their strategy in five driving cycles and come to satisfying results for all driving cycles. Fifthly, Pohlenz [10] mentioned in his comprehensive work on including an additional DC/DC converter to the drivetrain also the application of different simple optimization algorithms. He also compares their results on a general level, but his research is still on a more basic level and he draws no final conclusion of his comparison to give further general recommendations

Concluding, this paper aims to fill the gap in literature and compares results from different problem-invariant metaheuristics [32,33] for electric vehicle drivetrains and discuss the results in order to highlight the main advantages and disadvantages of these metaheuristics.

Heuristic optimization methods allow a real-time optimization of complex problems and achieve good (but not optimal) results. They are also applied for other energy systems such as smart homes [34] electric vehicles and combined heat and power units [35] or electric fleets [36]. There have been many efforts in the past regarding general guidelines for comparing the performances of optimization algorithms [37,38]. In order to draw correct conclusions about the best-performing algorithm it is important to monitor decisive performance measures. It is therefore essential to conduct comprehensive statistical analysis and reporting rather than just checking the best solutions [39,40]. There are also several papers comparing the general performance of some algorithms considered later in this paper [41,42]. These works focused mainly on benchmark functions.

3 Problem Description and Formulation

3.1 The Simulation Model

A detailed mathematical model of the power flow within an electric vehicle's drivetrain provides the basis for the analysis in this paper. The model comprises four main components: the model user interface, the mechanical vehicle model, the electric drivetrain model and the optimization model. **Fig. 1** illustrates their interfaces and interdependencies. The composition of all components that are mentioned sets up a user-friendly simulation model for loss minimization in an electric vehicle's drivetrain. All parts are implemented in the Matlab™ programming environment, ensuring platform independent use.

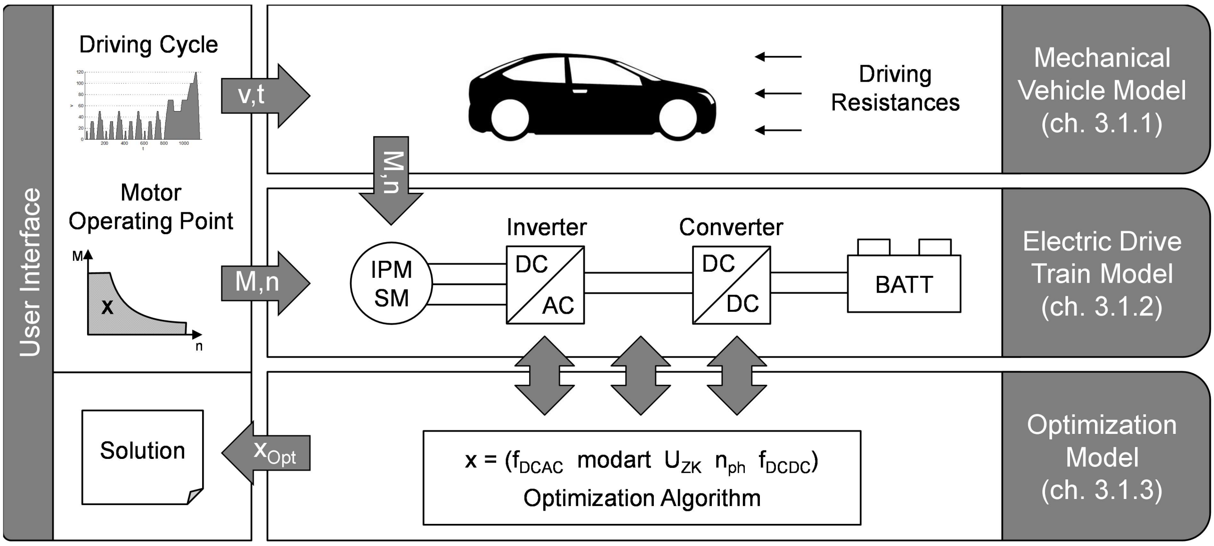


Fig. 1. Graphical representation of the modular simulation model structure

A driving cycle or a motor operating map specifies the drive requirements speed v and time t. While the operating point of the Interior Permanent Magnet Synchronous Motor (IPMSM) provides the motor torque M and speed n requirements directly, the driving cycle delivers a vehicle speed-time-vector, which requires the application of the mechanical vehicle model. All four model parts are briefly explained in the subsequent sections. For reasons of simplicity it is avoided to explain the implemented underlying electrical and mechanical functions, but focus on the step-by-step construction of the objective function (cf. equation 6 below).

Modifications of parameters, model output preferences, and different optimization algorithms in the simulation model can be easily applied via a developed graphic user interface (GUI).

3.1.1 The Mechanical Vehicle Model

The mechanical model of the simulated vehicle has the aim to derive the demanded motor torque and speed based on the model input parameters. The mechanical model's input parameters are specific for the considered vehicle (Smart EV, **Table A. 1**) on the one hand and the drive requirements on the other. The vehicle specifications are static parameters based on the vehicle supposed to be simulated.

The mechanical model simulates longitudinal dynamics. This includes air resistance, acceleration resistance as well as rolling resistance. According to regulatory driving cycles it is assumed that the vehicle only moves straight ahead on a horizontal plane. As a result of that, lateral dynamics can be neglected and the climbing resistance is set to zero. Based on the driving resistances the mechanical model delivers the required motor torque and speed (i.e. operation point, OP) to the electric drivetrain model. For each new OP within the driving cycle the mechanical model is executed again providing on demand input data for the subsequent simulation model layers.

3.1.2 The Electric Drivetrain Model

The electric drivetrain model is the main part of the simulation model and has the aim to systematically minimize the vehicle's energy consumption. It is based on previous work by Pohlenz [10]. As illustrated in **Fig. 1** the electric drivetrain consists of four main components: electric motor, DC/AC-inverter, DC/DC converter and traction battery. These subsystems influence each other. Since the best possible results will not be obtained by an isolated model of each single component a dynamic model of the entire system based on detailed mathematical descriptions is set up [43]. All components depend on variables that can be influenced in a way that the respective component power losses change. Later in this paper it will be the aim to adjust these optimization variables to minimize the overall drivetrain losses $P_{\mathrm{L,Drivetrain}}$ based on a given motor power demand.

$$P_{\mathrm{L,Drivetrain}} = P_{\mathrm{L,IPMSM}} + P_{\mathrm{L,DCAC}} + P_{\mathrm{L,DCDC}} + P_{\mathrm{L,Batt}} \tag{1}$$

The electric motor converts the electric power into mechanical power or the other way around in generator mode. The modeled vehicle utilizes an Interior Permanent Magnet Synchronous Motor (IPMSM), which is the most utilized motor in electric vehicles [44]. The stator and the rotor consist of a laminated core, while the rotor contains embedded permanent magnets. The fundamental advantages of the IPMSM are its high power density, its high efficiency within the basic speed range as well as its good ability to be field-weakened [45]. The IPMSM's power losses can be decomposed in a simplified way into ohmic losses $P_{\mathrm{L},\Omega}$ and iron losses $P_{\mathrm{L,Fe}}$. The ohmic losses are determined by the phase current $I_{\mathrm{S,rms}}$ which is determined through the application maximum torque per ampere

(MTPA) method. The iron losses are caused by hysteresis losses and eddy current losses [46-48].

$$P_{\mathrm{L,IPMSM}} = P_{\mathrm{L},\Omega} + P_{\mathrm{L,Fe}} \tag{2}$$

There are further friction losses appearing inside the IPMSM, which can be neglected since they account for a constant mechanical offset. They cannot be influenced by the electric optimization variables anyway. Once the IPMSM is in operation, the motor speed only depends on the frequency of the input voltage. In order to vary the frequency subject to the motor speed requirement a DC/AC-inverter is utilized. Switching the semiconductors inside the inverter creates additional harmonic motor losses based on harmonic voltages. These are subject to the applied pulse width modulation (PWM) pattern and the inverter's switching frequency. Although the harmonic losses are small, deviating them accurately is a complex process. Due to this disproportionate relation of high complexity (i.e. about 80% of computing time) and low contribution to drivetrain power losses (about 0.8% on average), the harmonic copper losses are approximated subject to Kolar et al. [49]. The harmonic iron losses are neglected because they are significantly smaller than the harmonic copper losses.

It is the task of the drivetrain's power electronics, especially the inverter, to adjust the electrical motor inputs according to the motor's requirement. This is achieved by converting the supplied direct voltage from the energy storage into a three-phase alternating voltage of variable frequency and amplitude. A state-of-the-art bidirectional self-commutated three-phase bridge inverter for automotive applications is utilized in this model. Based on the selected *modulation pattern* and *switching frequency*, the inverter's legs are switched and blocked differently, resulting in diverse inverter losses [49]. The inverter losses comprise the diode's conduction losses $P_{\mathrm{LC,Diode}}$, the transistor's conduction losses $P_{\mathrm{LC,IGBT}}$ and the inverter's switching losses $P_{\mathrm{L,Switch}}$ [50].

$$P_{\mathrm{L,DCAC}} = P_{\mathrm{LC,Diode}} + P_{\mathrm{LC,IGBT}} + P_{\mathrm{L,Switch}} \tag{3}$$

Integrating a DC/DC converter into the conventional electric drivetrain adds an additional degree of freedom to the drivetrain control, since it is then possible to boost the direct voltage supplied by the battery. This is not state-of-the-art so far. Furthermore, the *converter's switching frequency* and its *number of connected phases* can be adjusted in order to influence the converter's power losses. The input voltage of the inverter is called *intermediate circuit voltage* U_{IC}. It affects the operation performance of the IPMSM significantly and can therefore reduce the drivetrain power losses especially at high vehicle speeds. For detailed descriptions of the DC/DC converter's advantages the reader is referred to Tenner et al. [11], Schoenen et al. [51] and Klöffer [52]. The utilized converter is a multi-phase synchronous converter. The converter's power losses $P_{\mathrm{L,DCDC}}$ depend on the number of connected phases n_{ph} and can be decomposed into the basic electric component's losses: conduction losses P_{LC}, switching losses of the transistors $P_{\mathrm{LS,IGBT}}$ and the diodes $P_{\mathrm{LS,Diode}}$ [53], as well as inductor and capacitor losses $P_{\mathrm{L,Inductor}}$ [54,55] and $P_{\mathrm{L,Cap}}$.

$$P_{\mathrm{L,DCDC}} = n_{\mathrm{ph}} \cdot \left(P_{\mathrm{LC}} + P_{\mathrm{LS,IGBT}} + P_{\mathrm{LS,Diode}} + P_{\mathrm{L,Inductor}}\right) + P_{\mathrm{L,Cap}} \quad (4)$$

A battery serves as the energy storage of the drivetrain. In the presented case a Lithium-Ion battery comprising SAFT VL7P cells is utilized. The battery system is modeled as a direct voltage supply with variable internal resistance R_{i} subject to the battery output power P_{batt}. The battery losses $P_{\mathrm{L,Batt}}$ can then be derived from the battery current I_{Batt}. The nominal battery voltage is set to 250 V.

$$P_{\mathrm{L,Batt}} = R_{\mathrm{i}}(P_{\mathrm{batt}}) \cdot I_{\mathrm{Batt}}^2 \quad (5)$$

The mechanical vehicle model and the electric drivetrain model serves as a basis for the optimization model which is presented in the following section.

3.1.3 The Optimization Model

The objective function of the optimization problem is defined as the total drivetrain power losses at a given OP within the driving cycle and needs to be minimized. It is a complex function of a pre-defined parameter set (**Table A. 1**) and the optimization vector x. The following five variables influence the drivetrain power losses significantly and set up the optimization vector x: inverter's switching frequency f_{DCAC}, inverter's modulation pattern $modart$, intermediate circuit voltage U_{IC}, converter's number of connected phases n_{ph} and converter's switching frequency f_{DCDC}.

$$min \;\; P_{\mathrm{L,Drivetrain}} = f(paramSet, x) \quad (6)$$

$$x = \left(f_{\mathrm{DCAC}};\, modart;\, U_{\mathrm{IC}};\, n_{\mathrm{ph}};\, f_{\mathrm{DCDC}}\right) \quad (7)$$

The restricted domains and dimensions of the optimization variables are specified below. All variables are encoded as integer values. The domain bounds are chosen due to physical hardware limits and the lower bound of the switching frequencies are chosen for acoustic reasons due to human audibility.

$$5 \le f_{\mathrm{DCAC}} \le 15 \;\text{ in kHz} \quad (8)$$

$$1 \le modart \le 3 \quad (9)$$

$$U_{\mathrm{Bat}} \le U_{\mathrm{IC}} \le 400 \;\text{ in V} \quad (10)$$

$$1 \le n_{\mathrm{ph}} \le 3 \quad (11)$$

$$5 \le f_{\mathrm{DCDC}} \le 15 \;\text{ in kHz} \quad (12)$$

From the underlying equations for the power loss computation it follows that the given objective function $P_{\mathrm{L,Drivetrain}} = f(paramSet, x)$ is high-grade nonlinear, discontinuous and non-convex. Given this setup of the non-linear objective function and all optimization variables being encoded as integer values, the model can be classified as a non-linear integer problem. Hence, deterministic gradient-based methods (e.g. Newton methods) cannot be applied. Deterministic methods that decompose the problem into parts are not reasonable as well (e.g. Dynamic Programming or Branch & Bound). It is of special interest to simulate the interdependent behavior of all drivetrain components. Furthermore, it is the aim to find a solution that did not get stuck in a local optimum. Therefore, it is not reasonable to apply any simple deterministic method.

As an effect of the sequential calculation design of the model, almost all equations rely on each other. Furthermore, there are additional physical constraints, which need to be satisfied to obtain a technically feasible solution. E.g. they constrain a maximum possible modulation factor, minimum semiconductor interlocking times, the converter's coil flux maximum and maximum current densities within the electronic components. These constraints are already inherent. It evolved as the most practical method that each time an inherent constraint is violated the objective function value is automatically penalized with factor 10^9. Hence, the constraints do not have to be explicitly defined by the optimization model to avoid infeasible solutions.

In order to improve the computation time of the objective function several programmatic measures are implemented. All parameters are summarized into a structural set, ensuring simple handling between different programmatic functions and model parts. We implemented the so-called "Vectorization" approach [56] by replacing parallel scalar operations, e.g. for-loops. Besides reducing computation time, the approach allows the drivetrain model to handle complete optimization variable sets. This ability is advantageous for population-based optimization algorithms, which are applied in section 4.

3.2 Solution Approach

Our challenge is to find an approach which allows us to find a (preferably global) optimum in this non-linear integer problem on real-time during driving. The calculation has to be done with minimum calculation effort due to the high utilization of vehicle's control units. Therefore, the optimization is limited if computation capacity is sufficiently available. In time slots when other preferred highly complex control task (e.g. ABS, ESP) are necessary, our energy efficiency optimization is limited to basic operation. The major requirement for an online-optimization is a rapid model execution (< 1 s) and almost immediate provision of the obtained solution. Therefore, Exhaustive Search (ES) procedure which systematically checks each possible solution in the search space would deliver the global optimal solution is not applicable here for two reasons. First, even if only some of the optimization variables' domains are not constricted to integer values, it is impossible to find the optimum. The continuous domains would provide an infinite number of alternatives taking an infinite amount of time to evaluate them. Second, given the presented integer domains, there are about 348,000 solution alternatives. Setting the domains to integer values is meaningful, since a certain variable change is necessary in order to observe power loss changes. If each evaluation takes only 5 ms, almost 30 minutes would be necessary to obtain the final solution. This is too long for our intended online optimization. Alternatively, an offline ES approach can find an optimal solutions for every OP which than could be implemented into the vehicle by a respective map. Thus, the only remaining operation inside the vehicle would be a simple search for the respective optimal value at the current OP from the map. However, there is a significant drawback. The offline incorporation of parameter dependencies, that could be measured online in the future, adds additional dimensions to the model. An OP would then not be characterized by motor torque and speed anymore but also by additional values of the actual online-parameters (e.g. stator resistance and temperatures). The result would be a huge amount of possible OPs (> 10^{12}), each requiring 30 minutes optimization time as described above. Consequently, due to an unpredictable long optimization time and a considerable high number of potential combinations, the offline optimization of the model including online-measured parameters in the future is not feasible. This applies especially in the automotive industry, where shorter development processes can be decisive competitive advantages. In the end, developing a quick optimization algorithm that solves the described optimization problem online seems to be an appealing approach for future vehicles with a dynamically adjustable electric drivetrain.

4 Appropriate Optimization Algorithms

The availability of a vast amount of different optimization algorithms makes it impossible to find the one and only best-performing solution method for the given problem. But still the selection of a basic metaheuristic on the considered problem can help future studies in identifying efficient algorithms.

Consequently, the model is simplified by applying deterministic metaheuristic approaches in order to achieve a (nearly) optimal solution in real-time. As for this complex problem no standard heuristic is salient and no problem-specific heuristic for drivetrain optimization problems could be found in the literature, different heuristics are applied. The application of most metaheuristics is not straightforward and they have to be customized to each specific problem. Therefore, a sensitivity analysis with a representative set of different OPs to obtain knowledge about the problem behavior is applied. The metaheuristic of choice should be able to take this essential problem knowledge into account, which includes:

- The inverter and the synchronous IPMSM are the major loss contributors.
- The optimization variables modulation pattern, intermediate circuit voltage and number of connected converter-phases exert the highest influence (sensitivity) on the objective function value
- There exists one standard solution $x_{\text{bias}} = (f_{\text{DCAC,min}}, modart_{\text{Std}}, U_{\text{Bat}}, n_{\text{Ph,max}}, f_{\text{DCDC,min}},)$ due to the utilized reference system that ensures maximum efficient operation over a wide range of low and moderate vehicle speeds when the DC/DC converter operates in idle mode.

For those reasons, several optimization algorithms are selected trying to solve the presented problem at low computing time. This selection includes the Monte-Carlo Algorithm (MCA) as a simple random-based metaheuristic. It further includes three naturally inspired algorithms: Evolutionary Algorithm (EA) and Particle Swarm Optimization (PSO) as two population-based approaches and Simulated Annealing (SA) as a mix of trajectory-based and population-based approach [57,58]. Each algorithm is implemented in Matlab™. The performance of an algorithm is significantly influenced by its parameter setup [59]. Thus, additional effort is spent on the function to simply adjust the algorithms' parameters. It is necessary to manually identify parameter sets, which ensure a high performance operation of the respective algorithm on the presented problem. This parameter tuning was done by analyzing respective literature [41,57,60-62]. Further, the researcher's optimization experience plays a decisive role in this process and makes in our application comprehensive landscape analysis superfluous. Each of the following sections presents therefore a preselected and tuned parameter set (e.g. predefined starting point) for the further analysis in this paper.

4.1 Monte-Carlo Algorithm

MCA as a stochastic optimization approach is suitable for high-dimensional deterministic problems [63]. The main idea of this method is to randomly explore the search space by generating so-called pseudo-random numbers. The standard random number generator utilized in Matlab™ is the state-of-the-art Mersenne Twister [64]. **Fig. 2** illustrates the MCA that was implemented for this simulation model.

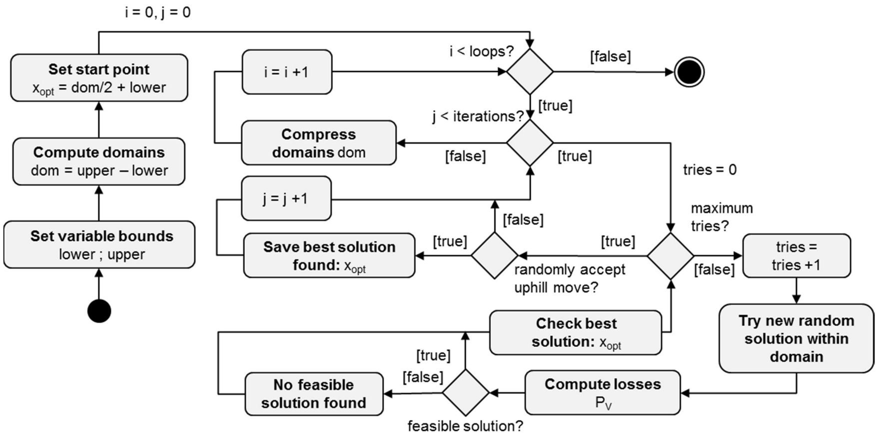


Fig. 2. Outline of the utilized MCA

It basically consists of three loops. The innermost loop (bottom of **Fig. 2**) ensures the generation of a feasible solution and computes the objective function value. The middle loop accounts for the exploration of the search space by random search. Finally the outmost loop exploits the best region found in the middle loop. By incrementally reducing the permitted domain of the optimization variables, a local optimum can be found. The outmost loop is generally known as a hill climbing procedure [65]. Whether the local optimum also represents the global optimum cannot be guaranteed. In order to ensure good performance of the MCA especially for low and moderate vehicle speeds, the high-quality solution x_{bias} is checked immediately after each MCA execution. It is then compared to the actual MCA solution and the best among these two forms the final result of the MCA. **Table 1** presents the pre-selection of three MCA parameter setups after tuning for deeper performance analysis later.

Table 1
Parameter setups of the preselected MCAs

Algorithm Setup	MCA1	MCA2	MCA3
loops	25	50	50
iterations	10	5	5
Acceptance Parameter	0.8	0.3	0.8

4.2 Evolutionary Algorithm

The idea behind an EA is to take the natural evolution as a model for the solution of optimization problems. There have been many new impulses to the research area in the recent years, resulting in various combinations of the approaches [61,66]. The population-based concept can be advantageous because many solutions are considered simultaneously and thus different areas of the search space are analyzed. The implemented EA structure (**Fig. 3**) offers different setup options, which are discussed in the following.

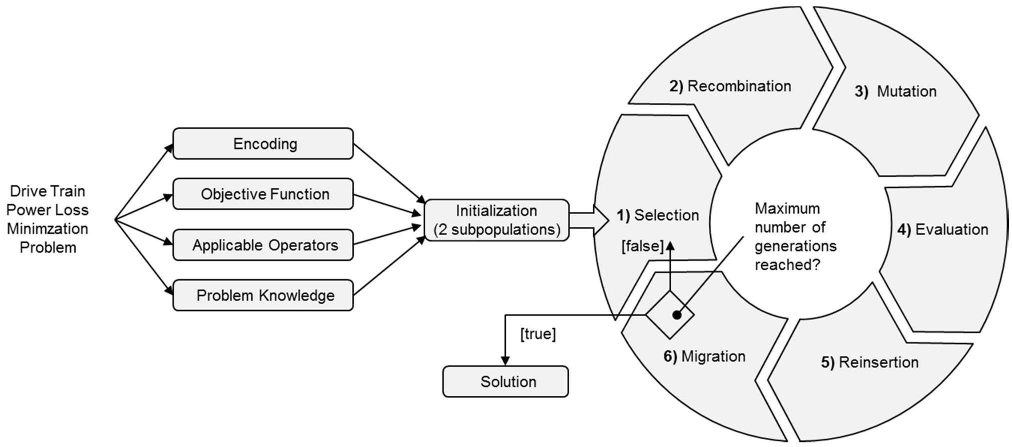


Fig. 3. Outline of the implemented EA approach

A population is at least partly generated randomly. As the use of simple random distributions cannot guarantee the desired level of diversification, it is suggested to incorporate more complex mathematical techniques. The Sobol sequence, a quasi-random method, generates a set of points that fills the search space in a highly uniform manner [67]. This uniformity can be further improved by applying an additional scrambling technique to the Sobol sequence. Matousek [68] suggests a scrambling method, which evolved as a standard and is implemented in our model. Population-based optimization procedures incorporating quasi-random, rather than random numbers, converge much faster [69]. When the gained problem knowledge is further included into the randomly generated initial population, it is biased towards high-quality solutions. Here, including x_{bias} generates the bias. The Gauss mutation operator is then applied to generate slightly modified copies of this individual until a certain share of the population size N_{Ind} is filled with this type of bias-individuals. But as there is only vague knowledge about the optimum, a strong bias should be avoided as it can mislead the search algorithm. In addition, biased initial populations should feature a sufficient diversity, in order to avoid a genetic drift [70].

In order to evaluate the quality of the individuals, two fitness-proportional methods are implemented. The first method is a simple assignment of the objective function value to the respective individual. In the presented problem, the objective function value represents the total drivetrain power losses. Drawbacks of this method are discussed in Goldberg and Deb [71]. The second method is subject to the following formula (13) based on and proved by Bäck and Hoffmeister [72]. It tries to overcome the drawbacks from the first method and scales the fitness linearly. The fitness Fit depends on the selection pressure SP, which is set in advance to a value between 1.5 and 2, the rank Pos of the respective individual within the current population (based on the objective function value) and the population size N_{Ind}.

$$Fit(i) = 2 - SP + 2 \cdot (SP - 1) \cdot \frac{(Pos - 1)}{(N_{\mathrm{Ind}} - 1)} \tag{13}$$

The parent selection procedures can be either deterministic or stochastic. In the case presented here, the $(\mu_{\mathrm{Par}} + \lambda)$ selection is implemented, since it increases the selection pressure [65]. Where λ is the number of offspring and μ_{Par} the number of parents, which are defined based on the pre-specified parameters generation gap $g < 1$ and fertility l (15). Strictly speaking, μ_{Par} and λ are further adjusted to a number that guarantees an equal amount of parents due to the necessity of parental pairs.

$$\lambda = N_{\mathrm{Ind}} \cdot g_{\mathrm{Gap}} \cdot l \tag{14}$$

$$\mu_{\mathrm{Par}} = \frac{2}{l} \cdot \lambda \tag{15}$$

Selection procedures involve the application of a selection function. Three such functions are implemented: stochastic universal sampling (SUS) [65], truncation selection, tournament selection. In the latter case a further parameter called tournament size can be specified by the user and can influence the optimization performance.

The offspring is generated by a pairwise recombination of two parent individuals. This is practically realized by the combination of the parents' optimization variable values. Two representatives of combinatorial operators are implemented: one-point crossover and uniform crossover. In the presented problem case the cutting position for one-point crossover can be placed after the first, second, third or fourth gene, because there are five genes in total. For both operators the fertility of the two parents needs to be set to two. Moreover, two representatives of arithmetic recombination operators are implemented: intermediate recombination and line recombination. Both operators can handle an arbitrary fertility, which has to be specified in advance to the execution of the EA. They also both employ almost the same formula (16), where D denotes the number of optimization variables. When intermediate recombination is applied, a_{d} is generated for each optimization variable. On the contrary, one and the same a_{d} is utilized for all variables when line recombination is applied.

$$x_{\mathrm{d}}^{\mathrm{offspring}} = x_{\mathrm{d}}^{\mathrm{parent1}} \cdot a_{\mathrm{d}} + x_{\mathrm{d}}^{\mathrm{parent2}} \cdot (1 - a_{\mathrm{d}}) \qquad d \in \{1, \dots, D\};\ a_{\mathrm{d}} \sim U(-0.5, 1.5) \tag{16}$$

Each individual within the offspring is faced to a small probability of being mutated, which refers to one or more optimization variable's changes. The mutation probability p_{M} is defined based on findings of Hesser and Männer [73] subject to the current generation t (17). The constants' values are set to $\xi/\varrho = 3$ and $\upsilon = 0.05$ based on empirical runs on the given problem.

$$p_{\mathrm{M}}(t) = \sqrt{\frac{\xi}{\varrho}} \cdot \frac{e^{\frac{-\upsilon \cdot t}{2}}}{N_{\mathrm{Ind}} \cdot \sqrt{D}} \tag{17}$$

The optimum mutation rate of $1/D$ (here 1/5) as proved by Bremermann et al. [74] is widely accepted in the literature [75]. The mutation operator finally executes the actual mutation, which is in our case eider based on the Gauss mutation with respective standard deviations (1.5, 0.5, 10, 0.5, 1.5) set for the optimization variables or the Breeder mutation operator [76]. The mutation range r_{d} and mutation precision K are two further user-specified parameters (18).

$$x_{\mathrm{d}}^{\mathrm{Mut}} = x_{\mathrm{d}} + s_{\mathrm{d}} \cdot r_{\mathrm{d}} \cdot b_{\mathrm{d}} \quad \text{with} \quad s_{\mathrm{d}} \sim U\{-1, 1\}$$
$$b_{\mathrm{d}} = 2^{-u \cdot K} \quad \text{with} \quad u \sim U(0,1) \tag{18}$$

The environmental selection procedure can be done by the same selection functions presented for parent selection. Either way, the elitist rule is applied for environmental selection. It ensures that the very best individual in the population is certain to survive to the next generation.

There exists a huge variety of hybrid approaches for EA which are characterized by the combination of different optimization algorithms and specific tuning [57]. Most of them are specialized on a very specific problem. But one promising concept that evolved from this intense research is the split of one population into several subpopulations [77]. The convincing advantages are that early convergences can be avoided, certain diversity can be maintained by the insertion of individuals from other populations and the number of necessary objective function evaluations can be reduced [78]. Thus, this setup could also be advantageous for sequential computing. This concept's implementation for our problem is limited to two possible subpopulations. Nevertheless, a parameter setup for migration interval and migration rate needs to be defined. The migration occurs always by best-worst replacement. **Table 2** presents the fastest EA parameter setups that could be found after several tuning procedures.

Table 2
Parameter setups of the preselected EAs

Alg. Setup	EA1	EA2	EA3	EA4		EA5	
Populations	1	1	1	2		2	
Subpopulations	-	-	-	Subpop 1	Subpop 2	Subpop 1	Subpop 2
N_{Ind}	25	25	25	25	25	25	25
Initialization	PK	PK	PK	PK	RND	PK	RND
Ranking	FF	FF	FF	FF	OF	FF	OF
SP	1.5	1.5	1.5	1.5	2	1.5	2
g_{Gap}	0.8	0.8	0.8	0.8	0.9	0.8	0.9
Par. Selection	Truncation	Truncation	SUS	Truncation	SUS	Truncation	SUS
l	2	2	2	2	2	2	2
Recombination	I	I	I	I	U	I	U
Mutation	Gauss	Breeder	Gauss	Gauss	Gauss	Gauss	Breeder
r_{d}	-	0.1	-	-	-	-	0.2
K	-	16	-	-		--	20
Env. Selection	Truncation	Truncation	Truncation	Truncation	Truncation	Truncation	Truncation
Migr. Interval	-	-	-	10		10	
Migr. Rate	-	-	-	4		4	
Termination	25 gen	25 gen	25 gen	25 gen		25 gen	

PK: some individuals based on problem knowledge; RND: only random individuals; FF: Fitness Function; OF: Objective Function; I: Intermediate Crossover; U: Uniform Crossover

4.3 Simulated Annealing

SA can be seen as a special simple case of an EA, where the population size is limited to a single individual and the selection procedure is based on an extrinsic parameter [79]. Nevertheless, in some cases SA performance can be superior to EA [80,81]. **Fig. 4** illustrates the steps of the implemented SA procedure. A random neighborhood setup operator is applied to generate a pre-defined number of solutions based on the Gaussian random distribution for each optimization variable, where the mean is kept at the current point of each optimization variable. In order to obtain reasonable neighbors, the standard deviation of the Gaussian distributions is set to the same values as for Gauss mutation described in section 4.2. The algorithm sets up a compressed neighborhood, incorporating random search directions and pre-defined step sizes for each optimization variable from the current point.

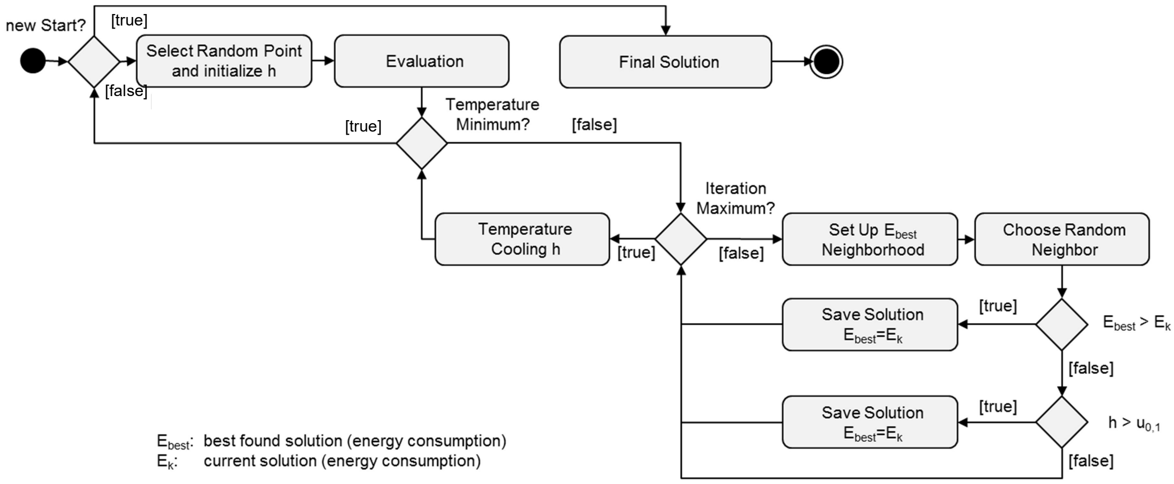


Fig. 4. Outline of the applied SA approach

In order to allow worse solutions to be accepted (uphill move), the acceptance probability h is utilized (19). In the presented case, "worse" corresponds to a total power loss increase. The term E_k refers to the new obtained objective function value, whereas E_best refers to the previous best solution. The suboptimum is accepted if $h > u_{0,1}$ with $u_{0,1} \sim U(0,1)$. In order to ensure that the final solution of an SA execution is not worse than the best solution found during this run, a storage of the best solution found is additionally implemented. This does not influence the algorithm's speed considerably. Table 3 presents the preselected and tuned parameter setups for SA.

$$h = \frac{1}{1 + e^{\frac{E_\mathrm{k} - E_\mathrm{best}}{T}}} \approx e^{\frac{E_\mathrm{best} - E_\mathrm{k}}{T}} \tag{19}$$

Three different SA schedules are implemented which differ with respect to the probability of acceptance: linear, logarithmic and exponential [82]. The linear schedule is represented by a proportional temperature function, as demonstrated in equation (20), where α_T is a constant [82,83]. The logarithmic schedule is represented by function (21), where k_0 is a starting index. An exponential schedule is demonstrated by equation (22), where the parameter c represents the decay rate, specified by the user.

$$T(t+1) = \alpha_\mathrm{T} \cdot T(t) \tag{20}$$

$$T(t) = T_0 \cdot \frac{\ln k_0}{\ln t} \tag{21}$$

$$T(t) = T_0 \cdot e^{-t/c} \tag{22}$$

Table 3
Parameter setups of the preselected SAs

Algorithm Setup	SA1	SA2	SA3	SA4	SA5
New Starts	10	10	10	20	10
Iterations	10	10	10	10	20
Initial Temperature	20	50	100	100	100
Minimum Temperature	1	1	0.001	0.001	0.001
Cooling Schedule	**linear**	**linear**	**exponential**	**exponential**	**exponential**
Neighborhood	random	random	random	random	random
Neighborhood Size	80	80	80	80	80
α_T	0.85	0.85	-	-	-
c	-	-	1.2	1.2	1.2

4.4 Particle Swarm Optimization

The PSO as a member of the swarm intelligence algorithms is the third naturally inspired method [84-86] that will be considered in this paper. A particle flying through the solution space represents a potential solution and follows simple rules [87]. **Fig. 5** illustrates the implemented version of the PSO.

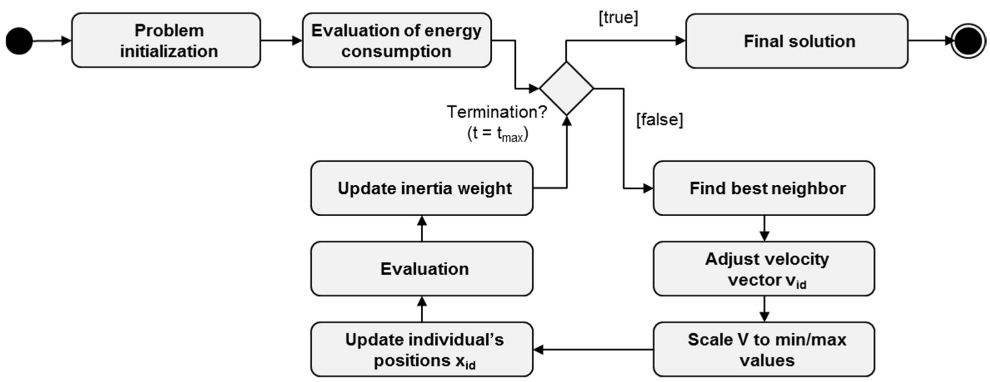


Fig. 5. Outline of the implemented PSO

As described in section 4.2 either a random population based on the scrambled Sobol sequence or a biased population incorporating the gained problem knowledge is generated. In this analysis each particle's neighborhood is simply defined by itself and its left and right neighbor. During the iteration loops the particles' velocity vector v_{id} will be adjusted just as the particles itself according to equations (23) and (24) where y_{i} represents the best position of the individual positions x_{id} found so far by particle i. This procedure is repeated until a maximum time t_{max} is achieved.

$$v_{\mathrm{id}} = \beta \cdot v_{\mathrm{id}} + \alpha_1 \cdot r_1 \cdot (y_{\mathrm{id}} - x_{\mathrm{id}}) + \alpha_2 \cdot r_2 \cdot (y_{\mathrm{bd}} - x_{\mathrm{id}}) \quad (23)$$

$$x_{\mathrm{id}} = x_{\mathrm{id}} + v_{\mathrm{id}} \quad (24)$$

The parameter α_1 is called the cognitive factor and represents the individual's ambition to return to its past successes. The parameter α_2 is the social factor and represents the individual's orientation on the best solutions of its neighbors. The parameter β is called the inertia weight and was suggested by Shi and Eberhart [88]. It is included, because it improves the PSOs local search ability and avoids premature convergence [89]. The inertia weight can be further dynamically adjusted rather than keeping it constant. In this case the linear-decreasing (25) and sigmoid (26) inertia weight adjustments are implemented. A small inertia weight at the end of the search supports the convergence towards an optimal solution with fine-tuning.

$$\beta(t) = \frac{(\beta_{\mathrm{init}} - \beta_{\mathrm{final}}) \cdot (t_{\max} - t)}{t_{\max}} + \beta_{\mathrm{final}} \quad (25)$$

$$\beta(t) = \frac{\beta_{\mathrm{init}} - \beta_{\mathrm{final}}}{1 + e^{q(t - \alpha \cdot t_{\max})}} + \beta_{\mathrm{final}} \quad \text{with} \quad q = 10^{(\log(t_{max}) - 2} \quad (26)$$

Table 4 presents the tuned PSO parameter setups for deeper performance analysis.

Table 4
Parameter setups of the preselected PSOs

Algorithm Setup	PS1	PS2	PS3	PS4	PS5
Population Size N_{Ind}	25	25	25	25	50
Initialization	PK	PK	PK	PK	PK
Inertia Weight β	Constant	Constant	linear	linear	linear
Initial Inertia β_{init}	0.9	0.9	0.9	0.9	0.9
Final Inertia β_{final}	-	-	0.4	0.4	0.4
Cognitive Factor α_1	1.75	1.75	1.75	1.75	1.75
Social Factor α_2	1.75	1.75	1.75	1.75	1.75
Termination	15 gen	25 gen	15 gen	25 gen	15 gen

PK: Initial population contains some individuals based on problem knowledge.

5 Comparison of the Algorithm's Performances

The objective in this section is to identify the algorithm among the presented selection, which features the following criteria best based on Barr et al. [37]: The algorithm should produce solutions of "higher quality" with less computing time than other algorithms. Robustness must be guaranteed, so that the algorithm is less sensitive to differences in the problem characteristic, e.g. adjustments of drivetrain component parameters. Moreover, a "simple" approach with and easy implementation is preferred.

5.1 Experimental Setting

In order to ensure comparability and competitive fairness, all algorithmic test runs are executed on the same computer configuration running lightly loaded.

- Apple MacBook Pro
- Processor: 2.9 GHz Intel Core i7 (1 processor, 2 cores)
- Memory: 8 GB 1600 MHz DDR3
- L2 Cache (per Core): 256 KB
- L3 Cache: 4 MB
- Software: OS X 10.9.4 (13E28)
- Programmer's coding expertise: professional

Furthermore, differences in coding skills, algorithm tuning and effort invested can be neglected since all competing optimization algorithms where programmed by the same person [38]. In order to ensure simple comparability, all algorithms are supposed to run on representative OPs, displayed in **Fig. 6**. The global optimal solutions for these OPs are known from long ES procedure runs. The high-quality bias-solution x_{bias} is implemented in all of the presented algorithms and forms an upperbound for the algorithms' solutions.

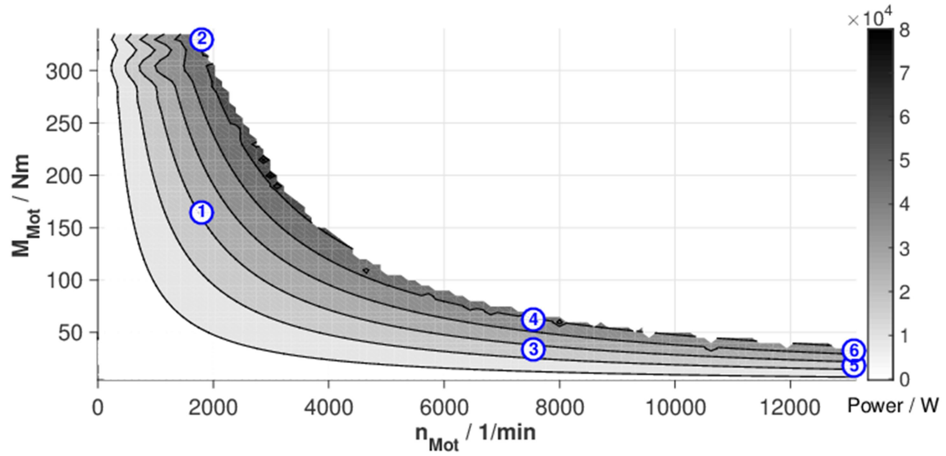


Fig. 6. Motor operating map with representative operating points

Three different performance criteria are taken into account: success probability of the algorithm, total computation time and time to best-found solution. Success probability is defined as the likelihood of the algorithm to obtain a final solution, which is considered to be successful. A solution is considered successful if the obtained power losses $P_{\mathrm{L,Drivetrain}}$ are not more than 1% higher than the global optimal solution (found through ES validation). Total run time is defined as the algorithm execution time prior to termination by its stopping rule. It comprises the complete processing time, including overhead. Finally, the time to the best-found solution indicates the time to produce the best solution. It is the solution, which is finally returned by the algorithm, but not necessarily produced at the end of the total running time. The best solution can be found much earlier. Time to best-found solution measures the convergence speed. As a recommended practice, all time measurements are done by measuring CPU-time and not the real time [37].

5.2 Computational Results

For statistical reasons it is necessary to perform a set of experiments in order to obtain a good representation of the results [40]. Taking the result of one single experiment could worst-case result in false conclusions. Consequently, each algorithm parameter set runs for 100 samples on the presented problem, represented by eight OPs. **Fig. 7** displays the success probability for each algorithm set, represents by the mean value over all samples and OPs. Except MCA, all algorithms perform very well on this criterion, reaching success probabilities over 98%.

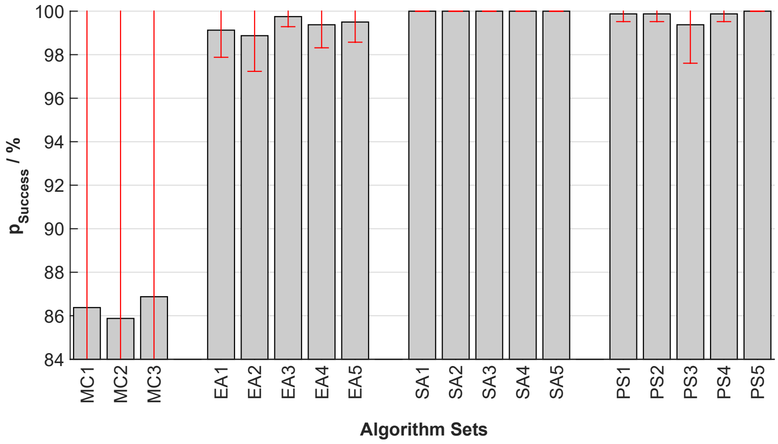


Fig. 7. Success probability comparison

Total run time is an important indicator on the algorithms' practical applicability for an online optimization. **Fig. 8** illustrates boxplot diagrams for the total run time performance of each algorithm. Points are drawn as outliers if they are larger than $q_3 + w_l \cdot (q_3 - q_1)$ or smaller than $q_1 + w_l \cdot (q_3 - q_1)$. q_1 and q_3 are the 25th and 75th percentile of the sample data. The whisker length w_l is set to 1.5 and corresponds to approximately $+/-2.7\sigma$. In order to facilitate clarity, the algorithm sets are plotted in two separate diagrams. In the left diagram it can be seen that even the best-performing SA algorithm requires a total run time of approximately 5 s. Thus, SA is not likely to be considered for quick online-optimization. Although the MCAs reach run times close to 1 s, they are still outperformed by all EAs and PSO procedures. EA4 and EA5 perform worst in the right diagram, because they involve the multiple subpopulations approach. But the advantages of this approach appear to be only exploited when parallel hardware architectures are utilized [77]. The remaining three EA algorithms feature a median total run time between 0.3 s and 0.4 s and are only outperformed by some PSO algorithms. PS1 and PS3 require the shortest running time.

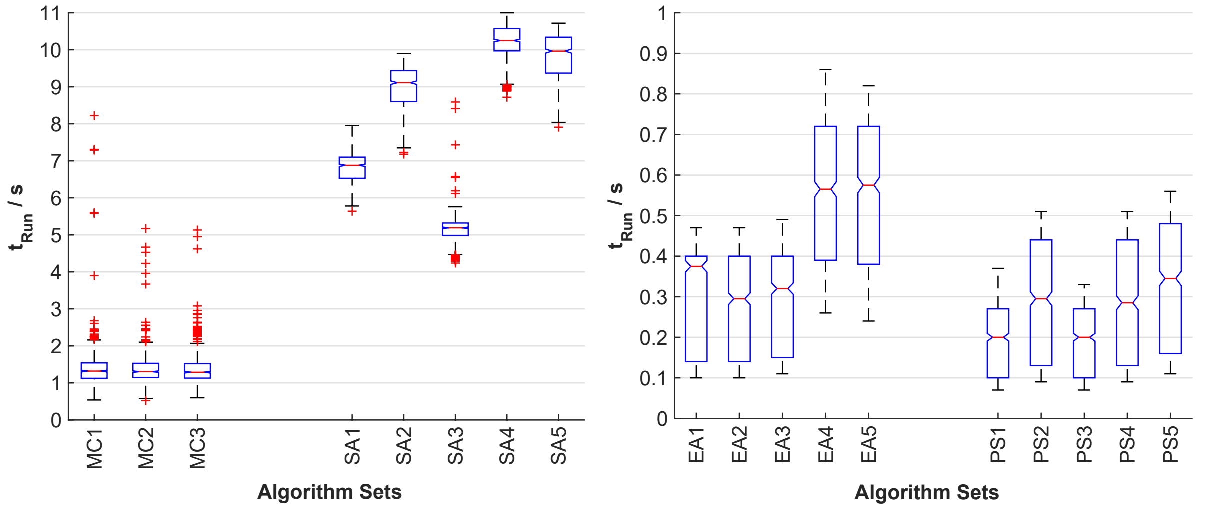


Fig. 8. Total run time comparison

As time to best-found solution measures the convergence speed, it is an indicator for robustness of an optimization algorithm. If the time to find the best solution lies very close to the total run time, the algorithm is faced to a high risk of not converging fast enough before the algorithm termination criterion is triggered. Thus, a certain time gap between total run time and time to best-found solution is a good indicator for algorithm robustness. The general outcome of **Fig. 9** is the higher variance of the sample times compared to the results regarding total run time. The lower this variance, the higher is the algorithm's robustness. It can be seen that algorithms PS1 and PS3 show the best performance on this criterion with regard to both, the median (red line) and variance (whisker) values.

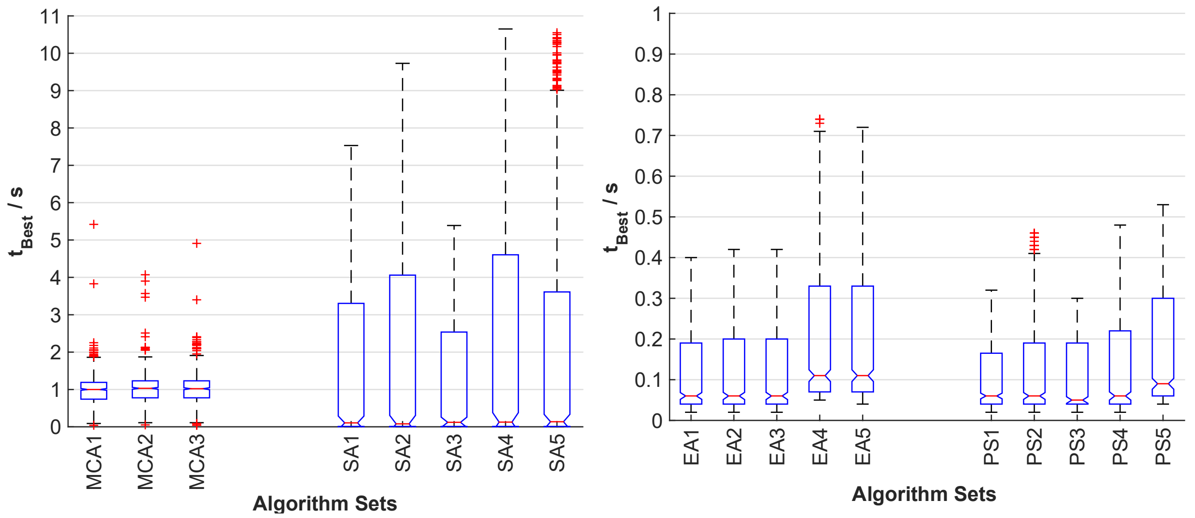


Fig. 9. Time to best-found solution comparison

There is always a key performance tradeoff between solution quality and running time. During the comparison process it is attempted to find the best possible fit of both. **Fig. 10** summarizes the algorithm's performances subject to these three-dimensional reference criteria. Although all SA algorithms feature a success probability of approximately 100%, their speed-performance with regard to median run time is by far the weakest among all compared algorithms. It was attempted to reduce the number of SA iterations to improve the speed-performance. But the outcome was a drop in success probability, which is even worse than bad speed-performance. The MCAs gather around success probabilities between 85% and 90% with very good run time performance. Since success probability is defined as the most substantial decision criterion, this group of algorithms can be further deselected because they are considered as “not successful enough”. In order to gain better insight into the performances of the algorithms, which are located around the origin, **Fig. 11** magnifies this area.

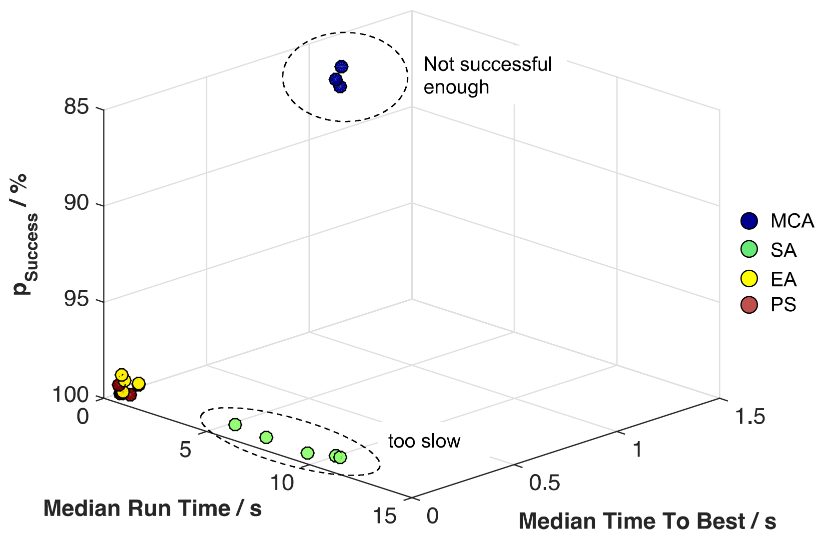


Fig. 10. Multi-criteria comparison of the algorithm performances (full view)

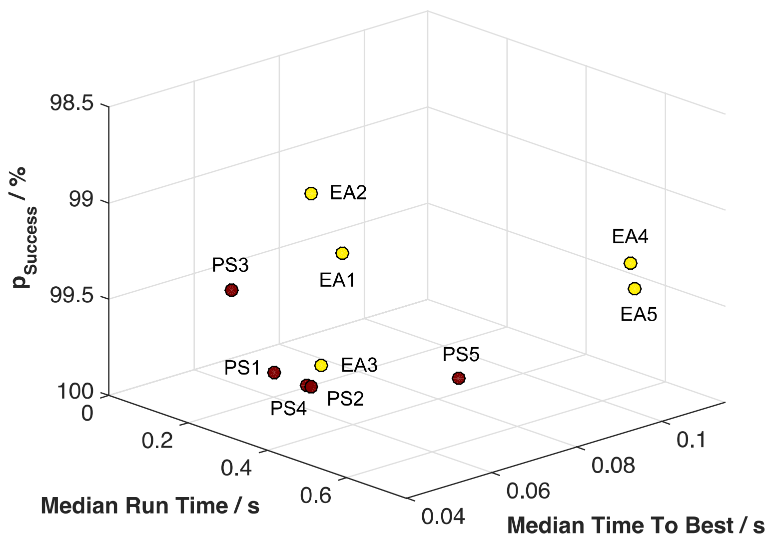


Fig. 11. Multi-criteria comparison of the algorithm performances (magnification)

Based on **Fig. 11**, the Euclidean distance to the origin (0, 0, 100%) of each scatter point is calculated. From this it follows that PS1 is selected as the best-performing algorithm on the presented problem followed by algorithms PS4 and PS2. Consequently, PS1 is considered the optimization algorithm of choice. The application of this optimization algorithm results in a success probability of 99.9%, a mean total run time of 0.186 s and mean time to best-found solution of 0.101s. Due to the normal distribution of the sample values, these mean values can be statistically validated by the Gauss-test [90]. It can be shown, that for a statistical significance of 99.99% the mean runtime of PS1 does not exceed 0.198 s and the mean time to best-found solution does not exceed 0.112 s. **Fig. 12** illustrates the convergence process of the mean best solution quality of all samples and for all OPs. Subsequently, the error bars indicate the standard deviation among the samples. Given the first four OPs, the optimal solution lies very close to x_{bias} because the electric converter operates in idle mode. Since x_{bias} is always inherent in the initial population of the PSO, PS1 finds the optimum immediately and no significant convergence process can be observed.

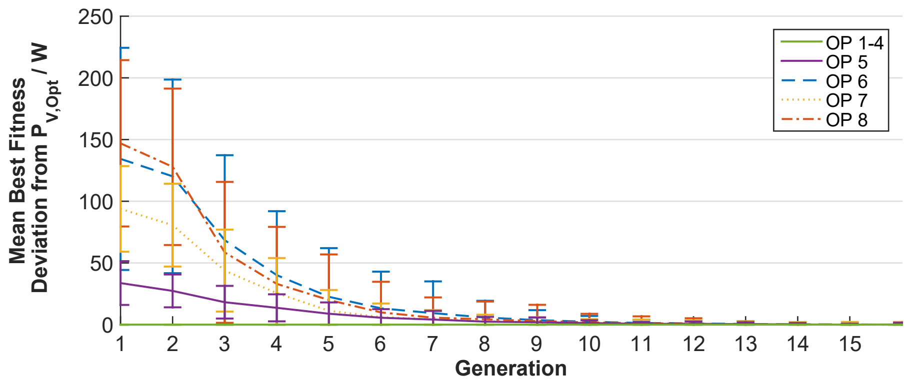


Fig. 12. Convergence process of PS1 for each operating point

5.3 Model Application and Result Validation

Finally, it is reasonable to get a general insight into the PS1 model operation and its effects on the drivetrain efficiency by applying it to the simulation model. The intermediate circuit voltage U_{IC} is adjusted to motor speed except at low motor speeds where the converter operates in idle mode and the intermediate circuit voltage equals the battery voltage. Since the electric vehicle incorporates a fixed transmission, motor speed directly corresponds to vehicle speed. This correlation is shown in **Fig. 13**, where the PS1 operation is applied in the Highway Fuel Economy Test (HWFET) cycle. The converter operates in boost mode at vehicle velocities above 50 km/h. This is a validation example for one optimization variable. The results for all remaining optimization variables are validated respectively by comparing the results to expected behavior.

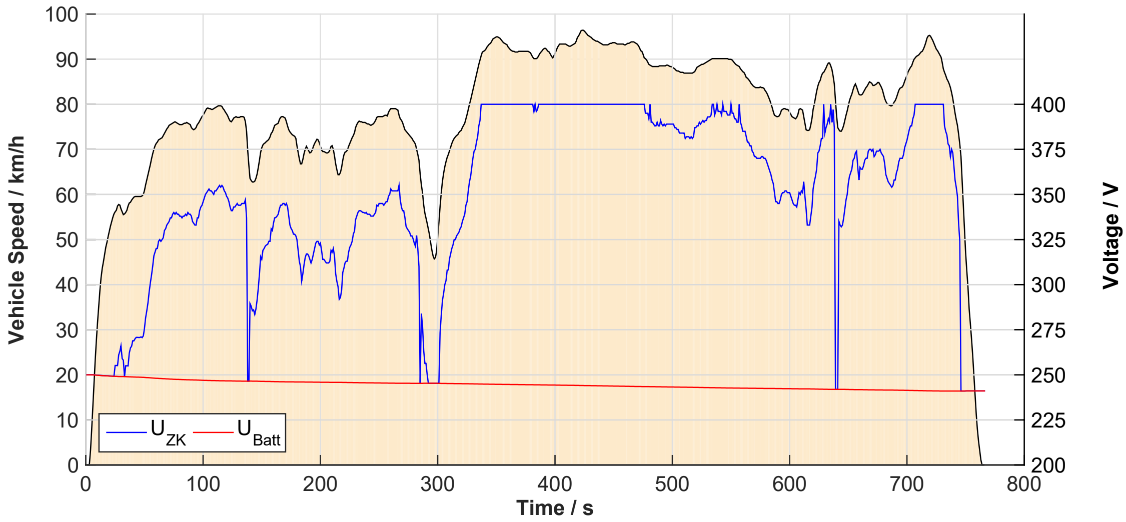


Fig. 13. Optimization results validation (here: intermediate circuit voltage)

Besides the HWFET cycle, we also applied our model to the New European Driving Cycle (NEDC) and identified the corresponding efficiency losses in the considered components (cf. **Fig. 14**). It can be seen that the major power loss contributors are the IPMSM, the DC/DC converter and the DC/AC inverter. If the converter operates in idle mode, then its power losses are considerably small. But in boost mode its losses increase significantly. Moreover, the battery losses and the harmonic IPMSM losses are small and even hardly visible. The cumulated energy losses are illustrated in the right side of **Fig. 14**. Benchmarks are given on the one hand by constantly operating the converter in idle mode and on the other hand by operating the drivetrain without DC/DC converter as a whole component. This validates the initial assumption that it is energetically advantageous to incorporate the component DC/DC converter into the drivetrain topology as it increases the overall drivetrain efficiency especially for high speed drive cycles. The resulting mean drivetrain efficiency reaches 92.95% for HWFET. Hence, energy savings of approximately 30.51% for HWFET can be realized, if the DC/DC converter is incorporated into the optimized drivetrain topology.

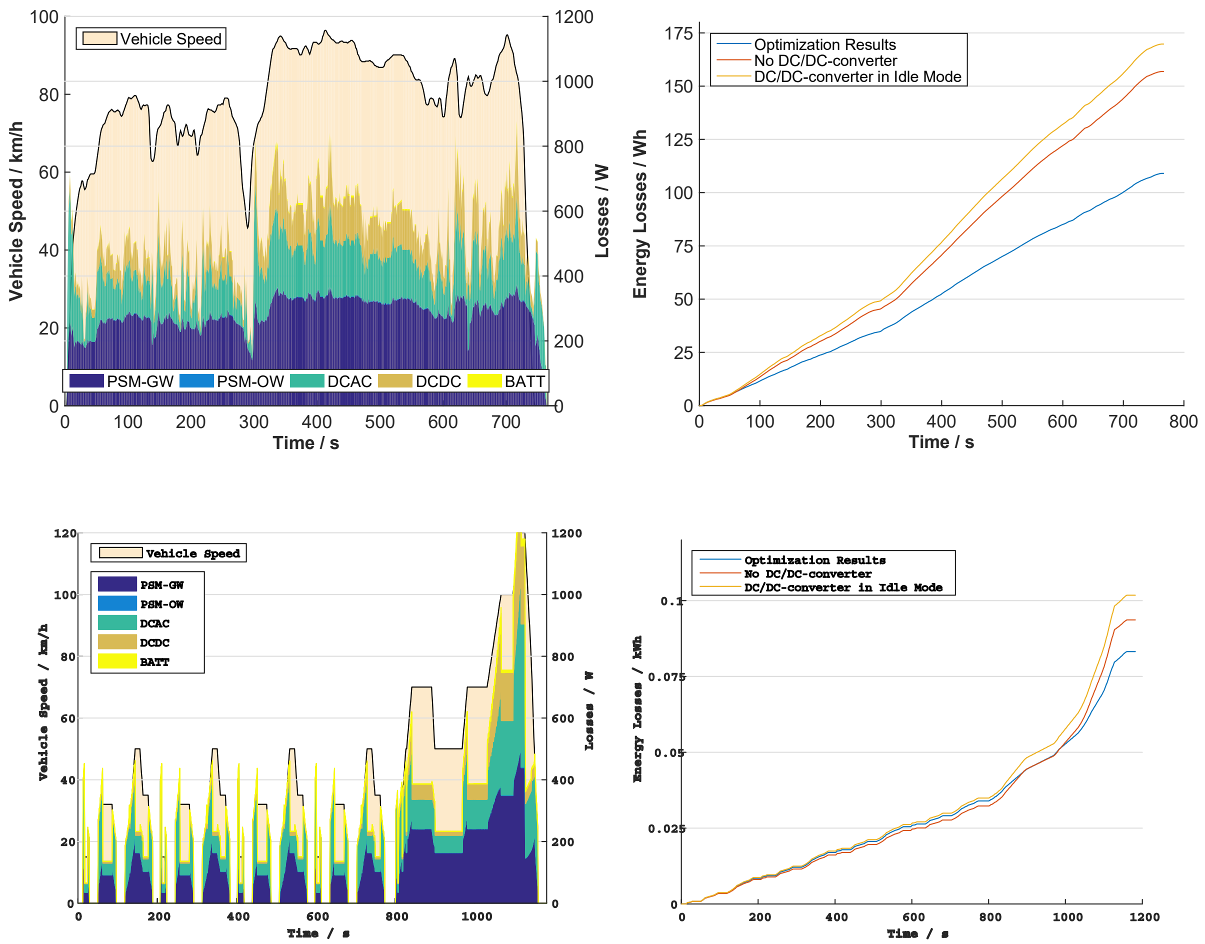


Fig. 14. Drivetrain efficiency results for driving cycles HWFET (above) and NEDC (below)

It should be noted, that the energy savings strongly depends on the voltage of the battery. If the battery voltage (for 100% State of Charge (SOC)) is equal to the maximum inverter voltage there is no benefit of the DC/DC converter. In this case the DC/DC converter leads to additional overall losses until the SOC of the battery decreases [91]. Additionally we would like to mention here that we only considered rather simple (but reliable) metaheuristics in order to identify the main general advantages of these five and robust optimization algorithms. Another (faster but more imprecise) solution for our problem might be to deposit a few hundred representative OPs connected to very easy commands.

6 Conclusion and Further Research

The initial objective was to establish a power loss model of the electric drivetrain as exact as possible and select an appropriate metaheuristic that enables quick and on-demand determination of the optimal drivetrain parameter setup. Based on previous research the drivetrain power loss model was successfully set up and embedded in a convenient simulation model framework.

Our results indicate that the Particle Swarm Optimization (PSO) algorithms outperform the much more complex presented Evolutionary Algorithms (EA). The programming and tuning efforts for PSO are significantly lower than for EAs, due to fewer parameters. Compared to the simple baseline, given by the Exhaustive Search (ES) procedure, the computing time could be reduced from about 30 minutes to less than 0.2 seconds, which enables an online optimization (see section 3.2). This emphasizes the great performance improvement on solving the presented energy efficiency problem by applying the PSO based PS1 algorithm. Nevertheless, especially EA3 can be seen as a high performing alternative optimization algorithm on the presented problem. In direct comparison with the conventional electric drivetrain, the energy savings of the online-optimized drivetrain by PS1 sum up to 30.51% for Highway Fuel Economy Test Cycle (HWFET). An analysis of several optimized parameter values further validated the reliability of PS1.

Based on the great performance improvement mainly two further research approaches are derived. On the one hand the problem of inaccuracies caused by the simplified simulation model can be solved by using a real test-bench. In this case the losses of a parameter set would no longer be calculated with a model but be measured at a real drivetrain. Hence the results would be much more precise. On the other hand it becomes possible to do the parameter optimization online in driving vehicles. The offline incorporation of parameter dependencies like changes in temperature or effects of aging would add additional dimensions to the model. An operating point (OP) would then not be characterized by motor torque and speed anymore but also by values of changing parameters. The result would be a huge amount of possible OPs ($> 10^{12}$). Consequently, due to an unpredictable long optimization time and a considerable high number of potential combinations, the offline optimization would lead to an unrealizable optimization problem. With the results of this paper it is feasible to overcome this drawback. It is now possible to measure changing parameters and to optimize the set of parameters in real-time when a new OP is demanded.

7 Appendix

Table A. 1
Vehicle parameters

Symbol	Value	Symbol	Value
Mechanical Vehicle Parameters			
ρ_{Air}	1.2041 kg/m^3	m_{Car}	900 kg
c_{W}	0.4	c_{R}	0.012
A_{Front}	2.05 m^2	g	9.81 m/s^2
r_{W}	0.25 m	i_{Trans}	10
IPMSM Parameters			
L_{q}	0.5 mH	ψ_{PM}	0.054 Vs
L_{d}	0.2 mH	p	4
R_{S}	12 mΩ	c_{fe}	2.4
c_{Str}	0.00234	κ	1.3
ε	0.5	τ	0.8
Power Electronics Parameters			
U_{D0}	1.0 V	$R_{\mathrm{L,Cu}}$	6.2 mΩ
U_{T0}	1.7 V	α	1.38
r_{D}	2.6 mΩ	γ	2.48
r_{T}	2.6 mΩ	ϑ	2.2
k_{T}	110 mJ / 627 A	V_{Core}	0.54459 dm^3
k_{D}	21 mJ / 627 A	I_{ref}	627 A
$B_{\mathrm{DCDC_{rated}}}$	0.38 T	U_{ref}	900 V
$J_{\mathrm{DCDC_{D_{rated}}}}$	2 A/cm^2	$J_{\mathrm{DCDC_{T_{rated}}}}$	2 A/cm^2
$J_{\mathrm{DCDC_{Cu_{rated}}}}$	3 A/cm^2		

Table A. 2

Model equations: This table intends to give a deeper insight of the model's complexity, discontinuity and non-linearity with regard to applicable optimization algorithms

Equation	Description
1. MECHANICAL VEHICLE MODEL	
$F_{\mathrm{Air}} = \frac{1}{2} \cdot \rho_{\mathrm{Air}} \cdot c_{\mathrm{W}} \cdot A_{\mathrm{Front}} \cdot v^2$	Air resistance of the vehicle
$F_{\mathrm{Acc}} = m_{\mathrm{Car}} \cdot a_{\mathrm{Car}}$	Acceleration resistance of the vehicle
$F_{\mathrm{Roll}} = m_{\mathrm{Car}} \cdot g \cdot c_{\mathrm{R}}$	Rolling resistance of the vehicle
$M_{\mathrm{Mot}} = \frac{(F_{\mathrm{Air}} + F_{\mathrm{Acc}} + F_{\mathrm{Roll}}) \cdot r_{\mathrm{W}}}{i_{\mathrm{Trans}}}$	Mechanical motor torque
$n_{\mathrm{Mot}} = \frac{v_{\mathrm{Car}} \cdot i_{\mathrm{Trans}}}{2\pi \cdot r_{\mathrm{W}}}$	Motor speed
$P_{\mathrm{Mot}} = 2\pi \cdot n_{\mathrm{Mot}} \cdot M_{\mathrm{Mot}}$	Motor power
2. ELECTRIC DRIVETRAIN MODEL	
$P_{L,\mathrm{Drivetrain}} = P_{\mathrm{L,IPMSM}} + P_{\mathrm{L,DCAC}} + P_{\mathrm{L,DCDC}} + P_{\mathrm{L,Batt}}$	Power losses of the complete electric drivetrain
2a. Internal permanent magnet synchronous motor	
$P_{\mathrm{L,IPMSM}} = P_{\mathrm{L},\Omega} + P_{\mathrm{L,Fe}}$	Total power losses of the electric motor
$P_{\mathrm{L},\Omega} = 3 \cdot R_{\mathrm{S}} \cdot I_{\mathrm{S,rms}}^2$	Ohmic power losses of the electric motor
$I_{\mathrm{S,rms}} = \frac{1}{\sqrt{2}} \cdot \sqrt{I_{\mathrm{d}}^2 + I_{\mathrm{q}}^2}$	Root mean square value of the stator current
$M_{\mathrm{Mot}} = \frac{3}{2} \cdot p \cdot \left(\psi_{\mathrm{PM}} \cdot I_{\mathrm{q}} + (L_{\mathrm{d}} - L_{\mathrm{q}}) \cdot I_{\mathrm{d}} \cdot I_{\mathrm{q}}\right)$	Electric motor torque
$I_{\mathrm{d,MTPA}} = \frac{\psi_{\mathrm{PM}}}{2 \cdot (L_{\mathrm{d}} - L_{\mathrm{q}})} - \sqrt{\frac{\psi_{\mathrm{PM}}^2}{4 \cdot (L_{\mathrm{d}} - L_{\mathrm{q}})^2} + I_{\mathrm{q}}^2}$	MTPA-optimized d-component of the stator current
$P_{\mathrm{L,Fe}} = c_{\mathrm{fe}} \cdot \omega^{\kappa} \cdot \psi_0^2 + c_{\mathrm{Str}} \cdot \omega^{\varepsilon} \cdot i^{\tau}$	Iron losses of the electric motor
2b. DC/AC-three-phase Bridge Inverter	
$P_{\mathrm{L,DCAC}} = P_{\mathrm{LC,Diode}} + P_{\mathrm{LC,IGBT}} + P_{\mathrm{L,Switch}}$	Total inverter power losses
$P_{\mathrm{LC,Diode}} = 6 \cdot \left[U_{\mathrm{D0}} \cdot \frac{\hat{I}_{\mathrm{S}}}{2} \cdot \left(\frac{1}{\pi} - \frac{M}{4} \cdot \cos\varphi\right) + r_{\mathrm{D}} \cdot \hat{I}_{\mathrm{S}}^{\,2} \cdot \sqrt{\frac{1}{8} - \frac{M}{3\pi} \cdot \cos\varphi}\right]$	Diode's conduction losses
$P_{\mathrm{LC,IGBT}} = 6 \cdot \left[U_{\mathrm{T0}} \cdot \frac{\hat{I}_{\mathrm{S}}}{2} \cdot \left(\frac{1}{\pi} + \frac{M}{4} \cdot \cos\varphi\right) + r_{\mathrm{T}} \cdot \hat{I}_{\mathrm{S}}^{\,2} \cdot \sqrt{\frac{1}{8} + \frac{M}{3\pi} \cdot \cos\varphi}\right]$	IGBT's conduction losses
$P_{\mathrm{L,Switch}} = 6 \cdot \frac{1}{\pi} + f_{\mathrm{DCAC}} \cdot (k_{\mathrm{T}} + k_{\mathrm{D}}) \cdot \hat{I}_{\mathrm{S}}$	Inverter's switching losses
2c. DC/DC converter	
$P_{\mathrm{L,DCDC}} = n_{\mathrm{ph}} \cdot \left(P_{\mathrm{LC}} + P_{\mathrm{LS,IGBT}} + P_{\mathrm{LS,Diode}} + P_{\mathrm{L,Coil}}\right) + P_{\mathrm{L,Cap}}$	Total converter power losses
$P_{\mathrm{LC}} = U_{\mathrm{D0}} \cdot I_{\mathrm{Diode,avg}} + \mathrm{r}_{\mathrm{D}} \cdot I_{\mathrm{Diode,rms}}^2 + U_{\mathrm{T0}} \cdot I_{\mathrm{IGBT,avg}} + r_{\mathrm{T}} \cdot I_{\mathrm{IGBT,rms}}^2$	Converter's conduction losses
$P_{\mathrm{LS,IGBT}} = f_{\mathrm{DCDC}} \cdot I_{\mathrm{IGBT,avg}} \cdot k_{\mathrm{T}} \cdot \frac{U_{\mathrm{IC}}}{U_{\mathrm{ref}}}$	IGBT's switching losses
$P_{\mathrm{LS,Diode}} = f_{\mathrm{DCDC}} \cdot \frac{Q_{\mathrm{rr}}}{2} \cdot \frac{U_{\mathrm{IC}}}{U_{\mathrm{ref}}} \cdot \frac{I_{\mathrm{Diode,avg}}}{I_{\mathrm{ref}}}$	Diode's switching losses

$P_{\mathrm{L,Coil}} = R_{\mathrm{L,Cu}} \cdot \left[I_{\mathrm{L}}^2 + \frac{I_{\mathrm{L}}^2}{4} \cdot \frac{d_{\mathrm{Cu}} \cdot \left(1 - e^{\frac{2 d_{\mathrm{Cu}}}{\delta} \delta}\right)}{\delta \cdot \left(1 - e^{\frac{2 d_{\mathrm{Cu}}}{\delta} \delta}\right)^2} \right] + \gamma \cdot f_{\mathrm{eq}}^{\alpha - 1} \cdot \left(\frac{\Delta B}{2}\right)^{\vartheta} \cdot V_{\mathrm{Core}} \cdot f_{\mathrm{DCDC}}$	Coil losses comprising copper and core losses
$f_{\mathrm{eq}} = \frac{2}{\Delta B^2 \cdot \pi^2} \cdot \int_0^T \left(\frac{\Delta B^2}{dt}\right) \cdot dt$	Equivalent frequency
$P_{\mathrm{L,Cap}} = R_C \cdot I_{\mathrm{Cap,rms}}^2$	Capacitor losses
2d. Traction battery	
$P_{L,\mathrm{Batt}} = R_{\mathrm{i}}(P_{\mathrm{a}}) \cdot I_{\mathrm{Batt}}^2$	Battery losses
$R_{\mathrm{i}}(P_{\mathrm{a}}) = 1.6 \cdot 10^{-3} + P_{\mathrm{a}} \cdot \frac{0.45}{11} \cdot 10^{-6}$	Dynamic internal battery resistance

References

[1] Creutzig F, P Jochem, OY Edelenbosch, L Mattauch, DP v. Vuuren, D McCollum, J Minx. Transport – a roadblock to climate change mitigation? Science 2015; 350 (6263):911-912.

[2] Jochem P, S Babrowski, W Fichtner. Assessing CO_2 Emissions of Electric Vehicles in Germany in 2030. Transportation Research A: Policy and Practice. 2015; 78:68-83.

[3] International Energy Agency (IEA). Global EV Outlook - 2015 Update. Paris: International Energy Agency (IEA); 2015.

[4] Thiel C, Schmidt J, Van Zyl A, Schmid E. Cost and well-to-wheel implications of the vehicle fleet CO_2 emission regulation in the European Union. Transportation Research Part A. 2014; 63:25–42.

[5] Hacker F, Harthan R, Matthes F, Zimmer W. Environmental impacts and impact on the electricity market of a large scale introduction of electric cars in Europe-Critical Review of Literature. ETC/ACC technical paper. 2009;4:56-90.

[6] Chen Z, Xia B, You C, Mi CC. A novel energy management method for series plug-in hybrid electric vehicles. Applied Energy. 2015; 145:172-179.

[7] Armenta J, Núñez C, Visairo N, Lázaro I. An advanced energy management system for controlling the ultracapacitor discharge and improving the electric vehicle range. Journal of Power Sources. 2015; 284:452-458.

[8] Song Z, Li J, Han X, Xu L, Lu L, Ouyang M, Hofmann H. Multi-objective optimization of a semi-active battery/supercapacitor energy storage system for electric vehicles. Applied Energy. 2014; 135:212-224.

[9] Xi J, Li M, Xu M. Optimal energy management strategy for battery powered electric vehicles. Applied Energy. 2014; 134:332-341.

[10] Pohlenz D. Wirkungsgradoptimale Regelung eines elektrischen Fahrantriebs mit variabler Zwischenkreisspannung. Paderborn: Universität Paderborn; 2012.

[11] Tenner S, Gunther S, Hofmann W. Loss minimization of electric drive systems using a DC/DC converter and an optimized battery voltage in automotive applications. Vehicle Power and Propulsion Conference (VPPC), 2011 IEEE2011. p. 1-7.

[12] Torres JL, Gonzalez R, Gimenez A, Lopez J. Energy management strategy for plug-in hybrid electric vehicles. A comparative study. Applied Energy. 2014;113:816-24.

[13] Butler KL, Ehsani M, Kamath P. A Matlab-based modeling and simulation package for electric and hybrid electric vehicle design. IEEE Transactions on Vehicular Technology. 1999;48:1770-8.

[14] Ade M, Binder A. Modeling the drive train for two parallel Hybrid Electric Vehicles in MATLAB/Simulink. 2009:592-600.

[15] Park G, Lee S, Jin S, Kwak S. Integrated modeling and analysis of dynamics for electric vehicle powertrains. Expert Systems with Applications. 2014; 41(5): 2595-2607.

[16] Trovão JP, Pereirinha PG, Jorge HM, Antunes CH. A multi-level energy management system for multi-source electric vehicles – An integrated rule-based meta-heuristic approach. Applied Energy. 2013; 105:304-318.

[17] Bertram C, Herzog H-G. Optimization method for drive train topology design and control of electric vehicles. 2013:1-8.

[18] Reinbold V, Vinot E, Gerbaud L. Global optimization of a parallel hybrid vehicle using optimal energy management. International Journal of Applied Electromagnetics and Mechanics. 2013;43:115-26.

[19] Wirasingha SG, Emadi A. Classification and Review of Control Strategies for Plug-In Hybrid Electric Vehicles. IEEE Transactions on Vehicular Technology. 2011;60:111-22.

[20] Koot M, Kessels JTBA, deJager B, Heemels WPMH, vandenBosch PPJ, Steinbuch M. Energy Management Strategies for Vehicular Electric Power Systems. IEEE Transactions on Vehicular Technology. 2005;54:771-82.

[21] Sciarretta A, Guzzella L. Control of hybrid electric vehicles. IEEE Control Systems Magazine. 2007;27:60-70.

[22] Vinot E, Trigui R, Cheng Y, Espanet C, Bouscayrol A, Reinbold V. Improvement of an EVT-Based HEV Using Dynamic Programming. IEEE Transactions on Vehicular Technology. 2014;63:40-50.

[23] Emanuel G, Naumann M, Jochem P, Fichtner W. Prädiktive Kraftstoffeinsatzoptimierung von Hybridfahrzeugen durch Metaheuristiken. Internationales Verkehrswesen. 2015;67:74-7.

[24] Shabbir W, Evangelou SA. Real-time control strategy to maximize hybrid electric vehicle powertrain efficiency. Applied Energy. 2014; 135:512-522.

[25] Chen BC, Wu YY, Tsai HC. Design and analysis of power management strategy for range extended electric vehicle using dynamic programming. Applied Energy. 2014;113: 1764-1774.

[26] Chen Z, Mi CC, Xiong R, Xu J, You C. Energy management of a power-split plug-in hybrid electric vehicle based on genetic algorithm and quadratic programming. Journal of Power Sources. 2014; 248:416-426.

[27] Song Z, Hofmann H, Li J, Hou J, Han X, Ouyang M. Energy management strategies comparison for electric vehicles with hybrid energy storage system. Applied Energy. 2014;134:321-331.

[28] Song Z, Hofmann H, Li J, Han X, Ouyang M. Optimization for a hybrid energy storage system in electric vehicles using dynamic programing approach. Applied Energy. 2015;139:151-162.

[29] Serrao L, Onori S, Rizzoni G. A Comparative Analysis of Energy Management Strategies for Hybrid Electric Vehicles. Journal of Dynamic Systems, Measurement, and Control. 2011;133:031012.

[30] Zou,Y, Liu T, Liu D, Sun F. Reinforcement learning-based real-time energy management for a hybrid tracked vehicle. Applied Energy. 2016; 171:372-382.

[31] Hou C, Ouyang M, Xu L, Wang H. Approximate Pontryagin's minimum principle applied to the energy management of plug-in hybrid electric vehicles. Applied Energy. 2016; 115:174-189.

[32] Osman I, Laporte G. Metaheuristics: A bibliography. Ann Oper Res. 1996;63:511-623.

[33] Vanhoucke M, Maenhout B. An empirical investigation of different solution strategies for meta-heuristic optimization: solution representation, diversity and space reduction. In: Siarry P, editor. Heuristics: Theory and Applications. New York: Nova Science Publishers, Inc.; 2013.

[34] Hu M, Weir JD, Wu T. Decentralized operation strategies for an integrated building energy system using a memetic algorithm. European Journal of Operational Research. 2012;217:185-97.

[35] Jochem P, Schönfelder M, Fichtner W. An efficient two-stage algorithm for decentralized scheduling of micro-CHP units. European Journal of Operational Research. 2015; 245(3):862-874.

[36] Goeke D, Schneider M. Routing a mixed fleet of electric and conventional vehicles. European Journal of Operational Research. 2015.

[37] Barr R, Golden B, Kelly J, Resende MC, Stewart W, Jr. Designing and reporting on computational experiments with heuristic methods. J Heuristics. 1995;1:9-32.

[38] Hooker JN. Testing heuristics: We have it all wrong. J Heuristics. 1995;1:33-42.

[39] Birattari M, Dorigo M. How to assess and report the performance of a stochastic algorithm on a benchmark problem: mean or best result on a number of runs? Optimization Letters. 2007;1:309-11.

[40] Coffin M, Saltzman MJ. Statistical Analysis of Computational Tests of Algorithms and Heuristics. INFORMS Journal on Computing. 2000;12:24-44.

[41] Angeline P. Evolutionary optimization versus particle swarm optimization: Philosophy and performance differences. In: Porto VW, Saravanan N, Waagen D, Eiben AE, editors. Evolutionary Programming VII: Springer Berlin Heidelberg; 1998. p. 601-10.

[42] Keane AJ. A brief comparison of some evolutionary optimization methods. In: Rayward-Smith VJ, Osman IH, Reeves CR, Smith GD, editors. Modern Heuristic Search Methods. Chichester: Wiley; 1996.

[43] Guzzella L, Sciarretta A. Vehicle Propulsion Systems: Introduction to Modeling and Optimization. Heidelberg: Springer; 2013.

[44] Hofmann P. Hybridfahrzeuge: Ein alternatives Antriebskonzept für die Zukunft. Wien: Springer; 2014.

[45] Schröder D. Elektrische Antriebe: Grundlagen. Heidelberg: Springer Vieweg; 2013.

[46] Schröder D. Elektrische Antriebe: Regelung von Antriebssystemen. Heidelberg: Springer; 2009.

[47] Lee J, Nam K, Choi S, Kwon S. Loss-Minimizing Control of PMSM With the Use of Polynomial Approximations. IEEE Transactions on Power Electronics. 2009;24:1071-82.

[48] Decker S. Verlustoptimaler Betrieb einer permanenterregten Synchronmaschine mit Reluktanz. Karlsruhe: Karlsruhe Institute of Technology (KIT); 2012.

[49] Kolar JW, Ertl H, Zach FC. Influence of the modulation method on the conduction and switching losses of a PWM converter system. Industry Applications, IEEE Transactions on. 1991;27:1063-75.

[50] Nohan N, Undeland TM, Robbins WP. Power Electronics - Converters, Applications, and Design. 3rd ed. Hoboken: John Wiley & Sons; 2003.

[51] Schoenen T, Kunter MS, Hennen MD, De Doncker RW. Advantages of a variable DC-link voltage by using a DC-DC converter in hybrid-electric vehicles. Vehicle Power and Propulsion Conference (VPPC), 2010 IEEE2010. p. 1-5.

[52] Klöffer C. Reichweitenverlängerung von Elektrofahrzeugen durch Optimierung des elektrischen Antriebsstranges, Herzogenrath, Shaker; 2015.

[53] Erickson RW, Maksimovic D. Fundamentals of Power Electronics. 2nd ed. New York: Kluwer Academic; 2001.

[54] Reinert J, Brockmeyer A, De Doncker RWA. Calculation of losses in ferro- and ferrimagnetic materials based on the modified Steinmetz equation. Industry Applications, IEEE Transactions on. 2001;37:1055-61.

[55] Rafiq M, Mohammed F-U-H, Yaqoob M, Thiringer T. Analysis of power losses and efficiency up to 60 kW DC-DC converter for hybrid electric vehicle with different inductive core materials. 2013:1-5.

[56] Getreuer P. Writing Fast Matlab Code. 2009.

[57] Dréo J, Siarry P, Pétrowski A, Taillard E. Metaheuristics for Hard Optimization: Methods and Case Studies. Heidelberg: Springer; 2006.

[58] Blum C, Roli A. Metaheuristics in combinatorial optimization. ACM Computing Surveys. 2003;35:268-308.

[59] Eiben AE, Smit SK. Parameter tuning for configuring and analyzing evolutionary algorithms. Swarm and Evolutionary Computation. 2011;1:19-31.

[60] Baghaee HR, Jannati M, Vahidi B, Hosseinian SH, Rastegar H. Improvement of voltage stability and reduce power system losses by optimal GA-based allocation of multi-type FACTS devices. 2008:209-14.

[61] Eiben G, Schut MC. New Ways to Calibrate Evolutionary Algorithms. In: Siarry P, Michalewicz Z, editors. Advances in Metaheuristics for Hard Optimization. Berlin: Springer; 2008.

[62] Abraham A, Liu H, Zhao M. Particle Swarm Scheduling for Work-Flow Applications in Distributed Computing Environments. In: Xhafa F, Abraham A, editors. Metaheuristics for Scheduling in Industrial and Manufacturing Applications. Berlin: Springer; 2008.

[63] Graham C, Talay D. Stochastic Simulation and Monte Carlo Methods: Mathematical Foundations of Stochastic Simulation. Heidelberg: Springer; 2013.

[64] Matsumoto M, Nishimura T. Mersenn twister: a 623-dimensional equidistributed uniform pseudorandom number generator. ACM Transactions on Modeling and Computer Simulation - Special issue on uniform random number generation. 1989;8:3-30.

[65] Michalewicz Z, Fogel DB. How to Solve It: Modern Heuristics. Berlin: Springer; 2004.

[66] Beasley D, Bull DR, Martin RR. An Overview of Genetic Algorithms: Part 2, Research Topics. University Computing. 1993;15:170-81.

[67] Joe S, Kuo FY. Remark on Algorithm 659: Implementing Sobol's Quasirandom Sequence Generator. ACM Transactions on Mathematical Software. 2003;29:49-57.

[68] Matousek J. On the L2-Discrepancy for Anchored Boxes. Journal of Complexity. 1998;14:527-56.

[69] Chi H, Beerli P, Evans D, Mascagni M. On the Scrambled Soboĺ Sequence. In: Sunderam V, van Albada G, Sloot PA, Dongarra J, editors. Computational Science – ICCS 2005: Springer; 2005. p. 775-82.

[70] Rogers A, Prügel-Bennett A. Genetic drift in genetic algorithm selection schemes. Evolutionary Computation, IEEE Transactions on. 1999;3:298-303.

[71] Goldberg DE, Deb K. A comparative analysis of selection schemes used in genetic algorithms. Foundations of genetic algorithms. 1991;1:69-93.

[72] Bäck T, Hoffmeister F. Extended Selection Mechanisms in Genetic Algorithms. In: Belew RK, Booker LB, editors. Fourth International Conference on Genetic Algorithms. San Mateo: Morgan Kaufmann; 1991. p. 92-9.

[73] Hesser J, Männer R. Towards an optimal mutation probability for genetic algorithms. In: Schwefel H-P, Männer R, editors. Parallel Problem Solving from Nature: Springer Berlin Heidelberg; 1991. p. 23-32.

[74] Bremermann HJ, Rogson M, Salaff S. Global properties of evolution processes. In: Pattee HH, Edlsack EA, Fein L, Callahan AB, editors. Natural Automata and Useful Simulations. Washington D.C.: Spartan Books; 1966.

[75] Mühlenbein H, Schlierkamp-Voosen D. Analysis of selection, mutation and recombination in genetic algorithms. In: Banzhaf W, Eeckman F, editors. Evolution and Biocomputation: Springer Berlin Heidelberg; 1995. p. 142-68.

[76] Mühlenbein H, Schlierkamp-Voosen D. Predictive Models for the Breeder Genetic Algorithm I. Continuous Parameter Optimization. Evolutionary Computation. 1993;1:25-49.

[77] Luque G, Alba E. Parallel Genetic Algorithms - Theory and Real World Applications. Berlin Heidelberg: Springer; 2011.

[78] Gordon VS, Whitley D. Serial and Parallel Genetic Algorithms as Function Optimizers. In: Forrest S, editor. Fifth International Conference on Genetic Algorithms. San Mateo: Morgan Kaufmann Publishers; 1993. p. 177-83.

[79] Kirkpatrick S, Gelatt CD, Vecchi MP. Optimization by Simulated Annealing. Science. 1983;220:671-80.

[80] Corne D, Ross P. Some combinatorial landscapes on which a Genetic Algorithm outperforms other Stochastic iterative methods. In: Fogarty T, editor. Evolutionary Computing: Springer; 1995. p. 1-13.

[81] Chen S, Flann NS. Parallel simulated annealing and genetic algorithms: A space of hybrid methods. In: Davidor Y, Schwefel H-P, Männer R, editors. Parallel Solving from Nature – PPSN III. Berlin: Springer; 1994.

[82] Ingber L. Simulated annealing: Practice versus theory. Mathematical and Computer Modelling. 1993;18:29-57.

[83] Eglese RW. Simulated annealing: A tool for operational research. European Journal of Operational Research. 1990;46:271-81.

[84] Kennedy J, Eberhart R. Particle swarm optimization. Neural Networks, 1995 Proceedings, IEEE International Conference on1995. p. 1942-8 vol.4.

[85] Banks A, Vincent J, Anyakoha C. A review of particle swarm optimization. Part I: background and development. Nat Comput. 2007;6:467-84.

[86] Banks A, Vincent J, Anyakoha C. A review of particle swarm optimization. Part II: hybridisation, combinatorial, multicriteria and constrained optimization, and indicative applications. Nat Comput. 2008;7:109-24.

[87] Sun J, Lai C-H, Wu X-J. Particle Swarm optimisation: classical and quantum perspectives. In: Lai C-H, Magoulès F, editors. Numerical Analysis and Scientific Computing. Boca Raton: CRC Press; 2012.

[88] Shi Y, Eberhart R. A modified particle swarm optimizer. Evolutionary Computation Proceedings, 1998 IEEE World Congress on Computational Intelligence, The 1998 IEEE International Conference on1998. p. 69-73.

[89] Czogalla J, Fink A. On the Effectiveness of Particle Swarm Optimization and Variable Neighborhood Descent for the Continuous Flow-Shop Scheduling Problem. In: Xhafa F, Abraham A, editors. Metaheuristics for Scheduling in Industrial and Manufacturing Applications. Berlin: Springer; 2008. p. 61-9.

[90] Bluman A. Elementary Statistics: A Step By Step Approach. 8th ed. New York: McGraw-Hill; 2011.

[91] Klöffer C. Reichweitenverlängerung von Elektrofahrzeugen durch Optimierung des elektrischen Antriebsstranges. Karlsruhe: Karlsruhe Institute of Technology (KIT); 2015.

Working Paper Series in Production and Energy

recent issues

No. 7 Tobias Jäger, Russell McKenna, Wolf Fichtner:
Onshore wind energy in Baden-Württemberg: a bottom-up economic assessment of the socio-technical potential

No. 8 Axel Ensslen, Alexandra-Gwyn Paetz, Sonja Babrowski, Patrick Jochem, Wolf Fichtner:
On the road to an electric mobility mass market - How can early adopters be characterized?

No. 9 Kai Mainzer, Russell McKenna, Wolf Fichtner:
Charakterisierung der verwendeten Modellansätze im Wettbewerb Energieeffiziente Stadt

No. 10 Hannes Schwarz, Valentin Bertsch, Wolf Fichtner:
Two-stage stochastic, large-scale optimization of a decentralized energy system – a residential quarter as case study

No. 11 Russell McKenna, Erik Merkel, Wolf Fichtner:
Development of a multi-energy residential service demand model for evaluation of prosumers' effects on current and future residential load profiles for heat and electricity

No. 12 Russell McKenna, Erik Merkel, Wolf Fichtner:
Energy autonomy in residential buildings: a techno-economic model-based analysis of the scale effects

No. 13 Johannes Schäuble, Silvia Balaban, Peter Krasselt, Patrick Jochem, Mahmut Özkan, Friederike Schellhas-Mende, Wolf Fichtner, Thomas Leibfried, Oliver Raabe:
Vergleichsstudie von Systemansätzen für das Schnellladen von Elektrofahrzeugen

No. 14 Marian Hayn, Valentin Bertsch, Anne Zander, Stefan Nickel, Wolf Fichtner:
The impact of electricity tariffs on residential demand side flexibility

No. 15 Erik Merkel, Robert Kunze, Russell McKenna, Wolf Fichtner:
Modellgestützte Bewertung des Kraft-Wärme-Kopplungsgesetzes 2016 anhand ausgewählter Anwendungsfälle in Wohngebäuden

No. 16 Russell McKenna, Valentin Bertsch, Kai Mainzer, Wolf Fichtner:
Combining local preferences with multi-criteria decision analysis and linear optimisation to develop feasible energy concepts in small communities

Impressum

Karlsruher Institut für Technologie

Institut für Industriebetriebslehre und Industrielle Produktion (IIP)
Deutsch-Französisches Institut für Umweltforschung (DFIU)

Hertzstr. 16
D-76187 Karlsruhe

KIT – Universität des Landes Baden-Württemberg und
nationales Forschungszentrum in der Helmholtz-Gemeinschaft

Working Paper Series in Production and Energy
No. 17, December 2016

ISSN 2196-7296