Applying (Hybrid) Metaheuristics to Fuel Consumption Optimization of Hybrid Electric Vehicles

Thorsten Krenek[1], Mario Ruthmair[2], Günther R. Raidl[2], and Michael Planer[1]

[1] Institute for Powertrains and Automotive Technology,
Vienna University of Technology, Vienna, Austria
{thorsten.krenek,michael.planer}@ifa.tuwien.ac.at
[2] Institute of Computer Graphics and Algorithms,
Vienna University of Technology, Vienna, Austria
{ruthmair,raidl}@ads.tuwien.ac.at

Abstract. This work deals with the application of metaheuristics to the fuel consumption minimization problem of hybrid electric vehicles (HEV) considering exactly specified driving cycles. A genetic algorithm, a downhill-simplex method and an algorithm based on swarm intelligence are used to find appropriate parameter values aiming at fuel consumption minimization. Finally, the individual metaheuristics are combined to a hybrid optimization algorithm taking into account the strengths and weaknesses of the single procedures. Due to the required time-consuming simulations it is crucial to keep the number of candidate solutions to be evaluated low. This is partly achieved by starting the heuristic search with already meaningful solutions identified by a Monte-Carlo procedure. Experimental results indicate that the implemented hybrid algorithm achieves better results than previously existing optimization methods on a simplified HEV model.

Keywords: hybrid metaheuristic, genetic algorithm, downhill-simplex, particle-swarm-optimization, hybrid electric vehicles, driving cycles

1 Introduction

Due to the requirement of lower fuel consumption and emissions it is necessary that the automotive industry comes up with new approaches. One of these are hybrid electric vehicles (HEV) which have a much higher flexibility concerning operation strategies and components compared to conventional vehicles utilizing only a combustion engine. The propulsion system of HEVs consists of a conventional combustion engine and electric machines. With the assistance of electric machines it is possible to achieve higher efficiency, in particular by providing energy recuperation in deceleration phases.

Nowadays engines and vehicles can be numerically simulated with high accuracy, which makes it easier to analyze different operation strategies and the consequences of their modification. Our aim is to minimize the fuel consumption in exactly specified driving cycles of such HEV computer models. The vehicle is simulated by the software GT-SUITE[3] using physics-based one-dimensional modeling thus being able to calculate the fuel consumption and the battery state of charge (SOC) for a specific driving cycle. Depending on the duration of the driving cycle, this can take several minutes on current hardware. In general, the fuel consumption is influenced by a large number of adjustable parameters from which we preselected a meaningful subset for optimization: velocities at which the vehicle switches from parallel to series hybrid mode and vice versa, the SOC operating limits and the gear shifting strategy. In parallel mode the internal combustion engine (ICE) and/or the electric machines are used for propulsion while in series mode only electric propulsion is provided utilizing the ICE to power the electric generator. A detailed parameter description is given in Section 5. All n parameters $p = (p_1, \ldots, p_n)$ of the HEV model are real-valued and have individual lower and upper bounds $[p_i^{\min}, p_i^{\max}],\ \forall i = 1 \ldots n$. The battery SOC is required to be nearly identical at the beginning and the end of a driving cycle in order to guarantee a fair comparison to other vehicles. So we considered the quadratic deviation between the SOC at the beginning (SOC_{begin}) and at the end (SOC_{end}) of the driving cycle. The objective function to be minimized is:

$$f(p) = w_{\text{cons}} \cdot cons(p) + w_{\text{sdev}} \cdot (SOC_{\text{begin}}(p) - SOC_{\text{end}}(p))^2$$

The fuel consumption is denoted by $cons(p)$ and constants $w_{\text{cons}} \geq 0$ and $w_{\text{sdev}} \geq 0$ are used for weighting the individual terms appropriately. A solution p^* is optimal if $f(p^*)$ is minimal, so $f(p^*) \leq f(p),\ \forall\ p$. A direct determination of proven optimal parameter settings is practically impossible due to the high complexity of f, even obtaining the objective for one set of parameters by simulation is quite time-consuming. So the goal was to find a heuristic optimization strategy making it possible to reliably find a solution that is close to optimal only requiring a limited number of simulations. Beginning with standard optimization techniques diverse in most cases more efficient algorithms than Design Of Experiments (DOE) [11], which is included in GT-SUITE, have been developed by considering special properties of the problem. A genetic algorithm (GA) [9], a downhill-simplex method [12], and an algorithm based on swarm intelligence (PSO) [5] provided, after some specific tailoring, in preliminary experiments the best results. Major features are: Starting solutions are not initialized randomly but by a Monte Carlo search procedure to reduce the number of required iterations. In the GA's recombination operator the choice which value is passed on depends on the deviation of the parameter values from the two parent solutions to the best solutions in the population. The simplex reduction in the downhill simplex method is not applied here because it re-calculates all points of the new simplex and this mostly ends up in worse objective function values due to possibly unbalanced SOCs. The best solution from the PSO algorithm is additionally improved by a surface-fitting algorithm. Finally, the individual metaheuristics are combined to a hybrid optimization approach taking into account the strengths and weaknesses of the single procedures.

[3] GT-SUITE is a software by Gamma Technologies, Inc., `http://www.gtisoft.com`

For a model of an existing HEV with complex operation strategy a fuel saving of about 33% compared to a related conventionally powered vehicle could be achieved. The part our hybrid optimization algorithm contributes is about five percent in comparison to setting the parameters by the methods implemented in GT-SUITE. These standard optimization methods in particular have problems with the high number of parameters. Furthermore, we are able to show that our proposed algorithm achieves better results on another simplified HEV benchmark model too, see Section 5.

The following Section discusses related work, Section 3 presents the individual metaheuristics which are then combined in Section 4 to a hybrid algorithm, Section 5 shows experimental results, and Section 6 concludes the article.

2 Related Work

In GT-SUITE a *Design of Experiments* optimization method is implemented. Here the search space is typically approximated by a quadratic or cubic polynomial function based on a large number of simulated parameter sets distributed in the search space. The minimum of this function is then derived analytically. In [7] and [14] several optimization algorithms are applied to HEV models and the authors state that the considered search space is highly non-linear with non-continuous areas. Similarly to our problem, the goal is to minimize the fuel consumption for a given driving cycle. As additional constraint they consider a minimum requirement on vehicle dynamics. As simulation software ADVISOR[4] is used and the applied optimization algorithms are taken from iSIGHT[5], VisualDOC[6] and MATLAB[7]. As optimization procedures *fmincon* from MATLAB, VisualDOC's DGO and RSA, as well as the search strategies Sequential Quadratic Programming (SQP) [13], DIviding RECTangle (DIRECT) [1] and a GA are applied. Unfortunately, there is no information given about the implementation and configuration of the used algorithms, in particular concerning the GA. The best result is achieved by the DIRECT method, the gradient strategies can only find rather poor local optima.

In [2] and [3] among others the simulation software PSAT[8] and its DIRECT optimization algorithms, a GA, Simulated Annealing (SA) and PSO are applied to a HEV model whereas SA and DIRECT are the most successful approaches. The objective is the same as in [7] and [14].

Furthermore, in [3] a hybrid algorithm combining SQP with DIRECT is presented but only applied on a simpler test function. However, in few iterations the global optimum is found in most cases. In [4] and [10] a multi-objective GA

---

[4] ADVISOR (Advanced Vehicle Simulator) is a software from AVL, `http://www.avl.com`

[5] iSIGHT is a software from Simulia, `http://www.simulia.com`

[6] VisualDOC is a software from VR & D, `http://www.vrand.com`

[7] MATLAB is a software from MathWorks `http://www.mathworks.de`

[8] PSAT (Powertrain System Analysis Toolkit) was developed by Argonne National Laboratory, `http://www.transportation.anl.gov/modeling_simulation/PSAT`

is successfully applied to a HEV model, considering fuel consumption and emissions minimization. Comparisons with other methods are not presented. In [15] and [16] a PSO algorithm was proposed for a HEV model for improving a given operation strategy. ADVISOR is used as simulation software. The SOC deviation on the defined driving cycles is integrated in the objective function. Given the characteristics of the vehicle the operation strategy is optimized resulting in an improvement compared to the strategy before. How the original strategy has already been optimized before is not stated. Compared to GT-SUITE parts of the objective function can be calculated much faster in ADVISOR and PSAT by directly solving mathematical functions. As a consequence, such models can be simulated significantly faster and gradient strategies can be applied. However, the benefit of using GT-SUITE is the much higher accuracy of the HEV model. In the mentioned related work not only the operation strategy but also other criteria, e.g. the battery capacity and the number of battery cells, are optimized. The requirement of a balanced SOC is either considered as a side constraint or by adding the difference to a balanced SOC as a penalty term to the objective function. In the first case a large number of infeasible solutions are possibly calculated, mainly by methods like DOE [11].

Most previous works use standard optimization methods from existing libraries without problem-specific adaptations, and different articles report different optimization methods to work best. Unfortunately, a direct comparison of these approaches is hardly possible since only few algorithmic details are available. Thus, it is difficult to draw general conclusions about appropriate methods for the optimization of HEVs. GT-SUITE provides a DOE-based optimization method too, however, in our studies we recognized that DOE can only handle up to five parameters in reasonable time for our HEV models.

3 Metaheuristics

We now describe the new metaheuristic approaches we developed. For more details, in particular also deeper studies of the individual algorithms' performances and influences of strategy parameters, we refer to the first author's master thesis [6], on which this article is based.

Monte-Carlo Search Method The Monte-Carlo method [8] is primarily used to generate manifold initial solutions for the other algorithms. The initial range of values for each parameter is set to the entire range of possible values. Consequently, in a first step only random solutions are generated. After each iteration the parameter range is reduced by a factor and moved towards the best known solution. Due to the fact that the algorithm mainly generates initial solutions subject to further improvement we choose a factor between 0.8 and 0.9 and keep the number of computed solutions constant.

Downhill-Simplex Method This method [12], also known as Nelder-Mead method, is based on a v-simplex, which is a polytope of dimension v defined by $v + 1$ points spanning the convex hull. Each point corresponds to a particular set of parameters together with its objective function value. By comparing the different function values the tendency of the values and gradient directions are approximated. In each iteration, the point with the worst value is replaced by a newly derived one. In our implementation we omit the otherwise usual shrinking of the whole simplex because it would be very time-consuming to re-calculate the objective values of all points of the simplex. Furthermore, these new points are likely to have an unbalanced SOC.

Genetic Algorithm (GA) In our GA [9] each individual is directly represented by a vector of real parameter values. The selection of solutions from the population for pairwise recombination occurs uniformly at random. To recombine two solutions $p^1,\ p^2$, for each parameter $i = 1 \dots n$, either p_i^1 or p_i^2 is adopted. The choice which value is passed on considers the average deviation to the d best solutions $q^j,\ j = 1 \dots d$, in the population:

$$dev_i(p^k) = \frac{1}{d} \sum_{j=1}^{d} |q_i^j - p_i^k| \qquad \forall k \in \{1, 2\},\ \forall i = 1 \dots n$$

The probability of adopting the i-th parameter from parent p^k is then defined as $P_i^{\mathrm{comb}}(p^k) = 1 - dev_i(p^k)/(dev_i(p^1) + dev_i(p^2))$. Furthermore, each parameter is mutated with a small probability P^{mut} by assigning it a new random value within its bounds. Once an offspring solution p' has been generated and its objective value $f(p')$ determined via simulation, a solution r is randomly selected from the population and replaced with probability $P^{\mathrm{rep}} = (f(r) - c)/(f(p') + f(r) - 2c)$. The correction value $c \in [0, \min\{f(p'), f(r)\})$ is used to control the influence of the objective values: the higher c the higher the probability of a new solution with better objective value being chosen as new member of the population.

Particle-Swarm-Optimization (PSO) This optimization method was originally derived from the behavior of birds and shoals of fish [5]. Each solution $p^j,\ j = 1 \dots m$, corresponds to an individual of a swarm of size m moving within the search space. The motion depends both on the best known solution of the individual and the best solution of the entire swarm. First, m solutions are randomly selected from the solution set of the Monte-Carlo search procedure to form the initial population. For each individual j the so-far best "local" solution $p^{\mathrm{L},j}$ encountered on its path is stored. Moreover, p^{G} denotes the overall best known solution. In each iteration the parameter set of each individual is modified depending on both the local and global best solutions. For each individual s^j a velocity vector $v^j \in [-1, 1]^n$ is defined and updated as follows:

$$v_i^j \leftarrow v_i^j + \alpha^{\mathrm{L}} \cdot \frac{p_i^{\mathrm{L},j} - p_i^j}{p_i^{\max} - p_i^{\min}} + \alpha^G \cdot \frac{p_i^{\mathrm{G}} - p_i^j}{p_i^{\max} - p_i^{\min}} + rand \qquad \begin{array}{l} \forall j = 1 \dots m, \\ \forall i = 1 \dots n. \end{array}$$

Constants $\alpha^L, \alpha^G \geq 0$, with $\alpha^L + \alpha^G = 1$, control the influence of the local and global best solutions, respectively, and $rand$ is a random value uniformly distributed in $[-0.1, 0.1]$. The positions (solutions) of the individuals are then updated by $p_i^j \leftarrow p_i^j + v_i^j \cdot (p_i^{\max} - p_i^{\min})/\delta$, where $\delta \geq 1$ controls the step size. If a parameter steps out of its corresponding range, it is set to the corresponding limit. The algorithm terminates after a specified number of iterations.

Surface-Fitting We use surface-fitting to improve the best solution obtained by the PSO algorithm in our hybrid metaheuristic approach, see Section 4. In each iteration $e \geq 6$ solutions are derived from the so far best solution by varying two randomly selected parameters p_1, p_2 slightly. The range of the variation is limited by the following factors: factor *area* is initialized with 1 and increases by 1 after every fourth solution. The factors $(\mathit{fit}_1, \mathit{fit}_2)$ are continuously assigned the values $(-1,-1)$, $(1,-1)$, $(-1,1)$ and $(1,1)$. The constant *rad* denotes the step size relative to the range of feasible parameter values. For the chosen parameters $i = 1 \ldots 2$ the parameter values are calculated by $p_i = p_i + \mathit{area} \cdot \mathit{fit}_i \cdot \mathit{rad} \cdot (p_i^{\max} - p_i^{\min})$. The new solutions are evaluated and the objective function is approximated by function $c_1 + c_2 \cdot p_1 + c_3 \cdot p_2 + c_4 \cdot p_1^2 + c_5 \cdot p_2^2 + c_6 \cdot p_1 \cdot p_2$. Coefficients $c_1 \ldots c_6$ and the minimum of the approximation function are calculated using the GNU Scientific Library and finally evaluated by simulation.

4 Hybrid Meta-Heuristic (PSAGADO)

Each presented method has its own strenghts and weaknesses. On average the GA was able to achieve the best results since by mutation it was possible to escape from unpromising areas of the search space. However, rather good solutions often could not be further improved. The results of the PSO and the downhill simplex method are highly dependent on the chosen initial solutions. If only the PSO is applied, the solutions have to be broader distributed in the search space and should have nearly a balanced SOC. Our hybrid approach (*Particle-Swarm And Genetic Algorithm with Downhill-simplex Optimization*, PSAGADO) combines the previously presented algorithms trying to exploit their strengths. Initial solutions are determined by the Monte-Carlo search method and stored in a solution pool. As not much is known about the search space the PSO is well suited to be the central algorithm, since it is a robust method considering solutions with high diversity. After a certain number of iterations the best solution of the PSO is improved by the surface-fitting procedure if possible. Surface-fitting is only applied to the best solution because of runtime considerations. The GA is applied next using the final swarm of the PSO as initial population. If most of the individuals are similar, the GA still can lead to new best solutions by increasing diversity by mutation. If the solutions are well distributed in the search space recombination is frequently able to combine two good parameter sets to a better one. After recombination two solutions are randomly chosen from the population. If the new solution is better than both selected, one solution is replaced by the new solution and the other one by a random solution from the initial solution pool to restrict similar solutions in the pool. Otherwise only the chosen solution with the lower objective value will be replaced by the new solution. If the GA is able to find a new best solution, half of the solutions closest

Algorithm 1: PSAGADO

```
execute Monte-Carlo search and store all solutions as initial pool
while termination criterion not met do
    execute PSO
    apply surface-fitting on the best solution of PSO
    execute GA
    if new best solution found then replace half of the solutions closest to best
    else
        execute downhill-simplex
        if no new best solution found then replace all solutions
```

Table 1. Algorithm settings.

Monte-Carlo	$resize = 0.89$
SIMPLEX	$v = 15$
SURFACE-FITTING	$e = 12$, $rad = 0.02$
PSO	$\alpha^L = 0.3$, $\alpha^G = 0.7$, $m = 30$, $\delta = 10$
GA	$c = \min\{f(p'), f(r)\} - 2$, $d = 10$, $P^{\text{mut}} = 10\%$

to the best solution are replaced by random solutions from the pool to increase diversity and prevent too much focus on the best solution. The distance $D(p)$ of parameter set p to the best solution p^{best} is calculated by

$$D(p) = \sum_{i=1}^{n} \left(\frac{|p_i - p_i^{\text{best}}|}{p_i^{\max} - p_i^{\min}} \right)^2 .$$

If the GA is not able to achieve any improvement, the simplex method is applied. This usually occurs when most of the PSO solutions are very similar. Although this could mean that most solutions are near the global optimum bad solutions may still exist possibly resulting in a shift of the simplex and leading to a new best solution. If the simplex method leads to an improvement, the process continues with the PSO. However, if most solutions are quite similar and the PSO and GA cannot achieve new best solutions then the simplex method usually results in no improvement, too. In this case a restart is performed by replacing all solutions but the so-far best with solutions from the initial pool and continuing with the PSO. Algorithm 1 shows the implementation of PSAGADO.

5 Experimental Results

We applied PSAGADO to a complex real-world and a simplified benchmark HEV model. Unfortunately we are not allowed to publish details for the real-world model due to a non-disclosure agreement with the manufacturer. Overall, a fuel saving of about 33% compared to a related conventionally powered vehicle could be achieved, and the remarkable part PSAGADO contributes is about five percent in comparison to the parameter setting found by DOE integrated in GT-SUITE. As simplified benchmark HEV model we used the “parallel-series” example supplied by GT-SUITE and compare PSAGADO to the integrated DOE

Table 2. Final objective values of PSAGADO, DOE, GA, SIMPLEX and DOE.

	Runs	Sol. p. Run	Worst	Best	Average	Std.Dev.
PSAGADO	10	3600	207.52	206.69	206.92	0.23
PSO	10	3600	229.93	207.22	212.43	12.85
GA	10	3600	208.64	206.98	207.23	0.25
SIMPLEX	10	3600	230.57	207.94	215.93	14.10
DOE	10	3600	210.87	210.19	210.40	0.23

and the individual metaheuristics. To further reduce simulation times a shorter driving-cycle is used here altogether leading to an evaluation time for one parameter set of about 30 seconds. Thus, the runtime of the optimization algorithms can be neglected compared to the simulation times. Important algorithm specific settings are shown in Table 1. The Monte-Carlo search method calculates 35 solutions at each of 15 total iterations. The population size for the PSO and GA is 25. In each optimization cycle the PSO is iterated ten times, the surface-fitting method is applied five times and in the GA 60 new solutions are derived. In case of no improvement, the simplex will be updated 15 times. The constants in the objective function are set to $w_{\mathrm{cons}} = 3.6$ and $w_{\mathrm{sdev}} = 9$. All parameter values have been determined in preliminary tests to fit the limited number of simulations. The fuel consumption *cons* is measured in mg, the SOC in percent. The parameters to be optimized are the gearshift strategy defined by *gear1up* to *gear4up*, the charging limits of the battery $SOC_{\mathrm{min}}, SOC_{\mathrm{max}}$ and hybrid mode thresholds *hev1*, *hev2* specifying the velocities switching from parallel to series mode and vice versa. DOE uses the latin-hypercube method to select the parameter sets and approximates the mathematical model by a cubic replacement function. Results obtained from 10 runs with 3600 evaluated solutions per run for each considered algorithm are summarized in Table 2.

In the optimization progress we observed several local optima from which one cannot escape by changing only one parameter. If the Monte-Carlo method leads to a poor local optimum it may take some time until PSAGADO gets out of it mainly because of the low diversity of the initial solution pool. To prevent this the range reduction factor could be increased or the number of iterations in the Monte-Carlo search procedure could be reduced. Another possibility would be to entirely skip the Monte-Carlo method and use only random solutions. However, since the number of simulations is strictly limited we decided to initially restrict the search space even if there is a risk of getting stuck in a local optimum. DOE often fails because of an inaccurate model approximation in the relevant areas containing good solutions which can be explained by the rather naive uniform sampling strategy. Table 3 shows the best solutions obtained by the individual algorithms; notable are the remarkably strong differences in the parameter values. Among PSO, SIMPLEX and the GA, the GA performed best, using mutation to escape from unfavorable areas of the search space. The results of the PSO strongly depend on the diversity and the SOC balance of the initial solutions. In the downhill-simplex method it is necessary to start with solutions with almost balanced SOC otherwise it is difficult to find good solutions.

Table 3. Obtained best parameter sets of PSAGADO, PSO, SIMPLEX, GA and DOE.

Parameter	Boundaries	PSAGADO	PSO	SIMPLEX	GA	DOE
hev1 [km/h]	65–100	65.00	65.00	65.04	65.02	100.00
hev2 [km/h]	10–60	60.00	60.00	59.95	59.82	60.00
SOC_{max}	0.7–0.9	0.79	0.70	0.78	0.73	0.90
SOC_{min}	0.1–0.7	0.50	0.57	0.47	0.55	0.10
gear1up [km/h]	12–30	29.93	12.93	27.08	29.62	25.87
gear2up [km/h]	32–50	47.84	42.01	38.45	47.19	44.69
gear3up [km/h]	52–70	57.53	57.87	59.11	57.52	53.72
gear4up [km/h]	72–100	72.00	72.10	87.63	72.00	76.13

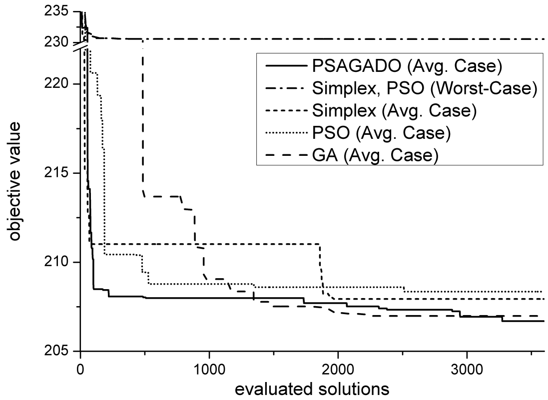

Fig. 1. Characteristic optimization progresses.

Characteristic optimization progresses of all methods are shown in Fig. 1, where worst-case scenarios of downhill-simplex method and PSO are shown together in one curve.

6 Conclusions and Future Work

We considered the problem of optimizing diverse control strategy parameters of hybrid vehicles in order to minimize fuel consumption over a given driving-cycle. This problem is characterized by the relatively large number of real-valued parameters, the multi-modality and discontinuity of the search space, and in particular the expensive simulations required for evaluating solutions. Consequently, we investigated diverse heuristic strategies including Monte Carlo and Downhill-Simplex approaches, a specifically adapted GA, and a PSO. Considering the individual properties of these methods, we finally combined them in the hybrid PSAGADO. Results on a complex real-world scenario were remarkable, with PSAGADO's solution leading to a reduction of the fuel consumption of about five percent in comparison to a standard optimization strategy provided by the GT-SUITE simulator. As we are not allowed to give more details here on these results, a simplified benchmark model was further used for comparison, also indicating the superiority of PSAGADO over the individual metaheuristics and GT-SUITE's DOE.

In future work more testing is necessary and the search space should be studied in more detail in order to possibly exploit certain features in the optimization in better ways. A promising idea seems to be to approximate the objective function with a neural network which is refined at the same time as the optimization is performed.

References

1. Finkel, D.: Direct optimization algorithm user guide. North Carolina State University, Center for Research in Scientific Computation 2 (2003)
2. Gao, D., Mi, C., Emadi, A.: Modeling and simulation of electric and hybrid vehicles. Proceedings of the IEEE 95(4), 729–745 (2007)
3. Gao, W., Porandla, S.: Design optimization of a parallel hybrid electric powertrain. In: IEEE Conference on Vehicle Power and Propulsion. pp. 6–12. IEEE (2005)
4. Huang, B., Wang, Z., Xu, Y.: Multi-objective genetic algorithm for hybrid electric vehicle parameter optimization. In: IEEE/RSJ International Conference on Intelligent Robots and Systems. pp. 5177–5182. Beijing, China (2006)
5. Kennedy, J., Eberhart, R.: Particle swarm optimization. In: Proceedings of the IEEE International Conference on Neural Networks. vol. 4, pp. 1942–1948. IEEE (1995)
6. Krenek, T.: Verbrauchsminimierung eines Hybridfahrzeuges im Neuen Europäischen Fahrzyklus. Master's thesis, Vienna University of Technology, Institute of Computer Graphics and Algorithms, Vienna, Austria (July 2011)
7. Markel, T., Wipke, K.: Optimization techniques for hybrid electric vehicle analysis using advisor. In: Proceedings of the ASME International Mechanical Engineering Congress and Exposition. pp. 11–16. New York, USA (2001)
8. Meywerk, M.: CAE-Methoden in der Fahrzeugtechnik. Springer, Heidelberg (2007)
9. Michalewicz, Z.: Heuristic methods for evolutionary computation techniques. Journal of Heuristics 1(2), 177–206 (1996)
10. Montazeri-Gh, M., Poursamad, A., Ghalichi, B.: Application of genetic algorithm for optimization of control strategy in parallel hybrid electric vehicles. Journal of the Franklin Institute 343(4-5), 420–435 (2006)
11. Myers, R., Montgomery, D., Anderson-Cook, C.: Response surface methodology: process and product optimization using designed experiments. Wiley (2009)
12. Nelder, J., Mead, R.: A Simplex Method for Function Minimization. Oxford Journals - The Computer Journal, British Computer Society 7(4), 308–313 (1965)
13. Nocedal, J., Wright, S.: Numerical optimization. Springer (2006)
14. Wipke, K., Markel, T., Nelson, D.: Optimizing energy management strategy and degree of hybridization for a hydrogen fuel cell SUV. In: Proceedings of 18th Electric Vehicle Symposium. Berlin (2001)
15. Wu, J., Zhang, C., Cui, N.: PSO algorithm-based parameter optimization for HEV powertrain and its control strategy. International Journal of Automotive Technology 9(1), 53–59 (2008)
16. Wu, X., Cao, B., Wen, J., Bian, Y.: Particle swarm optimization for plug-in hybrid electric vehicle control strategy parameter. In: IEEE Conference on Vehicle Power and Propulsion. pp. 1–5. Harbin, China (2008)