Hybrid vehicle design using global optimisation algorithms

Wenzhong Gao

Department of Electrical and Computer Engineering,
Center for Energy Systems Research,
Tennessee Technological University, 1020 Stadium Dr, PH414,
Box 5032, Cookeville, TN 38505, USA
E-mail: wgao@tntech.edu

Chris Mi*

Department of Electrical and Computer Engineering,
University of Michigan, 4901 Evergreen Road, Dearborn,
MI 48128 USA
Fax: 313-583-6336 E-mail: mi@ieee.org
*Corresponding author

Abstract: Four global optimisation algorithms are applied in the design optimisation of a hybrid electric vehicle (HEV). These four algorithms are: DIRECT, Simulated Annealing, Genetic Algorithm, and Particle Swarm Optimisation. The optimisation objective is to achieve maximum fuel economy, subject to the constraints of vehicle performance. The model in the loop methodology is adopted for our design process, in which a vehicle model named PSAT is used as the analysis tool. The design optimisation results and the performance of the four optimisation algorithms are compared. Our initial study shows that DIRECT and Simulated Annealing algorithms are efficient for the complex HEV engineering design problem.

Keywords: design optimisation; global optimisation algorithm; hybrid electric vehicle; PSAT.

Reference to this paper should be made as follows: Gao, W. and Mi, C. (2007) 'Hybrid vehicle design using global optimisation algorithms', *Int. J. Electric and Hybrid Vehicles*, Vol. 1, No. 1, pp.57–70.

Biographical notes: Wenzhong Gao received his BSc in Aeronautical Propulsion Control Engineering from Northwestern Polytechnic University, Xi'an, China, in 1988, and his MSc and PhD in Electrical and Computer Engineering from Georgia Institute of Technology, Atlanta, USA, in 1999 and 2002, respectively. From 2002 to 2006, he worked as an assistant research professor in the University of South Carolina and Mississippi State University. Since 2006, he has worked as an assistant professor in Tennessee Tech University. His current research interests include hybrid electric propulsion systems, power system modelling and simulation, alternative power systems, renewable energy and electric machinery and drives.

Chris Mi is associate professor of electrical engineering at the University of Michigan, Dearborn. He received BSEE and MSEE degrees from Northwestern Polytechnical University, China, and a PhD degree from

University of Toronto, Canada. His research interests are in power electronics, hybrid electric vehicles, electric machines, and renewable energy systems. He was with Northwestern Polytechnical University from 1988 to 1994. He joined Xi'an Petroleum Institute as an associate professor and Associate Chair of Department of Automation in 1994. He worked for General Electric Canada Inc. from 2000 to 2001. Chris Mi is the recipient of numerous awards including the Distinguished Teaching Award from the University of Michigan, National Innovation Award and Government Special Allowance Award from China, and many IEEE Awards.

1 Introduction

Due to environmental and energy concerns, the hybrid electric vehicle (HEV) is becoming an important research topic. Its heart lies at an innovative hybrid powertrain, whose parameters must be tuned for a better performance of the hybrid vehicle (Moore, 1996). A hybrid powertrain is comprised of electric motors with power electronic converters, energy storage devices such as batteries and ultracapacitors, and sophisticated controllers, in addition to such classical components as internal combustion engines, transmissions, clutch, drive shafts, differentials, etc. Therefore, a hybrid powertrain is much more complicated than a conventional powertrain. The component sizing and system prototyping of a hybrid powertrain is difficult because of the many design options and the rapidly developing technologies in the automotive industries (Miller, 2003). The cost and performance of the designed hybrid powertrain are determined by the chosen configuration and hundreds of design variables and parameters. Engineering design optimisation can refine a rough design so as to maximise fuel economy and minimise emission, weight and cost. In the meantime, vehicle performance requirements must be satisfied (Fellini et al., 1999; Gao and Porandla, 2005).

There are a variety of optimisation algorithms available. They can be categorised in different ways; for example, local optimisation algorithm versus global optimisation algorithm or deterministic optimisation algorithm versus stochastic optimisation algorithm or gradient-based algorithm versus derivative-free algorithm. A good selection of optimisation algorithms for the application of hybrid powertrain design is not very obvious. In this paper, four optimisation algorithms are thoroughly investigated in the design optimisation of an example parallel hybrid electric vehicle. Since the analytical expression of the objective function does not exist, a vehicle simulation model is used for function evaluations.

This paper explores the feasibility of different global optimisation algorithms by comparing their performance and accuracy. The rest of the paper is organised as follows: Section 2 reviews the principles and procedures of four global optimisation algorithms. Section 3 presents the methodology of the model-in-the-loop design process used for this study. In Section 4, the constrained HEV design optimisation problem is set up. Section 5 provides the HEV design optimisation results from the different algorithms and the associated comparison. Finally, conclusions and discussions are given in Section 6.

2 Global optimisation algorithms for HEV design

The response function of a parallel HEV is multi-modal (involving many local minima), and sometimes noisy and discontinuous (Fellini et al., 1999). Gradient based algorithms such as sequential quadratic programming (SQP) (Schittkowski, 1985) use the derivative information to find the local minima. The major disadvantage of local optimisers is that they do not search the entire design space and cannot find the global minimum. Derivative-free algorithms such as DIRECT (Jones, 2001; Jones et al., 1993), simulated annealing (SA) (Kirkpatrick et al., 1983), genetic algorithm (GA) (Holland, 1975), and particle swarm optimisation (PSO) (Kennedy and Eberhart, 1995; Trelea, 2003) do not rely on the derivatives and can, therefore, work exceptionally well when the objective function is noisy and discontinuous. Derivative-free methods are often the best global algorithms because they must often sample a large portion of the design space to be successful. A comparison of the gradient-based and the derivative-free algorithms for the optimisation of a hybrid electric vehicle is given in Fellini et al. (1999) and Wipke and Markel (2001). Note here that even though DIRECT, SA, GA, and PSO algorithms search the design space globally, the main difference is that DIRECT is a deterministic algorithm, whereas SA, GA, and PSO are stochastic algorithms.

2.1 DIRECT Algorithm

DIRECT (DIvided RECTangles) is a sampling algorithm, developed by Donald R. Jones (2001). This global optimisation algorithm is a modification of the standard Lipschitzian approach that eliminates the need to specify the Lipschitz constant (Jones et al., 1993). The Lipschitz constant is a weighing parameter, which decides the emphasis on the global and the local search (Jones et al., 1993). The use of the Lipschitz constant is eliminated in Jones (2001) by searching all possible values for the Lipschitz constant, thus putting a balanced emphasis on both the global and the local search.

The algorithm begins by scaling the design box to an n-dimensional unit hypercube. DIRECT initiates its search by evaluating the objective function at the centre point of the hypercube. DIRECT then divides the potentially optimal hyper-rectangles by sampling the longest coordinate directions of the hyper-rectangle. The sampling is done such that each sampled point becomes the centre of its own n-dimensional rectangle or box. This division continues until termination (pre-specified iteration limit is reached) or convergence is achieved. The division of rectangles in the first three iterations of a two dimensional problem is illustrated in Figure 1, where d represents the centre to vertex distance and each centre point is labelled with a numeral for identifying the rectangles.

In the first iteration, the unit hypercube is trisected into three rectangles. The objective function value is evaluated at the centre points of the three resulted rectangles. The objective function values are plotted against the centre – vertex distance as shown in Figure 2(a). Then the rectangle with the least objective value in each column of dots, which represent the design points, is selected as the optimal rectangle. In the first iteration there is only one column of dots; therefore rectangle 1 is selected as the optimal rectangle and trisected in the second iteration. Similarly in the second iteration, rectangles 4 and 2 have the least objective function values, as shown in Figure2(b). These two rectangles are selected as potential optimal rectangles and trisected in the third iteration. This process is continued until the maximum number of function evaluations is exhausted.

Figure 1 The first three iterations of the DIRECT algorithm

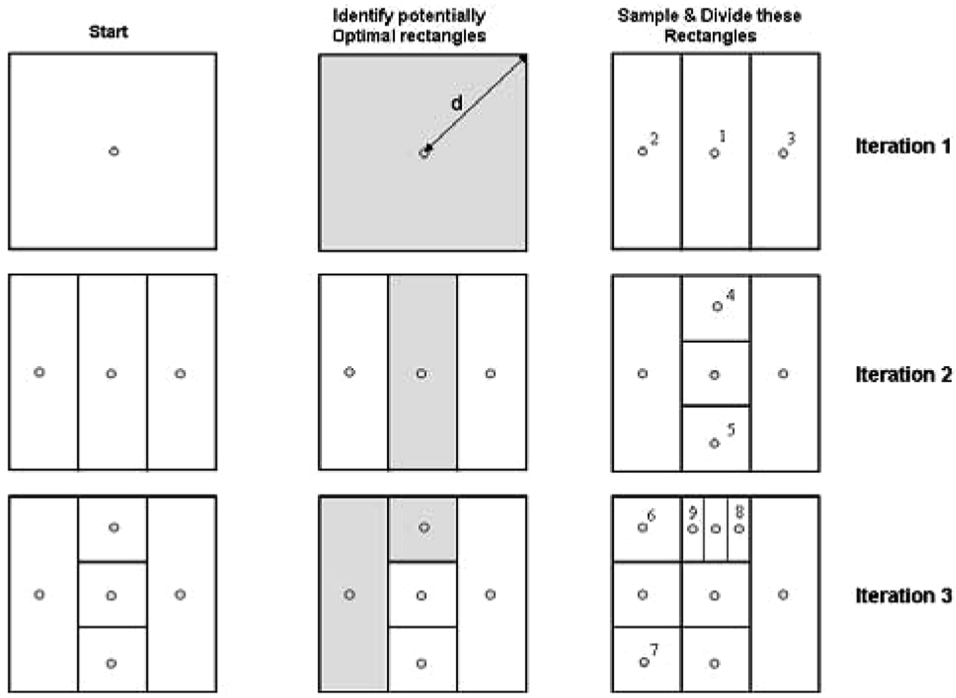
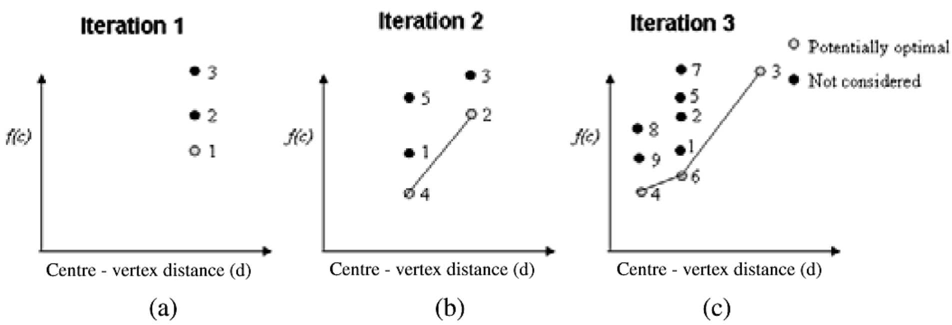


Figure 2 Rectangles selected by DIRECT for further subdivision



2.2 Simulated Annealing

Simulated Annealing belongs to the class of stochastic algorithms, which means that they follow a random path in every searching process for global optimum. A simulated annealing algorithm, based on the Metropolis Monte Carlo Simulation proposed by Kirkpatrick et al. (1983) is used for this study. As the name suggests, this algorithm is based on the analogy of the annealing process of metals. When metals are at a high temperature, the atoms can move relatively freely, but as the temperature is decreased slowly, the atoms can move freely enough to begin adopting the most stable orientation, by taking the lowest possible energy state. Attaining the lowest possible state can be thought of as reaching the global minimum in the optimisation process.

The algorithm starts by evaluating the objective function at a random design point. From this design point, the algorithm jumps to a new random design point and evaluates the objective function value and feasibility. If the current point is better than the previous point, then the current point is accepted to be the potentially optimal point and if the current point is worse than the previous point then its acceptance or rejection depends on the Metropolis probability P criterion given below:

$$P(f, T) = \mathrm{e}^{\left[\frac{f_{new} - f_{current}}{T}\right]}, \tag{1}$$

where f is the objective function value of the optimisation problem and T is the temperature.

From the above equation, it can be seen that the new point is more likely to be accepted, if the new design point function value is close to the current design point function value. And also, the probability of acceptance is more when the temperature is high. Note here that the system design may move to the new design point even when it is worse (has a higher function value) than the current one. It is this feature that prevents the method from becoming stuck in a local minimum. This shows that the simulated annealing algorithm does a global search initially when the temperature is high where even worse design points are more likely to be accepted, and switches to local search when the temperature is decreased, where worse design points are less likely to be selected. Thus, the switching from the global search to the local search depends on the value of the temperature. Another parameter which is responsible for the switching from the global to local search is the maximum step size. This process of selection is continued as the temperature is decreased by a certain factor until the pre-specified number of iterations or the convergence criteria are met.

2.3 Genetic Algorithm

Genetic Algorithms (Holland, 1975) are based on evolutionary processes and Darwin's theory of natural selection. In this selection, only the fittest populations survive, while the less fit populations are tossed out. During the process, several natural processes like crossover, mutation and natural selection are used for selecting the best-fit population. The same concept is extended to mathematical optimisation problems, where only good design points are selected, while less suitable design points are neglected. In this context, the objective function is usually referred to as a fitness function, and the process of 'survival of the fittest' implies a minimisation (or maximisation) procedure. GAs begin by randomly generating, or seeding, an initial population of candidate solutions. Starting with the initial random population, GA then applies a sequence of operations like the design crossover where two individuals from the initial population (parents) are reproduced to get two new individuals (children) and mutation where one individual from the initial population is slightly changed to get a new individual. Then the worst designs are weeded out from the population in order to improve the fitness function. The entire process outlined above can be termed as one generation and is continued for several generations to further improve the fitness function. This process is continued until some termination criteria are satisfied or for a pre-specified number of generations.

2.4 PSO

Particle Swarm Optimisation (PSO) is an evolution-based stochastic global optimisation technique developed by Kennedy and Eberhart (1995) (Trelea, 2003). PSO is based on the swarm intelligence found in natural systems. Such systems are typically made up of a population (swarm) of simple agents or particles interacting locally with one another and with their environment. Bird flocking, ant colonies, and animal herding. are a few examples of such natural systems. In these systems, the local interactions between the agents such as changing the position and velocity lead to the global behaviour. The same technique can be applied in the optimisation problems to find global minima (or maxima).

PSO starts by initialising random design points, called particles, in the multi-dimensional design space. In a PSO system, each particle flies in the multi-dimensional design space looking for the global optimum. Each particle in the PSO is defined by a point in the design space called *position* and its flight speed called *velocity*. In addition, each particle is aware of its best position reached so far (*pbest*) and the best position of the group so far (*gbest*). During flight, each particle adjusts its position according to its own experience (*pbest* value), and according to the experience of its neighbouring particles (*gbest* value). The position is modified using the concept of velocity. The velocity of each particle is updated as follows:

$$v_i^{n+1} = kv_i^n + \alpha_1 rand_1 \left(pbest_i - p_i^n\right) + \alpha_2 rand_2 \left(gbest - p_i^n\right), \tag{2}$$

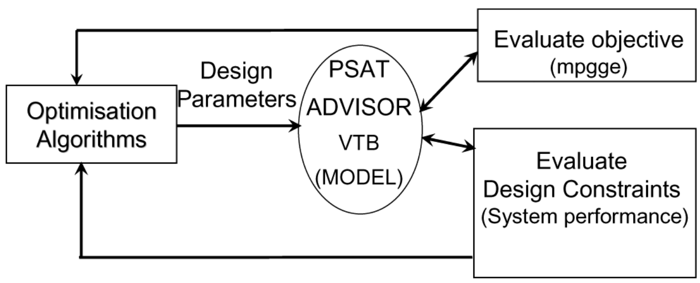
where v_i^{n+1} is the velocity of the particle i at iteration $(n+1)$; k is the weighing function; α_1 and α_2 are the weighing factors; $rand_1$ and $rand_2$ are two random numbers between 0 and 1; p_i^n is the position of the particle i at iteration n; $pbest_i$ is the best position of the particle i; *gbest* is the best position of the group (best of all *pbests*). Similarly, the position is updated towards the *gbest* position as follows:

$$p_i^{n+1} = p_i^n + v_i^{n+1}. \tag{3}$$

3 Model-in-the-loop design optimisation process

The approach that we used is a model-in-the-loop design optimisation process, as illustrated in Figure 3. In the middle of the diagram, the vehicle is modelled in a simulation tool such as PSAT,[1] ADVISOR (Wipke et al., 1999), or VTB (Gao et al., 2004), and this model is embedded in a computational loop. Initially, the vehicle model is simulated using the initial values of the design variables; and we get the numerical values of the objective function, in this case, the composite fuel economy in terms of mpgge (miles per gallon gasoline equivalent). In the meantime, the constraint functions, in this case, the vehicle performance, are evaluated. These simulated results are then fed back to the optimisation algorithm, which generates a new set of values for the design variables. Subsequently, the vehicle model is simulated again to get the values for the objective function and the constraint functions. The simulation results are fed back to the optimisation algorithm, again to generate yet another new set of design variables. This iteration process goes on and on until we have reached some stopping criteria for the optimisation process. Notice that the design variables are restricted within their bounds during this process.

Figure 3 Model-in-the-loop design optimisation process



For this study, powertrain system analysis toolkit (PSAT) is used as the modelling and simulation tool. PSAT has been developed by the Argonne National Laboratory and sponsored by the US Department of Energy (DOE).[1] It can help a vehicle designer to size components and develop realistic hybrid powertrain and its control system. PSAT can accurately simulate vehicle performance, fuel economy and emissions. In using PSAT, we mainly need to select powertrain topology, define component sizes, and construct control strategy. Note that the component sizing is automated by the model-in-the-loop process.

4 HEV design optimisation problem setup

As an application example, PSAT is used to optimise a parallel HEV for maximum fuel economy on a composite driving cycle. Four global algorithms, DIvided RECTangle (DIRECT), simulated annealing (SA), genetic algorithm (GA), and particle swarm optimisation (PSO) are used in the model-based design optimisation. The vehicle model 'gui_par_midsize_cavalier_ISG_in' (available in the PSAT model library) has been chosen for this optimisation study. This vehicle is a two wheel-drive starter-alternator parallel configuration with manual transmission. The basic configuration of the parallel HEV used for simulation study is illustrated in Figure 4 and main components of the HEV are listed in Table 1.

Figure 4 Configuration of the selected parallel HEV in PSAT

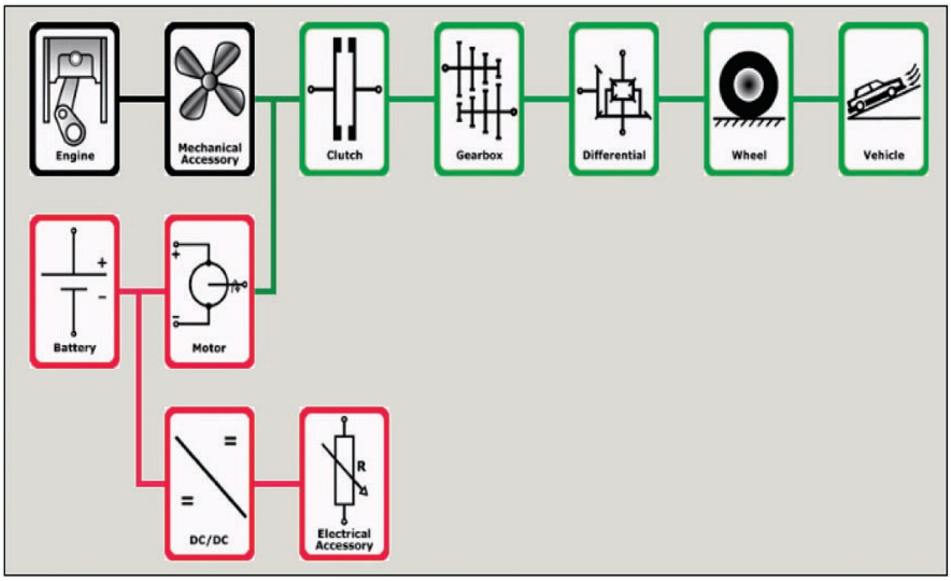


Table 1 Parallel HEV components

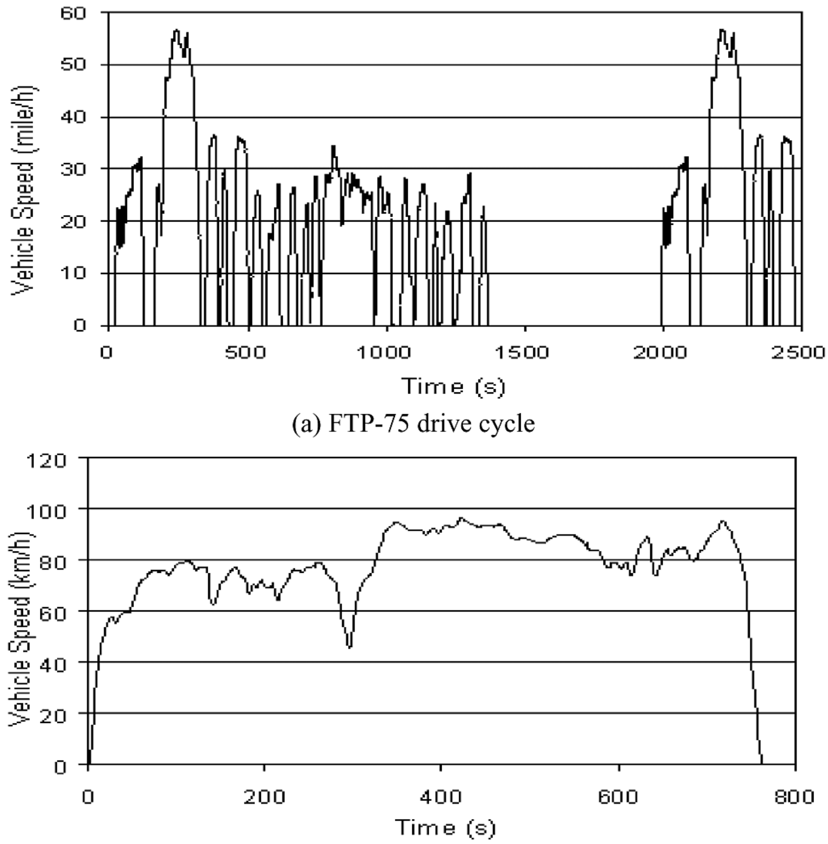
Component	*Description*
Fuel Converter	84 kW and 2.2 L Cavalier gasoline engine
Motor	ECOSTAR motor model with continuous power of 33 kW and peak power of 66 kW
Battery	Panasonic NiMH battery with capacity 6.5 Ah and 240 cells
Transmission	Four speed manual gearbox with final drive ratio 3.63
Control strategy	Default propelling, shifting and braking strategies

The objective is to maximise the composite fuel economy, which is computed based on city fuel economy and highway fuel economy. By definition, composite fuel economy is the harmonic average of the SOC-balanced fuel economy values during the two separate drive cycles (Wipke et al., 2001). Specifically, the composite fuel economy is calculated as given by the following formula:

$$\text{composite fuel economy} = \frac{1}{\dfrac{0.55}{City\_FE} + \dfrac{0.45}{Hwy\_FE}}, \tag{4}$$

where *City_FE* and *Hwy_FE* represent the city and highway fuel economy values respectively. The driving cycle is composed of city driving represented by FTP-75 (Federal Test Procedure) and highway driving represented by HWFET (Highway Fuel Economy Test). The two drive cycles are shown in Figures 5(a) and 5(b), respectively.

Figure 5 The drive cycles



(a) FTP-75 drive cycle

(b) HWFET drive cycle

The following vehicle performance constraints are imposed on the design problem.

Acceleration time	0–60 mph	$<=$ 18.1 s
Acceleration time	40–60 mph	$<=$ 7 s
Acceleration time	0–85 mph	$<=$ 35.1 s
Maximum acceleration	$>= 3.583\,\mathrm{m/s^2}$	

Table 2 shows the six design variables used in this study. The first two define the power ratings of the fuel converter (the engine) and motor controller. The third, fourth and fifth variables define the number of battery modules, minimum battery state of charge (SOC) allowed and maximum battery SOC allowed. Note that the SOC values are part of the control strategy parameters. Although they are not related to component-sizing, they have a direct impact on the fuel economy of an HEV design. The sixth design variable defines final drive ratio. Each design variable is also restricted within a lower and an upper bound.

Table 2 Upper and lower bounds of design variables

Design variable	*Description*	*Lower bound*	*Upper bound*
eng.scale.pwr_max_des	Fuel converter power rating	40 kW	100 kW
mc.scale.pwr_max_des	Motor controller power rating	10 kW	80 kW
ess.init.num_module	Battery number of cells	150	350
ess.init.soc_min	Minimum SOC allowed	0.2	0.4
ess.init.soc_max	Maximum SOC allowed	0.6	0.9
fd.init.ratio	Final drive ratio	2	4

5 HEV design optimisation results in PSAT

The problem now becomes quite challenging since this is a constrained multi-variable optimisation problem.

Firstly, the default vehicle with the initial values of design variables given in Table 3 is simulated in PSAT. The fuel economy was observed to be 35.1 mpg as given in Table 4 under the first column.

Table 3 Initial design variable values

Design variable	*Initial value*
eng.pwr_max_des	86 kW
mc.pwr_max_des	65.9 kW
ess.init.num_module	240
ess.init.soc_min	0.2
ess.init.soc_max	0.9
fd.init.ratio	3.63

Table 4 Comparison of fuel economy

Fuel economy				
	After optimisation			
Before optimisation	*DIRECT*	*SA*	*GA*	*PSO*
35.1 mpg	39.64 mpg	40.37 mpg	37.6 mpg	37.1 mpg

Secondly, the optimisation algorithms, DIRECT, Simulated Annealing, Genetic Algorithms, and PSO, are looped with the PSAT Vehicle Simulator and the optimisation is carried on. For this step, the same default vehicle configuration given in Figure 4 and Table 1 is taken and the bounds for the design variables are taken as given in Table 2. The four algorithms are allowed to run for 400 function evaluations. Using the same number of function evaluations will allow us to compare the performance of the different algorithms. A comparison of the fuel economy before and after the optimisation is given in Table 4. A significant improvement in the fuel economy is seen due to optimisation (to a lesser extent in the case of PSO and GA, though). Of all the four algorithms, SA performs well with an approximate improvement of 5 mpg.

Table 5 shows the final values of the six design variables after optimisation. We can notice that the rating of the electric motor is greatly reduced, implying that down-sizing of the electric motor has been achieved. On the other hand, the engine is down-sized to a lesser extent in DIRECT and SA cases, while up-sized in GA and PSO cases. Given the vehicle performance constraints, the trade-off of engine down-sizing and motor down-sizing can be realised by adjusting the lower and upper bounds of the design variables.

Table 5 Final design variable values

		Final value			
Design variable	*Initial value*	*DIRECT*	*SA*	*GA*	*PSO*
eng.pwr_max_des	86 kW	83.1 kW	82.4 kW	95.5 kW	87.1 kW
mc.pwr_max_des	65.9 kW	20.2 kW	21.9 kW	24.2 kW	14.8 kW
ess.init.num_module	240	245	311	300	238
ess.init.soc_min	0.2	0.25	0.22	0.34	0.26
ess.init.soc_max	0.9	0.84	0.78	0.89	0.78
fd.init.ratio	3.63	3.9	4.0	3.49	3.42

Table 6 shows the performance results of the hybrid powertrain after optimisation. Essentially, all the optimisation algorithms resulted in improved vehicle performance.

Table 6 Comparison of the HEV performance

			After opt.			
Constraint	*Constr. value*	*Before opt.*	*DIRECT*	*SA*	*GA*	*PSO*
0–60 mph	<=18.1 s	18.1 s	15.5 s	10.8 s	11.9 s	11.1 s
40–60 mph	<=7 s	7 s	6.8 s	5 s	4.4 s	4.9 s
0–85 mph	<=35.1 s	35.1 s	30.6 s	20.7 s	21.2 s	20 s
Max. accel.	>=3.583 m/s^2	3.583 m/s^2	3.97 m/s^2	4.07 m/s^2	3.94 m/s^2	3.99 m/s^2

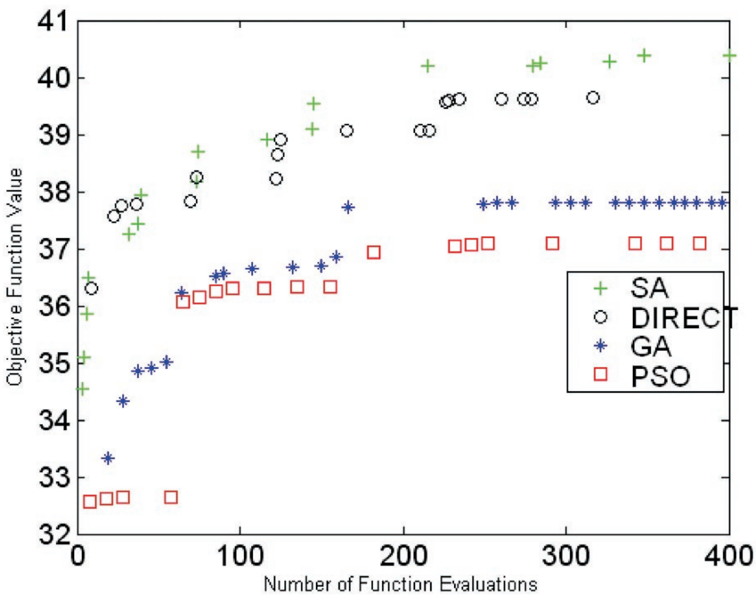
The mass of the vehicle changes as the design variables change because the mass of the vehicle depends directly on some design variables. In particular, of the chosen six design variables, three design variables (power ratings of engine and motor and energy modules) affect the mass of the vehicle. The mass of the vehicle before and after the optimisation is given in Table 7. The vehicle mass decreased for DIRECT and SA cases, while the vehicle weight increased slightly in the case of GA and PSO.

Table 7 Mass of HEV before and after optimisation

Mass of the vehicle				
	After optimisation			
Before optimisation	*DIRECT*	*SA*	*GA*	*PSO*
1683 kg	1635 kg	1656 kg	1694 kg	1690 kg

Figure 6 shows how objective function (fuel economy) value improves against the design iteration number. The cross curve is for the SA case; the circle curve is for the DIRECT case; the star curve is for the GA case; and the square curve is for the PSO case. We can see that the fuel economy improvement with the SA and DIRECT algorithms is very close until about 125 function evaluations, after which SA leapt ahead of DIRECT. GA is slow to catch SA and DIRECT initially because it takes some function evaluations to generate the initial populations. After about 200 function evaluations, GA did not find any better design point to get further improvement in the fuel economy. The performance of PSO is similar to that of GA. Overall; SA performed the best for this particular design optimisation problem.

Figure 6 Performance comparison of DIRECT, SA, GA, and PSO



6 Conclusion and discussions

Based on the optimisation results, the following observations can be made. The fuel economy of the parallel HEV is increased from 35.1 to 39.64 mpg with the DIRECT algorithm, while from 35.1 to 40.37 mpg with the SA algorithm. The performance of the optimised HEV shows a great improvement. The power rating of the traction motor is reduced significantly.

In this study, only global optimisation algorithms are tested for a hybrid vehicle design, and generally have slower convergence. On the other hand, derivative-based algorithms are known for their faster convergence. In fact, a hybrid optimisation algorithm can be used that combines the benefits of both a global and a local algorithm. The global algorithm can reach a design point near the global optimum region after a certain number of optimisation steps. Then a local algorithm kicks in and the process is continued until a global optimum is found.

The design optimisation takes about 100 hours running PSAT on a single PC. This long design time necessitates the development of a more efficient optimisation methodology, such as using parallel and distributed computing, which is part of our ongoing research effort.

References

Fellini, R., Michelena, N., Papalambros, P. and Sasena, M. (1999) 'Optimal design of automotive hybrid powertrain systems', *Proceedings of EcoDesign 99 – First Int. Symp. On Environmentally Conscious Design and Inverse Manufacturing*, H. Yoshikawa et al. (Eds), Tokyo, Japan, February 1999, pp.400–405.

Gao, W. and Porandla, S. (2005) 'Design optimization of a parallel hybrid electric powertrain', *Proceedings of the IEEE Vehicle Power and Propulsion Conference*, pp.530–535, Chicago, IL, September.

Gao, W., Solodovnik, E. and Dougal, R. (2004) 'Symbolically-aided model development for an induction machine in virtual test bed', *IEEE Transactions on Energy Conversion*, Vol. 19, No. 1, pp.125–135, March.

Holland, H.J. (1975) *Adaptation in Natural and Artificial Systems*, The University of Michigan.

Jones, D.R. (2001) 'DIRECT Global optimization algorithm', *Encyclopedia of Optimization*, Kluwer Academic Publications.

Jones, D.R., Perttunen, C.D. and Stuckman, B.E. (1993) 'Lipschitz optimization without Lipschitz Constant', *Journal of Optimization Theory and Applications*, Vol. 79, No. 1, Oct.

Kennedy, J. and Eberhart, R. (1995) 'Particle swarm optimization', *Proc. IEEE Intl. Conference on Neural Networks*, Vol. IV, pp.1942–1948.

Kirkpatrick, S., Gelett, C. and Vechhi, M. (1983) 'Optimization by simulated annealing', *Science*, Vol. 220, No. 4598, pp.671–680, May.

Miller, J. (2003) *Propulsion Systems for Hybrid Vehicles*, Peter Peregrinus Ltd.

Moore, T.C. (1996) 'Tools and strategies for hybrid-electric drive system optimisation', *SAE Technical Paper Series*, No. SP-1189.

Schittkowski, K. (1985) 'NLQPL: A FORTRAN – Subroutine solving constrained nonlinear programming problems', *Annals of Operations Research*, Vol. 5, pp.485–500.

Trelea, I.C. (2003) 'The particle swarm optimization algorithm: convergence analysis and parameter selection', *Information Processing Letters*, Vol. 85, No. 6, Mar.

Wipke, K. and Markel, T. (2001) 'Optimization techniques for hybrid electric vehicle analysis Using ADVISOR', *Proc. ASME, International Mechanical Engineering Congress and Exposition*, New York.

Wipke, K., Markel, T. and Nelson, D. (2001) 'Optimizing energy management strategy and a degree of hybridization for a hydrogen fuel cell SUV', *18th Electric Vehicle Symposium (EVS-18)*, Berlin, Germany, Oct. 20–24.

Wipke, K.B., Cuddy, M.R. and Burch, S.D. (1999) 'ADVISOR 2.1: A user-friendly advanced powertrain simulation using a combined backward/forward approach', NREL/JA-540-26839, Sept.

Note

[1] PSAT documentation (online), available: http://www.transportation.anl.gov/software/PSAT.