Honda Research Institute EU
Honda Research Institute Europe GmbH
https://www.honda-ri.de/

Evolutionary Many-objective Optimization of Hybrid Electric Vehicle Control: from General Optimization to Preference Articulation

Ran Cheng, Tobias Rodemann, Michael Fischer, Markus Olhofer, Yaochu Jin

2017

Preprint:

This is an accepted article published in IEEE Transactions on Emerging Topics in Computational Intelligence. The final authenticated version is available online at: https://doi.org/10.1109/TETCI.2017.2669104

Evolutionary Many-objective Optimization of Hybrid Electric Vehicle Control: From General Optimization to Preference Articulation

Ran Cheng, Tobias Rodemann, Michael Fischer, Markus Olhofer, and Yaochu Jin, *Fellow, IEEE*

***Abstract*—Many real-world optimization problems have more than three objectives, which has triggered increasing research interest in developing efficient and effective evolutionary algorithms for solving many-objective optimization problems. However, most many-objective evolutionary algorithms have only been evaluated on benchmark test functions and few applied to real-world optimization problems. In solving benchmark problems, most algorithms aim to find a representative subset of the Pareto optimal solutions, which becomes increasingly challenging as the number of objectives increases. To move a step forward, this paper presents a case study of solving a many-objective hybrid electric vehicle controller design problem using three state-of-the-art evolutionary algorithms, namely, a decomposition based evolutionary algorithm (MOEA/D), a non-dominated sorting based genetic algorithm (NSGA-III), and a reference vector based evolutionary algorithm (RVEA). We start with a typical setting aiming at approximating the Pareto front without introducing any user preferences. Based on the analyses of the approximated Pareto front, we introduce a preference articulation method and embed it in the three evolutionary algorithms for identifying solutions that the decision-maker prefers. Our experimental results demonstrate that by incorporating user preferences into many-objective evolutionary algorithms, we are not only able to gain deep insight into the trade-off relationships between the objectives, but also to achieve high-quality solutions reflecting the decision-maker's preferences. In addition, our experimental results indicate that each of the three algorithms examined in this work has its unique advantages that can be exploited when applied to the optimization of real-world problems.**

***Index Terms*—Many-objective optimization, hybrid electric vehicle, preference articulation, reference vector, evolutionary algorithm**

I. Introduction

MANY real-world optimization problems involve more than one conflicting objective to be optimized, known as multiobjective optimization problems (MOPs) [1]–[3]. Generally, a minimization MOP can be formulated as follows:

$$\begin{aligned} &\text{minimize } \mathbf{f}(\mathbf{x}) = (f_1(\mathbf{x}), f_2(\mathbf{x}), ..., f_M(\mathbf{x})) \\ &\text{s.t. } \mathbf{x} \in X, \quad \mathbf{f} \in Y \end{aligned} \tag{1}$$

Ran Cheng and Yaochu Jin are with the Department of Computer Science, University of Surrey, Guildford, Surrey, GU2 7XH, United Kingdom (e-mail: {r.cheng;yaochu.jin}@surrey.ac.uk). Y. Jin is also with the College of Information Sciences and Technology, Donghua University, Shanghai 201620, P. R. China. *(Corresponding author: Yaochu Jin)*

Tobias Rodemann and Markus Olhofer are with the Honda Research Institute Europe GmbH, 63073 Offenbach, Germany (e-mail: {tobias.rodemann;markus.olhofer}@honda-ri.de).

Michael Fischer is with Honda R&D Europe Germany GmbH, 63073 Offenbach, Germany (email: michael_fischer@de.hrdeu.com)

where $X \subset \mathbb{R}^n$ and $Y \subset \mathbb{R}^M$ are known as the decision space and objective space, respectively, with $\mathbf{x} = (x_1, x_2, ..., x_D) \in X$ and $\mathbf{f} \in Y$ denoting the decision vector and objective vector in the two spaces, D and M are the number of decision variables and the number of objectives, respectively. Due to the conflicts between different objectives as formulated in (1), there does not exist one single solution that is able to optimize all objectives simultaneously. Instead, a set of solutions, known as the Pareto optimal solutions, can be achieved as the trade-offs between different objectives. To be specific, the Pareto optimal solutions are also known as the Pareto set (PS) in the decision space, while its image in the objective space is also known as the Pareto front (PF).

Evolutionary algorithms (EAs), which are population based, are well suited for multiobjecitve optimization in that they are able to achieve a set of solutions in one run. As most popular multiobjective evolutionary algorithms (MOEAs), e.g. the fast non-dominated sorting based genetic algorithm (NSGA-II) [4], the improved strength Pareto evolutionary algorithm (SPEA2) [5], a region based selection algorithm (PESA-II) [6], among many others [7], were originally proposed for solving MOPs which involve two or three objectives, the performance of these MOEAs deteriorates seriously when the number of objectives becomes more than three [8]–[11]. Nowadays, MOPs with more than three objectives are often called many-objective optimization problems (MaOPs) [12]. In order to enhance the performance of MOEAs on MaOPs, various approaches haven been proposed in recent years, which can be roughly categorized into the following three classes [13]–[15].

The first class is to enhance the convergence capability of traditional MOEAs. One major challenge in many-objective optimization is known as dominance resistance, resulting in loss of convergence pressure in MOEAs. Therefore, the most straightforward idea is to adapt MOEAs to MaOPs is to enhance the convergence pressure. Among various convergence enhancement approaches, dominance modification is the most intuitive one. Examples of modified dominance definitions include L-optimality [16], ϵ-dominance [17], [18], fuzzy dominance [19], and grid based dominance [20]. Another typical idea of this category is to introduce new convergence metrics in addition to traditional dominance based mechanisms. For example, a shift-based density estimation strategy is proposed to penalize poorly converged solutions that cannot be distinguished by Pareto dominance [21]; a knee point based secondary selection is introduced on top of non-dominated sorting to enhance convergence pressure in the recently proposed knee

point driven evolutionary algorithm (KnEA) [22]. Some other recent work along this line includes the two-archive algorithm for many-objective optimization (Two_Arch2) [23] and the many-objective evolutionary algorithm based on both objective space reduction and diversity enhancement (MaOEA-R&D) [24].

The second class is decomposition based MOEAs that decompos a complex MOP into a number of simpler subproblems and solve them collaboratively [25], [26]. Decomposition based MOEAs can be further categorized into two groups [27]. The first class decomposes an MOP into a group of single-objective problems (SOPs), such as dynamic weighted aggregation (DWA) [28], cellular multi-objectige genetic algorithm (C-MOGA) [29] and multi-objective evolutionary algorithm based on decompostion (MOEA/D) [25], [30], [31]. The second class decomposes an MOP into a group of sub-MOPs, such as MOEA/D-M2M [26], [32] and MOEA/DD [33], which are variants of MOEA/D, a variant of NSGA-II, termed NSGA-III [34], and a recently proposed reference vector guided evolutionary algorithm (RVEA) [35]. It is worthing note that NSGA-III and RVEA can also be seen as preference based MOEAs, as suggested in [35], since they can easily be used for locate user preferred solutions.

The third class is performance indicator based MOEAs. Performance indicators are originally designed to evaluate the quality of solution sets but later found to be useful as a selection criterion in MOEAs. Representative algorithms of this class include the S-metric selection based evolutionary multi-objective algorithm (SMS-EMOA) [36], the indicator based evolutionary algorithm (IBEA) [37] and the fast hypervolume based evolutionary algorithm (HypE) [38]. Although this class of MOEAs do not suffer from dominance resistance, unfortunately, the computational cost for indicator calculation becomes prohibitively as the number of objectives increases [39]–[41].

Most of MOEAs for solving MaOPs have been designed mainly to achieve a representative subset of the PF, which is termed general optimization in this work. In real-world engineering optimization, it is particularly useful if an algorithm is able to incorporate preferences of decision makers, especially in many-objective optimization, where approximation of a high-dimensional PF with a limited number of solutions becomes impractical.

In the literature, however, most existing preference articulation approaches have been developed only for optimization problems having two or three objectives [42]. These algorithms include the fuzzy preference based approaches [43], [44], the reference based approaches [45]–[48], and the recently proposed inverse modelling based approaches [49]–[51]. By contrast, there is little study focused on preference articulation in many-objective optimization with few exceptions [52]–[54].

In summary, research on evolutionary many-objective optimization is still in its infancy and there is a big gap between academic research and demand from industry for the following main reasons. Firstly, most algorithms have merely been assessed on benchmark test problems for general optimization, where the performance is mostly evaluated using performance indicators originally proposed for multiobjective optimization with two or three objectives. Secondly, existing preference articulation approaches remain to be tested on real-world problems. Thus, there is a strong need to demonstrate the capability of the proposed MOEAs for MaOPs to solve real-world problems.

To bridge the gap between academic research and industrial applications, this paper presents a case study where three state-of-the-art MOEAs, namely, MOEA/D [25], NSGA-III [34], and RVEA [35] are employed to solve a seven-objective optimization problem in hybrid electric vehicle (HEV) control. By proposing a generic preference articulation method, we demonstrate that the three algorithms are not only able to perform general optimization, but also capable of identifying optimal solutions preferred by the decision-maker. The main new contributions of this work can be summarized as follows.

1) The paper exemplifies a successful application of MOEAs for real-world many-objective optimization. As previously mentioned, most MOEAs have only been assessed in general optimization on benchmark test problems using performance indicators. In practice, however, it is more useful if user preferences can be articulated in the optimization process. In this case study, we illustrate how existing MOEAs can meet such demands in the optimization of a seven-objective HEV controller design problem.
2) As an extension of our previous work, this paper details a seven-objective HEV controller model and demonstrates why the seven-objective optimization model can offer additional value compared to traditional models that consider a single objective. We empirically show that the improvement potential of the additional objectives is much larger than the potential for fuel consumption by using the state-of-art evolutionary many-objective optimization algorithms.
3) A preference articulation approach is proposed to enable the three state-of-the-art MOEAs for many-objective optimization to obtain solutions in the preferred regions of the objective space specified by the decision-maker. With the successful application of the three MOEAs assisted by the proposed preference articulation approach, the decision-maker is able to achieve high-quality solutions that significantly improve the performance of the HEV controller design, gleaning useful knowledge in design of HEV controllers.

The remainder of this paper is organized as follows. Section II presents some background knowledge, including an introduction to three MOEAs adopted in this work, namely, MOEA/D, NSGA-III and RVEA. Section III presents the preference articulation approach to be embedded in the three MOEAs. Section IV details the seven-objective HEV controller model. Section V presents the optimization results obtained, including the analyses of the results from general optimization, as well as the results when preferences are articulated. Finally, Section VI draws the conclusion.

II. Background

As background knowledge, this section presents a brief introduction to MOEA/D [25], NSGA-III [34] and RVEA [35].

The framework of each algorithm can be found in Appendix A, and more details of the three algorithms can be found in the original publications.

A. MOEA/D

MOEA/D is a representative decomposition based algorithm originally proposed for solving multiobjective optimization problems with two or three objectives [25]. Recently, it has been reported that the performance of MOEA/D is promising for many-objective optimization as well [27], [33]. As shown by Algorithm 1 in Appendix A, MOEA/D adopts a *steady-state* selection strategy [55], where there is only one offspring candidate solution created at each time in the reproduction process. In addition, both reproduction and selection in MOEA/D are performed locally in the predefined neighborhoods specified by a set of weight vectors.

The most important step in Algorithm 1 is the update process in Line 8. In this step, the algorithm updates each neighborhood by replacing the candidate solutions in it with the one having the best scalarization function value, where a scalarization function is used to transform a vector of objective values into a scalar fitness value. Two most commonly used scalarization functions are the Weighted Tchebycheff (TCH) scalarization function and the penalty-based boundary intersection (PBI) scalarization function, which can be formulated as follows:

1) Weighted Tchebycheff:

$$\text{minimize } g^{tch}(\mathbf{x}|\mathbf{w},\mathbf{z}^{min}) = \max_{i=1}^{M}\{w_i|f_i(\mathbf{w}) - z_i^{min}|\}, \quad (2)$$

2) PBI:

$$\text{minimize } g^{pbi}(\mathbf{x}|\mathbf{w},\mathbf{z}^{min}) = d_1 + \theta d_2, \quad (3)$$

with

$$d_1 = \frac{\|(\mathbf{F}(\mathbf{x}) - \mathbf{z}^{min})^\top \mathbf{w}\|}{\|\mathbf{w}\|}, \quad (4)$$

and

$$d_2 = \|\mathbf{F}(\mathbf{x}) - (\mathbf{z}^{min} + d_1\frac{\mathbf{w}}{\|\mathbf{w}\|})\|, \quad (5)$$

where $\mathbf{w} = (w_1, w_2, ...w_M)$ is a weight vector, $\mathbf{z} = (z_1^{min}, z_2^{min}, ...z_M^{min})$ is the ideal point, and θ in PBI is a user-specified parameter that balances the weights of d_1 and d_2.

In this work, we apply the PBI scalarization function in optimization of the seven-objective HEV controller model.

B. NSGA-III

NSGA-II is a classic MOEA proposed for solving bi-/three-objective MOPs [4]. Recently, NSGA-II has been extended for many-objective optimization, which was termed NSGA-III [34]. As shown by Algorithm 2 in Appendix A, NSGA-III still applies a similar elite framework as NSGA-II, where the selection is also *two-stage*: the first stage of selection (Line 8 in Algorithm 2) is to maintain convergence pressure to push the candidate solutions towards the PF, and the second stage of selection (Line 10 in Algorithm 2) is to manage population diversity for a wide spreading of the candidate solutions on the PF.

The dominance based selection in NSGA-III is exactly the same as the one in NSGA-II, where a fast non-dominated sorting approach is used to sort the population P_t into a number of non-dominated fronts as (F_1, F_2, ...), according to the dominance relationships between the candidate solutions. Afterwards, the candidate solutions in the first l fronts are selected into S_t which satisfies $|S_t| \geq N$ and $|S_t - F_{l-1}| < N$. In this way, the dominance based selection in NSGA-III guarantees that the candidate solutions with the best convergence are selected with the highest priority.

While the secondary selection in NSGA-II is based on crowding distances, the one in NSGA-III is based on the niching counts which are calculated by associating the candidate solutions with the closest reference line specified by each weight vector. The motivation behind such a niching based selection is that in many-objective optimization, where the candidate solutions are sparsely distributed, setting up fixed references is more efficient in diversity management than dynamically adapting the distribution of the candidate solutions according to the Euclidean distances between them.

Apart from the dominance based selection and the niching based selection, another important component in NSGA-III is the normalization (Line 9 in Algorithm 2). The normalization process in NSGA-III is designed to normalize the objective function values into the same range such that the candidate solutions are still uniformly distributed with respect to the weight vector even on a non-normalized PF where the objective function values are scaled to different ranges.

C. RVEA

RVEA has been recently proposed for many-objective optimization [35]. The basic idea of RVEA[1] is to guide the search of an evolutionary algorithm with a set of predefined reference vectors in the objective space, where each reference vector specifies a direction that a candidate solution should converge towards, and it is expected that each reference vector is finally associated with a unique Pareto optimal solution. As shown by Algorithm 3 in Appendix A, RVEA shares the same framework as traditional elitism based EAs. In the main loop of RVEA, the recombination component is the same as those adopted in many other MOEAs [22], [25], [34], [38], where the simulated binary crossover (SBX) [56] and the polynomial mutation [57] are applied. In addition to the recombination component, there are another two components in the main loop: reference vector guided selection and reference vector adaptation, which are detailed as follows.

The reference vector guided selection in RVEA is designed to perform single-objective selections inside the subpopulations generated by associating the candidate solutions with their closest reference vectors. As the selection criterion, a scalarization function known as the angle penalized distance (APD) is proposed to aggregate the convergence criterion and the diversity criterion:

$$d_{t,i,j} = (1 + P(\theta_{t,i,j})) \cdot \|\mathbf{f}_{t,i} - \mathbf{z}_t^{min}\|, \quad (6)$$

[1]The Matlab source code of RVEA can be downloaded from: http://www.soft-computing.de/jin-pub_year.html

where $d_{t,i,j}$ denotes the APD value of solution $p_{t,i}$ with respect to reference vector $\mathbf{v}_{t,j}$ in the t-th generation; $\|\mathbf{f}_{t,i} - \mathbf{z}_t^{min}\|$, as the convergence criterion, is the Euclidean distance from the candidate solution $p_{t,i}$ to the ideal vector $\mathbf{z}_t^{min}$; and $P(\theta_{t,i,j})$, as the diversity criterion, is a penalty function related to the angle $\theta_{t,i,j}$ between candidate solution $p_{t,i}$ and reference vector $\mathbf{v}_{t,j}$:

$$P(\theta_{t,i,j}) = M \cdot (\frac{t}{t_{max}})^{\alpha} \cdot \frac{\theta_{t,i,j}}{\gamma_{\mathbf{v}_{t,j}}}, \tag{7}$$

with

$$\gamma_{\mathbf{v}_{t,j}} = \min_{i \in \{1,...,N\}, i \neq j} \langle \mathbf{v}_{t,i}, \mathbf{v}_{t,j} \rangle, \tag{8}$$

where M is the number of objectives, t_{max} is the predefined maximum number of generations, $\gamma_{\mathbf{v}_{t,j}}$ is the smallest angle value between reference vector $\mathbf{v}_{t,j}$, and α is a user defined parameter controlling the changing rate of the diversity criterion with respect to the number of generations. It is worth noting that, the design of penalty function $P(\theta_{t,i,j})$ is out of several important empirical observations in many-objective optimization. Firstly, in many-objective optimization, since there are only a limited number of candidate solutions that are sparsely distributed in the high-dimensional objective space, it is difficult to guarantee convergence and diversity at the same time. Therefore, $P(\theta_{t,i,j})$ is designed to be linked to the generation number t. Secondly, due to the sparse distribution of the candidate solutions, the angles between the candidate solutions and reference vectors can vary a lot. Therefore, the angles are normalized into $[0, 1]$ with $\frac{\theta_{t,i,j}}{\gamma_{\mathbf{v}_{t,j}}}$. Thirdly, the more objectives there are, the more sparsely the candidate solutions are distributed. Therefore, $P(\theta_{t,i,j})$ is designed to be related with the number of objectives M.

The reference vector adaption in RVEA is designed to adjust the distribution of the reference vectors according to the different ranges of objectives on the PFs. As pointed out in [35], for dominance based algorithms such as the NSGA-III, normalization of objective functions does not change the partial orders of dominance relations, and thus has no influence on the selection process; while in RVEA, where the selection criterion is related to the specific objective function values of the candidate solutions, normalizing the objective functions will substantially perturb the convergence of the algorithm. Therefore, instead of normalizing the objective functions, Cheng *et al.* [35] have proposed to adapt the reference vectors in the following manner:

$$\mathbf{v}_{t+1,i} = \frac{\mathbf{v}_{0,i} \circ (\mathbf{z}_{t+1}^{max} - \mathbf{z}_{t+1}^{min})}{\|\mathbf{v}_{0,i} \circ (\mathbf{z}_{t+1}^{max} - \mathbf{z}_{t+1}^{min})\|}, \tag{9}$$

where $\circ$ operator denotes the Hadamard product that element-wisely multiplies two vectors (or matrices) of the same size, $i = 1, ..., N$, $\mathbf{v}_{0,i}$ denotes the i-th uniformly distributed reference vector, which is generated in the initialization stage (on Line 1 in Algorithm 3) and $\mathbf{v}_{t+1,i}$ denotes the i-th adapted reference vector for the next generation $t+1$, $\mathbf{z}_{t+1}^{max} = (z_{t+1,1}^{max}, z_{t+1,2}^{max}, ..., z_{t+1,m}^{max})$ and $\mathbf{z}_{t+1}^{min} = (z_{t+1,1}^{min}, z_{t+1,2}^{min}, ..., z_{t+1,m}^{min})$ denote the nadir point and the ideal point estimated with the candidate solutions in population P_{t+1}, respectively. In addition, to control the frequency of the reference vector adaptation, a predefined parameter f_r is introduced.

It is worth noting that, in original RVEA, it may occasionally happen that a subpopulation becomes empty because there is no solution that falls into the subspace specified by the corresponding reference vector. As demonstrated by the empirical results in [35], this does not influence the performance of RVEA as long as most subpopulations are non-empty. However, when RVEA is applied to the seven-objective HEV controller model, it turns out that most subpopulations are empty due to the limited optimization potential for many objectives (e.g. there is no way to reduce fuel consumption to zero). As a consequence, this leads to a severe loss of population diversity, thus causing failure of the algorithm. In this work, we propose a simple and efficient strategy to tackle the above issue: if a subpopulation is empty, the candidate solution with the smallest APD with respect to the corresponding reference vector is selected from the whole population instead of the subpopulation.

D. Discussions

Although the frameworks of MOEA/D, NSGA-III and RVEA show significant differences as in Algorithm 1, Algorithm 2 and Algorithm 3, the three algorithms share a common feature: in the initialization process (Line 1), a set of predefined vectors are required as the input of the algorithms. To be specific, in MOEA/D, the weight vectors are part of the scalarization functions which are used to transform the objective vectors into scalar fitness values; in NSGA-III, each weight vector specifies a niching center, with respect to which the niching based selection is performed; in RVEA, the objective space is divided into several subspaces by a set of reference vectors, and a subpopulation is then related to a subspace.

Since the predefined weight/reference vectors are highly relevant to the selection mechanism in each algorithm, the distribution of these vectors will consequently determine the distribution of the candidate solutions obtained by each algorithm. In general optimization, it is always expected that the candidate solutions can be evenly distributed as an approximation to the true PF. To meet such a requirement, the reference/weight vectors in MOEA/D, NSGA-III and RVEA are usually uniformly sampled using the simplex-lattice design approach [58]. Nevertheless, as pointed out in [35], these algorithms have the inherent capability of preference articulation [48]. More details about preference articulation using weight/reference vectors will be provided below.

III. Generation of Weight/Reference Vectors for General Optimization and Preference Articulation

In the following section, we first present how to generate uniformly distributed weight/reference vectors for general optimization. Then we show how to articulate user preferences by means of weight/reference vectors using the proposed preference articulation approach.

A. Generation of Uniformly Distributed Weight/Reference Vectors

To achieve a subset of representative Pareto optimal solutions, the weight/reference vectors used in MOEA/D and NSGA-III can be uniformly sampled using the simplex lattice design approach as follows [58]:

$$\begin{cases} \mathbf{w_i} = (w_i^1, w_i^2, ..., w_i^M), \\ w_i^j \in \{\frac{0}{H}, \frac{1}{H}, ..., \frac{H}{H}\}, \sum_{j=1}^{M} w_i^j = 1, \end{cases} \tag{10}$$

where $i = 1, ..., N$ with N being the number of uniformly distributed weight vectors, M is the number of objectives, and H is a positive integer for the simplex-lattice design [58] that samples uniformly distributed points on a unit hyperplane. Given M and H, the number of uniformly distributed weight vectors that can be sampled will be $\frac{(M+H-1)!}{H!(M-1)!}$.

Based on the weight vectors sampled using (10), the reference vectors used in RVEA can be generated by mapping the points from the unit hyperplane to a unit hypersphere [50], [59]:

$$\mathbf{v_i} = \frac{\mathbf{w_i}}{\|\mathbf{w_i}\|}. \tag{11}$$

B. Articulation of Preferences with Weight/Reference Vectors

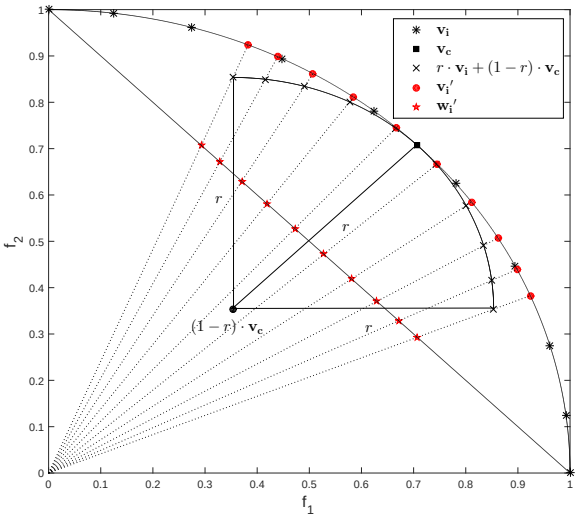


Fig. 1. A visualized illustration of the transformation procedure to generate weight/reference vectors inside a region specified by a central vector $\mathbf{v_c}$ and a radius r. In this example, 10 uniformly distributed weight/reference vectors are generated inside a region in a bi-objective space specified by $\mathbf{v_c} = (\frac{\sqrt{2}}{2}, \frac{\sqrt{2}}{2})$ and $r = 0.5$.

So far we have shown how to generate uniformly distributed weight/reference vectors for general optimization using MOEA/D, NSGA-III and RVEA. In practice, a decision-maker may only be interested in some specific regions instead of the whole objective space. In the following, we will show how to articulate decision-maker preferences with the weight/reference vectors.

As suggested in [35], decision-maker preferences can be articulated with reference vectors by defining a central vector $\mathbf{v_c}$ together with a radius r to specify a preferred region in the normalized objective space:

$$\mathbf{v_i}' = \frac{r \cdot \mathbf{v_i} + (1-r) \cdot \mathbf{v_c}}{\|r \cdot \mathbf{v_i} + (1-r) \cdot \mathbf{v_c}\|}, \tag{12}$$

where $\mathbf{v_i}$ with $i = 1, ..., N$ denotes the reference vectors generated using (11), and $\mathbf{v_i}'$ denotes the reference vectors inside the preference region specified by $\mathbf{v_c}$ and r. To define a central vector $\mathbf{v_c}$, the decision-maker can first specify a vector $\mathbf{u_c}$ according to his/her preference and then normalize it into a unit vector using $\mathbf{v_c} = \frac{\mathbf{u_c}}{\|\mathbf{u_c}\|}$, and if the decision-maker has multiple preferences, he/she can also specify multiple central vectors. As to the radius r, any value of $r \in (0, 1)$ is applicable, where a smaller r indicates a relatively smaller preference region around the central vector, and a larger r will result in a wider range of solutions around the central vector. It should be noted that, for better robustness, it is suggested to also include the extreme vectors (i.e. $(1, 0, ..., 0)$, $(0, 1, ..., 0)$, ..., $(0, 0, ..., 1)$) and the central vector $\mathbf{v_c}$ into the preference based reference vector set. Therefore, given a number of K central preference vectors, the size of a full preference based reference vector set is $K \cdot (N + 1) + M$, where N is the size of the original vector set generated using (11) and M is the number of objectives.

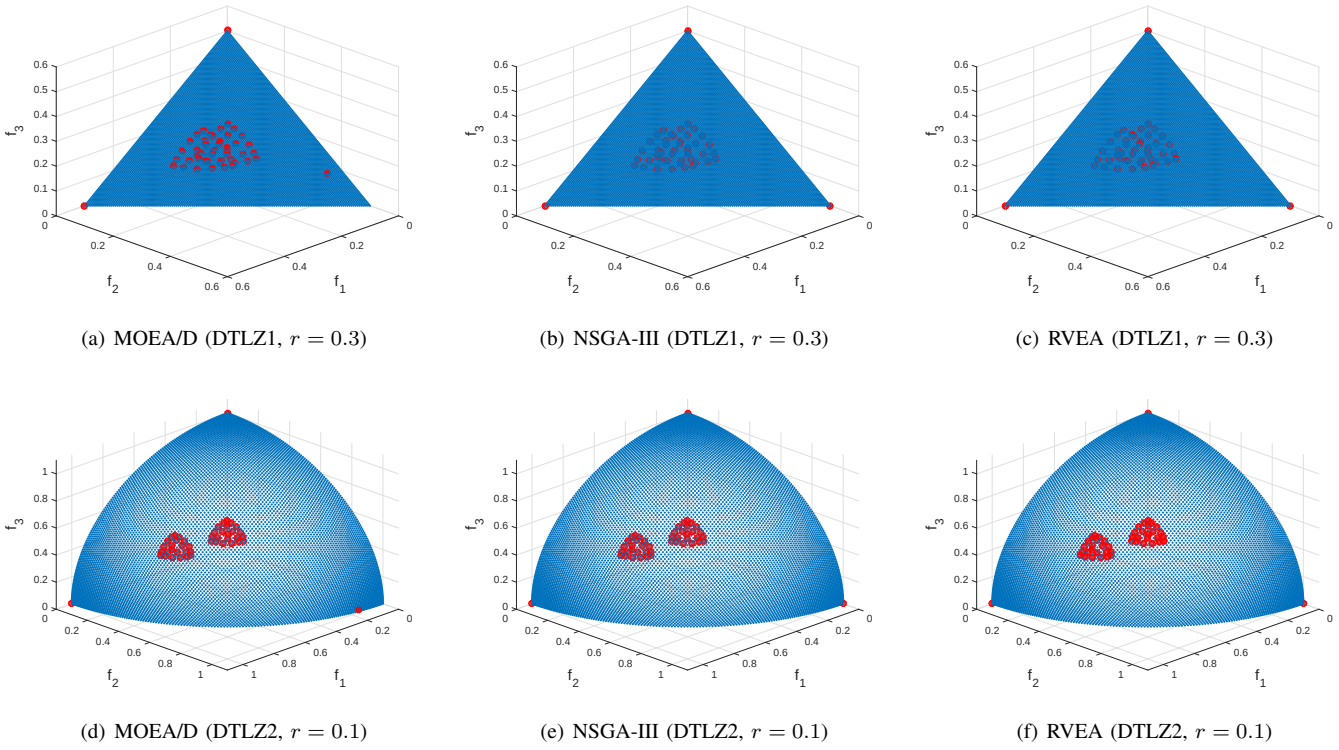
Similarly, to articulate decision-maker preferences with weight vectors, the only step is to map the reference vectors generated by (12) onto a hyperplane using the following transformation:

$$\mathbf{w_i}' = \frac{\mathbf{v_i}'}{\|\mathbf{v_i}'\|_1}, \tag{13}$$

where $\mathbf{w_i}'$ with $i = 1, ..., N$ denotes the weight vectors that articulate decision-maker preferences. The above geometrical transformations to generate preference based weight/reference vectors is illustrated in Fig. 1.

Note that the proposed approach described above is applicabe to all decomposition based MOEAs for generating weight vectors. In addition, the preference articulation approach suggested in [35], [48] can handle only one central vector $\mathbf{v_c}$ in each preference set, while the above approach is able to support multiple central vectors inside one reference set.

To better understand how the proposed preference articulation approach works, we will show in the following some empirical results obtained on selected benchmark test problems, i.e., DTLZ1 and DTLZ2 from the DTLZ test suite [60]. As shown by the results summarized in Fig. 2, a general observation is that all three algorithms work well with the proposed preference articulation approach. In addition, the following observations can also be made. First, as evidenced in Fig. 2(d) to Fig. 2(f), the proposed preference articulation approach is capable of dealing with multiple independent preference regions during one single run of the algorithm. Second, from Fig. 2(a) to Fig. 2(c), we can see that when the preference regions are overlapped, the proposed preference articulation approach also works well. Third, by tuning the preference radius r (e.g., $r = 0.3$ in Fig. 2(a) to Fig. 2(c) and $r = 0.1$ in Fig. 2(d) to Fig. 2(f)), the proposed preference



(a) MOEA/D (DTLZ1, $r = 0.3$) (b) NSGA-III (DTLZ1, $r = 0.3$) (c) RVEA (DTLZ1, $r = 0.3$)

(d) MOEA/D (DTLZ2, $r = 0.1$) (e) NSGA-III (DTLZ2, $r = 0.1$) (f) RVEA (DTLZ2, $r = 0.1$)

Fig. 2. An example to assess the proposed preference articulation approach used by MOEA/D, NSGA-III and RVEA on three-objective DTLZ1 and DTLZ2. In this example, two central vectors, namely, $(0.557, 0.557, 0.557)$ and $(0.774, 0.417, 0.477)$ are given simultaneously to specify multiple decision-maker preferences. Preference radius $r = 0.3$ and $r = 0.1$ are applied for DTLZ1 and DTLZ2, respectively. To generate the preference based weight/reference set, $H = 5$ is adopted for the simplex-lattice design in (10) to generate a number of 21 vectors around each central vector. The red dots, which are the preferred solutions approximated by each algorithm, are obtained with 100,000 fitness evaluations during one single run. The shape in blue in each figure denotes the true Pareto front.

articulation approach is able to adjust the size of the preference region covered by the weight/reference vectors.

It is worth noting that, although MOEA/D, NSGA-III and RVEA show very similar performance on the benchmark test problems using the proposed preference articulation approach, the results to be presented in Section V-C indicate that the performance of the three algorithms varies a lot on the seven-objective HEV controller model. This might be due to the fact that most existing benchmark test problems originally proposed for general optimization are not very well suited for assessing performance for preference based optimization.

IV. Seven-objective HEV Controller Model

In hybrid electric vehicles (HEVs), the propulsion is provided by a combination of internal combustion engine (ICE) and electric motor (EM). As a key element in the design of HEVs, the energy management controller (controller for short hereafter) plays an important role in guaranteeing peak performance of the hybrid power unit. As illustrated in Fig. 3, a general controller is used to determine in which condition the ICE and EM should be used [61]. In our case, since vehicle speed and torque are controlled by the driver or an externally given drive cycle, there is little freedom in the operation of the EM. Therefore, the controller basically controls the operation of the ICE only.

To be specific, HEV controller models are often designed on the basis of a small set of rules configured by a set of design parameters, i.e. the decision variables to the optimization objective(s). As listed in Appendix B, the proposed seven-objective HEV controller model is associated with a set of 11 design parameters that configure six basic control rules detailed in Appendix C.

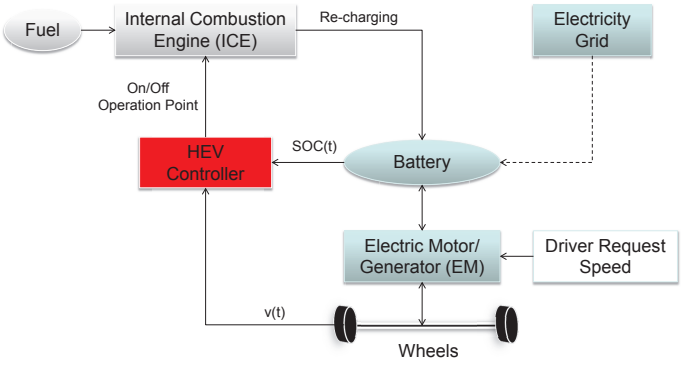


Fig. 3. The sketch of the employed HEV architecture used for the study in this work. A conventional internal combustion engine (ICE), powered by fuel, charges a battery that drives an electric motor (EM) to propel the car. Before the start of the trip, the battery can also be charged from the electricity grid. During braking, the EM can act as a generator to re-charge the battery using kinetic energy. The amount of torque generated by the EM is defined by the driver (or in this case the driving cycle). The controller uses current speed, denoted as $v(t)$, and current battery state-of-charge (SOC), denoted as $SOC(t)$, as inputs and determines the operation of the ICE.

TABLE I
SUMMARY OF THE SEVEN OPTIMIZATION OBJECTIVES IN THE HEV CONTROLLER MODEL.

Denotation	Objective Name
FC	Fuel consumption and CO_2
BS	Battery stress
OPC	ICE operation changes
Emission	ICE emissions
Noise	Perceived ICE noise
UO	Urban operation
SOC	Average battery state of charge level

One of the most important objectives of a controller is to minimize the fuel consumption, and thus most existing work on optimal controller design has exclusively focused on this aspect [62]–[65]. To achieve the optimal control with a focus on fuel consumption, some traditional single-objective optimization methods such as dynamic programming (DP) [66]–[68] have been applied using a discretization of the decision variables in the control models and investigating all possible actions. As pointed out in [69], however, apart from fuel consumption, there are also other factors that may influence the overall performance of HEVs. For example, the driving experience, including noise, vibration, and harshness (NVH), is an important criterion for most customers while the battery lifetime is a major concern for manufacturers. Considering that such additional factors may also have significant influence on the overall performance of HEVs, recently, an optimization model has been designed to simultaneously take seven objectives into consideration [69], where the objective names are summarized in Table I and detail definitions of the objectives can be found in Appendix D.

To deploy the proposed controller, the seven-objective HEV controller model is applied to a plug-in serial HEV simulator with a battery capacity of 24.4 kWh, which is approximately 150 km of pure electric driving range. Instead of detailed ICE, EM, and battery models, we employ efficiency maps based on test-bench measurements in the form of look-up-tables. From a pre-specified speed profile, also known as driving cycle, the corresponding acceleration value to follow this speed profile is computed, and then a basic model of the HEV is used to determine the resulting information such as torque demand. With the information generated by the simulator, the function values of the seven optimization objectives can be calculated, which can be further used in the optimization procedure as fitness values. An illustration to the architecture of the seven-objective HEV controller model and its relationship to the optimizer as well as the HEV controller can be found in Fig. 4.

V. Optimization Results

This section presents the optimization results obtained by MOEA/D, NSGA-III, and RVEA on the seven-objective HEV controller model. We first perform general optimization using the three algorithms as usually done in the literature and analyze the solutions. On the basis of these analyses, we incorporate user preferences into the three algorithms to get solutions of particular interest to the decision-maker.

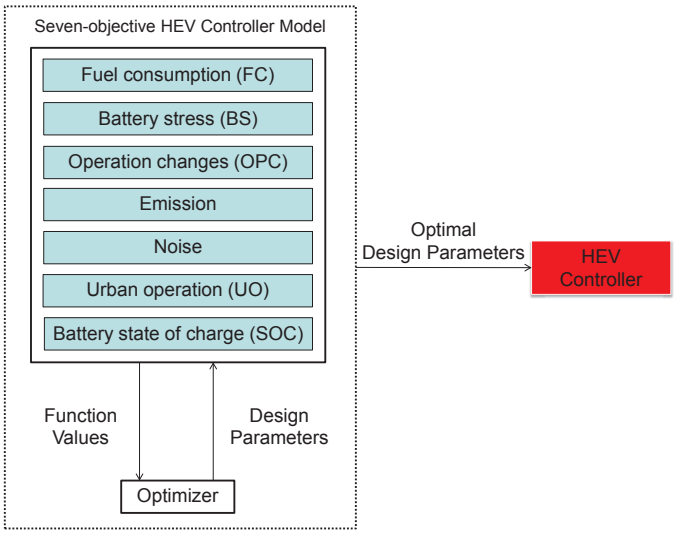


Fig. 4. An illustration of the architecture of the seven-objective HEV controller model and its relationship to the optimizer as well as the HEV controller. The seven-objective HEV controller model is connected to an optimizer, e.g., a many-objective evolutionary algorithm. During the optimization process, the optimizer iteratively outputs candidate solutions of optimal design parameters to the seven-objective HEV controller model. As a feedback, the HEV controller model outputs the objective function values to the optimizer as fitness of the candidate solutions. Once the optimization is finished, the final optimal design parameters will serve as an input to the HEV controller to deploy the control rules.

A. General Optimization

Most MOEAs are designed for general optimization, i.e., for approximation of the whole Pareto fronts of MOPs as well as MaOPs with a limited number of candidate solutions. In this subsection, we apply MOEA/D, NSGA-III and RVEA to the optimization of the seven-objective HEV controller model without decision-maker preferences.

To sample uniformly distributed reference vectors for such general optimization, the method introduced in Section III-A is adopted with settings of $H = 4$, thus generating a number of 210 weight/reference vectors in total for each algorithm. Other parameter settings of the three algorithms are all the same as recommended in [25], [34] and [35], respectively.

To evaluate the quality of the solution sets obtained by each of the three algorithms, the hypervolume metric is applied as the performance indicator [39]. Let P be the approximate PF obtained by an MOEA, $y^* = (y_1^*, ..., y_M^*)$ be a reference point in the objective space. The HV value of P (with respect to y^*) is the volume of the region which dominates point y^* and is dominated by all points in P. Here, we choose $[4, 4, 4, 4, 4, 4, 4]$ as the reference point y^*, where 4 is the worst possible value for all objectives that will be returned by the fitness functions (HEV simulator).

A maximum number of 10000 fitness evaluations is used as the termination criterion for each algorithm in each run, and the statistical results of the hypervolume values obtained by each algorithm over 20 independent runs are summarized in Fig. 5. From the figure, we can see RVEA has achieved the largest mean value as well as the largest maximum value in HV among the three algorithms. On this optimization problem,

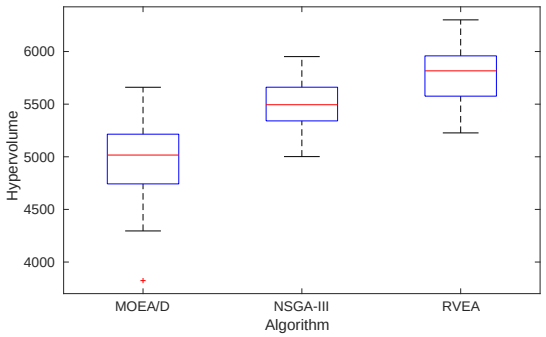


Fig. 5. Statistical results for hypervolume values of the solutions obtained by the MOEA/D, NSGA-III and RVEA over 20 runs each.

MOEA/D performs slightly worse than NSGA-III.

While most benchmark tests of multi-/many-objective optimization merely make statistical comparisons in terms of performance indicators of the solution sets, in real-world applications such as the optimization of a HEV controller, it is also of great interest to gain more insight into the problem by analyzing the trade-off relationships between the objectives. Therefore, in the following subsections, we will perform some analyses of the solutions obtained by all three algorithms, based on which, the decision-maker preferences will be incorporated into the optimizers.

B. Analyses of Solutions

In order to perform analyses of the solutions obtained by the three MOEAs, we use a configuration manually tuned by an experienced engineer as the baseline solution $(1, 1, 1, 1, 1, 1, 1)$ to normalize the objective values of the obtained solutions.

We first combine all solutions obtained by the three algorithms in 20 runs and remove the dominated solutions. As shown by the parallel coordinate plots in Fig. 6(a), the combined non-dominated solution set is well distributed around the baseline solution, showing highly diversified trade-offs between different objectives. In general, the majority of objective values are below the baseline for a large part of solutions, with a total 1086 solutions dominating the baseline solution, as shown in Fig. 6(b).

As shown in Fig. 7, MOEA/D has obtained 611 of the 1086 solutions that dominate the baseline solution, while NSGA-III and RVEA have obtained 246 and 229 solutions, respectively. This is an interesting observation as the ranking of the number of obtained non-dominated solutions is exactly opposite to the ranking of HV values as shown in Fig. 5. This is due to the fact that although MOEA/D achieves a large number of well-converged solutions, most of them are located in relatively small regions, leading to worse HV values. Empirical results to be presented in Section V-C confirm this observation.

Due to the conflicts between objectives such as FC, OPC, Emission and SOC, some solutions are significantly better than the baseline on these objectives, while others are significantly worse. For further investigations, the parallel coordinates of the extreme solutions that have the worst and best values on objectives of FC, OPC, Emission and SOC are plotted in Fig. 6(c) and Fig. 6(d), respectively. From these results, the following observations can be made.

First, as shown in both Fig. 6(c) and Fig. 6(d), OPC and Emission are highly consistent, as the solutions with the worst (as well as the best) OPC and Emission strongly overlap. These results indicate that the optima of OPC and Emission can very likely be achieved simultaneously. This is due to the fact that both objectives are very sensitive to the on/off changes of the ICE, which is consistent with the experience in HEV design.

Second, Fig. 6(c) reveals that the objective of Noise is in conflict with the objective of SOC, where the solution with the best SOC has the worst Noise value. Such a relationship between Noise and SOC is not obvious as there is no direct physical connection between the two objectives. This relationship is definitely very useful in HEV design as a guideline to make trade-offs between the two objectives when tuning the design parameters.

Finally, as shown in Fig. 6(d), the solution with the best FC achieves around 10% improvement compared to the baseline solution. Such an improvement is significant because in practice, it is very likely that the improvement of FC is at the cost of other objectives such as Noise or SOC, while in the solutions obtained, most of the other objectives are also significantly improved in addition to the 10% improvement on the objective of FC.

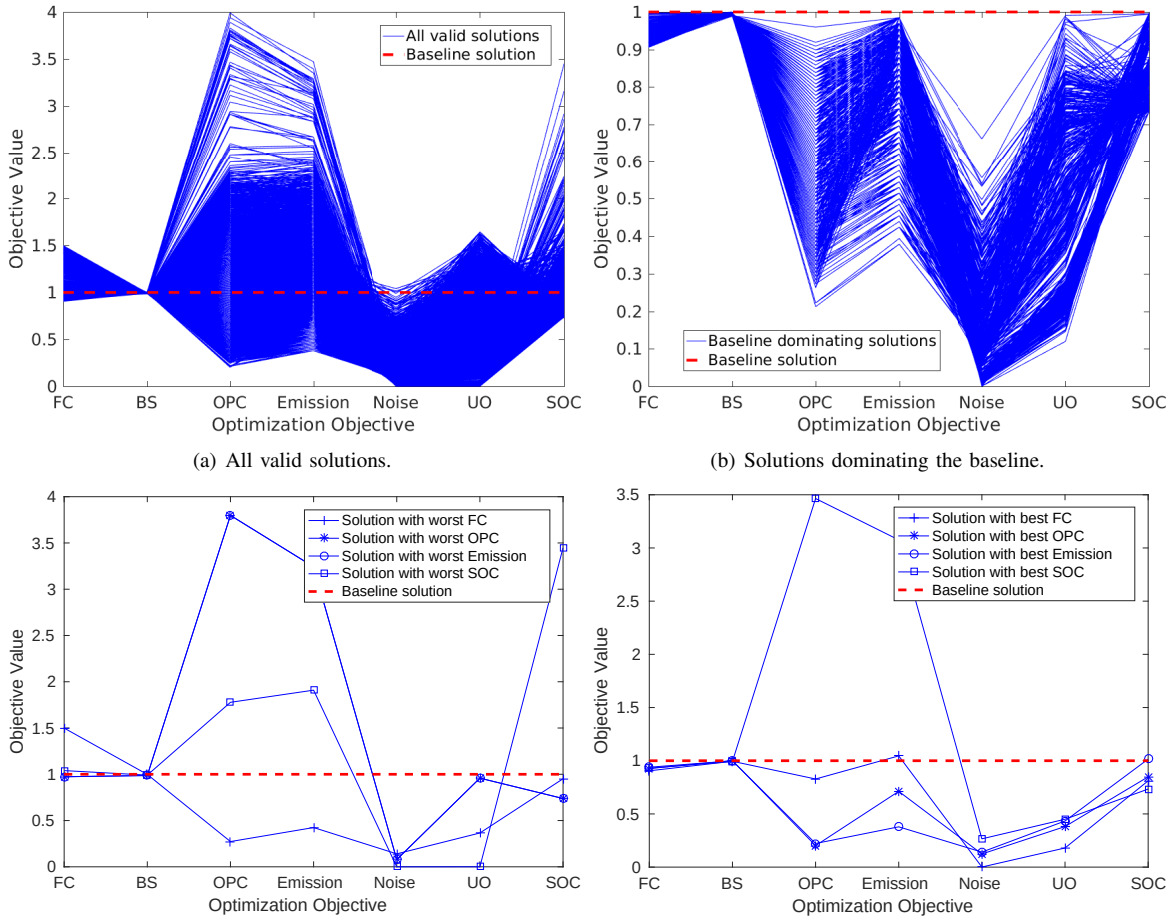
TABLE II
DESIGN PARAMETERS FOR THREE DIFFERENT CONTROLLER CONFIGURATIONS.

Parameter	Baseline	Min-Fuel	Balanced
SOC_{max} (%)	70	34.494	35.314
SOC_{min} (%)	40	28.699	26.082
v_1 (km/h)	20	27.151	20.634
v_2 (km/h)	30	50.262	50.204
rev_1 (/m)	2500	2941	2907
$torque_1$ (N · m)	9.42	8.6269	8.2231
rev_2 (/m)	3500	3981	3070
$torque_2$ (N · m)	15.03	16.991	12.392
rev_3 (/m)	5500	4008	4946
$torque_3$ (N · m)	25.92	17.622	22.228
v_{off} (km/h)	20	39.057	20.298

TABLE III
OBJECTIVE VALUES FOR THREE DIFFERENT CONTROLLER CONFIGURATIONS. BEST RESULTS ARE HIGHLIGHTED.

Objective	Baseline	Min-Fuel	Balanced
Fuel consumption	1.0	**0.905**	0.94519
Battery stress	1.0	**0.99471**	0.99914
OPC	1.0	0.82828	**0.21212**
Emission	1.0	1.0455	**0.37879**
Noise	1.0	**0.001**	0.093003
UO	1.0	**0.17986**	0.41935
SOC	1.0	**0.8111**	0.96108

In addition to the observations made above, we have also taken a more detailed look at two interesting solutions, the solution with the best FC and the one with the best overall improvement (a plain sum of objective values), in comparison with the baseline solution. As summarized in Table II and Table III, the baseline solution is also given as a reference normalized to 1, and the solution with the best FC and the



(a) All valid solutions.

(b) Solutions dominating the baseline.

(c) Solutions with the worst values on objectives of FC, OPC, Emission and SOC.

(d) Solutions with the best values on objectives of FC, OPC, Emission and SOC.

Fig. 6. Parallel coordinate plots of solutions obtained by MOEA/D, NSGA-III and RVEA in the final populations of 20 runs, in comparison with the baseline solution which is manually tuned by an experienced engineer.

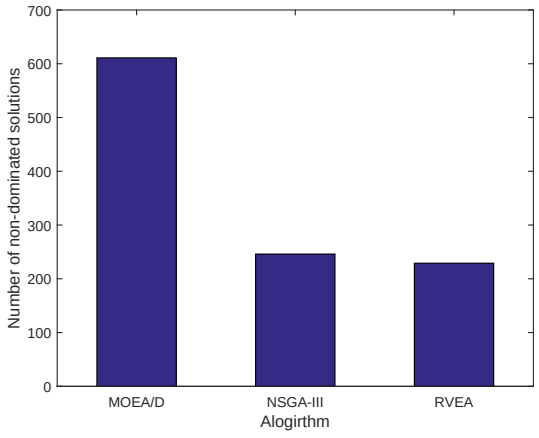


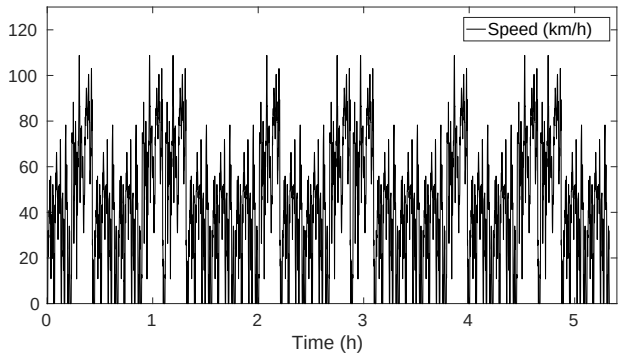
Fig. 7. Number of non-dominated solutions obtained by MOEA/D, NSGA-III and RVEA, respectively among the combined non-dominated solution set achieved by the three algorithm in 20 runs.

solution with the best overall improvement are listed in column "Min-Fuel" and column "Balanced", respectively. While Table II presents the design parameters deployed by the controller of the HEV simulator, Table III presents the objective values output by the seven-objective HEV controller model.

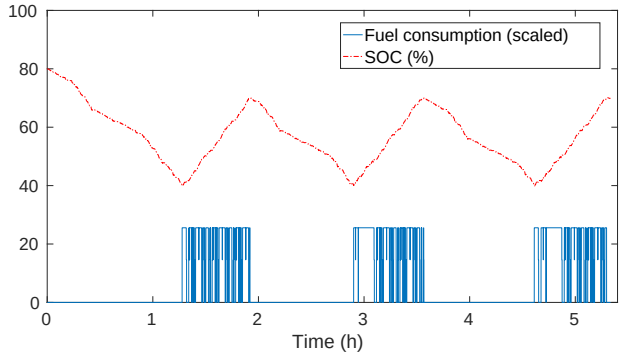
For the solution with the best FC, it can be observed from column "Min-Fuel" in Table III that FC has been improved by approximately 10% in comparison with the baseline solution, at the cost of a slight increase of Emission. For the solution with the best overall improvement, as presented in column "Balanced" in Table III, although FC is slightly higher than that of the solution with the best FC, the other objectives have substantially improved, by 43% averaged over all objectives, in comparison with the baseline solution, especially on Emission and Noise.

To further assess the quality of the representative solutions listed in Table II and Table III, we run the HEV simulator that deploys the control strategy using the design parameters listed in Table II and record the simulated data of FC and SOC generated during our project-specific drive cycle. As shown in Fig. 8, the data generated by the HEV simulator also shows observations that are consistent with our analyses above.

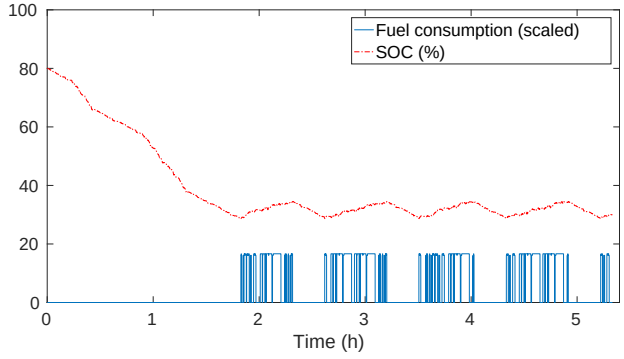
Based on the above analyses, the decision-maker (HEV engineer) is particularly interested in further exploiting the potential improvement on one of the objectives such as FC and striking a better balance between other objectives. In the following, the above user preferences will be incorporated in



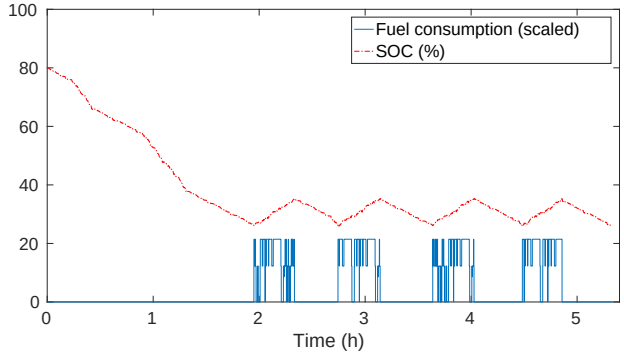
(a) Speed profile (drive cycle)



(b) Baseline configuration



(c) Min-Fuel configuration



(d) Balanced configuration

Fig. 8. The data of (instantaneous) fuel consumption (FC) and battery state-of-charge (SOC) with respect to the speed profile defined by the used driving cycle. The first figure shows the speed profile of the HEV, and the other three figures show the data generated by the HEV simulator which deploys controllers configured by different sets of design parameters (*Baseline, Min-Fuel, Balanced*), as defined in Table II.

the MOEAs to see if better solutions can be achieved.

C. Preference Articulation

As we previously discussed, general optimization is realistic for MOPs that have two or three objectives. For MaOPs having more than three objectives, however, approximation of the whole Pareto front in a very high dimensional space with a limited number of solutions becomes hardly possible. This challenge, however, has not been fully recognized. In the following, we illustrate using the HEV controller design example why user preferences are important for solving practical many-objective optimization problems.

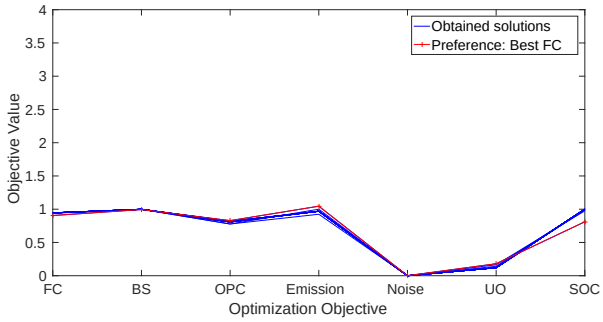
In the following experiments we show how to perform preference-driven optimization using MOEA/D, NSGA-III and RVEA with the proposed preference articulation approach. First, by using the solution with the best FC as a single preference central vector, we see if it is possible to further minimize FC. In addition, using solutions with the best SOC and best Noise as two simultaneous preference central vectors, we investigate the trade-off relationship between these two objectives.

1) Preference Articulation for Minimizing FC: To articulate preference for minimizing the objective of fuel consumption (FC), we first specify the solution with the lowest FC value obtained in the general optimization as a preference central vector $\mathbf{v_c}$ (refer to (12)), and then set the preference radius as $r = 0.1$. The solutions obtained by MOEA/D, NSGA-III and RVEA are shown in Fig. 9, from which the following remarks can be made.

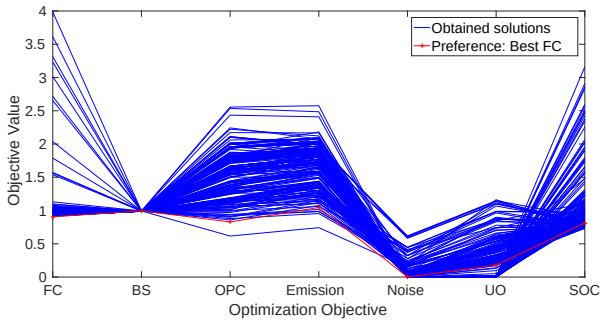
First, NSGA-III is able to obtain a large number of very different solutions, most of which are away from the preference central vector. By contrast, RVEA and MOEA/D have obtained a smaller number of solutions that are close to or even better than the preferred solution on most objectives. These results imply that NSGA-III tends to generate more widely spread candidate solutions, but fails to stick to the preferences.

Second, as observed from Fig. 9(c), with a slightly higher FC, the HEV simulator is able to achieve a significantly better performance in OPC and Emission. This is due to the fact that turning on/off the ICE too frequently (to reduce FC) will cost not only additional operation changes (OPC), but also extra start-up operations of the ICE, both of which will produce more emissions. By contrast, if the controller adopts optimal design parameters which aim to reduce the number of OPC operations at a slight higher cost of FC, as shown in Fig. 9(c), the objective of Emission can be further improved.

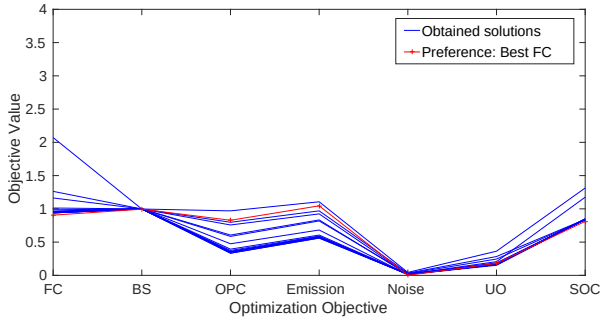
Third, we can see from Fig. 9(a) that solutions obtained by MOEA/D distribute very closely to the preference central vector and thus lack a certain degree of diversity. This small degree of diversity in solutions might be caused by the neighborhood based steady-state reproduction mechanism adopted in MOEA/D. By contrast, NSGA-III and RVEA, which adopt a generational reproduction mechanism performed on the whole population, show significantly better population diversity. Among the three algorithms, RVEA seems to have achieved the best balance between convergence and diversity.



(a) Solutions obtained by MOEA/D.

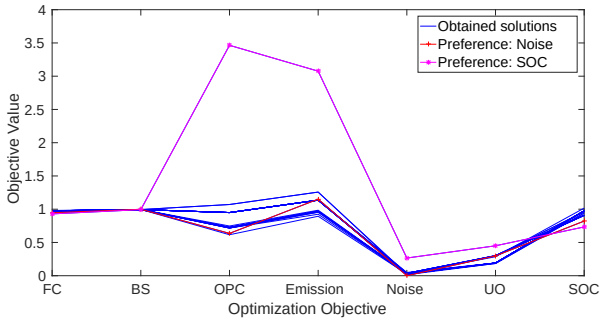


(b) Solutions obtained by NSGA-III.

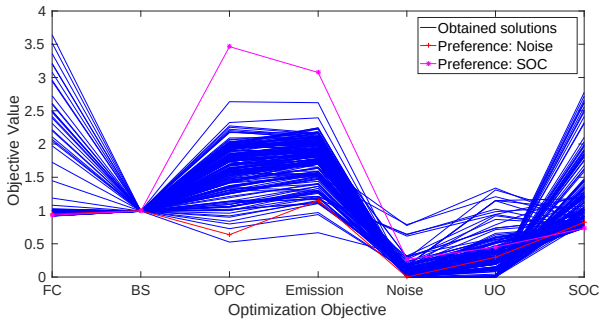


(c) Solutions obtained by RVEA.

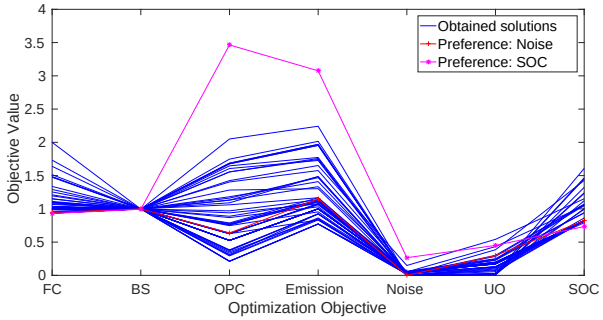
Fig. 9. Parallel coordinate plots of solutions obtained by MOEA/D, NSGA-III, and RVEA using the solution with the best FC in general optimization as the preference central vector.



(a) Solutions obtained by MOEA/D



(b) Solutions obtained by NSGA-III



(c) Solutions obtained by RVEA

Fig. 10. Parallel coordinate plots of solutions obtained by MOEA/D, NSGA-III, and RVEA using the solutions with the best Noise and the best SOC in general optimization together as the preference central vectors.

2) Preference Articulation for Trade-off between Noise and SOC: To articulate preference for trade-offs between Noise and SOC, the solutions with the best SOC and the best Noise obtained from the general optimization are used together as two preference central vectors $\mathbf{v_c}$ (refer to (12) with the preference radius being set to $r = 0.1$. From the results obtained by MOEA/D, NSGA-III and RVEA in Fig. 10, we make the following observations.

First, the solutions obtained by all three algorithms (especially MOEA/D) have a bias towards the solution with the best (lowest) Noise. Such an observation indicates that, in comparison with SOC, Noise is an objective that is relatively easier to optimize. This is due to the fact that, according to the definition of objective Noise in Appendix D, keeping the ICE noise lower than the rolling noise will directly result in the minimum noise value, which can be achieved in HEV design by keeping the ICE in lower-power operation points. By contrast, keeping the SOC objective low will require to define a very narrow band of permissible SOC values, severely impairing other objectives and making it more difficult to achieve low SOC values in the optimization.

Second, reducing Noise will consequently increase SOC, and vice versa, which indicates that the two objectives are in natural conflict and simultaneous optimization of both objectives are impossible. This knowledge is very helpful in HEV controller design as engineers can select a best trade-off solution from those obtained by the three algorithms, as shown in Fig. 10, instead of taking the time to fine tune design variables to simultaneously minimize both objectives. Here, we randomly select one trade-off solution obtained by RVEA in Fig. 10(c) to compare the ICE noise and the rolling noise recorded by the simulator. As shown in Fig. 11, by deploying the controller configured using the trade-off solution we select, the HEV controller manages to completely keep ICE noise below rolling noise, which satisfactorily meets the requirements of a low noise level in HEV design.

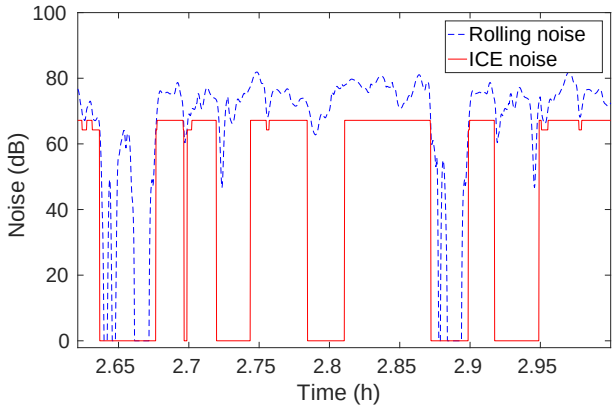


Fig. 11. A comparison between ICE noise and rolling noise generated from wind/tires recorded by the simulator that deploys the controller configured by one of the solutions that trade off between Noise and SOC as shown in Fig. 10(c). Here only a representative part of the total drive cycle is shown.

D. Discussions

As evidenced by the experimental results above, each of the three algorithms shows unique advantages over the other two when applied to the optimization of the seven-objective HEV controller model. On the one hand, MOEA/D has exhibited the best convergence performance, although the solutions may be densely distributed in a narrow region, leading to a smaller degree of diversity. This implies that in practical applications, if the decision-maker is more interested in convergence, MOEA/D is a good choice.

By contrast, NSGA-III shows the highest degree of diversity. No matter how the preference central vectors are set, the solutions obtained by NSGA-III are usually widely spread over the objective space, although most of the solutions are of relatively worse convergence in comparison with those obtained by the other two algorithms. Therefore, if an optimization problem has a simple fitness landscape and convergence is relatively easy to achieve, NSGA-III might be a good optimizer as it provides a wide distribution of the candidate solutions.

Finally, RVEA appears to have achieved the best balance between convergence and diversity. In general optimization, RVEA has achieved the highest HV value among the three algorithms. Meanwhile, in the preference based optimization, RVEA seems the most effective in achieving solutions preferred by the decision-maker, no matter whether a single or multiple preferences are given. Therefore, if the decision-maker has little priori knowledge about an optimization problem, RVEA might be the best option.

VI. Conclusion

In this paper, we have applied three state-of-the-art MOEAs, namely, MOEA/D, NSGA-III and RVEA to the optimization of a seven-objective HEV controller model. We also have proposed a preference articulate method by means of weight/reference vectors to guide the three MOEAs to search for the solutions preferred by the decison-maker. The proposed preference articulation approach, which only requires a set of preference central vector(s) and a preference radius, can be easily used to specify preferred regions in the objective space by generating uniformly distributed weight/reference vectors inside them.

A number of experiments have been performed on the seven-objective HEV controller model. At first, we aimed to approximate the Pareto front using using uniformly distributed weights/vectors across the whole objective space. The obtained solutions were carefully analyzed to gain useful information between the objectives, based on which preferences were articulated in the MOEAs using the proposed preference articulation approach to further exploit the solution space according to decision-maker preferences. Our experimental results have shown that, by applying the three MOEAs together with the preference articulation approach, the decision-maker is not only able to obtain a deeper understanding of the relationships (i.e., conflicts and consistencies) between different optimization objectives, but also obtain even better solutions in the preferred regions.

From the results of the compared solutions on the real-world many-objective optimization problem, we can conclude that the state-of-the-art MOEAs studied in this work have shown very good performance yet slightly different search properties. These finding are instructive for practitioners who are interested in solving real-world many-objective problems using evolutionary algorithms. We hope this work will trigger more interests in the academia in addressing real-world hard problems and attract more attention from industry on the recent developments in evolutionary computation research.

Acknowledgments

This work was supported in part by the Honda Research Institute Europe, and in part by the Joint Research Fund for Overseas Chinese, Hong Kong and Macao Scholars of the National Natural Science Foundation of China (Grant No. 61428302). The authors thank Mohammed Awada for support with the modeling of objectives and Kaname Narukawa for developing the initial many-objective optimization set-up.

References

[1] R. T. Marler and J. S. Arora, "Survey of multi-objective optimization methods for engineering," *Structural and multidisciplinary optimization*, vol. 26, no. 6, pp. 369–395, 2004.

[2] A. Ponsich, A. L. Jaimes, and C. A. C. Coello, "A survey on multiobjective evolutionary algorithms for the solution of the portfolio optimization problem and other finance and economics applications," *IEEE Transactions on Evolutionary Computation*, vol. 17, no. 3, pp. 321–344, 2013.

[3] P. M. Reed, D. Hadka, J. D. Herman, J. R. Kasprzyk, and J. B. Kollat, "Evolutionary multiobjective optimization in water resources: The past, present, and future," *Advances in water resources*, vol. 51, pp. 438–456, 2013.

[4] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan, "A fast and elitist multiobjective genetic algorithm: NSGA-II," *IEEE Transactions on Evolutionary Computation*, vol. 6, no. 2, pp. 182–197, 2002.

[5] E. Zitzler, M. Laumanns, and L. Thiele, "SPEA2: improving the strength Pareto evolutionary algorithm for multiobjective optimization," in *Fifth Conference on Evolutionary Methods for Design, Optimization and Control with Applications to Industrial Problems*, 2001, pp. 95–100.

[6] D. W. Corne, N. R. Jerram, J. D. Knowles, and M. J. Oates, "PESA-II: region-based selection in evolutionary multi-objective optimization," in *2001 Genetic and Evolutionary Computation Conference*, 2001, pp. 283–290.

[7] A. Zhou, B.-Y. Qu, H. Li, S.-Z. Zhao, P. N. Suganthan, and Q. Zhang, "Multiobjective evolutionary algorithms: A survey of the state of the art," *Swarm and Evolutionary Computation*, vol. 1, no. 1, pp. 32–49, 2011.

[8] K. Ikeda, H. Kita, and S. Kobayashi, "Failure of Pareto-based MOEAs: does non-dominated really mean near to optimal?" in *Proceedings of the IEEE Congress on Evolutionary Computation*, 2001, pp. 957–962.

[9] D. Brockhoff, T. Friedrich, N. Hebbinghaus, C. Klein, F. Neumann, and E. Zitzler, "On the effects of adding objectives to plateau functions," *IEEE Transactions on Evolutionary Computation*, vol. 13, no. 3, pp. 591–603, 2009.

[10] O. Schütze, A. Lara, and C. A. C. Coello, "On the influence of the number of objectives on the hardness of a multiobjective optimization problem," *IEEE Transactions on Evolutionary Computation*, vol. 15, no. 4, pp. 444–455, 2011.

[11] L. S. Batista, F. Campelo, F. G. Guimaraes, and J. A. Ramírez, "A comparison of dominance criteria in many-objective optimization problems," in *Proceedings of the IEEE Congress on Evolutionary Computation*, 2011, pp. 2359–2366.

[12] P. J. Fleming, R. C. Purshouse, and R. J. Lygoe, "Many-objective optimization: An engineering design perspective," in *Proceedings of the International Conference on Evolutionary multi-criterion optimization*. Springer, 2005, pp. 14–32.

[13] H. Ishibuchi, N. Tsukamoto, and Y. Nojima, "Evolutionary many-objective optimization: A short review," in *IEEE Congress on Evolutionary Computation*, 2008, pp. 2419–2426.

[14] Z. He and G. G. Yen, "Ranking many-objective evolutionary algorithms using performance metrics ensemble," in *Proceedings of the IEEE Congress on Evolutionary Computation*, 2013, pp. 2480–2487.

[15] B. Li, J. Li, K. Tang, and X. Yao, "Many-objective evolutionary algorithms: A survey," *ACM Computing Surveys*, vol. 48, no. 1, p. 13, 2015.

[16] X. Zou, Y. Chen, M. Liu, and L. Kang, "A new evolutionary algorithm for solving many-objective optimization problems," *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics*, vol. 38, no. 5, pp. 1402–1412, 2008.

[17] M. Laumanns, L. Thiele, K. Deb, and E. Zitzler, "Combining convergence and diversity in evolutionary multiobjective optimization," *Evolutionary Computation*, vol. 10, no. 3, pp. 263–282, 2002.

[18] D. Hadka and P. Reed, "Borg: An auto-adaptive many-objective evolutionary computing framework," *Evolutionary Computation*, vol. 21, no. 2, pp. 231–259, 2013.

[19] G. Wang and H. Jiang, "Fuzzy-dominance and its application in evolutionary many objective optimization," in *2007 International Conference on Computational Intelligence and Security Workshops*, 2007, pp. 195–198.

[20] S. Yang, M. Li, X. Liu, and J. Zheng, "A grid-based evolutionary algorithm for many-objective optimization," *IEEE Transactions on Evolutionary Computation*, vol. 17, no. 5, pp. 721–736, 2013.

[21] M. Li, S. Yang, and X. Liu, "Shift-based density estimation for pareto-based algorithms in many-objective optimization," *IEEE Transactions on Evolutionary Computation*, vol. 18, no. 3, pp. 348–365, 2014.

[22] X. Zhang, Y. Tian, and Y. Jin, "A knee point driven evolutionary algorithm for many-objective optimization," *IEEE Transactions on Evolutionary Computation*, vol. 19, no. 6, pp. 761–776, 2015.

[23] H. Wang, L. Jiao, and X. Yao, "Two_arch2: An improved two-archive algorithm for many-objective optimization," *Evolutionary Computation, IEEE Transactions on*, vol. 19, no. 4, pp. 524–541, Aug 2015.

[24] Z. He and G. G. Yen, "Many-objective evolutionary algorithm: objective space reduction and diversity improvement," *IEEE Transactions on Evolutionary Computation*, vol. 20, no. 1, pp. 145–160, 2016.

[25] Q. Zhang and H. Li, "MOEA/D: a multi-objective evolutionary algorithm based on decomposition," *IEEE Transactions on Evolutionary Computation*, vol. 11, no. 6, pp. 712–731, 2007.

[26] H.-L. Liu, F. Gu, and Q. Zhang, "Decomposition of a multiobjective optimization problem into a number of simple multiobjective subproblems," *IEEE Transactions on Evolutionary Computation*, vol. 18, no. 3, pp. 2450–455, 2014.

[27] Y. Yuan, H. Xu, B. Wang, B. Zhang, and X. Yao, "Balancing convergence and diversity in decomposition-based many-objective optimizers," *IEEE Transactions on Evolutionary Computation*, accepted in 2015.

[28] Y. Jin, M. Olhofer, and B. Sendhoff, "Dynamic weighted aggregation for evolutionary multi-objective optimization: Why does it work and how?" in *Proceedings of the Genetic and Evolutionary Computation Conference*. ACM, 2001, pp. 1042–1049.

[29] T. Murata, H. Ishibuchi, and M. Gen, "Specification of genetic search directions in cellular multiobjective genetic algorithms," in *Proceedings of the International Conference on Evolutionary Multi-Criterion Optimization*, 2001, pp. 82–95.

[30] K. Li, Q. Zhang, S. Kwong, M. Li, and R. Wang, "Stable matching-based selection in evolutionary multiobjective optimization," *IEEE Transactions on Evolutionary Computation*, vol. 18, no. 6, pp. 909–923, 2014.

[31] K. Li, S. Kwong, Q. Zhang, and K. Deb, "Interrelationship-based selection for decomposition multiobjective optimization," *IEEE Transactions on Cybernetics*, vol. 45, no. 10, pp. 2076–2088, 2014.

[32] L. Chen, H.-L. Liu, C. Lu, Y.-m. Cheung, and J. Zhang, "A novel evolutionary multi-objective algorithm based on S Metric selection and M2M population decomposition," in *Proceedings of the Asia Pacific Symposium on Intelligent and Evolutionary Systems*. Springer, 2015, pp. 441–452.

[33] K. Li, Q. Zhang, and S. Kwong, "An evolutionary many-objective optimization algorithm based on dominance and decomposition," *IEEE Transactions on Evolutionary Computation*, vol. 19, no. 5, pp. 694–716, 2015.

[34] K. Deb and H. Jain, "An evolutionary many-objective optimization algorithm using reference-point-based nondominated sorting approach, part I: solving problems with box constraints," *IEEE Transactions on Evolutionary Computation*, vol. 18, no. 4, pp. 577–601, 2014.

[35] R. Cheng, Y. Jin, M. Olhofer, and B. Sendhoff, "A reference vector guided evolutionary algorithm for many-objective optimization," *IEEE Transactions on Evolutionary Computation*, 2016(in press).

[36] N. Beume, B. Naujoks, and M. Emmerich, "SMS-EMOA: multiobjective selection based on dominated hypervolume," *European Journal of Operational Research*, vol. 181, no. 3, pp. 1653–1669, 2007.

[37] E. Zitzler and S. Künzli, "Indicator-based selection in multiobjective search," in *8th International Conference on Parallel Problem Solving from Nature*, 2004, pp. 832–842.

[38] J. Bader and E. Zitzler, "HypE: an algorithm for fast hypervolume-based many-objective optimization," *Evolutionary Computation*, vol. 19, no. 1, pp. 45–76, 2011.

[39] L. While, P. Hingston, L. Barone, and S. Huband, "A faster algorithm for calculating hypervolume," *IEEE Transactions on Evolutionary Computation*, vol. 10, no. 1, pp. 29–38, 2006.

[40] L. While, L. Bradstreet, and L. Barone, "A fast way of calculating exact hypervolumes," *IEEE Transactions on Evolutionary Computation*, vol. 16, p. 86, 2012.

[41] C. Priester, K. Narukawa, and T. Rodemann, "A comparison of different algorithms for the calculation of dominated hypervolumes," in *Proc. of 2013 Genetic and Evolutionary Computation Conference*. ACM, 2013, pp. 655–662.

[42] C. M. Fonseca and P. J. Fleming, "Multiobjective genetic algorithms made easy: selection sharing and mating restriction," in *Proceedings of the Genetic Algorithms in Engineering Systems: Innovations and Applications*. IET, 1995, pp. 45–52.

[43] Y. Jin and B. Sendhoff, "Incorporation of fuzzy preferences into evolutionary multiobjective optimization." in *Proc. of the 4th Asia-Pacific Conference on Simulated Evolution and Learning*, 2002, pp. 26–30.

[44] D. Cvetkovic and I. C. Parmee, "Preferences and their application in evolutionary multiobjective optimization," *IEEE Transactions on evolutionary computation*, vol. 6, no. 1, pp. 42–57, 2002.

[45] K. Deb, J. Sundar, N. Udaya Bhaskara Rao, and S. Chaudhuri, "Reference point based multi-objective optimization using evolutionary algorithms," *International Journal of Computational Intelligence Research*, vol. 2, no. 3, pp. 273–286, 2006.

[46] H. Ishibuchi, Y. Nojima, K. Narukawa, and T. Doi, "Incorporation of decision maker's preference into evolutionary multiobjective optimization algorithms," in *Proc. of 2006 Genetic and Evolutionary Computation Conference*. ACM, 2006, pp. 741–742.

[47] F. Siegmund, J. Bernedixen, L. Pehrsson, A. H. Ng, and K. Deb, "Reference point-based evolutionary multi-objective optimization for industrial systems simulation," in *Proceedings of the winter simulation conference*. Winter Simulation Conference, 2012, p. 130.

[48] R. Cheng, M. Olhofer, and Y. Jin, "Reference vector based a posteriori preference articulation for evolutionary multiobjective optimization," in *Proceedings of the IEEE Congress on Evolutionary Computation*. IEEE, 2015.

[49] I. Giagkiozis and P. J. Fleming, "Pareto front estimation for decision making," *Evolutionary Computation*, vol. 22, no. 4, pp. 651–678, 2014.

[50] R. Cheng, Y. Jin, K. Narukawa, and B. Sendhoff, "A multiobjective evoltuionary algorithm using Gaussian process based inverse modeling," *IEEE Transactions on Evolutionary Computation*, vol. 19, no. 6, pp. 838–856, 2015.

[51] Y. Yan, I. Giagkiozis, and P. J. Fleming, "Improved sampling of decision space for pareto estimation," in *Proceedings of the Annual Conference on Genetic and Evolutionary Computation*. ACM, 2015, pp. 767–774.
[52] A. Mohammadi, M. N. Omidvar, X. Li, and K. Deb, "Integrating user preferences and decomposition methods for many-objective optimization," in *2014 IEEE Congress on Evolutionary Computation (CEC)*. IEEE, 2014, pp. 421–428.
[53] G. Yu, J. Zheng, R. Shen, and M. Li, "Decomposing the user-preference in multiobjective optimization," *Soft Computing*, pp. 1–17, 2015.
[54] K. Narukawa, Y. Setoguchi, Y. Tanigaki, M. Olhofer, B. Sendhoff, and H. Ishibuchi, "Preference representation using gaussian functions on a hyperplane in evolutionary multi-objective optimization," *Soft Computing*, vol. 20, no. 7, pp. 2733–2757, 2016.
[55] G. Syswerda, "A study of reproduction in generational and steady state genetic algorithms," *Foundations of genetic algorithms*, vol. 2, pp. 94–101, 1991.
[56] K. Deb and R. B. Agrawal, "Simulated binary crossover for continuous search space," *Complex Systems*, vol. 9, no. 4, pp. 115–148, 1995.
[57] K. Deb and M. Goyal, "A combined genetic adaptive search (GeneAS) for engineering design," *Computer Science and Informatics*, vol. 26, no. 4, pp. 30–45, 1996.
[58] J. A. Cornell, *Experiments with mixtures: designs, models, and the analysis of mixture data*. John Wiley & Sons, 2011.
[59] R. Cheng, Y. Jin, and K. Narukawa, "Adaptive reference vector generation for inverse model based evolutionary multiobjective optimization with degenerate and disconnected pareto fronts," in *Proceedings of the Evolutionary Multi-Criterion Optimization*. Springer, 2015, pp. 127–140.
[60] K. Deb, L. Thiele, M. Laumanns, and E. Zitzler, "Scalable test problems for evolutionary multi-objective optimization," in *2001 TIK-Technical Report*, no. 112, 2001.
[61] F. R. Salmasi, "Control strategies for hybrid electric vehicles: Evolution, classification, comparison, and future trends," *Vehicular Technology, IEEE Transactions on*, vol. 56, no. 5, pp. 2393–2404, 2007.
[62] S. Delprat, T. Guerra, and J. Rimaux, "Optimal control of a parallel powertrain: from global optimization to real time control strategy," in *EVS 18, Berlin, October*, 2001.
[63] A. Sciarretta, M. Back, and L. Guzzella, "Optimal control of parallel hybrid electric vehicles," *Control Systems Technology, IEEE Transactions on*, vol. 12, no. 3, pp. 352–363, 2004.
[64] C. Bacher, T. Krenek, and G. Raidl, "Reducing the number of simulations in operation strategy optimization for hybrid electric vehicles," in *Proceedings of EvoStar Conference (EvoApplications)*, ser. LNCS. Springer, 2014, pp. 553–564.
[65] O. Cassebaum and B. Bäker, "Predictive supervisory control strategy for parallel HEVs using former velocity trajectories," in *Vehicle Power and Propulsion Conference (VPPC)*. IEEE, 2011, pp. 1–6.
[66] R. Bellman, "Dynamic programming and stochastic control processes," *Information and control*, vol. 1, no. 3, pp. 228–239, 1958.
[67] D. Sinoquet, G. Rousseau, and Y. Milhau, "Design optimization and optimal control for hybrid vehicles," *Optimization and Engineering*, vol. 12, no. 1-2, pp. 199–213, 2011.
[68] M. Guemri, A. Neffati, S. Caux, and S. U. Ngueveu, "Management of distributed power in hybrid vehicles based on DP or Fuzzy Logic," *Optimization and Engineering*, vol. October, pp. 1–20, 2013.
[69] T. Rodemann, K. Narukawa, M. Fischer, and M. Awada, "Many-objective optimization of a hybrid car controller," in *Applications of Evolutionary Computation*, ser. Lecture Notes in Computer Science, A. M. Mora and G. Squillero, Eds., vol. 9028. Springer International Publishing, 2015, pp. 593–603.
[70] U. Sandberg, "Tyre/road noise: myths and realities," Swedish National Road and Transport Research Institute, Tech. Rep., 2001.

APPENDIX A
FRAMEWORKS OF MOEA/D, NSGA-III AND RVEA

Algorithm 1 Main framework of MOEA/D.

Input: the maximal number of generations t_{max}, a set of N weight vectors W, neighborhood set B with neighborhood size T;
Output: final population $P_{t_{max}}$;
Initialization: create the initial population P_0 with N randomized individuals;
/*Main Loop*/
while $t < t_{max}$ **do**
  for $i = 1, ..., N$ **do**
    $q_t(i)$ = recombination($P_t(B(i))$): randomly select two candidate solutions from the neighborhood $B(i)$ in population P_t, and create a new candidate solution using recombination operators such as crossover and mutation for neighborhood $B(i)$;
    $P_t(B(i))$ = update($q_t(i)$, $P_t(B(i))$): update the candidate solutions in neighborhood $B(i)$ by comparing the scalarization function value of $q_t(i)$ with those in $P_t(B(i))$, where the scalarization function is as formulated in (2) or (3);
  end for
  $P_{t+1} = P_t$;
  $t = t + 1$;
end while

APPENDIX B
DEFINITIONS OF THE DESIGN PARAMETERS IN THE HEV CONTROLLER MODEL

TABLE IV
RANGES AND PHYSICAL MEANINGS OF THE 11 DESIGN PARAMETERS

Design Parameter	Range	Physical Meaning
SOC_{max} (%)	[25,50]	SOC threshold to turn off ICE
SOC_{min} (%)	[20,30]	SOC threshold to turn on ICE
v_1 (km/h)	[20,60]	Lower speed for operation points
v_2 (km/h)	[50,100]	Upper speed for operation points
rev_1 (/min)	[2500,3000]	ICE speed for operation point 1
$torque_1$ (N · m)	[5.23,12.56]	Torque for operation point 1
rev_2 (/min)	[3000,4000]	ICE speed for operation point 2
$torque_2$ (N · m)	[5.65,17.59]	Torque for operation point 2
rev_3 (/min)	[4000,5000]	ICE speed for operation point 3
$torque_3$ (N · m)	[7.77,23.03]	Torque for operation point 3
v_{off} (km/h)	[20,50]	Speed threshold to turn off ICE

APPENDIX C
DEFINITIONS OF THE SIX CONTROL RULES IN THE HEV CONTROLLER MODEL

Rule 1: If speed is below v_{off}, turn off ICE;
Rule 2: If battery state-of-charge (SOC) goes above SOC_{max}, turn off ICE. Note that SOC_{max} is not the maximum SOC of the battery but the SOC level above which the ICE will no longer recharge the battery;

Algorithm 2 Main framework of NSGA-III.

Input: the maximal number of generations t_{max}, a set of N weight vectors W;
Output: final population $P_{t_{max}}$;
Initialization: create the initial population P_0 with N randomized individuals;
/*Main Loop*/
while $t < t_{max}$ **do**
  Q_t = recombination(P_t): perform genetic operations to generate the offspring population;
  $P_t = P_t \cup Q_t$;
  S_t = dominance-based-selection(P_t): perform non-dominated sorting on P_t to divide the population into a number of non-dominated fronts $F_1, F_2, ...$, and select candidate solutions one by one until front F_l is reached such that $|S_t| \geq N$ and $|S_t| - |F_{l-1}| < N$;
  S'_t = normalization(S_t): normalize the objective values of the candidate solutions in S_t to $[0, 1]$;
  P_{t+1} = niching-based-selection(S'_t): associate each candidate solution in S'_t with the closest weight vector, calculate the niche count of each weight vector according to the number of candidate solutions associated with it, and select a number of $|F_l| - (|S'_t| - N)$ candidate solutions from S'_t which have minimal niche counts together with the candidate solutions on the first $l-1$ fronts to create P_{t+1};
  $t = t + 1$;
end while

Algorithm 3 Main framework of RVEA.

Input: the maximal number of generations t_{max}, a set of N unit reference vectors $V_0 = \{\mathbf{v}_{0,1}, \mathbf{v}_{0,2}..., \mathbf{v}_{0,N}\}$;
Output: final population $P_{t_{max}}$;
Initialization: create the initial population P_0 with N randomized individuals;
/*Main Loop*/
while $t < t_{max}$ **do**
  Q_t = recombination(P_t): perform genetic operations to generate the offspring population;
  $P_t = P_t \cup Q_t$;
  P_{t+1} = reference-vector-guided-selection(t, P_t, V_t): translate the objective values of the individuals in P_t by subtracting the estimated ideal point, divide P_t into a number of N subpopulations by associating the candidate solutions with the closest reference vectors, and select one solution from each subpopulation using the scalarization function called the angle penalized distance (ADP) as formulated in (6), (7) and (8);
  V_{t+1} = reference-vector-adaptation(t, P_{t+1}, V_t, V_0): adapt the distribution of the reference vectors using the strategy formulated in (9) according to a given frequency f_r;
  $t = t + 1$;
end while

Rule 3: If SOC falls below SOC_{min}, turn on ICE (overrides first rule);

Rule 4: If ICE is ON and speed is below v_1, use engine operation point 1 (revolutions per minute (rpm) rev_1 and torque $torque_1$);

Rule 5: If ICE is ON and speed is between v_1 and v_2, use operation point 2 (rev_2, $torque_2$);

Rule 6: If ICE is ON and speed is above v_2, use operation point 3 (rev_3, $torque_3$).

APPENDIX D
DEFINITIONS OF THE SEVEN OBJECTIVES IN THE HEV CONTROLLER MODEL

Fuel consumption (FC):

This objective also correlates with CO_2 emissions. The simulator directly computes fuel consumption using ICE efficiency maps and also considers the battery charge remaining at the end of the trip by reducing fuel consumption by 0.5 l for every kWh left.

Battery stress (BS):

Due to the importance of battery lifetime in HEVs, the operation strategy has to consider the impact of controller actions (battery usage) on battery lifetime. However, since accurate battery aging models are very complex and difficult to calibrate, in our simulator, a simplified model is adopted by considering (dis-)charging current and battery temperature. If temperature or current leaves a specified safety zone, it is considered that battery stress is generated. Therefore, the optimization target is to minimize these stress events. To be specific, battery stress BS is computed as a function over all discrete time (simulation) steps t as:

$$BS = \sum_t \left(BS_{current}(t) + c_T * BS_{temp}(t)\right), \quad (14)$$

with current-induced stress $BS_{current}(t)$ given by:

$$BS_{current}(t) = \Delta I(t)^2, \quad (15)$$

where $\Delta I(t)$ is the deviation of the battery current at time t from the safe operating range, assumed here as $[-60, +160]$ A; and battery temperature stress $BS_{temp}(t)$ is computed similarly as

$$BS_{temp}(t) = \Delta T(t)^2, \quad (16)$$

with $\Delta T(t)$ as the deviation of battery temperature from the safe operating range (here assumed as $[15, 25]$ $^\circ C$). The relative battery temperature stress weight c_T is manually chosen as $c_T = 0.08$. Battery temperature and current are taken directly from the simulation software.

Operation changes (OPC):

A start and stop of the ICE can often be felt by the driver. It is therefore advisable to keep the frequency of on/off switches of the ICE low. We compute the number of on/off switches that happen within less than a given interval time (60 s) after the last on/off switch and try to minimize this number:

$$OPC = \sum_k H(|t_k - t_{k-1}| < 60s), \quad (17)$$

where k is the index of ICE on/off change events, and t_k the time when the k-th on/off switch event occurs. The step function $H(x)$ is 1 (true) if and only if x is 1 (true), and zero otherwise.

Emission:

While CO_2 emissions are directly proportional to the consumed fuel, other emissions like NO_x mostly occur when the ICE is on and the catalytic converter has not yet reached its operation temperature. We therefore compute emissions (excluding CO_2) considering the current system temperature in a simple model of a catalytic converter:

$$S_{cat}(t) = 1.0 - (T_{cat}(t)/T_{cat}^{op}), \quad (18)$$

where S_{cat} is the status of the catalytic converter, T_{cat} is the current catalytic converter temperature taken from the simulator, and T_{cat}^{op} is the nominal (maximum) operation temperature of the catalytic converter. T_{cat} will increase when the ICE is on and vice versa.

Based on catalytic converter status, the emissions are calculated as:

$$\text{Emission} = \sum_t \Theta_{ICE}(t) * S_{cat}(t) \quad (19)$$

which is the sum over the complete driving cycle with the product of ICE on/off status ($\Theta_{ICE} = [0 = \text{off}, 1 = \text{on}]$) times the current catalytic converter status.

Noise:

A vehicle generates noise from various sources, among which the two most important ones are tire/wind noise (a.k.a rolling noise) and ICE noise. While the former cannot be avoided given a target speed of the vehicle, the latter is influenced by the control strategy. We model rolling noise after [70] and ICE noise using measured data. The noise objective will only consider instances when the engine noise exceeds rolling noise, because otherwise it will not be perceived by the driver:

$$\text{Noise} = \sum_t \Theta_{Noise}(t) * (N_{ICE}(t) - N_{Rolling}(t))^2 \quad , \quad (20)$$

where Θ_{Noise} is one if ICE noise $N_{ICE}(t)$ exceeds tire/wind noise $N_{Rolling}(t)$ and zero otherwise (values are in dB).

Urban operation (UO):

HEVs are a potential answer to pollution and noise problems in metropolitan areas, if the control strategy manages to keep the ICE off while driving in the city. For this objective, we count the number of simulation

time steps when the ICE is on while the vehicle is driving with $v \leq 50$ km/h (urban speed limit).

$$UO = \sum_t \Theta_{ICE}(t) * H(v(t) \leq 50) \tag{21}$$

Battery state of charge (SOC):

Since we translate the remaining state of charge into reduced fuel consumption, the control strategy might decide to end the trip with a high final SOC. But there are several reasons to prefer a lower SOC: Firstly, for some battery types, lifetime is higher when the battery is kept at a lower SOC; secondly, when the vehicle arrives with full batteries at a charging station, it cannot use grid electricity to recharge. Therefore, we compute the accumulated SOC level (the area under the SOC curve) as our final objective:

$$SOC = \sum_t (SOC(t) - SOC_{min})^2, \tag{22}$$

with an SOC_{min} of 20%.