Investigating the Use of Linear Programming and Evolutionary Algorithms for Multi-objective Electric Vehicle Charging Problem

Hemant Singh, Tapabrata Ray, Mohammad Rana, Steffen Limmer, Tobias Rodemann, Markus Olhofer

2022

Preprint:

This is an accepted article published in IEEE Access. The final authenticated version is available online at: https://doi.org/10.1109/ACCESS.2022.3218058

IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Date of publication xxxx 00, 0000, date of current version xxxx 00, 0000.

Digital Object Identifier xxxxx

Investigating the Use of Linear Programming and Evolutionary Algorithms for Multi-objective Electric Vehicle Charging Problem

HEMANT KUMAR SINGH[1], TAPABRATA RAY[1], MD JUEL RANA[1], STEFFEN LIMMER[2], TOBIAS RODEMANN[2], MARKUS OLHOFER[2]

[1]The University of New South Wales, Canberra ACT 2600, Australia. Email: {t.ray,h.singh}@unsw.edu.au, m.rana@student.unsw.edu.au
[2]Honda Research Institute Europe, Offenbach/Main, Germany. Email: {steffen.limmer,tobias.rodemann,markus.olhofer}@honda-ri.de

Corresponding author: Hemant Kumar Singh (e-mail: h.singh@unsw.edu.au).

The authors would like to acknowledge the financial support from Honda Research Institute Europe.

ABSTRACT With the increasing uptake of electric vehicles (EVs), the need for efficient scheduling of EV charging is becoming increasingly important. A charging station operator needs to identify charging/discharging power of the client EVs over a time horizon while considering multiple objectives, such as operating costs and the peak power drawn from the grid. Evolutionary algorithms (EAs) are a popular choice when faced with problems involving multiple objectives. However, since the objectives and constraints of this problem can be expressed using linear functions, it is also possible to come up with improvised multi-objective formulations which can be solved with exact techniques such as mixed-integer linear programming (MILP). With both approaches having their potential strengths and pitfalls, it is worth investigating their use to inform the algorithmic choices, which this study aims to address. In doing so, it makes a number of contributions to the topic, including extension of an existing EV charging problem to a multi-objective form; observing some interesting properties of the problem to improve both the MILP and EA solution approaches; and comparing the performance of MILP and EA. The study provides some useful insights into the problem, initial results and quantitative basis for selecting solution approaches, and highlights some areas of further development.

INDEX TERMS Evolutionary algorithms, electric vehicle charging, mixed-integer linear programming, multi-objective optimization

I. INTRODUCTION

The transportation sector is one of the major contributors of greenhouse gas emissions globally. In Australia, for example, it accounts for 17% of the emissions, nearly half of which is attributed to cars [1]. Likewise, in the USA, transportation accounts for 29% of the emissions, of which 58% is due to light duty vehicles [2]. Therefore, it comes as no surprise that transition to electric vehicles (EVs) forms a crucial part of the strategy towards realizing the net-zero emission targets. With the improvements in the renewable energy infrastructure, government incentives as well as advancements in battery technology, the uptake of EVs has surged rapidly in the past decade. This widespread adoption, which is projected to only grow in the foreseeable future, comes with the challenges and opportunities of their integration in the electricity power grid.

A range of energy management problems that stem from the domain have therefore been of significant research interest recently [3]–[5]. These studies cover many different aspects such as dynamic pricing, scheduling, forecasting, grid stability, efficiency, etc., formulated often as different optimization problems depending upon the focus area(s). It is also evident that the different optimization problems may require different solution approaches to be solved well, with no single approach likely to be superior universally (also referred to as the no free lunch theorem [6]). Among the optimization approaches available, two most common ones include mixed-integer linear programming (MILP) [7], [8] and meta-heuristics [3] (predominantly evolutionary algorithms (EAs)). However, an aspect that has garnered less attention is the justification of the choice of the approach adopted, which is not always well-motivated through systematic comparison between different approaches. In this regard, there can be certain cases where the choice is more obvious - for example, continuous linear problem with single objective would naturally yield itself very well to MILP technique. On the other hand, for multi-objective/non-linear/black-box problems, EAs might be a more suitable approach, usually combined with expert knowledge and/or customized operators. Just to give examples of some cases where the algorithm choice is relatively straightforward, consider the works [9], [10] which deal with non-linear formulations, and hence metaheuristic approaches were applied to solve them. On the other hand, some works consider formulations that are linear [8] or otherwise convex [11] and hence classical approaches such as MILP and primal-dual interior point methods are used to solve them directly.

There can, however, be indeterminable cases where the choice of approach is not very obvious. The main motivation of this work is to make a systematic and fair comparison between MILP and EA for one such indeterminable case, in order to gain insights into the performance of these approaches and help practitioners make an informed choice regarding the solution approach. To achieve the above-stated aim, we use the smart scheduling of EV charging as a case study, which is of wide interest in the energy management domain. In this context, the intended contributions of this work are:

1) extending the problem presented in [8] to include multiple power modulation levels and multiple objectives; which can be used as benchmarks for further research on this topic,
2) formulating solution approaches to solve the MO problem based on MILP and EA, which include their basic versions as well as improvements and customizations based on the understanding of the problem,
3) conducting numerical experiments to systematically compare and analyze the relative performance differences between various approaches for this problem, to provide useful insights for informed algorithm selection for such classes of problems.

The rest of the paper is organized as follows. The background of the problem and related works are discussed in Section II, followed by a detailed problem description in Section III. The MILP and EA based solution approaches are detailed in Sections IV and V, respectively. The numerical experiments and discussions are presented in Section VI, followed by concluding remarks and future work in Section VII.

II. BACKGROUND AND RELATED WORK

In practice currently, the charging of electric vehicles is often done in an uncoordinated manner, by plugging in and charging it at its full power until its required state-of-charge (SOC) is reached [8]. However, this can have a number of downsides, including (but not limited to) higher charging costs and unexpected spikes in the load on the grid. This can be overcome by coordinated scheduling of vehicles, which inherently involves solving an optimization problem wherein the charging power is carefully modulated to provide optimal schedules [12]. Moreover, the vehicle-to-grid (V2G) technology also allows for feeding the power back to the grid (by discharging the vehicles) to reduce the peak load, which could be further incorporated in the problem to come up with desirable schedules [13].

The EV charging problem considered here is an optimization problem from the viewpoint of a charging station operator (CSO). A CSO needs to construct a charging/ discharging schedule for all the EVs that enter the charging station over a certain time horizon. The CSO purchases energy from the electricity grid at real-time electricity prices to charge the vehicles. Likewise, it can sell energy (by discharging vehicles) to the grid at real-time prices. In constructing such a schedule for charging/discharging of each vehicle, the CSO is interested in reducing its charging costs (to maximize its profits) as well as reducing the peak power drawn, while satisfying the charging requirements of the EV customers [8]. This as well as a number of other related problems have typically been posed as single-objective (SO) problems in the literature, where the other objectives (such as peak power drawn) if present are converted to constraints [8]. Usually, the multi-objective (MO) problem is not considered in its inherent form. Another aspect that is scarcely studied is to gain an understanding of relative performance of different solution approaches to inform the algorithmic choices. In the literature, the use of EAs [14]–[16] as well as MILP [8], [17], [18], among others, has been explored to solve EV charging problems. In [19], both MILP and EAs were used concurrently, but for solving different parts of the problem. More specifically, EA was used to optimize dynamic prices, combined with MILP for the charging scheduling. In [15], interestingly, a linear problem was solved using genetic algorithms. It is perceivable that there are instances where the choice of the algorithm is based on familiarity and accessibility of the implemented approach by the users, rather than on comparative analysis.

In this study, we focus on an extended, multi-objective version of the problem presented in [8] and compare the performance of different solution approaches to observe and understand their relative strengths and weaknesses, since they may have competing advantages. As indicated before, for the cases where the problem of interest is linear and SO, it seems most efficient to use exact techniques such as MILP. For (generic) MO problems on the other hand, it is common to resort to EAs due to their ability to handle multiple objectives through specialized ranking techniques such as non-dominated sorting, and use the population to cover the Pareto-optimal front (PF) [20]. However, for linear problems (which is the case here), it is also possible to reformulate the problem and convert it into a set of sub-problems that can be solved individually to yield the overall PF by running multiple MILPs. For example, some of the recent works use ϵ-constraint reformulation to achieve this for different versions of multi-objective energy management problems [21]–[24].

Both the above approaches have their perceived strengths and weaknesses. EAs handle MO problems in their native form, whereas MILP (or any other SO technique) requires an appropriate scalarization (or a generic parametrization) of the problem to convert it into SO - and the choice of methods to do so can have an impact on the outcome. EAs can use a direct encoding with lower number of variables compared to MILP formulations. For example, in the problem considered here, the ratio of number of variables between EA:MILP is 1:8 to represent the same problem. EAs are also able to exchange the information between solutions in its population through evolutionary operators, whereas MILP reformulation needs to run multiple times independently to come up with the individual solutions of the sub-problems. On the other hand, MILP can search the variable space very efficiently for linear problems; with a potential that running it many times may still yield better results in the same time as it takes to run a single EA. Given these competing advantages, we systematically construct an extended version of the problem presented in [8] to include multiple power modulation levels and multiple objectives. Then, we systematically evaluate the EA and MILP approaches on the problem, as well as propose customizations to improve the performance of both approaches. Numerical experiments are conducted across multiple instances to compare the performance, and to gain insights that would be helpful for choosing appropriate algorithms for solving this class of problems.

Some other recent trends in the domain are also worth mentioning. Notably, Machine learning (ML) techniques are increasingly being used in the domain of energy management and EV charging scheduling. ML methods are data-driven and can help in identifying consumer charging behavior patterns and provide insights via predictive analytics [25], [26]. For dynamically changing environments, reinforcement learning (RL) has also gathered significant attention [27]. On the other hand, more traditional optimization methods (such as MILP/EA) can help in maximizing/minimizing the performance metrics of interest, taking into account the predictions (including those from ML) in order to find the best EV scheduling strategies. These two classes of methods can be considered complementary and their combined application can help in identification of competitive solutions to energy management problems. Within the scope of this work, we focus on the traditional optimization based approaches for scheduling, and hence the ML techniques are excluded from further discussion. For the interested readers, recent reviews of ML and RL techniques in the domain of EV charging can be found in [25], [27].

III. PROBLEM DESCRIPTION

The MO EV charging scheduling problem studied herein considers charging cost for the CSO as the first objective, and the minimization of instantaneous peak power drawn within the charging horizon as the second objective. In [8], SO (cost minimization) problem was studied. There were four different variants of the problem, with two variants (C-D and C-C) assuming that the vehicles can only be charged; they cannot feed power back to the grid by discharging. The remaining two variants (CD-D and CD-C) assumed that the vehicles can charge by drawing energy from the grid, as well as discharge to feed energy back to the grid. Among these, CD-D variant assumed the power modulation to be discrete ($P \in \{P^{max}, 0, -P^{max}\}$), while CD-C variant assumed the power modulation to be continuous ($P \in [-P^{max}, P^{max}]$). As indicated in [8], the variants with both charging and discharging (CD-*) pose greater challenges to MILP. Hence in this study we have focused on the CD problems and created extended versions as follows.

- The first extension in the problem set is to generalize the power modulation levels (N_L), as indicated in Table 1. The CD-D version from [8] is referred to as variant 1 (V1), with 3 discrete levels. In variant V2, we allow the charging/discharging levels to take more discrete states, in total 5. V3 further expands this to 9 levels. The continuous version is labeled as V4 and is the same as the CD-C variant in [8]).

TABLE 1. Variants of the EV charging problem

Problem	#levels (N_L)	Charging/discharging power (P_L)
V1	3	[1,0,-1]P^{max}
V2	5	[0.5,1,0,-0.5,-1]P^{max}
V3	9	[0.25,0.5,0.75,1,0,-0.25,-0.5,-0.75,-1]P^{max}
V4	∞	[1,-1]P^{max}

- The second extension is the inclusion of the second objective, minimization of peak power. The intent is to investigate and understand the nature of the trade-offs between these two objectives relevant to the CSO. Further, the MO problems will help in assessing the capability of MILP and EA based approaches in identifying the set of trade-off solutions.

Note that throughout this paper, we refer to the term 'instance' as a specific realization of the problem parameters such as maximum charging powers, initial/final EV charging status, charging/discharging costs, etc. The parameters are sampled using the same approach as well as within the same ranges as suggested in [8], presented in Table 2. Based on the above preliminary experiments, we focus this study on version V3 of the problem, and assume the scheduling of 2 vehicles at the given instant.

TABLE 2. Parameters and parameter ranges used in the construction of random problem instances [8]

Parameters	Values	Parameters	Values
P_n^{max} (kW)	[3,5]	Δt (min)	15
E_n^{max} (kWh)	[8,16]	T	32
E_n^{min} (kWh)	1	P^{max} (kW)	$3N$
E_n^{init} (kWh)	[0.2,0.5]E_n^{max}	P^{min} (kW)	-P^{max}
E_n^{ref} (kWh)	[0.55,0.8]E_n^{max}	e_t^c (cents/kWh)	[1.9,3.5]
δ_n	[0.015,0.075]	e_t^d (cents/kWh)	$1.1e_t^c$

We conducted some preliminary experiments to determine the relative difficulty of different versions of the problem, including the continuous version (denoted as V4). To this end, we ran MILP on the cost minimization (single-objective) problem with a time limit of 40 s. Gurobi 9.12[1] is used as the MILP solver using the MATLAB[2] API for this and all subsequent MILP runs. A total of 21 problem instances were created by sampling the parameters from Table 2. In Figure 1, we indicate the % of problem instances (out of 21 instances) where MILP was able to achieve proven optimal solutions. It is clear from Figure 1, that MILP could identify optimal solutions of variants V1, V2 and V4 with 2 vehicles for 100% of the instances. It can also be concluded that variant V3 poses significant challenges to the MILP solver as 9 discrete levels increases branching possibilities; while the continuous version (V4) is the easiest to solve. The main reason for sampling multiple instances is so that the observations and the conclusions regarding the relative performance of the algorithms do not become too specialized or biased to one single instance, both for the single-objective as well as for the multi-objective version of the problem.

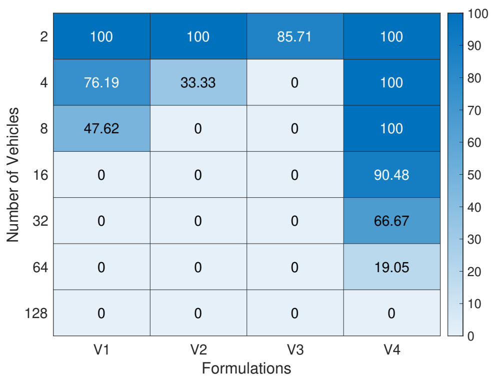


FIGURE 1. Percentage of single-objective (cost minimization) instances solved to optimality within a time limit of 40 seconds using MILP

From the above observations, it could be inferred that V3 is significantly more challenging for MILP compared to V1, V2, and also the continuous version (denoted as V4). Since cost minimization is one of the objectives in the extended multi-objective problem (discussed shortly), this is likely to remain the case for the MO problem. This initial test was used as a basis to choose variant V3 for the studies presented in this work, since there might be potential improvements to gain from metaheuristics such as evolutionary algorithms (EA) in this case.

The representation of the solutions and resulting equations vary depending on the solution approaches used, and are discussed in the respective sections next. Throughout the discussion of the solution approaches, we use indicative proof-of-concept results on one problem instance (denoted Instance-1) of the V3 problem for illustration. Subsequently, we show more comprehensive sets of results on multiple (21) instances in the numerical experiments.

IV. SOLUTION APPROACHES USING MILP

To formulate the above problem in a linear form, the decision variables at any time instant t can be represented by a vector $\mathbf{x_t}$ with

$$\mathbf{x}_{t,n} = [c_{t,n}^1 \dots c_{t,n}^{(N_L-1)/2} \quad d_{t,n}^1 \dots d_{t,n}^{(N_L-1)/2}], \tag{1}$$

where $c_{t,n}^l$ and $d_{t,n}^l$ are variables representing charging and discharging status of EV n at level l. The complete solution vector $\mathbf{x}$ is then constructed by concatenating $\mathbf{x}_{t,n}$ for all t and n along with the peak power (P^{peak}), as shown in Equation (2):

$$\mathbf{x} = [\mathbf{x}_{t,n} \quad P^{peak}], \quad t = 1,\dots,T, n = 1,\dots,N. \tag{2}$$

P^{peak} is modeled as a continuous variable (and the second objective). For the subproblem where only the cost is being minimized to obtain one of the extreme solutions, it only acts as a placeholder. For the problem configuration considered ($T = 32$), the total number of variables for 2-vehicle V3 problem is thus $9 \times 2 \times 32 + 1 = 513$.

The mathematical expressions for V1-V3 can then be represented by the generic equations (3)–(10):

$$\min_{\mathbf{x}} f_1(\mathbf{x}) = \sum_{t=1}^{T}\sum_{n=1}^{N}\left\{\sum_{l=1}^{\frac{N_L-1}{2}} c_{t,n}^l \cdot P_L^n(l) \cdot e_t^c - \sum_{l=1}^{\frac{N_L-1}{2}} d_{t,n}^l \cdot P_L^n(l) \cdot e_t^d\right\}\Delta t, \tag{3}$$

$$\min_{\mathbf{x}} f_2(\mathbf{x}) = P^{peak}, \tag{4}$$

subject to:

$$E_{t,n} = E_n^{init} + \sum_{s=1}^{t}\left\{\sum_{l=1}^{(N_L-1)/2} c_{s,n}^l \cdot P_L^n(l) \cdot (1-\delta_n) - \sum_{l=1}^{(N_L-1)/2} d_{s,n}^l \cdot P_L^n(l) \cdot (1+\delta_n)\right\}\Delta t \qquad \forall t, n, \tag{5}$$

$$E_{T,n} \geq E_n^{ref} \qquad \forall n, \tag{6}$$

[1]https://www.gurobi.com/
[2]https://au.mathworks.com/products/matlab.html

$$E_n^{min} \leq E_{t,n} \leq E_n^{max} \qquad \forall t, n, \quad (7)$$

$$\sum_{l=1}^{(N_L-1)/2} c_{t,n}^l + \sum_{l=1}^{(N_L-1)/2} d_{t,n}^l \leq 1 \qquad \forall t, n, \quad (8)$$

$$P^{min} \leq \sum_{n=1}^{N} \left\{ \sum_{l=1}^{(N_L-1)/2} c_{t,n}^l \cdot P_L^n(l) - \sum_{l=1}^{(N_L-1)/2} d_{t,n}^l \cdot P_L^n(l) \right\} \leq P^{max} \qquad \forall t, \quad (9)$$

$$\sum_{n=1}^{N} \left\{ \sum_{l=1}^{(N_L-1)/2} c_{t,n}^l \cdot P_L^n(l) - \sum_{l=1}^{(N_L-1)/2} d_{t,n}^l \cdot P_L^n(l) \right\} \leq P^{peak} \qquad \forall t, \quad (10)$$

$$c_{t,n}^l, d_{t,n}^l \in \{0, 1\} \qquad \forall t, n.$$

Setting the value of N_L to 3 ,5 or 9 would equate to solving variants V1, V2 or V3 respectively. The objectives are denoted as f_1 (cost) and f_2 (peak power). The Eqn. (5) calculates the charging state of the vehicle n at a given time step t. The constraint (6) ensures that all vehicles meet the required SOC at the end of time step T, while the constraint (7) ensures that the SOC stays between the desirable limits to avoid battery stress. The constraint (8) is applied on the binary variables $(c_{t,n}, d_{t,n})$ to ensure that charging and discharging are not both active at the same time. The constraint (9) imposes limitations on the total power drawn by individual vehicles, where a negative value represents discharging. P_{min} is typically set as $-P^{max}$ for the study, i.e., the vehicles are able to charge/discharge up to the same (absolute value of) power threshold. The constraint (10) is added in the formulation for the optimization of second objective (P^{peak}).

Given that MILP can only optimize one objective at a time, the MO problem needs to be re-formulated such that the solution(s) to multiple SO problems collectively provide the PF approximation of the MO problem. There exist many scalarization methods to achieve this [28]. However, two that are of particular interest in this study are the weighted sum (WS) [29] and $\epsilon-$constraint (EC) [30] approaches. The reason for this choice is that these two methods maintain the linearity of the problem unlike others such as Chebyshev scalarization.

A. WEIGHTED SUM FORMULATION (MILP-WS)

A set of N_W weights $\{\mathbf{W}; W_i \leq 1 \forall i\}$ is provided as an input to the framework. For each $i \in \{1, \dots N_W\}$, the MILP optimizes a weighted sum of the two normalized objectives (the weights being $\{W_i, 1 - W_i\}$). The normalization/scaling of the objectives is required so that the combined objective is not biased by one of the objectives due to the disparity in the magnitude of the objectives. The scaling factors are determined by running the MILP on each objective individually, and noting the corresponding value of the other objective. To summarize, three steps are involved in the solution process:

1) Minimize $f_1(\mathbf{x})$ independently subject to all operational constraints; by setting $W = 1$. Let the objective values of the solution be denoted as (f_1^{min}, f_2^{max}).
2) Minimize $f_2(\mathbf{x})$ independently subject to all operational constraints; by setting $W = 0$. Let the objective values of the solution be denoted as (f_1^{max}, f_2^{min}).
3) Minimize for all $W_i \in \mathbf{W}$ the weighted function:

$$F_{Wi} = W_i \frac{f_1(\mathbf{x}) - f_1^{min}}{f_1^{max} - f_1^{min}} + (1 - W_i) \frac{f_2(\mathbf{x}) - f_2^{min}}{f_2^{max} - f_2^{min}} \quad (11)$$

subject to (5)–(10).

Note that the constraint (10) only applies to the steps 2) and 3) above. Essentially, it ensures that the value of P^{peak} is greater than or equal to the net power drawn (charging − discharging) at every instant. The equality condition in Equation (10) will be true at the minimum f_2.

B. EPSILON CONSTRAINT FORMULATION (MILP-EC)

In this formulation, the general idea is to minimize one of the objectives (f_1), while imposing an inequality constraint on the other (f_2) bounded by an ϵ value. The exercise can be repeated for multiple uniformly distributed values of ϵ within the bounds of f_2, with each optimization yielding one solution to the MO problem at a time. To determine the range of ϵ, the extreme values of peak power f_2^{min} and f_2^{max} are obtained analogous to the steps 1-2 of the weighted sum approach. Then, the range of ϵ is set based on a user defined step size $\Delta\epsilon$ as in (12).

$$\epsilon = \{f_2^{min} : \Delta\epsilon : f_2^{max}\}. \quad (12)$$

The solution vector of decision variables for discrete variants is similar to (2). Then, the optimization problem for a specific $\epsilon_k \in \epsilon$ can be formulated utilizing $f_1(\mathbf{x})$ from (3) as follows:

$$\min_{\mathbf{x}} f(\mathbf{x}) = f_1(\mathbf{x}) \quad (13)$$

subject to:

$$(5) - (10),$$

$$P^{peak} \leq \epsilon_k. \quad (14)$$

For an initial assessment, we solve the V3 problem, Instance-1, with 2 vehicles using $\Delta\epsilon = 0.05$, resulting in a total of 85 levels between f_2^{min} and f_2^{max}. The PF approximation achieved through this approach for Instance-1 is shown in Figure 2, where a limit of 40 s is set for each run of MILP (Gurobi) corresponding to one value. The results of this approach are indicated as MILP-EC40 in the figure. Interestingly, the shape of the PF is non-convex, consisting of 26 solutions. On closer observation of the objective values, it is also apparent that these 26 solutions correspond to all possible discrete power combinations that result in peak power levels (f_2 values) between and including f_2^{min} and f_2^{max} values for this instance. Thus, it can be asserted that no other PF solutions exist for a different power level (than those already covered) for this problem.

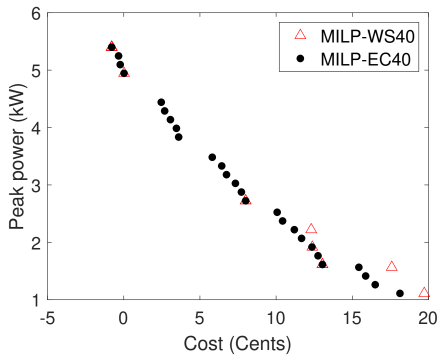


FIGURE 2. PF approximation obtained for V3 Instance-1 using MILP-WS and MILP-EC, with max. 40 s for each subproblem.

A realization from the above experiments is that increasing the number of steps by using lower values of $\Delta\epsilon$ will bring no further benefits in terms of achieving a better PF approximation. In fact, given the nature of the problem, the number of levels can be reduced significantly to yield the same results, by identifying the feasible combinations of charging/discharging power that yield a peak power between f_2^{min} and f_2^{max}.

The above proposed improvement to reduce run time (number of MILPs solved) is implemented as follows:

1) Enumerate all possible power (charging/discharging) combinations for the vehicles first. For the case of V3 each vehicle can assume 9 charging/discharging states at any give time instant, so the total number of combinations are $9 \times 9 = 81$. This automatically becomes the hard upper bound on the number of ϵ levels.
2) For each of the above possibilities, the power drawn is calculated. Let's call this set of combinations P_A where $|P_A| = n_A$ (81 in this case).
3) The two extremities of the PF from the individual objective optimizations give us lowest/highest peak power values (f_2^{min}, f_2^{max}).
4) The set of power levels are then designated as $P_C = \{p_i | p_i \in P_A; f_2^{min} \leq p_i \leq f_2^{max}; i = 1, 2 \dots n_A\}$
5) P_C is thus essentially the minimal set to obtain exactly $n_C = |P_C|$ solutions on the PF, provided the limits (f_2^{min}, f_2^{max}) are correct PF extremities.

The impact of the above improvement, by customizing the number (and values) of the peak power levels is that the same result as shown in Figure 2 is obtained by running less than 1/3rd the number of MILPs compared to the case of equidistant sampling with $\Delta\epsilon = 0.05$ on V3 Instance-1, making it significantly more computationally efficient. Hence we use this version going forward for the MILP-EC runs.

For comparison, we also observe the results of the weighted sum approach on the the same problem instance. In Figure 2, we show the results obtained by using 40 s time limit for each MILP, and using $N_W = 30$ weights obtained by equidistant sampling the W_i in [0,1]. The results are labeled as MILP-WS40 in the figure. The performance of MILP-WS is observed to be inferior to MILP-EC40 due to the non-convex nature of the PF for this particular instance. It is also interesting to note that *some of* the solutions from MILP-WS40 did not land exactly on the PF approximation obtained by MILP-EC40. The reason in this case is that the 40 s time limit given to the MILP solver was not always sufficient for it to solve the WS formulation to optimality. In separate experiments (not presented here for brevity), we verified that with a longer runtime, these solutions are able to reach the PF. This also indicates that the WS formulation is relatively more difficult to solve for the MILP solver in this case compared to the EC formulation.

V. SOLUTION APPROACH USING EA

For solving the problem using an EA, the generic framework is used, a pseudo-code for which is provided in Algorithm 1. Most of the components involved are standard in the EA domain. They are outlined below in the baseline EA subsection, followed by the proposed improvements.

Algorithm 1 Pseudo-code for Canonical Evolutionary Algorithm

Input: Population size (N_P), Generations (N_G), Crossover probability (P_c), Mutation probability (P_m), DE Scaling factor (F_R), Mutation index for PM (μ_m)

1: Initialize the population Pop_0 (of size N_p) and evaluate all solutions.
2: **for** g=1 to N_G **do**
3: Select parent solutions P_g from current population (Pop_{g-1})
4: Generate child solutions C_g from P_g via crossover and mutation
5: Evaluate child solutions C_g
6: Rank $T_g = Pop_{g-1} \cup C_g$
7: Select top N_P individuals in T_g as the surviving population Pop_g
8: **end for**
9: Output the non-dominated solutions $\mathcal{N} \in Pop_G$ as the best solution/PF approximation

A. BASELINE EA (EAB)

1) Solution representation

The representation scheme is relatively straightforward for the EA since there's no imperative on the linearity of the problem. Hence the solution vector for each time instant can be directly encoded as the charging power of each vehicle; where a negative value automatically implies discharging. The overall solution vector can be constructed as shown in

Equation (15). The lower and upper bounds of each variable are P_n^{min} and P_n^{max}, respectively, of the corresponding vehicle n. The values they can take are determined by the variant under consideration (Table 1). For the problem configuration considered ($T = 32$), the total number of variables for 2-vehicle V3 problem is thus $2 \times 32 = 64$.

$$\mathbf{x} = [p_{1,1}, \dots p_{n,1} \quad \dots p_{1,t}, \dots p_{n,t} \quad \dots p_{1,T}, \dots p_{n,T}] \tag{15}$$

The calculation of the total cost and peak power, as well as the related constraints can then be easily derived based on the charging schedule as shown below; noting that the problem becomes non-linear in this representation.

$$\min_{\mathbf{x}} f_1(\mathbf{x}) = \sum_{t=1}^{T} \sum_{n=1}^{N} \begin{cases} p_{n,t} e_t^c \Delta t & if\, p_{n,t} \geq 0 \\ p_{n,t} e_t^d \Delta t & if\, p_{n,t} < 0 \end{cases} \tag{16}$$

$$\min_{\mathbf{x}} f_2(\mathbf{x}) = P^{peak} = \max_{\forall t} \sum_{n=1}^{N} p_{n,t} \tag{17}$$

subject to:

$$P_n^{min} \leq p_{n,t} \leq P_n^{max} \qquad \forall t, n, \tag{18}$$

$$E_{t,n} = E_n^{init} + \sum_{s=1}^{t} \begin{cases} p_{n,s}(1 - \delta_n) & if\, p_{n,s} \geq 0 \\ p_{n,s}(1 + \delta_n) & if\, p_{n,s} < 0 \end{cases} \qquad \forall t, n, \tag{19}$$

$$E_{T,n} \geq E_n^{ref} \qquad \forall n, \tag{20}$$

$$E_n^{min} \leq E_{t,n} \leq E_n^{max} \qquad \forall t, n, \tag{21}$$

$$P^{min} \leq \sum_{n=1}^{N} p_{n,t} \leq P^{max} \qquad \forall t. \tag{22}$$

2) Initialization

The solutions are randomly initialized within the variable bounds at the beginning of the EA. A uniform random discrete number generator is used to generate the required charging/discharging powers at each time instant for each vehicle.

3) Evolution

From the given population in any generation, the parent solutions are first selected through pairwise tournament selection. The parent solutions then undergo crossover and mutation to generate the child solutions. The well-known differential evolution (DE) crossover [31] and polynomial mutation (PM) [32] are used for these operations. A discretization is performed on the child solutions by casting the real values to the nearest allowable discrete charging/discharging power as per the variant being solved.

4) Ranking/environmental selection

The parent and child solutions undergo a ranking process to order them based on fitness. The most common approach for ranking is the parameter-less feasibility-first (FF) scheme, eg., as used in the non-domination sorting genetic algorithm II (NSGA-II) [32]. In this approach, the feasible and infeasible solutions are ranked separately. The feasible solutions are ranked using non-dominated (ND) sorting and crowding distance (CD); whereas the infeasible solutions are ranked based on the sum of constraint violations. Thereafter, the overall ranks are established as the set of ranked feasible solutions followed by the set of ranked infeasible solutions; giving unconditional preference to the feasible over infeasible solutions. From the ranked parent + child population, the top N_P are then kept as the surviving population for the next generation, while the remainder are discarded. Note that for single-objective formulation, non-dominated sorting degenerates automatically to simple sorting, hence no change is required to deal with SO vs MO version of the problems.

For an initial estimate of the performance of the above baseline EA, we evolve a population of 200 over 10000 generations (2 million objective function evaluations in total) on problem instance Instance-1. The results are presented in Figure 3. It is evident that the baseline EA struggles significantly both in terms of convergence and diversity.

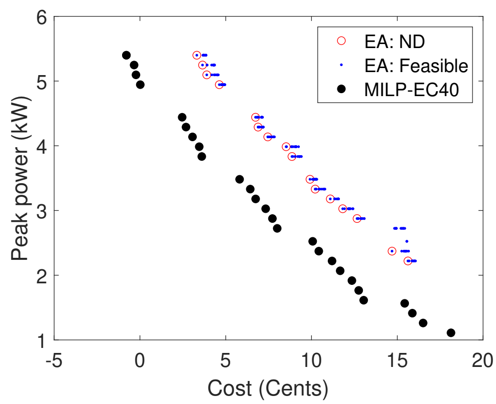


FIGURE 3. PF approximation obtained by the baseline EA for Instance-1, with MILP-EC40 results shown for reference. Blue dots indicate feasible solutions in the final EA population, with red circles indicating the final non-dominated set.

B. IMPROVED EA (EAI)

On examining how the solutions evolve over generations, the poor performance of the baseline EA can be attributed to the following:

1) For the lower half of the PF, the minimal cost charging schedules need to be searched while restricting the peak power to a low value. The chance of a randomly generated solution to be feasible in this region is relatively small. This bias is an interesting problem characteristic and implies that the lower half of the PF is harder to obtain compared to the upper half. This is also reflected by the performance of MILP, as it was observed that MILP took more time to solve problems

with lower values of ϵ compared to higher ones. For the EA, given its dependence on stochastic solution generation process, the challenge is significantly amplified. For certain low peak power levels, not even a single feasible solution was generated by the EA during the entire run.

2) The second challenge relates to the use of non-domination (ND) based sorting as the primary mechanism to drive the solutions towards the PF. It was observed that during ranking, solutions at a certain peak power level can get dominated by the solution(s) at other power levels, and get eliminated from the surviving population of the next generations. This means that the solutions corresponding to certain peak power levels may be generated and lost frequently; adversely impacting the final distribution obtained.

To overcome the above challenges, a few enhancements were incorporated in the EA. These include (a) generation of some initial feasible solutions at each power level through a heuristic initialization (b) maintenance of solutions at each power level through a customized ranking (c) a repair mechanism to reduce costs and (d) custom pairing during recombination. These are described below, which form a part of the improved EA (EAI).

1) Heuristic initialization

To address the diversity issue, during the initialization, certain solutions are constructed using heuristics in order to provide representative *feasible* solutions at each power level. This is to overcome the fact that purely through random initialization and evolution, EA could not generate feasible solutions for the lower part of the PF. The underlying rationale is that if these feasible solutions can be inserted at the beginning, the EA could maintain/improve it over the run to achieve better diversity at the end. The steps for the initialization are discussed below. Mainly it assumes a very conservative charging schedule, where vehicles are charged at their lowest possible charging power for most of the slots, so that the given peak power is never exceeded.

1) First of all, we identify the set of power levels $P_C = \{p_i | p_i \in P_A; f_2^{min} \leq p_i \leq f_2^{max}; i = 1, 2 \ldots n_A\}$, similar to how it is done for the MILP. The two extremities are not known in this case, so they are set empirically: f_2^{min} is set as the maximum of the lowest charging powers, and f_2^{max} is set as P^{max}.
2) Then, for each power level p_i, we construct a feasible schedule in the following manner:
   - For $t = 1$, the charge/discharge combination that gives the resulting power p_i is used.
   - For each vehicle j, the remaining number of slots N_j required to charge it to its reference energy E_j^{ref} is calculated, assuming it is charged at its lowest possible power.
   - The schedule from $t = 2 : T$ is constructed by choosing N_j slots for charging vehicle j; where $j = \{1, 2\}$, while the other vehicle is kept idle (no charging/discharging).
   - The unused slots from the above step are set to use 0 power (both vehicles idle).

To ensure that the starting number of solutions are equal to the population size N_p, the remaining solutions (N_p - $|P_C|$ solutions) are generated randomly, similar to EAB.

2) Ranking

Since the main problem identified in ND-sorting was the loss of solutions at certain peak power levels, a more suitable ranking scheme needs to ensure that good solutions are maintained at each peak power level. We propose the following ranking strategy to achieve this goal.

1) We first do a lexicographic sorting of the solutions, based on peak power (f_2) followed by cost (f_1). That is, the solutions are sorted based on the peak power, and those that have the same value of peak power are sorted based on their cost.
2) Starting from the lowest p_i, pick the first solution from each peak power level. Then move on to the next solution in each power level, and so on, until all solutions have been picked (ordered).

This simple ranking strategy ensures that at least one representative solution from each of the power levels is maintained in the population. Some of these solutions may not be non-dominated within the current population, but this ranking provides them a chance to maintain diversity and generate lower-cost solutions at each power level, eventually leading to a better PF approximation.

An indicative improvement from the use of the customized initialization and ranking scheme discussed above can be seen in Figure 4. Importantly, the diversity of the solutions has improved significantly, identifying a wider range of representative solutions across different peak power levels. At the same time, the scope of improvement in convergence (mainly the reduction in cost for solution at each peak power level) is also apparent. Thus, the next two customizations focus on improving convergence.

3) Repair

One plausible way to improve the convergence at any given power level is by attempting to reduce its cost while maintaining the peak power. This can be done by shuffling the schedule *horizontally*. That is, while keeping the combinations of charge/discharge for the vehicles the same, the corresponding time instants can be changed. In order to incur lower costs, the charging/discharging combinations that draw higher amount of energy can be scheduled at the time instant where the energy costs are low. Similarly, the instances of low (or no) power drawn can be scheduled when the charging costs are high. The repair is particularly useful for the initial feasible solutions since the costs were not accounted for at all during initialization. The implementation of the repair is quite simple. First of all, the charging costs are sorted

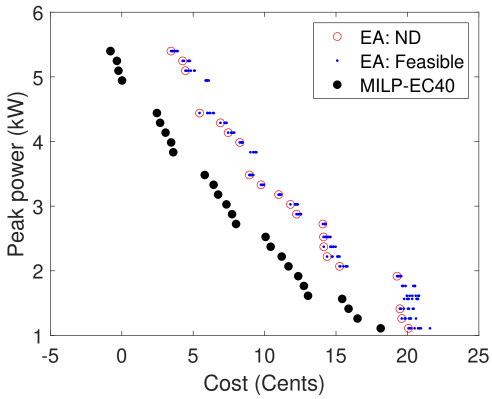


FIGURE 4. PF approximation obtained by the EA with customized initialization and ranking for Instance-1, with MILP-EC40 results shown for reference. Blue dots indicate feasible solutions in the final EA population, with red circles indicating the final ND set.

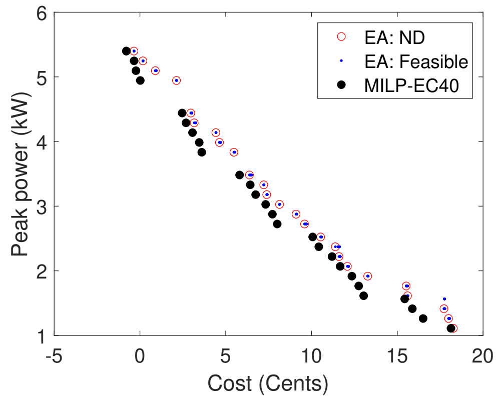


FIGURE 5. PF approximation obtained by the EA with customized initialization, ranking, repair and recombination strategy for Instance-1, with MILP-EC40 results shown for reference. Blue dots indicate feasible solutions in the final EA population, with red circles indicating the final ND set.

in decreasing order, and their corresponding time slots are identified. Then, the power drawn is sorted in the increasing order, and assigned to these time slots. Thus, the lowest power drawn will correspond to the highest cost, and vice versa.

4) Recombination strategy

There is evidence in the literature that restricting the crossover within a neighborhood can improve the convergence especially for cases where the shape of the Pareto set (in the variable space) is complex [33]. In this work, since the peak power levels are discrete, we use it to define the neighborhood of a solution. For the crossover, the pool of parents considered are formed by those that have the neighboring peak power (next higher or lower value relative to the current solution).

Visually, the impact of the above strategies implemented within the EA can be seen from Figure 5. Compared to the previous versions, it can be seen that the convergence and diversity have significantly improved, and the resulting PF approximation is reasonably close to the one obtained using MILP.

VI. NUMERICAL EXPERIMENTS

In this section, we present the results of numerical experiments, which are designed to understand the nature of PF across the instances, as well as to compare the solutions obtained using EA and MILP based approaches.

A. EXPERIMENTAL SET-UP

A set of 21 problem instances is generated in this study, by sampling the problem parameters listed in Table 2. For the EA, a population size of 200 is evolved over 10,000 generations. For generating offspring solutions, a DE crossover is applied with a probability of 1 and scaling factor of 0.7, followed by polynomial mutation with a probability of 0.1 and mutation index of 10. Each individual in the parent population is considered as the base parent sequentially, while the othe two are randomly drawn from the parent pool. Two variants of EA are considered: baseline EA (**EAB**) and improved EA (**EAI**) as discussed in Sections V-A-V-B.

While a reasonable basis for comparison among different EAs assumes equal number of function evaluations, the same cannot be used when comparing EA and MILP. Therefore, multiple different versions of MILP are run, which fall under two categories:

1) Some MILP runs, listed below, are setup as “timed tests”, where the *total* runtime is restricted to that of the proposed EAI, which is approximately 140 seconds. Further, noting that identification of good estimates of the two extremities of the PF are important for both weighted sum and $\epsilon-$constrained formulations, a longer runtime is allowed for finding the extremities, while the rest of the time is divided equally between the remaining subproblems.
   - **MILP-WS**: Refers to the weighted sum formulation with 30 weight values uniformly generated in [0,1]. The MILP is allowed up to 40 s to search for each of the extreme solutions, while the remaining time (out of 140 s) is equally divided to solve the problems corresponding to the intermediate weight values.
   - **MILP-EC**: The same paradigm as above, except $\epsilon-$constraint formulation is used. The number and values of ϵ levels are automatically and efficiently identified within the f_2 bounds as discussed in Section IV-B; thus no additional parameter is required.
2) For others, an extended runtime is allowed to observe the performance corresponding to a longer run. The different versions are listed below:
   - **MILP-WS40**: The weighted sum version, with 40s allowed for solving each subproblem. The overall runtime across instances is observed to be in the range of 700–1000 s.

- **MILP-EC40**: The $\epsilon-$constraint version, with 40 s allowed for solving each subproblem. The overall runtime across instances is observed to be typically in the range of 700–1000 s, although it can get under 300 s for some instances.
- **MILP-EC80**: The $\epsilon-$constraint version, with 80 s allowed for solving each subproblem. The overall runtime across instances is observed to reach up to 1500 s, though some instances can still be solved under 300 s. The results obtained out of this version are considered as the reference set in computing the metrics discussed next.

The EA is implemented in MATLAB 2021, whereas Gurobi 9.12 is used as the MILP solver using the MATLAB API. All experiments are executed on an Intel i7-3770 desktop running Windows 7. A single-thread mode is used for all approaches for a fair comparison in time tests. For quantitative comparisons of the PF approximations, two widely adopted metrics are used, namely, hypervolume (HV) metric [34] and inverted generational distance (IGD) [35]. The reason for choosing these metrics is that both these are composite metrics that assess convergence and diversity of the PF approximation concurrently, and therefore are the most widely used quantitative metrics in the field of evolutionary multi-objective optimization (EMO) for benchmarking [36]. Additionally, IGD is fast to compute, and HV is Pareto compliant, which are desirable properties for benchmarking. A reference set R_F is required for objective-space normalization and computation of the metrics. We use the results MILP-EC80 as R_F, since they were observed to be the best among all compared algorithms. For HV calculations, (1.1,1.1) is used as the reference point in the normalized objective space. Note that due to different number of variables when applying these techniques, as well as different platforms of their implementation, time complexity analysis is not considered as a relevant metric in this study. More extensive analysis regarding scalability of these approaches can be considered in the future work.

B. RESULTS AND DISCUSSION

The comparisons between all algorithms in terms of IGD and HV are shown in Tables 3 and Tables 4, respectively. Due to the stochastic nature of the EAs, the best and median values across 21 runs are shown; whereas for the MILP (deterministic) algorithms, the single run results are shown.

The visualization of the results is organized in two parts. Figure 6 compares EAB, EAI and the time test versions of MILP (MILP-WS, MILP-EC) on all 21 problem instances. Figure 7 compares the lengthier runs of MILP (MILP-WS40, MILP-EC40 and MILP-EC80) with those obtained from EA on all instances. A closer look into the results obtained from the algorithms reveal some interesting aspects of the problem and help understand the relative performance trends of the solution approaches.

Outright, it can be observed from the Figures 6 and 7 that, the problem parameters can impact the shape and convexity of the PF significantly, even though all problems have the same underlying mathematical formulation. The number of distinct segments of the solutions on the PF, as well as their slopes vary significantly across the instances. As an example, Instance-1 (Figure 7(a)) has five distinct segments, Instance-3 (Figure 7(c)) has four segments, whereas Instance-12 (Figure 7(l)) has one segment spanning the whole PF. Further, the disconnected segments can have significantly different slopes (e.g. compare Instance-1,3,8), affecting the convexity of the problem. Thus, the proposed MO extension of the EV charging problem constitutes an interesting and diverse set of benchmarks that can be used to develop efficient solution approaches. It is important during the benchmarking in multi-objective optimization domain to consider a diversity in the nature or PFs; otherwise there is a risk of developing algorithms that are effective only for certain shapes of PFs. This issue is discussed, for example in the context of decomposition-based algorithms in the papers [37], [38]. Typically, the shape of the PF is not known a priori, so it is important that the algorithms perform well on diverse shapes during benchmarking.

The above aspects particularly affect the performance of MILP-WS adversely. Given the non-convex nature of PF for many of the instances, it often obtains only a limited set of solutions on each of the disconnected segments (e.g. see Instance-1, 8 in Figure 6). Moreover, due to the fixed time test setting, the time allocated to the interim weight values is small (only a few 2–5 seconds), which proves insufficient for it to converge sufficiently to the PF. The MILP-WS tends to do better on the PFs with convex shape, such as Instances-7,12 in Figure 6 where it gets relatively better spread across the set of solutions. Considering all instances, the average quality of the PF approximation obtained by MILP-WS is generally poor and lies between EAB and EAI, with some exceptions, discussed shortly.

The preliminary observation regarding baseline EA from the earlier sections on Instance-1 remains true for the remaining instances as well. It achieves poor convergence and diversity for all instances. Of particular note are Instances-3,7,11 where it could not achieve *any* feasible solution at all in the median IGD run. This highlights the challenges faced by EAs in generating solutions through stochastic operators without embedding problem-specific knowledge, given the large search space of 9^{64} possible schedules. Also it is important to note that in all cases, the improved EA provides a much better PF approximation. For the Instances-3,7,11, where EAB could not find feasible solutions, EAI is able to generate a set of solutions close to the PF, though it still struggles to cover the entire range of the PF. One of the reasons why some solutions, particularly in the bottom half of the PF are not identified is because of the heuristic initialization discussed in Section V-B1 which may not *always* be able to generate a feasible solution. If the energy required to fully charge both the vehicles cannot be achieved through the lowest level of charging within the time horizon ($T = 32$), then the resulting solutions do not meet the charging constraint. Thus, while the

TABLE 3. Comparison of IGD obtained by different approaches. A lower value indicates better performance. For EA, the statistics are derived from 21 independent runs.

Instance	EAB(Best)	EAB(Median)	EAI(Best)	EAI(Median)	MILP-WS	MILP-WS40	MILP-EC	MILP-EC40	MILP-EC80
1	0.1650	0.1820	0.0343	0.0415	0.0895	0.0999	0.0128	0.0001	0
2	0.1491	0.1589	0.0415	0.0600	0.0457	0.0456	0.0021	0.0001	0
3	-	-	0.0321	0.1274	0.0533	0.1015	0.0050	0.0002	0
4	0.1660	0.1841	0.0330	0.0368	0.0728	0.0551	0.0016	0.0000	0
5	0.1761	0.1912	0.0460	0.0487	0.0620	0.0563	0.0020	0.0008	0
6	0.2027	0.2195	0.0620	0.0805	0.0738	0.0678	0.0083	0.0000	0
7	0.1781	-	0.1108	0.1485	0.0715	0.0561	0.0023	0.0000	0
8	0.1979	0.2024	0.0277	0.0334	0.0422	0.0417	0.0044	0.0026	0
9	0.1573	0.1766	0.0432	0.0473	0.0642	0.0453	0.0052	0.0000	0
10	0.1528	0.1640	0.0363	0.0400	0.0431	0.0613	0.0008	0.0000	0
11	-	-	0.1465	0.1829	0.0386	0.0356	0.0005	0.0000	0
12	0.1715	0.1815	0.0621	0.0752	0.0999	0.0913	0.0039	0.0004	0
13	0.1655	0.1814	0.0432	0.0566	0.0791	0.0884	0.0000	0.0000	0
14	0.1733	0.1948	0.0320	0.0522	0.0892	0.1366	0.0047	0.0000	0
15	0.1825	0.1964	0.0495	0.0605	0.0680	0.0548	0.0066	0.0010	0
16	0.1563	0.1635	0.0412	0.0692	0.1184	0.0863	0.0031	0.0001	0
17	0.1650	0.1766	0.0256	0.0303	0.0615	0.0485	0.0250	0.0229	0
18	0.1879	0.2232	0.0469	0.0585	0.0385	0.0374	0.0076	0.0007	0
19	0.1691	0.1829	0.0411	0.0453	0.0726	0.0548	0.0077	0.0002	0
20	0.1627	0.1812	0.0327	0.0391	0.0501	0.0521	0.0004	0.0000	0
21	0.1336	0.1419	0.0257	0.0276	0.0472	0.0409	0.0101	0.0000	0

TABLE 4. Comparison of Hypervolume obtained by different approaches. A higher value indicates better performance. For EA, the statistics are derived from 21 independent runs.

Instance	EAB(Best)	EAB(Median)	EAI(Best)	EAI(Median)	MILP-WS	MILP-WS40	MILP-EC	MILP-EC40	MILP-EC80
1	0.5270	0.5098	0.7269	0.7134	0.6676	0.6567	0.7648	0.7751	0.7752
2	0.5845	0.5690	0.7586	0.7315	0.7490	0.7607	0.8089	0.8103	0.8103
3	0.0000	0.0000	0.6573	0.5970	0.6209	0.5832	0.6967	0.6991	0.7001
4	0.5336	0.5118	0.7315	0.7228	0.6773	0.7280	0.7713	0.7724	0.7724
5	0.5223	0.5013	0.7173	0.7064	0.7035	0.7249	0.7667	0.7674	0.7676
6	0.5268	0.5029	0.7546	0.7215	0.7692	0.7871	0.8304	0.8352	0.8352
7	0.4942	0.0000	0.6522	0.6042	0.6567	0.7127	0.7623	0.7646	0.7646
8	0.5313	0.5015	0.7095	0.7018	0.7198	0.7213	0.7449	0.7454	0.7464
9	0.5058	0.4884	0.6815	0.6736	0.6498	0.6827	0.7331	0.7388	0.7388
10	0.5733	0.5540	0.7662	0.7543	0.7479	0.7418	0.8034	0.8039	0.8039
11	0.0000	0.0000	0.5903	0.5629	0.6523	0.6746	0.7126	0.7128	0.7128
12	0.5034	0.4836	0.6793	0.6632	0.6091	0.6330	0.7500	0.7525	0.7527
13	0.5102	0.4891	0.6980	0.6819	0.6344	0.6620	0.7530	0.7530	0.7530
14	0.4638	0.4394	0.6911	0.6707	0.6351	0.5870	0.7259	0.7275	0.7276
15	0.5190	0.4924	0.7099	0.6901	0.6582	0.7095	0.7745	0.7805	0.7806
16	0.5107	0.4943	0.6902	0.6553	0.5916	0.6465	0.7440	0.7467	0.7468
17	0.5237	0.5088	0.7300	0.7207	0.6675	0.7126	0.7437	0.7454	0.7634
18	0.4584	0.4151	0.6132	0.5933	0.6357	0.6478	0.6717	0.6747	0.6748
19	0.5146	0.5006	0.7119	0.7065	0.6576	0.6976	0.7642	0.7720	0.7722
20	0.5187	0.4968	0.6985	0.6879	0.6792	0.6882	0.7392	0.7395	0.7395
21	0.5407	0.5299	0.7151	0.7090	0.6926	0.7103	0.7407	0.7462	0.7462

proposed initialization heuristic works well for most cases, there are occasional problem settings where it may generate feasible solutions only for a subset of peak power levels. For such problems, EAB and EAI will have inferior solutions than MILP-WS. For all the rest of the problems, EAI has significantly better median IGD than MILP-WS, as well as MILP-WS40. Compared to EAB, MILP-WS and MILP-WS40 always achieve better metrics.

Lastly, the performance obtained using MILP-EC is better than MILP-WS, EAB, and EAI. It's comparison with EAI is quite interesting, since they both are given approximately the same time to run. The identification of the extreme solutions are used to establish the ϵ bounds for MILP, hence these two were given a higher runtime (40 s each). This leaves relatively short time to be allotted to the remaining subproblems (2–5 s each). It could be anticipated that the performance of MILP-EC will suffer as a result. However, the results obtained are still better than the median IGD run of EAI. This implies that even when the allotted time was not sufficient to reach optimality, it could still get a good feasible solution for the subproblem. It was observed that the higher runtimes are needed to only find good solutions in the lower half of the PF (i.e., for the low peak power levels). This can be visualized from the comparisons in Figure 6. For Instance-

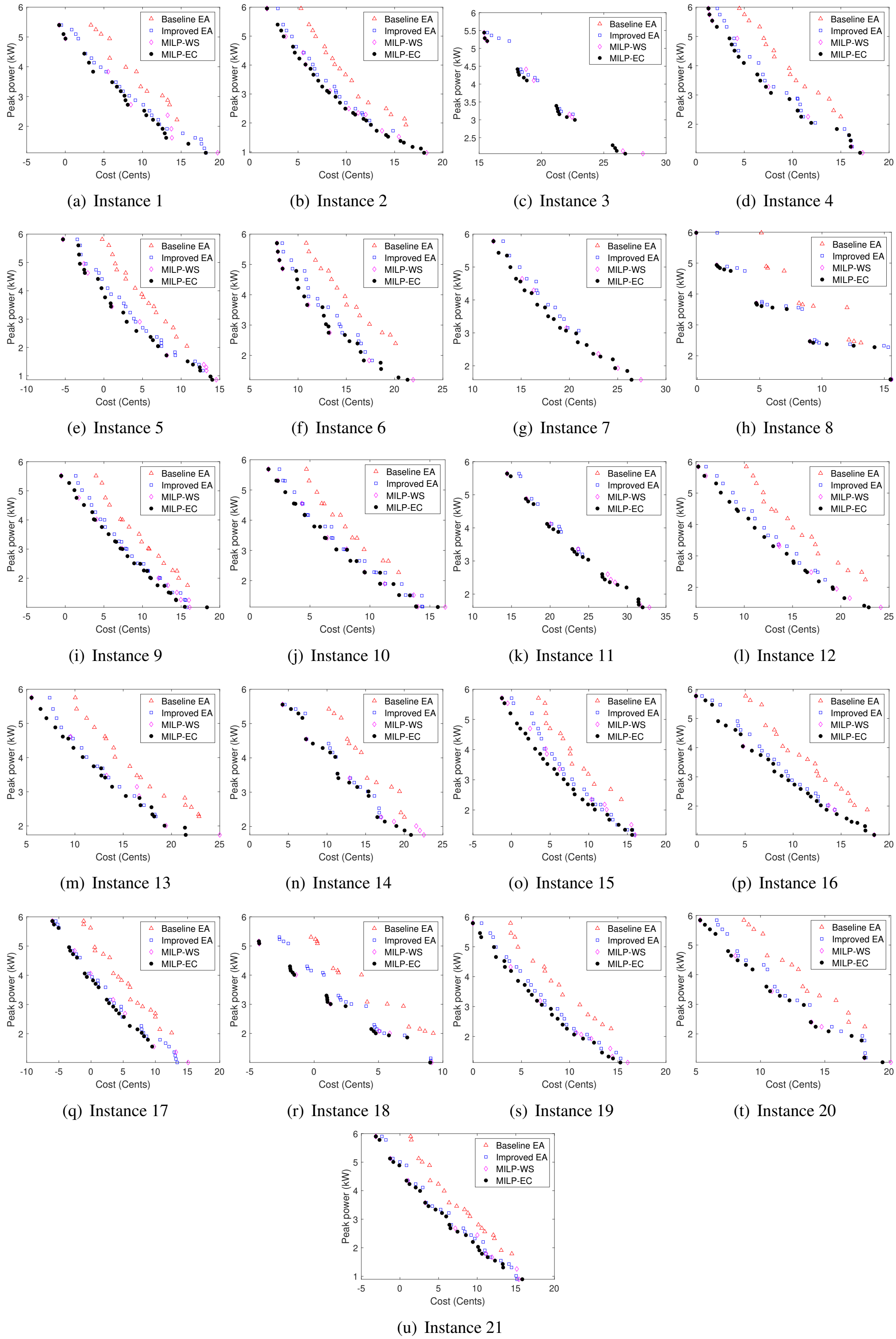
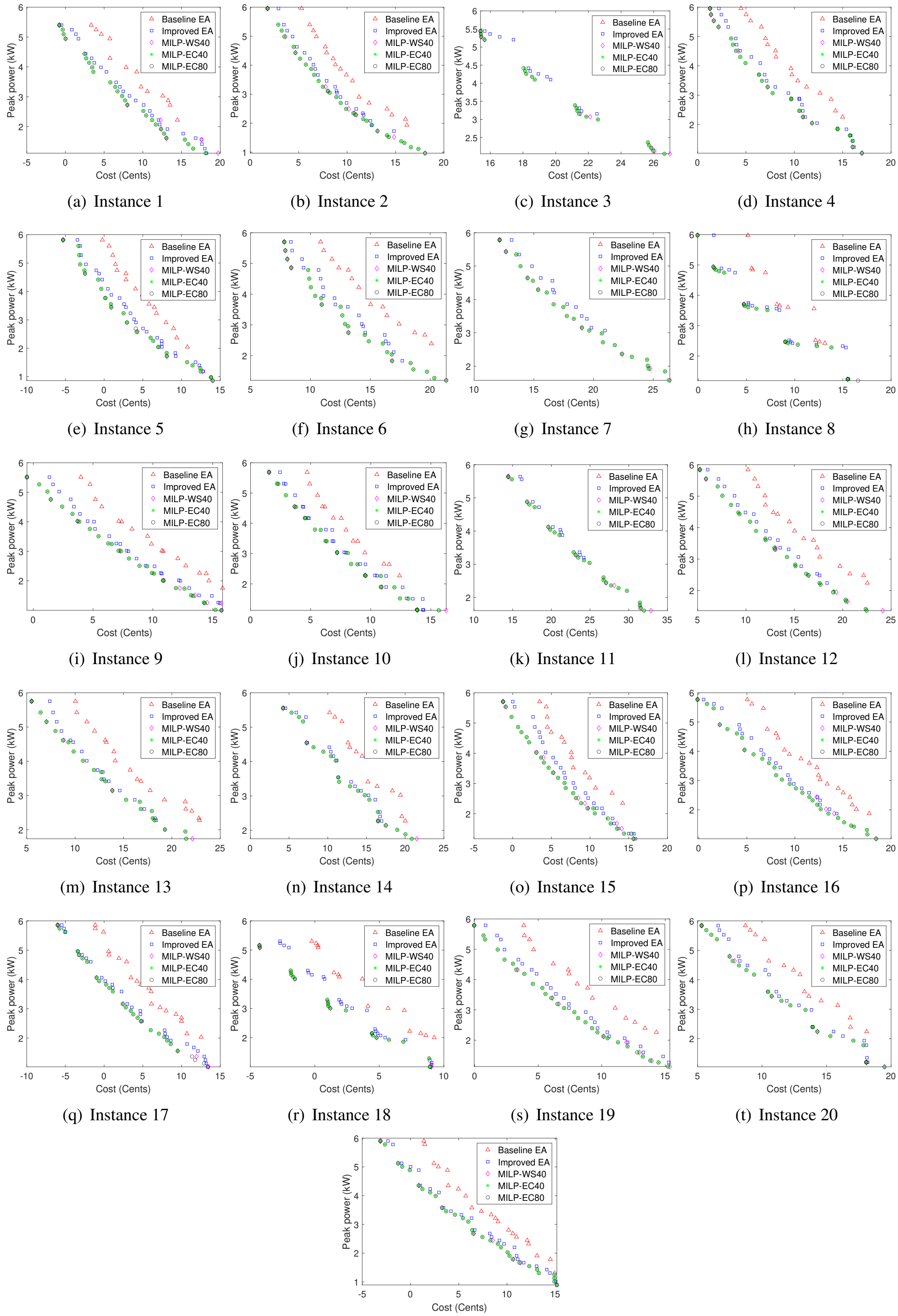


FIGURE 6. Comparison of results obtained for all 21 instances using the Baseline EA, Improved EA, MILP-WS and MILP-EC. For the EA, the median IGD results are shown. For the MILP versions, the total time is restricted to be equal to the EA runtime (140s), with up to 40s allotted to the search of extreme points.



(a) Instance 1 (b) Instance 2 (c) Instance 3 (d) Instance 4

(e) Instance 5 (f) Instance 6 (g) Instance 7 (h) Instance 8

(i) Instance 9 (j) Instance 10 (k) Instance 11 (l) Instance 12

(m) Instance 13 (n) Instance 14 (o) Instance 15 (p) Instance 16

(q) Instance 17 (r) Instance 18 (s) Instance 19 (t) Instance 20

(u) Instance 21

FIGURE 7. Comparison of results obtained for all 21 instances using the Baseline EA, Improved EA, MILP-WS40, MILP-EC40 and MILP-EC80. The MILP-EC80 results shown here are used as reference sets during normalization/computation for IGD and HV. For the EA, the median IGD results are shown.

1, e.g., some of the solutions in the lowest segment were not obtained for the MILP-EC (Figure 6), compared to the results in MILP-EC40 (Figure 7). The Instance-17 (Figure 6(q) shows a very interesting scenario, where EAI was actually able to get a larger range of solutions than the MILP-EC. For the lowermost part, MILP-EC could not obtain a feasible solution. This is in-fact also the case when Instance-17 was solved using MILP-EC40 (Figure 7), implying that even 40s was not sufficient to obtain solutions in this part. The existence of PF beyond the range obtained from MILP-EC is confirmed by looking at the results of MILP-EC80 for Instance-17 (Figure 7). The lengthier versions, i.e., MILP-EC40 and MILP-EC80 produce successively improved results, as intuitively expected.

The visual observations above are also consistent with the numerical metrics that quantify the convergence and diversity (IGD and HV) observed in the Tables 3-4. Overall, the performance can be summarized as EAB < MILP-WS < MILP-WS40 < EAI < MILP-EC < MILP-EC40 < MILP-EC80. Although the EA performance improved through the customizations, it could not outperform the MILP-EC, even for the case of the time test where possibly its chances were the highest. Of course, given the linear nature of the problem, this is not entirely unexpected. However, when multiple solutions need to be found, one premise and motivation of turning to EAs is that the time taken in solving MILP multiple times could be higher than that for a single EA to achieve the same quality of results. Through the algorithm and experiments designed along this line, it was established that even with the above customizations, the performance of EAI was not competitive. There are indeed opportunities for further improvement. Researchers in the EA community should consider this aspect while choosing solution approaches for MO problem, i.e., not directly use a (especially out-of-the-box) EA as a preferred method to solve the MO problem without considering the MILP based alternatives. At the same time, the MILP users should be careful not to use the intuitive weighted sum method as a default for solving MO version, as it is likely to result in poor results. Aside from establishing this through numerical experiments, the main value this study brings to the topic is the generation of interesting MO benchmarks themselves, gaining insights into the challenges involved, and designing the improved solution strategies. We hope that the study will provide a quantitative basis for the researchers for informed algorithm selection, and encourage further development of algorithms to solve the problem more efficiently.

VII. CONCLUSIONS AND FUTURE WORK

In this study we set out with an aim of systematically comparing evolutionary and exact techniques for solving a multi-objective version of an EV charging problem. Towards this end, an existing EV charging problem was extended by generalizing the number of power levels for charging/discharging of each vehicle, as well as including the peak power as a second objective. Thereafter, solution strategies were developed for both MILP and EA approaches to solve the problem. Intermediate results were observed to understand the nature of the MO problem, and embed improvisations in the MILP and EA algorithms leading to their enhanced versions. For MILP, the number of ϵ levels were significantly reduced by utilizing problem-specific information. EA on the other hand was significantly improved from its baseline version by recognizing the nature of the bias in the PF, as well as the downsides of non-dominated sorting for this problem. Numerical experiments were conducted to quantify the performance of the algorithms across multiple instances of the problem. The analysis of the results improved the understanding of the nature of the problems, as well as how they impacted the relative performance of the solution strategies. In the subsequent work, the scalability of the optimization methods to higher number of vehicles will be investigated. The proposed operators could be further improved by, e.g., using ranking strategies that promote marginally infeasible solutions [39]. As discussed in Section II, the potential of ML techniques and their combinations with optimization methods could be also explored. In the current context, for example, the profiles of charging and discharging prices are generated numerically. In a real-world scenario, these profiles might need to predicted using techniques such as ML, since dynamic pricing might be in place. The use of ML could also be used to pose constraints to the problem that would eliminate parts of the search space that are predicted to outright be sub-optimal. Lastly, dynamic version of the scheduling problem, as well as other hybrid solution approaches also form interesting topics for further research.

REFERENCES

[1] Climate Council Australia, "Transport emissions: Driving down car pollution in cities," https://www.climatecouncil.org.au/wp-content/uploads/2017/09/FactSheet-Transport.pdf, (retrieved 27/01/22), 2017.
[2] United States EPA, "Fast facts on transportation greenhouse gas emissions," https://www.epa.gov/greenvehicles/fast-facts-transportation-greenhouse-gas-emissions, (retrieved 27/01/22), 2019.
[3] D. Espín-Sarzosa, R. Palma-Behnke, and O. Núñez-Mata, "Energy management systems for microgrids: Main existing trends in centralized control architectures," *Energies*, vol. 13, no. 3, p. 547, 2020.
[4] Z.-J. M. Shen, B. Feng, C. Mao, and L. Ran, "Optimization models for electric vehicle service operations: A literature review," *Transportation Research Part B: Methodological*, vol. 128, pp. 462–477, 2019.
[5] J. Hu, H. Morais, T. Sousa, and M. Lind, "Electric vehicle fleet management in smart grids: A review of services, optimization and control aspects," *Renewable and Sustainable Energy Reviews*, vol. 56, pp. 1207–1226, 2016.
[6] D. H. Wolpert and W. G. Macready, "No free lunch theorems for optimization," *IEEE transactions on evolutionary computation*, vol. 1, no. 1, pp. 67–82, 1997.
[7] O. Sassi and A. Oulamara, "Electric vehicle scheduling and optimal charging problem: complexity, exact and heuristic approaches," *International Journal of Production Research*, vol. 55, no. 2, pp. 519–535, 2017.
[8] T. Ishihara and S. Limmer, "Optimizing the hyperparameters of a mixed integer linear programming solver to speed up electric vehicle charging control," in *EvoAPPS (EvoStar)*. Springer, 2020, pp. 37–53.
[9] D. Orazgaliyev, A. Tleubayev, B. Zholdaskhan, H. K. Nunna, A. Dadlani, and S. Doolla, "Adaptive coordination mechanism of overcurrent relays using evolutionary optimization algorithms for distribution systems with

dgs," in *2019 International Conference on Smart Energy Systems and Technologies (SEST)*. IEEE, 2019, pp. 1–6.
[10] H. K. Nunna, S. Battula, S. Doolla, and D. Srinivasan, "Energy management in smart distribution systems with vehicle-to-grid integrated microgrids," *IEEE transactions on smart grid*, vol. 9, no. 5, pp. 4004–4016, 2016.
[11] Y. Cao and Y. Wang, "Robust charging schedule for autonomous electric vehicles with uncertain covariates," *IEEE Access*, vol. 9, pp. 161 565–161 575, 2021.
[12] Q. Wang, X. Liu, J. Du, and F. Kong, "Smart charging for electric vehicles: A survey from the algorithmic perspective," *IEEE Communications Surveys Tutorials*, vol. 18, no. 2, pp. 1500–1517, 2016.
[13] M. Yilmaz and P. T. Krein, "Review of benefits and challenges of vehicle-to-grid technology," in *2012 IEEE Energy Conversion Congress and Exposition (ECCE)*, 2012, pp. 3082–3089.
[14] R. Mehta, D. Srinivasan, and A. Trivedi, "Optimal charging scheduling of plug-in electric vehicles for maximizing penetration within a workplace car park," in *IEEE Congress on Evolutionary Computation (CEC)*, 2016, pp. 3646–3653.
[15] M. Elmehdi and M. Abdelilah, "Genetic algorithm for optimal charge scheduling of electric vehicle fleet," in *ICPS/NISS*, 2019, pp. 1–7.
[16] T. Mao, X. Zhang, and B. Zhou, "Intelligent energy management algorithms for EV-charging scheduling with consideration of multiple EV charging modes," *Energies*, vol. 12, no. 2, 2019.
[17] E. Kontou, Y. Yin, and Y.-E. Ge, "Cost-effective and ecofriendly plug-in hybrid electric vehicle charging management," *Transportation Research Record*, vol. 2628, no. 1, pp. 87–98, 2017.
[18] C. Jin, J. Tang, and P. Ghosh, "Optimizing electric vehicle charging with energy storage in the electricity market," *IEEE Transactions on Smart Grid*, vol. 4, no. 1, pp. 311–320, 2013.
[19] S. Limmer and T. Rodemann, "Peak load reduction through dynamic pricing for electric vehicle charging," *International Journal of Electrical Power & Energy Systems*, vol. 113, pp. 117–128, 2019.
[20] E. Zitzler, "Evolutionary algorithms for multiobjective optimization: Methods and applications," Ph.D. dissertation, Swiss Federal Institute of Technology Zurich, 1999.
[21] R. Das, Y. Wang, G. Putrus, R. Kotter, M. Marzband, B. Herteleer, and J. Warmerdam, "Multi-objective techno-economic-environmental optimisation of electric vehicle for energy services," *Applied Energy*, vol. 257, p. 113965, 2020.
[22] R. Das, Y. Wang, K. Busawon, G. Putrus, and M. Neaimeh, "Real-time multi-objective optimisation for electric vehicle charging management," *Journal of Cleaner Production*, vol. 292, p. 126066, 2021.
[23] A. Zakariazadeh, S. Jadid, and P. Siano, "Multi-objective scheduling of electric vehicles in smart distribution system," *Energy Conversion and Management*, vol. 79, pp. 43–53, 2014.
[24] H. Ahmadi-Nezamabad, M. Zand, A. Alizadeh, M. Vosoogh, and S. Nojavan, "Multi-objective optimization based robust scheduling of electric vehicles aggregator," *Sustainable Cities and Society*, vol. 47, p. 101494, 2019.
[25] S. Shahriar, A.-R. Al-Ali, A. H. Osman, S. Dhou, and M. Nijim, "Machine learning approaches for ev charging behavior: A review," *IEEE Access*, vol. 8, pp. 168 980–168 993, 2020.
[26] I. Kalysh, M. Kenzhina, N. Kaiyrbekov, H. K. Nunna, A. Dadlani, and S. Doolla, "Machine learning-based service restoration scheme for smart distribution systems with dgs and high priority loads," in *2019 International Conference on Smart Energy Systems and Technologies (SEST)*. IEEE, 2019, pp. 1–6.
[27] H. M. Abdullah, A. Gastli, and L. Ben-Brahim, "Reinforcement learning based ev charging management systems–a review," *IEEE Access*, vol. 9, pp. 41 506–41 531, 2021.
[28] R. Kasimbeyli, Z. K. Ozturk, N. Kasimbeyli, G. D. Yalcin, and B. I. Erdem, "Comparison of some scalarization methods in multiobjective optimization," *Bulletin of the Malaysian Mathematical Sciences Society*, vol. 42, no. 5, pp. 1875–1905, 2019.
[29] S. Gass and T. Saaty, "The computational algorithm for the parametric objective function," *Naval research logistics quarterly*, vol. 2, no. 1-2, pp. 39–45, 1955.
[30] Y. Haimes, "On a bicriterion formulation of the problems of integrated system identification and system optimization," *IEEE transactions on systems, man, and cybernetics*, vol. 1, no. 3, pp. 296–297, 1971.
[31] R. Storn and K. Price, "Differential evolution – A simple and efficient heuristic for global optimization over continuous spaces," *Journal of global optimization*, vol. 11, no. 4, pp. 341–359, 1997.
[32] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan, "A fast and elitist multi-objective genetic algorithm: NSGA-II," *IEEE transactions on evolutionary computation*, vol. 6, no. 2, pp. 182–197, 2002.
[33] H. Li and Q. Zhang, "Multiobjective optimization problems with complicated pareto sets, MOEA/D and NSGA-II," *IEEE transactions on evolutionary computation*, vol. 13, no. 2, pp. 284–302, 2008.
[34] K. Shang, H. Ishibuchi, L. He, and L. M. Pang, "A survey on the hypervolume indicator in evolutionary multiobjective optimization," *IEEE Transactions on Evolutionary Computation*, vol. 25, no. 1, pp. 1–20, 2020.
[35] C. A. C. Coello and M. R. Sierra, "A study of the parallelization of a coevolutionary multi-objective evolutionary algorithm," in *Mexican international conference on artificial intelligence (MICAI)*. Springer, 2004, pp. 688–697.
[36] H. Ishibuchi, R. Imada, N. Masuyama, and Y. Nojima, "Comparison of hypervolume, igd and igd+ from the viewpoint of optimal distributions of solutions," in *International conference on evolutionary multi-criterion optimization*. Springer, 2019, pp. 332–345.
[37] M. Asafuddoula, H. K. Singh, and T. Ray, "An enhanced decomposition-based evolutionary algorithm with adaptive reference vectors," *IEEE transactions on cybernetics*, vol. 48, no. 8, pp. 2321–2334, 2017.
[38] H. Ishibuchi, Y. Setoguchi, H. Masuda, and Y. Nojima, "Performance of decomposition-based many-objective algorithms strongly depends on pareto front shapes," *IEEE Transactions on Evolutionary Computation*, vol. 21, no. 2, pp. 169–190, 2016.
[39] T. Ray, H. K. Singh, A. Isaacs, and W. Smith, "Infeasibility driven evolutionary algorithm for constrained optimization," in *Constraint-handling in evolutionary optimization*. Springer, 2009, pp. 145–165.

HEMANT KUMAR SINGH completed his PhD in 2011 from The University of New South Wales (UNSW), Australia and Bachelor of Technology in 2007 from Indian Institute of Technology, Kanpur, India. He is currently a Senior Lecturer at UNSW Canberra. His research interest lies in evolutionary computation methods for optimization and decision-making. Within these domains, he has investigated various aspects, such as constraint handling, multiple conflicting criteria, robustness, and hierarchical objectives. More details of his research and professional contributions can be found on his webpage http://www.mdolab.net/Hemant.

TAPABRATA RAY received his Ph.D. from the Indian Institute of Technology, Kharagpur, India, in 1997. He is a Professor with The University of New South Wales, Australia, where he leads the Multidisciplinary Design Optimization Group. His current research interests include surrogate assisted/multifidelity optimization, multi/many-objective optimization, bilevel optimization, robust design and decision-making.

MD JUEL RANA received B.Sc. degree in electrical and electronic engineering from the Rajshahi University of Engineering and Technology (RUET), Rajshahi, Bangladesh, in 2011, and M.Sc. degree in electrical engineering from the King Fahd University of Petroleum and Minerals (KFUPM), Saudi Arabia, in 2017. He is currently pursuing Ph.D. with the School of Engineering and Information Technology, University of New South Wales (UNSW) Canberra, Australia. His current research interests include power system oper- ation and control, energy management of microgrid, electricity market, evolutionary algorithms, multiobjective optimization, and game theory.

STEFFEN LIMMER received the M.Sc. (Diploma) in computer science from the University of Jena, Germany in 2009. In 2016, he received the Ph.D. degree in engineering from the University of Erlangen-Nürnberg, Germany, where he worked from 2009 to 2016 as scientific assistant at the chair of computer architecture. Since December 2016, Dr. Limmer is senior scientist at the Honda Research Institute Europe. His current research topics are optimization and data-driven modeling in the context of energy management systems.

TOBIAS RODEMANN received his Diploma (MSc) in Neuroinformatics from the University of Bochum (Germany) in 1998. In the same year he joined Honda R&D Europe and in 2003 he got a PhD degree in Computational Neuroscience from the University of Bielefeld. He is currently a chief scientist at the Honda Research Institute Europe, working on cooperative intelligence for energy and mobility applications.

MARKUS OLHOFER is Chief Scientist at the Honda Research Institute Europe GmbH since 2010 and responsible for the Complex Systems Optimisation and Analysis Group of the institute. Markus Olhofer graduated in Electrical Engineering with a Diplom Ingenieur degree in 1997. In 2000 he received the Dr.-Ing. degree from the Institute for Neural Computation from Ruhr-Universität Bochum, Germany. He joined the Future Technology Research Division at Honda R&D Europe (Deutschland) GmbH in 1998 and since 2001 he works at the Honda Research Institute Europe GmbH.