

Design of an Autonomous Intelligent Demand-Side Management System Using Stochastic Optimisation Evolutionary Algorithms

Edgar Galván-López^a, Tom Curran^b, James McDermott^c, Paula Carroll^c

^aTAO Project, INRIA Saclay & LRI - Univ. Paris-Sud and CNRS, Orsay, France

^bSchool of Computer Science & Statistics, Trinity College Dublin

^cManagement Information Systems, Lochlann Quinn School of Business, University College Dublin

Abstract

Demand-Side Management systems aim to modulate energy consumption at the customer side of the meter using price incentives. Current incentive schemes allow consumers to reduce their costs, and from the point of view of the supplier play a role in load balancing, but do not lead to optimal demand patterns. In the context of charging fleets of electric vehicles, we propose a centralised method for setting overnight charging schedules. This method uses evolutionary algorithms to automatically search for optimal plans, representing both the charging schedule and the energy drawn from the grid at each time-step. In successive experiments, we optimise for increased state of charge, reduced peak demand, and reduced consumer costs. In simulations, the centralised method achieves improvements in performance relative to simple models of non-centralised consumer behaviour.

Keywords: Demand-Side Management Systems, Evolutionary Algorithms, Genetic Algorithms, Evolution Strategies, Electric Vehicles, State of Charge, Peak-to-average ratio, Electricity Costs, Smart grid time-of-use pricing

1. Introduction

EU policy aims to reduce greenhouse gas emissions and reduce dependency on imported fossil fuels. The “20-20-20” targets [1] mandate the reduction in member states to 20% below the 1990 emission levels, the supply of 20% of all energy from renewable energy sources (RESs) and a reduction in energy consumption by 20% by the year 2020.

Electric vehicles (EVs) are viewed as playing a role in reducing emissions in the transport sector, but their usage causes an increase in electricity demand. The use of RESs also causes problems for the efficient operation of a power plant [2]. Increased cycling (starting up and shutting down) of the power system results in increased wear and tear on the plant and can cause an increase of greenhouse gas emissions.

Therefore, new methods are required to increase electricity grid efficiency and reduce emissions. The smart grid (SG) is one main approach. A SG is a type of electrical power grid whose goal is to respond to the behaviour and actions of energy suppliers and consumers to efficiently deliver economic, reliable and sustainable electricity services. Multiple research areas have been explored in SG over recent years as a result of different challenges that have been posed to the electrical grid. One of the most explored areas in SG is Demand-Side Management (DSM) systems as shown by the increasing number of publications, ranging from the use of intelligent algorithms (e.g., game theory [3],

Email addresses: edgar.galvan@inria.fr (Edgar Galván-López), tcurran@tcd.ie (Tom Curran), jmmcd@jmmcd.net (James McDermott), paula.carroll@ucd.ie (Paula Carroll)

Monte Carlo-based methods [4], evolutionary algorithms [5], multi-agent systems [6]), real-time systems [7], up to challenges and in-depth surveys on the area [8, 9, 10, 11].

DSM is a set of measures to improve the energy system at the consumer side. DSM ranges from improving energy efficiency through the use of better insulation or better materials up to the use of autonomous systems to control energy resources [10]. DSM programs include different approaches, e.g. manual conservation and energy efficiency programs [12] and Residential Load Management (RLM) [6, 3]). RLM programs based on smart pricing are amongst the most popular methods.

The motivation for smart grid tariff structures is twofold. They allow consumers to reduce their electricity costs. At the same time, the utility company achieves a reduction in the peak-to-average ratio (PAR) in load demand resulting from the shifted consumption [6]. If no special measures are taken to avoid them, high PAR values come about naturally because consumer electricity demand follows a diurnal pattern, with increased load in the morning, a dip in the afternoon, a rise in the evening, and a stronger dip in the middle of the night.

Some of these smart pricing methods are very popular. In particular, time-of-use (ToU) pricing has been widely adopted in some European countries [11]. Other smart pricing types include critical-peak pricing, extreme day pricing, and smart grid real-time pricing, to mention a few.

Motivated by smart price-based approaches, we are interested in developing an *autonomous intelligent* DSM system that shifts electricity consumption of electric vehicles (EVs). To this end, we use stochastic optimisation evolutionary algorithms (EAs). The main contribution of this work focuses on the notion of load shifting, borrowed from popular smart pricing-based methods. In contrast to typical DSM approaches such as dynamic pricing, which are based on an interaction between the utility and the user, we use a centralised approach, wherein the consumption schedule is set centrally based on complete information of all EVs. The motivation is to achieve improvements in performance. To do so, we use EAs to *automatically* generate (optimal) solutions. The use of all EVs is considered in the solution representation used in our EAs (described in detail in Section 2). We also use this in the evaluation of candidate solutions.

To test this idea, we considered a dynamic scenario of 28 simulated days, with the charging period from 18:00 to 07:30, divided into 28 time-slots of 30 minutes each. An action (switching EV charging on or off) can be taken at the beginning of each time-slot. We defined three different goals:

- (a) that EVs' batteries are as fully charged as possible;
- (b) we add an extra goal to (a) by aiming for a low fluctuation at the transformer load (i.e., low PAR); and finally,
- (c) we add a third and final goal that aims to reduce electricity costs to the consumer by using a pricing signal based on ToU.

To achieve these three goals, we propose three fitness functions. Each will be used independently in our EA and will guide our evolutionary search to automatically create an (optimal) plan.

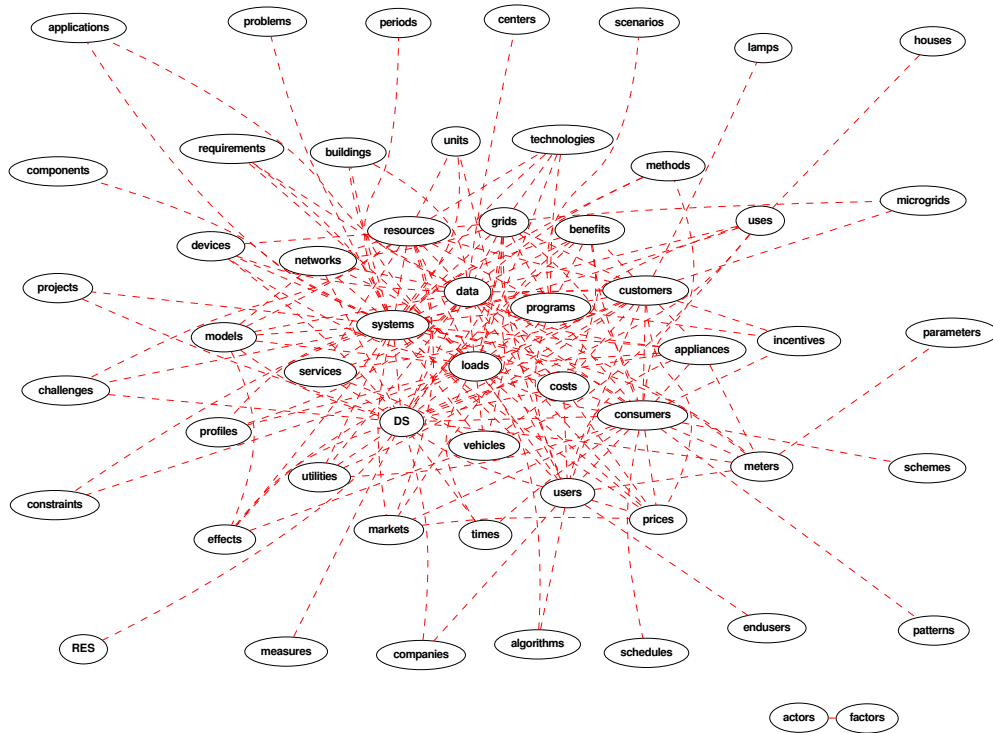
The core elements in this work are:

1. We study the impact of the representation and functions proposed in this work when scaling the problem up (i.e., from using a few EVs to using dozens of them) by measuring the transformer load, the initial and final state of charge (SoC), the PAR and electricity costs.
2. To do so, we used two EA approaches: a genetic algorithm and an evolution strategy and compared their performance against three non-intelligent approaches (i.e., Greedy-, Midnight- and Random method), each of them simulating a specific user behaviour.
3. A dynamic scenario was used to study all these approaches by allowing having a variety of changes, i.e. different SoC for each EV for each of the simulated days, over a 28-day simulated period.

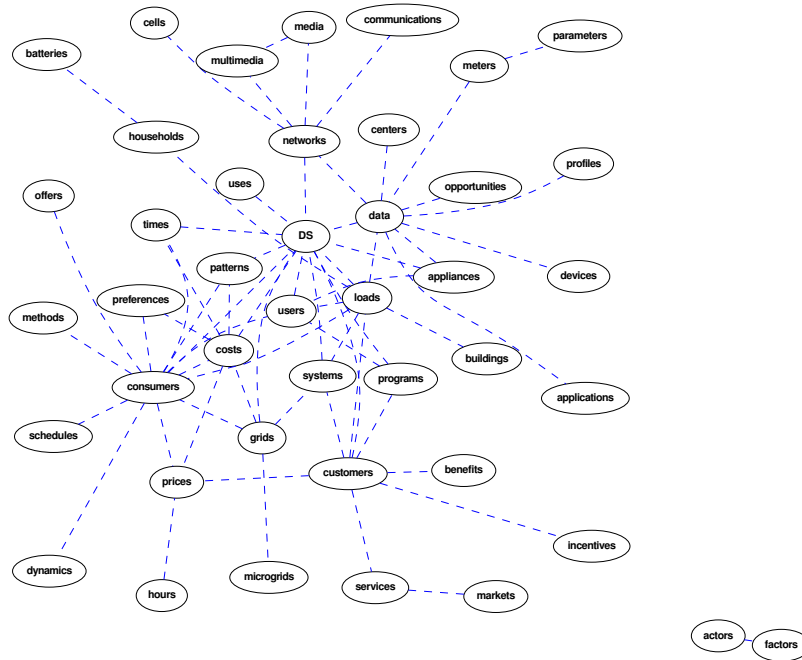
1.1. Importance of this Research in DSM

DSM has been investigated extensively over recent years. For instance, it has been shown that more than 2,000 scientific papers have been published in this area since the 1980s [4], with more than half in this decade. Figure 1 shows a visual representation of the research trends followed in DSM (a) from 2010 until now, and (b) in 2014 only¹.

¹Source: <http://ieeexplore.ieee.org/Xplore/home.jsp>. Last accessed date: 31/08/2014. Links of strength less than 55 (in (a)) or 20 (in (b)) are filtered out. Details on how this was produced can be found in [13].



(a) Analysis from 2010 - 2014



(b) Analysis 2014 only

Figure 1. Research trend on ‘Demand-side management’ systems (a) 2010 until 2014, and, (b) 2014 only. Only links (similarities) with strength greater than 55 (in (a)) or 20 (in (b)) are shown.

As can be seen in Figure 1 multiple topics have been covered in DSM, ranging from electricity costs, the use of electric vehicles, up to the use of data. The research conducted in this work lies at the very core of the research trend observed in this figure.

The challenges continuously presented to the grid, such as the aggregation of new electric devices (e.g., electric vehicles using the grid can double the average household load [3]) make the use of intelligent algorithms suitable to be used in the design/implementation of DSM systems. Figure 1 shows this trend. For instance, notice the presence of “algorithms”, “programs”, and “methods”. In fact, one could consider the presence of “algorithms” in the core of Figure 1 if researchers had unified their use around this unique term instead of various synonyms.

Several algorithms have been used in DSM system and each has focused attention on different areas within DSM. For example, it has been shown that by adopting pricing tariffs which differentiate energy usage by time and level, a global optimal performance can be achieved by means of a Nash equilibrium of the formulated consumption scheduling game [3]. Multi-agent systems have also been used in DSM. For instance, the research conducted in [14] aimed to create a DSM based on these type of systems and studied different types of smart pricing, concluding that in all studied scenarios, a high PAR was observed under the use of these smart pricing models (e.g., ToU, critical peak price, real-time pricing).

In this work we use EAs to automatically create plans to intelligently charge EVs’ batteries, aiming at reducing PAR, reducing load at the substation transformer, and reducing costs to the consumer.

This paper is organised as follows. In the following section we introduce our proposed approach. Section 3 shows the experimental setup used in this study. In Section 4, we present and discuss our findings. Section 5 draws some conclusions and presents some future work.

2. Proposed Approach

2.1. Background

Evolutionary Algorithms (EAs) [15, 16], also known as Evolutionary Computation systems, are influenced by the theory of evolution by natural selection. These algorithms have been with us for some decades and are very popular, perhaps, due to their successful application in a range of different problems, ranging from the automated design of an antenna carried out by NASA [17], the automated optimisation of game controllers [18, 19], the automated design of combinational logic circuits [20, 21], to automated optimal localisation for building seismic sensing stations [22]. EAs are “black-box”, that is, they do not require any specific knowledge of the fitness function. They work even when, for example, it is not possible to define a gradient on the fitness function or to decompose the fitness function into a sum of per-variable objective functions. The fitness functions used in our work (described in Section 2.2) are not amenable to analytic solution or simple gradient-based optimisation, hence search algorithms such as EAs are required.

The idea behind EAs is to automatically generate (nearly) optimal solutions by “evolving” potential solutions (individuals forming a population) over time (generations) by using bio-inspired operators (e.g., crossover, mutation). More specifically, the evolutionary process includes the initialisation of the population $P(0)$ at generation $g = 0$. The population consists of a number of individuals which represent potential solutions to the particular problem. At each iteration or generation (g), every individual within the population ($P(g)$) is evaluated using a *fitness function* that determines its fitness (i.e., how good or bad an individual is). Then, a selection mechanism takes place to stochastically pick the fittest individuals from the population. Some of the selected individuals are modified by genetic operators and the new population $P(g + 1)$ at generation $g + 1$ is created. The process stops when some halting condition is satisfied. Further details on how these stochastic optimisation algorithms work can be found in [15, 16].

2.2. Representation and Evaluation

In this work, we use a fixed-length bitstring representation, where each bit indicates whether a particular EV should be charged or not during a particular time period.

Let N denote the number of household units (users). Let M denote the number of electric vehicles (EVs) available in N . In our study $M = N$. In our work the time-slots are of length 30 minutes, starting at t_i and running to t_f .

Therefore an energy consumption schedule can be naturally represented by a matrix of bits:

$$E_M \triangleq \begin{bmatrix} E_1^{t_i}, \dots, E_1^{t_f} \\ E_2^{t_i}, \dots, E_2^{t_f} \\ \vdots \\ E_M^{t_i}, \dots, E_M^{t_f} \end{bmatrix} \quad (1)$$

where each E_m^t is a single bit representing whether EV m is charging at time t or not. Each row represents the behaviour of a single EV over the full period; each column represents the behaviour of all EVs at a single time-slot. An individual in the EA is just a matrix E_M , unrolled to give a bitstring, that is:

$$E_1^{t_i}, \dots, E_1^{t_f}, E_2^{t_i}, \dots, E_2^{t_f}, \dots, E_M^{t_i}, \dots, E_M^{t_f} \quad (2)$$

Now, we need to define a fitness function (cost function) to automatically evaluate the candidate solution shown in Equation 2. To do so, we need to define several elements, discussed next.

For each $m \in \{1, \dots, M\}$, let l_m^t denote the load drawn by EV m at time t . If $E_m^t = 0$ then $l_m^t = 0$; if $E_m^t = 1$ then $l_m^t = 1.7\text{kW}$ (this constant is characteristic of the EV). The total load across all EVs at each time $t \in \{t_i, \dots, t_f\}$ is

$$L_t \triangleq \sum_{m \in \{1, \dots, M\}} l_m^t \quad (3)$$

As indicated previously, we are interested in the first instance in *automatically* creating (near-)optimal schedules so that the final state of charge SoC_{t_f} for all $m \in \{1, \dots, M\}$ is as high as possible (goal (a) as described in Section 1). Let us call this the ‘Charging’ function, f_c . Thus, we aim to **maximise**

$$f_c(E_M) \triangleq \frac{1}{M} \sum_{m \in \{1, \dots, M\}} \frac{\text{SoC}_{t_f}(E_m)}{\text{SoC}_{\max_m}} \quad (4)$$

where we denote the maximum possible charge as SoC_{\max_m} . Equation 4 guides our EA towards a solution that aims to charge each EV as much as possible.

We now proceed to consider goal (b), which is a low fluctuation in the total EV load over time. The peak-to-average ratio (PAR) is calculated as the maximum load demand for a period of time divided by the average load demand. Therefore, the PAR in demand is obtained by using Equation 3 as a basis, and so we have

$$\text{PAR} \triangleq \frac{\max_{t \in \{t_i, \dots, t_f\}} L_t}{\frac{1}{T} \sum_{t \in \{t_i, \dots, t_f\}} L_t} \quad (5)$$

To define a suitable fitness function, we start by defining the variance of the total load (Equation 3) over time. We take the negative (and add a scaling constant) so that we aim to **maximise**:

$$S(E_M) \triangleq -\frac{1}{C} \sigma[L_{t_i}, \dots, L_{t_f}] + \frac{1}{C} \quad (6)$$

where $\sigma[L_{t_i}, \dots, L_{t_f}]$ is the variance in the total load over time, and $C \approx 80\%M$. Equation 6 calculates the constancy of load. Now, we are in position to define a ‘Steady Charging’ function:

$$f_s(E_M) \triangleq \begin{cases} f_c(E_M) & \text{if } \exists m : \text{SoC}(E_m) < \text{SoC}_{\min}, \\ f_c(E_M) + S(E_M) & \text{otherwise.} \end{cases} \quad (7)$$

where SoC_{\min} is the minimum state of charge that an EV should achieve at time t_f , and in this work we have arbitrarily chosen $\text{SoC}_{\min} = 80\%$ of capacity. Equation 7 aims first at reaching this minimum SoC for all EVs and then tries to achieve constancy of EV load at the transformer. This in consequence translates into having both a constancy and a low PAR (Equation 5).

We finally consider our final goal (c): reduction of electricity costs. So, we define our third and last function, to be called ‘Price-Based Charging’ (f_p). This function works in three stages. It first tries to charge an EV to a certain minimum SoC, SoC_{\min} . Once this is achieved, it tries to reduce costs, by taking advantage of cheaper electricity at certain times (details are given in Section 3). It then tries to reduce variance at the transformer load (Equation 6). The reduction of electricity cost is defined by

$$R(E_M) \triangleq \frac{C}{M} \sum_{m \in M} \left(1 - \frac{P_m(E_M)}{P_{\max_m} - P_{\min_m}} \right) \quad (8)$$

where P_m , P_{\max_m} , and P_{\min_m} indicate the price of a given scheduling E_m for a single EV m , the highest possible price for that EV, and the optimal price for it, respectively.

Since $R(E_M)$ is a *reduction*, higher values are better. We will denote the minimum desired price reduction as R_{\min} . In this work we use the value $R_{\min} = 0.15$, chosen empirically. Other values are also possible. By using the expressions in (4), (6) and (8) we are now in a position to define our ‘Price-Based Charging’ function f_p . Thus, we have

$$f_p(E_M) \triangleq \begin{cases} f_c(E_M) & \text{if } \exists m : \text{SoC}(E_m) < \text{SoC}_{\min}, \\ f_c(E_M) + R(E_M) & \text{if } \forall m : \text{SoC}(E_m) \geq \text{SoC}_{\min} \text{ and } R(E_M) < R_{\min}, \\ f_c(E_M) + R(E_M) + S(E_M) & \text{otherwise.} \end{cases} \quad (9)$$

Next, we describe the experimental setup used in this work to test the three proposed functions: Charging (f_c), Steady Charging (f_s) and Price-Based Charging (f_p).

3. Experimental Setup

3.1. Grid Scenario

We assume that EVs are charged only at home, and between the hours of $t_i = 18:00$ and $t_f = 07:30$, inclusive. This charging period is divided into 30-minute slots. This scenario is common in previous research [5, 23, 24, 25]. There are several reasons why a 30-minute slot is used: sometimes electricity costs are recorded every 30 minutes [25]; another reason is to reduce the decision space in the context of reinforcement learning [6]. In this study, we decided to use 30-minute slots in order to have candidate solutions of reasonable size (see Equation 1), which results in a smaller search space. Our representation allows for charging to switch on or off every 30 minutes, but modern EV batteries are not as susceptible to damage from switching charging on and off frequently [26].

In our considered benchmark smart grid system there are $N = \{10, 30, 60, 90\}$ users, each with one EV (hence $N = M$). Each EV $m \in \{1, \dots, M\}$ can only be charged at home. These small numbers of EVs are not unrealistic: in real-world electricity markets, a Demand Aggregator is a market participant which plays the role of a middleman between the consumer and the supplier (see e.g. [27]). In our scenario, a Demand Aggregator would be responsible for the centralised control of a (potentially small) number of EV charging schedules.

We simulated a dynamic scenario, where the initial state of charge (SoC) for each EV at time of arrival (t_i) at home varies for each of the 28 simulated days.

To calculate the electricity costs yielded by the fitness functions proposed in Section 2, we use the smart pricing time-of-use ‘Standard Electricity NightSaver’ plan implemented by electric Ireland² on the summertime schedule. The peak time is from 09:00 until midnight, whereas the remaining time is off-peak time (Table 1 shows the electricity costs for these two periods). In our scenario, this translates as having a high-peak time from 18:00 until 23:30, whereas the off-peak time is from midnight until 07:30.

Table 1 summarises the parameters used to simulate our grid scenario. We ran our simulations using the GridLab-D simulator [28] (version 2.3). This is an open source electrical grid simulator developed by the US Department of Energy.

²Source: <https://www.electricireland.ie/switchchange/detailsValueRewardNightSaver.htm>

Parameter	Value
Number of EVs (N)	{10, 30, 60, 90}
Initial time and latest time to charge	$t_i = 18:00$, $t_f = 07:30$
Frequency of making a decision	30 minutes
Number of time slots T	28
State of Charge (SoC) at t_i (%)	Uniform in [40, 60]
Battery capacity (kW)	Random from {24, 60}
Plug-in charging (kW)	1.7
Time-of-use pricing (kWh)	€0.2062 (peak) €0.1019 (off peak)

Table 1. Summary of parameters used for our smart grid system.

Parameter	Value
Length of the chromosome	NT (see Table 1)
Population size	100
Generations	200
Crossover rate	0.5 (Only for the EA)
Mutation rate	0.01

Table 2. Summary of parameters used for our evolutionary algorithm.

3.2. Evolutionary Algorithms

The experiments were conducted using two evolutionary algorithms: (1) a population-based EA, with a steady state approach with tournament selection, one-point crossover and bit-flip mutation; and (2) a 1+1-evolution strategy (only one individual is evolved using mutation). We will refer to them in this work as “EA” and “1+1-ES”, respectively.

The parameters used for the EA are shown in Table 2. The 1+1-ES uses mutation as the only genetic operator. It was run for 20,000 generations, giving the same total number of fitness evaluations as the EA. The parameter values and type of operators were determined empirically in pilot experiments. From these experiments there are some elements worth mentioning. For example, a two-point crossover decreased the performance of our EA, whereas the one-point crossover, which we used in our work, exhibited a slightly better performance. We also realised that a much lower number of generations were required when using 10 and 30 EVs to find good results. However, we decided to use the same number of generations over all the EVs scenarios (from 10 up to 90 EVs) to make a fair comparison.

Because of the stochasticity in the initial conditions on each day and in the EA, we performed extensive independent runs (we executed 672 runs in total, see Table 3 for details). Runs were stopped when the maximum number of generations was reached.

3.3. Baselines

We will provide three simple methods of setting charging schedules as baselines against which to compare EA performance, named “Greedy”, “Midnight” and “Random”.

The Greedy baseline represents one typical customer behaviour. It represents the behaviour of a customer who begins charging the EV at 18:00, ignoring cost, until the time of departure (07:30) or until it is fully charged (i.e. $SoC_{max,m}$). This is not an unrealistic behaviour for many customers. It is fully deterministic. For evaluation purposes, the Greedy approach can be represented in the EA framework by a matrix where each row is of the form

$$\begin{aligned}
 & [1, 1, \dots, 1, 1, 1, \dots, 1] \quad (\text{if charging has not finished at } 07:30) \\
 & [1, 1, \dots, 1, 1, 0, \dots, 0] \quad (\text{if charging finishes before } 07:30)
 \end{aligned}$$

where the transition point from charging (1) and non-charging (0) is determined by the EV’s initial SoC, capacity, and target SoC.

The Midnight baseline represents the behaviour of a customer who aims to use the cheapest possible electricity, and so starts charging at midnight when the off-peak price applies, and continues until either the EV is charged or the customer leaves home at 07:30. For evaluation purposes, the Midnight approach can be represented in the EA framework by a matrix where each row is of the form

$$\begin{aligned}
 & [0, 0, \dots, 0, 1, 1, \dots, 1] \quad (\text{if charging continues until } 07:30), \text{ or} \\
 & [0, 0, \dots, 0, 1, 1, \dots, 1, 0, 0, \dots, 0] \quad (\text{if charging finishes before } 07:30)
 \end{aligned}$$

Description	Value
Number of simulated days	28
Number of fitness functions	3
Number of EV fleets	4
Number of algorithms (EA and 1+1-ES)	2
Total number of runs	672*

*672 = 28 * 3 * 4 * 2.

Table 3. Total number of runs executed for our experiments.

Approach	Electric Vehicles			
	10	30	60	90
1+1 ES Charging	1'43"	3'53"	2'05"	2'47"
1+1 ES Steady	3'57"	10'10"	13'52"	12'21"
1+1 ES Price-Based	4'27"	6'50"	4'28"	4'32"
EA Charging	2'00"	5'21"	5'29"	8'01"
EA Steady	4'09"	11'00"	12'53"	15'51"
EA Price-Based	4'11"	9'52"	11'28"	15'32"

Table 4. CPU time using different sizes of EV fleets over a period of 28 days, using the three proposed fitness functions (i.e., Charging, Steady Charging, Price-Based Charging) and using two variants of EAs (i.e., 1+1-ES and EA). Notice that this CPU time excludes compiling the simulator, which takes approximately one minute. This has to be compiled for each of scenarios shown in this table.

The Random baseline represents, perhaps, a more realistic behaviour of customers given that it incorporates randomness. Each EV begins charging at a random time and ends at a random (later) time. For evaluation purposes, the Random approach can be represented in the EA framework by a matrix where each row is of the form

$$[0, 0, \dots, 0, 1, 1, \dots, 1, 0, 0, \dots, 0]$$

4. Results and Discussion

As discussed throughout the paper, we are interested, broadly speaking, in maximising the charge of the EVs' batteries, while at the same time balancing the transformer load.

Thus, in the following paragraphs we focus our attention on the performance of our proposed approach by analysing the total EV load at the transformer over time and how this translates to charge in the EVs' batteries by analysing the initial and final state of charge. We also analyse what the impact is on the peak-to-average ratio. Finally, we discuss and analyse the implications of the proposed approach in terms of electricity costs.

4.1. Overall Performance

Let us start our analysis by considering the performance of our proposed approaches in terms of their fitness values, as defined in Section 2.2. The three fitness functions are named (a) Charging, (b) Steady Charging and (c) Price-Based Charging. According to all three definitions of fitness, higher fitness values are preferred. Their performance is shown in Figure 2 when using a 1+1-ES (top row) and when using a population-based EA (bottom row). Evolutionary time, i.e. generations, is shown on the 'x-axis', and fitness on the 'y-axis'. As mentioned before, both the 1+1-ES and the EA use the same number of fitness evaluations (e.g., 20,000 generations, as seen in the 'x-axis' of Figure 2 (top), for the 1+1-ES evolving 1 individual; and 200 generations, shown in the 'x-axis' of Figure 2 (bottom), for the EA using 100 individuals). All three fitness functions show the expected tendency to yield higher fitness values as search progresses. CPU time required to run the experiments is shown in Table 4 using a desktop computer with Intel Core i7-2600 CPU clocked at 3.40GHz and 8GB RAM.

Broadly speaking, the results show that the 1+1-ES slightly out-performs the EA, and that performance for both algorithms tends to decrease slightly as problem size increases (the 10- up to 90-EV cases).

The Charging fitness function, formally defined in Equation 4, aims to charge the EVs' batteries as much as possible. This is the simplest function defined in our work. The 1+1-ES achieves very good results for this function (Figure 2 (a)), with the EA (Figure 2 (d)) achieving equally good results for the smallest problem (10 EVs) and slightly worse for larger ones.

The Steady Charging fitness function, formally described in Equation 7, is built on top of the Charging fitness function (it aims to charge the EVs' batteries as much as possible) but also considers a steady charging of the EVs' batteries. As explained in Section 2, this function works in two stages: it aims at charging the EVs' batteries up to a certain point, $SoC_{min} = 80\%$ of capacity in Equation 6) and then it tries to find solutions that draw power steadily from the transformer (low PAR). When using the 1+1-ES to optimise this fitness function (Figure 2 (b)), we can observe

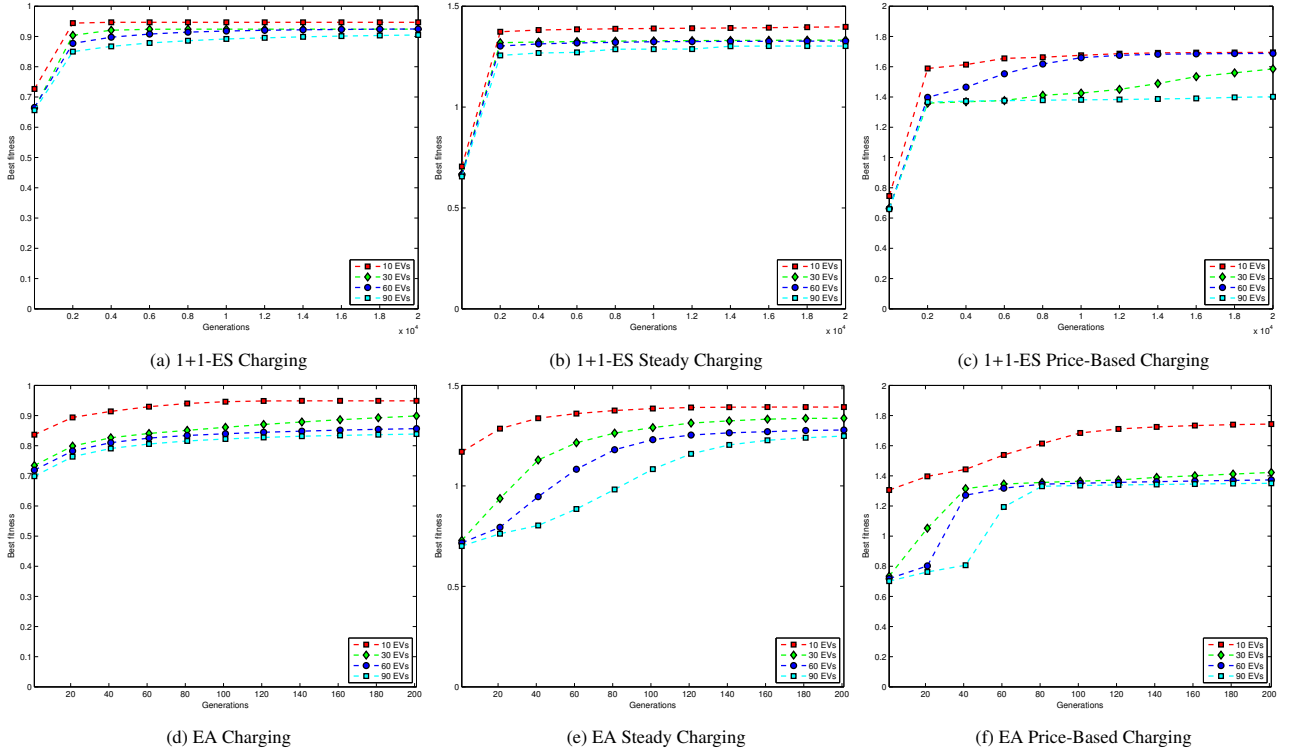


Figure 2. Average of best fitness values over 28 independent runs for each of the three fitness functions proposed in this work, named (a) Charging, (b) Steady Charging and (c) Price-Based Charging, using a 1+1-ES (top row) and a EA (bottom row). The fitness values denote the performance of our proposed fitness functions using different number of EVs: 10, 30, 60 and 90, specified with the red-filled squares, green-filled diamonds, blue-filled circles and magenta-filled squares, respectively. Higher fitness values denote better performance.

good performance after only 2,000 generations (the equivalent of 20 generations with the EA). Thereafter the fitness values improve marginally if at all. In contrast, the EA achieves good results slowly (Figure 2 (e)). Even at the end of the run, the results on the largest case (90 EVs) are improving but are still not quite as good as for the 1+1-ES. This may suggest that longer runs would allow the EA to continue to improve, eventually out-performing the 1+1-ES.

The third and last fitness function proposed in our work, named Price-Based Charging, formally described in Equation 9, works in three stages: it aims to charge the EVs' batteries up to a certain point (i.e., SoC 80%), then it tries to reduce electricity costs given a pricing signal. Once search is able to meet these two targets, it tries to achieve a constancy at the transformer load (Equation 6). The performance of this function when using the 1+1-ES is depicted in Figure 2 (c). We can see a trend similar to that of the previous two fitness functions when using this particular algorithm. That is, performance drops for the larger problems (60- and 90-EV cases). The EA (Figure 2 (f)) marginally out-performs the 1+1-ES on the 10-EV case, but again performance is somewhat worse for larger problems.

From the above discussion, we have learned that both algorithms are capable of improving all three fitness functions, and that the 1+1-ES has a slight advantage over the EA. However the functions' impact on the individual goals of the work (transformer load, SoC, PAR, and electricity costs) is not yet clear. This is particularly so for the functions that work in two phases (i.e., the Steady- and Price-Based Charging), due to the fact that the fitness values are a combination of these stages. In the following sections, the fitness functions' impact on the individual goals will be examined in turn and compared to the baselines described in Section 3 (i.e. the Greedy, Midnight, and Random baselines).

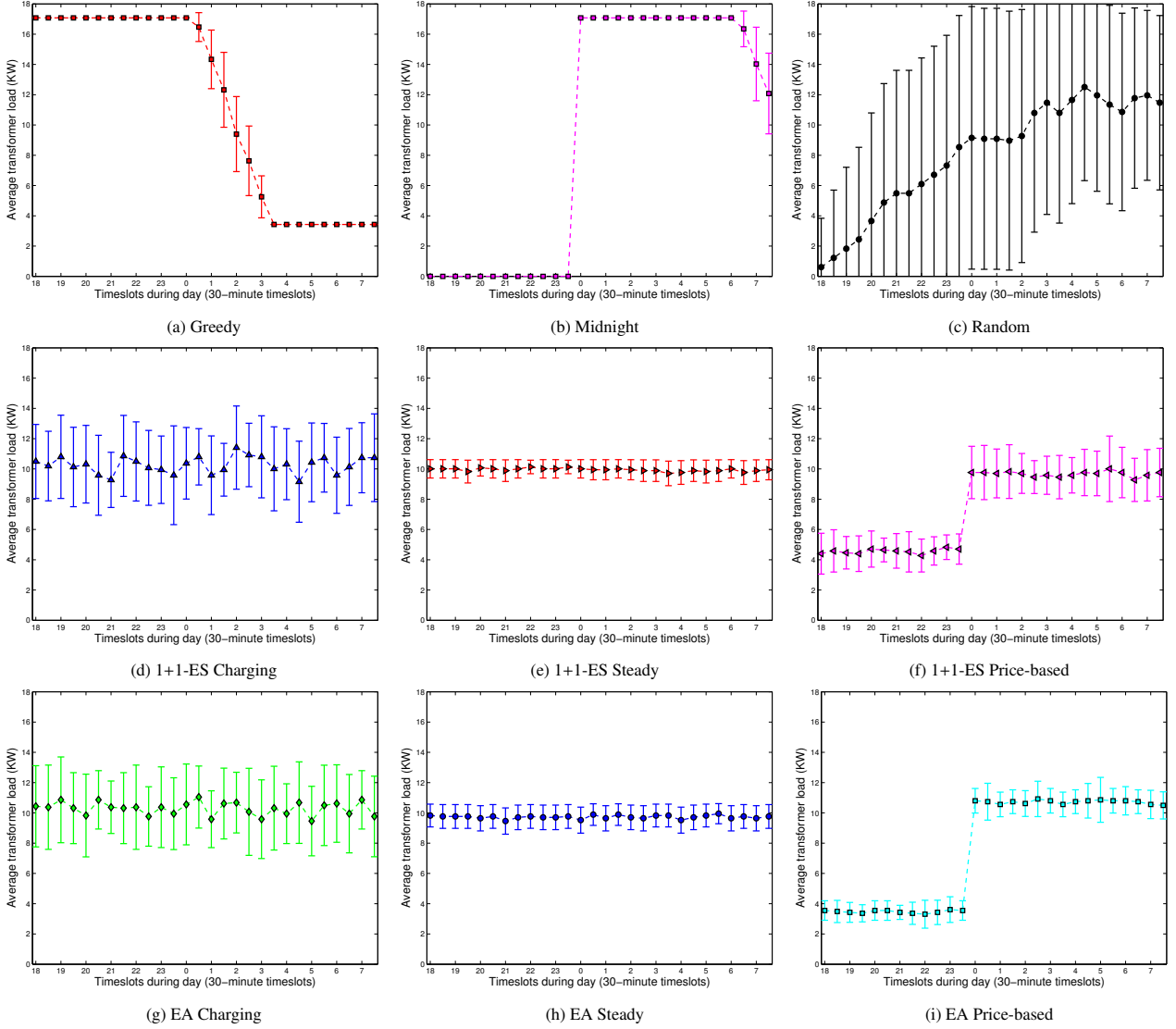


Figure 3. Transformer load, averaged for 28 days, for 10 electric vehicles (EVs). From top to bottom and from left to right, the figures show the transformer load for the (a) Greedy-, (b) Midnight-, (c) Random approach, (d) 1+1-ES Charging, (e) 1+1-ES Steady Charging, (f) 1+1-ES Price-Based Charging, (g) EA Charging, (h) EA Steady Charging and (i) EA Price-Based Charging.

4.2. Transformer Load

Let us consider the load over time, averaged over a period of 28 simulated days, depicted in Figures 3, 4, 5 and 6, for 10, 30, 60 and 90 EVs, respectively, when using the Greedy, Midnight and Random baselines, and using both the 1+1-ES and a EA with the three fitness functions.

Results for the Greedy approach are shown with red squares in Figures 3 (a), 4 (a), 5 (a) and 6 (a) (top-left image in all cases). It results in a high transformer load from 18:00. Demand begins to drop from 00:00 as some EVs become fully charged. It represents one of the two worst-case scenarios for transformer load (the other is the Midnight approach, discussed in the next paragraph) because it simulates that an EV starts charging as soon as it reaches home. As expected, this approach gives a higher transformer load during the 18:00-00:00 period than either of the algorithms with any of the three fitness functions. We can also see that with the Greedy approach it is not possible to fully charge all the EVs' batteries for each of the simulated days since with this method the transformer load never drops to zero, as shown in Figures 3 (a), 4 (a), 5 (a) and 6 (a) for 10, 30, 60 and 90 EVs, respectively. The reason why this happens

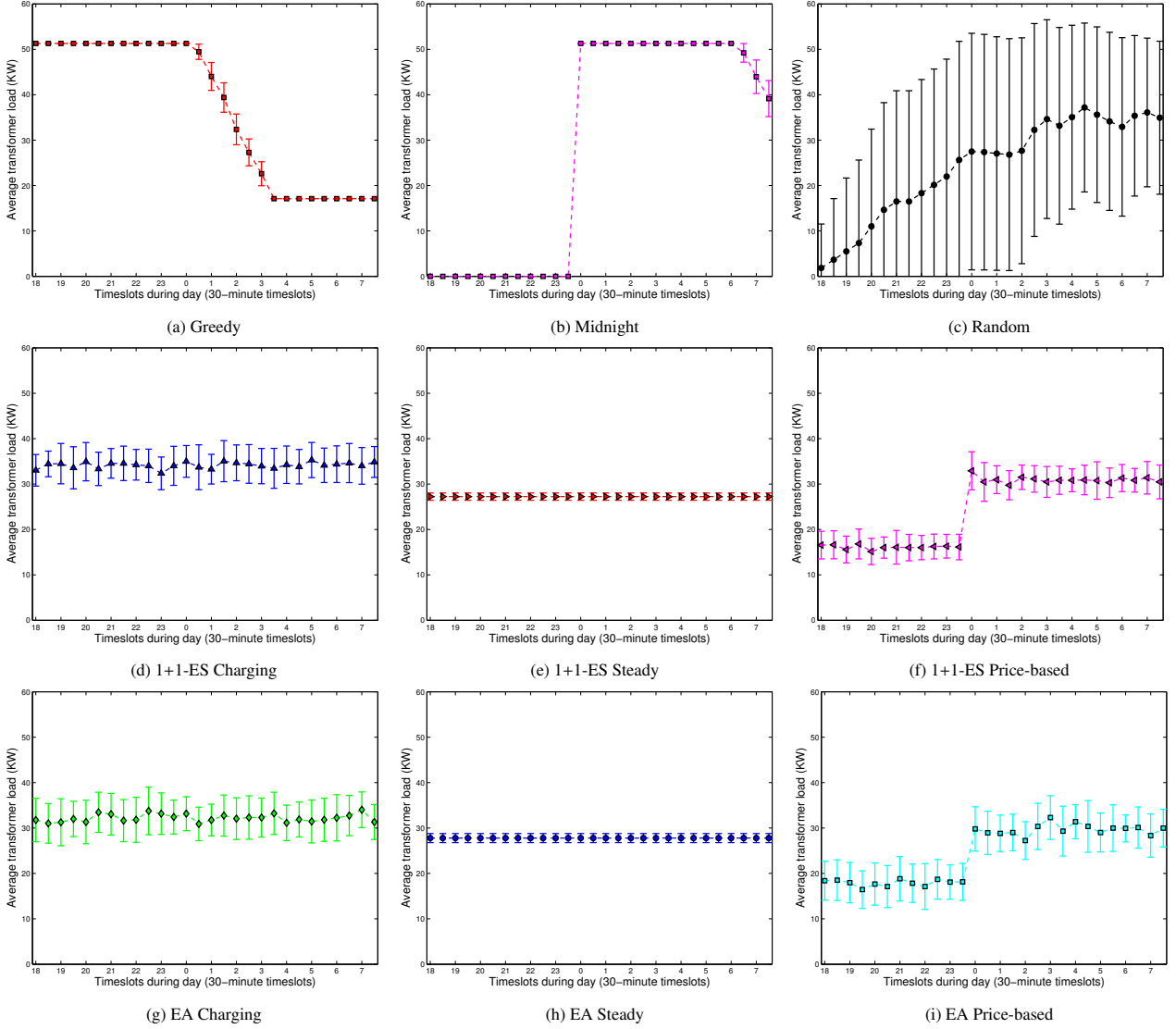


Figure 4. Transformer load, averaged for 28 days, for 30 electric vehicles (EVs). From top to bottom and from left to right, the figures show the transformer load for the (a) Greedy-, (b) Midnight-, (c) Random approach, (d) 1+1-ES Charging, (e) 1+1-ES Steady Charging, (f) 1+1-ES Price-Based Charging, (g) EA Charging, (h) EA Steady Charging and (i) EA Price-Based Charging.

is due to three main factors: the battery sizes, the limit on charging rate, and the period of time during which EVs can be charged (these specifications and their corresponding values are shown in Table 1).

The Midnight approach is depicted with magenta squares in Figures 3 (b), 4 (b), 5 (b) and 6 (b) for 10, 30, 60 and 90 EVs, respectively (top-centre image in all cases). Here, load is zero until 00:00, then jumps to a maximum and remains at this value until some EVs become fully charged. It is equally as bad as the Greedy approach, and in fact, it shows the same transformer load achieved by the Greedy approach, but at a different time. Because of the nature of this approach (i.e., starts to charge the EVs at midnight, when the electricity cost is the lowest) and by considering the previous approach, which starts charging the EVs 6 hours before compared to this approach, it is clear that none of the EV fleets will be fully charged at the time of departure, as can be observed in the referred figures. For instance, if we consider the case with 10 EVs, shown in Figure 3 (b) we can see that at 07:30, the average transformer load is around 12kW, whereas for the Greedy approach, the average transformer load is around 4kW, depicted in Figure 3 (a). The same trend is observed for the rest of the EV fleets (i.e., 30, 60 and 90 EVs). We will further discuss the implications

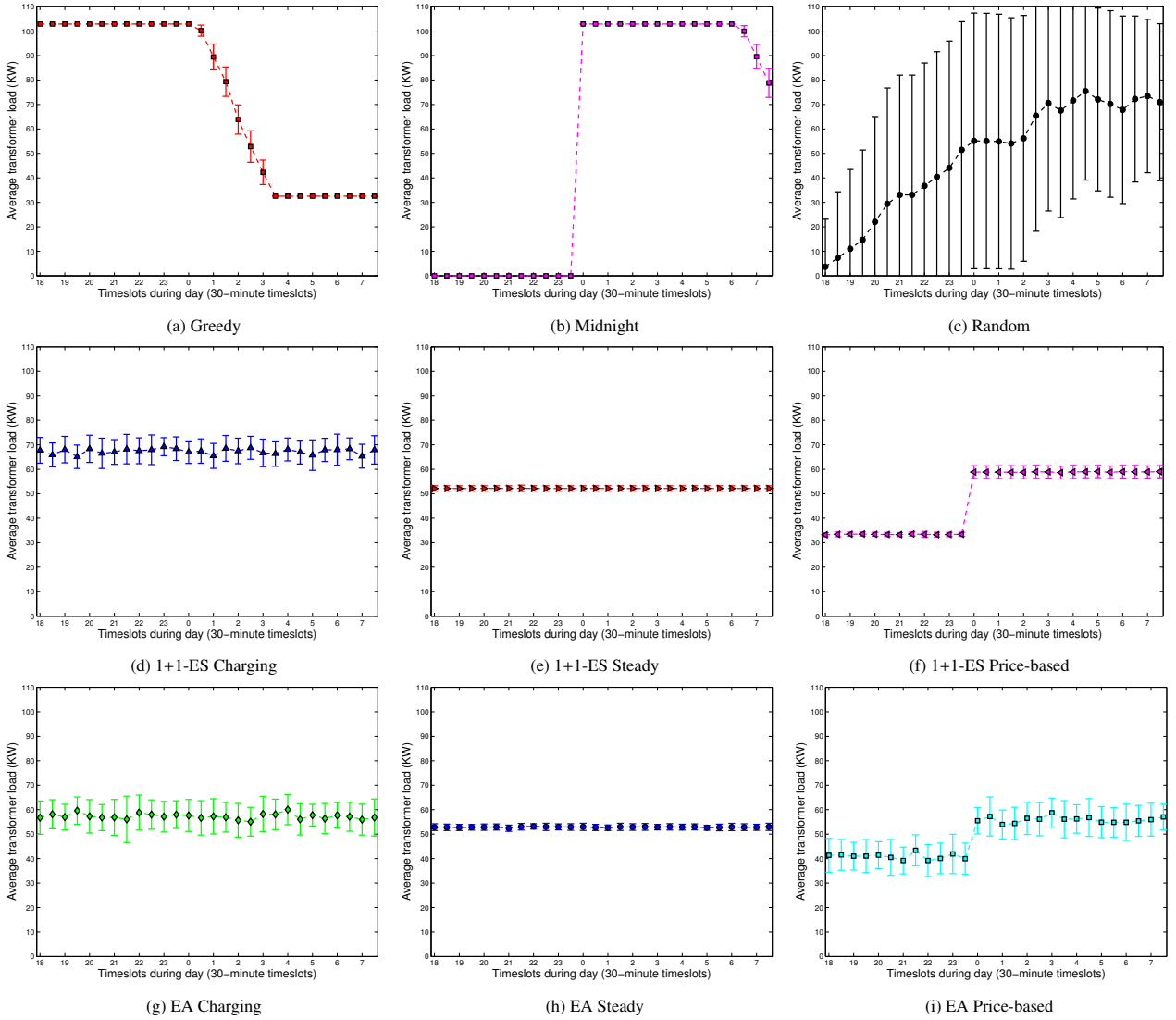


Figure 5. Transformer load, averaged for 28 days, for 60 electric vehicles (EVs). From top to bottom and from left to right, the figures show the transformer load for the (a) Greedy-, (b) Midnight-, (c) Random approach, (d) 1+1-ES Charging, (e) 1+1-ES Steady Charging, (f) 1+1-ES Price-Based Charging, (g) EA Charging, (h) EA Steady Charging and (i) EA Price-Based Charging.

of this transformer load when analysing the state of charge, peak to average ratio and electricity costs later in this section.

We now turn our attention to the last non-intelligent approach, the Random approach, shown with black circles in Figures 3 (c), 4 (c), 5 (c) and 6 (c) for 10, 30, 60 and 90 EVs, respectively (top-right image in each case). Unsurprisingly, we can see that the average transformer load is reduced substantially compared to the values found during charging periods by the other two non-intelligent approaches (i.e., Greedy and Midnight), discussed previously. This is to be expected given the nature of this approach. That is, each EV may begin to charge at any point in the time period (i.e., from 18:00 until 07:30 as specified in Table 1). Clearly, this approach is the best amongst the non-intelligent algorithms in terms of keeping the transformer load low. Its effect on the final state of charge, peak to average ratio, and electricity costs will be discussed later in this section.

Let us now turn our attention to the transformer load using a 1+1-ES approach with each of the three fitness functions proposed in this work: Charging, Steady Charging and Price-Based Charging, denoted by the triangles, in

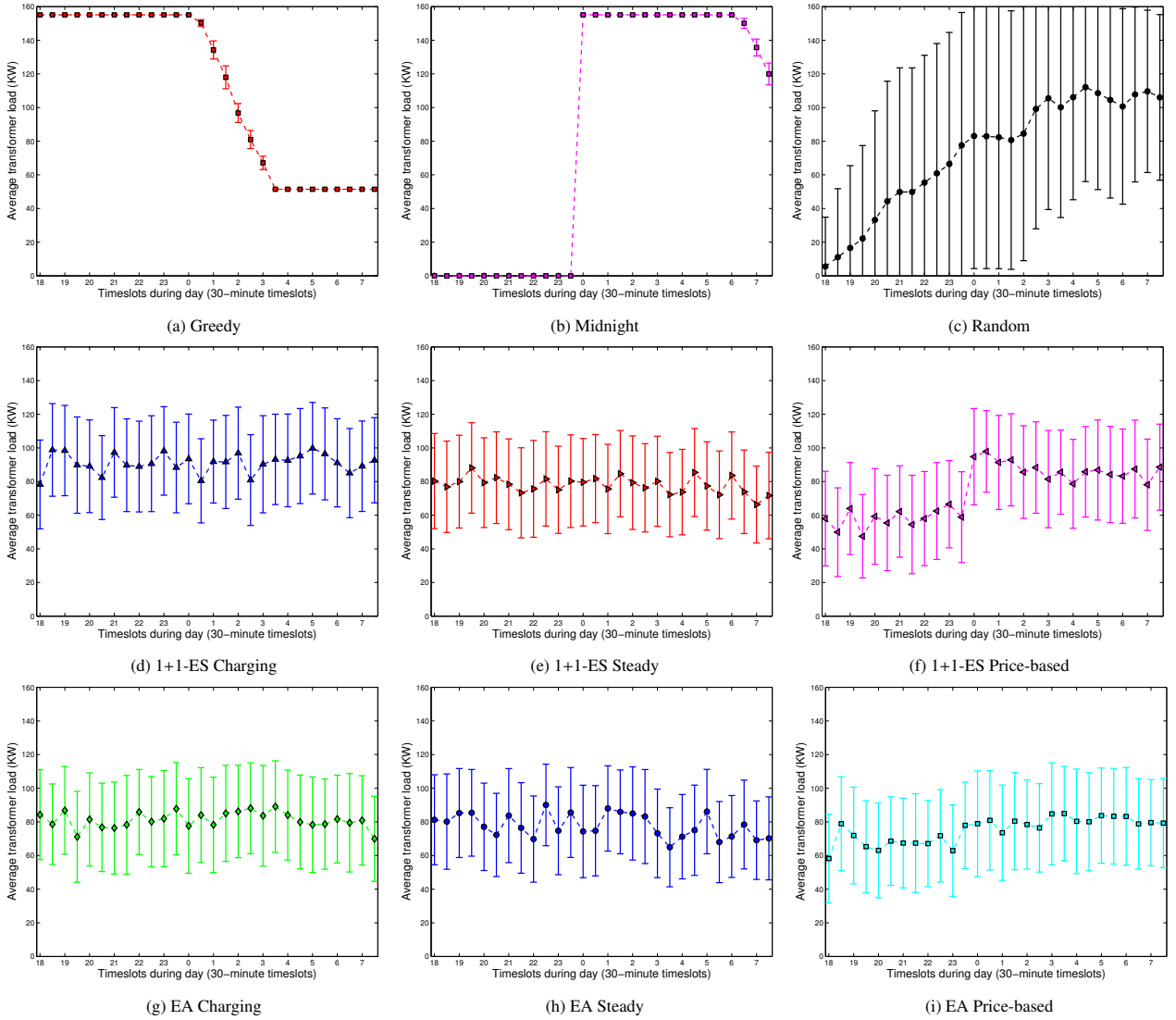


Figure 6. Transformer load, averaged for 28 days, for 90 electric vehicles (EVs). From top to bottom and from left to right, the figures show the transformer load for the (a) Greedy-, (b) Midnight-, (c) Random approach, (d) 1+1-ES Charging, (e) 1+1-ES Steady Charging, (f) 1+1-ES Price-Based Charging, (g) EA Charging, (h) EA Steady Charging and (i) EA Price-Based Charging.

the middle row (d)–(f), from left to right, of Figures 3, 4, 5 and 6 for 10, 30, 60 and 90 EVs, respectively. If we focus our attention on the 10-EV case, shown in Figure 3, and consider the Charging function, we can observe that the algorithm tends to reduce the transformer load compared to the first two non-intelligent approaches, despite not explicitly aiming to minimise it. In fact, the average transformer load is something over half of that achieved during charging with the Greedy and Midnight approaches, with a high variability (the standard deviation is shown by the vertical lines). A similar trend can be seen when using 30 EVs, as seen in Figure 4 (d), 60 EVs as shown in Figure 5 (d), and 90 EVs as depicted in Figure 6 (d). A similar tendency can be seen when using the same algorithm (1+1-ES) and the two other fitness functions proposed in this work. For instance, if we continue analysing the 10 EV case, we can see that the Steady Charging function, formally defined in Equation 6 yields similar results: see Figure 3 (e), compared to the Charging function as shown in Figure 3 (a), discussed in the previous paragraph. Moreover, the Price-Based Charging function again yields similar results (see Figure 3 (f) compared to the Charging function). This intuitively means that when using the 1+1-ES and these fitness functions, one should expect similar results when

analysing the final SoC, electricity costs, and PAR. We will discuss these later in this section.

We now analyse behaviour of the population-based EA (or simply EA) with the three proposed fitness functions. The results obtained by the EA are shown in the bottom row (g)–(i) of Figures 3, 4, 5 and 6 for 10, 30, 60 and 90 EVs, respectively. We see that results are very similar to those of the 1+1-ES, and again are superior to those of the Greedy, Midnight, and Random approaches. There are some elements worth mentioning for each of the functions presented in this work.

For instance, when using the Charging fitness function (bottom left-hand corner, Figures 3 (g), 4 (g), 5 (g) and 6 (g)), it can be perceived that the average transformer load is, for the smaller problem sizes (10, 30 and 60 EVs), larger than when using the Steady Charging function (bottom centre, Figures 3 (h), 4 (h), 5 (h) and 6 (h)). However, the Steady Charging function has a much more smooth load compared to the Charging fitness function. This is to be expected since the latter function rewards candidate solutions that tend to show less variance in terms of the actions executed by the EVs. The difference between these two functions is less clear when using 90 EVs, as depicted in Figure 6 (g) and (h).

Finally, let us focus our attention on the effect of the Price-Based Charging functions on the average load at the transformer, shown in Figures 3 (i), 4 (i), 5 (i) and 6 (i) (bottom-right image in each case), when using 10, 30, 60 and 90 EVs, respectively. This function is built on the previously discussed functions. It aims to charge the EVs' batteries up to a certain level (i.e., 80%). Once they reach this level, the function tries to achieve a certain reduction in electricity cost. Finally it aims for balanced load. As can be seen, this function works fairly well as it follows the pricing signal (recall that the lowest price is from midnight onward as specified in Table 1), which results in the 'stair' shape seen in Figures 3 (i), 4 (i) and 5 (i). The situation is less clear when using 90 EVs, as seen in Figure 6 (i).

As indicated throughout the paper, the proposed approach, discussed in Section 2, aims at charging the EVs' batteries as much as possible at the time of departure, balancing transformer load, and reducing consumer electricity costs. Only the first of these goals is captured in the Charging function defined in Equation 4, whereas the Steady Charging considers the first two goals, and Price-Based Charging considers all three.

4.3. State of Charge

From our previous analysis on the transformer load and the three fitness functions, we know that the Charging function will achieve a higher final state of charge (SoC) compared to that achieved by the two other functions. However, it remains unclear exactly what final SoC will be achieved for both the non-intelligent approaches (i.e., Greedy, Midnight and Random) and the EA approaches (i.e., 1+1-ES and population-based EA) using each of the three fitness functions.

Thus, to shed some light on this, we calculated the final SoC for all the approaches discussed previously using 10, 30, 60 and 90 EVs. The results are shown in Figure 7 (a)–(d), respectively. Recall that the initial SoC for each of the EVs, for each of the 28 simulated days, is between 40% and 60%, as specified in Table 1. It can be seen that, unsurprisingly, the Greedy approach, shown in the first boxplot, achieves the highest final SoC, regardless of the number of EVs used. This, however, is achieved at a great cost since the transformer load is fairly high, as discussed in the previous paragraphs. The Midnight approach also shows a high final SoC given that all EVs start to charge as soon as the electricity price drops at 00:00, and continue until the time of departure (18 slots over 28 available time slots, see Table 1 for details). This intuitively means that one should expect to see a low electricity price when using this approach, but it has the drawback of having a high transformer load from 00:00, as discussed in the previous paragraphs. The Random approach shows the lowest SoC among all the approaches (see third boxplot in each of the plots of Figure 7), including the EA methods used in this work. This is to be expected by its nature. Recall that any EV could be charged during any time over the charging period. Moreover, it could be charged for a short period (e.g., 1 slot = 30 minutes).

Let us now analyse the results yielded by both EA methods: 1+1-ES and population-based EA. To facilitate our analysis, we have put together in Figure 7 the results from methods using each of the fitness functions proposed in this work. From the formal definition of these functions, introduced in Section 2 and from the analysis conducted throughout all the paper, we know that the Charging function should result in having a high SoC, and this is found in the fourth and fifth boxplots of Figure 7 (b)–(d). For the 10-EV case (Figure 7 (a)) there is no difference.

If we now focus our attention on the results yielded by the Steady Charging function, shown in the seventh and eighth boxplots of Figure 7 (a)–(d), we can see a similar scenario compared to the Charging function, described above.

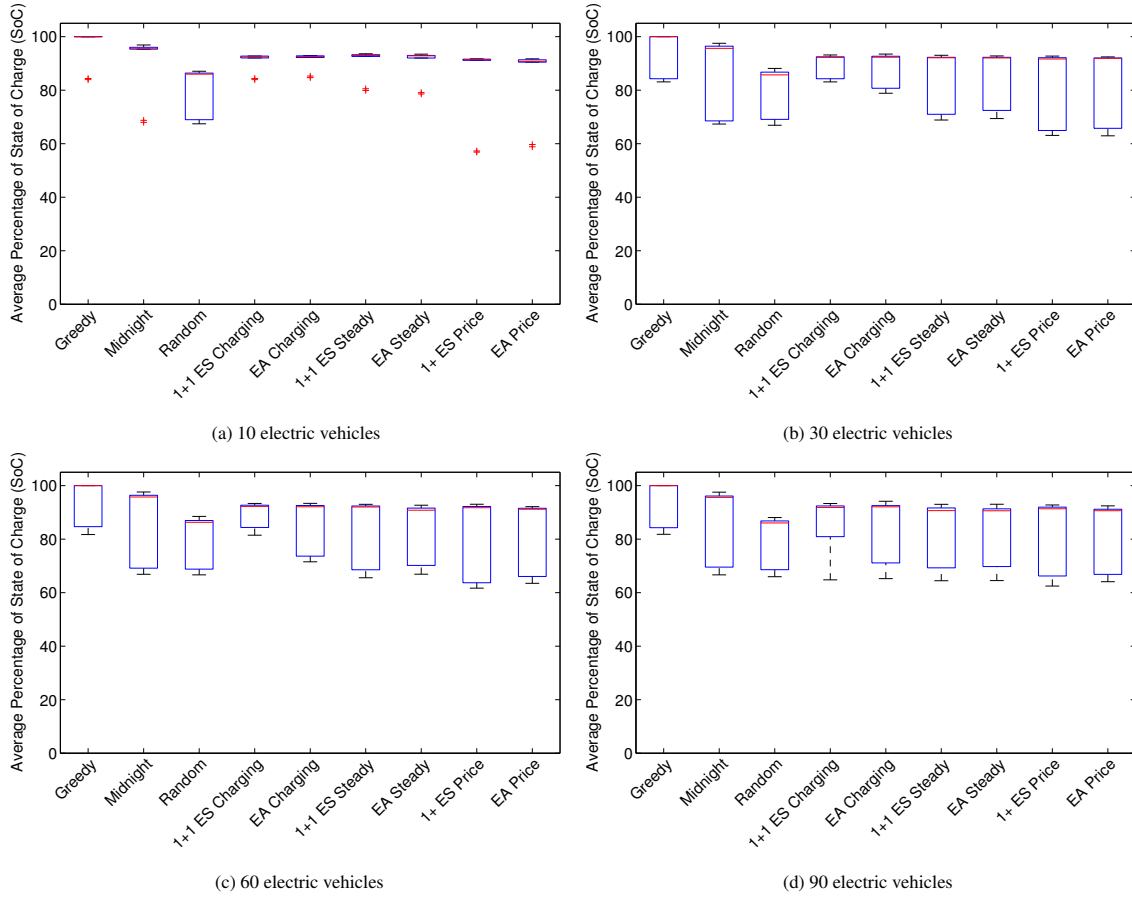


Figure 7. Percentage of state of charge (*SoC*), averaged for 28 simulated days using (a) 10, (b) 30, (c) 60 and (d) 90 electric vehicles (EVs). The figures show a summary of the results achieved by the non-intelligent algorithms (i.e., Greedy, Midnight and Random approach) and the two variants of EAs (i.e., 1+1-ES and population-based EA) using the three proposed fitness functions: Charging, Steady Charging and Price-Based Charging, specified in the ‘x-axis’ of the figures. The variance within each scenario comes about partly through the fact that the initial conditions vary during the 28 simulated days, as well as the non-deterministic behaviour of the EA.

The differences between the 1+1-ES and EA, if any, are small. However, the SoC achieved by this Steady Charging function is often lower than that of the Charging function. This is also to be expected given the features of the former function. That is, it tends to charge an EV as much as possible while at the same trying to make a low fluctuation at the transformer load. Thus, as a result of the last constraint, one should expect a lower SoC.

The final fitness function, Price-Based Charging function, shown in the last two boxplots of Figure 7 (a)–(d), shows a slightly lower SoC compared to the previous fitness functions, regardless of the EA approach used. Again, by analysing both the fitness function and the results on the transformer load, we can see that the rather erratic behaviour observed on the transformer load should result on having a lower SoC compared to the other two functions, as discussed in the previous paragraphs.

4.4. Peak-to-average ratio

The peak-to-average ratio (PAR) is calculated by the maximum load demand for a period of time divided by the average load demand (see Equation 5). A lower PAR is preferred [29] via [3]. The total required generating capacity is determined by the peak load, hence a low PAR allows for a reduction in the total required generating capacity. PAR is related to but distinct from the Steady Charging objective, in that Steady Charging is motivated by a desire to reduce the inter-time-step fluctuation in load, whereas PAR is concerned only with the peak and the average load. It is possible to achieve a relatively low PAR despite a relatively bad (“non-steady”) Steady Charging behaviour.

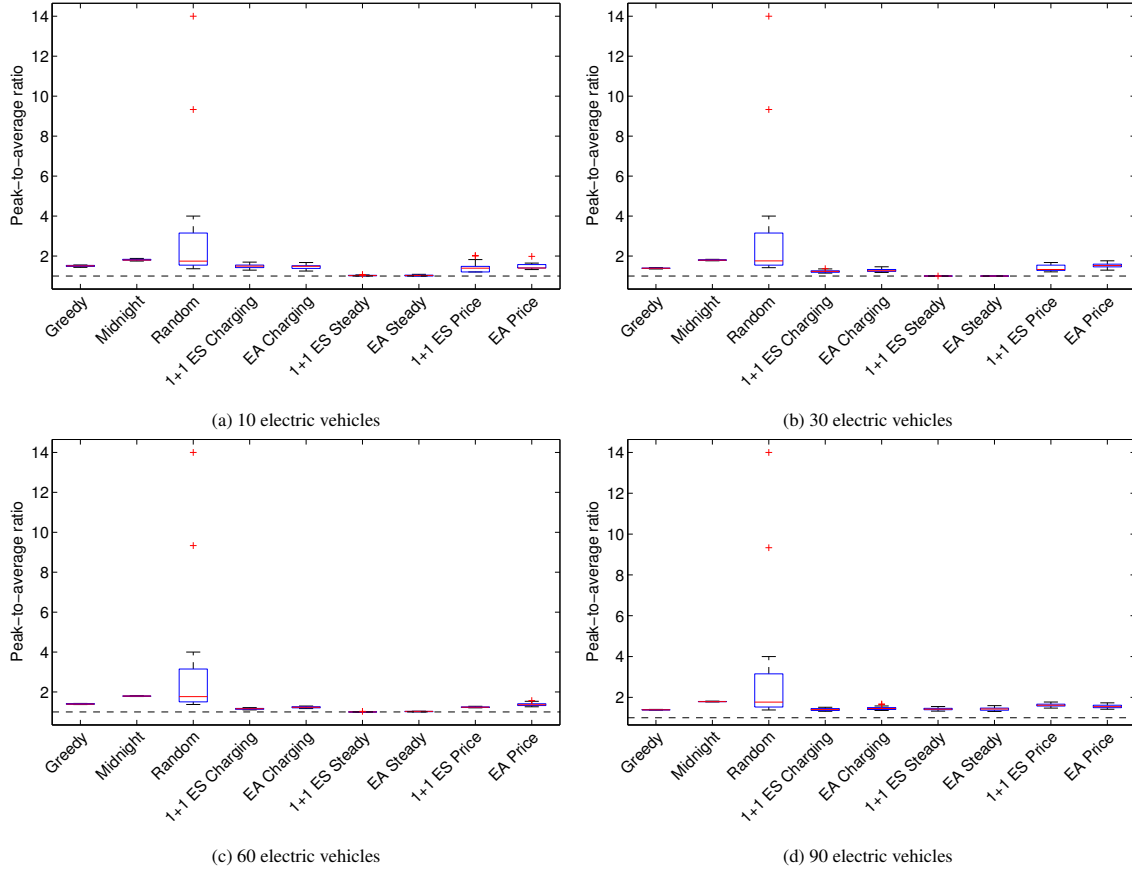


Figure 8. Peak-to-average ratio (PAR), averaged for 28 simulated days using (a) 10, (b) 30, (c) 60 and (d) 90 EVs. The figures show a summary of the PAR when using the non-intelligent algorithms (i.e., Greedy, Midnight and Random approach) and the two variants of EAs (i.e., 1+1-ES and population-based EA) using the three proposed fitness functions: Charging, Steady Charging and Price-Based Charging, specified in the ‘x-axis’ of the figures.

The PAR achieved by the non-intelligent approaches (i.e., Greedy, Midnight and Random) and our proposed approach using each of the three fitness functions when using a 1+1-ES and a population-based EA is summarised in Figure 8. The horizontal dotted line shown in the figure is the optimal value that can be obtained and it is taken as a baseline against which to compare all the approaches used in this work.

Considering only the non-intelligent approaches, the Greedy approach achieves the lowest PAR, followed by the Midnight approach and the Random approach. To see why this happens, one should recall how each of these approaches work. For instance, the Midnight approach achieves higher PAR compared to the Greedy approach because although both reach the same peak (at 00:00 and 18:00 respectively), the Midnight approach has a zero load before 00:00 whereas the Greedy approach never draws a load of zero. The Random approach achieves the highest PAR amongst these approaches given its constant fluctuation.

If we now consider the EA approaches using the three proposed fitness functions, we see some differences worth mentioning. Let us start with the best (lowest) PAR, yielded by the Steady Charging function and using both the population-based EA and the 1+1-ES, shown in the sixth and seventh boxplots of Figure 8 (a)–(d). We can see that for the 10, 30 and 60 EV scenarios, shown in Figure 8 (a) - (c), the PAR value is almost the ideal as it is close to the horizontal line, which denotes the optimal PAR value. This, however, is not observed for the case of 90 EVs, which agrees with the rather erratic result observed at the transformer load shown in Figure 6 (i). The other two fitness functions, Charging and Price-Based Charging, yield high PAR values regardless of both the number of EVs and the type of EA employed. This is in fact to be expected. For the Charging function, one should recall that this function does not consider to reward candidate solutions based on a low fluctuation of the load. On the other hand,

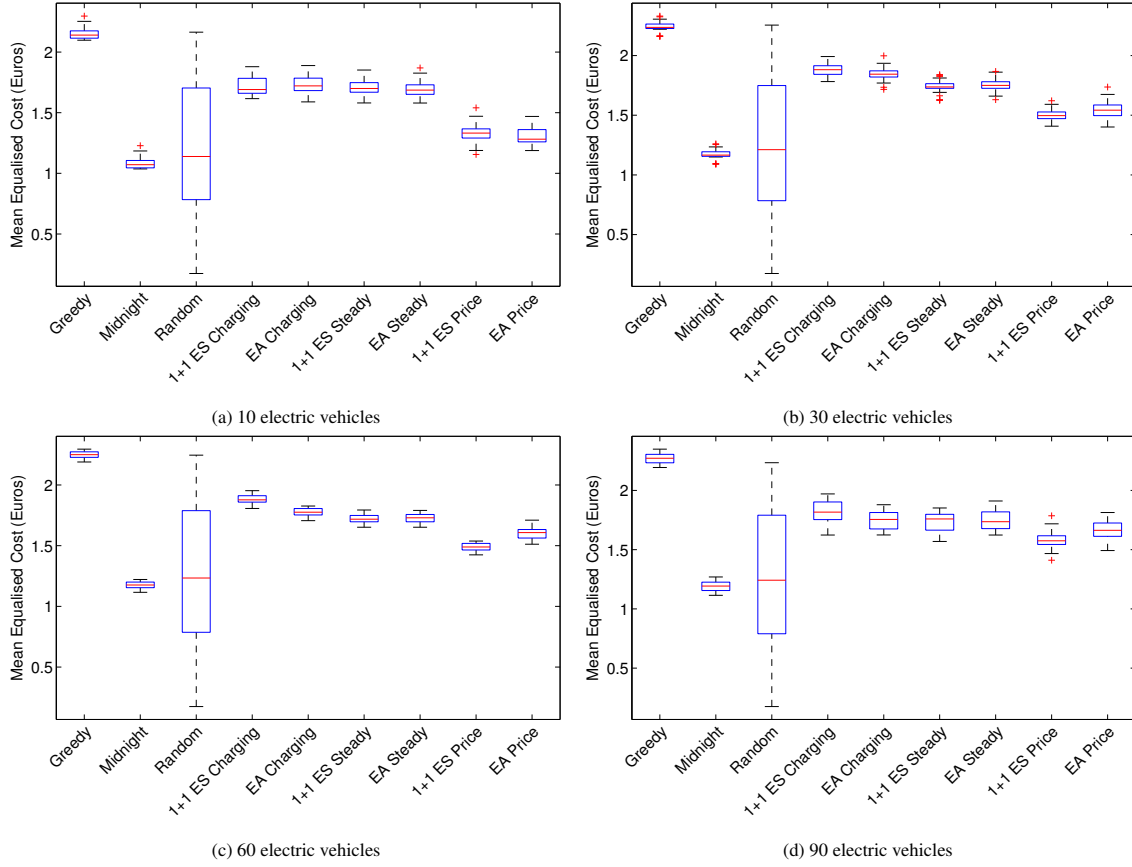


Figure 9. Electricity costs, in euros, averaged for the number of electric vehicles used (a) 10, (b) 30, (c) 60 and (d) 90 electric vehicles. The figures show an average of the electricity cost when using the non-intelligent algorithms (i.e. Greedy, Midnight and Random approach) and the two variants of EAs (i.e., 1+1-ES and population-based EA) using the three proposed fitness functions: Charging, Steady Charging and Price-Based Charging, specified in the ‘x-axis’ of the figures.

the Price-Based Charging function does consider the fluctuation. However, the high PAR observed by this function is mainly caused by the drastic change in price, resulting in having a fairly big jump at the transformer load at 00:00, as observed by the ‘stair’ shape in Figures 3 (i), 4 (i), 5 (i), for 10, 30 and 60 EVs, respectively.

4.5. Electricity costs

As measures of the performance of the plans produced, we have considered transformer load over time (Figures 3–6), the state of charge (Figure 7), and the peak-to-average ratio (Figure 8). It remains to consider the electricity cost. The natural way to measure this is as the total cost of the electricity drawn by all EVs over the 28 simulated days. However, it is necessary to account for the fact that some plans draw more electricity than others: this is reflected in the fact that the SoC at the end of the charging period is not uniform. Therefore, we have implemented a measure of electricity cost we will refer to as Equalised Cost. It uses the *minimum* SoC achieved for each EV as a reference point. Since all EVs achieve this SoC or higher, the cost up to this SoC can be compared fairly. Therefore, we simply calculate the cost of electricity per EV up to this minimum SoC, and disregard the electricity drawn after this SoC. That is, within this metric all EVs are seen as drawing the same amount of electricity. Furthermore, we calculate the electricity costs by considering the ToU and the peak and off-peak times, as discussed in Section 3 (a summary is provided in Table 1).

The results of this analysis are shown in Figure 9, with more details shown in Table 5. For all scenarios (10, 30, 60, and 90 EVs) there is quite a clear trend of decrease in Equalised Cost as we move from the Greedy approach, through the Charging and Steady Charging to the Price-Based Charging fitness function. The price per EV for the Greedy

<i>Approach</i>	<i>Consumption in kWh during peak hours</i>	<i>Consumption in kWh during off-peak hours</i>	<i>Equalised Cost</i>	<i>Equalised Cost per EV</i>
10 electric vehicles				
Greedy	102.21	7.92	21.88	2.18
Midnight	0	110.13	11.22	1.12
Random	38.73	71.40	14.32	1.43
1+1-ES Charging	60.87	49.26	17.20	1.72
EA Charging	65.99	44.14	18.10	1.81
1+1-ES Steady Charging	60.05	50.08	17.12	1.71
EA Steady Charging	60.96	49.17	17.58	1.75
1+1-ES Price-Based Charging	27.35	82.78	13.34	1.33
EA Price-Based Charging	27.14	82.99	14.05	1.40
30 electric vehicles				
Greedy	307.77	37.06	67.24	2.24
Midnight	0	344.84	35.13	1.17
Random	81.52	263.31	36.95	1.23
1+1-ES Charging	203.80	141.03	56.39	1.87
EA Charging	193.34	151.50	55.30	1.84
1+1-ES Steady Charging	163.43	181.41	52.18	1.73
EA Steady Charging	166.72	178.11	52.52	1.75
1+1-ES Price-Based Charging	96.78	248.05	44.88	1.49
EA Price-Based Charging	107.40	237.44	46.34	1.54
60 electric vehicles				
Greedy	617.41	75.38	134.99	2.24
Midnight	0	692.8	70.59	1.17
Random	163.54	529.26	73.91	1.23
1+1-ES Charging	405.05	287.75	112.79	1.87
EA Charging	342.31	350.48	106.30	1.77
1+1-ES Steady Charging	313.08	379.72	103.20	1.72
EA Steady Charging	319.23	373.56	103.89	1.73
1+1-ES Price-Based Charging	200.29	492.51	89.31	1.48
EA Price-Based Charging	244.61	448.18	96.10	1.60
90 electric vehicles				
Greedy	930.59	128.45	204.97	2.27
Midnight	0	1059.05	107.91	1.20
Random	246.49	812.56	112.90	1.25
1+1-ES Charging	544.94	514.11	163.93	1.82
EA Charging	490.65	568.39	159.09	1.76
1+1-ES Steady Charging	475.54	583.51	156.34	1.73
EA Steady Charging	481.31	577.73	158.11	1.75
1+1-ES Price-Based Charging	348.26	710.79	142.25	1.58
EA Price-Based Charging	414.73	644.31	151.17	1.67

Table 5. The Equalised Cost is calculated based on the number of kilowatt-hours drawn during peak and off-peak hours, equalized between schedules by taking into account only electricity up to a minimum SoC. The Equalised Cost per EV is calculated based on the Equalised Cost and the number of EVs.

approach is relatively constant, at about EUR2.18 – EUR2.27 per charging period. This relatively high price reflects the fact that the Greedy approach carries out its charging as early as possible, as seen in Figures 3–6, coinciding with the peak charging period. Obviously, the Midnight approach results in the lowest Equalised Cost given that it only uses the cheapest electricity cost. The Random approach also yields some fairly low Equalised Costs, although with a high variance. The reason the latter approach yields some low Equalised Costs is due to the fact that the off-peak period is longer than the peak period (see Table 1 for details).

The Charging and Steady Charging fitness functions tend to spread the load out more, hence as a by-product tend to take better advantage of the off-peak charging period, again as seen in Figures 3–6. The two are comparable, though Steady Charging spreads the load out slightly more and achieves slightly lower overall price.

However, the Price-Based Charging fitness function explicitly rewards low prices, and so is capable of achieving lower prices: approximately EUR1.40 for 10 EVs, up to EUR1.67 for 90 EVs, when using the population-based EA approach. However, its advantage over the other fitness functions, which is very clear for 10, 30 or 60 EVs, is far less clear for 90 EVs.

5. Conclusions and Future Work

We have implemented two variants of evolutionary algorithms (EAs): a 1+1-ES and a population-based EA to search for efficient charging schedules for fleets of EVs, achieving good results in terms of reducing peak demand and reducing consumers' electricity costs, while maintaining a high overall state of charge of EVs' batteries. We have tested these approaches on small to medium fleet sizes – 10, 30, 60 and 90 EVs – using realistic data generated by a state of the art grid simulator, over the course of 28 simulated days.

We have found that the 1+1-ES is capable of slightly out-performing a population-based EA, referred to in this work as EA. We have also shown that both the EA and the 1+1-ES approach exhibit better performance compared against the non-intelligent methods (i.e., Greedy-, Midnight- and Random approach) used in this work. This is a significant result because each of these non-intelligent methods reflects likely default behaviour for most consumers: in the Greedy approach, the EV is simply plugged in and charged up fully as soon as it arrives home each evening; in the Midnight approach, the EV is plugged in at midnight to take advantage of the cheapest electricity cost; and in the Random approach, the EV could be charged at any time during the simulated period for any length of time. In contrast, either of the EAs used in this work produces plans which take advantage of lower-cost pricing in the middle of the night, and at the same time reduce peak demand.

Although numbers of EVs are projected to be in the thousands or millions, Demand Aggregators may have much smaller fleet sizes. Therefore, the smaller fleet sizes considered here represent an important real-world case. The slight disimprovement in the EA's results noted in Section 4 for the 90-EV case is not a cause for great concern. However, in future work we hope to scale our results up further.

It is an assumption of this work that customers are willing to submit their charging schedules to a central authority, e.g. a Demand Aggregator. The intelligent algorithms used in this work (i.e., 1+1-ES and EA) require centralised knowledge (the number of EVs and their initial SoC) and centralised control (specification of when each EV should charge, up to 30-minute granularity). This assumption is not unrealistic. As we have seen, customers will do better through centralised control than through the most likely individualised behaviour, the Greedy approach. However, more informed and more price-conscious customers will be willing to deviate from the Greedy approach in order to avail of lower cost periods in the middle of the night. Such behaviour would lead to decreases in performance, in particular increases in peak demand. Therefore, to apply our work customers would have to be either contracted to submit control of charging schedules to the central authority, or induced to do so via a monetary reward. (Any such monetary reward is not considered in our calculations of electricity costs.)

In future work, this assumption could be removed, by modelling consumers as independently evolving agents seeking to reduce their own costs and maximise their own SoC. The price signal would then have to be modulated to induce a steady charging behaviour. Both the consumer's behaviours and the price signal policy could then be optimised in a coevolutionary setup.

We hope that the results achieved by our EA approach using the fitness functions proposed in this work and by using a well-developed grid simulator could attract the attention of companies to adapt this form of machine learning technique.

Acknowledgements

Edgar Galván López's research is funded by an ELEVATE Fellowship, the Irish Research Council's Career Development Fellowship co-funded by Marie Curie Actions. The first author would also like to thank the TAO group at INRIA Saclay & LRI - Univ. Paris-Sud and CNRS, Orsay, France for hosting him during the outgoing phase of the ELEVATE Fellowship. Paula Carroll is funded by a Science Foundation Ireland Sustainable Electrical Energy Systems (SEES) Strategic Research Cluster Award. The authors would like to thank all the reviewers for their useful comments that helped us to significantly improve our work.

- [1] COM, Energy 2020 – a strategy for competitive, sustainable and secure energy, Tech. Rep. 639, European Union (2010).
URL http://ec.europa.eu/energy/publications/doc/2011_energy2020_en.pdf
- [2] Y. Tang, H. He, Z. Ni, J. Wen, X. Sui, Reactive power control of grid-connected wind farm based on adaptive dynamic programming, *Neurocomputing* 125 (0) (2014) 125 – 133, advances in Neural Network Research and Applications Selected papers from the 9th International Symposium of Neural Networks, July 2012 *Advances in Bio-Inspired Computing: Techniques and Applications*. doi:<http://dx.doi.org/10.1016/j.neucom.2012.07.046>.
URL <http://www.sciencedirect.com/science/article/pii/S0925231213001628>

- [3] A. Mohsenian-Rad, V. Wong, J. Jatskevich, R. Schober, A. Leon-Garcia, Autonomous demand-side management based on game-theoretic energy consumption scheduling for the future smart grid, *Smart Grid, IEEE Transactions on* 1 (3) (2010) 320–331. doi:10.1109/TSG.2010.2089069.
- [4] E. Galván-López, C. Harris, L. Trujillo, K. R. Vázquez, S. Clarke, V. Cahill, Autonomous demand-side management system based on Monte Carlo tree search, in: *IEEE International Energy Conference (EnergyCon)*, IEEE Press, 2014, pp. 1325 – 1332.
- [5] E. Galván-López, A. Taylor, S. Clarke, V. Cahill, Design of an automatic demand-side management system based on evolutionary algorithms, in: *Proceedings of the 29th Annual Symposium on Applied Computing, SAC '14*, ACM, Gyeongju, Korea, 2014, pp. 525 – 530.
- [6] E. Galvan, C. Harris, I. Dusparic, S. Clarke, V. Cahill, Reducing electricity costs in a dynamic pricing environment, in: *Proc. Third IEEE International Conference on Smart Grid Communications (SmartGridComm)*, IEEE Press, Tainan, Taiwan, 2012, pp. 169 – 174.
- [7] A. Collin, J. Acosta, I. Hernando-Gil, S. Djokic, An 11 kv steady state residential aggregate load model. part 2: Microgeneration and demand-side management, in: *PowerTech, 2011 IEEE Trondheim*, 2011, pp. 1–8. doi:10.1109/PTC.2011.6019384.
- [8] C. Gellings, The concept of demand-side management for electric utilities, *Proceedings of the IEEE* 73 (10) (1985) 1468–1470. doi:10.1109/PROC.1985.13318.
- [9] G. M. Masters, *Renewable and Efficient Electric Power Systems*, Wiley-Interscience, 2004.
- [10] P. Palensky, D. Dietrich, Demand side management: Demand response, intelligent energy systems, and smart loads, *Industrial Informatics, IEEE Transactions on* 7 (3) (Aug.) 381–388. doi:10.1109/TII.2011.2158841.
- [11] G. Strbac, Demand side management: Benefits and challenges, *Energy Policy* 36 (12) (2008) 4419 – 4426, foresight Sustainable Energy Management and the Built Environment Project. doi:http://dx.doi.org/10.1016/j.enpol.2008.09.030. URL <http://www.sciencedirect.com/science/article/pii/S0301421508004606>
- [12] Pacific Northwest GridWise Testbed Demonstration Projects, Part I. Olympic Peninsula Testbed Project (October 2007).
- [13] R. Poli, Analysis of the publications on the applications of particle swarm optimisation, *J. Artif. Evol. App.* 2008 (2008) 4:1–4:10. doi:10.1155/2008/685175. URL <http://dx.doi.org/10.1155/2008/685175>
- [14] S. D. Ramchurn, P. Vytelingum, A. Rogers, N. Jennings, Agent-based control for decentralised demand side management in the smart grid, in: *The 10th International Conference on Autonomous Agents and Multiagent Systems - Volume 1, AAMAS '11*, International Foundation for Autonomous Agents and Multiagent Systems, Richland, SC, 2011, pp. 5–12. URL <http://dl.acm.org/citation.cfm?id=2030470.2030472>
- [15] A. E. Eiben, J. E. Smith, *Introduction to Evolutionary Computing*, Springer Verlag, 2003.
- [16] T. Bäck, D. B. Fogel, Z. Michalewicz (Eds.), *Evolutionary Computation 1: Basic Algorithms and Operators*, IOP Publishing Ltd., Bristol, UK, 1999.
- [17] J. Lohn, G. Hornby, D. Linden, An evolved antenna for deployment on NASA's Space Technology 5 mission, in: U.-M. O'Reilly, T. Yu, R. Riolo, B. Worzel (Eds.), *Genetic Programming Theory and Practice II*, Vol. 8 of Genetic Programming, Springer US, 2005, pp. 301–315.
- [18] E. Galván-López, J. M. Swafford, M. O'Neill, A. Brabazon, Evolving a Ms. PacMan controller using grammatical evolution, in: C. D. Chio, S. Cagnoni, C. Cotta, M. Ebner, A. Ekárt, A. Esparcia-Alcázar, C. K. Goh, J. J. M. Guervós, F. Neri, M. Preuss, J. Togelius, G. N. Yannakakis (Eds.), *EvoApplications (1)*, Vol. 6024 of Lecture Notes in Computer Science, Springer, 2010, pp. 161–170.
- [19] E. Galván-López, D. Fagan, E. Murphy, J. Swafford, A. Agapitos, M. O'Neill, A. Brabazon, Comparing the performance of the evolvable grammatical evolution genotype-phenotype map to grammatical evolution in the dynamic ms. pac-man environment, in: *Evolutionary Computation (CEC)*, 2010 IEEE Congress on, 2010, pp. 1–8. doi:10.1109/CEC.2010.5586508.
- [20] E. Galván-López, R. Poli, C. Coello, Reusing code in genetic programming, in: M. Keijzer, U.-M. O'Reilly, S. Lucas, E. Costa, T. Soule (Eds.), *Genetic Programming*, Vol. 3003 of Lecture Notes in Computer Science, Springer Berlin Heidelberg, 2004, pp. 359–368. doi:10.1007/978-3-540-24650-3_34. URL http://dx.doi.org/10.1007/978-3-540-24650-3_34
- [21] E. Galvan-Lopez, Efficient graph-based genetic programming representation with multiple outputs, *International Journal of Automation and Computing* 5 (1) (2008) 81–89. doi:10.1007/s11633-008-0081-4. URL <http://dx.doi.org/10.1007/s11633-008-0081-4>
- [22] J. R. Koza, Human-competitive results produced by genetic programming, *Genetic Programming and Evolvable Machines* 11 (3-4) (2010) 251–284. doi:10.1007/s10710-010-9112-3. URL <http://dx.doi.org/10.1007/s10710-010-9112-3>
- [23] J. A. P. Lopes, F. J. Soares, P. M. Almeida, M. Moreira da Silva, Smart Charging Strategies for Electric Vehicles: Enhancing Grid Performance and Maximizing the Use of Variable Renewable Energy Resources, in: *Proceedings 24th International Electric Vehicle Symposium and Exposition (EVS24)*, World Electric Vehicle Association (WEVA), World Electric Vehicle Association (WEVA), 2009, pp. 1–11.
- [24] J. Soares, B. Canizes, C. Lobo, Z. Vale, H. Morais, Electric vehicle scenario simulator tool for smart grid operators, *Energies* 5 (6) (2012) 1881–1899. doi:10.3390/en5061881. URL <http://www.mdpi.com/1996-1073/5/6/1881>
- [25] P. Wang, J. Huang, Y. Ding, P. Loh, L. Goel, Demand side load management of smart grids using intelligent trading/metering/ billing system, in: *Power and Energy Society General Meeting, 2010 IEEE*, 2010, pp. 1–6. doi:10.1109/PES.2010.5588163.
- [26] S. J. Gerssen-Gondelach, A. P. Faaij, Performance of batteries for electric vehicles on short and longer term, *Journal of power sources* 212 (2012) 111–129.
- [27] M. Parvania, M. Fotuhi-Firuzabad, M. Shahidehpour, Optimal demand response aggregation in wholesale electricity markets, *Smart Grid, IEEE Transactions on* 4 (4) (2013) 1957–1965.
- [28] D. Chassin, K. Schneider, C. Gerkenmeyer, Gridlab-d: An open-source power systems modeling and simulation environment, in: *Transmission and Distribution Conference and Exposition, 2008. IEEE/PES*, 2008, pp. 1–5. doi:10.1109/TDC.2008.4517260.
- [29] A. J. Wood, B. F. Wollenberg, *Power Generation, Operation, and Control*, 2nd Edition, Wiley-Interscience. URL <http://www.amazon.com/exec/obidos/redirect?tag=citeulike07-20&path=ASIN/0471586994>