

Fair Online Allocation of Perishable Goods and its Application to Electric Vehicle Charging

Enrico H. Gerding¹, Alvaro Perez-Diaz¹, Haris Aziz^{2,3},
Serge Gaspers^{2,3}, Antonia Marcu¹, Nicholas Mattei⁴ and Toby Walsh^{2,3,5}

¹University of Southampton

²UNSW Sydney

³Data61, CSIRO

⁴Tulane University

⁵TU Berlin

eg@ecs.soton.ac.uk

Abstract

We consider mechanisms for the online allocation of perishable resources such as energy or computational power. The main application is electric vehicle charging where agents arrive and leave over time. Unlike previous work, we consider mechanisms without money, and a range of objectives including fairness and efficiency. In doing so, we extend the concept of envy-freeness to online settings. Furthermore, we explore the trade-offs between different objectives and analyse their theoretical properties both in online and offline settings. We then introduce novel online scheduling algorithms and compare them in terms of both their theoretical properties and empirical performance.

1 Introduction

In this paper, we consider an online scheduling problem where a number of agents compete for a perishable resource and a fair allocation needs to be found. A resource is perishable when it cannot be stored and, hence, needs to be allocated at every time step to prevent it from being wasted. Relevant examples include energy for charging electric vehicles (EVs) and processing power for executing computational tasks. We assume that agents (EVs or computational tasks) arrive in an online manner and, on arrival, announce their departure times or execution deadlines. In addition, the agents declare the maximum amount of resource they can consume per time step, as well as the total resource required. The aim is to allocate resources optimally and/or fairly according to various objectives without information about future arrivals. For ease of exposition, we will focus on the EV charging setting, as this has received considerable attention in recent years, although the model and results can be applied to other settings.

These types of scenarios are extensively studied in various fields. Firstly, works that consider scheduling specifically for EV applications include [Mehta *et al.*, 2018; Sun *et al.*, 2018; Liu *et al.*, 2018]. These works are concerned with cost minimisation, but are not interested in fair energy allocation and

propose offline rather than online algorithms. Other works, more in line with this paper, consider online EV scheduling, including [Gerding *et al.*, 2011; Gerding *et al.*, 2016; Stein *et al.*, 2012; Bilh *et al.*, 2018]. However, all of these works consider mechanisms with money instead of fairness. A recent closely related work is [de Weerd *et al.*, 2018]. They also analyse the computational complexity of battery charging algorithms. However, they focus on offline algorithms and monetary payments. Also, they consider binary valuation functions (demand fulfilled/non-fulfilled). Instead, we consider partial fulfilment, as well as fairness, and do not assume any information about the valuations (only the scheduling constraints).

Secondly, the extensive field of scheduling has studied related problems (see [Pinedo, 2006] for an overview). However, to our knowledge, none of the existing models covers the same set of online features as ours; specifically a model with dynamic arrivals and departures; soft completion times, and where each element of capacity is perishable. The area of power balancing and bandwidth allocation [Shi *et al.*, 2014; Shah, 2012] has a similar flavour to our work but focuses on the coordination of multiple decentralised actors instead. Finally, Porter [2004] investigates strategic aspects of maximizing weighted completion in online hard real-time scheduling where tasks have weights, release times, deadlines, and durations. However, in our model, we define metrics around fairness including the proportion of a job that is completed.

Thirdly, another relevant field is online fair division (e.g. [Aleksandrov *et al.*, 2015; Aleksandrov and Walsh, 2017]) that considers the problem of food allocation from food banks. While this literature presents some similarities, such as using an online setting and indivisible goods, our model significantly differs from theirs in that agents dynamically arrive and depart over time.

The fourth and final area considered is that of energy efficient processor scheduling [Albers, 2010; Yao *et al.*, 1995]. Here, a number of jobs need to be processed on a variable-speed processor where each job has a release time, a deadline, and a processing volume. Certain processing power can be allocated to each job, and this needs to be suitably scheduled. In comparison, our model is unique in that it has a multi-

processor flavour, as each EV is independent.

Contributions. We consider a novel setting without monetary payments and where agents do not specify a valuation function but only requirements such as their maximum charging speed, the amount of resource required and a deadline. In doing so, we focus on trade-offs between 3 different objectives: the fairness concept of *envy-freeness*; overall resource utilisation; and the number of satisfied agents (i.e. agents who receive their full demand). In more detail, this paper makes the following contributions. First, we present the first extension of the notion of envy-freeness to online settings. Second, we show that for offline settings there is always an allocation that maximises both resource utilisation and the number of satisfied agents, but neither implies the other. Moreover, the former can be computed in polynomial time, while the latter is NP-hard even in very restricted settings. Third, focusing on online settings, we present two novel online scheduling algorithms which prioritise early allocations. However, we show that neither efficiency objective can always be achieved. Fourth, we present an empirical evaluation in an EV charging scenario which shows that, despite the negative theoretical results, our algorithms perform well in practice compared to standard benchmarks. We also show that Equal Contention, which satisfies envy-freeness, performs very poorly in terms of the other objectives.

2 Model

We consider a setting with n agents $A = \{1, \dots, n\}$ (e.g. EVs) who arrive in an online manner with arrival time $a_i \in T$ and departure time $d_i \in T$ ($d_i \geq a_i$) and who require up to a certain quantity of resource q_i on departure. The value q_i could be seen as the total energy needed to fully charge the battery of an electric vehicle. We assume that both time and quantity are discrete, i.e. $T = \{1, \dots, \tau\}$ and $q_i \in \mathbb{N}^+$. Let $x_{i,t} \in \mathbb{N}^+$ denote the amount of resource allocated to agent i at time t . Moreover, we say an agent is *satisfied* if its allocation is equal to its required quantity, i.e. if $\sum_{t=a_i}^{d_i} x_{i,t} = q_i$. We assume that agents can consume resources at a maximum charging rate of $\bar{r}_i \in \mathbb{N}^+$. However, since no algorithm will allocate more resources than an agent needs, the *effective* maximum charging rate is given by $\bar{r}_{i,t} = \min(\bar{r}_i, q_i - \sum_{t'=a_i}^{t-1} x_{i,t'})$. Finally, at each point in time $t \in T$ a supply $s_t \in \mathbb{N}^+$ of the resource becomes available, which can be allocated only to agents who are in the market, i.e. $A_t = \{i \in A : a_i \leq t \leq d_i\}$. We assume that supply is *perishable* meaning that supply needs to be al-

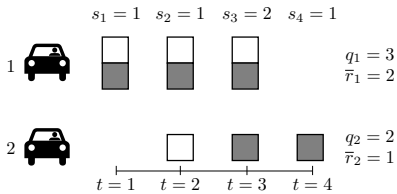


Figure 1: Example allocation. For each EV and time step, each empty square represents one potential unit of energy, while a grey square represents an allocated unit.

located immediately and any unallocated supply is lost. Consequently, we have that $\forall t : \sum_i x_{i,t} \leq s_t$.

This results in the following *problem constraints* for any allocation we consider:

$$\forall i \in A, \forall t < a_i : x_{i,t} = 0 \quad (1)$$

$$\forall i \in A, \forall t > d_i : x_{i,t} = 0 \quad (2)$$

$$\forall i, a_i \leq t \leq d_i : x_{i,t} \leq \bar{r}_i \quad (3)$$

$$\forall i \in A : \sum_{t=a_i}^{d_i} x_{i,t} \leq q_i \quad (4)$$

$$\forall t : \sum_{i \in A} x_{i,t} \leq s_t \quad (5)$$

Example 1. Consider two EVs arriving at a charging station with parameters $a_1 = 1$, $d_1 = 3$ and $a_2 = 2$, $d_2 = 4$ respectively. See Fig. 1. The first vehicle has a maximum charging rate $\bar{r}_1 = 2$ and needs three units of energy, $q_1 = 3$. The second vehicle has a maximum charging rate $\bar{r}_2 = 1$ and needs two units of energy, $q_2 = 2$. The available supply to the charging station is one unit at time steps 1, 2, 4, and two units at time step 3. We give an example allocation that fully satisfies both EVs by allocating $x_{1,1} = x_{1,2} = x_{1,3} = 1$ and $x_{2,3} = x_{2,4} = 1$, and zero otherwise.

3 Objectives

This work is motivated by the fact that agent preferences are often not easily obtainable in practice, e.g. because of the additional burden on the user and/or the difficult for users to express their willingness to pay. Hence, although it is reasonable to assume the agent constraints (a_i , d_i , q_i and \bar{r}_i) can be elicited with relatively little effort, we would like to avoid assumptions about the utility functions, and hence objectives such as social welfare (sum of utilities) maximisation are no longer applicable. Instead, we consider other objectives a system designer may reasonably want to achieve in our setting. In so doing, our goal is to study their computational complexity, as well as the theoretical and empirical trade offs between the various objectives. Specifically, we consider the following set of objectives:

MaxDelivered: Maximise the total resource allocated:

$$\sum_{i \in A} \sum_{t=a_i}^{d_i} x_{i,t}.$$

MaxSatisfied: Maximise the number of agents satisfied:

$$\sum_{i \in A} \{ \sum_{t=a_i}^{d_i} x_{i,t} = q_i \}.$$

Envy-Freeness: We say agent i is envious of agent j 's allocation if $\min(q_i, \sum_{t=a_i}^{d_i} \min(\bar{r}_i, x_{j,t})) > \sum_{t=a_i}^{d_i} x_{i,t}$. In words, the allocation to agent j while agent i is in the market is greater than the allocation i receives, subject to i 's maximum charging rate and total demand. We say that an allocation is *envy-free* if no agent is envious of another agent's allocation.

The first objective is arguably the closest to the common social welfare maximisation objective and, indeed, is equivalent if agents have the same preferences and these are linear in the number of units received. The second objective is reasonable in cases where agents have some complementary preferences, e.g. in the EV setting a driver will need a minimum charge to

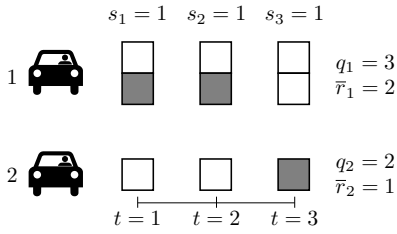


Figure 2: Example showing that MaxDelivered does not imply MaxSatisfied.

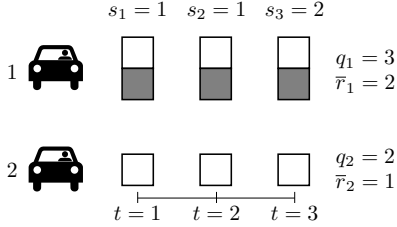


Figure 3: Example showing that MaxSatisfied does not imply MaxDelivered.

complete a day’s journeys and it is also reasonable to expect that users would want their vehicle fully charged by the time they need it. Similarly, in a cloud computing setting, partial completion of a computational task may not be feasible. The third objective is about allocation fairness, where we build on the established notion of envy-freeness, which requires that no agent should prefer another agent’s allocation. Our definition is the first generalisation to online settings where agents may be in the markets at different times (and so it is not considered unfair when an agent is receiving more resources while another agent is not present). We note that, although our definition is less restrictive than the original offline one, it would be interesting to study even less restrictive extensions, and we briefly return to this in our future work discussion.

Note that, in general, not all objectives can be satisfied simultaneously, and none are equivalent. We explore this in more detail in the remainder of this paper and present now an illustrative example.

Example 2. In this example, we show that MaxDelivered and MaxSatisfied are not equivalent and neither implies the other. First, consider the setup depicted in Fig. 2. The proposed allocation satisfies MaxDelivered, as all the available supply is allocated, but fails to satisfy any of the two EVs. Alternatively, if the unit of energy allocated to EV 2 at $t = 3$ was instead allocated to EV 1, we would have both MaxDelivered and MaxSatisfied.

Secondly, consider the setup depicted in Fig. 3. In this case, the proposed allocation verifies MaxSatisfied, given that it satisfies EV 1 and no other allocation could possibly satisfy the two EVs given the available supply. However, this allocation is not MaxDelivered, as there is an unused unit of supply at time $t = 3$.

4 Offline Algorithms

We start by considering the complexity of solving the offline problem (i.e. where we have complete foresight of the agents arriving in the future). Specifically, for the MaxSatisfied objective, this is generally hard, but it can be computed in polynomial time when the supply is a single unit per timestep:

Theorem 1. *Computing a MaxSatisfied allocation is NP-hard, even when $s_t = 2$ at every time step, and we have $\forall i : \bar{r}_i = 1$.*

Proof. We reduce from the scheduling problem $P2|pmtn, r_i|\sum U_i$, which is NP-hard [Du *et al.*, 1992]. This is the scheduling problem with two identical machines, tasks can be preempted, each task has its own release time and deadline, and the objective is to minimize the number of tasks for which the deadline is not met. This corresponds to our setting where we have one agent for each task whose arrival time equals the task’s release time, whose departure time equals the task’s deadline and whose charging requirement q_i equals the task’s processing time. We also have that $s_t = 2$ at every time step t , and the charging rates are $r_i \in \{0, 1\}$ for every agent i . Finally, T equals the largest deadline. \square

Theorem 2. *A MaxSatisfied allocation can be computed in polynomial time when $s_t = 1$ at every time step, and we have $\forall i : \bar{r}_i = 1$.*

Proof. The problem corresponds to the classic scheduling problem $1|pmtn, r_i|\sum U_i$; the case where we have a single machine, tasks can be preempted, there are release times and deadlines for every task, and we want to minimize the unit penalty which is $U_i := 0$ if i is completed after d_i and 1 otherwise. This problem can be solved in polynomial time in the number of agents using an algorithm from [Lawler, 1990]. \square

Interestingly, the MaxDelivered objective is easier computationally:

Theorem 3. *A MaxDelivered allocation can be computed in polynomial time.*

Proof. We present a polynomial-time reduction to that of computing a feasible flow of a flow network with upper capacity constraints.

We build a network flow with nodes as follows:

$$\{s, t\} \cup A \cup \{v_t \mid t \in T\} \cup \{a_i^t \mid t \in T, i \in A\}.$$

The arcs between the nodes are as follows.

- Source s points to the n agents.
- Each agent node $a_i \in A$ points to each node a_i^t for all time slots $t \in T$.
- A node a_i^t only points to node v_t corresponding to time slot t if agent a_i is active in time slot t .
- Each node v_t points to sink t .

The only capacities of the nodes are as follows.

- Each node a_i has upper capacity q_i .
- Each node v_t has upper capacity s_t .

We compute a maximum flow of the network. We can read off an actual schedule by looking at the corresponding feasible flow: if an agent node a_i send some flow f through a vertex v_t , then a_i gets f units of electricity in time slot t . \square

We now consider the tradeoffs between MaxDelivered and MaxSatisfied. Although a MaxDelivered allocation process need not optimize MaxSatisfied, as shown in Fig. 2, we now prove that it is always possible to modify a MaxSatisfied schedule to also be MaxDelivered, and so achieve both objectives simultaneously.

Proposition 1. *There always exists a solution which maximizes both MaxDelivered and MaxSatisfied.*

Proof. To find a schedule that is both MaxSatisfied and MaxDelivered we first find a MaxSatisfied schedule (e.g. using our linear programming formulation). We can then take this schedule, which may not allocate all units of charge s_t at each time step, and augment it by assigning any unused charge to any available agent not covered in the returned MaxSatisfied schedule. If no such agent exists, then we are allocating as much charge in each time step as possible and the allocation is MaxDelivered. \square

We note that this result critically depends on the foresight given by the offline setting. In an online setting, it will be impossible for any algorithm to always maximise MaxDelivered or MaxSatisfied, as we prove in the next section.

Finally, in terms of the envy-freeness objective, a solution can be trivially obtained in polynomial time (both in online and offline settings) by an *equal contention* algorithm, i.e. an algorithm which, subject to charging constraints, allocates everyone the same amount at each time step. Formally:

Definition 1 (Equal Contention (EC)). *A mechanism satisfies equal contention if, at every time step, the charging rate of any agent in the market which is below their maximum effective rate, is at least as fast as that of any other agent in the market. Formally, let $\bar{r}_{i,t}$ be the effective maximum charging rate (as defined in Section 2). Then, $\forall t, i \in A_t$, whenever $x_{i,t} < \bar{r}_{i,t}$ we have that $\max_{i' \in A_t} x_{i',t} = x_{i,t}$.*

5 Online Algorithms

We now consider online algorithms, where, at every time step, the algorithm only has knowledge of the agents arrived so far and has no information about future arrivals. We will assume that future supply is known. We start by adapting the following well known online scheduling algorithms to our setting: earliest deadline first, least laxity first and value density. In these algorithms, at each time t , the agents are prioritised based on different criteria, and the agent $i \in A_t$ with the highest priority is allocated resources at the maximum possible rate, i.e. $x_{i,t} = \min(s_t, \bar{r}_{i,t})$. If there is still supply remaining (i.e. if $s_t - x_{i,t} > 0$), the next agent in the priority list is allocated resources in the same manner until no more supply or agents are available. Ties can be broken in any way desirable. Where unspecified, in our empirical evaluation (Section 6) we use a random tie breaking rule.

- **Earliest Deadline First (EDF):** Prioritise agents based on their deadline, with earlier deadline having higher priority. In our implementation any ties are resolved using arrival time (earlier is better) followed by random.
- **Least Laxity First (LLF):** Let τ_i^{\max} denote the latest possible time that agent i can be scheduled any units in order to fully satisfy its demand, given its current the effective maximum charging rate and the available supply over time. Allocate the agents with the lowest τ_i^{\max} first. If there are agents who cannot be fully satisfied they get top priority and least satisfied agents are prioritised first. Ties are broken based on arrival time.
- **Value Density (VD):** Prioritise agents by *density* defined as: $\frac{q_i - \sum_{t' < t} x_{i,t'}}{(d_i - t + 1) \cdot \bar{r}_i}$. Ties are broken according to EDF.

These algorithms are simple as they are very short-sighted. Moreover, they do not directly optimise our objectives of MaxDelivered and MaxSatisfied. Therefore, we now introduce novel algorithms which are based on their offline counterparts, but without considering future arrivals. That is, an optimal schedule is computed based on the agents currently in the market, assuming no new agents arrive. The schedule is then recomputed at each timestep, considering any new agents who have arrived (noting that past allocations cannot be revoked). A challenge is that, typically, many such schedules exist and a random tie breaking rule turns out to perform poorly empirically. Therefore, our algorithm prioritises schedules where more resources are allocated to the agents earlier on in the schedule. Formally, for MaxDelivered the algorithm is as follows:

Definition 2 (OnlineMaxDelivered (OMDel)). *Let t be a current time step and A_t the agents currently in the market. The algorithm proceeds in two steps. First, find the maximum number of units, \bar{x}_t , we can allocate at the current time, t , i.e. find $\bar{x}_t = \max \sum_{i \in A_t} x_{i,t}$ subject to the problem constraints. Then, an online MaxDelivered solution is found by maximising $\sum_{i \in A_t} \sum_{t'=a_i}^{d_i} x_{i,t'}$ subject to the problem constraints and $\sum_{i \in A_t} x_{i,t} = \bar{x}_t$.*

Note that, first, the additional constraint of allocating \bar{x}_t units at time t does not reduce the optimality of the solution. That is, OnlineMaxDelivered will allocate the same number of units regardless of the constraint (we leave out a formal proof due to space). Second, imposing similar constraints at later timesteps is superfluous since any future constraints do not influence the choice at time t . Finally, it is easy to see that the additional constraint does not affect the computational time:

Theorem 4. *OnlineMaxDelivered runs in polynomial time.*

Proof. The argument is similar as that of Theorem 3 except that, since we are in an online setting and the allocation in previous time steps is irrevocable, we enforce the prior history of the allocation by having lower capacity constraints for the a_i^t nodes representing the agents for a particular time step. \square

We can similarly define an algorithm for MaxSatisfied. Due to Proposition 1 we can always achieve an allocation

which maximises both MaxDelivered and MaxSatisfied, and this can be achieved by combining both criteria as follows:

Definition 3 (OnlineMaxSatisfied (OMSat)). *Find \bar{x}_t as in Definition 2. Then, an online MaxSatisfied solution is found by maximising both MaxDelivered as well as MaxSatisfied, i.e. $\sum_{i \in A_t} \left(\sum_{t'=a_i}^{d_i} x_{i,t'} + \{ \sum_{t=a_i}^{d_i} x_{i,t} = q_i \} \right)$ subject to the problem constraints and $\sum_{i \in A_t} x_{i,t} = \bar{x}_t$.*

Although this algorithm satisfies both objectives, it can be trivially shown that its computational hardness is the same as that of solving MaxSatisfied offline (see Section 4).

Finally, we show that, unfortunately, even though these algorithms look ahead, none of the algorithms are always optimal in online settings. This is surprising, especially for MaxDelivered.

Theorem 5. *No online mechanism exists which always optimises MaxDelivered.*

Proof. The proof is by counterexample. Consider a setting with initially 2 agents entering the market at time $t = 1$, where $\bar{r}_1 = 1$, $q_1 = 2$ and $\bar{r}_2 = 2$, $q_2 = 2$. Agent 1 is in the market for 4 timesteps, and agent 2 for only 2 timesteps ($d_1 = 4$, $d_2 = 2$). Supply is 2 units per time step.

We consider two options. In option 1, the algorithm allocates 0 units to agent 1 and 2 units to agent 2. In option 2, the algorithm allocates 1 unit to both. It is easy to see that any other option (i.e. which does not allocate the full supply in the first round) is suboptimal.

If option 2 is chosen, suppose that a third agent enters the market at $t = 2$ with $\bar{r}_3 = 2$, $q_3 = 2$, $d_3 = 2$ (i.e. the agent is in the market for 1 timestep and requires 2 units). The maximum total units allocated is 5. However, if option 1 was chosen, 6 units could have been allocated in total, showing that option 2 is not optimal.

If option 1 is chosen, suppose that 2 agents enter the market at times $t = 3$ and $t = 4$ with $\bar{r}_3 = \bar{r}_4 = 2$, $q_3 = q_4 = 2$, $d_3 = 3$, $d_4 = 4$ (they stay in the market for 1 timestep each). At timestep 2 agent 2 requires no more resources, so the only option is to allocate 1 unit to agent 1. Then, at timesteps 3 and 4 at most 4 units can be allocated, making the total delivery 7. However, if option 2 is chosen, at timestep 2 we give 1 unit to each. The total delivery after 4 timesteps is then 8.

Hence, neither option is always optimal, showing that no online algorithm can always optimise MaxDelivered. \square

Theorem 6. *No online mechanism exists which always optimises MaxSatisfied.*

Proof. The proof is by counterexample. Consider the setting detailed in the proof of Theorem 5. If option 2 is chosen and, following the previous proof, a new agent arrives at $t = 2$ with $\bar{r}_3 = 2$, $q_3 = 2$, $d_3 = 2$, then only two agents would be satisfied. It is easy to see that choosing option 1 would satisfy all three agents. Similarly, if option 1 is chosen and two agents enter the market at times $t = 3$ and $t = 4$ with $\bar{r}_3 = \bar{r}_4 = 2$, $q_3 = q_4 = 2$, $d_3 = 3$, $d_4 = 4$, only three agents would be satisfied. Again, choosing option 2 would satisfy all four agents. Hence, neither option is always optimal, showing that no online algorithm can always optimise MaxSatisfied. \square

6 Empirical Evaluation

Despite the negative theoretical results for online settings, the algorithms may perform well in practice. Hence, in this section, we compare the performance of the proposed algorithms in a realistic simulation. We will start by detailing the setup considered, and then proceed to describe the results.

We consider a 24-hour period, i.e. $T = \{0, 1, \dots, 23\}$, each time unit representing one hour. Specifically, we consider a night-time charging scenario, in which EVs arrive to the charging station in the evening and depart in the morning. For every considered day, random EVs and supply conditions are generated, as detailed in the next sections. Also, we run 50 instances of each given simulation and present averaged results. Note that all the data from this empirical evaluation is publicly available [Perez-Diaz *et al.*, 2019].

6.1 Supply Settings

Each supply unit represents 3 kWh [Binetti *et al.*, 2015]. In terms of the available supply, high availability is characteristic of night hours. In more detail, Robu *et al.* [2013] identify the average available energy in a small neighbourhood to be 615 kWh between 2:00 and 6:00, corresponding to 5-12 units/time step. On the contrary, reduced supply is available during the rest of the day, corresponding to 99 kWh, hence 0-4 units/time step. Thus, the supply available at time t , s_t , is drawn from a Gaussian distribution with parameters specified in Table 1, both for high and low supply intervals. Specifically, mean, standard deviation, and minimum and maximum limits are presented for each.

6.2 EV Settings

Similarly to the available supply, the characteristics of every EV are drawn independently from a Gaussian distribution with the parameters specified in Table 1.

Robu *et al.* [2013] reported that the majority of agents arrive in the time interval 16:00-19:00 (peak at 17:00). With regards to the standard deviation of this arrival time, values such as 1.5 and 3.13 have been reported [Benetti *et al.*, 2015]. Similarly, Robu *et al.* [2013] find that EV departures peak between 8:00 and 10:00. Thus, most EVs are plugged-in for at most 15h every night. Average and standard deviations are set to 85% and 5% of the online time limit, respectively.

With respect to battery capacity, it is assumed to be approximately 25 kWh, which corresponds to 9 units of supply. In terms of required charge, it is assumed that the battery's health is taken into account by the EV owner. Specifically,

	min	average	std	max
high supply	1	10.5	2.25	15
low supply	1	5.25	6	6
arrival	0	17	2.3	23
online time	1	12.75	0.75	remaining hours
charging rate	1	2.5	1	5
quantity	1	5.5	1.5	10

Table 1: Limits, averages and standard deviations of the Gaussian distributions of each of the scenario's stochastic variables.

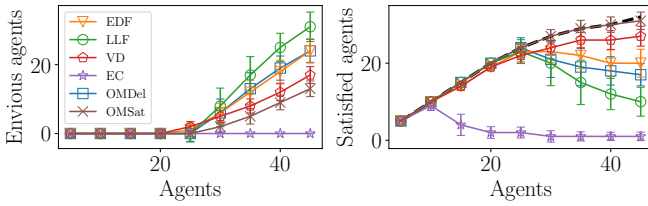


Figure 4: Number of envious agents (left) and number of satisfied agents (right) when varying the number of agents present in the simulation. The stochastic available supply units follow the Gaussian distribution specified in Table 1. The dashed line corresponds to the optimal offline MaxSatisfied allocation.

Neaimah *et al.* [2015] report a minimum charge value of 39%, with a median of 53%, and a maximum value of 68%.

Finally, the maximum charging rate of an EV depends on the vehicle itself and on the charging infrastructure. Given that we consider a night-time charging scenario, presumably in residential areas, we consider that the EVs do not have access to industrial-grade charging speeds. Typical charging speeds in this type of scenario are around 3-6 kW and 11 kW [Heydarian-Forushani *et al.*, 2016].

6.3 Results

We now compare how each of the considered algorithms performs with respect to the considered objectives. Fig. 4 shows the number of envious agents (as per our envy-freeness definition, see Section 3) and the number of satisfied agents in a setting where we vary the total number of agents. We can see that, as the number of agents grows, it becomes more difficult to keep them envy-free and to satisfy their requirements. In fact, the number of satisfied agents drops as more agents are in the market for most algorithms, since more agents only get partially satisfied. Moreover, even though EqualContention is the only envy-free algorithm, it is extremely inefficient and scores very low on agent satisfaction. Interestingly, while LLF is the worst in terms of both measures, EDF and OnlineMaxDelivered are comparable, and Value Density is the best of the simple algorithms and performs surprisingly well. OnlineMaxSatisfied presents the highest efficiency in terms of the number of satisfied agents, achieving over 96% efficiency in all cases, and the best envy-freeness after EqualContention, but incurs more costly computation. Furthermore, all the algorithms perform similarly with respect to MaxDelivered, achieving between 95 and 100% of the offline optimal result, apart from EqualContention, which is only 13% optimal on average (graphs omitted).

These trends are similar when we vary the amount of available supply, as depicted in Fig. 5. We can see that, as expected, overall results improve as the available supply grows, since there is less competition for the resources. Again, EqualContention is very inefficient, and OnlineMaxSatisfied presents, on the whole, the best results. Results with respect to MaxDelivered are also similar to before.

7 Conclusions and Future Work

In this paper we study online allocations of perishable resources (*e.g.* energy or computational power) in a setting

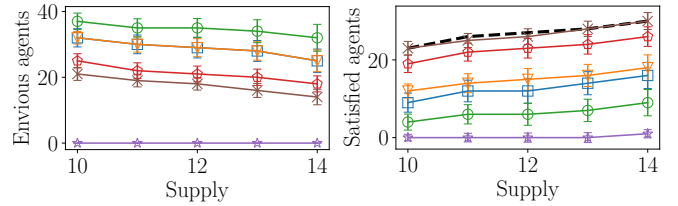


Figure 5: Number of envious agents (left) and number of satisfied agents (right) when varying the amount of supply available at each time step. Total number of agents is 45. The dashed line corresponds to the optimal offline MaxSatisfied allocation, averaged over 20 instances.

without money and where agents dynamically enter and leave the market. In doing so, we consider efficiency objectives (MaxDelivered and MaxSatisfied) as well as fairness (envy-freeness). Specifically, we provide computational complexity results as well as the trade-offs between the objectives. We also introduce novel online algorithms, called OnlineMaxDelivered and OnlineMaxSatisfied, and prove that their computational complexity is polynomial and NP-hard, respectively. Moreover, we prove that no algorithm exists which can always guarantee a solution satisfying either efficiency objective in online settings. Also, we extend the definition of envy-freeness to online settings, and show that an envy-free allocation can be found in polynomial time using an equal contention algorithm, but this algorithm performs poorly in terms of the other objectives. Finally, our empirical results show that the proposed OnlineMaxSatisfied algorithm achieves the best overall performance across the considered objectives.

An interesting direction for further research is to consider additional objectives, such as truthfulness and other notions of fairness. In more detail, different from existing work in mechanism design, in our settings agents do not need to report valuation functions, but only constraints. Nevertheless, agents can still misreport these. For example, it is already known that the earliest deadline first algorithm is not incentive compatible since it is sometimes beneficial to report an earlier deadline. In future work, we will analyse which algorithms are truthful. Furthermore, we would like to study other algorithms besides equal contention that meet online envy-freeness. Preliminary work indicates that, for agents arriving at the same time, envy-freeness can only be guaranteed by applying equal contention between those agents. However, for agents arriving at different times, other variants can be considered. Moreover, it would be interesting to consider approximately envy-free solutions.

Acknowledgements

A. P.-D. acknowledges financial support from an EPSRC Doctoral Training Centre grant (EP/L015382/1). H. A. is supported by a UNSW Scientia Fellowship. S. G. is the recipient of an Australian Research Council (ARC) Future Fellowship (FT140100048) and acknowledges support under the ARC’s Discovery Projects funding scheme (DP150101134). T. W. is funded by the European Research Council under the Horizon 2020 Programme via AMPLify 670077.

References

- [Albers, 2010] Susanne Albers. Energy-efficient algorithms. *Communications of the ACM*, 53(5):86–96, 2010.
- [Aleksandrov and Walsh, 2017] Martin Aleksandrov and Toby Walsh. Pure Nash equilibria in online fair division. *International Joint Conference on Artificial Intelligence*, pages 42–48, 2017.
- [Aleksandrov et al., 2015] Martin Aleksandrov, Haris Aziz, Serge Gaspers, and Toby Walsh. Online Fair Division: Analysing a Food Bank problem. *International Joint Conference on Artificial Intelligence*, pages 2540–2546, 2015.
- [Benetti et al., 2015] Guido Benetti, Maurizio Delfanti, Tullio Facchinetti, Davide Falabretti, and Marco Merlo. Real-Time Modeling and Control of Electric Vehicles Charging Processes. *IEEE Transactions on Smart Grid*, 6(3):1375–1385, 2015.
- [Bilh et al., 2018] Abdoulmenim Bilh, Kshirasagar Naik, and Ramadan El-Shatshat. A Novel Online Charging Algorithm for Electric Vehicles Under Stochastic Net-Load. *IEEE Transactions on Smart Grid*, 9(3):1787–1799, 2018.
- [Binetti et al., 2015] Giulio Binetti, Aali Davoudi, David Naso, Biagio Turchiano, and Frank L. Lewis. Scalable Real-Time Electric Vehicles Charging With Discrete Charging Rates. *IEEE Transactions on Smart Grid*, 6(5):2211–2220, 2015.
- [de Weerd et al., 2018] Mathijs de Weerd, Michael Albert, and Vincent Conitzer. Complexity of Scheduling Charging in the Smart Grid. *Proc. of the 17th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2018)*, pages 1924–1926, 2018.
- [Du et al., 1992] J. Du, Joseph Y.-T. Leung, and C. S. Wong. Minimizing the number of late jobs with release time constraint. *Journal of Combinatorial Mathematics and Combinatorial Computing*, 11:97–107, 1992.
- [Gerding et al., 2011] Enrico H. Gerding, Valentin Robu, Sebastian Stein, and D. Parkes. Online mechanism design for electric vehicle charging. *Proceedings of the 10th Int. Conf. on Autonomous Agents and Multiagent Systems (AAMAS 2011)*, (Aamas):811–818, 2011.
- [Gerding et al., 2016] Enrico H. Gerding, Sebastian Stein, Sofia Ceppi, and Valentin Robu. Online mechanism design for vehicle-to-grid car parks. *International Joint Conference on Artificial Intelligence*, 2016-Janua:286–293, 2016.
- [Heydarian-Forushani et al., 2016] Ehsan Heydarian-Forushani, Mohamad E. H. Golshan, and Miadreza Shafie-khah. Flexible interaction of plug-in electric vehicle parking lots for efficient wind integration. *Applied Energy*, 179:338–349, 2016.
- [Lawler, 1990] Eugene L. Lawler. A dynamic programming algorithm for preemptive scheduling of a single machine to minimize the number of late jobs. *Annals of Operations Research*, 26(1):125–133, 1990.
- [Liu et al., 2018] Zhaoxi Liu, Qiuwei Wu, Shaojun Huang, Lingfeng Wang, Mohammad Shahidehpour, and Yusheng Xue. Optimal Day-ahead Charging Scheduling of Electric Vehicles through an Aggregative Game Model. *IEEE Transactions on Smart Grid*, 9(5):5173–5184, 2018.
- [Mehta et al., 2018] Rahul Mehta, Dipti Srinivasan, Ashwin M. Khambadkone, Jing Yang, and Anupam Trivedi. Smart charging strategies for optimal integration of plug-in electric vehicles within existing distribution system infrastructure. *IEEE Transactions on Smart Grid*, 9(1):299–312, 2018.
- [Neaimeh et al., 2015] Myriam Neaimeh, Robin Wardle, Andrew M Jenkins, Jialiang Yi, Graeme Hill, Pdraig F Lyons, Yvonne Hübner, Phil T Blythe, and Phil C Taylor. A probabilistic approach to combining smart meter and electric vehicle charging data to investigate distribution network impacts. *Applied Energy*, 157:688–698, 2015.
- [Perez-Diaz et al., 2019] Alvaro Perez-Diaz, Antonia Marcu, Enrico H. Gerding, Haris Aziz, Serge Gaspers, Nicholas Mattei, and Toby Walsh. Dataset for “Fair Online Allocation of Perishable Goods and its Application to Electric Vehicle Charging”, <https://eprints.soton.ac.uk/430960/>, 2019. Last accessed on 04/06/2019.
- [Pinedo, 2006] Michael L. Pinedo. *Scheduling: Theory, Algorithms, and Systems*. Springer, 5th edition, 2006.
- [Porter, 2004] R Porter. Mechanism design for online real-time scheduling. In *Proceedings of the 5th ACM conference on Electronic commerce*, pages 61–70, 2004.
- [Robu et al., 2013] Valentin Robu, Enrico H. Gerding, Sebastian Stein, David C. Parkes, Alex Rogers, and Nicholas R. Jennings. An Online Mechanism for Multi-unit Demand and its Application to Plug-in Hybrid Electric Vehicle Charging. *J. Artif. Int. Res.*, 48(1):175–230, 2013.
- [Shah, 2012] Ashin D. Shah. *On Coordinating Electricity Markets: Smart Power Scheduling for Demand Side Management and Economic Dispatch*. PhD thesis, Harvard University, 2012.
- [Shi et al., 2014] Huaizhou Shi, Venkatesha R. Prasad, Ertan Onur, and I.G.M.M Niemegeers. Fairness in wireless networks: Issues, measures and challenges. *IEEE Communications Surveys*, 16(1):5–24, 2014.
- [Stein et al., 2012] Sebastian Stein, Enrico H. Gerding, Valentin Robu, and Nicholas R. Jennings. A Model-Based Online Mechanism with Pre-Commitment and its Application to Electric Vehicle Charging. In *Proceedings of the 11th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2012)*, pages 4–8, 2012.
- [Sun et al., 2018] Bo Sun, Zhe Huang, Xiaoqi Tan, and Danny H.K. Tsang. Optimal Scheduling for Electric Vehicle Charging with Discrete Charging Levels in Distribution Grid. *IEEE Transactions on Smart Grid*, 9(2):624–634, 2018.
- [Yao et al., 1995] Frances Yao, Alan Demers, and Scott J. Shenker. A scheduling model for reduced CPU energy. In *Proceedings of IEEE 36th Annual Foundations of Computer Science*, pages 374–382, 1995.