Optimal resource allocation strategies for electric vehicles in smart grids

Bahram Alinia

Doctor of Philosophy (PhD) Thesis
Institut Télécom SudParis

Specialization

Computer Networks

presented by

Bahram Alinia

Submitted for the partial requirement of

Doctor of Philosophy
from
Institut Télécom SudParis

Optimal Resource Allocation Strategies for Electric Vehicles in Smart Grids

Thesis number: 2018TELE0012

10^{th} July 2018

Commitee:

Lila Boukhatem	Reviewer	Assistant Professor , Université Paris-Sud - France
Nadia Maïzi	Reviewer	Professor, MINES ParisTech - France
Maurice Gagnaire	Examiner	Professor, Télécom Paris-Tech - France
Guy Pujolle	Examiner	Professor, Pierre et Marie Curie University (Paris 6) - France
Mehdi Mani	Examiner	PhD, Itron - France
Noël Crespi	Thesis Advisor	Professor, Télécom SudParis - France

Thèse de Doctorat (PhD) de l'Institut Télécom SudParis

Spécialité

RÉSEAUX INFORMATIQUES

présentée par

Bahram Alinia

pour obtenir le grade de

Docteur de l'Institut Télécom SudParis

Stratégies Optimales d'Allocation des Ressources pour les Véhicules Electriques dans les Smart Grids

Numéro de thèse : 2018TELE0012

10 Juillet 2018

Jury composé de :

Lila Boukhatem	Rapporteur	Maitre de conférences , Université Paris-Sud - France
Nadia Maïzi	Rapporteur	Professor, MINES ParisTech - France
Maurice Gagnaire	Examinateur	Professor, Télécom Paris-Tech - France
Guy Pujolle	Examinateur	Professor, Pierre et Marie Curie University (Paris 6) - France
Mehdi Mani	Examinateur	PhD, Itron - France
Noël Crespi	Directeur	Professor, Télécom SudParis - France

Dedication

This thesis is dedicated to my lovely Sana and my parents.

Acknowledgements

I would like to thank my PhD advisor Prof. Noël Crespi for his guidance and support during my PhD. He always inspire me to become a better researcher. I would also like to thank my thesis committee members Prof. Lila Boukhatem, Prof. Nadia Maïzi, Prof. Maurice Gagnaire, Prof. Guy Pujolle, and Dr. Mehdi Mani. Their comments and feedback are invaluable and improves the quality of this thesis. I am grateful to RS2M department and the administrative staff of Telecom SudParis. I would like to thank my co-authors for their contribution and collaboration. I also thank all my colleagues and friends in service architecture lab.

I am forever indebted to my family, academic advisors and friends for their constant support and encouragement. Finally, I would like to thank my family and my parents for their endless love and support.

Bahram Alinia
10^{th} July 2018

Abstract

With the increased environmental concerns related to carbon emission, and rapid drop in battery prices (e.g., 35% drop in 2017), the market share of Electric Vehicles (EVs) is rapidly growing. It is predicted that 2020s will be the decade of EV and the global EV market share will be up to 10% and 20% by 2020 and 2030, respectively. The growing number of EVs along with the unprecedented advances in battery capacity and technology results in drastic increase in the total energy demand of EVs. This large charging demand makes the EV charging scheduling problem challenging. An apparent challenge is that even with taking the advantage of deferrable property of charging demands and performing proper scheduling, the aggregate demand might be beyond the tolerable charging rate of the station, given physical constraints of charger devices and transformers. The critical challenge is the need for online solution design since in practical scenario the scheduler has no information of future arrivals of EVs in a time-coupled underlying problem. This thesis studies online EV scheduling problem and provides three main contributions.

First, we demonstrate that the classical problem of online scheduling of deadline-sensitive jobs with partial values is similar to the EV scheduling problem and study the extension to EV charging scheduling by taking into account the processing rate limit of jobs as an additional constraint to the original problem. The problem lies in the category of time-coupled online scheduling problems without availability of future information. Using competitive ratio, as a well-established performance metric, two online algorithms, both of which are shown to be $(2 - \frac{1}{U})$-competitive are proposed, where U is the maximum scarcity level, a parameter that indicates demand-to-supply ratio. The first proposed algorithm is deterministic, whereas the second is randomized and enjoys a lower computational complexity. The performance of both algorithm matches the state-of-the-art as U grows large. Nonetheless in realistic cases, where U is typically small, they achieve a much lower competitive ratio. To carry out the competitive analysis of our algorithms, we present a proof technique, which is novel to the best of our knowledge. This technique may be used to simplify the competitive analysis of some existing algorithms, and thus could be of independent interest.

Second, we formulate a social welfare maximization problem for EV charging scheduling with charging capacity constraint. Even though the underlying problem is linear, it is difficult to tackle since the input to the problem, i.e., type of EVs, reveals in online fashion. We devise charging scheduling algorithms that not only work in online scenario, but also they address the following two key challenges: (i) to provide on-arrival commitment; respecting the capacity constraint may hinder fulfilling charging requirement of deadline-constrained EVs entirely. Therefore, committing a guaranteed charging amount upon arrival of each EV is highly required; (ii) to guarantee (group)-strategy-proofness as a salient feature to promote EVs to reveal their true type and do not collude with other EVs. Extensive simulations using real-world traces demonstrate the effectiveness of our online scheduling algorithms as compared to the optimal non-committed offline solution.

Third, we tackle online scheduling of EVs in an adaptive charging network (ACN) with local and global peak constraints. Given the aggregate charging demand of the EVs and the peak constraints of the ACN, it might be infeasible to fully charge all the EVs according to their charging demand. Two alternatives in such resource-limited scenarios are to maximize the social welfare by partially charging the EVs (fractional model) or selecting a subset of EVs and fully charge them (integral model). For the fractional model, both offline and online algorithms are devised. We prove that the offline algorithm is optimal. We prove the online algorithm achieves a competitive ratio of 2. The integral model, however, is more challenging since the underlying problem is NP-hard due to 0/1 selection criteria of EVs. Hence, efficient solution design is challenging even in offline setting. We devise a low-complexity primal-dual scheduling algorithm that achieves a bounded approximation ratio. Built upon the offline approximate algorithm, we propose an online algorithm and analyze its competitive ratio in special cases. Extensive trace-driven experimental results show that the performance of the proposed online algorithms is close to the offline optimum, and outperform the existing solutions.

Keywords

Smart grids, electric vehicles, resource allocation, approximation algorithms, competitive analysis

Résumé

Avec les préoccupations environnementales croissantes liées aux émissions de carbone et la chute rapide des prix des batteries, la part de marché des véhicules électriques (EV) augmente rapidement. Le nombre croissant de EV ainsi que les progrès sans précédent dans la capacité de la batterie et de la technologie entraîne une augmentation drastique de la demande totale d'énergie destinée aux véhicules électriques. Cette forte demande de charge rend complexe le problème de planification de la charge. Même en prenant avantage de la propriété reportable des demandes de charge et d'une planification adéquate, la demande globale pourrait dépasser le taux de charge tolérable des stations, étant donné les contraintes physiques des dispositifs de charge et des transformateurs. Le principal défi est la nécessité de concevoir des solutions en ligne puisque, dans la pratique, l'ordonnanceur ne dispose d'aucune information sur les arrivées futures d'EV. Cette thèse étudie le problème d'ordonnancement des EV en ligne et fournit trois contributions principales.

Premièrement, nous démontrons que le problème classique de la programmation en ligne des tâches sensibles aux échéances avec des valeurs partielles est similaire au problème d'ordonnancement EV et étudions l'extension de la programmation des charges EV en prenant en compte de la limite de traitement des travaux. Le problème réside dans la catégorie des problèmes d'ordonnancement en ligne couplés dans le temps sans disponibilité d'informations futures. Le premier algorithme proposé est déterministe, tandis que le second est randomisé et bénéficie d'une complexité de calcul plus faible.

Deuxièmement, nous formulons un problème de maximisation du bien-être social pour la planification de la charge des EV avec une contrainte de capacité de charge. Nous avons conçu des algorithmes d'ordonnancement de charge qui non seulement fonctionnent dans un scénario en ligne, mais aussi qui répondent aux deux principaux défis suivants : (i) fournir un engagement á l'arrivée ; (ii) garantir la résistance aux stratégies (de groupe). Des simulations approfondies utilisant des traces réelles démontrent l'efficacité de nos algorithmes d'ordonnancement en ligne par rapport á la solution hors-ligne optimale non-engagée.

La troisième contribution concerne la planification en ligne des véhicules électriques dans un réseau de recharge adaptatif (ACN) avec des contraintes de pics locaux et globaux. Nous avons conçu un algorithme d'ordonnancement primal-dual de faible complexité qui atteint un rapport d'approximation borné. Des résultats expérimentaux détaillés basés sur des traces montrent que les performances des algorithmes en ligne proposés sont proches de l'optimum hors ligne et surpassent les solutions existantes.

Mots-clés

Smart grids, véhicules électriques, allocation de ressources, algorithmes d'approximation, analyse concurrentielle

Table of contents

Chapter 1

Introduction

Contents

1.1 Motivation

To promote quick adoption of green renewable energy sources, electrification of vehicles is a trend that has been globally advocated in recent years; the global sale of Electric Vehicles (EVs) increased by 80% in 2015 [1]. Consequently, it is expected that the EV charging demand constitute a considerable portion of total energy demand, e.g., currently transportation consumes 29% of total energy in the US, while electricity, in total, consumes around 40% [1]. The growing number of EVs along with the unprecedented advances in battery capacity and technology results in drastic increase in the total energy demand of EVs. This large charging demand makes the EV charging scheduling problem challenging. An apparent challenge is that even with taking the advantage of deferrable property of charging demands and performing proper scheduling, the aggregate demand might be beyond the tolerable charging rate of the station, given physical constraints of charger devices and transformers [2]. For example, the power capacity of a transformer in North America is limited to 25 kVA [3]. Furthermore, in practice, EVs arrive to charging station in online fashion and the charging station has no information about the arrival and demand of the future EVs. This makes the charging scheduling even more challenging.

It turns out that the problem of EV charging scheduling in a charging station and the cloud job scheduling problem have similar basic structure. Similarly to the job scheduling problem, EVs arrive to the charging station in an online fashion, each of which with different arrival time, deadline, demand, and value. The resource constraint in EV charging scenario is the limited power of the charging station to be allocated to the EVs at each time slot. The power constraint is determined by the chargers' or transformers' output power or is set manually by the station operator. Despite these similarities, EV scheduling problem poses an additional constraint that makes the corresponding classic job scheduling problem more challenging. More specifically, the input power of EV's battery is limited to a specific amount called *maximum charging rate*. Therefore, unlike the traditional scheduling problems, the completion time of a demand in EV scheduling problem not only depends on the availability of the resources, but it is also dependent to the maximum charging rate of its battery.

Another challenge is to design algorithms that work well in strategic environments where users are selfish and try to maximize their benefit by cheating the system. Put it another way, a user might lie about its preferences (e.g., deadline) to increase its utility. Hopefully, the field of mechanism design provides us with guidelines to cope with these users by designing *strategyproof* and *group-strategyproof* algorithms.

The goal of this thesis is to develop online scheduling algorithms for EVs with a focus on the performance and complexity of the algorithms, their application in strategic environments and in a network of multiple charging stations.

1.2 General Model

We consider one (or multiple) charging station (CS) for EV charging purpose. The time horizon is divided to T equal length time slots $t = \{1, 2, \dots, T\}$ (e.g., $T = 24$ with time slots of 1 hour length). There are n EVs (user or player, used interchangeably) denoted by set $\mathcal{N}$. Each EV i is characterized by its *"charging profile"* $\pi_i = \langle a_i, d_i, v_i, D_i, k_i \rangle$ indicating its arrival time, departure time, valuation of the job or willingness to pay, charging demand, and maximum charging rate, respectively. The maximum charging rate of an EV is a parameter dependent to physical properties of its battery. We refer to time interval $\mathcal{T}_i = [a_i, d_i]$ as *availability window* of EV i. At each time slot t in availability window of EV i, the scheduler can set the charging rate of i, denoted by y_i^t, to a value less than or equal to its maximum charging rate, k_i. We assume that for each EV i, its charging profile represent a feasible demand, i.e., we have $D_i \leq k_i(d_i - a_i + 1)$. We distinguish between *online* and *offline* scheduling. In this thesis, we consider online scenarios where the charging profile of an EV is revealed to the CS only upon its arrival. Therefore, the CS is uncertain about future demands which makes the scheduling problem challenging. In contrast, an offline scheduling algorithms has access to all charging profile of current and future EVs which cannot represent a real world scenario.

The variable v_i can carry different meanings in different scheduling problems (e.g., priority, revenue, gain and value for the user). When generally speaking, we will refer to v_i as valuation or gain. If EV i is charged before its deadline the gain is v_i. Otherwise, the gain could be zero (*integral revenue model*) or calculated proportionally (*fractional revenue model*) depending on the business model. It is assumed that for each EV i, its demand and deadline represent a feasible charging profile with respect to its maximum charging rate k_i. More specifically, we have $D_i \leq k_i(d_i - a_i + 1)$. At each time slot, the total electricity flowed to the EVs is restricted to P kW and is referred to as *peak constraint.* Depending on the scenario, there might be local and global peak constraint (e.g., in a multiple CS setting in Chapter 4). The peak constraint is set based on cost effective consumption policy or due to the fact that charger devices or installed transformers have constraint on the maximum power that they can output in a time slot [4–6].

A main part of this thesis discusses on the performance of the proposed algorithms. We use the notion of *competitive ratio* to evaluate our online algorithms. For a maximization problem, we say that an online algorithm is c-competitive with optimal offline solution if for all input sequences, the value of the optimal objective is at most c times the algorithm result. We have the notion of *approximation ratio* for the offline algorithms that is defined similarly.

1.3 Thesis Overview

This thesis is organized as follows. We first start by studying classic job scheduling problem and consider its extension to EVs in Chapter 2. The main difference is that EVs' battery have limitation on their charging rate while in the job scheduling problem where jobs are normally going to be processed by a CPU, there is no constraint on processing limit of the jobs (in fact, the processing rate is limited to CPU processing rate which is assumed to be high enough). The aim of this chapter is to provide performance analysis for some proposed algorithms that contribute to the classic job scheduling problem as well. In particular, the notion of competitive ratio is used to evaluate our scheduling algorithms in worst-case scenarios. The objective function of the optimization problem in this chapter is simply total valuation of EVs who get charged. This problem considers the profit of the aggregator but not the users. To extend this work, in Chapter 3, we study the EV scheduling problem with the objective of maximizing social welfare for the charging network which includes the profit of both users and aggregator. To define social welfare we considered two criteria including on-arrival charging notifications given by charging station to the EVs at their arrival time and valuation of the demands. To reflect a real-world scenario, we assume that each EV (or user) is selfish and might try to maximize its utility by misreporting its private data (including arrival time, deadline, valuation, demand and maximum charging rate). To cope with this, this chapter relies on the result from *mechanism design* field that aims to provide solutions for strategic environments. In both Chapter 2 and Chapter 3 the scheduling problem is solved for a single CS. Therefore, in Chapter 4, we proceed to address the EV scheduling problem in multiple station scenario where in addition to the local peak constraint in each CS, there is a global peak constraint where the aggregate allocated power in all CSs at a single time slot should not exceed the global peak. The global and local peak constraints could be set according to the maximum output power of transformers installed in charging network or according to a cost-effective policy. We inspired the system model from a real-world charging network consisting several CS. Also, different business models for EV charging is considered.

In what follows, we review the main results of each chapter.

Chapter 2: Competitive Online Scheduling Algorithms for EVs

In this chapter, we revisit the deadline constrained job scheduling problem with partial values and limiting maximum processing rate of the jobs and make the following key contributions: We propose a deterministic algorithm, WFair, along with a simple randomized algorithm, WRand that take into account the additional maximum charging rate constraint in EV charging scheduling problem. We show that both algorithms are $(2 - \frac{1}{U})$-competitive, where U is the maximum scarcity level of the system. To the best of our knowledge, amongst existing algorithms capable of respecting processing limit of the jobs, none of them attains a competitive ratio better than 2. To accomplish the competitive analysis of the two algorithms, we propose a new proof technique that can be applied to a wider class of algorithms beyond this work.

Chapter 3: Online Scheduling to Maximize Social Welfare

In this chapter we aim to study a social welfare maximization problem and tackling two challenges: *(1) Online scheduling with on-arrival commitment.* Enforcing capacity constraint may result in partial or no charging of some EVs. In a proper design, the scheduling mechanism must provide *on-arrival commitment* for the EVs, meaning that the mechanism must notify each EV upon receiving its charging demand whether or not it can receive (entirely or partially) the requested demand by the submitted departure time. Without on-arrival commitment, at departure time, an EV may realize that its charging request is not fulfilled, which definitely degrades user satisfaction. *(2) Strategy-proof and* *group-strategy-proof scheduling design.* The second challenge is a highly desired feature in social maximization problems which tries to propose mechanisms that are robust against selfish users and groups. Generally speaking, algorithmic mechanism design [7] is a field of game theory, that tries to devise *truthful* (also known as strategy-proof) mechanisms such that it is guaranteed that reporting true values is the best strategy for the players (EVs in our problem) regardless of the behavior of the others. Group-strategy-proofness is a natural generalization of strategy-proofness that tries to guarantee that not only truth-telling is the dominant strategy for individual players, but also, no group of players can improve the utility of at least one member of the group by lying, when the values of the other players are fixed.

Chapter 4: Online Scheduling in a Network of Charging Stations

This chapter studies EV charging scheduling in an adaptive charging network (ACN) governed by a single operator in a campus-scale location such as a university, a headquarter, etc. [8]. A notable example is the Caltech ACN [9] where several *charging stations* (CSs) dispersed in a *charging network* with the capability of adaptive charging of the electric vehicles. The problem is different from EV charging scheduling in single station scenarios, because of the essential need to respect the aggregate peak demand of the ACN. More specifically, the ACN operator might limit the total power drawn from EVs to control costs [10, 11], reserve the capacity for other loads, and/or participate in demand-response events. The aim of this thesis is to use the deferrable property of EVs and schedule their charging jobs, so as to intelligently control the global peak demand of the ACN.

1.4 Related Works

The charging scheduling problem for EVs in smart grids is a variation of classical job shop scheduling problem [12,13] and thus is studied in a large extent. The job scheduling problem has appeared in different application domains including task scheduling in processors [14, 15], cloud computing [16–19], and network buffer management [20]. In this problem, a decision maker aims to maximize the total value of processed jobs in the presence of deadline and resource constraints under heterogeneity in the value of jobs. In this section, we first look over the existing works in the related domains (Section 1.4.1), and then review the EV scheduling problem (Section 1.4.2).

1.4.1 Classic Job Scheduling Algorithms

As there is a plethora of real-world applications for the problem, extensive studies have been conducted on the basic form of the problem [16, 17, 19–30] with a focus on online algorithm design. These studies can be classified into full execution and partial execution models. We refer to [31] as the offline result, and [17,20,21,24,25,29] as online results.

1.4.1.1 Full Execution Model

A basic challenge in full execution form of the scheduling problem is that there is no partial gain for partially processed jobs. For the offline setting, efficient algorithms are developed [31,32] while the online version of the problem is more challenging and addressed in [17,20, 21,24,25,29,33–36]. A primal-dual algorithm in [21] is proposed and its competitive ratio is computed according to a slackness parameter, s, which indicates how much users can wait. The competitive ratio is $3+O\big(\frac{1}{(s-1)^2}\big)$ for $1 < s < 2$ and $2+O\big(\frac{1}{\sqrt[3]{s}}\big)$ for $s \geq 2$. The study is extended in [17] to provide incentive compatible scheduling mechanism design. However, in practice, the slackness parameter is expected to be close to 1 as users are usually desire to finish their job as soon as possible. [29] addressed the scheduling problem with commitment where users are notified in their arrival whether they will receive their demand or not. The system is still allowed to not complete an accepted job by accepting a penalty related to job's value. A simple online algorithm that achieves the optimal competitive ratio of $3 - 2\sqrt{2}$ is proposed. For packet scheduling problem, [36] is developed a 1.93-competitive algorithm. When packets are unit-length, it is shown in [35] that the competitive ratio lies in interval $[2, 1.63]$. Except [21], non of the above mentioned online solutions considered jobs' processing limit (a.k.a parallelism bound) in the formulation.

1.4.1.2 Partial Execution Model

Studies in [16, 19, 22, 23, 26–28, 30] considered partial execution model. Two simple and natural greedy algorithms named FirstFit and EndFit are proposed in [30], where both algorithms are 2-competitive and the bound is tight. For non-decreasing concave utility functions, the ISPEED algorithm in [16] provides competitive ratio of $2 + \alpha$, where α is a shape parameter. The study is extended in [19] to the case of multi-resource scenario by taking into account the processing limit of jobs and providing a competitive ratio of 2. In [26], the authors provide a lower bound of 1.236 for the competitive ratio and propose MIXED, which is shown to be 1.8-competitive. An improvement to this result appears in [27], where the authors propose MIX, and show that it is $\frac{e}{e-1} \approx 1.582$-competitive. The idea is that each job receives some resources according to its unit value unless its unit value is less than a threshold. Furthermore, a lower bound of 1.25 is provided for the competitive ratio of any randomized (and hence deterministic) online algorithm. We stress that filling the gap between the lower and the upper bounds is still an open problem. Moreover, authors provide an upper-bound of 1.618 when time sharing is not allowed (i.e., only one job can be processed at each time). [28] studied the problem when time sharing is not allowed and the number of concurrent jobs, m, is limited. Their proposed algorithm, GAP, is 1.618-competitive when there are only two concurrent jobs. However, GAP attains a larger competitive ratio when the number of concurrent jobs increases. Studies in [22,23] address scheduling in interactive services such as web servers and finance services but do not provide competitive analysis for the proposed methods. We emphasize that in our model jobs (EVs' charging demand in our case) have limited processing rate, which adds to the complexity of the problem.

1.4.2 EV Charging Scheduling

There is a growing number of studies in the EV scheduling problem (see, e.g., [37, 38]) to provide efficient algorithms aiming to optimize different objective functions including aggregator profit, users' comfort level, etc. In this section, we will focus on the studies that propose competitive algorithms, i.e., those whose worst-case performance with respect to the optimal offline solution is bounded.

The EV scheduling problem is a special case of job scheduling problem where the processing limit of jobs is an essential constraint to be considered. Therefore, the studies reviewed in Section 1.4.1 cannot be directly applied to the EV problem. Although there are some exceptions [19,21,31], yet none of them provide a competitive ratio better than 2. Moreover, [21,31] consider a slackness parameter in their model (see Section 1.4.1), which we believe cannot capture the real world scenarios. Also, the algorithm in [19] reduces to FirstFit algorithm [30] which is compared to our algorithms in simulation section. An online $\frac{e}{e-1} \approx 1.582$-competitive algorithm is developed in [39] but the constraint on the charging speed of the EVs is missing from the formulation. Moreover, the authors assume that all EVs have the same demand. Assuming that there is no resource constraint in charging station and the objective is to minimize the cost for the aggregator, [40] and [41] proposed online algorithms, called SOCA and ORCHARD respectively, that achieve the optimal competitive ratio of 2.39. The studied problem in this thesis is fundamentally different than [40] and [41] in the constraint set and the objective function. [42] considers same model as described in this thesis and proposes a truthful online scheduling algorithm assuming that discharging of EVs can be done instantaneously in their departure (referred to as on-departure burning) and EVs have the same charging rate. The authors extended the work in [43] and [44] for heterogeneous charging rates and proved that the proposed algorithm is 2-competitive. However, the assumption of on-departure burning is not realistic. [45] proposed TAGS algorithm and proved that the algorithm has optimal competitive ratio. However, in their model, all EVs have the same unit value and there is no limit on the charging speed. Under the same model, [46] proposed DSAC, an online scheduling algorithm with admission control (i.e., on-arrival notification). DSAC achieves an optimal competitive ratio for the case of linear utility function. In the next chapter, we will propose competitive online algorithms for EV scheduling problem where their competitive ratio are less than the best known results under fairly reasonable assumptions. To the best of our knowledge, there is no online algorithm achieving a competitive ratio better than 2 that respects charging rate limitations of the EVs (or jobs).

1.4.2.1 Peak-Constrained EV Charging Scheduling

There is an extensive literature on EV scheduling problem where most of them focused on single CS [4, 47] and the local and global peak constraints are usually omitted or only the local peak is considered. As discussed in Section 4.2.2, the global optimal solution cannot be obtained by separately solving the single station problems. Hence, those solutions cannot be directly applied to our problem.

A scenario that local and global peak constraints exist is the case that scheduling is required for a charging network for multiple CSs. Studies in [48–53] addressed charging scheduling problem in multiple CSs. The authors in [48] tackled a global EV charging scheduling problem in a system consisting of a central controller and multiple local controllers to minimize total cost. However, there is no limit on the maximum peak demand that the system can tolerate. Consequently, the peak value can be arbitrary high depending on the total demand. This may increase the electricity bill of large costumers substantially mainly due to very high peak demand. In addition, high EV penetration level leads to high peak which may pose danger for the grid system [51]. Besides, the charger devices installed in the CSs have limitation on the maximum power that they can transfer in each time unit [6]. We solve the issue by constraining local and global peaks. However, to meet the peak constraints, it may not be feasible to respond to all charging demands. Consequently, only a subset of EVs can be charged [5], which is captured in our integral charging model. [51] considered a multi-microgrid system with global peak constraint where each microgrid has a CS and the goal is to minimize the operating cost of the system and the exchanged electricity between microgrids and main grid. The authors assume that charging aggregators are able to forecast required information about individual EVs which may not represent a real scenario. A similar assumption is made in [52, 53] where the objective is to maximize total utility of EVs and aggregators in a distribution network.

To avoid big billing cost in peak hours, [50] proposed a solution based on genetic algorithm to find optimal capacity and location of parking lots for serving demands in peak hours with the goal of maximizing total benefit of all stations. Although authors studied the problem under multiple CS setting, their solution is not applicable to our setting when the CSs are already set up. [49] employed a similar model used in this thesis, where both local and global peak constraints in a charging network are considered. The objective in [49] is to maximize user convenience level which is different from the aim of the problem studied in this work. More importantly, the authors solve the single-slot problem, which fails to provide a general solution taking into account EVs' arrival and departure times which are considered in our study. As an alternative approach to control the peak, some studies directly targeted minimizing the peak [54, 55]. In [54], an online algorithm is developed for EV charging to minimize the peak by minimizing the impact uncertainty of renewable energies. [55] proposed a valley filling method by leveraging V2G in peak hours. Although the peak is minimized in above works, it cannot guarantee that the minimized peak is tolerable.

1.4.2.2 Scheduling Under Demand Uncertainty

A main challenge in EV scheduling problems is to cope with demand uncertainty. Many studies including [5, 47, 54, 56–59] addressed online scheduling problem with different objectives. [5, 56] studied the problem of maximizing social welfare considering the benefit for both users and service provider. [47] and [57] developed algorithms to minimize the charging price for the CS, where the proposed algorithm in [47] is 2.39-competitive. In [59], an online 2-competitive algorithm is proposed for a single CS which provides incentives for the users to truthfully report their data.

Our problem in this thesis is unique from above works in different aspects. First, we study the problem in an ACN where several CSs exist. None of the above studies solve the problem under this setting. Second, the previous algorithms do not work for both integral and fractional charging models. In addition, [57, 58] put no limit on the charging rate of EVs which makes their solution impractical in real scenarios. Also, [47, 54, 56–58] do not consider the peak limitation of the CS. Finally, the study in [59] assumes that EVs are capable of getting discharged in a negligible amount of time, resulting in a 2-competitive algorithm. However, the assumption is not realistic for EVs. In this thesis, we consider these limitations and develop online and offline algorithms for fractional and integral revenue models in an ACN and provide theoretical bounds on their performance.

Chapter 2

Single Station Online EV Scheduling to Maximize Revenue

Contents

2.1 Introduction

Online scheduling of heterogeneous deadline-constrained jobs in the presence of limited resources is a fundamental, yet challenging problem in different application scenarios. Notable examples are network buffer management [20], processors sharing [14, 15], as traditional applications, and cloud job scheduling [18] and electric vehicle (EV) charging scheduling [37, 46, 60], as the state-of-the-art examples.

In the classic form of the online scheduling of deadline-sensitive jobs, there is a limited resource (e.g., router's buffer, CPU time, or the maximum power capacity of EV charging station) that is shared among a set of jobs (users, task, or EVs) that arrive over time in an online fashion. The jobs are heterogeneous in terms of arrival, deadline, demand, and value (or weight), and the goal is to maximize total value obtained from the jobs, subject to the resource capacity constraints. The target applications could be categorized into *full* [17, 31] and *partial* [16, 19, 23, 26–28, 30] execution models. In the present work, we focus on the latter, where partially completed jobs get partial values proportional to their received resource. Notable examples of partial models are job scheduling in web search applications [22], multimedia content transmission [30], and EV charging scheduling [37].

The underlying classic problem under partial execution model has been first introduced in [30] and two simple greedy heuristics are proposed. Our focus in this chapter is on online algorithms with a bounded worst-case performance determined by their *competitive ratio*[1] to maximize charging station gain. Using the competitive analysis [61], the authors in [30] demonstrated that both algorithms achieves the competitive ratio of 2 Using the ideas of prioritizing the valuable jobs and timesharing among the jobs, in [26] and [27], the competitive ratio has been improved to 1.8 and 1.582, respectively. With extensive applications in the recent research topics, the problem has been extended to several other settings such as multi-resource allocation [19], providing resource commitment [29], and truthful analysis [17], among others.

This chapter especially focuses on the application of scheduling that is identified with the advent of EVs. EVs are a promising alternative for the conventional vehicles considering their significant advantages in energy efficiency, zero emission, and relieve reliance on fossil fuels. With increasing number of EVs, their charging demand can pose a tremendous challenge to the power system operation [37, 46, 60]. EV charging demand, however, is usually deferrable implying that there is often considerable flexibility in charging schedule.

It turns out that the problem of EV charging scheduling in a charging station and the cloud job scheduling problem have similar basic structure. Similarly to the job scheduling

[1] An online algorithm $\mathcal{A}$ is c-competitive for $c \geq 1$ if for any input instance the optimal gain is at most c times the algorithm's gain.

problem, EVs arrive to the charging station in an online fashion, each of which with different arrival time, deadline, demand, and value. The resource constraint in EV charging scenario is the limited power of the charging station to be allocated to the EVs at each time slot. The power constraint is determined by the chargers' or transformers' output power or is set manually by the station operator. Despite these similarities, EV scheduling problem poses an additional constraint that makes the corresponding classic job scheduling problem more challenging. More specifically, the input power of EV's battery is limited to a specific amount called *maximum charging rate*. Therefore, unlike the traditional scheduling problems, the completion time of a demand in EV scheduling problem not only depends on the availability of the resources, but it is also dependent to the maximum charging rate of its battery.

In this chapter, we revisit the deadline constrained job scheduling problem with partial values and limiting maximum processing rate of the jobs and make the following key contributions:

1. We propose a deterministic algorithm, WFair, along with a simple randomized algorithm, WRand that take into account the additional maximum charging rate constraint in EV charging scheduling problem. We show that both algorithms are $(2 - \frac{1}{U})$-competitive, where U is the maximum scarcity level of the system. To the best of our knowledge, amongst existing algorithms capable of respecting processing limit of the jobs, none of them attains a competitive ratio better than 2.

2. We examine the performance of the proposed algorithms by trace-driven experiments. As our results show, the empirical cost ratios of our algorithms are much better than the obtained theoretical competitive ratios.

3. To accomplish the competitive analysis of the two algorithms, we propose a new proof technique that can be applied to a wider class of algorithms beyond this work. In particular, when applied to derive competitive performance bounds of some existing algorithm (e.g., [30]), the presented technique recovers the same results using much simpler proofs. We therefore believe that it could be of independent interest beyond EV charge scheduling problem as well.

The rest of this chapter is organized as follows. In Section 2.2, the tailored system model for EV charging application is introduced and the problem is formulated. Section 2.3 proposes two deterministic and randomized algorithms. The deviation method is introduced in Section 2.4 and used for the competitive analysis. The results of simulations are reported in Section 2.5. Finally, Section 2.6 concludes the chapter and highlights future directions.

Table 2.1 – Summary of notations

Notation	**Description**
$\mathcal{T}$	Set of time slots with $\|\mathcal{T}\| = T$, indexed by t
$\mathcal{N}_t$	Set of available users at t with $\|\mathcal{N}_t\| = n_t$
$\mathcal{M}_t$	Set of active users at t with $\|\mathcal{M}_t\| = M_t$
$\mathcal{T}_j$	Availability window of user j
D_j	Demand of user j
v_j	Value of user j for receiving demand D_j
K	The maximum charging rate of users
ρ_j	Unit value of user j, i.e., $\rho_j = v_j/D_j$
$R_{j,t}$	Residual demand of user j at t: $D_j - \sum_{t'\in\mathcal{T}_j,t'\le t} y_{j,t'}$
P	Capacity constraint of the charging station
$y_{j,t}$	**decision variable**, the amount that user j is charged at t

2.2 System Model and Problem Formulation

We consider a discrete time-slotted system where the time horizon is divided into T time slots indexed by $t \in \mathcal{T} := \{1, \dots, T\}$. Time slots are assumed to be of equal lengths. We present our model in the context of EV charging scheduling. Consider a single charging station with capacity (resource) constraint of P (in kWh, say) to serve a set comprising n users (EVs, or jobs, used interchangeably) indexed by j. User j is represented by its demand profile $\pi_j = \langle \mathcal{T}_j, v_j, D_j \rangle$, where $\mathcal{T}_j$ denotes the *availability window* and v_j is the value for receiving demand D_j. The availability window $\mathcal{T}_j$ consists of all time slots from the arrival to the departure of user j. Let K denote the maximum charging rate of the EV, which is assumed to be fixed for all EVs[2]. We denote by $\rho_j = \frac{v_j}{D_j}$ the *unit value* (a.k.a. marginal value [31] or value density [24]) of user j.

We consider an online setting in which the profile of each user is only known to the scheduler upon its arrival.

We assume a preemptive model in which the scheduler is allowed to pause charging of an EV at any time and resume it later. We denote by $y_{j,t} \in [0, K]$ the allocated resource to the EV j at slot t. Moreover, $R_{j,t} = D_j - \sum_{t':t'\le t} y_{j,t'}$ is the residual demand of EV j at time slot t. We consider the partial charging model, where if EV j receives its total demand D_j within its availability window, the obtained value is v_j. Otherwise, its gain would be $\sum_{t\in\mathcal{T}_j} y_{j,t}\rho_j$.

Next we introduce some definitions. We say that EV j is *available* at time slot t if

[2] Our algorithms can be straightforwardly extended to the setting with heterogeneous charging rate demands. We consider fixed rates to facilitate our competitive analysis.

$t \in \mathcal{T}_j$. Moreover, given the scheduling policy, EV j is *active* at time slot t if it is available at t but its charging demand is not fulfilled yet. Finally, EV j is said to be *selected* at time slot t if $y_{j,t} > 0$. For any time t, let $\mathcal{N}_t$ and $\mathcal{M}_t$ denote the set of available and active EVs at time slot t, respectively. Further, let n_t and M_t be the cardinality of $\mathcal{N}_t$ and $\mathcal{M}_t$, respectively. Under a given algorithm $\mathcal{A}$, introduce $\mathcal{S}_{\mathcal{A},t}$ as the set of selected EVs at time t by $\mathcal{A}$:

$$\mathcal{S}_{\mathcal{A},t} := \{j : y^{\mathcal{A}}_{j,t} > 0\}.$$

Moreover, we define $\mathcal{S}_{\mathrm{OPT},t} := \{j : y^{\star}_{j,t} > 0\}$, where $y^{\star}_{j,t}$ is the allocated resource to j at t by the optimal solution.

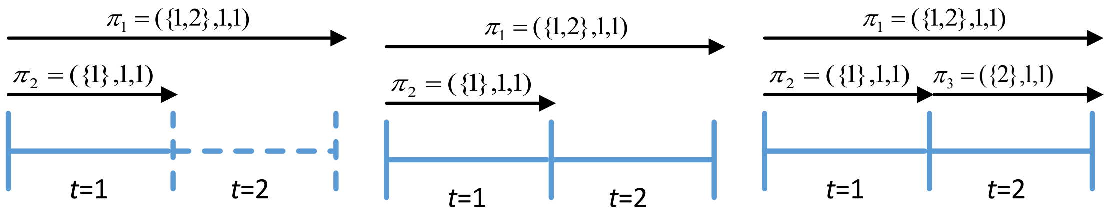
The key notations used in this chapter are listed in Table 2.1.

Having introduced these notations and definitions, we may formulate the EV scheduling problem under partial execution model as follows:

$$\begin{aligned}
\textsc{rjsp}: \quad \max_{\vec{y}} \quad & \sum_{j=1}^{n} \rho_j \sum_{t\in\mathcal{T}_j} y_{j,t} && \text{(2.1a)}\\
\text{s.t.} \quad & \sum_{t\in\mathcal{T}_j} y_{j,t} \le D_j, \qquad \forall j && \text{(2.1b)}\\
& \sum_{j:t\in\mathcal{T}_j} y_{j,t} \le P, \qquad \forall t && \text{(2.1c)}\\
& 0 \le y_{j,t} \le K, \quad \forall j,t, && \text{(2.1d)}\\
& y_{j,t} = 0, \quad \forall (j,t) : t \notin \mathcal{T}_j && \text{(2.1e)}
\end{aligned}$$

The RJSP in Eq. (2.1a) maximizes the charging station gain. The constraint in (2.1b) limits the total resources received by an EV to its demand as there is no benefit for the charging station to overcharge the EVs. The second constraint in (2.1c) is the capacity constraint, and the third and the fourth constraints enforce the charging station to respect the maximum charging rate and to charge EVs only during their availability window.

First observe that RJSP is a linear program and can hence be solved efficiently in offline scenarios. Second, in online scenarios the problem is less challenging to solve if the charging rate constraint in (2.1d) is omitted. In fact, with the charging rate constraint, part of the resources at some time slots might remain unused while there are some users that have not received their entire demand yet. Such users may also not receive their total demand in the next time slots if they are not selected for charging due to resource scarcity. Third, any c-competitive algorithm for RJSP is also a c-competitive solution for the basic form of RJSP (i.e., the form without maximum charging rate constraint). However, the inverse is not necessarily true.



(a) Jobs 1 and 2 arrive at $t = 1$.

(b) **Scenario 1:** No arrival at $t = 2$.

(c) **Scenario 2:** Job 3 arrives at $t = 2$.

Figure 2.1 – A simple scheduling scenario. Dotted line indicates a time slot that is not visited yet and the scheduler has no information about the arriving EVs in that slot.

2.3 Online Scheduling Algorithms

In this section, we propose two online algorithms for RJSP. The competitive ratios of the two algorithms as well as their computational complexities are summarized in Table 2.2.

Table 2.2 – Summary of the proposed algorithms

Algorithm	**comp. ratio**	**Complexity**	**Type**
WFair	$2 - \frac{1}{U}$	$O(n^2T)$	Deterministic
WRand	$2 - \frac{1}{U}$	$O(nT \log n)$	Randomized

2.3.1 The WFair Algorithm

In this subsection, we present a deterministic algorithm, which we refer to as WFair, as an online algorithm for RJSP. The pseudo-code of WFair is listed as Algorithm 1.

WFair allocates the available resources to the users proportional to their unit values. More precisely, at each time slot t, the algorithm runs in multiple rounds, where at each round an active user j receives

$$\min\Big\{\frac{\rho_j}{\sum_{i\in\mathcal{M}_t}\rho_i}\Big(P - \sum_{i\in\mathcal{N}} y_{i,t}\Big), R_{j,t}, K - y_{j,t}\Big\} \tag{2.2}$$

units of the resource (Line 6 of the algorithm). The received resource by each user is linearly correlated to its unit value. Therefore, for all active users at t, it holds that $y_{j,t} > 0$ as unit values are non-zero, i.e., no user will be left unallocated but it may receive an infinitesimal amount if $\frac{\rho_j}{\sum_{i\in\mathcal{M}_t}\rho_i}$ is very small. Note that for some users, the second or the third term in Eq. (2.2) might be selected. In this case, the aggregate allocated amount might be less than the total capacity. This potential issue is resolved by re-allocating the residual resource

Algorithm 1: WFair (for time slot t)

```
$\mathcal{L}_t \leftarrow \mathcal{M}_t$
$y_{j,t} \leftarrow 0, \forall j$
while $\sum_j y_{j,t} < P$ and $\mathcal{L}_t \neq \emptyset$ do
    for all $j \in \mathcal{L}_t$ do
        $\delta_{j,t} \leftarrow \min\{\frac{\rho_j}{\sum_{i \in \mathcal{L}_t} \rho_i}(P - \sum_{i \in \mathcal{N}_t} y_{i,t}), R_{j,t}, K - y_{j,t}\}$
    for all $j \in \mathcal{L}_t$ do
        $y_{j,t} \leftarrow y_{j,t} + \delta_{j,t}$
        $R_{j,t} \leftarrow R_{j,t} - \delta_{j,t}$
        if $R_{j,t} = 0$ then
            $\mathcal{L}_t \leftarrow \mathcal{L}_t \backslash j$
```

in multiple iterations until the entire resource allocated or all the active users get their maximum possible requirements.

We stress that this allocation rule is in contrast to that of FirstFit algorithm [30], which allocates the resources to the most valuable users first.

Fig. 2.1 shows a general example, which we will refer to frequently to clarify the technical discussions. If we run WFair over the scenario of Fig. 2.1, it will share the resources in the first time slot equally between users 1 and 2 (because $\rho_1 = \rho_2$) and set $y_{1,1} = y_{2,1} = 0.5$. Therefore, the gain (i.e., total valuation of allocated resources) of WFair at $t = 1$ is 1. The worst-case for WFair happens when no user arrives at $t = 2$. In this case, Opt $= 2$ (by allocating user 2 in the first slot and user 1 in the second slot) and WFair will set $y_{1,2} = 0.5$. So, the total gain by WFair is 1.5.

We now illustrate a worst-case instance for WFair. For more in-depth analysis on the competitive ratio of WFair see Section 2.4.

Worst-case instance for WFair: Consider a single time slot scenario with $P = K = T = 1$ and n users where n is sufficiently big. The charging profiles are $\pi_1 = \langle\{1\}, \frac{1}{2}, \frac{1}{2}\rangle$ and $\pi_2 = \cdots = \pi_n = \langle\{1\}, \frac{1}{2n}, \frac{1}{2}\rangle$. Hence, we have $\rho_1 = 0.5$ and $\rho_2 = \cdots = \rho_n = \frac{1}{2n}$. Therefore, $\sum_j \rho_j = 0.5 + \frac{n-1}{2n}$ which approximates to 1 as n is large. The optimal solution is to fully schedule user 1 while giving no resources to the other users. WFair shares the resources between all the EVs such that EV 1 only receives half of the resource and the other half is allocated to rest of the users. This leads to a total gain of 0.5 while the optimal gain is 1.

The above example indicates that the competitive ratio of WFair could not be better than 2. We however note that the presented worst-case scenario is quite unrealistic as the ratio of demand-to-supply is a small constant in practice. Under this assumption, the competitive ratio can be improved.

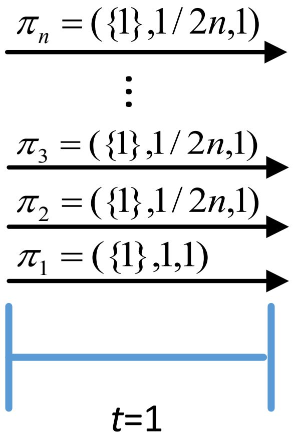


Figure 2.2 – A worst-case instance for WFair.

Next we define the notion of scarcity level:

Definition 2.3.1.1 (Scarcity Level [62]). *The* scarcity level U_t *at time slot t is defined as* $U_t = \frac{n_t K}{P}$. *Moreover, the maximum scarcity level of the system is* $U = \max_t U_t$.

Indeed the scarcity level U_t is an indication of *demand-to-supply ratio*, where the demand is roughly U_t times higher than the available resource.

The following theorem provides the competitive ratio of WFair:

Theorem 2.3.1.1. WFair *is* $(2 - \frac{1}{U})$*-competitive.*

2.3.2 The WRand Algorithm

In this subsection, we present WRand, a *randomized* algorithm for rjsp. In general, randomized algorithms bring two main advantages over deterministic ones. First, they are usually more efficient in terms of the algorithm cost. Second, a randomized algorithm usually admits a simpler design than deterministic ones, which in turn makes the implementation easier. The competitive ratio of a randomized algorithm is measured with respect to an *adversary model*, which determines the way the input sequence to the problem is generated. We distinguish between two notions of adversary: *oblivious adversary* and *adaptive online adversary*. An oblivious adversary knows the algorithm code but should choose the entire input sequence in advance (i.e., before the start of the algorithm), whereas an adaptive online adversary, can well condition the input at each time step on the algorithm's history of plays.

The WRand algorithm is motivated as follows (we refer to Algorithm 2 for its pseudocode). At each slot t, the algorithm selects one or multiple active users randomly with a

Algorithm 2: WRand (for time slot t)

```
1 $\mathcal{L}_t \leftarrow \mathcal{M}_t$
2 $y_{j,t} \leftarrow 0, \forall j$
3 while $\sum_i y_{i,t} < P$ and $\mathcal{L}_t \neq \emptyset$ do
4     Select user $j$ with probability $\frac{\rho_j}{\sum_{i \in \mathcal{L}_t} \rho_i} \mathbb{I}_{\{j \in \mathcal{L}_t\}}$
5     $y_{j,t} \leftarrow \min\{K, R_{j,t}, P - \sum_i y_{i,t}\}$
6     $\mathcal{L}_t \leftarrow \mathcal{L}_t \backslash j$
```

probability proportional to their unit values: the more the unit value of a user, the higher the probability it will be selected. More specifically, the algorithm maintains a set $\mathcal{L}_t$ that comprises all active users whose demand has not been met. Then, at each round of the 'while' loop, it selects a user j with probability proportional to $\rho_j \mathbb{I}_{\{j \in \mathcal{L}_t\}}$, where for an event X, $\mathbb{I}_X = 1$ if X holds, and $\mathbb{I}_X = 0$ otherwise. Then, the selected user is processed with the highest rate (Line 5). The process continues until no more user can be processed.

Theorem 2.3.2.1. WRand *is* $(2 - \frac{1}{U})$*-competitive against an oblivious adversary.*

2.3.3 Discussion

We provide some remarks on the proposed algorithms.

- First, WFair and WRand characterize the competitive ratio as a function of the scarcity level. The worst competitive ratio bound (equal to 2) for these algorithm, which occurs when U tends to infinity, matches the existing results with maximum charging rate [19, 21, 31]. In practice, however, the scarcity level is expected to be a small constant as the capacity is usually set based on the expected demand (as in, e.g., a cloud). Fig. 2.3 depicts the competitive ratio of the proposed algorithms against different values of U.

- The time complexity of WFair and WRand are $O(n^2 T)$ and $O(nT \log n)$, respectively. Thus, WRand is a better choice in terms of computational complexity while attaining the same competitive ratio. Due to space constraints, we omit the details of the time complexity analysis.

- Finally, we mention that both our proposed algorithms are deadline-oblivious as they do not use the users' deadline in decision making. This property, on the one hand, proves useful in scenarios where the users' deadline are not provided to the system. It also makes the implementation easier. On the other hand, deadline-aware scheduling

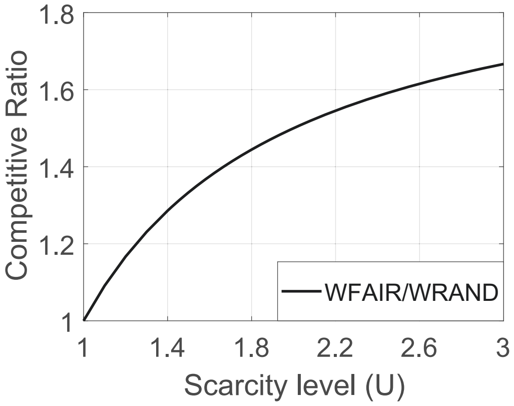


Figure 2.3 – Competitive ratio of proposed algorithms w.r.t. scarcity level (Definition 2.3.1.1).

algorithms may enjoy a better competitive ratio (than that of deadline-oblivious ones) by utilizing the deadline information. Probably, no deadline-oblivious scheduling algorithm for RJSP can attain a competitive ratio better than $2 - \epsilon$ for all $\epsilon > 0$ when U grows large. An intuitive proof for this could be obtained by considering a scheduling problem in two time slots (i.e., $T = 2$) with $n + 2$ users, $P = K = 1$, and setting $\pi_1 = \cdots = \pi_n = \langle\{1, 2\}, 1, 1\rangle$ and $\pi_{n+1} = \langle\{1\}, 1, 1\rangle$. Since users 1 to $n + 1$ only differ in their deadlines, they could not be distinguished by a deadline-oblivious algorithm. This would lead to a situation in which, with a high probability, user $n + 1$ at $t = 1$ would not be allocated any resource. In this case, the adversary can set $\pi_{n+2} = \langle\{2\}, 1, 1\rangle$, thus resulting in a competitive ratio of 2.

2.4 Competitive Analysis

2.4.1 Preliminaries

The competitive analysis of our proposed algorithms relies on a proof technique, which is novel to the best of our knowledge. In this subsection, we describe our proof technique and illustrate it through some examples.

Let $\mathcal{A}$ be an online algorithm that outputs a feasible solution for RJSP. Let $y^{\mathcal{A}}_{j,t}$ denote the resource (charging rate) allocated to user j at time t under $\mathcal{A}$, and ALG be the corresponding objective value. Fix an optimal offline algorithm with objective value OPT and charging rates $y^{\star}_{j,t}, j \in \mathcal{N}, t \in \mathcal{T}$. If $y^{\mathcal{A}}_{j,t} \geq y^{\star}_{j,t}$ for all j and t, then $\mathcal{A}$ is optimal. However, if there exists a user j and a time slot t such that $y^{\star}_{j,t} > y^{\mathcal{A}}_{j,t}$, then the difference $y^{\star}_{j,t} - y^{\mathcal{A}}_{j,t}$ *might* increase the gap between ALG and OPT (by the amount $(y^{\star}_{j,t} - y^{\mathcal{A}}_{j,t})\rho_j$). Let $B_{j,t}$ be the *block*[3] of the resource that $\mathcal{A}$ allocated to user j at t with $|B_{j,t}| = y^{\mathcal{A}}_{j,t}$. Furthermore, denote by $\bar{B}_{j,t}$ the block corresponding to the additional resource that the optimal algorithm allocated to user j at t as compared to $\mathcal{A}$, which could be *feasibly* allocated by $\mathcal{A}$ to user j at t.

We denote by $\Phi_{j,t}$ the total gain that could be obtained by $\mathcal{A}$ if it had allocated $B_{j,t} \cup \bar{B}_{j,t}$ to j at time t. We have

$$\Phi_{j,t} = \rho_j y^{\mathcal{A}}_{j,t} + \bar{g}_{j,t}\,, \tag{2.3}$$

where $\bar{g}_{j,t}$ denotes the gain of $\mathcal{A}$ in block $\bar{B}_{j,t}$. To calculate $\bar{g}_{j,t}$, we need to know the valuation of EV(s) (if any) that occupied block $\bar{B}_{j,t}$ as well as the size of $\bar{B}_{j,t}$, which we denote by $\Delta_{j,t}$. Based on the previous discussion, $\Delta_{j,t}$ can be determined as follows:

$$\Delta_{j,t} = \begin{cases} \min\{y^{\star}_{j,t} - y^{\mathcal{A}}_{j,t}, R_{j,t}\} & y^{\star}_{j,t} \geq y^{\mathcal{A}}_{j,t} \\ 0 & \text{otherwise.} \end{cases} \tag{2.4}$$

The gain of $\mathcal{A}$ in sum of the two blocks $\bar{B}_{j,t}$ and $B_{j,t}$ is $\rho_j \Delta_{j,t}$ units less than that of the optimal solution unless $\mathcal{A}$ allocates the difference $\Delta_{j,t}$ to some other EVs and obtains the corresponding gain, $\bar{g}_{j,t}$. If $\mathcal{A}$ allocates the whole block $\bar{B}_{j,t}$ to a single EV i, then $\bar{g}_{j,t} = \Delta_{j,t}\rho_i$. More complex cases where there are more than one EV that occupy $\bar{B}_{j,t}$ will be considered later in the competitive analysis of our algorithms.

Next we introduce the notions of *gain* and *loss*. The gain of algorithm $\mathcal{A}$ at time t is defined as follows:

$$\Gamma_{\mathcal{A},t} = \sum_j \rho_j y^{\mathcal{A}}_{j,t}. \tag{2.5}$$

Furthermore, we note that $\textsc{Alg} = \sum_{t\in\mathcal{T}} \Gamma_{\mathcal{A},t}$.

Define $L_{\mathcal{A},t}$ as the *loss* of $\mathcal{A}$ at t expressed as

$$L_{\mathcal{A},t} = \textsc{Opt} - \textsc{Opt}^{-t}_{\mathcal{A}},$$

where $\textsc{Opt}^{-t}_{\mathcal{A}}$ is the optimal value of a variant of RJSP where the resource allocated to any user i at time t coincides to that allocated by $\mathcal{A}$. Equivalently, $\textsc{Opt}^{-t}_{\mathcal{A}}$ is the optimal value of RJSP with the following additional constraint: for all i, $y_{i,t} = y^{\mathcal{A}}_{i,t}$. Moreover, the total loss of $\mathcal{A}$ is given by $L_{\mathcal{A}} = \sum_{t\in\mathcal{T}} L_{\mathcal{A},t}$. The value $L_{\mathcal{A},t}$ characterizes the amount that $\mathcal{A}$ deviates from $\textsc{Opt}$ at slot t. Define the loss of user j at slot t as

$$L_{j,t} = \rho_j \Delta_{j,t}, \quad t \in \mathcal{T}_j. \tag{2.6}$$

[3] Block is a conceptual term that facilitates our theoretical analysis and is not appeared in the main body of algorithm design.

Then an upper bound on $L_{\mathcal{A},t}$ can be obtained as follows:

$$L_{\mathcal{A},t} \leq \sum_{j \in \mathcal{S}_{\textsc{Opt},t}} L_{j,t}.$$

In the following theorem, we relate the notion of loss of an algorithm $\mathcal{A}$ to its competitive guarantee.

Theorem 2.4.1.1. *If $L_{j,t} \leq c\Phi_{j,t}$ for all j and t, for some $c \geq 0$, then $\mathcal{A}$ is $(1+c)$-competitive.*

Proof. To prove the theorem, we provide lower and upper bounds on $L_{\mathcal{A}}$. First observe from the definition of $L_{\mathcal{A},t}$ that the gap between $\textsc{Opt}$ and $\Gamma_{\mathcal{A}}$ is less than or equal to the aggregate loss over the time horizon:

$$\textsc{Opt} - \textsc{Alg} \leq L_{\mathcal{A}}.$$

On the other hand, we have

$$\begin{aligned} L_{\mathcal{A}} &= \sum_{t \in \mathcal{T}} L_{\mathcal{A},t} \leq \sum_{t \in \mathcal{T}} \sum_{j \in \mathcal{S}_{\textsc{Opt},t}} L_{j,t} \\ &\leq \sum_{t \in \mathcal{T}} \sum_{j \in \mathcal{S}_{\textsc{Opt},t}} c\Phi_{j,t} \\ &= c \sum_{t \in \mathcal{T}} \Gamma_{\mathcal{A},t} = c\textsc{Alg}. \end{aligned}$$

Putting the two bounds together gives $\textsc{Opt} \leq (1+c)\textsc{Alg}$ and completes the proof. □

In what follows, we first define the notion of work-conserving algorithm and then provide two examples to illustrate the application of the technical tool described above.

Definition 2.4.1.1 (Work-Conserving Algorithm [63]). *A scheduling algorithm is work-conserving if it processes requests as long as there is some resources to allocate.*

Example 1: Consider a scheduling problem during 2 time slots ($T = 2$) with $P = 1$ and $K = 1$ as shown in Fig. 2.1a. At the first time slot, users 1 and 2 arrive with demand profiles $\pi_1 = (\{1,2\}, 1, 1)$ and $\pi_2 = (\{1\}, 1, 1)$. Consider an algorithm $\mathcal{A}$ that selects user 1 to process at time slot 1. The gain at the first time slot is $\Gamma_{\mathcal{A},1} = 1$. For the second slot, we consider two scenarios as shown in Figs. 2.1b-2.1c. In the first scenario (Fig. 2.1b), where no EV arrives, we get $\textsc{Opt} = 2$ (by setting $\vec{y}_1^\star = [0,1]$ and $\vec{y}_2^\star = [1,0]$). Since $\mathcal{A}$ already fully charged EV 1, we get $\textsc{Alg} = 1$ (with $\vec{y}_1 = [1,0]$ and $\vec{y}_2 = [0,0]$). To obtain $\textsc{Opt}_{\mathcal{A}}^{-1}$ we fix algorithm $\mathcal{A}$'s decision at time slot 1 (that is selecting user 1) and find the maximum objective value that can be obtained by $\mathcal{A}$ which is 1. Therefore, $\textsc{Opt}_{\mathcal{A}}^{-1} = 1$. The loss of $\mathcal{A}$ at $t = 1$ is then $L_{\mathcal{A},1} = \textsc{Opt} - \textsc{Opt}_{\mathcal{A}}^{-1} = 2 - 1 = 1$. Now let us consider the second scenario, where user 3 arrives at $t = 2$ with $\pi_3 = (\{2\}, 1, 1)$ (in Fig. 2.1c). In this case, if $\mathcal{A}$ sets $y_{3,2} = 1$, then $\Gamma_{\mathcal{A},2} = 1$ and thus $\Gamma_{\mathcal{A}} = 2$. We have $\textsc{Opt} = 2$ and so, $\textsc{Opt}_{\mathcal{A}}^{-1} = 2$ and $L_{\mathcal{A},1} = \textsc{Opt} - \textsc{Opt}_{\mathcal{A}}^{-1} = 2 - 2 = 0$.

Example 2 (Competitive analysis of FirstFit *[30])*: FirstFit [30] is a natural 2-competitive greedy scheduling algorithm that sorts the users based on their unit values and selects them one at a time until no more user can be allocated. Each time, the most valuable EV from the sorted list is selected and the processing rate is set to the maximum feasible rate. The algorithm continues until no more feasible allocation is possible (therefore, the algorithm is work-conserving). According to Theorem 2.4.1.1, the competitive ratio of FirstFit is $(1 + \max_{j,t} L_{j,t}/\Phi_{j,t})$.

In what follows we apply our technique to derive the competitive ratio of FirstFit. For selected user j at slot t by the optimal solution, we have $L_{j,t} \leq \rho_j y_{j,t}^{\star}$. If $\Delta_{j,t} = 0$, it means that the block $\bar{B}_{j,t}$ is allocated to some other users with at least the same unit values as j otherwise, FirstFit would process j with a higher speed. Therefore, $\bar{g}_{j,t} \geq \rho_j \Delta_{j,t}$. Thus, noting that $y_{j,t} + \Delta_{j,t} = y_{j,t}^{\star}$, we get

$$\frac{L_{j,t}}{\Phi_{j,t}} = \frac{\rho_j \Delta_{j,t}}{\rho_j y_{j,t} + \bar{g}_{j,t}} \leq \frac{\rho_j \Delta_{j,t}}{\rho_j \Delta_{j,t}} = 1.$$

Applying Theorem 2.4.1.1 proves that FirstFit is 2-competitive.

We conclude this subsection by the following definition:

Definition 2.4.1.2 (Saturated Time Slot)**.** *A time slot t is said to be* saturated *if it satisfies $\sum_j y_{j,t} = P$.*

2.4.2 WFair Analysis (Proof of Theorem 2.3.1.1)

We first note that we assumed $U > 1$. For the case where $U \leq 1$, it is trivial to show that WFair is optimal as there will always be sufficient resources to schedule all users with the maximum speed.

To prove the theorem, we compute $L_{j,t}$ and $\Phi_{j,t}$ for WFair, and then apply Theorem 2.4.1.1. Without loss of generality, assume $0 < \rho_i \leq 1$ for all i and $\sum_{i \in \mathcal{M}_t} \rho_i = 1$. This is always possible through normalization, namely by dividing the unit value of each user to the sum of unit values of all active users. Moreover, we assume that for any active job $i \neq j$ at time t, it holds that $R_{i,t} \leq K$. This assumption can be relaxed by temporarily aggregating multiple demands into a single demand for the current slot and then splitting them at the next slot.

Let $\mathcal{A} = \text{WFAIR}$, and to ease notation, in the rest of the proof we omit the dependence of $y^{\mathcal{A}}_{j,t}$ on $\mathcal{A}$ for all j and t (so $y_{j,t} := y^{\mathcal{A}}_{j,t}$). Fix an optimal solution and a user $j \in \mathcal{S}_{\text{OPT},t}$. Let $y^{\star}_{j,t}$ denote the amount of resource allocated by the optimal solution to j at time t. Since we consider the worst-case, in the rest of the proof we assume that j is not completed by WFAIR (otherwise, $L_{j,t} = 0$) and thus, $j \in \mathcal{M}_t$.

If $\sum_{i\in\mathcal{M}_t} \min\{K, R_{i,t}\} \le P$, then all active users can be scheduled with the maximum feasible rate at t and the gain is $\rho_i \min\{K, R_{i,t}\}$ for all $i \in \mathcal{M}_t$. In this case, $L_{\mathcal{A},t} = 0$ since for any available user i, $y_{i,t} = \min\{K, R_{i,t}\} \ge y^{\star}_{j,t}$ or i is completed in an earlier time slot.

Now, we focus on the case where $\sum_{j\in\mathcal{M}_t} \min\{K, R_{i,t}\} > P$. First, observe that WFAIR is work-conserving since the "while" loop in WFAIR will not terminate if more resources can be allocated to the users. We further deduce that time slot t is saturated. This implies that there must be a non-empty set $\mathcal{H} \subseteq \mathcal{N}_t \backslash \{j\}$ of users such that they received the difference $\Delta_{j,t}$. Let $H = |\mathcal{H}|$ and note that $H \le n_t - 1$. Let $y_{i,t}, i = 1, \dots, H$ be the amount that WFAIR allocated to user $i \in \mathcal{H}$ with

$$\Delta_{j,t} = \sum_{i\in\mathcal{H}} y_{i,t}$$

Then, we have $\bar{g}_{j,t} = \sum_{i\in\mathcal{H}} \rho_i y_{i,t}$. According to allocation strategy of WFAIR, $y_{j,t} = \min\{\rho_j P, K, R_{j,t}\}$. Since $\Delta_{j,t} > 0$, thus, $y_{j,t} < K$. Also, as j is not yet finished by WFAIR at t, $y_{j,t} < R_{j,t}$. Therefore, $y_{j,t} = \rho_j P$. Since $y_{i,t} \le \rho_i P$ for all $i \in \mathcal{H}$. Thus,

$$\sum_{i\in\mathcal{H}} \rho_i y_{i,t} \ge \frac{1}{P} \sum_{i\in\mathcal{H}} y^2_{i,t},$$

which further gives $\bar{g}_{j,t} \ge \frac{1}{P} \sum_{i\in\mathcal{H}} y^2_{i,t}$. The right-hand side of the above is minimized with $y_{i,t} = \frac{1}{H}\Delta_{j,t}$. Hence,

$$\bar{g}_{j,t} \ge \frac{1}{P} \sum_{i\in\mathcal{H}} \frac{\Delta^2_{j,t}}{H^2} = \frac{\Delta^2_{j,t}}{PH},$$

and we get

$$\frac{L_{j,t}}{\Phi_{j,t}} \le \frac{\rho_j \Delta_{j,t}}{\rho_j^2 P + \frac{\Delta^2_{j,t}}{PH}}.$$

Let $\Delta_{j,t} = a\rho_j$ where $a > 0$ is a constant to be identified. By replacing $\Delta_{j,t}$ we have $\frac{L_{j,t}}{\Phi_{j,t}} \le \frac{aPH}{P^2H + a^2}$. The maximum value of this term is obtained by setting $a = P$. Therefore,

$$\frac{L_{j,t}}{\Phi_{j,t}} \le \frac{P^2 H}{P^2 H + P^2} = 1 - \frac{1}{H+1}.$$

It just remains to find an upper bound for H. To this end, we define the notion of *importance ratio.*

Definition 2.4.2.1 (Importance Ratio [27]). *Given a set $\mathcal{M}$ of users, the* importance ratio *of $\mathcal{M}$ is defined as the maximum ratio of unit values of users in $\mathcal{M}$, i.e., $r_{\mathcal{M}} := \max_{i,j\in\mathcal{M}} \frac{\rho_j}{\rho_i}$.*

In this chapter, we assume that the importance ratio does not grow with the number n of users. We have:

Lemma 2.4.2.1. *Assume that at each time slot t, the unit values of active users are normalized and add up to 1. Then,*

$$\rho_j \geq \frac{1}{n + r_{\mathcal{M}_t} - 1}, \quad \forall j \in \mathcal{M}_t.$$

Proof. Let $\rho_{\max,t}$ and $\rho_{\min,t}$ denote the maximum and minimum unit values at t, respectively. Recall that by definition, $r_{\mathcal{M}_t} = \frac{\rho_{\max,t}}{\rho_{\min,t}}$. To prove the lemma, it suffices to derive a lower bound on $\rho_{\min,t}$. Observe that the minimal value of $\rho_{\min,t}$ occurs when there are $n_t - 1$ users with unit value $\rho_{\min,t}$ and one user with $\rho_{\max,t}$. It then follows that

$$(n_t - 1)\rho_{\min,t} + r_{\mathcal{M}_t}\rho_{\min,t} = 1,$$

since unit values are normalized. Using $n_t \leq n$ gives the desired result. □

Having $y_{j,t} + \sum_{i=1}^{H} y_{i,t} \leq K$ and $R_{i,t} \geq K$ for all $i \in \mathcal{H}$, we get $y_{i,t} = \rho_i P$ for all $i \in \mathcal{H}$. Using Lemma 2.4.2.1, we get $\frac{HP}{n+r_{\mathcal{M}_t}-1} + \rho_j P \leq K$, thus giving

$$H \leq \frac{K(n + r_{\mathcal{M}_t} - 1) - P}{P} \approx \frac{nK - P}{P} = U - 1.$$

Here, we made the approximation based on the fact that $r_{\mathcal{M}_t}$ is a constant and n is large. Therefore,

$$\frac{L_{j,t}}{\Phi_{j,t}} \leq \frac{U-1}{U}.$$

Finally, applying Theorem 2.4.1.1 we conclude that WFair is $(2 - \frac{1}{U})$-competitive. □

2.4.3 WRand Analysis (Proof of Theorem 2.3.2.1)

To analyze competitive ratio of WRand, we assume an *oblivious adversary* model [64], which is reasonable in practical scenarios. Recall that an oblivious adversary has complete knowledge about the algorithm's code but has no information about the random choices made by the algorithm during its execution.

Let $\mathcal{A}$ = WRand and for brevity, in the rest of the proof, omit the dependence of $y_{j,t}^{\mathcal{A}}$ on $\mathcal{A}$ for all j and t (so $y_{j,t} := y_{j,t}^{\mathcal{A}}$). Fix an optimal solution, and consider a user j and a time slot t such that $j \in \mathcal{S}_{\text{Opt},t}$. Without loss of generality, we make the following assumptions:

(i) Using a similar argument as in the proof of Theorem 2.3.1.1 and to consider a worst-case scenario, we assume that j is not completed by WRand at t and slot t is saturated.

(ii) We assume that $y^{\star}_{j,t} > y_{j,t}$, since otherwise $L_{j,t} = 0$.

(iii) If $j \in \mathcal{S}_{\mathcal{A},t}$, then $R_{j,t} > y^{\star}_{j,t}$. This is because in the otherwise case (i.e., $R_{j,t} \leq y^{\star}_{j,t}$), we get $R_{j,t} \leq K$ and considering the fact that WRand allocates the maximum feasible resource to selected users, then j should be finished at t and thus $\Delta_{j,t} = 0$ and subsequently $L_{j,t} = 0$.

Given that $R_{j,t} > y^{\star}_{j,t}$, we get $\Delta_{j,t} = \min\{R_{j,t}, y^{\star}_{j,t} - y_{j,t}\} = y^{\star}_{j,t} - y_{j,t}$, and using Eq. (2.6), we have

$$L_{j,t} = \begin{cases} \rho_j y^{\star}_{j,t} & j \notin \mathcal{S}_{\mathcal{A},t}, \\ 0 & j \in \mathcal{S}_{\mathcal{A},t}, \end{cases} \tag{2.7}$$

thus giving

$$\mathbb{E}[L_{j,t}] = \Pr(j \notin \mathcal{S}_{\mathcal{A},t}) \rho_j y^{\star}_{j,t}.$$

If $j \notin \mathcal{S}_{\mathcal{A},t}$, the algorithm prefers another user, name i, with occupied block $\bar{B}_{j,t}$. Note that the case that $\bar{B}_{j,t}$ is allocated to more than one user does not affect the analysis of WRand as in this case the weighted average of users' unit value can be considered. Therefore, the gain $\bar{g}_{j,t} = \rho_i y^{\star}_{j,t}$ and $\rho_j y_{j,t} = 0$. On the other hand, when $j \in \mathcal{S}_{\mathcal{A},t}$, since $\Delta_{j,t} = 0$, then $\bar{g}_{j,t} = 0$. Therefore, using Eq. (2.3), we can calculate $\Phi_{j,t}$ as

$$\Phi_{j,t} = \begin{cases} \rho_i y^{\star}_{j,t} & j \notin \mathcal{S}_{\mathcal{A},t}, \\ \rho_j y^{\star}_{j,t} & j \in \mathcal{S}_{\mathcal{A},t}, R_{j,t} \geq y^{\star}_{j,t}, \\ \rho_j R_{j,t} & j \in \mathcal{S}_{\mathcal{A},t}, R_{j,t} < y^{\star}_{j,t}. \end{cases}$$

Using (iii), we ignore the third case and so,

$$\Phi_{j,t} \geq \begin{cases} \rho_i y^{\star}_{j,t} & j \notin \mathcal{S}_{\mathcal{A},t}, \\ \rho_j y^{\star}_{j,t} & j \in \mathcal{S}_{\mathcal{A},t}. \end{cases} \tag{2.8}$$

Let $h : \mathcal{S}_{\text{OPT},t} \times \mathcal{T} \to \mathbb{R}_+$ be a function with $h(j', t') = L_{j',t'}/\Phi_{j',t'}$ if $j' \notin \mathcal{S}_{\text{OPT},t}$ and $h(j', t') = 0$, otherwise. Note that $\Phi_{j',t'} > 0$ as unit values are positive numbers. Using Eqs. (2.7) and (2.8), we obtain

$$h(j,t) \leq \begin{cases} \frac{\rho_j}{\rho_i} & j \notin \mathcal{S}_{\mathcal{A},t}, \\ 0 & j \in \mathcal{S}_{\mathcal{A},t}, \end{cases}$$

and thus,

$$\mathbb{E}[h(j,t)] \leq \frac{\rho_j}{\rho_i} \Pr(j \notin \mathcal{S}_{\mathcal{A},t}).$$

By the design of WRand, the selection probabilities are proportional to the unit values. Hence,

$$\Pr(j \notin \mathcal{S}_{\mathcal{A},t}) = \frac{\rho_i}{\rho_j} \Pr(i \notin \mathcal{S}_{\mathcal{A},t}).$$

Lemma 2.4.3.1. *Let t be a saturated time slot. Then, under $\mathcal{A} =$ WRand,*

$$\Pr(j \notin \mathcal{S}_{\mathcal{A},t}) \leq 1 - \frac{1}{U}, \quad \forall j \in \mathcal{M}_t.$$

Proof. Let $j \in \mathcal{M}_t$ and consider a saturated time slot t. First note that by definition, at least P/k users are selected in time slot t (i.e., $n_t \geq P/k$), where at each round the selected users in previous rounds are excluded from the selection pool. This can be modeled as a hypergeometric distribution with n_t balls, where $\rho_j n_t$ of them are of our interest. Furthermore, the number of draws is P/k and we will succeed if at least one of those $\rho_j n_t$ balls are selected (to simplify the presentation, we assume that $\rho_j n_t, P/k \in \mathbb{N}$). It then follows that the probability that user j is not selected is given by:

$$\Pr(j \notin \mathcal{S}_{\mathcal{A},t}) = \frac{\binom{n_t - \rho_j n_t}{P/k}}{\binom{n_t}{P/k}}.$$

Using Lemma 2.4.2.1, $\rho_j \geq \frac{1}{n + r_{\mathcal{M}_t} - 1} \approx \frac{1}{n}$. Moreover, $\frac{n_t}{U_t} = \frac{P}{k}$ so that

$$\Pr(j \notin \mathcal{S}_{\mathcal{A},t}) \approx \frac{\binom{n_t - 1}{n_t/U_t}}{\binom{n_t}{U_t}} = 1 - \frac{1}{U_t} \leq 1 - \frac{1}{U},$$

which concludes the proof. □

Now applying Lemma 2.4.3.1 gives

$$\mathbb{E}[h(j,t)] \leq \frac{\rho_j}{\rho_i} \frac{1}{\rho_j/\rho_i} \Pr(i \notin \mathcal{S}_{\mathcal{A},t}) \leq 1 - \frac{1}{U}.$$

Applying Theorem 2.4.1.1, we finally conclude that WRand is $(2 - \frac{1}{U})$-competitive. □

2.5 Simulation Results

In this section, we evaluate the *average performance* of the proposed algorithms. Although we provided theoretical bounds for the worst-case performance of our methods, the average case performance is still important. We note that it is possible that an algorithm with poor competitive ratio can beat another algorithm with a good competitive ratio in the average case.

2.5.1 Setup

The default parameter setting, unless otherwise mentioned, is as follows: we consider a charging station in a time period of 16 hours with $T = 16$. The resource constraint at each time slot is 200 kWh. Similarly to [19], we assume that the number of arrivals at each time slot follows a Poisson distribution with a mean of 10. Moreover, the length of availability window of an EV is independent from the others and follows an exponential distribution. For each EV, the maximum charging rate is drawn uniformly at random from the interval $[1, 10]$. The demand of each EV j is sampled uniformly at random from the interval $[\frac{1}{3}K|\mathcal{T}_j|, K|\mathcal{T}_j|]$, and the value v_j is sampled uniformly from the interval $[\frac{1}{2}D_j, 5D_j]$.

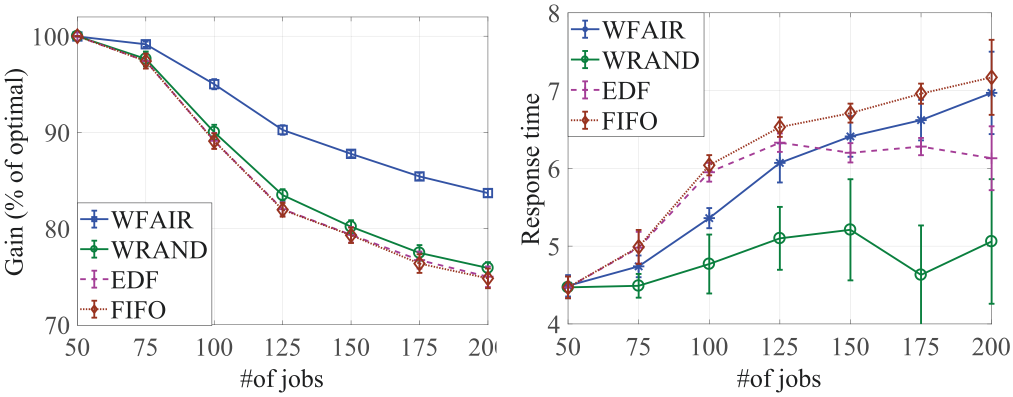
We used Gurobi solver [65] to find the optimal solution and compare the performance of WFair and WRand to the optimal offline solution as well as two other benchmarks:

- *FIFO (First In First Out)*: At each time slot, the priority is given to the EVs with earlier arrival time.
- *EDF (Earliest Deadline First)*: At each time slot, the priority is given to the EVs that are closer to their deadline.

Two major metrics are studied in the simulation: a) the gain of the system which is identified by the objective function in rjsp, and b) average response time of the EVs, defined as the average number of time slots to complete EVs' demand. To compute the response time, we only considered EVs who received their total demand and ignored partially charged EVs.

2.5.2 Impact of Number of Users

In the first scenario in Fig. 2.4, the number of EVs is changed from 50 to 200 while the other parameters are set to their default values as described in Section 2.5.1. When the number of EVs is small, the scheduling problem is less challenging. As it can be seen in Fig. 2.4a, for $n = 50$ the gain of all methods is close to the optimal one. As n increases, the gain falls down for all algorithms. On average, WFair has the best performance by achieving 91% of the optimal while the difference between WRand, EDF and FIFO is minuscule (86%, 85% and 85% of the optimal, respectively). The average response time of all methods increases by increasing number of EVs, where WFair and FIFO show more sensitivity to this change. Another observation is that the response time of WFair is higher than WRand while this is reverse for the gain in Fig. 2.4a.



(a) Gain (b) Response time

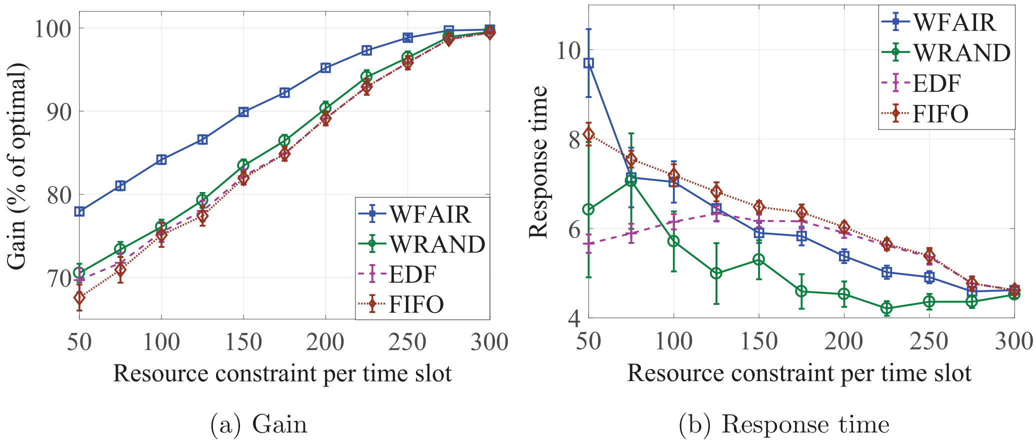
Figure 2.4 – Varying number of EVs.

2.5.3 Impact of Resource Constraint

In the second scenario, the resource constraint at each time slot, P, is varied from 50 kWh to 300 kWh and the methods are compared based on their gain and average response time. By increasing the resource constraint, it is expected that both gain and response time of the algorithms improve. The reason is that with more available resources, EVs have not to wait too long to be allocated. Besides, more EVs can be served at each time slot. This is observable in Figs. 2.5a and 2.5b. Similarly to the scenario of Fig. 2.4, WFair outperforms WRand in terms of gain while there is a small gap between WRand, EDF, and FIFO. When the available resource is sufficient to easily serve all EVs (at $P = 300$), the gain of all methods converges to the optimal gain.

2.5.4 Confirming the Theoretical Bounds

The analysis in Section 2.3.1 demonstrates that the performance of WFair should not fall down its competitive ratio under any input scenario. To verify, we generated 50 random scenarios with $n = 100$, $P = 200$ and set $K = 5$ which gives $U = 2.5$ and the competitive ratio of 1.6. Then, we compared the WRand algorithm with the optimal solution in each single scenario and plotted the result in Fig. 2.6. It can be observed that the gain of WFair is *always* significantly better than the worst-case gain suggested by the competitive ratio. The simulation result here confirms the theory.



(a) Gain (b) Response time

Figure 2.5 – Varying resource constraint.

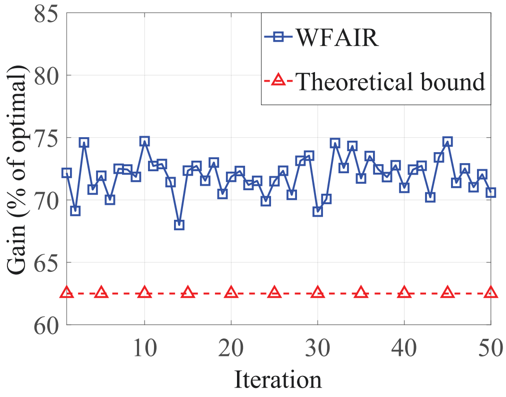


Figure 2.6 – Tracking worst-case performance in 50 random scenarios for WFair.

2.6 Conclusion

This chapter tackled deadline constrained job scheduling problem with its application in electric vehicles. Two deadline-oblivious online algorithms (one deterministic and one randomized) have been developed and their performance are analyzed by a new proof technique which can be used to find upper bound for competitive ratio of a class of algorithms designed for the studied problem. Under realistic scenarios where the demand-to-supply ratio is not too high, the proposed algorithms improve the state of the art result. Further research should be conducted on the scheduling algorithms which can utilize deadline information of the users. Moreover, the proposed proof technique could be extended to support a wider range of problems.

Chapter 3

Mechanism Design in Single Station Scenario

Contents

3.1 Introduction

With the increase in environmental concerns related to carbon emission, and rapid drop in battery prices (e.g., 35% drop in 2015 [66]), the market share of electric vehicles (EVs) is rapidly growing. Bloomberg predicts that 2020s will be the decade of EV [66]. Also, Gartner [67] reports that the global EV market share boosts up to around 10% and 20% by 2020 and 2030, respectively.

The growing number of EVs along with the unprecedented advances in battery capacity and technology results in drastic increase in the total energy demand of EVs. This large charging demand makes the EV charging scheduling problem challenging. An apparent challenge is that even with taking the advantage of deferrable property of charging demands and performing proper scheduling, the aggregate demand might be beyond the tolerable charging rate of the station, given physical constraints of charger devices and transformers [2]. For example, the power capacity of a transformer in North America is limited to 25 kVA [3]. Furthermore, in practice, EVs arrive to charging station in online fashion and the charging station has no information about the arrival and demand of the future EVs. This makes the charging scheduling even more challenging.

In the recent years, the EV charging scheduling problem has attracted much attention from the research community [41,42,45,68–74]. Several studies [41,42,45,69,70,72,73] have tackled different scenarios of EV charging scheduling problems (with different objectives and set of constraints) in online scenario. However, none of the above works, explicitly formulate the problem considering the capacity constraint of the stations. The problems studied in [75, 76] have considered peak shaving in EV charging problem by trying to minimize the peak demand. However, the aggregate EV charging requests might be too large such that even with the goal of minimizing the peak demand, the total demand at some slots is beyond the charging station's capacity. Consequently, this approach fails to guarantee respecting the capacity of the charging station.

In this thesis, we focus on a promising alternative advocated in the recent studies [29,77–79], where the limited capacity of station is incorporated as a constraint in the underlying problem. More specifically, we study online EV charging scheduling, where the EVs arrive at charging station at different times in online manner, and the station has no information about future arrivals. Upon arrival of an EV, it announces its departure time (or deadline), charging demand, maximum instantaneous charging rate, and (potentially) willingness to pay. The goal is to schedule the charging of EVs, such that the social welfare (defined precisely in Sec. 3.2) is maximized, and the charging capacity of the station is respected.

In addition to the inherent challenge raised by the need for online solution design [80], we aim to tackle two other challenges as follows:

(1) Online scheduling with on-arrival commitment. Enforcing capacity constraint may result in partial or no charging of some EVs. In a proper design, the scheduling mechanism must provide *on-arrival commitment* for the EVs, meaning that the mechanism must notify each EV upon receiving its charging demand whether or not it can receive (entirely or partially) the requested demand by the submitted departure time. Without on-arrival commitment, at departure time, an EV may realize that its charging request is not fulfilled, which definitely degrades user satisfaction. Providing on-arrival commitment, however, is challenging in online setting since the scheduler has no information about the demand profile of future arrivals, and giving commitment to the arrived EVs may come at the expense of loosing more valuable EVs in future, even with "re-optimizing" the offline solutions.

None of the related designs [17,21,29,77], to the best of our knowledge, can support on-arrival commitment. The authors in [21] focus on scheduling of deadline-constrained jobs and propose an algorithm that commits to finish a job only once it begins to process it, that might not be upon the arrival. The study is extended in [17] for a truthful scheduling. The online algorithm in [77] commits to charge EVs in arbitrarily time after their arrival. In [29], a competitive online algorithm with on-arrival commitment is proposed for deadline-constrained jobs. However, instantaneous charging rate limit of EV batteries hinders direct application of the design in [29] to EV charging scheduling problem.

(2) Strategy-proof and group-strategy-proof scheduling design. The second challenge is a highly desired feature in social maximization problems which tries to propose mechanisms that are robust against selfish users and groups. Generally speaking, algorithmic mechanism design [7] is a field of game theory, that tries to devise *truthful* (also known as strategy-proof) mechanisms such that it is guaranteed that reporting true values is the best strategy for the players (EVs in our problem) regardless of the behavior of the others. Group-strategy-proofness is a natural generalization of strategy-proofness that tries to guarantee that not only truth-telling is the dominant strategy for individual players, but also, no group of players can improve the utility of at least one member of the group by lying, when the values of the other players are fixed. Studies in [17,77,81] analyze the strategy-proofness of their scheduling mechanisms, however, their algorithms fail to guarantee group-strategy-proofness of the scheduling algorithms. To the best of our knowledge, there is no scheduling mechanism design that can provide full incentive (strategy-proofness and group-strategy-proofness) for EV charging scheduling problem.

Putting together the above challenges, we aim to propose (group)-strategy-proof online EV charging scheduling problem with on-arrival commitment. Toward this, we make the following contributions:

▷ In Section 3.2, we formulate EV charging scheduling problem to maximize social welfare of the users, with the charging rate capacity constraint of the station and instantaneous charging rate constraint of EVs. Even though the problem is a linear one, it is coupled with the time, thereby challenging to solve in online manner.

▷ In Section 3.3, we propose a simple, yet effective online scheduling algorithm (sCommit) that addresses the first challenge and provides on-arrival commitment for the EVs. The sCommit analyzes the recent demands as a clue to make scheduling and commitment decisions. The sCommit relies only on the information of available EVs and has no assumptions on the probabilistic modeling of future arrivals.

▷ In Section 3.4, we tackle the second challenge and propose tCommit that extends the sCommit to guarantee strategy-proofness. By illustrative examples, we demonstrate that if users collude, tCommit cannot guarantee the group-strategy-proofness. In Section 3.5, we design another algorithm (gCommit) that guarantees group-strategy-proofness. To the best of our knowledge, this is the first work that studies developing group-strategy-proof algorithms in the context of EV scheduling problems.

▷ In Section 3.6, we analyze the performance of the proposed algorithms. In particular, we prove that there is no online competitive algorithm with on-arrival commitment for the problem, i.e., no online algorithm can simultaneously provide on-arrival commitment and performance guarantee as compared to the non-committed offline optimum. However, we demonstrate that in a special case that the charging commitment is excluded from the definition of the social welfare (defined in Section 3.2), our proposed algorithms are 2-competitive with optimal offline solution.

▷ In Section 3.7, we extend our basic algorithm, sCommit, when the charging station has partial access to the future charging demands.

▷ In Section 3.8 and through extensive simulations, we evaluate the efficiency of the proposed algorithms and compare them to the optimal offline solution. In a representative set of simulations, the sCommit, tCommit, and gCommit achieve respectively 84%, 85%, and 70% of the offline optimum, on average.

3.2 System Model and Problem Formulation

3.2.1 System Model

We consider a time-slotted system model in which the time horizon is divided to T equal length time slots, e.g., 1 hour, denoted by $\mathcal{T} = \{1, 2, \dots, T\}$. There are n EVs (user or player, used interchangeably) denoted by set $\mathcal{N}$.

Definition 3.2.1.1 (Type of each EV). *Each EV i is characterized by its* "type" $\pi_i = \langle a_i, d_i, v_i, D_i, k_i \rangle$ *indicating its arrival time, departure time, value for the user or willingness to pay, charging demand, and maximum charging rate, respectively.*

Table 3.1 – Summary of notations

Notation	**Description**
$\mathcal{N}$	Set of all EVs with $\|\mathcal{N}\| = n$, indexed by i
$\mathcal{N}^t$	Set of available EVs at t
$\mathcal{C}^t$	Set of active EVs at t with $y_i^t > 0$
$\mathcal{W}^t$	Set of active EVs at t with $y_i^t = 0, r_i^t > 0$
T	Number of time slots, indexed by t
$\mathcal{T}$	$\{1, 2, \ldots, T\}$
$\mathcal{T}_i$	$\{a_i, a_i + 1, \ldots, d_i\}$
a_i	Arrival time of EV i
d_i	Departure time of EV i
D_i	Demand of EV i
v_i	Valuation of EV i for receiving its demand D_i
k_i	Maximum charging rate of EV i in kW
r_i^t	Residual demand of EV i at t i.e., $D_i - \sum_{t'=a_i}^{t} y_i^{t'}$
P	Capacity constraint (in kWh) in charging station
C	Number of charger slots in the station
y_i^t	**opt. variable**, The amount that EV i is charged at t
γ_i	**opt. variable**, Commitment given to EV i on its arrival
x_i^t	**auxiliary binary variable**, $x_i^t = 1$, if $y_i^t > 0$ and 0, otherwise

We refer to time interval $\mathcal{T}_i = [a_i, d_i]$ as *availability window* of EV i. At each time slot t in availability window of EV i, the scheduler can set the charging rate of i, denoted by y_i^t, to a value less than or equal to its maximum charging rate, k_i. We also define an auxiliary binary variable x_i^t with $x_i^t = 1$, if $y_i^t > 0$ and $x_i^t = 0$, otherwise.

We assume that for each EV i, its type represent a feasible demand, i.e., we have $D_i \leq k_i(d_i - a_i + 1)$. The maximum charging rate k_i depends on the physical specification of EV's battery[1] (See Table 3.3 for maximum charging rate of popular EV models). The valuation v_i indicates the worthiness of receiving the submitted demand D_i before the departure time d_i. Note that v_i is the willingness of EV i to pay and it is different from actual payment (see Section 3.4 for details).

We study the charging scheduling problem in online setting where the type of an EV is not revealed to the scheduler until it arrives at the station. Also, we do not have any assumptions on the underlying stochastic process of EV arrivals.

At each time slot, the total power that can be flowed to the EVs is limited to a specific

[1] We assume that the maximum charging rate of the EVs' battery is always less than the maximum output power of the installed chargers in the station. In case that this does not hold, k_i is equal to minimum of the two.

amount of P kW, which we refer to it as the capacity constraint throughout the chapter. The capacity parameter P is a system parameter that is identified based on the maximum power that the chargers in the station are able to deliver at any slot [6,69,82], or set by the charging station owner through participating in a demand-response program. Moreover, we denote by C as the number of charging slots available in the station which restricts the number of EVs that can concurrently get charged at each time slot. As a direct consequence of the capacity constraint and limited number of charging slots, it may not be feasible to fulfill all demands of the vehicles within their availability windows. Consequently, some users may leave the charging station with partial or no charging. Hence, in such a scenario, it is crucial to notify the user upon its arrival on how much charging amount is guaranteed during its availability window.

3.2.2 Social Welfare Maximization Problem

In our design, we aim to devise a scheduler that gives on-arrival commitment. More specifically, let $0 \leq \gamma_i \leq 1$ be *commitment degree* assigned by the charging station to EV i at the beginning of time slot a_i. Once the scheduler decides on the commitment degree γ_i, it is committed to deliver *at least* $\gamma_i D_i$ kWh of power before the departure time d_i. The extreme cases are (i) $\gamma_i = 0$ where there is no commitment on the amount of electricity that EV i will receive and, (ii) $\gamma_i = 1$ where it is guaranteed that EV i will receive all its demand D_i before departure.

Deciding on the commitment degree is highly challenging in online setting, since we assume that at each time instance there is no information about the future coming EVs and it is possible to loose opportunity of charging future high valuable EVs because of commitments given in the previous time slots (in Section 3.7, we will relax this assumption by considering the case that some future information is available).

Taking into account the valuation of demands for the EVs and the commitment degrees, we use two criteria (referred to as J_1 and J_2) to measure users' social welfare. The first criteria measures the aggregate value of allocated resources:

$$J_1 = \sum_{i=1}^{n} \frac{v_i}{D_i} \sum_{t \in \mathcal{T}_i} y_i^t. \tag{3.1}$$

Note that in Equation (3.1), we assume that if EV i receives all its demand, i.e., $\sum_{t \in \mathcal{T}_i} y_i^t = D_i$, the value for the user is v_i, otherwise, the value is proportionally calculated based on the amount of resource the EV received as $(v_i \times \sum_{t \in \mathcal{T}_i} y_i^t)/D_i$.

The second criteria, J_2, is defined based on charging commitments given to the users:

$$J_2 = \sum_{i=1}^{n} v_i \gamma_i. \tag{3.2}$$

Definition 3.2.2.1 (Social welfare maximization problem)**.** *Assuming truthful bidding (see Section 3.4), the social welfare in the EV charging scheduling scenario is defined as the aggregate utility of the charging station, i.e., the total payments obtained from the EVs, and the aggregate utility of the users, that is $J_1 + J_2$ (as defined in Equations.* (3.1) *and* (3.2)*) subtracted by their payment (see Equation* (3.4) *for the formal definition of utility of each user). The payments between the charging station and users cancel themselves, hence, the social welfare of the entire system considering utility of both users and charging station is equivalent to $J_1 + J_2$. Consequently, in the social welfare maximization problem, the objective is to maximize $J_1 + J_2$.*

Given the social welfare definition above, we formulate *social welfare maximization problem* (SWMP) as follows:

$$\begin{aligned}
\mathsf{SWMP:}\quad \max \quad & J_1 + J_2 && (3.3a)\\
\text{s.t.} \quad & \sum_{t\in\mathcal{T}_i} y_i^t \le D_i, \quad \forall i \in \mathcal{N}, && (3.3b)\\
& \sum_{t\in\mathcal{T}_i} y_i^t \ge \gamma_i D_i, \quad \forall i \in \mathcal{N}, && (3.3c)\\
& \sum_{i:t\in\mathcal{T}_i} y_i^t \le P, \quad \forall t \in \mathcal{T}, && (3.3d)\\
& \sum_i x_i^t \le C, \qquad \forall t \in \mathcal{T}, && (3.3e)\\
& y_i^t \in [0, k_i], \quad \forall i, t \in \mathcal{T}_i, && (3.3f)\\
& x_i^t = 1, i, t \in \{i \in \mathcal{N}, t \in \mathcal{T}_i, y_i^t = 1\}, && (3.3g)\\
& x_i^t = 0, i, t \in \{i \in \mathcal{N}, t \in \mathcal{T}, y_i^t = 0\}, && (3.3h)\\
& y_i^t = 0, \qquad \forall i, t : t \notin \mathcal{T}_i && (3.3i)\\
& \gamma_i \in [0, 1], \forall i \in \mathcal{N}. && (3.3j)
\end{aligned}$$

The objective function of the SWMP is sum of J_1 and J_2 with $0 \le J_1, J_2 \le \sum_i v_i$. Note that for each EV i, $\sum_t y_i^t$ is the total power received by the EV where according to the charging commitment definition (variable γ_i) and Constraint (3c), it must hold that $\sum_t y_i^t \ge \gamma_i D_i$. The difference $\sum_t y_i^t - \gamma_i D_i$ corresponds to the amount of power that is delivered but was not committed to user i on its arrival.

The optimization variables are charging commitment γ_i for each EV i and its charging rate y_i^t at each slot t in $\mathcal{T}_i$. Note that x_i^t is a function of y_i^t and thus it is an auxiliary optimization variable to facilitate the formulation of the chapter. Constraint (3.3b) restricts the charging of each EV to its charging demand. Constraint (3.3c) ensures the charging

Table 3.2 – Brief description of the algorithms

Algorithm	Description
sCommit	Online scheduling with on-arrival commitment
tCommit	Online scheduling with on-arrival commitment and strategy-proofness
gCommit	Online scheduling with on-arrival commitment and group-strategy-proofness

commitments are adhered to by the scheduler. The capacity constraint is represented in (3.3d), where at each time slot, total power to be allocated is restricted to P kW. Constraint (3.3e) restricts the total number of EVs to get charged at each time slot. Finally, Constraints (3.3g) and (3.3h) restricts the values of auxiliary binary variable as a function of the original optimization variable.

3.3 Online Scheduling Design with On-arrival Commitment

In this section, we propose sCommit as an online scheduling algorithm for the SWMP, assuming that all the EVs report their true values. Extension to the case that promotes truth-telling is addressed in Section 3.4.

Generally speaking, the SWMP problem can be considered as a time-expanded online version of the well-established fractional knapsack problem [83] where the latter can be optimally solved using a greedy algorithm that sorts items based on the unit values and selects the most valuable items until the reaching the capacity of the knapsack. Our problem is more complicated due to (i) expansion over time, and more importantly, (ii) the online nature of the problem, i.e., items arrive in online manner. However, the general ideas in devising our algorithms in this study utilize the similar sorting ideas. An overview of the proposed algorithms in this chapter is given in Table 3.2.

3.3.0.1 The Details of the sCommit in Algorithm 3

The sCommit runs at each time slot and is developed based on two main ideas. First, the EVs with higher value-demand ratio (i.e., v_i/D_i, hereafter, we refer to it as unit value) are in priority. Second, the commitment decision is made based on whether or not (i) the unit value of the new EV is higher than a *threshold*, or (ii) a specific amount of the resource in availability window of the EV is available.

A high level description of the sCommit and its truthful version, tCommit, is given in Fig. 3.1 using a flowchart. As we will explain later in Section 3.4, the only difference between sCommit and tCommit is in SetGamma sub-procedure. The details of the

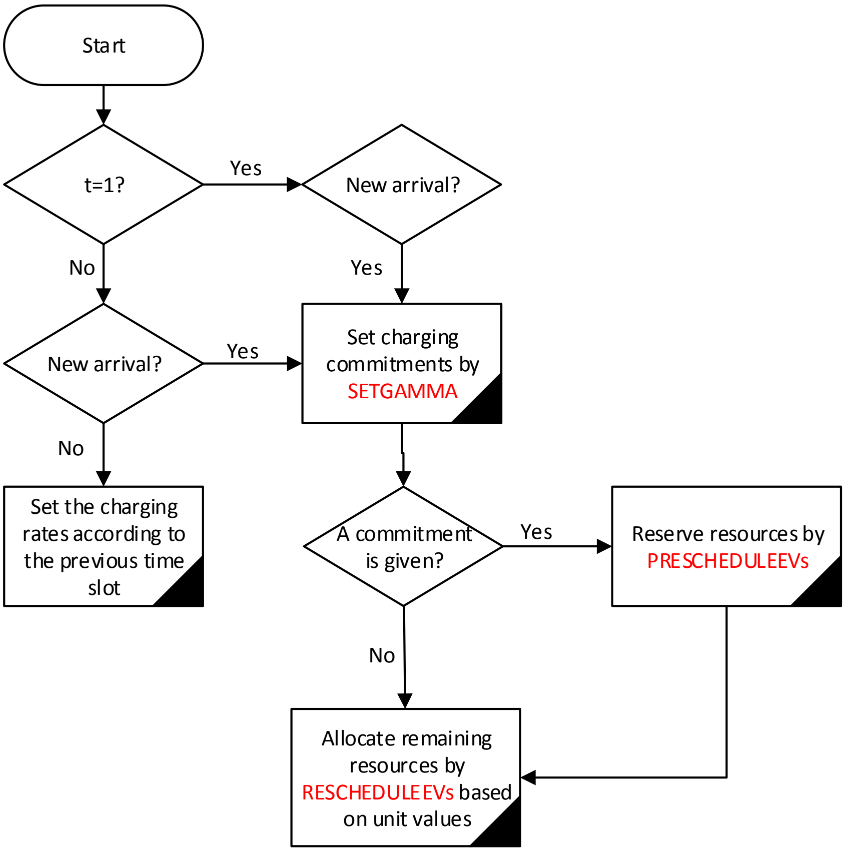

Start
t=1?
Yes
New arrival?
No
Yes
New arrival?
Yes
Set charging commitments by SETGAMMA
No
Set the charging rates according to the previous time slot
A commitment is given?
Yes
Reserve resources by PRESCHEDULEEVs
No
Allocate remaining resources by RESCHEDULEEVs based on unit values

Figure 3.1 – sCommit and tCommit's charging mechanism at time slot t. In case of new arrivals, the main algorithm calls SetGamma to assign charging commitments. The promised resources (if any) are then reserved by PreScheduleEV. Any remaining resource at time slot t will be allocated then by ReScheduleEVs.

Algorithm 3: sCommit: $\forall t \in \{1, \ldots, T\}$

```
   Input: EVs to arrive on the fly, capacity constraint $P$
   Output: A feasible charging scheduling
 1 $\mathcal{N}^t \leftarrow \{i \in \mathcal{N} : t \in \mathcal{T}_i\}$
 2 $r_i^t \leftarrow D_i - \sum_{u=a_i}^{t} y_i^u, \forall i \in \mathcal{N}^t$
 3 $\mathcal{W}^t \leftarrow \{i \in \mathcal{N}^t : y_i^t = 0, r_i^t > 0\}$
 4 $\mathcal{C}^t \leftarrow \{i \in \mathcal{N}^t : y_i^t > 0\}$
 5 if there are new arrivals then
 6     $\mathcal{N}^t \leftarrow$ Sorted list of new arrived EVs in a non-increasing order of their unit
         value (i.e., $\frac{v_i}{D_i}$)
 7     foreach EV $i \in \mathcal{N}^t$ do
 8         $\gamma_i \leftarrow$ SetGamma($i$)
 9         if $\gamma_i > 0$ then
10             $\mathcal{C}^t \leftarrow \mathcal{C}^t \cup \{i\}$
11             PreScheduleEV($i$)
12         else
13             $\mathcal{W}^t \leftarrow \mathcal{W}^t \cup \{i\}$
14     if $\mathcal{W}^t \neq \emptyset$ then
15         ReScheduleEVs($\mathcal{N}^t$)
16 else
17     if $t > 1$ then
18         foreach EV $i \in \mathcal{N}^t$ do
19             if $y_i^t$ is not set yet (by PreScheduleEV) then
20                 $y_i^t \leftarrow \min\{y_i^{t-1}, r_i^t\}$
21                 Update $r_i^t$
```

sCommit are as follows. In Lines 5-6 and given that there are some new arrivals at the current slot, the algorithm first sorts the new EVs in a non-increasing order of their unit values. Then, it selects one EV i at a time and decides on the commitment value (i.e., γ_i) by calling the SetGamma procedure (see Section 3.3.0.2 for details) in Line 8 and provided that $\gamma_i > 0$, the algorithm reserves the promised resources (Line 11) by calling the PreScheduleEV procedure (see details in Section 3.3.0.3).

If there is no new arrival at the current slot, in Line 20, the sCommit sets the charging rate of all EVs with $\gamma_i = 0$ to the same amount of the previous time slot (or less if an EV needs less power to finish its charging). After this step, it is possible that the aggregate charging amount of EVs is less than the capacity, i.e., $\sum_{i \in \mathcal{N}^t} y_i^t < P$. In this case, the procedure ReScheduleEVs is called (Lines 14-15) to allocate the remaining resources to the EVs based on their unit value (details in Section 3.3.0.4). Note that in the ReScheduleEVs, provided that there are enough resources, the scheduler allocates resources to some EVs which is beyond its commitment.

3.3.0.2 The Details of the SetGamma in Algorithm 4

The SetGamma runs for each EV i upon its arrival and sets the commitment variable γ_i to a non-zero value if there is enough resource and one of the following conditions are satisfied: (i) less than α fraction of the resources in the availability window of EV i are allocated (or reserved), i.e., $\sum_{t \in \mathcal{T}_i} \sum_j y_j^t \leq \alpha(d_i - a_i + 1)P$; and (ii) the unit value of EV i (i.e., v_i/D_i) is more than average unit value of the previous EVs available in interval $[t - \Delta, t], \Delta \geq 0$ (index these EVs by j) with $\gamma_j = 1$. Here, Δ and α are parameters to be set by scheduler and affect the performance of the algorithm. As Δ grows, the algorithm looks to more historical data of the previous EVs. Intuitively, rule (ii) states that if a set of users with average unit value of ρ_{avg} deserve full charging commitment, a user with unit value greater than ρ_{avg} deserves some degree of charging commitment as well. In simulations, we investigate the impact of these design parameters on the performance of the algorithms.

3.3.0.3 The Details of the PreScheduleEV in Algorithm 5

If the sCommit gives charging commitment to EV i (i.e., $\gamma_i > 0$), then procedure PreScheduleEV is called to reserve resources for EV i. The reservation policy applied by PreScheduleEV is to charge the EV with the maximum possible charging rate starting from arrival slot a_i.

3.3.0.4 The Details of the ReScheduleEVs in Algorithm 6

If $\gamma_i = 0$, the type of EV i still could be evaluated for an ordinary uncommitted charging. Therefore, the procedure ReScheduleEVs listed in Algorithm 6 is called to check EV i's eligibility to receive resource at the current time slot. The set $\mathcal{W}^t$ keeps the list of EVs waiting to get charged at time slot t. The ReScheduleEVs evaluates EVs' profile in $\mathcal{W}^t$ for possibility of allocating released resources. The procedure gives priority to the EVs with higher unit values.

3.4 Mechanism Design for Self-Interested Users

In this section, we first represent the EV scheduling scenario as a game model in Section 3.4.1, and then in Section 3.4.2, we extend our algorithm design in the previous section to satisfy the game theoretical properties.

Algorithm 4: SetGamma

```
Input: Profile of EV $i$, parameters $\Delta \in \mathbb{Z}^+$, and $\alpha \in [0, 1]$
Output: $\gamma_i$
$\gamma_i \leftarrow 0, s \leftarrow 0$
for each $t$ in $\mathcal{T}_i$ do
    if $\sum_i x_i^t < C$ then
        $s \leftarrow s + \min\{k_i, P - \sum_{j \in \mathcal{N}^t} y_j^t\}$
if $\sum_{t \in \mathcal{T}_i} \sum_{j \in \mathcal{N}^t} y_j^t \leq \alpha(d_i - a_i + 1)P$ then
    $\gamma_i \leftarrow \min\{1, s/D_i\}$
else
    $\mathcal{A}_j \leftarrow \big\{j \in \mathcal{N} : \mathcal{T}_j \cap [a_i - \Delta, a_i] \neq \emptyset \text{ and } \gamma_j = 1\big\}$
    $\mathsf{th} \leftarrow \frac{\sum_{j \in \mathcal{A}_j} v_j/D_j}{|\mathcal{A}_j|}$
    if $\frac{v_i}{D_i} > \mathsf{th}$ then
        $\gamma_i \leftarrow \min\{1, s/D_i\}$
```

Algorithm 5: PreScheduleEV

```
Input: EV $i$ to be scheduled for charging
Output: Charging plan for EV $i$
$R_i \leftarrow \gamma_i D_i$
$t \leftarrow a_i$
while $R_i > 0$ do
    if $\sum_i x_i^t < C$ then
        $r \leftarrow \min\{k_i, R_i, P - \sum_{j \in \mathcal{N}^t} y_j^t\}$
        $y_i^t \leftarrow r$
        $R_i \leftarrow R_i - r$
        $t \leftarrow t + 1$
```

3.4.1 Formal Game Model

In the previous section, we assumed that the users report their type (see Definition 3.2.1.1) truthfully. However, in reality a self-interested user may misreport his preference to increase his own *utility*. Such scenario can be modeled as a game, with the players (denote by $\mathcal{P}$) being EVs and the aggregator (charging station). Denote by $\pi_i = \langle a_i, d_i, v_i, D_i, k_i \rangle$ and $\widehat{\pi}_i = \langle \widehat{a}_i, \widehat{d}_i, \widehat{v}_i, \widehat{D}_i, \widehat{k}_i \rangle$ the true and reported types of user i, respectively. We consider direct revelation mechanisms where each user submits its type $\widehat{\pi}_i$ chosen from set $\mathcal{S}$ of all possible types. Then, a *mechanism* denoted by $(\mathcal{A}, p)$ is composed of an *allocation rule* (a.k.a *social choice rule*) $\mathcal{A} : \mathcal{S}^{\mathcal{N}} \to \{0, 1\}^{\mathcal{N}}$ and a payment rule $p : \mathcal{S}^{\mathcal{N}} \to \mathbb{R}^{\mathcal{N}}$. To cope with the selfish

Algorithm 6: ReScheduleEVs

```
Input: $\mathcal{N}^t$
Output: A new charging decision for time slot $t$
Sort EVs in $\mathcal{W}^t$ in a non-increasing order of their unit value
while $(\sum_{j\in\mathcal{N}^t} y_j^t < P) \wedge (\mathcal{W}^t \neq \emptyset)$ do
    $i \leftarrow$ the next EV in ordered set $\mathcal{W}^t$
    if $(\exists j \in \mathcal{C}^t : v_i/D_i > v_j/D_j) \vee \Big[(\sum_{j\in\mathcal{N}^t} y_j^t < P) \wedge (\sum_i x_i^t < C)\Big]$ then
        Pause charging of EVs with lower priority (if necessary) without violating
          charging commitments
        $y_i^t \leftarrow \min\{k_i, r_i^t, P - \sum_{j\in\mathcal{N}^t} y_j^t\}$
```

users and implement desired allocation rule in a strategic setting, the goal in this section is to design mechanisms that are able to satisfy the game theoretical propeties such as to promote truthfulness.

Note that the reported type $\widehat{\pi}_i$ may not be equal to the true type π_i which is private for each user. Similar to the prior works [24, 77], we assume *no early arrivals* and *no late departures*. More formally, we assume $\widehat{a}_i \geq a_i$ and $\widehat{d}_i \leq d_i$ for all $i \in \mathcal{N}$. These assumptions make sense in practical scenarios. Therefore, the strategy space for each user includes any type that satisfies above conditions. Let "$\succeq$" denotes the partial order of types, where

$$\begin{aligned}\widehat{\pi}_1 \succeq \widehat{\pi}_2 \quad \equiv \quad & (\widehat{a}_1 \leq \widehat{a}_2) \wedge (\widehat{d}_1 \geq \widehat{d}_2) \wedge (\widehat{v}_1 \geq \widehat{v}_2) \\ & \wedge (\widehat{D}_1 \leq \widehat{D}_2) \wedge (\widehat{k}_1 \geq \widehat{k}_2).\end{aligned}$$

If $\widehat{\pi}_1 \succeq \widehat{\pi}_2$, we say $\widehat{\pi}_1$ *dominates* $\widehat{\pi}_2$. Generally, $\widehat{\pi}_1 \succeq \widehat{\pi}_2$ if $\widehat{\pi}_1$ is more valuable and easier to handle for charging station compared to $\widehat{\pi}_2$. $\widehat{\pi}_1 \succ \widehat{\pi}_2$ is also defined similarly and equals to $(\widehat{\pi}_1 \neq \widehat{\pi}_2) \wedge (\widehat{\pi}_1 \succeq \widehat{\pi}_2)$. We also define payment rule $p_i(\widehat{\pi}_{\mathcal{N}})$ which determines the payment of user i at departure.

We use a quasi-linear utility function for user i as follows:

$$u_i(\widehat{\pi}_{\mathcal{N}}) = (\gamma_i + \frac{1}{\widehat{D}_i} \sum_{t=a_i}^{d_i} y_i^t)\widehat{v}_i - p_i(\widehat{\pi}_{\mathcal{N}}) \tag{3.4}$$

where $\widehat{\pi}_{\mathcal{N}}$ is the set of all reported types. The maximum utility is achieved when the user receives full commitment (i.e., $\gamma_i = 1$) along with entire demand.

3.4.2 Extending the sCommit to a Dominant Strategy Incentive Compatible Mechanism

To design an efficient mechanism, several desirable properties are required by the underlying game theory model. These properties include *individual rationality* (IR), *budget* *balanced* (BB), *allocative efficiency* (AE), and *dominant strategy incentive compatibility* (DSIC) (a.k.a *truthfulness* or *strategyproofness*), and generally need that the allocation rule meet some specific conditions. In this chapter, we will focus on IR, BB and particularly DSIC properties where the latter is important for practical mechanism design. First, we define the above properties formally.

Definition 3.4.2.1 (Individual Rationality). *Mechanism $(\mathcal{A}, p)$ is "individually rational" if players always get non-negative utility.*

The IR property is important as it ensures that users are not forced to participate.

Definition 3.4.2.2 (Budget Balanced). *Mechanism $(\mathcal{A}, p)$ is "budget-balanced" if the total payment by the players (i.e., including EVs and charging station in our scenario) is zero i.e., $\sum_{i \in \mathcal{P}} p_i(\widehat{\pi}_i) = 0$.*

With the BB property, there are no net transfers in or out of the system.

Definition 3.4.2.3 (Dominant Strategy Incentive Compatibility). *Mechanism $(\mathcal{A}, p)$ is "dominant strategy incentive compatible", "truthful", or "strategyproof" if the best strategy of each user is to adapt the strategy $\widehat{\pi}_i = \pi_i, \forall i \in \mathcal{P}$.*

The truthfulness property ensures that no user can benefit by deviating from its true type. Toward our goal to design a mechanism satisfying the aforementioned properties, we first design a truthful mechanism, then show that it satisfies IR and BB properties, as well.

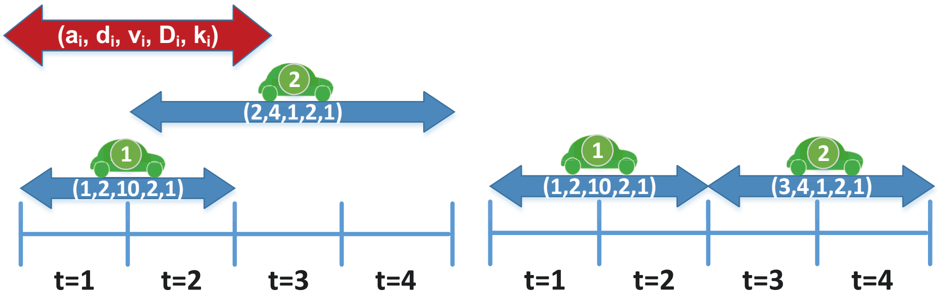
The following definition of monotonicity from a celebrated result by Myerson [84] is the key in the game theoretical analysis in the rest of this section.

Definition 3.4.2.4 (Monotonicity). *Allocation rule $\mathcal{A}$ is* monotone *if for any types $\widehat{\pi}_i$ and $\widehat{\pi}'_i$ where $\widehat{\pi}'_i \succeq \widehat{\pi}_i$, we have $u_i(\widehat{\pi}_{-i} \cup \{\widehat{\pi}'_i\}) \succeq u_i(\widehat{\pi}_{-i} \cup \{\widehat{\pi}_i\})$.*

In above definition, $\widehat{\pi}_{-i}$ denotes all reported types except $\widehat{\pi}_i$ and $u_i(\widehat{\pi}_{-i} \cup \{\widehat{\pi}_i\})$ is utility of user i with profile $\widehat{\pi}_i$ when the reported types of other users are fixed.

Theorem 3.4.2.1. [24] *Let $\mathcal{A}$ be a scheduling mechanism. There is a payment rule p such that the mechanism $(\mathcal{A}, p)$ is strategy-proof if and only if $\mathcal{A}$ is monotone.*

According to [24, 77], for a deterministic allocation mechanism to be monotone, $p_i(\widehat{\pi}_{\mathcal{N}})$ for each user who received full service (in our case, all the demand with full charging commitment) should be equal to its *critical value* which is essentially the minimum $\widehat{v}_i$ that user i can report and receive the same service. In our case where users can be partially allocated (i.e., $\sum_{t=a_i}^{d_i} y_i^t < \widehat{D}_i$) or receive partial charging commitment (i.e., $\gamma_i < 1$) the



(a) (b)

Figure 3.2 – Failure of monotonicity by the SetGamma

payment proportionally calculated according to the received service which is less than the critical value. More formally, let v_i^{cr} be the critical value of user i i.e., $v_i^{\text{cr}} = \min\{v'_i : v'_i \geq 0, \gamma'_i = \gamma_i \wedge \sum_t y'^t_i = \sum_t y^t_i\}$ where γ'_i and $\sum_t y'^t_i$ are respectively the charging commitment and total received power when the reported valuation is v'_i. Then, the payment of user i is defined as follows:

$$p_i(\widehat{\pi}_{\mathcal{N}}) = (\gamma_i + \frac{1}{\widehat{D}_i} \sum_t y_i^t) v_i^{\text{cr}}. \tag{3.5}$$

If user i receives no resources, $p_i(\widehat{\pi}_{\mathcal{N}}) = 0$. In practical cases, it is straightforward to calculate $p_i(\widehat{\pi}_{\mathcal{N}})$ at EV i's departure time by removing the EV from users of interval $[a_i, d_i]$ and running scheduling algorithm again to find the payment value. For the details, we refer to [42, 85]. Given the definitions of $p_i(\widehat{\pi}_{\mathcal{N}})$ in Equation (3.5) and $u_i(\widehat{\pi}_{\mathcal{N}})$ in Equation (3.4), it just remains to prove the monotonicity of the proposed algorithm.

The following example, however, shows that the sCommit is not truthful since the SetGamma as a sub-procedure called in the sCommit is not monotone. In fact, both conditions that the SetGamma checks for giving commitments can be misused by a selfish user to increase its utility.

Example 1: Consider the scheduling problem with $T = 4$ and $P = 1$ as shown in Fig. 3.2a and assume that $\alpha < 1/3$ and $\mathtt{th} > 1/2$ in the SetGamma. At slot 1, EV 1 arrives with type $\widehat{\pi}_1 = \langle 1, 2, 10, 2, 1 \rangle$. According to condition specified in Line 5 of the SetGamma, EV 1 receives charging commitment of $\gamma_1 = 1$. Subsequently, the scheduler reserves all resources at slots 1 and 2 for EV 1 by PreScheduleEV. Later at slot 2, EV 2 arrives with type $\widehat{\pi}_2 = \langle 2, 4, 1, 2, 1 \rangle$. With this type, however, EV 2 cannot pass any of two conditions in the SetGamma. Surprisingly, EV 2 can postpone its arrival to slot 3 (as shown in Fig. 3.2b) without delaying its departure (i.e., $\widehat{\pi}'_2 = \langle 3, 4, 1, 2, 1 \rangle$), and receives charging commitment by passing the first eligibility condition. This violates monotonicity of the SetGamma since $\widehat{\pi}_2 \succeq \widehat{\pi}'_2$.

The reason that the SetGamma cannot provide monotonicity is that the decision on commitment degree is made based on the *fraction* of free resources at the availability window of the EV and not the *actual* amount. It is straightforward to construct another similar example to show that how the monotonicity of the SetGamma can be violated by the second eligibility condition (Line 10 of the SetGamma) as the threshold is calculated based on the recent users' profile and is not a pre-determined value where if a user arrives just after a group of users with low value demand ratio, he is more likely to receive charging commitments.

To overcome these issues, our proposal is to replace the "if condition" in Line 5 of the SetGamma with the following

$$\sum_{t\in\mathcal{T}_i}\sum_{j\in\mathcal{N}^t} y_j^t \le \delta_1 D_i, \tag{3.6}$$

where δ_1 is a constant design parameter. Similarly, we change the "if condition" in Line 10 with

$$v_i/D_i > \delta_2, \tag{3.7}$$

where δ_2 is constant. Note that δ_1 and δ_2 are independent from EVs' arrival and departure time.

Based on the above discussion, we can finally propose a mechanism which is truthful.

Theorem 3.4.2.2. *Let* tCommit *be the online algorithm that replaces the if conditions in Lines 5 and 10 in the* SetGamma *with Equations* (3.6) *and* (3.7)*, and uses utility and payment functions defined in Equations* (3.4) *and* (3.5)*, respectively. Then, the scheduling mechanism* (tCommit, p) *provides DSIC, IR, and BB properties.*

Proof. We first prove the truthfulness property by contradiction by utilizing Theorem 3.4.2.1.

Assume tCommit is not monotone. Therefore, there should exists a scenario that if a user i submits two different types π_i and π_i' with $\pi_i \succ \pi_i'$ it should hold that $u_i(\pi_{-i} \cup \{\pi_i'\}) > u_i(\pi_{-i} \cup \{\pi_i\})$. $\pi_i \succ \pi_i'$ requires that at least one of the following cases hold: a) $(a_i < a_i') \wedge (d_i = d_i') \wedge (v_i = v_i') \wedge (D_i = D_i') \wedge (k_i = k_i')$, b) $(d_i > d_i') \wedge (a_i = a_i') \wedge (v_i = v_i') \wedge (D_i = D_i') \wedge (k_i = k_i')$, c) $(v_i > v_i') \wedge (a_i = a_i') \wedge (d_i = d_i') \wedge (D_i = D_i') \wedge (k_i = k_i')$, d) $(D_i < D_i') \wedge (a_i = a_i') \wedge (d_i = d_i') \wedge (v_i = v_i') \wedge (k_i = k_i')$, and e) $(k_i > k_i') \wedge (a_i = a_i') \wedge (d_i = d_i') \wedge (v_i = v_i') \wedge (D_i = D_i')$. We show that if any of the cases a-e holds then $u_i(\pi_{-i} \cup \{\pi_i\}) \ge u_i(\pi_{-i} \cup \{\pi_i'\})$ which is contradiction. We first prove theorem when exactly one of the conditions a-e holds and then generalize the proof:

a) In this case, EV i with profile π_i arrives to the charging station $a_i' - a_i$ time slots earlier. Note that tCommit allocates resources to an EV either by reserving resource by SetGammaT or through normal competition of users at each time slot. With earlier arrival, the probability of meeting first eligibility condition in SetGammaT increases while it has no effect on the second eligibility condition (See SetGammaT). Therefore, γ_i cannot be decreased in this case but might be increased. If SetGammaT does not set $\gamma_i > 0$, the amount of resources that EV i can receive will be the same for arrival times a_i and a'_i since the value demand ratio of the user did not change and the EV preserves the same priority to receive resources. Hence, in this case $u_i(\pi_{-i} \cup \{\pi_i\}) \geq u_i(\pi_{-i} \cup \{\pi'_i\})$.

b) If a user extend his deadline, it is more likely to receive charging commitment and resources. The argument here is similar to the previous case.

c) When the valuation of a user increases, its value demand ratio increases and the EV can get higher priority to receive charging commitment and resources according to first eligibility condition in SetGammaT and the criterion (i.e., value-demand ratio) used to determine priority of the user. Therefore, a higher reported valuation by a user may increase his utility. Consequently, $u_i(\pi_{-i} \cup \{\pi_i\}) \geq u_i(\pi_{-i} \cup \{\pi'_i\})$

d) The argument for this case is similar to the previous one as the value demand ratio increases by reporting lower demand.

e) In no part of the sCommit algorithm the charging rate of EVs affects the *selection* of EVs to charge. Only when sCommit is going to charge an EV, it sets the charging speed of EVs based on their maximum charging rate. The charging speed is always set to maximum value according to PreScheduleEV and ReScheduleEVs. $u_i(\pi_{-i} \cup \{\pi_i\}) \geq u_i(\pi_{-i} \cup \{\pi'_i\})$ holds as with higher charging speed, an EV may receive more resources by its deadline but not less.

According to the given discussion, all the above cases result in $u_i(\pi_{-i} \cup \{\pi_i\}) \geq u_i(\pi_{-i} \cup \{\pi'_i\})$ which is a contradiction. If two or more of the above cases hold simultaneously, the same contradiction still exists as the scenario can be transformed to multiple scenarios where each one refers to one of the cases a-e.

It remains to show that the proposed mechanism is budget-balanced and individually rational. Each EV owner pays to the charging station according to the payment rule in (3.5). If a user receives no resources its payment is zero. Therefore, the total payment by the EV owners is equal to total money received by the charging station. This ensures budget-balanced property. We now show that $u_i(\widehat{\pi}_i)$ in Equation (3.4) is always non-negative. Using the payment function in Equation (3.5), the utility function is simplified to $u_i(\widehat{\pi}_{\mathcal{N}}) = (\gamma_i + \frac{1}{\widehat{D}_i} \sum_{t=a_i}^{d_i} y_i^t)(\widehat{v}_i - v_i^{\text{cr}})$ which is non-negative as $v_i^{\text{cr}} \leq \widehat{v}_i$. □

3.5 Scheduling Design with Group-strategy-proofness

As mentioned in the previous section, the aim of designing truthful mechanisms is to encourage users to report their true profiles. However, this goal cannot be fully achieved by *individual* strategy-proofness. More specifically, in a truthful scheduling mechanism, it might be possible that a group of users collude to increase the utility of members by being untruthful. In another scenario, it is also possible that a user behaves hostilely against others by reporting its profile in a way that its own utility does not change but at least one other user's utility degrades.

Definition 3.5.0.1. *Scheduling mechanism $\mathcal{A}$ is weak group-strategy-proof if no group of users can collude to increase the utility of all members of the group by deviating from their true values. Also, $\mathcal{A}$ is strong group-strategy-proof if none of the members can obtain higher utility by gaming the system.*

Hereafter, when we write group-strategy-proofness it refers to the strong version as a weak group-strategy-proof mechanism does not provide full incentive for users to not lie. Now we formally define group-strategy-proofness. Let $\mathcal{J} \subseteq \mathcal{N}$ be a coalition of users. In addition, we assume $\pi_i = \widehat{\pi}_i, \forall i \in \mathcal{N} \backslash \mathcal{J}$. Then, group-strategy-proofness states that if

$$u_i(\widehat{\pi}_{-i} \cup \{\widehat{\pi}_i\}) \geq u_i(\widehat{\pi}_{-i} \cup \{\pi_i\}), \forall i \in \mathcal{J},$$

it implies that

$$u_i(\widehat{\pi}_{-i} \cup \{\widehat{\pi}_i\}) = u_i(\widehat{\pi}_{-i} \cup \{\pi_i\}), \forall i \in \mathcal{J}.$$

In other words, if utility of no member of group $\mathcal{J}$ has lessen when the group members lie about their profile then, none of them should end up with a better utility. Truthfulness is a special case of group-strategy-proofness with $|\mathcal{J}| = 1$.

An investigation on the tCommit reveals that it is not group-strategy-proof. As a simple example assume that $\mathcal{J} = \mathcal{N}$ and $\widehat{v}_i = \frac{v_i}{z}, \forall i \in \mathcal{N}, z > 1$. Then, considering that the reported value of all users divided by constant $z > 1$, the charging priorities stay unchanged and all users receive the same service as in the first case. However, according to payment function in Equation (3.5), all users pay $1/z$ of the price they should have paid in the first scenario. In fact, users can decrease their payment arbitrarily by choosing larger values of z.

Unfortunately, it is not straightforward to make the tCommit group-strategy-proof. Rather, we employ the existing group-strategy-proof algorithms in other domains by adjusting them into our model. To this end, we first provide some definitions. Let $\mathcal{Q}^t$ and $\mathcal{Q}_i \subseteq \mathcal{N}$ be set of users who get charged at time slot t and interval $\widehat{\mathcal{T}}_i = [\widehat{a}_i, \widehat{d}_i]$ respectively where $\mathcal{Q}_i = \sum_{t \in \widehat{\mathcal{T}}_i} \mathcal{Q}^t$. Also, $\mathcal{Q}$ is set of all charged EVs in interval $[1, T]$ satisfying

$\mathcal{Q} = \bigcup_{i=1}^{n} \mathcal{Q}_i = \sum_{t=1}^{T} \mathcal{Q}^t$. Let $C(\mathcal{Q})$ be the total payment by the users in set $\mathcal{Q}$. Define $p_i(\widehat{\pi}_{\mathcal{Q}})$ as the payment rule (also known as cost sharing method) such that: (i) $p_i(\widehat{\pi}_{\mathcal{Q}}) = 0$ if $i \notin \mathcal{Q}$, and (ii) $\sum_{i \in \mathcal{Q}} p_i(\widehat{\pi}_{\mathcal{Q}}) = C(\mathcal{Q})$. The key of designing a group-strategy-proof mechanism is to make payment function *cross monotonic* [86]. A payment rule is cross monotonic if for each user $i \in \mathcal{Q}$ it holds that $p_i(\widehat{\pi}_{\mathcal{Q}}) \geq p_i(\widehat{\pi}_{\mathcal{N}})$.

In [86], a general mechanism, called $M(p)$, is designed for a binary system where each user receives the entire service or nothing. Moreover, it is assumed that the service is always available for a user unless he is not willing to pay the corresponding bill. The mechanism $M(p)$ is as follows:

1. $\mathcal{Q} \leftarrow \mathcal{N}$
2. Select an arbitrary user and drop it from $\mathcal{Q}$ if $u_i(\widehat{\pi}_{\mathcal{Q}}) \leq 0$
3. Repeat step 2 until for all users in $\mathcal{Q}$, $u_i(\widehat{\pi}_{\mathcal{Q}}) > 0$

Theorem 3.5.0.1. *[86] Mechanism $M(p)$ is group-strategy-proof for any cross-monotonic payment rule p.*

To have a group-strategy-proof scheduling algorithm for our problem, mechanism $M(p)$ should be justified into our model. The main steps include designing a cross-monotonic payment function and to consider the fact that in our model, some users may not be able to get their service regardless of the amount they are willing to pay. Besides, in our model partial charging is allowed and an EV may receive only a fraction of its demand. Developing a cross-monotonic payment function requires that the total payment $C(\mathcal{Q})$ by the users is known beforehand. However, in our online setting users arrive on-the-fly and the total payment cannot be calculated without having information about other types. To overcome the issue, we develop a time slot based scheduling mechanism. The idea is to design a mechanism which is group-strategy-proof for group of EVs of a single time slot and run the mechanism for all time slots $t = 1, \ldots, T$. In this case, $u_i^t(\widehat{\pi}_{\mathcal{Q}^t})$ denotes the utility of user i at time slot t with the set of charged EVs $\mathcal{Q}^t = \{i : y_i^t > 0\}$, i.e.,

$$u_i^t(\widehat{\pi}_{\mathcal{Q}^t}) = \widehat{v}_i(\frac{y_i^t}{\widehat{D}_i} + \frac{\gamma_i}{\widehat{d}_i - \widehat{a}_i + 1}) - p_i^t(\widehat{\pi}_{\mathcal{Q}^t}), \tag{3.8}$$

and $u_i(\widehat{\pi}_{\mathcal{N}}) = \sum_{t \in \widehat{\mathcal{T}}_i} u_i^t(\widehat{\pi}_{\mathcal{Q}^t})$. Moreover, $p_i^t(\widehat{\pi}_{\mathcal{Q}^t})$ is the price that user i pays for the amount of resource that he receives at time slot t. The payment for user i at each time slot $t \in \widehat{\mathcal{T}}_i$ is defined as

$$p_i^t(\widehat{\pi}_{\mathcal{Q}^t}) = \frac{y_i^t}{\widehat{D}_i} + \frac{\gamma_i}{\widehat{d}_i - \widehat{a}_i + 1} - \frac{|\mathcal{Q}^t|}{c}, \tag{3.9}$$

where $c > 0$ is a constant set by the charging station. In a case that $\mathcal{Q}$ is empty, $p_i^t(\widehat{\pi}_{\mathcal{Q}^t}) = 0$. The total payment by user i is the summation of its payments at different time slots:

$$p_i(\widehat{\pi}_{\mathcal{Q}}) = \sum_{t \in \widehat{\mathcal{T}}_i} p_i^t(\widehat{\pi}_{\mathcal{Q}^t}) = (1/\widehat{D}_i \sum_{t \in \widehat{\mathcal{T}}_i} y_i^t) + \gamma_i - |\mathcal{Q}_i|/c. \tag{3.10}$$

Corollary 3.5.0.1. *The payment function in Equation* (3.10) *is cross-monotonic.*

Proof. Equation (3.9) provides a cross-monotonic definition for payment at each time slot t since as the number of EVs who get charged at the the time slot grows, the payment for each user decreases. Similarly, $p_i(\pi_{\mathcal{Q}})$ which is a summation over time slot payments in Equation (3.9) is cross-monotonic. □

Based on the above discussion, we propose an online allocation algorithm with group-strategy-proofness (gCommit) as listed in Algorithm 7.

Theorem 3.5.0.2. gCommit *is group-strategy-proof.*

Proof. We first show that gCommit applies a special version of mechanism $M(p)$ at each time slot and hence is group-strategyproof when $T = 1$. Then, we extend the proof for $T > 1$.

$T = 1$: Consider a single time slot t. In step 2 of mechanism $M(p)$, users are processed in an arbitrary order and each user receives the service if its utility is greater than its payment. Otherwise, it is dropped from set $\mathcal{Q}$. A special form of this step is employed by gCommit where it processes all users in time slot t according to their marginal valuation and not randomly. As the information about all EVs who are present in the charging station in time slot t is available, it is straightforward to identify users who will get charged at the current time slot denoted by $\mathcal{Q}^t$, based on the sorted list and resource constraint P. This is done in Lines $4 - 6$ of the algorithm. Once the set $\mathcal{Q}^t$ is identified, the next step in mechanism $M(p)$ is to allocate resources to the selected users. The equivalent action in gCommit is done inside the "for" loop in Line 11. Finally, we drop users from set $\mathcal{Q}$ who reach their deadline and did not receive any resources which is equal to last Step of mechanism $M(p)$. Notice that we defined the payment function in Equation (3.5) such that the user utility is always greater than or equal to zero as this is required in mechanism $M(p)$ for selected users i.e, $u_i^t(\pi_{\mathcal{Q}^t}) > p_i^t(\pi_{\mathcal{Q}^t})$. Moreover, $p_i^t(\pi_{\mathcal{Q}^t})$ is cross-monotonic (see proof of Corollary 3.5.0.1). Therefore, it can be observed that gCommit is group-strategyproof for a single time slot t. That is, in each time slot $t = 1, \ldots, T$ if $u_i^t(\pi_{-i} \cup \{\widehat{\pi}_i\}) \geq u_i^t(\pi_{-i} \cup \{\pi_i\}), \forall i \in \mathcal{Q}^t$ then it holds that $u_i^t(\pi_{-i} \cup \{\widehat{\pi}_i\}) = u_i^t(\pi_{-i} \cup \{\pi_i\}), \forall i \in \mathcal{Q}^t$.

$T > 1$: We assume $T = 2$ and prove the theorem for a collusion of two users i, j. The proof is similar when $T > 2$ and with a larger group of users. For notational convenience,

Algorithm 7: gCommit:$\forall t \in \{1, 2, \dots, T\}$

```
Input: Available EVs at time slot $t$, capacity constraint $P$, parameter $c$ and
       parking size $C$
Output: A feasible scheduling for time slot $t$
$\mathcal{Q} \leftarrow \mathcal{N}$, $p_i(\widehat{\pi}_{\mathcal{Q}}) \leftarrow 0$
$\mathcal{M}^t \leftarrow$ the ordered set of EVs available at time slot $t$ such that
  $\widehat{v}_1/\widehat{D}_1 \geq \widehat{v}_2/\widehat{D}_2 \geq \cdots \geq \widehat{v}_{|\mathcal{M}^t|}/\widehat{D}_{|\mathcal{M}^t|}$
$\mathcal{Q}^t \leftarrow \mathcal{M}^t$
$s_i \leftarrow \sum_{j=1}^{i}(\widehat{k}_j - y_j^t)$ for $i = 1, \dots, |\mathcal{M}^t|$
$n_t \leftarrow (\arg\max_{s_i < P} i) + 1$
$\mathcal{Q}^t \leftarrow \{j : j \leq n_t\}$
for $j = 1, \dots, n_t$ do
    if $\sum_i x_i^t = C$ then
        Break the “for” loop
    $\delta \leftarrow \min\{\widehat{k}_j - y_j^t, \sum_{\tau=\widehat{a}_j}^{t} y_j^{\tau} - \widehat{D}_j, P - \sum_{j \in \mathcal{M}^t} y_j^t\}$
    $y_j^t \leftarrow \delta$
    update $x_j^t$
    if $t = \widehat{a}_j$ then
        $\gamma_j \leftarrow y_j^t/\widehat{D}_j$
    $p_j^t(\widehat{\pi}_{\mathcal{Q}^t}) = \frac{y_j^t}{\widehat{D}_j} + \frac{\gamma_j}{\widehat{d}_j - \widehat{a}_j + 1} - \frac{|\mathcal{Q}^t|}{c}$
    if $\widehat{d}_j = t$ then
        $p_j(\widehat{\pi}_{\mathcal{Q}}) = \sum_{t' \in \widehat{\mathcal{T}}_j} p_j^{t'}(\widehat{\pi}_{\mathcal{Q}^{t'}})$
        if $\sum_{t' \in \widehat{\mathcal{T}}_j} y_j^{t'} = 0$ then
            Drop $j$ from $\mathcal{Q}$
```

we define $a_1 = u_i^1(\pi_{-i} \cup \pi_i), a_1' = u_i^1(\pi_{-i} \cup \widehat{\pi}_i), a_2 = u_i^2(\pi_{-i} \cup \pi_i)$ and $a_2' = u_i^2(\pi_{-i} \cup \widehat{\pi}_i)$ to indicate the utilities of user i with its true and reported type in time slot 1 and 2, respectively. b_1, b_1', b_2 and b_2' are also defined similarly for user j. Since gCommit is group-strategyproof for a single time slot, the following deductions are true:

$$(a_1' \geq a_1) \wedge (b_1' \geq b_1) \rightarrow (a_1' = a_1) \wedge (b_1' = b_1) \tag{3.11}$$

$$(a_2' \geq a_2) \wedge (b_2' \geq b_2) \rightarrow (a_2' = a_2) \wedge (b_2' = b_2) \tag{3.12}$$

The total utility of a user is summation of his utilities in different time slots. Therefore, to prove that gCommit is group-strategyproof in general, we should show that the following deduction is true:

$$\begin{aligned} &(a_1' + a_2' \geq a_1 + a_2) \wedge (b_1' + b_2' \geq b_1 + b_2) \rightarrow \\ &\quad (a_1' + a_2' = a_1 + a_2) \wedge (b_1' + b_2' = b_1 + b_2) \end{aligned} \tag{3.13}$$

The first part of the hypothesis (i.e.,$a_1' + a_2' \geq a_1 + a_2$) requires that one of the following cases hold:

$$\begin{aligned} A_1 &\equiv (a_1' \geq a_1) \wedge (a_2' \geq a_2), \\ A_2 &\equiv (a_1' \geq a_1) \wedge (a_2' \leq a_2), \\ A_3 &\equiv (a_1' \leq a_1) \wedge (a_2' \geq a_2) \end{aligned}$$

Similarly, one of the following cases should hold according to the second part of the hypothesis (i.e., $b_1' + b_2' \geq b_1 + b_2$):

$$\begin{aligned} B_1 &\equiv (b_1' \geq b_1) \wedge (b_2' \geq b_2), \\ B_2 &\equiv (b_1' \geq b_1) \wedge (b_2' \leq b_2), \\ B_3 &\equiv (b_1' \leq b_1) \wedge (b_2' \geq b_2) \end{aligned}$$

Considering different combinations of the above cases, the hypothesis of Equation (3.13) can be stated in 9 different forms. We now show that in all of these forms, the conclusion in Equation (3.13) holds.

$A_1 \wedge B_1$: In this case, the hypothesis of Equations (3.11) and (3.12) hold and using the corresponding conclusions we have $(a_1' + a_2' = a_1 + a_2) \wedge (b_1' + b_2' = b_1 + b_2)$.

$A_1 \wedge B_2$: Let's define $p_1 = b_1' - b_1, p_2 = b_2 - b_2'$. Then, from definition of B_2 and hypothesis of Equation (3.13) we have $p_1 \geq p_2$. Therefore, $b_2' + p_1 \geq b_2$ and we also

have $a'_2 \geq a_2$ from A_1. With this observation and deduction (3.12) we can write $(a'_2 = a_2) \wedge (b'_2 + p_1 = b_2)$ which results in $p_1 = 0$ and turns the case to $A_1 \wedge B_1$. Thus, the conclusion in deduction (3.13) holds.

$A_1 \wedge B_3$: The argument here is similar to the previous case.

$A_2 \wedge B_1$: The argument here is similar to the case $A_1 \wedge B_2$.

$A_2 \wedge B_2$: Combining $A_2 \wedge B_2$ with (3.11) we have $(a'_1 = a_1) \wedge (b'_1 = b_1)$. Therefore, hypothesis of (3.13) simplifies to $(a'_2 \geq a_2) \wedge (b'_2 \geq b_2)$ which results $(a'_2 = a_2) \wedge (b'_2 = b_2)$ using (3.12). Thus, the conclusion in deduction (3.13) holds.

$A_2 \wedge B_3$: Let $p_1 = a'_1 - a_1, p_2 = a_2 - a'_2$. Since we have $p_1 \geq p_2$ then, $a'_2 + p_1 \geq a_2$. Similarly, by defining $q_1 = b_1 - b'_1, q_2 = b'_2 - b_2$ we have $b'_1 + q_2 \geq b_1$. With this observation and deductions (3.11) and (3.12) we can conclude $(a'_1 = a_1) \wedge (b'_1 + q_2 = b_1)$ and $(a'_2 + p_1 = a_2) \wedge (b'_2 = b_2)$. This results in $p_1 = q_2 = 0$ and thus the conclusion in (3.13) holds.

The remaining cases $A_3 \wedge B_1$, $A_3 \wedge B_2$ and $A_3 \wedge B_3$ are similar to cases $A_1 \wedge B_3$, $A_2 \wedge B_3$ and $A_2 \wedge B_2$, respectively. □

3.6 On the Competitive Ratio of the Proposed Online Algorithms

The performance of an online algorithm is determined by its *competitive ratio* [80] in which the algorithm is compared to the offline optimal solution in the worst case. Let U_{ALG} and U_{OPT} denote the utility provided by the online algorithm and the offline optimal solution, respectively. Then, the online algorithm is c-competitive for $c \geq 1$, if for any input sequence we have $\frac{U_{\text{OPT}}}{U_{\text{ALG}}} \leq c$. Our proposed online algorithms in this chapter provide on-arrival commitment. A commitment given at time slot t can be fully adhered at the same time slot if $D_i \leq k_i$, or it may require resource reservation in subsequent time slots $t+1, t+2, \ldots$, which we refer to it in this case as *commitment with reservation.* When we talk about commitment it refers to the latter case.

Theorem 3.6.0.1. *There is no competitive online scheduling algorithm* "with reservation" *for SWMP.*

Proof. Assume $\mathcal{A}$ is an online c-competitive algorithm with on-arrival charging commitment with reservation $(c \geq 1)$. We show by a counter-example that c can be arbitrary large. Consider the moment that algorithm $\mathcal{A}$ gives on arrival charging commitment with reservation to an EV. Assume that the commitment is given to EV i with type $\pi_i = \langle a_i, d_i, v_i, D_i, k_i \rangle$ at time slot a_i and the EV received an amount of $y_i^{(a_i)}$ at the first time slot. Let $\delta = \gamma_i D_i - y_i^{(a_i)}$ be the amount of resource that should be reserved at interval $[a_i + 1, d_i]$ for the EV. Also, let Δ be the total amount of resources reserved in interval $[a_i + 1, T]$ for EVs arrived before time slot $a_i + 1$. Note that since algorithm $\mathcal{A}$ is online, it has no information about EVs arriving in interval $[a_i+1, T]$ and we are allowed to set their type arbitrary as an adversarial input for the algorithm. We set the adversarial input as follows. At time slot a_i+1, EV n arrives with type $\pi_n = \langle a_i+1, d_n, v_n, D_n, k_n \rangle$ as the last EV arriving to the charging station with $d_n = T, v_n = L, D_n = P(T - a_i + 1) - \Delta + \delta$ and $k_n = P$ where L is a large enough number to satisfy $\frac{v_n}{D_n} > \frac{v_j}{D_j}, \forall j \in \{1, \dots, n-1\}$. Since EV n has the highest value demand ratio, it should receive all its demand in the optimal solution. Assume that algorithm $\mathcal{A}$ is smart enough to assign all remaining resources (i.e., $P(T - a_i + 1) - \Delta$) to EV n and obtain an objective value of A_1 for the SWMP. However, algorithm $\mathcal{A}$ could obtain better result (denote as A_2) if it does not reserve δ kWh to EV i and instead allocate D_n kWh to EV n to fully charge it. By increasing the value of v_n the performance gap between the two cases (i.e., $A_2 - A_1$) increases as well. If user n sets v_n large enough to satisfy $A_2 > c \times \mathrm{OPT} + A_1$ which is possible as there is no upper limit on v_n, then the competitive ratio of algorithm $\mathcal{A}$ is greater than c which is a contradiction. Therefore, algorithm $\mathcal{A}$ cannot be c-competitive. □

The result in Theorem 3.6.0.1 expresses that regardless of how intelligent the scheduling algorithm with on-arrival commitment is, the adversary can construct a worst-case input, such that the social welfare obtained by the online algorithm as compared to the offline optimum is arbitrarily small. Consequently, this result demonstrates that deciding on the charging commitments is highly challenging and implies that it is not possible to provide an upper bound for the competitive ratio of the sCommit, tCommit and gCommit. However, it is possible to obtain a competitive ratio for the algorithms in special case that they give no charging commitment i.e., J_2 is omitted from the definition of the social welfare. In this case, J_1 which reflects total amount of resources received by the EVs represents the social welfare and SWMP can be simplified as follows:

$$
\begin{aligned}
\textsf{SWMP-R} : \max \quad & J_1 = \sum_{i=1}^{n} \frac{v_i}{D_i} \sum_{t \in \mathcal{T}_i} y_i^t && \text{(3.14a)}\\
\text{s.t.} \quad & \sum_{t \in \mathcal{T}_i} y_i^t \le D_i, \quad \forall i \in \mathcal{N}, && \text{(3.14b)}\\
& \sum_{i : t \in \mathcal{T}_i} y_i^t \le P, \quad \forall t \in \mathcal{T}, && \text{(3.14c)}\\
& \sum_{i} x_i^t \le C, \qquad \forall t, && \text{(3.14d)}\\
& y_i^t \in [0, k_i], \quad \forall i, t \in \mathcal{T}_i, && \text{(3.14e)}\\
& y_i^t = 0, \qquad \forall i, t : t \notin \mathcal{T}_i && \text{(3.14f)}
\end{aligned}
$$

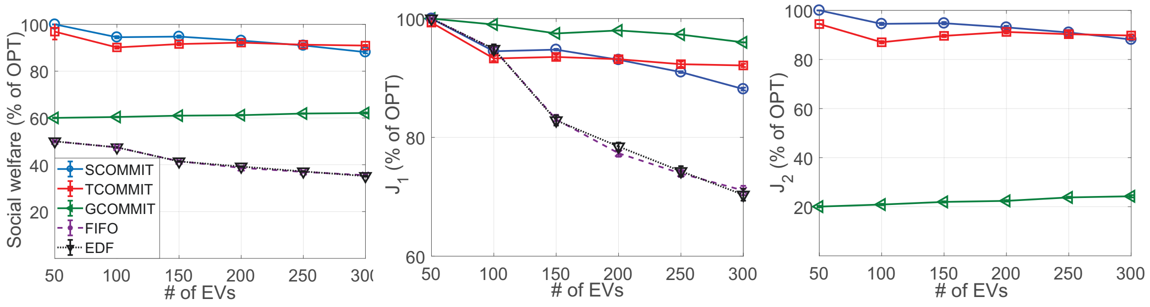
Theorem 3.6.0.2. *Assume that (i) there is always enough charging slot to charge EVs and, (ii)* sCommit*,* tCommit *and* gCommit *are modified such that they set $\gamma_i = 0, \forall i$ while the rest of their code remains intact. Then,* sCommit*,* tCommit *and* gCommit *are* 2*-competitive with optimal offline solution of SWMP-R.*

Proof. The proof follows from fact that when sCommit, tCommit and gCommit provide no charging commitment, they are equal to FirstFit algorithm [30] which is proved to be 2-competitive. In the FirstFit algorithm, at each time slot with a new arrival, the jobs (EVs in our case) are sorted in a non-increasing order of their unit values. Then, the algorithm process jobs according to the sorted list such that for any two jobs i and j with $\rho_i > \rho_j$, j can only receive some resources if it cannot be allocated to i.

If sCommit set $\gamma_j = 0, \forall j$, then the charging decisions are made by ReScheduleEVs. Observe that ReScheduleEVs uses the same sorted list used by FirstFit and follows the same allocation policy. Moreover, tCommit is different from sCommit only in the part that it sets the charging commitment. Therefore, when no charging commitment is given by the algorithms, tCommit is equal to sCommit. Finally, we can observe that gCommit also follows the same approach where it sorts the EVs at each time slot and allocates the maximum resource for each selected EV from the sorted list (Lines $10 - 11$ of Algorithm 7). Therefore, under SWMP-R (formulated in Section 3.6), our proposed algorithms behave as the FirstFit. □

3.7 Extension: Scheduling Under Partial Availability of Future Information

The scheduling algorithms in this chapter are *pure online* as they are designed based on the assumption that zero information about the type of future coming EVs is available. In practice, however, it might be possible that the charging station has some knowledge of the future demands [87]. For example, mobile EVs in a city can submit their charging demand using onboard units (OBUs) before they arrive to the charging station [88]. In this case, the charging station can improve the scheduling by utilizing the amount of the time that it takes for the EVs to drive to the charging station. In a simple form, we can assume that the charging station is always aware of the EVs' type W time slots before their arrival where $0 \leq W \leq T$. With $W = 0$, the scheduling is pure online while $W = T$ represents the offline scenario. Our scheduling algorithms can be modified to adapt to this scenario. We give here explanations to extend sCommit but leave the extension of our truthful algorithms for the future works.



(a) Social welfare (b) J_1 (c) J_2

Figure 3.3 – The impact of the number of EVs on the performance of the proposed algorithms.

Let $\mathcal{A}_t$ be set of EVs which will arrive in time interval $[t, \min\{T, t+W\}])$. Then, to extend sCommit, we add following steps in the beginning of SetGamma and after Line 1: (a) sort EVs in set $\mathcal{A}_t$ in non-increasing order of their unit value where t is the current slot. (b) select one EV (say i) at a time from sorted list and allocate maximum feasible amount of resource in interval $[\max\{t, a_i\}, \min\left\{T, t+W\right\}]$. (c) update γ_i according to allocated resources in $[a_i, d_i]$. (d) set $D_i \leftarrow D_i - \sum_t y_i^t$ and continue running SetGamma as explained in the algorithm box (Lines $2-11$).

The effect of parameter W on the performance of sCommit is investigated in Section 3.8.2.3. Due to the reasons that are explained in Section 3.4, a similar extension cannot be applied to tCommit and gCommit without violating their truthfulness.

3.8 Simulation Results

3.8.1 Settings

We consider charging scheduling of EVs during a day divided to 24 time slots of length 1 hour. In the simulations, we use the battery capacity and maximum charging rate of 10 popular EV models in the market as summarized in Table 3.3. As in [41] and [45], we assume that arrival times follow a Poisson distribution and parking times follow an exponential distribution with the mean arrival and parking duration indicated in Table 3.4. The peak intervals include 08:00-10:00, 12:00-14:00 and 18:00-20:00 which is in accordance to NHTS survey 2009 [41, 89]. Demands are uniform random values from $[k_i(d_i - a_i + 1)/2s, \min\{k_i(d_i - a_i + 1)/s, c_i\}]$ where c_i is the battery capacity of EV i. Based on US national electricity price average ($0.11 per kW) [90] the willingness of a user to pay for 1 kW of electricity is a uniform random number from interval $[0.08, 0.2]$. The parameters Δ and α in the sCommit algorithm are set to 3 and 1, respectively. Moreover, δ_1 and δ_2 in the

Table 3.3 – Characteristics of popular EV models

Model	Max. charging rate	Battery capacity
BMW i3	7.4 kW	22 kWh / 33 kWh
Chevy Spark EV	3.3 kW	19 kWh
Fiat 500e	6.6 kW	24 kWh
Ford Focus Electric	6.6 kW	23 kWh
Kia Soul EV	6.6 kW	27 kWh
Mercedes B-Class Electric	10 kW	28 kWh
Mitsubishi i-MiEV	3.3 kW	16 kWh
Nissan LEAF	3.3 kW / 6.6 kW	20 kWh / 24 kWh
Tesla Model S	10 kW / 20 kW	60 kWh / 100 kWh
Tesla Model X	10 kW /20 kW	60 kWh / 100 kWh

Table 3.4 – EV arrival rates and mean parking times

Time interval	Arrival rate	Mean parking time
08:00-10:00	14	10
10:00-12:00	10	1/2
12:00-14:00	20	2
14:00-18:00	10	1/2
18:00-20:00	20	2
20:00-24:00	10	10
24:00-08:00	0	0

tCommit by default are set to 20 and 0.2, respectively. Recall that when an EV arrives to the charging station, sCommit tends to give a higher charging commitment as α increase from 0 to 1. The same holds for tCommit when δ_1 increases and δ_2 decreases. Therefore, higher values of α, δ_1 and $1/\delta_2$ means that the algorithms give charging commitment blindly.

For the charging station, the default value of capacity constraint is 200 kW and the number of charging slots is 100. In simulation figures, the results are plotted with a 95% confidence level and each data point represents average result of 50 random scenarios. We compared the proposed methods to non-truthful non-committed optimal offline solution labeled as Opt and two classic scheduling algorithms, i.e., *Earliest Deadline First* (EDF) and *First-In First-Out* (FIFO). As the names suggest, EDF always schedules an EV with earliest deadline and FIFO gives the priority to a user which is arrived earlier. To obtain optimal values, we use Gurobi solver [65].

3.8.2 Results

3.8.2.1 The impact of the number of EVs

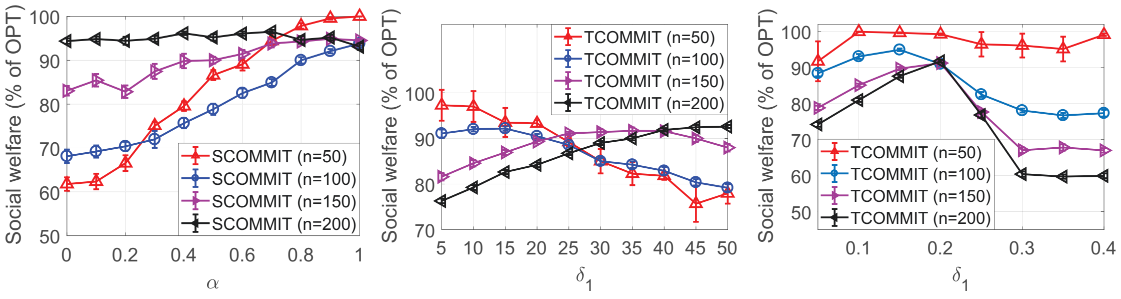
We investigated the performance of our solution when the number of EVs varies. Toward this, we changed the number of EVs from 50 to 300 and reported the results in Fig. 3.3. In Fig. 3.3a, we compared different algorithms based on their social welfare, i.e., the value of objective function in Equation (3.3a). Generally, higher performance is expected when the charging capacity (200 kW) and number of charging slots are enough to charge all or most of EVs. This is because in such conditions, the scheduling problem is less challenging and our algorithms need less intelligence to be close to the Opt. However, as the number of EVs (and thus total demand) grows without increasing capacity constraint and number of slots, more and more EVs loose the opportunity of getting charged. In such scenarios the scheduling is more challenging as less number of EVs can get charged. Consequently, the expansion of solution space makes the scheduling more challenging for the proposed algorithms and the optimality gap slightly increases. This can be observed in Fig. 3.3a-3.3c. Note that FIFO and EDF only consider J_1 (i.e., total valuation of processed demands). Therefore, they cannot provide a good level of social welfare in Fig. 3.3a. Fig. 3.3b reveals that the two algorithms, which perform almost similar, do not scale well as number of EVs increases.

In terms of social welfare, sCommit, tCommit and gCommit are $93\%, 92\%$ and 65% of the optimal solution, on average. The sCommit works better than tCommit when the EV numbers is " $\leq$ 125" and it is reverse for larger values. The reason is that the value of input parameters α, δ_1 and δ_2 in sCommit and tCommit are fixed regardless of the number of EVs. Therefore, depending on the values of the input parameters, either sCommit or tCommit could act better than the other one.

To investigate the strengths and weaknesses of different methods in more details, in Figs. 3.3b and 3.3c we reported the performance of the algorithms in terms of different components of social welfare, i.e., J_1 (as a measure of total power received by the EVs) and J_2 (as a measure of given commitments) as defined in Equations (3.1) and (3.2), respectively. According to the results, the gCommit has very small optimality gap when the comparison is made based on J_1 while in terms of J_2, in Fig. 3.3c, the gap is as large as 76%. This large gap is a result of the gCommit's behavior which puts the priority to provide group-strategy-proofness and gives no commitment that requires reservation (See Section 3.3).

3.8.2.2 The Impact of Design Parameters

In this simulation, we examined the impact of design parameters $0 \leq \alpha \leq 1$ in the sCommit, and $\delta_1 > 0$ and $0 \leq \delta_2 \leq 1$ in the tCommit to find appropriate default parameter setting. In Figs. 3.4a-3.4c, the result for each algorithm is plotted for different number of EVs and different values of input parameters. An immediate observation is that the performance of both sCommit and tCommit are sensitive to design parameters. The impact of parameters, however, is different for different number of EVs. The results in Fig. 3.4a show that when the number of EVs is less than 150, the performance of the sCommit improves as α increases. Observe that with higher values of α the sCommit gives more



(a) Effect of α in sCommit (b) Effect of δ_1 in tCommit (c) Effect of δ_2 in tCommit

Figure 3.4 – The effect of input parameter α on sCommit, and parameters δ_1, δ_2 on tCommit.

commitments. In low load regime with small number of EVs, always enough resources are available to charge all or the majority of EVs. Therefore, the best strategy in such scenarios is to set α to its maximum value. In high load regimes, e.g., $n = 200$, large values of α may degrade the performance as the algorithm gives charging commitments to EVs blindly in the presence of resource shortage. Based on Fig. 3.4a, when $n = 200$, α has an optimal value of 0.6.

In the tCommit, δ_1 is similar to α in the sCommit but it acts reversely. According to Equation (3.6), more charging commitments are given with lower values of δ_1. With a similar justification discussed regarding the effect of α in the sCommit, the results for tCommit in Fig. 3.4b indicate that small (resp. large) values of δ_1 should be used for small (resp. large) number of EVs (approximately, $\delta_1 = n/5$ results in maximum social welfare). Moreover, it can be seen from Fig. 3.4c that $0.15 \leq \delta_2 \leq 0.2$ eventuate to the best results for tCommit. Notice that the online algorithms have no information on the number of EVs. However, one can estimate the number of EVs based on the historical data and set the input parameters accordingly.

3.8.2.3 Performance of sCommit with Partial Access to Future Information

In this section, we investigated the performance of an extended version of sCommit explained in Section 3.7, assuming that at each time slot t the algorithm is aware of the type of EVs which are available in the next W times slots for $W = 0, 1, \dots, 12$. It can be seen from Fig. 3.5 that by increasing W, the sCommit can provide a better scheduling and achieves a higher social welfare. The improvements for $100, 200$ and 300 EVs are 7%, 8% and 11% respectively when W changes from 0 to 12.

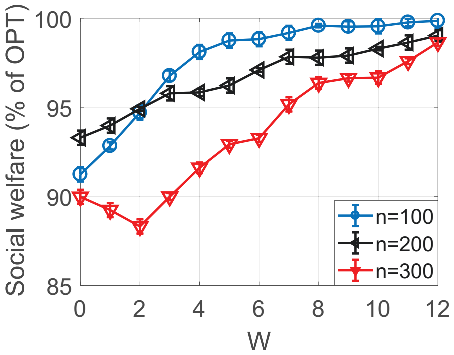


Figure 3.5 – The effect of parameter W in sCommit.

3.9 Conclusion

This chapter studies online EV charging scheduling with on-arrival commitment and group-strategy-proofness. Given the rapid increase in EV charging demand, the aggregate request of deadline-constrained EVs may be beyond the maximum tolerable rate of the charging station. Consequently, some EVs may leave the station by the deadline without receiving their charging request, thereby providing on-arrival commitment is vital for a proper design. We propose several online scheduling algorithms with on-arrival commitment. We then analyze their (group)-strategy-proofness, as a salient feature that simplifies the system design by enforcing users to report their true profiles. Our extensive simulations demonstrate that beside the apparent benefit of on-arrival commitment on improving user satisfaction, the performance of our scheduling algorithms is close to the optimal offline scheduling without commitment.

As a future work, we plan to study the scheduling problem in a network of charging stations where the goal is to achieve a global optimal solution or a near-optimal distributed solution.

Chapter 4

From Single Station to a Network of Charging Stations

Contents

4.1 Introduction

To promote quick adoption of renewable energy sources, electrification of vehicles is a trend that has been globally advocated in recent years. According to a Bloomberg report, EVs will account for more than half of the new car sales by 2040 [66]. Consequently, it is expected that demand from EV charging will constitute a considerable portion of total energy demand. Currently, transportation consumes 29% of total energy in the US, while electricity production consumes 40% [54]. Hence, rapid electrification of the vehicles makes the total electricity demand of EVs significant.

This thesis studies EV charging scheduling in an adaptive charging network (ACN) governed by a single operator in a campus-scale location such as a university, a headquarters, etc. [8]. A notable example is the Caltech ACN [9] where individual charging ports are organized into several *charging stations* (CSs) which are dispersed in a *charging network* with the capability of adaptive charging of the electric vehicles. The problem is different from EV charging scheduling in single station scenarios [4, 5, 47, 47, 54, 56–59] (We refer to [91] for detailed discussions on related work), because of the essential need to respect the aggregate peak demand of the ACN. More specifically, the ACN operator might limit the total power drawn from EVs to control costs [10, 11], reserve the capacity for other loads, and/or participate in demand-response events. The aim of this thesis is to use the deferrable property of EVs and schedule their charging jobs, so as to maximize the revenue of the ACN.

We consider a scenario with multiple EVs where each EV has different availability, charging demand, charging rate limit, and valuation of getting charged. We formulate an online EV charging scheduling problem with the goal of *selecting* and *scheduling* a subset of EVs such that: (1) the charging demand of the selected EVs are (fully or partially) satisfied; (2) the charging rate limit of EV batteries are respected, (3) the global peak constraint of the ACN is satisfied [9]; (4) the local peak constraint of each CS is respected; and finally, (5) the total revenue obtained by the valuation of the EVs is maximized.

There are two main challenges in the design and implementation of scheduling algorithms for EVs satisfying the goals mentioned above. Firstly, the problem calls for online scheduling design. In practice, EVs arrive to the CS in online fashion and the scheduler has no information about the arrival and demand of the future EVs. Secondly, the underlying optimization problem in integral model is NP-hard (see Section 4.4). This is because the problem is a mixed integer linear problem and a "time-expanded" extension of knapsack problem which is known as a classic NP-hard problem. In this thesis, we tackle the challenge of online design by following competitive algorithm design [80] and the challenge of NP-hardness by pursuing approximation algorithm design [92] and make the following contributions:

$\triangleright$ We first consider a *fractional model* (where EVs can be charged partially and the revenue is proportional accordingly) and design an optimal offline scheduling algorithm. We then develop an efficient online algorithm in which no exact or stochastic information about the future EV arrivals is given. Despite its simplicity, the algorithm is proved to be 2-competitive with optimal offline solution, i.e., the revenue of the proposed online algorithm is at least 1/2 of the offline optimum, regardless of input sequence. Even though there are competitive algorithms in the literature for similar problems, to the best of our knowledge, our algorithm is the first 2-competitive algorithm which considers the charging rate limit of EV batteries.

$\triangleright$ We next study the more challenging scenario of the *integral model*, where EVs must receive all their demand to make revenue. We first propose a polynomial-time primal-dual offline approximate algorithm. We analyze the approximation ratio of the algorithm and by strengthening the linear relaxed version of the mixed integer problem [93], we obtain an approximation ratio of $\alpha = 1 + \sum_{j=1}^{m} \frac{p_j}{p_j - q_j} . \frac{s}{s-1}$, where p_j is local peak constraint in station j, q_j is the maximum charging rate of the EVs in station j and s is a slackness parameter. We highlight that when $p_j \gg q_j$ and s is large enough, then $\alpha \approx m + 1$, where m is the number of stations in the ACN. Built on top of the offline algorithm, we devise an online algorithm, and discuss its competitive ratio in special cases.

$\triangleright$ We conduct a set of extensive simulations to evaluate the performance of our proposed algorithms. The results of online algorithms for both integral and fractional settings are close to the offline algorithms (within 89% and 92% for integral and fractional models in a representative scenario). In addition, our algorithm outperforms the existing scheduling algorithm in Caltech ACN [9] by 9% and 3% for integral and fractional revenue models, respectively.

4.2 System Model and Problem Formulation

4.2.1 System Model

We consider a time-slotted system model in which the time horizon is divided to T equal length slots $t = \{1, 2, \ldots, T\}$ (e.g., $T = 24$ with time slots of 1 hour length).

4.2.1.1 Charging Network

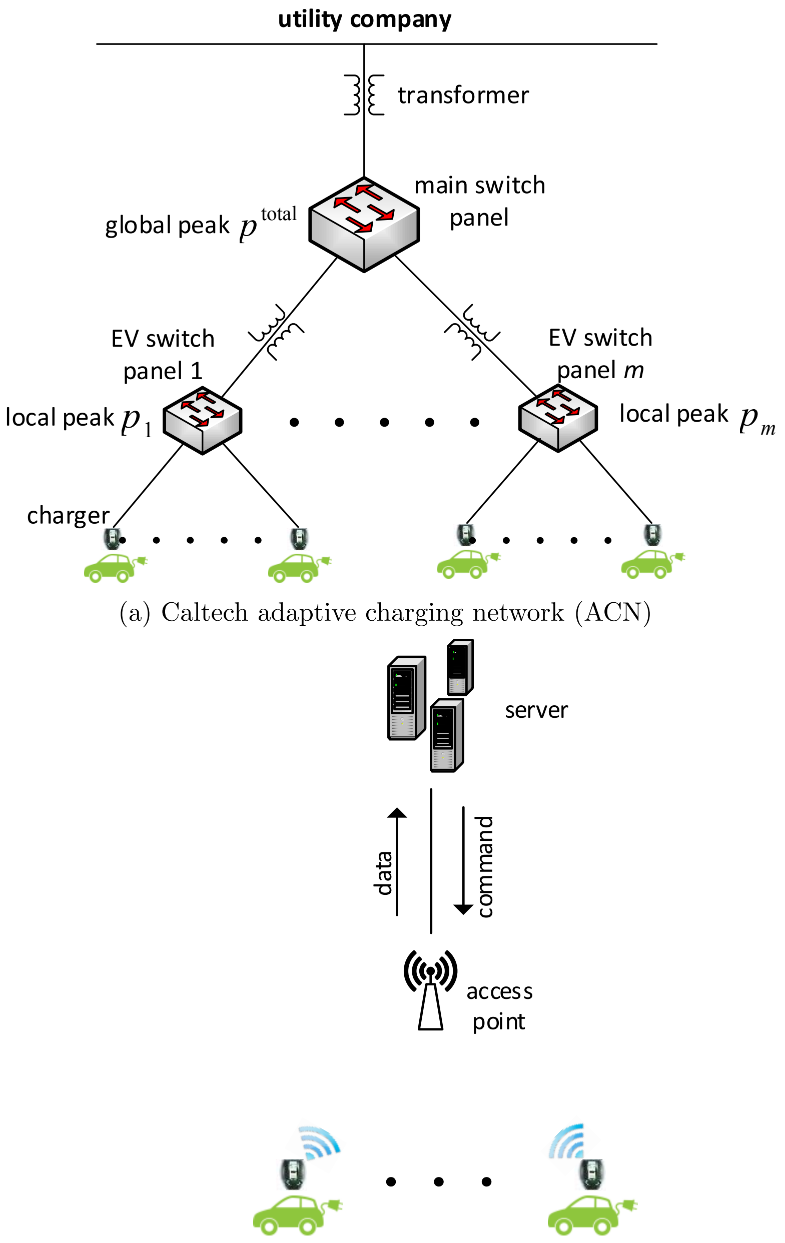
Our charging network model is inspired by the Caltech ACN [9] as illustrated in Fig. 4.1a. In the Caltech EV charging network (located in a parking garage), electricity is distributed through a two-level transformer architecture from a main switch panel to multiple EV

Table 4.1 – Summary of key notations

Notation	Description
T	Number of time slots, indexed by t
m	Number of CSs, indexed by j
n	Number of EVs, indexed by i
a_i	Arrival time of EV i
d_i	Departure time of EV i
D_i	Demand of EV i
v_i	Valuation of EV i for receiving its demand D_i
k_i	Maximum charging rate of EV i
$h(i)$	CS of EV i
q_j	Maximum k_i among all EVs in CS j
p_j	Maximum aggregate charging rate in station j
p^{total}	Maximum aggregated charging rate of all stations
y_i^t	**opt. variable**, The amount that EV i is charged at t

switch panels (2 panels in the current Caltech ACN). Each EV switch panel then is connected to several chargers (≈25 chargers per panel in the Caltech ACN). The total power drawn from the main switch panel by the charging network has a power limit of p^{total}, that is determined by the facility operator to control the costs, reserve the capacity for other loads, and/or participate in demand-response events. In other words, p^{total}, which we refer to it as the *global peak*, hereafter, limits the maximum aggregate EV charging load at each time slot.

We assume that there are m EV switch panels that represent m CSs. In addition to the global peak constraint, each CS j has a capacity constraint on its total power drawn, indicated by p_j, and referred to it as the *local peak* constraint. The value of p_j is determined by the output power limit of the transformers installed between the main switch panel and EV switch panels and could be different for each EV switch panel. It is often observed that the charging demand of different CSs (EV switch panels in Fig. 4.1a) are well below the local peak constraints. To increase the flexibility due to heterogeneous charging demand of CSs, in the Caltech ACN, the global peak constraint of the main switch can be *over-provisioned*, i.e., p^{total} is less than the aggregate local peaks, $(\sum_j p_j \geq p^{\mathsf{total}})$. While this increases flexibility, it also couples the problem of EV charging scheduling across different CSs. Our solutions in this thesis will be centralized ones which can be obtained through communication between the CSs and a central server as illustrated in Fig. 4.1b.



(a) Caltech adaptive charging network (ACN)

(b) Communication model with server

Figure 4.1 – System model [9].

Table 4.2 – Proposed algorithms and their properties.

Alg.	**Revenue**	**Type**	**Optimality**	**Complexity**
FCS	fractional	offline	Optimal	$O(n^2T + nT^2)$
ICS	integral	offline	$\left(1 + \sum_{j=1}^{m} \frac{p_j}{p_j - q_j} \cdot \frac{s}{s-1}\right)$-approximate	$O(nT \log T + n^2T)$
FOCS	fractional	online	2-competitive	$O(n^2T)$
IOCS	integral	online	$b(1 + \frac{p}{p-q} \cdot \frac{s}{s-1})$-competitive, $(m = 1)$	$O(n^2T)$

4.2.1.2 EVs

There are n EVs in the system, indexed by i. EV i is represented by a charging profile $\langle a_i, d_i, D_i, v_i, k_i \rangle$ indicating its arrival time, departure time, charging demand, valuation, and the maximum charging rate, respectively. More specifically, the charging of EV i can be scheduled within its *availability window*, $[a_i, d_i]$. The charging rate at each slot is bounded to k_i, a parameter that depends on the physical constraints of the battery and on-board charger. It is assumed that the charging profile of each EV is feasible with respect to its maximum charging rate and a slackness parameter $s \geq 1$, which is the minimum ratio between the park time of the EV and its minimum charging time, i.e., $D_i \leq k_i(d_i - a_i + 1)/s$. The slackness parameter is imposed to tune the flexibility of the charging scheduling. In extreme case $s = 1$, the flexibility is minimum and the flexibility improves as s increases. We assume that EV owners select their CS, perhaps the nearest to them, and so the assignments are given to the problem. Define $h(i)$ as the CS of EV i. Moreover, q_j denotes the maximum k_i among all EVs in CS j, i.e., $q_j = \max_{h(i)=j} k_i$. Finally, v_i is the revenue of CS given that EV i received its demand D_i before the departure time d_i.

4.2.1.3 Revenue Models

We consider two revenue models: *(i) Fractional revenue model*: In this model, the fractional charging is allowed, i.e., the revenue from each EV is proportional to the fraction of the demand that is fulfilled [5] (see Eq. (4.2)). We tackle this model in Section 4.3. *(ii) Integral revenue model*: In this model, when EV i pays v_i when it is fully charged, otherwise it pays nothing, i.e. no partial revenue is received for partial charging. This is the model that is considered in [56, 57]. We tackle this model in Section 4.4.

4.2.2 Problem Formulation

We formulate an optimization problem to schedule the charging of the EVs with the objective of maximizing total revenue obtained from charged EVs while respecting local and global peak constraints. Note that each revenue model makes the underlying optimization problem fundamentally different. More specifically, the fractional charging model is a linear problem. Integral revenue model, however, turns the underlying charging scheduling problem to a Mixed Integer Linear Program (MILP). The integer nature originates from the 0/1-selection decision on EVs. We formulate Scheduling Problem for Adaptive charging Network (SPAN) under fractional revenue model as follows:

$$
\begin{aligned}
\mathsf{SPAN}: \max \quad & \sum\nolimits_{i=1}^{n} \frac{v_i}{D_i} \sum\nolimits_{t=a_i}^{d_i} y_i^t \\
\text{s.t.} \quad & \sum\nolimits_{t=a_i}^{d_i} y_i^t \le D_i, \qquad \forall i, && (4.1a)\\
& \sum\nolimits_{i=1}^{n} y_i^t \le p^{\mathsf{total}}, \qquad \forall t, && (4.1b)\\
& \sum\nolimits_{i:h(i)=j} y_i^t \le p_j, \qquad \forall t, j, && (4.1c)\\
& y_i^t \le \frac{k_i}{D_i} \sum\nolimits_{t'=a_i}^{d_i} y_i^{t'}, \quad \forall i, t, && (4.1d)\\
\text{vars.} \quad & y_i^t \ge 0, \qquad \forall i, t,
\end{aligned}
$$

where y_i^t is the amount that EV i is charged at slot t. Constraint (4.1a) ensures that the aggregate amount received by EV i is at most the demand D_i. The global and local peak constraints are represented by constraints (4.1b) and (4.1c), respectively.

The constraint (4.1d) enforces the maximum charging rate of EVs. The straightforward way to express this constraint is to simply state that at each time slot t, the charging rate of EV i should be less than or equal to its maximum charging rate, i.e., $y_i^t \le k_i, \forall i, t$. However, for the sake of effective algorithm design for integral model and reducing the integrality gap of the relaxed linear problem, this constraint is strengthened in the form of Eq. (4.1d). Note that in case that the aggregated charging of EV i during its availability window is equal to its demand, i.e., $\sum_{t'=a_i}^{d_i} y_i^{t'} = D_i$, Eq. (4.1d) reduces to the simple form of $y_i^t \le k_i$. This is a natural way in approximation algorithm design to improve the performance of the algorithms under linear-relaxation based design [93].

The SPAN is an extension of formulated problem in [94] where a job scheduling problem in cloud applications is studied. It turns out that the resource allocation problem in cloud systems and the EV charging scheduling problem in a *single station* share similar structure. Indeed, each charging profile in our scheduling problem can be seen as a job in cloud system with a deadline, value, and CPU demand. The SPAN, however, comes with an additional constraint (4.1b), which makes it different from the problem in [94], such that the existing solution does not work in the new setting anymore. More importantly, this thesis studies both offline and online solutions for the problem in both fractional and integral settings, while [94] tackles only the offline integral model.

4.3 Fractional Revenue Model

4.3.1 Overview of Fractional Revenue Model

To develop a solution for the SPAN, we first consider fractional revenue model. In this model, the revenue of the CS from EV i is directly proportional to the amount of resource that the EV is received, i.e.,

$$v_i^{\mathsf{f}} = \min\left\{v_i, \frac{\sum_t y_i^t}{D_i} v_i\right\}, \tag{4.2}$$

where v_i is the gain, if the entire demand D_i is fulfilled and v_i^{f} is the fractional gain.

Under fractional revenue model and without the binary variable of selecting EVs, the underlying problem turns into a linear one. In Section 4.3.2, we propose a simple algorithm with low computational complexity for the fractional model that finds the optimal solution in offline setting. Note that even though linear programs can be solved in polynomial time in general, the complexity of our proposed algorithm is much lower than the general linear program algorithms. Moreover, our proposed offline algorithm applies a valley-filling strategy to reduce the peak. The online algorithm is designed in Section 4.3.3, where no exact information or stochastic modeling of future inputs of the problem is available. Our analysis demonstrates that the proposed online algorithm is a 2-competitive one, i.e., the revenue obtained by the algorithm is at least 1/2 of the offline optimum in worst-case.

4.3.2 Offline Scenario

We refer the proposed algorithm as the fCS and summarize it as Algorithm 8. The fCS works in two phases. In the first phase (Section 4.3.2.1), the algorithm decides on the amount of resource to be allocated to each EV within its availability window and reserves resources accordingly. In this phase, the details of allocation is not known. The actual resource allocation is done in the second phase (Section 4.3.2.2) by setting variables y_i^t.

Before discussing the details of the algorithm, we introduce some notations that facilitate our algorithm design. Let R_i be the amount of resource that is reserved for EV i by the fCS and $I_{t,t'}$ as time interval $[t, t']$. Then, assuming that charging demands are sorted in non-increasing order of their unit values, $A_j^i(t, t')$ is the aggregate residual resource in interval $I_{t,t'}$ at station j assuming that the reservation for EVs 1 to i is accomplished with a dummy $A_j^0(t, t')$ defined as

$$A_j^0(t, t') = (t' - t + 1) \times \min\{p^{\mathsf{total}}, p_j\},$$

that indicates the available resource when no charging request is processed.

Algorithm 8: FCS

```
    Input: $n$ EVs with their profile, local and global peak constraints $p_j, j = 1, \dots, m$
           and $p^{\text{total}}$
    Output: Optimal scheduling under fractional model
 1  Sort charging requests in non-increasing order of their unit values, i.e.,
    $\frac{v_1}{D_1} \geq \frac{v_2}{D_2} \geq \cdots \geq \frac{v_n}{D_n}$
 2  $\mathcal{L} \leftarrow \emptyset$
 3  //Phase I
 4  for $i = 1, \dots, n$ do
 5      $R_i \leftarrow \min\{D_i, \min_{t,t'} A^{i-1}_{h(i)}(t,t'), \forall t,t' : I_{t,t'} \in \mathcal{I}_{a_i,d_i}\}$
 6      if $R_i > 0$ then
 7          $A^{i}_{h(i)}(t,t') \leftarrow A^{i-1}_{h(i)}(t,t') - R_i, \forall t,t' : [t,t'] \in \mathcal{I}_{a_i,d_i}$
 8          $\mathcal{L} \leftarrow \mathcal{L} \cup i$
 9  //Phase II
10  Sort EVs in $\mathcal{L}$ in increasing order of their charging flexibility i.e., $\frac{(d_i - a_i + 1)k_i}{D_i}, i \in \mathcal{L}$.
11  for $i = 1, \dots, |\mathcal{L}|$ do
12      Pick EV $i$ from the sorted list $\mathcal{L}$.
13      feasible $\leftarrow \left(\sum_{t=a_i}^{d_i} \min\{k_i, A^{i-1}_{h(i)}(t,t)\}\right) - R_i$
14      if feasible $< 0$ then
15          Re-allocate previously allocated EVs such that feasible $\geq 0$
16      Arbitrarily allocate $R_i$ to EV $i$ in its availability window
```

Definition 4.3.2.1. *Time interval $[\delta, \delta']$ is a "super interval" for interval $[t, t']$ if $1 \leq \delta \leq t$ AND $t' \leq \delta' \leq T$. Moreover, $\mathcal{I}_{t,t'}$ is the set of all super intervals of interval $I_{t,t'}$ i.e., $\mathcal{I}_{t,t'} = \{[\delta, \delta'] : 1 \leq \delta \leq t$ AND $t' \leq \delta' \leq T\}$.*

The number of super intervals of an interval is at most T^2 and at least one (for interval $[1, T]$). We now explain in detail each phase of the algorithm.

4.3.2.1 Phase I-Reservation

In Line 1, the EVs are sorted in a non-increasing order of their unit values. In Line 5, the FCS processes demand of EV i, picked from top of the ordered list, and sets R_i as the amount to be reserved for EV i which will be allocated in Phase II. In Line 7, the residual resource of all intervals in set $\mathcal{I}_{a_i,d_i}$ decreases by R_i and EV i is added to the set of selected EVs.

Lemma 4.3.2.1. *Provided that for EV i we have*

$$R_i \leq \min\left\{D_i, \min_{t,t'} A_j^{i-1}(t,t'), \forall t,t' : I_{t,t'} \in \mathcal{I}_{a_i,d_i}\right\}, \tag{4.3}$$

then there is a feasible allocation to allocate R_i to EV i in its availability window $[a_i, d_i]$.

Proof. We prove the theorem by induction.

Base case: When $n = 1$, there is only one EV that can be easily scheduled in its interval since all charging profiles represent a feasible demand and all resources are free.

Induction step: Let $k \in \mathbb{Z}^+$ is given and the claim is true for $n = k$ i.e., EVs $1, \dots, k$ can be feasibly scheduled to receive their reserved resources. Now let $n = k+1$. We claim that EV $k+1$ can be feasibly scheduled in its interval. To prove, assume that this claim does not hold. Therefore, there should be at least one interval say $I_{t,t'}$ such that $A_{h(k+1)}^{k+1}(t,t') < 0$ and $A_{h(k+1)}^{k}(t,t') \geq 0$. Then, one of the following cases holds: *a*) $I_{t,t'} \notin \mathcal{I}_{a_{k+1},d_{k+1}}$ and *b*) $I_{t,t'} \in \mathcal{I}_{a_{k+1},d_{k+1}}$.

In case *a*, since $I_{t,t'} \notin \mathcal{I}_{a_{k+1},d_{k+1}}$, reserving resources in interval $I_{a_{k+1},d_{k+1}}$ does not affect remaining resource in $I_{t,t'}$. Therefore, we have $A_{h(k+1)}^{k+1}(t,t') = A_{h(k+1)}^{k}(t,t') \geq 0$.

In case *b*, according to Eq. (4.3), we have $R_{k+1} \leq A_{h(k+1)}^{k}(t,t')$. Also, $A_{h(k+1)}^{k+1}(t,t') = A_{h(k+1)}^{k}(t,t') - R_{k+1}$ which gives $A_{h(k+1)}^{k}(t,t') \geq 0$.

Consequently, in both cases $A_{h(k+1)}^{k}(t,t') \geq 0$ which is a contradiction. Therefore, the original claim holds.

□

In Eq. (4.3), the second term, i.e., $\min_{t,t'} A_j^{i-1}(t,t')$, indicates the minimum remaining resource in all super intervals of interval I_{a_i,d_i}. For $i \geq 1$, $A_j^i(a_i,d_i)$ is defined as follows:

$$A_j^i(a_i,d_i) = \begin{cases} A_j^{i-1}(a_i,d_i) - R_i & j = h(i), \\ A_j^{i-1}(a_i,d_i) & j \neq h(i). \end{cases}$$

The optimal value of $R_i, i = 1, \dots, n$ is set according to the following lemma:

Lemma 4.3.2.2. *Given n EVs sorted in a non-increasing order of the unit values, $v_1/D_1 \geq v_2/D_2 \geq \cdots \geq v$ and the value of R_i, where R_i is set after $R_{i-1}, i = 2, \dots, n$ by Eq. (4.3), then,*

$$R_i = \min\left\{D_i, \min_{t,t'} A_{h(i)}^{i-1}(t,t'), \forall t,t' : I_{t,t'} \in \mathcal{I}_{a_i,d_i}\right\}, \forall i,$$

is the optimal value for R_i.

Proof. By induction.

Base case: When $n = 1$, the claim holds since $R_i > \min\{D_i, \min_{t,t'} A_{h(i)}^{i-1}(t,t'), \forall t,t' : [t,t'] \in \mathcal{I}_{a_i,d_i}\}$ is not feasible and the gain is maximized with maximum value of R_i i.e., $R_i = \min\{D_i, \min_{t,t'} A_{h(i)}^{i-1}(t,t'), \forall t,t' : [t,t'] \in \mathcal{I}_{a_i,d_i}\}$.

Induction step: Assume the claim holds for $n = k, k > 1$ i.e., $R_i^\star = \min\{D_i, \min_{t,t'} A_{h(i)}^{i-1}(t,t'), \forall t,t' : [t,t'] \in \mathcal{I}_{a_i,d_i}\}$, is the optimal value for $R_i, i = 1,\ldots,k$. Now let $n = k+1$. We claim that $R_{k+1}^\star = \Gamma$ where $\Gamma = \min\{D_{k+1}, \min_{t,t'} A_{h(i)}^{k}(t,t'), \forall t,t' : (t,t') \in \mathcal{I}_{a_{k+1},d_{k+1}}\}$. To prove, assume Γ is not the optimal value of R_{k+1}. Therefore, since according to the definition of R_{k+1} it always holds that $R_{k+1} \le \Gamma$ thus, $R_{k+1}^\star < \Gamma$ and the amount of resource to be reserved for EV i is decreased by $\Gamma - R_{k+1}^\star$. This amount, can be only assigned to EVs $k+2$ to n since R_i is set to its maximum value for $i = 1,\ldots,k$. However, having $v_{k+1}/D_{k+1} \ge v_i/D_i, i = k+2,\ldots,n$ the optimal total revenue can be increased by setting $R_{k+1} = \Gamma$ which is a contradiction. □

4.3.2.2 Phase II- Allocation

Lemma 4.3.2.1 shows that there is a feasible scheduling to allocate the reserved resources. However, despite its feasibility, it is not straightforward to find such scheduling. For example, assume that for EV i it is set that $R_i = 10$ and $k_i = 4$. It is possible that all available resources are concentrated in a single time slot but EV i cannot use more than 4 kWh of it. In this situation, the previously allocated resources in interval I_{a_i,d_i} should be re-allocated such that the concentrated resources are *dispersed* and we have $\sum_{t=a_i}^{d_i} \sigma_t \ge R_i$ where $\sigma_t = \min\{k_i, A_{h(i)}^{i-1}(t,t)\}$ is the maximum resource that can be allocated to EV i at time slot t. Since the total amount of allocated resource does not change in the interval, such dispersion is possible and can be done by a simple algorithm in which allocates $\min\{k_i, A_{h(i)}^{i-1}(t,t)\}$ starting from time slot $t = a_i$ until R_i units is allocated. To further reduce the peak of the system, we will develop SmartAllocate algorithm (See Section 4.4) which acts more intelligent so that Line 16 of the fCS can be replaced by "Run SmartAllocate(i, R_i)".

Theorem 4.3.2.1. fCS *is an optimal solution under fractional revenue model.*

Proof. By utilizing Lemma 4.3.2.2, fCS sets R_i to its optimal value for EV $i, i = 1,\ldots,n$. Based on Theorem 4.3.2.1, since it is feasible to allocate R_i to EV $i, i = 1,\ldots,n$, the total gain by fCS is optimal. □

The following theorem characterizes the complexity of fCS.

Theorem 4.3.2.2. *The time complexity of* fCS *algorithm is $O(n^2T + nT^2)$ where n is the number of EVs and T is number of time slots.*

Proof. The algorithm starts by sorting the charging profiles which costs $O(n \log n)$. Then, in the first "for" loop in Lines $4-8$, the algorithm calculates R_i for $i = 1,\ldots,n$. This requires us to check that for each EV i, there are enough available resources in all time intervals in the set $\mathcal{I}_{a_i,d_i}$. By definition, number of these times slots is $(T - d_i + 1)a_i$ which is $O(T^2)$ and their length varies from 1 to T. Therefore, the complexity of the first "for" is $O(nT^2)$. Finally, in the second "for" loop where the algorithm makes re-allocations, it should check all previously allocated EVs in their availability interval which can be done in $O(n^2T)$ and dominates the cost of sub procedure SmartAllocate. Therefore, the total cost is $O(n^2T + nT^2)$. □

4.3.3 Online Scenario

In this section, we devise an algorithm for the scenario that EVs arrive in online fashion. The scheduling decisions at each time slot are made given the information of available EVs and neither exact values nor stochastic modeling of future arrivals is available. Our goal is to obtain a *competitive ratio* for the online algorithm. A scheduling algorithm $\mathcal{A}$ has a competitive ratio of c with $c \geq 1$, if for any input sequence, the utility $U(\mathcal{A})$ of $\mathcal{A}$ satisfies that

$$\frac{U^\star}{U(\mathcal{A})} \leq c,$$

where $U^\star$ is the maximum revenue over the input instance [80].

Our online algorithm for the fractional revenue model, referred to as foCS, is listed as Algorithm 9. The foCS is a simple yet efficient algorithm that always selects EVs with highest unit value to allocate. First, foCS sorts the available EVs at each slot t based on their unit values (Line 2). Then, it selects the EV with highest unit value and sets the charging rate of the EV to maximum possible rate taking into account its remaining demand, maximum charging rate, and peak constraints (Lines 3-4). The allocation is continued until all resources are allocated or there is no more EV which can receive more resources at the current time slot according to its maximum charging rate and remaining demand. For unselected EVs the value of y_i^t will be 0 in Line 4 of the algorithm. The time complexity of the foCS is $O(n^2T)$, determined by cost of its "**for**" loop multiplied by the number of running times of the algorithm i.e., T.

Despite the simplicity of foCS which makes it easy to implement, its performance is sound and within a constant factor of the offline optimum. We now proceed to analyze the performance of the foCS by first giving some preliminaries.

Fix an optimal scheduling and let $\mathcal{S}_{\text{foCS},t}$ and $\mathcal{S}_{\text{Opt},t}$ be the sets of EVs selected by the foCS and optimal solution at time slot t, respectively. Let y_i^t and z_i^t be the charging rate of EV i set by foCS and Opt, respectively. We define Δ_i^t as follows:

$$\Delta_i^t = \begin{cases} \min\{z_i^t - y_i^t, r_{i,t}\} & i \in \mathcal{S}_{\text{Opt},t}, z_i^t > y_i^t, \\ 0 & \text{otherwise,} \end{cases} \tag{4.4}$$

```
Algorithm 9: FOCS: $\forall t \in \{1, 2, \ldots, T\}$
  Input: Available EVs at time slot $t$, number of CSs $m$, local peak constraint
         $p_j, j = 1, \ldots, m$, global peak constraint $p^{\text{total}}$
  Output: A feasible charging scheduling
  $\mathcal{M}^t \leftarrow$ The set of EVs available but not finished at time slot $t$
  Sort EVs in set $\mathcal{M}^t$ indexed by $i = 1, \ldots, |\mathcal{M}^t|$:
    $v_1/D_1 \geq v_2/D_2 \geq \cdots \geq v_{|\mathcal{M}^t|}/D_{|\mathcal{M}^t|}$
  for $i = 1, \ldots, |\mathcal{M}^t|$ do
      $y_i^t \leftarrow \min\{k_i, \sum_{\tau=a_i}^{t} y_i^\tau - D_i, p_{h(i)} - \sum_{i':h(i')=h(i)} y_{i'}^t, p^{\text{total}} - \sum_{i'} y_{i'}^t\}$
```

where $r_{i,t}$ is the remaining demand of EV i by the end of time slot t. $\Delta_i^t > 0$ indicates that the optimal algorithm allocated Δ_i^t units more resources to EV i than the FOCS by time slot t that could be *feasibly* allocated by FOCS to the EV i. If for any EV $i \in \mathcal{S}_{\text{OPT},t}$ and time slot $t \in \mathcal{T}$ we have $y_i^t = z_i^t$, i.e., $\Delta_i^t = 0$, then the FOCS is obviously optimal because it gains whatever the optimal solution gains. For the case that $\exists i, t : \Delta_i^t > 0$, we define *loss* of the FOCS imposed by EV i as follows:

$$l_{i,t} = \Delta_i^t \frac{v_i}{D_i}, \tag{4.5}$$

When FOCS sets charging rate of an EV less than its rate in the optimal solution, it gains $\Delta_i^t v_i/D_i$ less than the optimal solution from that EV. An upper bound for the distance between the optimal objective value (denote by OPT) and the revenue of the FOCS (denote by ALG) is summation of the losses over all time slots and all EVs, i.e.,

$$\text{OPT} - \text{ALG} \leq \sum_{t=1}^{T} \sum_{i \in \mathcal{S}_{\text{OPT},t}} l_{i,t}. \tag{4.6}$$

Notice that it is possible that an algorithm does not choose any selected EV by a particular optimal solution but still provide an optimal or near-optimal solution as there can be multiple optimal solutions. In particular, $\mathcal{S}_{\text{FOCS},t} \cap \mathcal{S}_{\text{OPT},t} = \emptyset, \forall t$ cannot lead to any conclusion on the competitive ratio of the FOCS. In addition to the amount of loss, the gain of the algorithm from charging alternative EVs should also be taken into account in comparison between OPT and ALG.

Let $i \in \mathcal{S}_{\text{OPT},t}$, $i \notin \mathcal{S}_{\text{ALG},t}$ and $g_{i,t}$ be the gain that the FOCS obtains from charging another EV instead of i at time slot t. We are going to show that in the FOCS, for each EV i with $l_{i,t} > 0$ there must be another EV (denote by i') where $y_{i'}^t \geq \Delta_i^t$ and $\frac{v_{i'}}{D_{i'}} \geq \frac{v_i}{D_i}$ which means in resource allocation phase for EV i, the FOCS allocated the difference Δ_i^t to another EV with the same or higher unit value. This can be proved by considering the fact that (i) the selected EVs have higher unit values than the unselected EVs, and (ii) the charging rate of the selected EVs are set to the maximum feasible value. Moreover, the FOCS does not let any resource to remain unused if there are some EVs that can use it.

Let $g_{i,t}$ denote the gain that FOCS obtains from allocating the same amount of resource that optimal algorithm allocated to EV i (with size z_i^t) to another EV(s). If $\Delta_i^t = 0$, the loss is zero. If $\Delta_i^t > 0$, then by the charging strategy that the FOCS uses we can conclude that (i) i is not finished by the FOCS, and (ii) $\sum_{i'} y_{i'}^t = \min\{p_{h(i)}, p^{\text{total}}\}$ otherwise, the FOCS could allocate more resources to EV i. Therefore, the Δ_i^t units of the resource is allocated to one or multiple other EVs (denote them by set $\mathcal{J}_i^t$) by the FOCS. Moreover, it must hold that all the EVs in set $\mathcal{J}_i^t$ has a unit value equal to or higher than v_i/D_i which yields $g_{i,t} \geq l_{i,t}$, otherwise, the FOCS should not prefer the EVs in $\mathcal{J}_i^t$ to i. Having $g_{i,t} \geq l_{i,t}$, we obtain the following:

$$\sum_{i=1}^{n}\sum_{t=1}^{T} l_{i,t} \leq \sum_{i=1}^{n}\sum_{t=1}^{T} g_{i,t}. \tag{4.7}$$

The total gain of FOCS, i.e., ALG, is equal to sum of its gains from each single EV:

$$\text{ALG} = \sum_{i=1}^{n} \frac{v_i}{D_i} \sum_{t=1}^{T} y_i^t = \sum_{i=1}^{n}\sum_{t=1}^{T} g_{i,t}. \tag{4.8}$$

With the above discussion and using Eq. (4.7) we are able to show that the competitive ratio of the FOCS is 2.

Theorem 4.3.3.1. *The* FOCS *is* 2*-competitive.*

Proof. We first prove the theorem for single station scenario and then extend it to multiple stations. From (4.6), (4.7), and (4.8) we obtain the following result

$$\text{OPT} - \text{ALG} \leq \sum_{i=1}^{n}\sum_{t=1}^{T} g_{i,t} \leq \text{ALG}, \tag{4.9}$$

hence, $\text{OPT} \leq 2\text{ALG}$.

In multi-station setting, the difference with the previous case is that when $\Delta_i^t > 0, i \in \{1, \dots, n\}$, then FOCS may allocate the difference Δ_i^t to one or multiple EVs in any CS that might not be $h(i)$. However, the inequality $l_{i,t} \leq g_{i,t}$ is still valid as FOCS is centralized and uses a single sorted list for all EVs. Using similar deductions as in the single station setting, it is easy to verify that the competitive ratio of 2 is preserved.

□

Remarks: When there is only one CS and EVs have no limit on their charging rate, the foCS reduces to the FirstFit algorithm [95] which is known to be 2-competitive for classic job scheduling problem. However, the charging rate limitation is crucial for EV charging problem. Moreover, [95] uses "charging argument" to prove the competitive ratio of the proposed algorithm which cannot be directly applied to our problem. Thus, the foCS extends the FirstFit and makes it practical for the EV charging scenario. Moreover, the proof technique used for the competitive analysis of the foCS is fundamentally different from the one used in [95].

Theorem 4.3.3.2. *The time complexity of* foCS *is* $O(nT \log n)$.

Proof. The complexity of the algorithm is identified by sorting operation at each time slot which dominates the $O(n)$ cost of the "for" loop. Therefore, the total time complexity is $O(nT \log n)$. □

4.4 Integral Revenue Model

4.4.1 Overview of Integral Revenue Model

The MILP form of the SPAN in integral revenue model is a generalized form of the 0/1-knapsack problem which is a well-known NP-hard problem. We give an intuition to understand the similarity of these problems in Section 4.4.2.1, however, we skip to prove that there is a polynomial time algorithm to reduce 0/1-knapsack problem to the MILP form of the SPAN due to the straightforwardness of the proof. In Section 4.4.2, we propose a polynomial time approximation algorithm for the general integral problem and analyze its approximation ratio. Then, in Section 4.4.3, we propose an online algorithm for the integral model, where the EVs arrive in slot-by-slot fashion, and the scheduler has no information of the future arrivals.

4.4.2 Offline Scenario

We design our offline smart charging scheduling algorithm under integral revenue model referred to as iCS, to solve the SPAN with bounded approximation gap. Our algorithm design is inspired by the basic algorithm proposed in [94]. The algorithm in [94] works for a single CS where arrival time of all EVs are the same and there is no global peak constraint. Since the performance analysis of the proposed algorithm relies on a dual fitting method and utilizes weak duality property, we construct the dual problem of SPAN, referred to as dSPAN, as follows:

$$
\begin{aligned}
&\mathsf{dSPAN}: \\
&\min \quad \sum_{i=1}^{n} D_i\alpha_i + \sum_{j=1}^{m}\sum_{t=1}^{T} p_j\beta(t) + \sum_{t=1}^{T} p^{\mathsf{total}}\gamma(t) \\
&\text{s.t.} \quad \alpha_i + \beta(t) + \gamma_i + \pi(t) - \frac{k_i}{D_i}\sum_{t'=a_i}^{d_i} \pi_i(t') \geq \frac{v_i}{D_i}, \\
&\qquad\qquad \forall i, t \in [a_i, d_i] \qquad (4.10a) \\
&\text{vars.} \quad \alpha_i, \beta_i, \gamma, \pi_i(t) \geq 0, \qquad \forall i, t,
\end{aligned}
$$

In the dSPAN, the dual variables α, γ, β and $\pi_i(t)$ are associated with constraints (4.1a) (4.1b), (4.1c) and (4.1d) in the SPAN, respectively.

4.4.2.1 Explanation of the Main Algorithm

The iCS algorithm (listed as Algorithm 10) works in two phases. In the first phase it sorts the charging requests based on their unit values in a non-increasing order. Then, it selects most valuable demand from top of the list and if the remaining resource is enough for covering the *entire* demand of the the EV, it is admitted to receive the demand. When iCS processes EV i, the algorithm checks for the feasibility of allocating D_i units of the resource within its availability window $[a_i, d_i]$ without violating the constraints related to the maximum charging rate k_i, local and global peaks (Lines 6-8).

Scheduling of the Selected EV: If the feasibility check passed, iCS calls sub-procedure SmartAllocate to allocate required resources in interval $[a_i, d_i]$. Then, α_i is set to v_i/D_i in order to cover dual constraint in Eq. (4.10a) (Lines 9-10).

We explain SmartAllocate in more details. Let us define $W(t, h(i)) = \sum_{i':h(i')=h(i)} y_{i'}^t$ as total workload at time slot t in CS $h(i)$ and $\bar{W}(t, h(i))$ as total available load to allocate at time slot t for CS $h(i)$. We always have $\bar{W}(t, h(i)) + W(t, h(i)) = p_{h(i)}, \forall t, i$. For scheduling, SmartAllocate applies two main policies: 1) *flat allocation* and, 2) *right-to-left allocation.* With flat allocation, time slots with more available resources are preferred for charging purpose. This is in fact a *valley-filling* strategy which helps to reduce the peak of the system. The simulation results in Section 4.5 will confirm this claim. Right-to-left allocation is used when two or more time slots are equal in terms of their remaining resources. When this policy applies on scheduling of EV i, any EV i' with $i' > i$ and $d_{i'} < d_i$ has more chance to get charged since the algorithm tends to charge EV i in interval $[d_{i'} + 1, d_i]$ (i.e., right hand part of the EV's availability window) and keeps resources in $[a_{i'}, d_{i'}]$ for EV i'. A ranking based approach is used to apply the aforementioned policies. To charge EV i, SmartAllocate ranks time slots in interval $[a_i, d_i]$. Then, charging is done by allocating resources from the higher ranked time slot to lowest one. The rank of a time slot t is calculated based on remaining resources in the time slot (flat allocation) and value of t (right-to-left allocation).

Algorithm 10: iCS

```
Input: $n$ EVs with $a_i, d_i, v_i, D_i$, and $k_i$ associated with each EV $i$, $m$ CSs, local
       and global peak constraints $p_j, j = 1, \dots, m$, and $p^{\mathsf{total}}$
Output: A feasible scheduling of EVs
initialize: $y \leftarrow 0, \alpha \leftarrow 0, \beta \leftarrow 0, \gamma \leftarrow 0, \pi \leftarrow 0$
Sort charging requests in non-decreasing order of their unit values:
$\frac{v_1}{D_1} \geq \frac{v_2}{D_2} \geq \dots \geq \frac{v_n}{D_n}$
//Use sorted list to process demands
for (i=1...n) do
    for $t = a_i \dots d_i$ do
        $\sigma_t \leftarrow \min \Big\{ p_{h(i)} - \sum_{i':h(i')=h(i)} y_{i'}(t),$
          $p^{\mathsf{total}} - \sum_{i'=1}^{n} y_{i'}^{t}, k_i \Big\}$
    //if enough resources remain for EV i
    if $D_i \leq \sum_{t=a_i}^{d_i} \sigma_t$ then
        SmartAllocate$(i, D_i)$
        $\alpha_i \leftarrow \frac{v_i}{D_i}$
    else
        if $(\beta(d_i) = 0)$ then
            BetaCover$(i)$
for (i=1...n) do
    if EV i is not selected then
        ReConsider$(i)$;
```

Dual Feasibility on the EVs that are not Selected: If there is not enough resources to fully charge EV i, i.e., $D_i > \sum_{t=a_i}^{d_i} \sigma_t$, the EV cannot be selected. However, we still need to satisfy constraint (4.10a) in dual problem which is done by calling BetaCover(i). To cover the constraint (4.10a) for EV i, sum of dual variables for all $t \in [a_i, d_i]$ should be greater than or equal to v_i/D_i. Towards this, BetaCover(i) sets $\beta(t)$ to v_i/D_i for all time slots t in interval $[t_{\mathsf{cov}}, R(d_i)]$ (Lines 3-4 of Algorithm 3, the definitions of t_{cov} and $R(d_i)$ are given in Lines 1 and 2, respectively). Observe that when $t_{\mathsf{cov}} > 1$ we have $\beta(t') \geq v_i/D_i, \forall t' < t_{\mathsf{cov}}$ considering that the demands are sorted in a non-increasing order according to their unit values and $\beta(t')$ is already set to $v_{i'}/D_{i'}$ when processing the earlier charging demand of EV i' in the list which is not selected. Hence, $v_{i'}/D_{i'} \geq v_i/D_i$, thereby

$\beta(t) \geq v_i/D_i,\ \forall t \in [a_i, d_i]$ and the dual constraint in (4.10a) is satisfied.

Lines 1-4 of algorithm BetaCover(i) is enough to cover dual constraint. However, the algorithm continues in Lines 5-8 by setting a variable $\Phi_{i'}(t)$ for time slots $t = 1, \ldots, R(d_i)$ to a value dependent to amount of the resource that a selected EV i' received at slot t. $\Phi_{i'}(t)$ will be used in approximation analysis of the main algorithm in Section 4.4.2.2 and has no effect on the scheduling of EVs.

Algorithm 11: SmartAllocate(i, D_i)

```
Input: EV $i$ to receive $D_i$
Rank time slots in interval $[a_i, d_i]$ such that for $t_1$ and $t_2$: $\mathsf{rank}(t_1) > \mathsf{rank}(t_2)$ iff
  $\bar{W}(t_1, h(i)) > \bar{W}(t_2, h(i))$ OR $\bar{W}(t_1, h(i)) == \bar{W}(t_2, h(i)) \wedge t_1 > t_2$
while $\sum_{t'=a_i}^{d_i} y_i^{t'} \neq D_i$ do
    Select time slot $t$ with highest rank which is not selected before
    Allocate $\min\{k_i, \bar{W}(t, h(i))\}$ for EV $i$ at slot $t$
```

Algorithm 12: BetaCover(i)

```
  Input: EV $i$ which is not selected to charge
1 $t_{\mathsf{cov}} \leftarrow \min\{t : \beta(t) = 0\}$
2 $R(d_i) = \max\{t \geq d_i : \forall t' \in (d_i, t], \bar{W}(t') < q_{h(i)}\}$
3 for $(t = t_{\mathsf{cov}} \ldots R(d_i))$ do
4     $\beta(t) \leftarrow \frac{v_i}{D_i}$
5 for $(t = 1 \ldots R(d_i))$ do
6     for $(i' = 1 \ldots n)$ do
7         if $y_{i'}^t > 0 \wedge \Phi_{i'}(t) = 0$ then
8             $\Phi_{i'}(t) \leftarrow \left[\frac{p_{h(i)}}{p_{h(i)}-k_i}\frac{s}{s-1}\right].\frac{v_i}{D_i}y_{i'}^t$
```

Improving the Peak-demand: In the second phase of iCS, the algorithm tries to increase total value of selected EVs by calling ReConsider(i) on every unselected EV i (Lines 14-16). Before giving the details of ReConsider(i), we first explain the intuition behind this algorithm. Note that if in the scheduling problem we set $T = 1, m = 1, k_i = p^{\mathsf{total}}, a_i = d_i = 1\ \forall i$, then the problem is equal to the well-known 0-1 knapsack problem [92]. In the knapsack problem, a widely used greedy approach sorts items based on their unit values and selects items accordingly. It turns out that in this approach the approximation factor can be arbitrarily bad. For example, consider a knapsack problem with two items with $v_1 = 2, v_2 = p^{\mathsf{total}}, D_1 = 1$, and $D_2 = p^{\mathsf{total}}$. Given these values we have $v_1/D_1 > v_2/D_2$. To maximize total value of selected items, the optimal solution chooses item 2 while greedy algorithm selects item 1 which results in a worst-case approximation factor of $c/\textsc{Opt}$ in

Algorithm 13: ReConsider(i)

```
Input: EV $i$
Output: Updated schedule
$\mathcal{L} \leftarrow \emptyset$
$v_{\mathsf{inc}} \leftarrow v_i$
$\sigma_t \leftarrow 0, t = a_i, \ldots, d_i$
for ($i' = i - 1 \ldots 1$) do
    if EV $i'$ is selected $\wedge (h(i') = h(i)) \wedge (v_{\mathsf{inc}} - v_{i'}) > 0$ then
        Add EV $i'$ to list $\mathcal{L}$
        $v_{\mathsf{inc}} \leftarrow v_{\mathsf{inc}} - v_{i'}$
        for ($t = a_i \ldots d_i$) do
            $\sigma_t \leftarrow \sigma_t + \min\{k_i, y^t_{i'}\}$
if $\sum_{t=a_i}^{d_i} \sigma_t \geq D_i$ then
    Remove EVs in list $\mathcal{L}$ from charging schedule
    SmartAllocate($i, D_i$)
```

general where c is a constant (in this example $c = 2$). To resolve it, one approach is to *re-consider* unselected items after running greedy algorithm and replace some selected items in the knapsack with unselected ones and then check whether the result is improved or not. In a simple case, only the largest unselected item can be examined which makes a significant theoretical improvement by providing a worst case approximation factor of Opt/2.

iCS algorithm leverages the same idea but using a more intelligent replacing method called ReConsider(i). ReConsider(i) is called on every unselected EV i. It tries to find some selected EVs that if they are replaced by EV i, total revenue from the selected EVs increases.

4.4.2.2 Analysis

In primal-dual algorithm, the goal is to design an algorithm in a way that it produces a good solution for primal problem (with primal value Γ) and a feasible solution for the dual problem (with dual value Λ). Then, assuming that the primal problem is a maximization problem, to prove that the algorithm is $c-$approximation (for $c \geq 1$), the important part is to show that $\Lambda \leq c\Gamma$. Then, based on weak duality theorem we have $\Lambda \geq$ Opt, and it is concluded that $\Gamma \geq \frac{1}{c} \times$ Opt where Opt is the optimal value.

Based on the above understanding, we now analyze the approximation ratio of iCS algorithm assuming that arrival times are the same for all EVs. First note that the designed scheduling algorithm outputs a feasible scheduling since it respects the constraints in the primal problem. Also, the algorithm produces a feasible solution for the dual problem by covering the dual problem constraint in (4.10a) through setting α_i to $\frac{v_i}{D_i}$ when EV i is accepted and, $\beta(t)$ to a value greater than or equal to $\frac{v_i}{D_i}$ for $t \in [a_i, d_i]$ (according to the discussion in Section 4.4.2.1) if EV i is not selected. To obtain an approximation factor for the algorithm, it is enough to bound the total covering cost of the dual constraints.

Theorem 4.4.2.1. iCS *algorithm is a* $\left(1 + \sum_{j=1}^{m} \frac{p_j}{p_j - q_j} . \frac{s}{s-1}\right)$*-approximation when EVs have same arrival time.*

Proof. Without loss of generality we assume that $a_i = 1, \forall i$, however, the proof holds for any other constant value for arrival time. We sum up all costs of covering dual constraints and then provide a bound for it.

Each EV is either selected or not selected. For the unselected EVs, $\sum_{j=1}^{m} \sum_{t=1}^{T} p_j \beta(t)$ determines the cost. When BetaCover(i) is running as a result of charging request disapproval of EV i, for any previously accepted request i' the algorithm sets $\Phi_{i'}(t)$ to a value proportional to $y_{i'}^t$ for $t \leq R(d_i)$ (Line 8 of BetaCover(i) algorithm). The followings are proved in [94] for a single station $h(i') = h(i) = j$:

$$\sum_{i'=1}^{n} \sum_{t=1}^{d_j} \Phi_{i'}(t) \leq \left[\frac{p_j}{p_j - q_j} . \frac{s}{s-1}\right] . \sum_{i'=1}^{n} v_{i'} \tag{4.11}$$

$$\sum_{t=1}^{T} p_j \beta(t) \leq \sum_{i'=1}^{n} \sum_{t=1}^{d_j} \Phi_{i'}(t). \tag{4.12}$$

For m CSs, we can obtain the following inequality based on (4.11) and (4.12),

$$\sum_{j=1}^{m} \sum_{t=1}^{T} p_j \beta(t) \leq \sum_{j=1}^{m} \Big(\frac{p_j}{p_j - q_j} . \frac{s}{s-1} \sum_{i:h(i)=j} v_i\Big). \tag{4.13}$$

Now for notational convenience let's define A_j, B_j and C as follows:

$$\begin{aligned} A_j &= \frac{p_j}{p_j - q_j} . \frac{s}{s-1}, \\ B_j &= \sum_{i:h(i)=j} v_i, \\ C &= \sum_{j=1}^{m} B_j = \sum_{i=1}^{n} v_i. \end{aligned}$$

We can write the right hand side of Eq. (4.13) as follows:

$$\begin{aligned}
\sum_{j=1}^{m} A_j B_j &= \sum_{j=1}^{m} \Big[A_j\big(C - \sum_{i:h(i)\neq j} B_{h(i)}\big)\Big] \\
&= \sum_{j=1}^{m} A_j C - \sum_{j=1}^{m} \Big[A_j \sum_{i:h(i)\neq j} B_{h(i)}\Big] \\
&= \sum_{j=1}^{m} A_j C - \sum_{j=1}^{m} \Big[A_j\big(C - B_j\big)\Big] \\
&\leq \sum_{j=1}^{m} A_j C - \sum_{j=1}^{m} \Big[A_j\big(C - \max_j\{B_j\}\big)\Big] \\
&= \max_j\{B_j\} \sum_{j=1}^{m} A_j. \qquad (4.14)
\end{aligned}$$

From (4.13) and (4.14) we get

$$\sum_{j=1}^{m}\sum_{t=1}^{T} p_j \beta(t) \leq \max_j\{B_j\} \sum_{j=1}^{m} \frac{p_j}{p_j - q_j} . \frac{s}{s-1} \qquad (4.15)$$

For the selected EVs, the covering cost is determined by the term $\sum_i D_i \alpha_i$ in the dual objective which equals to $\sum_{i\in\mathcal{S}} v_i$ where $\mathcal{S}$ is the set of selected EVs. Therefore, the total cost of covering dual constraints equals to

$$\begin{aligned}
\Lambda &= \sum_{i\in\mathcal{S}} v_i + \max_j\{B_j\} \sum_{j=1}^{m} \frac{p_j}{p_j - q_j} . \frac{s}{s-1} \\
&\leq \sum_{i\in\mathcal{S}} v_i + \sum_{i\in\mathcal{S}} v_i \sum_{j=1}^{m} \frac{p_j}{p_j - q_j} . \frac{s}{s-1} \\
&= \Big[1 + \sum_{j=1}^{m} \frac{p_j}{p_j - q_j} . \frac{s}{s-1}\Big] \sum_{i\in\mathcal{S}} v_i \qquad (4.16)
\end{aligned}$$

Given that the primal value obtained from iCS is $\Gamma = \sum_{i\in\mathcal{S}} v_i$, we get

$$\Lambda \leq \Big[1 + \sum_{j=1}^{m} \frac{p_j}{p_j - q_j} . \frac{s}{s-1}\Big]\Gamma. \qquad (4.17)$$

Finally, considering the fact that $\Lambda \geq$ Opt, we conclude that iCS is $\Big[1 + \sum_{j=1}^{m} \frac{p_j}{p_j - q_j} . \frac{s}{s-1}\Big]$-approximation.

□

Note that in the case that the system is flexible enough, i.e., $s \gg 1$, and the maximum charging rates of stations are much bigger than those of EVs, i.e., $p_j \gg q_j, \forall j$, the approximation ratio approaches $m+1$. And in the case that there is one single station, the approximation ration is 2. Finally, we provide the time complexity of the iCS.

Theorem 4.4.2.2. *The time complexity of the* iCS *algorithm is* $O(nT \log T + n^2T)$.

Proof. iCS starts by sorting demands which costs $O(n \log n)$. The inner "for" loop in Line 5 costs $O(nT)$ and running SmartAllocate(i, D_i) and BetaCover(i) costs $O(T \log T)$ and $O(nT)$, respectively. Therefore, the total cost of outer "for" loop (Line 4) equals to $O(nT \log T + n^2T)$. The complexity of ReConsider(i) called in the "for" loop in Line 14 is $O(nT)$. Therefore, the total cost is $O(nT \log T + n^2T)$. □

4.4.3 Online Scenario

Due to existence of binary selection variable, the online solution design under integral revenue model is fundamentally more challenging than the one for fractional model. In this section, we propose the ioCS that is built upon the offline iCS algorithm. In particular, the ioCS calls the iCS at each time slot for the set of available EVs, however, any other algorithm that is designed for offline integral model can be used alternatively. Hereinafter, $\mathcal{A}$ refers to iCS or a similar algorithm and ρ denotes the approximation factor of $\mathcal{A}$.

The ioCS is summarized as Algorithm 14 and explained in the following. At each time slot t, the ioCS compares two scheduling results returned by $\mathcal{A}$ and chooses among them. In the first scheduling, the ioCS keeps all reserved resources in interval $[t, T]$ intact. Then, for utilizing the remaining resources, the algorithm runs $\mathcal{A}$ over arrived EVs at time slot t to allocate them. In this case, the total gain that can be obtained by the *active* EVs (i.e., EVs that are available but not received their entire demand yet) is denoted by $\widehat{\Gamma}_{\mathcal{A},\mathcal{R}^t}$ (Line 7 of the algorithm). In the second scheduling, the ioCS considers the case that it can sacrifice the previously admitted EVs by canceling their reservations and allocating the freed resources to more valuable demands. For this purpose, the algorithm modifies the demand and valuation of the previously admitted EVs such that each demand is replaced by the EV's remaining demand, and the valuation of the EV is proportionally calculated based on the remaining demand (Line 10 of the algorithm) so that the unit values of EVs does not change. Then, the ioCS runs $\mathcal{A}$ over $\mathcal{M}^t$ (defined in Line 3) where the corresponding gain is denoted by $\Gamma_{\mathcal{A},\mathcal{M}^t}$. If $\Gamma_{\mathcal{A},\mathcal{M}^t} > \widehat{\Gamma}_{\mathcal{A},\mathcal{R}^t}$, the ioCS forgets the previously admitted EVs and follows the second scheduling.

Let $\mathcal{S}_{\text{ioCS},t}$ and $\mathcal{S}_{\text{Opt},t}$ be set of active EVs at time slot t that are selected for charging by ioCS and Opt (fix a particular optimal solution), respectively. Note that the set of active EVs for ioCS and Opt might be different. Having $i \in \mathcal{S}_{\text{ioCS},t}$ means that based on

```
Algorithm 14: ioCS: $\forall t \in \{1, 2, \ldots, T\}$
  Input: $n$ EVs to arrive on the fly, number of CSs $m$, global peak constraint $p^{\text{total}}$,
         local peak constraints $p_j, j = 1, \ldots, m$
  Output: A feasible charging scheduling
  Let $\mathcal{A}$ be an algorithm that solves SPAN efficiently with $a_1 = \cdots = a_n$
  $\mathcal{R}^t \leftarrow$ set of EVs arrived at time slot $t$
  $\mathcal{M}^t \leftarrow \mathcal{R}^t \cup \{i : t \in \mathcal{T}_i$ AND $\sum_{t'} y_i^{t'} < D_i\}$
  \\Schedule 1
  Based on the remaining resources in interval $[t, T]$, use algorithm $\mathcal{A}$ to allocate
    EVs in set $\mathcal{R}^t$ assuming no further arrivals
  $\mathcal{S}_{\text{IOCS},t} \leftarrow \{i : i \in \mathcal{M}^t$ AND $i$ is admitted at $t\}$
  $\widehat{\Gamma}_{\mathcal{A},\mathcal{R}^t} \leftarrow \sum_{i \in \mathcal{S}_{\text{IOCS},t}} v_i$
  Assume all reserved resources are freed at time slots $t, t+1, \ldots, T$
  $r_{i,t} \leftarrow$ remaining demand of $i$ at $t, \forall i$
  $D'_i \leftarrow r_{i,t}, \quad v'_i \leftarrow \frac{r_{i,t}}{D_i} v_i, \forall i \in \mathcal{M}^t$
  \\Schedule 2
  Run $\mathcal{A}$ on $\mathcal{M}^t$ using $D'_i$ and $v'_i, \forall i$ and reconstruct $\mathcal{S}_{\text{IOCS},t}$
  $\Gamma_{\mathcal{A},\mathcal{M}^t} \leftarrow \sum_{i \in \mathcal{S}_{\text{IOCS},t}} v_i$ \\Use original values
  if $\Gamma_{\mathcal{A},\mathcal{M}^t} > \widehat{\Gamma}_{\mathcal{A},\mathcal{R}^t}$ then
      Use the second schedule
  else
      Use the first schedule
```

the updated schedule at time slot t, enough resources reserved for EV i so that the EV will receive its demand D_i before the deadline d_i. However, the schedule may change at each time slot and $i \in \mathcal{S}_{\text{IOCS},t}$ cannot be a guarantee for EV i to fully receive its demand. In fact, users and the scheduler itself have to wait until the deadline of the EV to find out that the demand is fulfilled or not. This is because the scheduler may cancel some reservations in the next time slots and allocate the freed resources to other EVs (when Algorithm 14 uses the second schedule in Line 15).

The following theorem characterizes the performance of the IOCS when EVs arrive in batch mode and $m = 1$.

Theorem 4.4.3.1. *Let $\mathcal{A}$ be* ICS *in* IOCS *algorithm and $m = 1$. Assuming that EVs are released in b distinct groups where arrival time of EVs in each group are the same then, the* IOCS *is $b\Big(1 + \frac{p}{p-q}\frac{s}{s-1}\Big)$-competitive with optimal offline solution, where p is the station peak and $q = \max_i k_i, i = 1, \ldots, n$.*

Proof. In the worst case $b = T$ i.e., there are T groups where at each time slot, a group of EVs arrive. Observe that

$$\textsc{Opt} \leq \rho\Gamma_{\mathcal{A},\mathcal{R}^1} + \cdots + \rho\Gamma_{\mathcal{A},\mathcal{R}^T}.$$

Put it simply, the increase in optimal gain at each time slot is at most equal to maximum gain that can be obtained from arrived EVs at time slot t. Moreover, according to the IOCS, the gain of the algorithm obtained from set of active jobs at each time slot t, Γ^t_{IOCS}, is as follows:

$$\Gamma^t_{\text{IOCS}} = \max\{\widehat{\Gamma}_{\mathcal{A},\mathcal{R}^t}, \Gamma_{\mathcal{A},\mathcal{M}^t}\}.$$

Therefore,

$$\Gamma_{\mathcal{A},\mathcal{R}^t} \leq \Gamma^t_{\text{IOCS}}, t = 1, \ldots, T$$

Thus, $\textsc{Opt} \leq T\Big(1 + \frac{p}{p-q}\frac{s}{s-1}\Big)$. If $b < T$, we can obtain $\textsc{Opt} \leq b\Big(1 + \frac{p}{p-q}\frac{s}{s-1}\Big)$ by the same analysis.

□

Theorem 4.4.3.2. *Let $\mathcal{A}$ be* ICS *in Line* 1 *of* IOCS*. Then, the time complexity of the* IOCS *is $O(n^2T)$.*

Proof. The time complexity of IOCS is determined by the cost of algorithm $\mathcal{A}$ that it calls at each time slot. Assuming that $\mathcal{A}$ is ICS, the complexity of IOCS in a single time slot is $O(n^2)$, according to Theorem 4.4.2.2. Therefore, the complexity of IOCS is $O(n^2T)$. □

4.5 Simulation Results

In this section, we perform simulation experiments to evaluate the performance of our proposed scheduling algorithms.

Table 4.3 – Arrival rates and mean parking times.

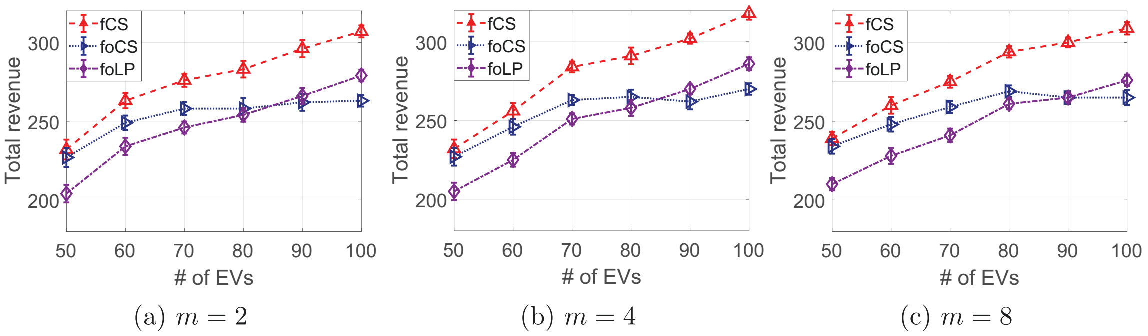
Interval	**Arrival rate**	**Mean parking time**
08:00-10:00	14	10
10:00-12:00	10	1/2
12:00-14:00	20	2
14:00-18:00	10	1/2
18:00-20:00	20	2
20:00-24:00	10	10
24:00-08:00	0	0

Table 4.4 – Acronyms for the algorithms

Notation	**Description**
iOpt	Optimal value under integral revenue model
iCS	Proposed offline algorithm for SPAN under integral revenue model
fCS	Proposed *optimal* algorithm for SPAN under fractional revenue model
ioCS	Proposed online algorithm for SPAN under integral revenue model
foCS	Proposed online algorithm for SPAN under fractional revenue model
ioLP	OLP algorithm [9] for integral revenue model
foLP	OLP algorithm [9] for fractional revenue model
GreedyRTL	The algorithm in [94] for single station scenario without global peak constraint

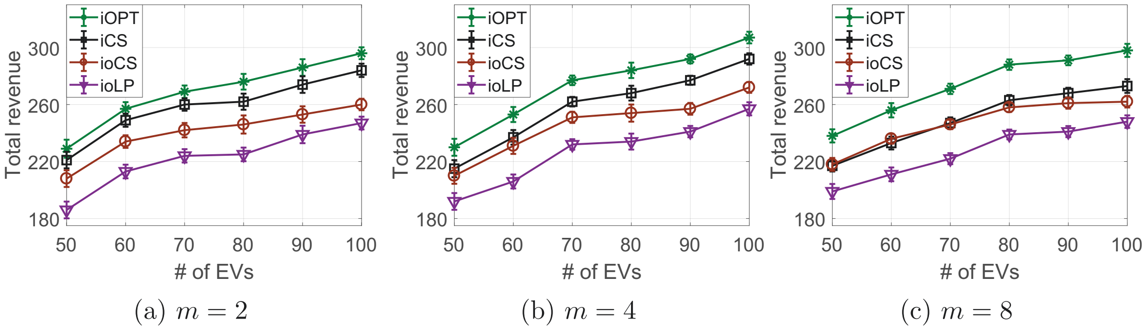
4.5.1 Simulation Setup and Overview

We consider charging scheduling of EVs during a day divided into 24 time slots of length 1 hour. We gathered information of 10 popular EV models in the market to use in the simulation. Each EV model is characterized by its battery capacity and maximum charging rate as shown in Table 3.3. Battery capacity varies from 16 kWh to 100 kWh and the maximum charging rate from 3.3 kW to 20 kW. As in [47, 58], we assume that arrival times follow a Poisson distribution and parking times follow an exponential distribution with the mean ar-



(a) $m = 2$ (b) $m = 4$ (c) $m = 8$

Figure 4.2 – Comparison results for 2, 4, and 8 CSs for fractional revenue model.



(a) $m = 2$ (b) $m = 4$ (c) $m = 8$

Figure 4.3 – Comparison results for 2, 4, and 8 CSs for integral revenue model.

rival and parking duration indicated in Table 4.3. The peak intervals include 08:00-10:00, 12:00-14:00, and 18:00-20:00 which is in accordance to national household travel survey (NHTS) 2009 [47, 96]. Demands are uniform random values from $[\min\{\frac{d_i - a_i + 1}{sk_i}, U_i\}, U_i]$ where U_i is the battery capacity for EV i. The minimum electricity cost is \$0.11 per kWh based on US national electricity price average [90]. In our setting, users can submit higher prices than the minimum price in order to ensure receive their requested demand. For each CS, the default value of local peak constraint is 30 kW and the global peak constraint is 200 kW. The slackness parameter is set by default to 1.2, however, in Section 4.5.4, we investigate the effect of this parameter on total revenue and response time of the algorithms in detail.

In the simulation figures, the results are plotted with 95% confidence level and each data point represents average result of 50 random scenarios. Table 4.4 explains the comparison algorithms. The measured performance metrics are total revenue (i.e., $\sum_{i \in S} v_i$), percentage of EVs that received all their requested demand, total peak of all CSs and response time which is only calculated for EVs who received their whole demand and defined as time length between an EV's arrival and the time that its charging is finished.

4.5.2 Total Revenue

Figs. 4.2-4.3 depict the comparison results based on total revenue under fractional and integral models respectively while the total number of EVs increases from 50 to 100 for 2, 4, and 8 CSs. The proposed algorithms for fractional revenue model, fCS and foCS, are compared to fractional version of OLP algorithm (referred to as foLP), which is implemented in Caltech ACN [9]. OLP [9] is a greedy algorithm that works as follows: (i) at $t = 1$ compute the optimal solution (e.g., by a solver) for the current set of EVs assuming that there will be no further arrival. (ii) use the solution until there is a new arrival. (iii) when an EV arrives, construct and solve a new problem by considering the new arrivals and the remaining demand of previously arrived EVs which are still active. (iv) go back to (ii). The original OLP algorithm which is implemented in Caltech ACN works only for integral revenue model and under different cost functions including the revenue of the charging stations, which is the case of our study. In this thesis, we implemented the algorithm for both integral and fractional model. The difference between the two versions is in the constraint set of the optimization problem that should be solved such that in integral model, the total amount of resource that an EV receives can only be equal to zero or its demand while in fractional model it can be any non-negative value less than or equal to the demand.

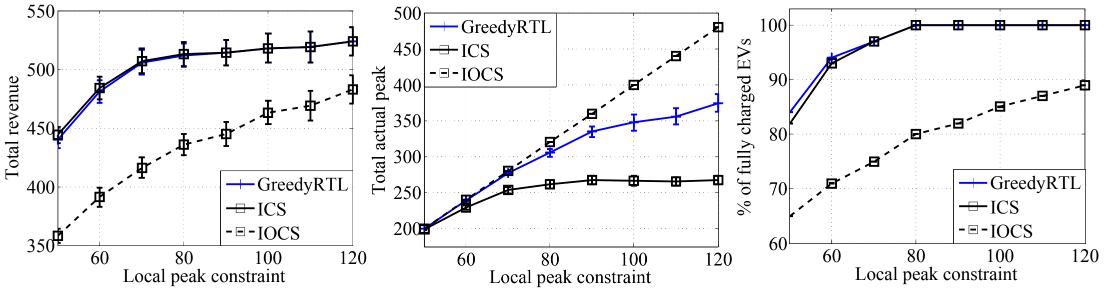
Recall that fCS is optimal offline solution. The notable observations are as follows: (i) the general trend in both scenarios of Fig. 4.2 and 4.3 is that by increasing the number of EVs, total revenue increases. This is because with more number of EVs, the scheduler has more freedom to choose more valuable EVs. (ii) as explained in Section 4.3, under fractional charging model, better results are expected due to increased scheduling flexibility in CSs. According to the simulation data that we extracted from Fig. 4.2 and Fig. 4.3, the gain obtained by fCS, foCS, and foLP in Fig. 4.2 are respectively 9%, 4% and 10% more than the gain of the iCS, ioCS and ioLP in Fig. 4.3. (iii) in fractional revenue model (Fig. 4.2), foLP performs better than foCS for $n \geq 90$, implying that foLP (foCS) is better option when the density of EVs in the CSs is high (low). (iv) in integral revenue model (Fig. 4.3), the proposed ioCS algorithm acts significantly better than ioLP. In particular, ioCS improves ioLP by $8\%, 8\%$ and 9% for $m = 2$, $m = 4$ and $m = 8$, respectively. We highlight that ioCS (foCS) is also better choice in terms of the algorithm complexity compared to ioLP (foLP). (v) iCS approximates iOpt by 96%, 94%, and 91% for $m = 2$, $m = 4$, and $m = 8$, respectively. On average, ioCS is 89% of iOpt and, foCS is 92% of its optimal offline solution, fCS. (vi) finally, the results depict that iCS achieves much better results in practice as compared to the theoretical approximation ratio that characterizes the performance in worst-case scenario.

4.5.3 Actual Peak

The constraint set in the SPAN assures that any feasible solution respects the local and global peak constraints, i.e., at each time slot, the electricity consumed by station j is less than or equal to p_j and accumulative charging rate of all stations are less than or equal to p^{total} (note that there is no assumption that $\sum_j p_j \leq p^{\mathsf{total}}$, in general). An efficient scheduling algorithm may take a further step by not only satisfying the peak constraints but to further reduce the peak as much as possible. As explained in Sections 4.3 and 4.4, our proposed offline algorithms (i.e., iCS and fCS) apply flat scheduling policy to reduce the peak. The online algorithms (i.e., ioCS and foCS) do not apply the same policy as at each time slot they do not have knowledge of future EV arrivals to be able to balance allocated resources. Besides, due to uncertainties in future EV charging load in online scheduling design, it is a natural heuristic to charge EVs at earliest time (i.e., setting the charging rates to the maximum feasible value at each time slot) as applied by ioCS an foCS. This heuristic works well as the uncertainty in EVs' arrival time and demands may not give a second opportunity to the scheduler to charge the current EVs at the next time slots. Therefore, although it is expected that ioCS and foCS have higher peak values compared to the proposed offline algorithms, using maximum charging rate and left-to-right allocation can decrease their average response time (see Section 4.5.4).

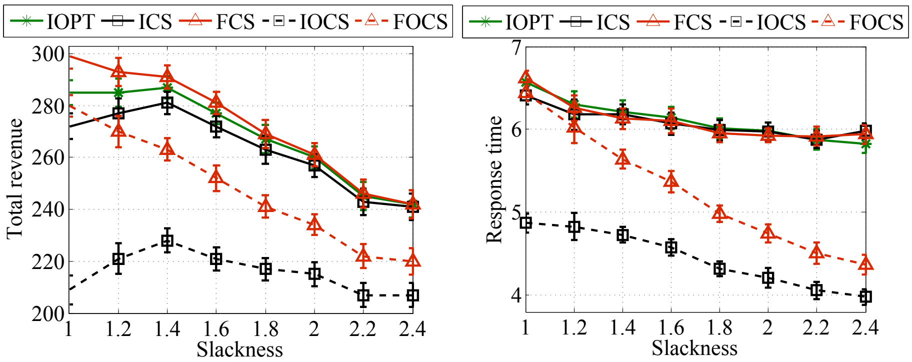
To show how flat scheduling policy improves the peak, we conducted a set of simulations by varying local peak constraints from 50 kWh to 120 kWh and 200 EVs which are dispersed in 4 different CSs. For this scenario, the results of iCS is compared to ioCS and GreedyRTL [94] where the latter is an approximation scheduling for single station scenario with EVs having same arrival time and no global peak constraint (see remark (i) after formulating the SPAN). The GreedyRTL algorithm works under integral revenue model. Therefore, results for fCS and foCS are not plotted. Since GreedyRTL assumes all EVs arrive at the same time, we set all arrival times to 1. Moreover, we assume that global peak constraint is big enough (i.e., $\sum_{j=1}^{m} p_j \leq p^{\mathsf{total}}$) such that the solution of GreedyRTL is feasible. We run GreedyRTL in each CS separately and combine the results. The final results are shown in Fig. 4.4. Along with total revenue in Fig. 4.4a, we also report total *actual* peak in Fig. 4.4b and percentage of fully charged EVs in Fig. 4.4c. As a high level trend, the results show that as the peak values increase, total revenue, total actual peak, and the number of EVs who received all their demand increase.

In Fig. 4.4a, the results for iCS and GreedyRTL are almost equal while ioCS is 84% of the other two algorithms, on average. When $p_j \geq 100$, the total revenue for iCS and GreedyRTL does not increase in Fig. 4.4a and percentage of fully charged EVs is 100 for both algorithms according to Fig. 4.4c. From this point and onward, the scheduling is not challenging for offline methods to obtain optimal answer because of resource sufficiency.



(a) Total revenue (b) Actual peak ($p^{\text{total}} = 480$) (c) % of full charged EVs

Figure 4.4 – Comparison in terms of total revenue, actual peak, and percentage of fully charged EVs by varying local peak value.



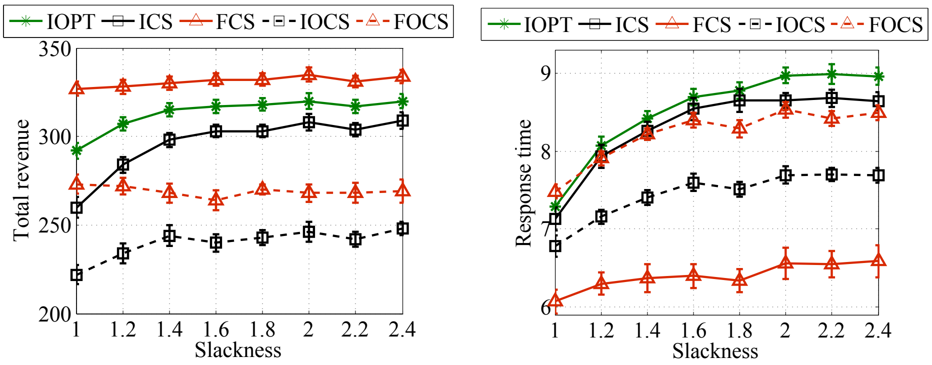
(a) Total revenue (\$) (b) Response time (hr))

Figure 4.5 – The impact of slackness parameter on total revenue and response time when users react by adjusting their demand.

However, it is still a challenge to control total actual peak for the system. As a result of proper scheduling policy through flat and right-to-left scheduling (see Section 4.4), the value of actual peak of ICS in Fig. 4.4b remains almost unchanged for $p_j \geq 100$. IOCS (as discussed earlier in this section) and GreedyRTL, however, continuously increase the peak demand, since they are not using a peak shaving method and only try to maximize total revenue. According to Fig. 4.4b, IOCS always reaches to the maximum peak of $\sum_j p_j$.

4.5.4 The Impact of Slackness Parameter

This section discusses on the benefits and disadvantages that can be brought by using slackness parameter. To give the charging scheduler more flexibility, a slackness parameter $s \geq 1$ is used and set by the system designer. Recall that charging profile of EV i is feasible if $D_i \leq k_i \frac{d_i - a_i + 1}{s}$ holds. Since users have no control on the slackness parameter



(a) Total revenue ($) (b) Response time (hr)

Figure 4.6 – The impact of slackness parameter on total revenue and response time when users react by adjusting their deadline.

and the maximum charging rate, they must adjust their demand and availability according to the imposed slackness value. Based on the charging profile feasibility equation, when the slackness value increases users have two choices to react: (i) decrease the demand and depart at the desired deadline or, (ii) extend the deadline and receive the desired demand. In the remaining of this section, we investigate the effects of either case by simulation. To have a clear picture of effects caused by slackness parameter, we generated 100 initial charging profiles $\langle a_i, d_i, v_i, D_i, k_i \rangle$ for 100 EVs randomly and uniformly chosen from 10 different models in Table 3.3 such that the initial profiles are set up assuming that the CS allows the charging operations to be finished at earliest possible time (i.e., $\frac{D_i}{k_i}$) with $s = 1$. For each EV, its demand is randomly chosen from 25% to 100% of its battery capacity and departure in interval $[a_i + \frac{sD_i}{k_i}, T]$, where $a_i + \frac{sD_i}{k_i}$ is the earliest feasible deadline. Then, the EVs are uniformly assigned to 4 CSs and should choose one of the above strategies to submit a feasible demand according to imposed slackness.

4.5.4.1 Case I- Adjusting Demands

In this case, if the initial charging profile of an EV does not reflect a feasible charging request based on the slackness parameter, the EV owner decreases its demand so it will be able to leave CS at its initial desired deadline. Note that the valuation of EVs decreases proportionally, as well. Fig. 4.5 depicts the result under this policy applied by users. As it can be seen in Fig. 4.5a and Fig. 4.5b, the general trend is that both total revenue and average response time decrease when slackness value increases. This is justifiable based on users reaction. When users decrease their demand, less electricity is sold which results in less revenue. When total demand decreases, charging can be finished in shorter time which decreases response time. Therefore, if users choose the first policy (adjusting their demand), total revenue degrades while response time improves. Notice that for the algorithms working under integral revenue model (i.e., iOPT, iCS and ioCS) the total revenue increases with slackness parameter at first (when s grows from 1 to 1.4) but then it decreases when the slackness increases more (for $s > 1.4$). In our view, this happens because increasing the slackness in integral revenue model makes it possible to *fully* charge more EVs at the beginning as the demands decrease. However, when the slackness increases more, it results in opposite effect because the valuation of demands decreases along with the demands' size.

4.5.4.2 Case II- Adjusting Deadlines

In this case, we assume that users are reluctant to decrease their desired demands. Instead, they can extend their departure. In Fig. 4.6 it can be observed that under this behavior of the users, the results are opposite as compared to the previous case (Fig. 4.5). When deadlines are extended without demand decrement, the scheduler has more chance to compliance the demands through improved scheduling flexibility. Consequently, the total revenue increases by increasing slackness value while average response time degrades.

We can conclude that when total revenue is more important than the response time, the CS should impose small values of slackness for users that apply the policy of the first case and impose higher values of slackness for the users that apply the second policy. The conclusion is reverse for the case that the objective is to have lower response times.

4.6 Conclusion

This chapter proposed offline and online algorithms for the EV charging scheduling problem under fractional and integral revenue models in an adaptive charging network (ACN). The problem is different, and more challenging than the existing single station EV charging scheduling problems since it requires respecting the aggregate peak charging demand of the ACN. As the notable contributions, our proposed online algorithm for fractional revenue model achieves constant competitive ratio of 2. Moreover, the offline integral algorithm achieves a theoretical bound on the optimality gap and approximates the optimum by 92%, on average in experiments. As a future work, we plan to study the problem under posted pricing mechanism where the charging station publishes the unit price of the power (which can be varied over time) and the users can accept or reject the offer.

Chapter 5

Summary

Contents

In this thesis, we developed efficient online charging scheduling algorithms for EVs where the power resource at charging station is limited. We first revisited the classic job scheduling problem in Chapter 2 by adding a constraint to the problem representing the maximum input power of the EVs' battery and proposed deterministic and randomized competitive algorithms. Next, in Chapter 3, we extended the problem from maximizing total valuation of charged EVs to maximizing social welfare of the users. In particular, we took into account both on-arrival charging commitment and valuation of demands for the users in the objective function. Finally, in chapter 4, we extended the scheduling problem to multiple-station setting where the goal is to find a globally optimal solution for the scheduling of EVs which are dispersed in different charging stations. We summarize the contribution of this thesis as follows.

5.1 Summary of Contributions

5.1.1 Competitive scheduling algorithm for EVs

In chapter 2, we focused on developing competitive online scheduling algorithms for charging scheduling of EVs which could be directly used in similar type of problems in other domains including cloud computing. The contributions of this chapter are twofold. First, we introduced a new proof technique for competitive analysis of scheduling algorithms devised for the studied problem in this thesis. Then, we proposed a deterministic algorithm, WFair, and a randomized algorithm, WRand, for the problem. We proved that both algorithms are $2 - \frac{1}{U}$-competitive with optimal offline solution, where U is scarcity level indicating the ratio of demand to supply. When U grow large, the competitive ratio is 2 which confirms the state of the art result. However, in practice, U is expected to be a constant which makes our result attractive.

5.1.2 Scheduling with On-Arrival Commitment

In Chapter 3, we addressed an interesting social welfare maximization problem by demonstrating that during peak hours, the charging station should be able to inform the users at their arrival time, about the minimum amount of resource that they will receive by their deadline (i.e., providing on-arrival charging commitment). We then formulate a social welfare maximization problem that has the charging commitment and valuation of the demands in the objective. In the second part of the chapter, the game theoretical aspects of the problem is studied to propose mechanisms that provide strategy-proofness and group-strategy-proofness. To the best of our knowledge, this is the first study that provides group-strategyproof scheduling algorithm for the EVs.

5.1.3 Scheduling in a Network of Charging Stations

In Chapter 4, we aim to employ the deferrable feature of high demand charging jobs of EVs and tackle EV charging scheduling problem in a network of charging stations to control the peak-demand of the network. In particular, we assume that in a charging network, multiple charging stations are available for a set of EVs to be charged. The goal is to *select* and *schedule* a subset of EVs such that: (i) the charging demand of the selected EVs are fulfilled; (ii) global and local peak constraints of the charging network are respected; and finally, (iii) the total charging network revenue obtained by the valuation of the selected EVs is maximized. As solution, we provided online and offline algorithms for two different business models and derived theoretical bounds for their worst-case performance.

5.2 Future Directions

In the remaining section, we describe what we learned from the thesis and give some open problems for future research.

5.2.1 Model-based Online Scheduling

The online scheduling algorithms proposed in this thesis are model-free i.e., they assume that information about future demand is neither available nor can be modeled. Although this is true for some cases in practice, there are scenarios that the aggregator can rely on some stochastic information about the future demand. In particular, the future demand can be modeled as a stochastic process and to be feed to the scheduling algorithm. In this way, more efficient scheduling can be generated that could increase the performance of the algorithms. It remains an open problem to extend the developed algorithms in this thesis to mode-based scheduling algorithms.

5.2.2 Deadline-Aware Scheduling Algorithms

Our proposed scheduling algorithms in this thesis are deadline-oblivious, meaning that they do not use deadline information of the users in their decision making at each time instance. Put it simply, our algorithms respect deadline but do not utilize it. In one hand, deadline-oblivious algorithms have some advantages. First, they are easier to implement in practice. Second, there might be some cases that the deadline information is not available where the deadline-oblivious algorithms can be still useful. On the other hand, in the scenarios that the deadline information is available, deadline-aware scheduling algorithms could be more efficient by utilizing the deadline information. Therefore, more research is needed to improve the proposed algorithms by making them deadline-aware.

5.2.3 Optimal Posted Price

In this thesis we assumed that each demand is determined by its valuation which is an input to the system. However, in realistic cases, the valuation of the demands may be determined by the charging system. Consider a posted pricing mechanism where the charging system announces a unit price for the power which may vary over time. Then, each user upon submitting its request to the system (e.g., mobile EVs in a city using ob-board units) either accepts the offer or refuses it without charging its EV. In such scenarios, one important challenge is to determine optimal power prices to be published at each time instance such that the revenue of the charging system is maximized during a time period.

5.2.4 Competitive Analysis for Scheduling with On-Arrival Commitment

We proved in this thesis that no scheduling algorithm can has a performance guarantee when giving on-arrival charging commitment. However, under some assumptions, it could be possible to derive performance bound for the algorithms in this case. For example, one could assume that given charging commitments can be violated in presence of a penalty. In particular, a penalty function can be added to objective function. Another example is to assume that importance ratio (i.e., the ratio of more valuable demand to less valuable one) is bounded. In these cases, a performance guarantee can be provided while keeping the practicality of the algorithms.

5.2.5 Scheduling Mobile EVs

In near feature, mobile EVs in the cities will be equipped with on-board units which to be able to submit their charging demand to the charging system or a central aggregator. The responsibility of the central aggregator in this case will be to process the received demands and produce an efficient scheduling which should include deciding on accepting/rejecting of each demand and assigning accepted demands to a charging station considering users' travel time and convenient level. This model is more complex than the multiple station setting that we studied in this thesis as the assignments of the EVs to the charging station is not given and should be computed.

Appendix A

Thesis Publications

- **B. Alinia**, M. S. Talebi, M. H. Hajiesmaili, A. Yekkehkhany, and N. Crespi, “Competitive Online Scheduling Algorithms with Applications in Deadline-Constrained EV Charging,” in *proc. IEEE/ACM International Symposium on Quality of Service (IWQoS)*, 2018.
  acceptance ratio: 20%

- **B. Alinia**, M. H. Hajiesmaili and, N. Crespi, “Online Charging Scheduling of EVs with On-Arrival Charging Commitments,” IEEE Transactions on Intelligent Transportation Systems (under revision).
  Journal Impact Factor: 3.72

- **B. Alinia**, M. H. Hajiesmaili, Z. J. Lee, N. Crespi, and E. Mallada “Online EV Scheduling Algorithms for Adaptive Charging Networks with Global Peak Constraints,” Under Submission, 2018.

- **B. Alinia**, M. H. Hajiesmaili, A. Khonsari and, N. Crespi, “Deadline-Constrained Maximum-Quality Aggregation Tree Construction in Wireless Sensor Networks,” IEEE Sensors Journal, 2017.
  Journal Impact Factor: 2.5

- H. Yousefi, , M. Mehrabi, **B. Alinia**, and K. G. Shin, “Maximizing Quality of Aggregation in WSNs Under Deadline and Interference Constraints,” in *proc. IEEE International Conference on Sensing, Communication and Networking (SECON)*, 2018.
  acceptance ratio: 23%

- S. N. Han, Q. H. Cao, **B. Alinia**, and N Crespi, “Design, implementation, and evaluation of 6LoWPAN for home and building automation in the Internet of Things,” in *proc. IEEE/ACS AICCSA*, 2015.

References

[1] http://energy.gov/eere/vehicles/.

[2] J. de Hoog, T. Alpcan, M. Brazil, D. A. Thomas, and I. Mareels, “Optimal charging of electric vehicles taking distribution network constraints into account,” *IEEE Transactions on Power Systems*, vol. 30, no. 1, pp. 365–375, 2015.

[3] T. A. Short, *Electric power distribution handbook.* CRC press, 2014.

[4] C.-K. Wen, J.-C. Chen, J.-H. Teng, and P. Ting, “Decentralized plug-in electric vehicle charging selection algorithm in power systems,” *IEEE Transactions on Smart Grid*, vol. 3, no. 4, pp. 1779–1789, 2012.

[5] Q. Xiang, F. Kong, X. Liu, X. Chen, L. Kong, and L. Rao, “Auc2charge: An online auction framework for eectric vehicle park-and-charge,” in *Proceedings of the 2015 ACM Sixth International Conference on Future Energy Systems.* ACM, 2015, pp. 151–160.

[6] https://www.teslamotors.com/fr_fr/supercharger?redirect=no.

[7] N. Nisan, T. Roughgarden, E. Tardos, and V. V. Vazirani, *Algorithmic game theory.* Cambridge University Press Cambridge, 2007, vol. 1.

[8] D. Wu, H. Zeng, C. Lu, and B. Boulet, “Two-stage energy management for office buildings with workplace ev charging and renewable energy,” *IEEE Transactions on Transportation Electrification*, vol. 3, no. 1, pp. 225–237, 2017.

[9] G. Lee, T. Lee, Z. Low, S. H. Low, and C. Ortega, “Adaptive charging network for electric vehicles,” in *Proc. of IEEE GlobalSIP*, 2016.

[10] P. Gas and R. I. Electric Company. http://www.pge.com/nots/rates/tariffs/rateinfo.shtml.

[11] Y. Zhang, M. Hajiesmaili, S. Cai, M. Chen, and Q. Zhu, “Peak-aware online economic dispatching for microgrids,” *IEEE Transactions on Smart Grid*, 2016.

[12] A. S. Manne, “On the job-shop scheduling problem,” *Operations Research*, vol. 8, no. 2, pp. 219–223, 1960.

[13] L. Davis, “Job shop scheduling with genetic algorithms,” in *International conference on genetic algorithms and their applications*, vol. 140, 1985.

[14] M. Weiser, B. Welch, A. Demers, and S. Shenker, “Scheduling for reduced CPU energy,” in *1st USENIX conference on Operating Systems Design and Implementation*, 1994.

[15] F. Yao, A. Demers, and S. Shenker, “A scheduling model for reduced CPU energy,” in *36th Annual Symposium on Foundations of Computer Science (FOCS)*, 1995, pp. 374–382.

[16] Y. Zheng, B. Ji, N. Shroff, and P. Sinha, “Forget the deadline: Scheduling interactive applications in data centers,” in *IEEE 8th International Conference on Cloud Computing (CLOUD)*, 2015, pp. 293–300.

[17] Y. Azar, I. Kalp-Shaltiel, B. Lucier, I. Menache, J. S. Naor, and J. Yaniv, “Truthful online scheduling with commitments,” in *Proceedings of the 16th ACM Conference on Economics and Computation (EC)*, 2015, pp. 715–732.

[18] Y. Zheng, *Scheduling and design in cloud computing systems.* PhD thesis, The Ohio State University, 2015.

[19] Z. Zheng and N. B. Shroff, “Online multi-resource allocation for deadline sensitive jobs with partial values in the cloud,” in *IEEE INFOCOM*, 2016, pp. 1–9.

[20] F. Li, “Competitive scheduling of packets with hard deadlines in a finite capacity queue,” in *IEEE INFOCOM*, 2009, pp. 1062–1070.

[21] B. Lucier, I. Menache, J. S. Naor, and J. Yaniv, “Efficient online scheduling for deadline-sensitive jobs,” in *Twenty-fifth annual ACM symposium on Parallelism in algorithms and architectures*, 2013, pp. 305–314.

[22] Y. He, S. Elnikety, J. Larus, and C. Yan, “Zeta: Scheduling interactive services with partial execution,” in *Proceedings of the ACM Symposium on Cloud Computing*, 2012.

[23] Y. He, Z. Ye, Q. Fu, and S. Elnikety, “Budget-based control for interactive services with adaptive execution,” in *Proceedings of the 9th International Conference on Autonomic Computing (ICAC)*, 2012, pp. 105–114.

[24] M. T. Hajiaghayi, “Online auctions with re-usable goods,” in *6th ACM Conference on Electronic Commerce*, 2005, pp. 165–174.

[25] A. Gupta, R. Krishnaswamy, and K. Pruhs, “Online primal-dual for non-linear optimization with applications to speed scaling,” in *International Workshop on Approximation and Online Algorithms*, 2012, pp. 173–186.

[26] M. Chrobak, L. Epstein, J. Noga, J. Sgall, R. van Stee, T. Tichỳ, and N. Vakhania, “Preemptive scheduling in overloaded systems,” *Automata, Languages and Programming*, pp. 800–811, 2002.

[27] F. Y. Chin and S. P. Fung, “Online scheduling with partial job values: Does timesharing or randomization help?” *Algorithmica*, vol. 37, no. 3, pp. 149–164, 2003.

[28] ——, “Improved competitive algorithms for online scheduling with partial job values,” in *International Computing and Combinatorics Conference*, 2003, pp. 425–434.

[29] S. Chen, L. Tong, and T. He, “Optimal deadline scheduling with commitment,” in *49th Annual Allerton Conference on Communication, Control, and Computing*, 2011, pp. 111–118.

[30] E.-C. Chang and C. Yap, “Competitive on-line scheduling with level of service,” *Journal of Scheduling*, vol. 6, no. 3, pp. 251–267, 2003.

[31] N. Jain, I. Menache, J. S. Naor, and J. Yaniv, “Near-optimal scheduling mechanisms for deadline-sensitive jobs in large computing clusters,” *ACM Transactions on Parallel Computing*, vol. 2, no. 1, pp. 3:1–3:29, 2015.

[32] X. Wu and P. Loiseau, “Algorithms for scheduling deadline-sensitive malleable tasks,” in *53rd Annual Allerton Conference on Communication, Control, and Computing)*, 2015, pp. 530–537.

[33] F. Li, J. Sethuraman, and C. Stein, “An optimal online algorithm for packet scheduling with agreeable deadlines,” in *Sixteenth annual ACM-SIAM symposium on Discrete algorithms*, 2005, pp. 801–802.

[34] A. Kesselman, Z. Lotker, Y. Mansour, B. Patt-Shamir, B. Schieber, and M. Sviridenko, “Buffer overflow management in qos switches,” *SIAM Journal on Computing*, vol. 33, no. 3, pp. 563–583, 2004.

[35] B. Hajek, “On the competitiveness of on-line scheduling of unit-length packets with hard deadlines in slotted time,” in *IEEE Information Sciences and Systems*, 2001.

[36] M. Chrobak, W. Jawor, J. Sgall, and T. Tichỳ, “Improved online algorithms for buffer management in qos switches,” in *European Symposium on Algorithms*, 2004, pp. 204–215.

[37] J. C. Mukherjee and A. Gupta, “A review of charge scheduling of electric vehicles in smart grid,” *IEEE Systems Journal*, vol. 9, no. 4, pp. 1541–1553, 2015.

[38] W. Tang, S. Bi, and Y. J. Zhang, “Online charging scheduling algorithms of electric vehicles in smart grid: An overview,” *IEEE Communications Magazine*, vol. 54, no. 12, pp. 76–83, 2016.

[39] R. Deng and H. Liang, “Whether to charge an electric vehicle or not? A near-optimal online approach,” in *IEEE Power and Energy Society General Meeting (PESGM)*, 2016, pp. 1–5.

[40] W. Tang, S. Bi, and Y. J. Zhang, “Online speeding optimal charging algorithm for electric vehicles without future information,” in *IEEE International Conference on Smart Grid Communications (SmartGridComm)*, 2013, pp. 175–180.

[41] W. Tang, S. Bi, and Y. J. A. Zhang, “Online coordinated charging decision algorithm for electric vehicles without future information,” *IEEE Transactions on Smart Grid*, vol. 5, no. 6, pp. 2810–2824, 2014.

[42] E. H. Gerding, V. Robu, S. Stein, D. C. Parkes, A. Rogers, and N. R. Jennings, “Online mechanism design for electric vehicle charging,” in *The 10th International Conference on Autonomous Agents and Multiagent Systems-Volume 2*, 2011, pp. 811–818.

[43] V. Robu, S. Stein, E. H. Gerding, D. C. Parkes, A. Rogers, and N. R. Jennings, “An online mechanism for multi-speed electric vehicle charging,” in *International Conference on Auctions, Market Mechanisms and Their Applications*, 2011, pp. 100–112.

[44] V. Robu, E. H. Gerding, S. Stein, D. C. Parkes, A. Rogers, and N. R. Jennings, “An online mechanism for multi-unit demand and its application to plug-in hybrid electric vehicle charging,” *Journal of Artificial Intelligence Research*, vol. 48, pp. 175–230, 2013.

[45] S. Chen and L. Tong, “iems for large scale charging of electric vehicles: Architecture and optimal online scheduling,” in *IEEE International Conference on Smart Grid Communications (SmartGridComm)*, 2012, pp. 629–634.

[46] S. Chen, Y. Ji, and L. Tong, “Large scale charging of electric vehicles,” in *IEEE Power and Energy Society General Meeting (PESGM)*, 2012, pp. 1–9.

[47] W. Tang, S. Bi, and Y. J. A. Zhang, “Online coordinated charging decision algorithm for electric vehicles without future information,” *IEEE Transactions on Smart Grid*, vol. 5, no. 6, pp. 2810–2824, 2014.

[48] Y. He, B. Venkatesh, and L. Guan, "Optimal scheduling for charging and discharging of electric vehicles," *IEEE transactions on smart grid*, vol. 3, no. 3, pp. 1095–1105, 2012.

[49] A. Malhotra, G. Binetti, A. Davoudi, and I. D. Schizas, "Distributed power profile tracking for heterogeneous charging of electric vehicles," *IEEE Transactions on Smart Grid*, 2016.

[50] M. Moradijoz, M. P. Moghaddam, M. Haghifam, and E. Alishahi, "A multi-objective optimization problem for allocating parking lots in a distribution network," *International Journal of Electrical Power & Energy Systems*, vol. 46, pp. 115–122, 2013.

[51] D. Wang, X. Guan, J. Wu, P. Li, P. Zan, and H. Xu, "Integrated energy exchange scheduling for multimicrogrid system with electric vehicles," *IEEE Transactions on Smart Grid*, vol. 7, no. 4, pp. 1762–1774, 2016.

[52] M. Zeng, S. Leng, Y. Zhang, and J. He, "Qoe-aware power management in vehicle-to-grid networks: a matching-theoretic approach," *IEEE Transactions on Smart Grid*, 2016.

[53] M. F. Shaaban, M. Ismail, E. F. El-Saadany, and W. Zhuang, "Real-time pev charging/discharging coordination in smart distribution systems," *IEEE Transactions on Smart Grid*, vol. 5, no. 4, pp. 1797–1807, 2014.

[54] S. Zhao, X. Lin, and M. Chen, "Peak-minimizing online ev charging: Price-of-uncertainty and algorithm robustification," in *Proc. of IEEE INFOCOM*, 2015.

[55] E. L. Karfopoulos and N. D. Hatziargyriou, "Distributed coordination of electric vehicles providing v2g services," *IEEE Transactions on Power Systems*, vol. 31, no. 1, pp. 329–338, 2016.

[56] Z. Zheng and N. Shroff, "Online welfare maximization for electric vehicle charging with electricity cost," in *Proceedings of the 5th international conference on Future energy systems*. ACM, 2014, pp. 253–263.

[57] W. Tang and Y. J. A. Zhang, "A model predictive control approach for low-complexity electric vehicle charging scheduling: optimality and scalability," *IEEE Transactions on Power Systems*, vol. 32, no. 2, pp. 1050–1063, 2017.

[58] S. Chen and L. Tong, "iEMS for large scale charging of electric vehicles: Architecture and optimal online scheduling," in *Proc. of SmartGridComm*, 2012.

[59] V. Robu, E. H. Gerding, S. Stein, D. C. Parkes, A. Rogers, and N. R. Jennings, "An online mechanism for multi-unit demand and its application to plug-in hybrid electric vehicle charging," *Journal of Artificial Intelligence Research*, vol. 48, pp. 175–230, 2013.

[60] Y. Zhou, D. Yau, P. You, and P. Cheng, "Optimal-cost scheduling of electrical vehicle charging under uncertainty," *IEEE Transactions on Smart Grid*, 2017.

[61] A. Borodin and R. El-Yaniv, *Online computation and competitive analysis.* Cambridge University Press, 1998.

[62] Z. Zhang, Z. Li, and C. Wu, "Optimal posted prices for online cloud resource allocation," in *ACM SIGMETRICS*, 2017.

[63] K. Funaoka, S. Kato, and N. Yamasaki, "Work-conserving optimal real-time scheduling on multiprocessors," in *Euromicro Conference on Real-Time Systems (ECRTS)*, 2008, pp. 13–22.

[64] R. Motwani and P. Raghavan, *Randomized algorithms.* Chapman & Hall/CRC, 2010.

[65] "Gurobi optimizer 5.0," *Gurobi: http://www. gurobi. com*, 2013.

[66] https://www.bloomberg.com/features/2016-ev-oil-crisis/.

[67] T. Koslowski, “The electric vehicle‚Äôs value chain and technology evolution,” Technical report, Gartner Inc, Tech. Rep., 2009.

[68] Q. Kang, S. Feng, M. Zhou, A. C. Ammari, and K. Sedraoui, “Optimal load scheduling of plug-in hybrid electric vehicles via weight-aggregation multi-objective evolutionary algorithms,” *IEEE Transactions on Intelligent Transportation Systems*, 2017.

[69] Q. Xiang, F. Kong, X. Liu, X. Chen, L. Kong, and L. Rao, “Auc2charge: An online auction framework for eectric vehicle park-and-charge,” in *Proceedings of the 2015 ACM Sixth International Conference on Future Energy Systems*. ACM, 2015, pp. 151–160.

[70] Z. Zheng and N. Shroff, “Online welfare maximization for electric vehicle charging with electricity cost,” in *Proceedings of the 5th international conference on Future energy systems*. ACM, 2014, pp. 253–263.

[71] Z. Darabi, P. Fajri, and M. Ferdowsi, “Intelligent charge rate optimization of phevs incorporating driver satisfaction and grid constraints,” *IEEE Transactions on Intelligent Transportation Systems*, vol. 18, no. 5, pp. 1325–1332, 2017.

[72] E. Akhavan-Rezai, M. F. Shaaban, E. F. El-Saadany, and F. Karray, “Online intelligent demand management of plug-in electric vehicles in future smart parking lots,” *IEEE Systems Journal*, vol. 10, no. 2, pp. 483–494, 2016.

[73] T. Zhang, W. Chen, Z. Han, and Z. Cao, “Charging scheduling of electric vehicles with local renewable energy under uncertain electric vehicle arrival and grid power price,” *IEEE Transactions on Vehicular Technology*, vol. 63, no. 6, pp. 2600–2612, 2014.

[74] A. Gusrialdi, Z. Qu, and M. A. Simaan, “Distributed scheduling and cooperative control for charging of electric vehicles at highway service stations,” *IEEE Transactions on Intelligent Transportation Systems*, 2017.

[75] S. Zhao, X. Lin, and M. Chen, “Peak-minimizing online ev charging: Price-of-uncertainty and algorithm robustification,” in *IEEE INFOCOM*, 2015, pp. 2335–2343.

[76] ——, “Robust online algorithms for peak-minimizing ev charging under multi-stage uncertainty,” *IEEE Transactions on Automatic Control*, 2017.

[77] S. Stein, E. Gerding, V. Robu, and N. 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-Volume 2*. International Foundation for Autonomous Agents and Multiagent Systems, 2012, pp. 669–676.

[78] A. Ghosh and V. Aggarwal, “Control of charging of electric vehicles through menu-based pricing,” *IEEE Transactions on Smart Grid*, 2017.

[79] B. Sun, Z. Huang, X. Tan, and D. H. Tsang, “Optimal scheduling for electric vehicle charging with discrete charging levels in distribution grid,” *IEEE Transactions on Smart Grid*, 2016.

[80] A. Borodin and R. El-Yaniv, *Online computation and competitive analysis.* cambridge university press, 2005.

[81] S. Han, U. Topcu, and G. J. Pappas, “An approximately truthful mechanism for electric vehicle charging via joint differential privacy,” in *American Control Conference (ACC), 2015.* IEEE, 2015, pp. 2469–2475.

[82] C.-K. Wen, J.-C. Chen, J.-H. Teng, and P. Ting, "Decentralized plug-in electric vehicle charging selection algorithm in power systems," *IEEE Transactions on Smart Grid*, vol. 3, no. 4, pp. 1779–1789, 2012.

[83] H. Kellerer, U. Pferschy, and D. Pisinger, "Knapsack problems," 2005.

[84] R. B. Myerson, "Optimal auction design," *Mathematics of operations research*, vol. 6, no. 1, pp. 58–73, 1981.

[85] D. C. Parkes and Q. Duong, "An ironing-based approach to adaptive online mechanism design in single-valued domains," in *AAAI*, vol. 7, 2007, pp. 94–101.

[86] H. Moulin, "Incremental cost sharing: Characterization by coalition strategy-proofness," *Social Choice and Welfare*, vol. 16, no. 2, pp. 279–320, 1999.

[87] Z. Darabi and M. Ferdowsi, "Plug-in hybrid electric vehicles: Charging load profile extraction based on transportation data," in *Power and Energy Society General Meeting, 2011 IEEE.* IEEE, 2011, pp. 1–8.

[88] Y. Cao and N. Wang, "Toward efficient electric-vehicle charging using vanet-based information dissemination," *IEEE Transactions on Vehicular Technology*, vol. 66, no. 4, pp. 2886–2901, 2017.

[89] A. Santos, N. McGuckin, H. Y. Nakamoto, D. Gray, and S. Liss, "Summary of travel trends: 2009 national household travel survey," Tech. Rep., 2011.

[90] https://www.teslamotors.com/fr_fr/supercharger?redirect=no.

[91] B. Alinia, M. H. Hajiesmaili, Z. J. Lee, N. Crespi, and E. Mallada, "Online ev scheduling algorithms for adaptive charging networks with global peak constraints," *https://github.com/bahramalinia/TechReport/blob/master/Appendix.pdf*, 2017.

[92] V. V. Vazirani, *Approximation algorithms.* Springer Science & Business Media, 2013.

[93] R. D. Carr, L. K. Fleischer, V. J. Leung, and C. A. Phillips, "Strengthening integrality gaps for capacitated network design and covering problems," Sandia National Labs., Albuquerque, NM (US); Sandia National Labs., Livermore, CA (US), Tech. Rep., 1999.

[94] N. Jain, I. Menache, J. S. Naor, and J. Yaniv, "Near-optimal scheduling mechanisms for deadline-sensitive jobs in large computing clusters," *ACM Transactions on Parallel Computing*, vol. 2, no. 1, p. 3, 2015.

[95] E.-C. Chang and C. Yap, "Competitive on-line scheduling with level of service," *Journal of Scheduling*, vol. 6, no. 3, pp. 251–267, 2003.

[96] A. Santos, N. McGuckin, H. Y. Nakamoto, D. Gray, and S. Liss, "Summary of travel trends: 2009 national household travel survey," Tech. Rep., 2011.

List of figures

List of tables