CHALMERS | GÖTEBORG UNIVERSITY

Technical Report No. 2015:01

Tailor your curves after your costume: Supply-following demand in Smart Grids through the Adwords problem

GIORGOS GEORGIADIS
IOSIF SALEM
MARINA PAPATRIANTAFILOU

Department of Computer Science and Engineering
CHALMERS UNIVERSITY OF TECHNOLOGY/
GÖTEBORG UNIVERSITY
Göteborg, Sweden, 2015

Tailor your curves after your costume: Supply-following demand in Smart Grids through the Adwords problem

Giorgos Georgiadis, Iosif Salem, Marina Papatriantafilou
Department of Computer Science and Engineering, Chalmers University of Technology
S-412 96 Göteborg, Sweden
Email: {georgiog,iosif,ptrianta}@chalmers.se, Fax: +46-31-7723663

Abstract

In deregulated energy markets, consumers -ranging from households to data centers- have access to multiple offers, often through multiple suppliers and energy carriers (i.e. electric, thermal) or through local generation, such as renewable energy sources and energy storage. Ideally, supply should match demand, leading to a balanced power grid, but this is challenging in practice: while some generation sources can be planned in advance (e.g. utility offers), others can be planned to a limited degree or cannot be planned altogether (e.g. storage and renewable energy sources respectively). In this context, we focus on how to address systematically this complex resource allocation problem in the presence of multiple actors.

In this work, utilizing a proposed modeling of the energy dispatch problem as an online scheduling problem, we model supply-following demand in terms of the Adwords problem, in order to provide algorithmic solutions of measurable quality. Building on previous work, we extend the Adwords problem to incorporate load credit (i.e. storage) and we present and analyze online algorithms that can schedule demand, given availability constraints on supply, with guaranteed competitive ratio. In systems where demands are small compared to the individual supply, we prove a $\left(1 - \frac{1}{e}\right)$-competitive ratio. For cases where this does not hold, we extend the Adwords problem to utilize dynamic budgets, and present an algorithm with a $\frac{1}{2}$-competitive ratio. We also provide examples of algorithmic performance in real world scenarios, by utilizing long term, fine-grained data from a pilot project in Sweden, while taking into account renewable generation on site.

1 Introduction

Until recently, electric power grids were operating under the paradigm of a utility service: the utility company must provide energy to the consumer consistently, by satisfying all incoming demands regardless of supply. Along with the increasing penetration of renewable energy sources (e.g. photovoltaic arrays, wind generator farms), as well as loads that are distributed or that may have less regular patterns compared to households (e.g. electric car fleets, data centers), this paradigm has began shifting towards a market-oriented one, where generated energy comes from many different sources and is brokered to consumers through utilities or electricity vendors.

Under such a paradigm and the associated enhanced power grid (often referred to as *Smart Grid* [1]), end consumers have the opportunity to utilize energy from various sources (e.g. utility company, local storage) and decide based on some information, most commonly price-related, associated to the sources. While research on the appropriate pricing information (or *pricing signal* [2]) and tariff structure needed on the utility side is ongoing, the question remains: how to ensure high resource (i.e. energy) utilization on a system level, in the presence of multiple energy offers, such as storage and renewable energy sources, that can be controlled up to a certain degree or not at all.

In this work, we look at the problem of resource utilization from a perspective orthogonal to pricing: instead of looking into techniques that can be used to influence end consumers into consuming energy at opportune times (such as pricing signals), we assume the existence of multiple such mechanisms (in the form of multiple offers) and address the utilization problem of available energy supply through flexibility in demand (or *supply-following demand* [3]).

This new perspective allows for connections to be made between the energy utilization problem and the well known Adwords problem [4] and, based on these connections, we provide two algorithms that can solve the utilization problem in a variety of supply-related assumptions, along with their rigorous analysis for online guarantees. In parallel, we identify an extension of the Adwords problem that includes dynamic budgets and is particularly relevant here, and address it through our algorithms. We also show the functionality of the proposed algorithms through the use of real generation and consumption data in relevant experiments, focusing on their ability to shape demand according to supply, as well as on the utilization of the renewable energy sources available.

The rest of the paper is structured as follows. In section 2 we define our problem formally and provide necessary background knowledge. A novel modeling for the energy utilization problem based on the Adwords problem is presented in section 3, along with two algorithms that utilize the modeling, together with their analysis. The experimental study that evaluates the algorithmic solution against real consumption data from a pilot housing project can be found in section 4. In section 5 we provide an extensive discussion on the properties of our algorithms, both analytically and experimentally, as well as of the proposed modeling. Finally, we conclude with some closing remarks in section 6.

1.1 Related work

A number of approaches in the literature exist regarding the problem of supply-following demand. In [5], Kok et al focus on both the supply and demand, and use a hierarchical mechanism and a market structure to match consumers with producers, with the ultimate goal of reducing peaks in consumption. On the other hand, Barker et al [6] focus on background consumption loads (i.e. loads that the consumer does not interact with), and by applying scheduling techniques such as a variation of the Earliest Deadline First algorithm, they shift demand during the day in order to reduce peaks in consumption. The works of Lu et al [7] and Tu et al [8] are closer to the context of the current work, since both present online algorithms with proven competitive ratios. However, they do so for special cases of interest: in [7], Lu et al focus on fast-responding generators (e.g. gas or diesel turbines) and present an algorithm that operates for any combination of demand, supply and price, and in [8], Tu et al focus on data centers and on a cost minimization problem where price is a parameter. In addition, Georgiadis et al [9] present a novel modeling and an online algorithm that can schedule flexible demand in order to reduce peaks in consumption in scenarios where forecasts are unreliable or not available (e.g. renewable energy sources, energy storage). Nevertheless, one common element of all approaches above is that the optimization goal is the reduction of peak demand in the considered time period, while the criterion in this work is utilization of all available supply[1]. To our knowledge, this is the first work that addresses the supply-following demand problem on this basis.

On the other hand, there is significant research interest on pricing policies about Smart Grids. For example, both [2] and [10] compare different pricing schemes in real world scenarios and make recomendations in their individual contexts: [2] shows that a real time tariff can have a positive outcome in reducing consumption, while [10] points the risk of automated solutions adopting the exact same behavior under the same pricing signals and suggests the introduction of different prices to different customers, among other remedies (cf also [10] for a short survey on pricing policies). In their respective works, Mohsenian-Rad et al [11], Caron and Kesidis [12], Wijaya et al [13] and Carpenter et al [14] look into pricing mechanisms from a game-theoretical approach, albeit with some differences. Both Mohsenian-Rad et al [11] and Caron and Kesidis [12] focus on the distributed elements of their respective approaches, while Caron and Kesidis [12] introduce elements from online analysis by identifying both cases of complete and incomplete knowledge. Wijaya et al [13] on the other hand proceeds to explicitly cut down peaks of demand, before applying their game-theoretic approach. Finally, Carpenter et al [14] conduct experiments in a real world setting and find surprisingly that their game-theoretic approach manages to increase peaks of demand, contrary to other finding in the literature. As mentioned above, the algorithms and the modeling presented in this work can be applied in parallel with any of the above approaches to pricing mechanisms.

[1] I.e. a peak might be desirable in a specific time where supply is too high, e.g. due to increased wind generation.

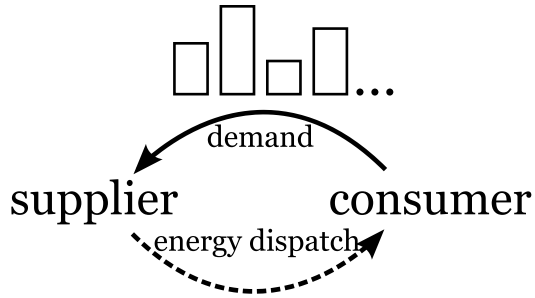

Figure 1: A schematic representation of the energy dispatch problem

2 Background

2.1 Scheduling for the energy dispatch problem

Following [9], this paper utilizes a high level abstraction of the distribution grid, with *consumer* sites (or *nodes*) issuing *load demand requests* at arbitrary intervals, and energy being dispatched from *supplier* sites (or nodes) to satisfy these demand requests (*energy dispatch problem*, cf figure 1). We identify properties of interest for these requests, such as elasticity, storage capabilities and specific energy carrier utilization (i.e. thermal/electric) in order to model time flexibility for the delivery of the requested energy, the ability to store the requested energy for later use and the ability to utilize specific energy carriers for the satisfaction of the request, respectively. By regarding demands originating from consumer nodes as *tasks* and the supplier nodes themselves as *machines*, the aforementioned problem of energy dispatch from suppliers to consumers can be modeled as a *scheduling* problem of tasks to machines: issued load demands correspond to incoming tasks that need to be assigned to a machine, or equivalently, to be serviced by a supplier node. The machines correspond both to the different energy carriers available to tasks but also to the *timeslots* that each carrier is subdivided in, i.e. if a carrier is available for the next window of slots, e.g. 24 hours, and we consider 1 hour timeslots, then 24 machines will be dedicated to this carrier. When an incoming task is assigned to a machine, it incurs a *load* equal to the corresponding demand request. More formally, let $M_i, i = 0, \ldots, n-1$ be a set of machines where variable load credit (i.e. storage) can accumulate and $t_0, \ldots, t_j, \ldots$ be an input task sequence of two task types, *simple* and *storage*, with the following properties: each task t_j of both types has load w_j and restrictions on the allowable machines it can run on, while storage tasks additionally create on all machines load credit equal or less to their load w_j (with the possibility of 0 on some but not all machines). Note that the total number of machines n is equal to the number of timeslots over all energy carriers considered.

The example of figure 2 illustrates the above modeling: two energy carriers exist (i.e. thermal and electric), each with 24 hourly timeslots (total of 48 machines), storage tasks are marked with an S while simple tasks are not and, as an example, task t_j has a set of allowed machines equal to $\{M_{46}, M_{47}\}$.

The scheduling problem described above is NP-complete [15], and as such, research

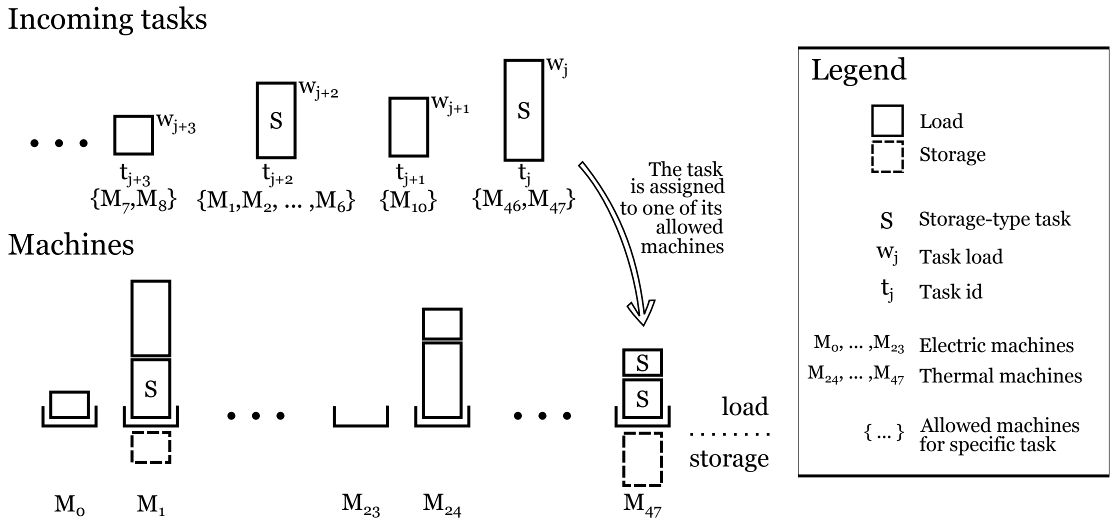

Figure 2: An example of the online load balancing problem for electrical and thermal energy requests with storage capabilities

efforts have been focused on approximation algorithms that solve it in a time complexity as close to the optimal algorithm as possible. Moreover, in this work we focus on the online version of the problem, and evaluate the proposed algorithmic solution using the concept of the *competitive ratio* [15]:

Definition 1 (Competitive ratio [15]) *For a maximization problem, we call an algorithm ALG* c-competitive *if there is a constant α such that for all finite input sequences I,*

$$ALG\,(I) \leq c \cdot OPT\,(I) + \alpha,$$

where $ALG\,(I)$ is the cost associated with the solution produced by ALG for input I and $OPT\,(I)$ is the corresponding cost for the optimal algorithm OPT and the input I.

2.2 The ADWORDS problem

The following generalization of the online vertex-weighted bipartite matching problem called ADWORDS [4] is considered here: in a bipartite graph $G = (U \cup V, E)$ with bipartition (U, V) where nodes $v \in V$ arrive (alt. are revealed) online, each node $u \in U$ has a budget B_u, and edges $(u, v) \in E$ have bids bid_{uv} (fig. 3). When an new node v arrives, it is matched to a neighbor u by spending an amount of bid_{uv} from its budget. When a node spends its entire budget it becomes unavailable, while the overall goal is to maximize the total budget spent. An important distinction that leads to different variations of the ADWORDS problem is whether the following assumption holds:

Definition 2 (Small bids assumption) *For each u, v, bid_{uv} is very small compared to B_u.*

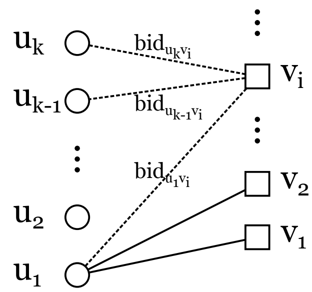

Figure 3: A schematic representation of the ADWORDS problem in graph form

The defining characteristic of cases where this assumption does not hold is the treatment of nodes with nearly exhausted budgets. Such nodes u continue to bid bid_{uv} for nodes $v \in V$ (since their budgets are not exhausted) but they may be unable to pay the full amount of the bid, due to their limited budget. A possible design choice regarding this problem is to consider a payable amount of $min\,(bid_{uv}, B_u - bid_{uv})$ instead of bid_{uv}, but this leads inevitably to fundamentally different solutions and analyses from cases where the small bids assumption holds [4].

In the continuation of this work, the above mentioned scheduling-based modeling for the energy dispatch problem will be connected to the ADWORDS problem, in order to present algorithms that can that can schedule demand, given availability constraints on supply, with guaranteed competitive ratio. The next section presents two different algorithms, addressing cases where the small bids assumption holds or not respectively, along with the formal proofs of their competitive ratios when compared to offline algorithms that assume perfect knowledge of the problem.

3 Modeling, algorithms and analysis

The fundamental modeling described above still applies here: incoming load demand requests are considered tasks that are arriving online and need to be assigned to machines, which in turn correspond to both the different energy carriers available to tasks, as well as the timeslots that each energy carrier is subdivided in. In the following, we will consider also different suppliers that provide services of possibly the same type of energy carriers, each of which corresponds to a distinct set of machines. For example, these can be different utilities or energy vendors that provide competitive electricity services to end customers. Note that the optimization goal considered in this paper, sometimes also called *supply-following demand* in the literature [3], is distinctly different from the minimization of consumption peaks in [9]:

Definition 3 (Supply-following energy dispatch) *We consider the energy dispatch problem from suppliers to consumers, with a given availability of the suppliers' energy as well as a set of features regarding the consumers' load requests (i.e. load elasticity, energy storage, utilization of alternative energy carriers) that may or may not be present on all requests. We define as* supply-following energy dispatch *the problem of maximizing the utilization of the given supply by using the individual features of load requests.*

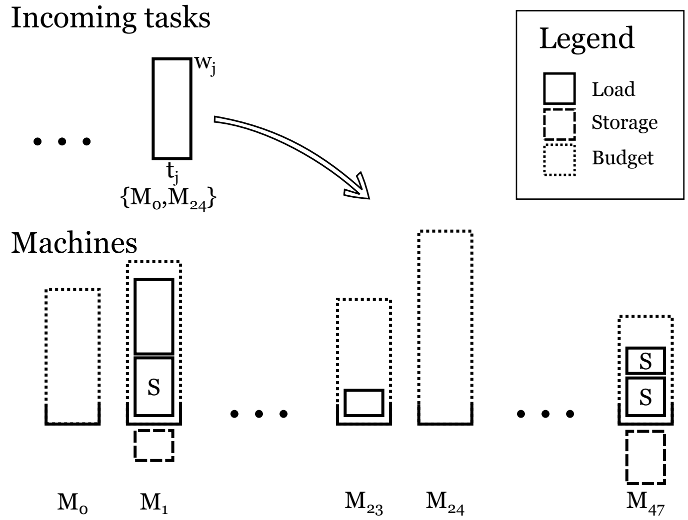

Figure 4: An example of task assignment to machines with budget information

In order to systematically address the supply-following energy dispatch problem, we refine the concept of *utilization*, mentioned in the previous definition, by introducing an additional parameter: that of an individual *budget* for each machine. This parameter expresses naturally the supplier's available quantity of energy through a particular energy carrier for a particular timeslot, and can be used to express an upper bound in a machine's allowable load and therefore quantify the machine's utilization.

Figure 4 shows an example of a single supplier with two energy carriers, electric and thermal, and different budget information for each timeslot. In this example, task t_j that can utilize both the electric machine M_0 and the thermal machine M_{24} can be assigned to machine M_0. However, since machine M_0 has a limited budget compared to M_{24}, this might not be possible for similar tasks in the future since the machine will stop accepting tasks as soon as its budget is exhausted. In fact, exhausting the budgets *of all machines* is the desirable state of the system, since a supplier's goal is to utilize (i.e. sell) all available capacity, while avoiding to go over budget due to unforeseen events (i.e. extreme weather conditions calling for extra capacity).

3.1 Problem modeling

Using the above mentioned budget information, we formulate the supply-following energy dispatch problem in terms of the modeling and utilize it as an optimization goal in the rest of the paper:

Definition 4 (SupplyBudgetUtilization) *Consider the energy dispatch problem from suppliers to consumers, modeled using the machines and tasks modeling of section 2 (i.e. including the possibility of storage accumulation on machines), where suppliers have budgets on the available energy for the different energy carriers and timeslots, and where servicing a demand by a supplier implies using part of its budget. We consider the problem of maximizing the budget utilization on the suppliers' energy carriers (equiv. machines) over all energy carriers and timeslots considered, in the* *presence of the consumers' expressed demands (equiv. tasks) with their associated restrictions.*

Note that due to a finite total budget, we consider a static timeframe equal to the set of timeslots, as well as a finite number of generated tasks within this timeframe. This assumption fits well the energy market practice of considering a daily operations cycle regarding energy procurement and distribution to the end consumer (day-ahead markets). In fact, the following algorithms, analysis and modeling apply also when considering a rolling window as a timeframe, since machines are allowed to enter or leave at any point (cf section 5).

In order to solve the SupplyBudgetUtilization problem, we allow machines to bid on incoming tasks (subjected to the tasks' individual characteristics) using the "funds" of their budgets, which allows the direct mapping of the SupplyBudgetUtilization problem to the Adwords problem mentioned above. In the continuation, we will use the generic expression bid_{uv} for the bid of machine u for task v, which represents any suitable bidding metric. We will also denote with U and V the sets of machines and tasks respectively and we will use the term "node" for both machines and tasks, when it is clear from the context where we are referring to.

Note that the concept of machines bidding for tasks may seem initially counterintuitive: for example, in electricity markets, consumers' demand requests are satisfied by paying suppliers for the requested amount of energy for a particular timeslot. However, the connection between the two cases can be easily understood by taking into account ordinary market *price* information about the individual machines, and noting that price and bid are inverse measures: the more expensive a machine is, the larger the bid that the machine will have to make for a particular task (cf also section 5).

Finally, note that the small bids assumption mentioned in section 2 is valid in the SupplyBudgetUtilization problem we described above: we assume that the energy budgets of utilities and energy vendors are big when compared to the bids towards individual energy demands of, e.g., household appliances and generic consumer devices. In section 3 we will see a version of the SupplyBudgetUtilization problem where the small bids assumption is not valid any more, corresponding to e.g. small, local scale generation or storage.

3.2 The SGSupplyAdwords algorithm

In order to solve the SupplyBudgetUtilization problem with the small bids assumption, we follow a bidding process, according to which machines bid on each new task, and the task gets assigned on the "winning" machine, which in turn subtracts the bid from its budget. Following [16], we do not simply award the task to the highest bidder but we modify the bid by a factor that takes into account the currently spent amount of the bidder's budget, and we award the task to the highest modified bidder instead. The introduction of this factor aims at "promoting" bidders that have spent only a small percentage of their respective budgets and "demoting" bidders that have nearly exhausted their budgets, in order to keep more machines (and thus more supplier options) available for a longer time whenever possible. It also takes into account any available storage, by "promoting" machines with storage against others of the same *spend budget*[2] level which do not possess any. The proposed SGSupplyAdwords algorithm (i.e. *Smart Grid Supply Adwords*), which solves the SupplyBudgetUtilization problem with a competitive ratio of $1-\frac{1}{e}$, can be found below.

Definition 5 (Algorithm SGSupplyAdwords) *Assign a new task v to machine u that maximizes the expression*

$$bid_{uv}\left(1-e^{\frac{l_u-s_u}{b_u}-1}\right), \tag{1}$$

where bid_{uv} is the bid of machine u for task v and l_u, s_u are the load and storage of machine u respectively at the time of arrival of task v, and b_u is the budget of machine u.

Note that the modified bid is used only to determine the winning machine but not to "pay" for the task: the winning machine u always subtracts the original bidding amount bid_{uv} for task v from its budget. Naturally, when a machine runs out of available budget it stays deactivated, does not participate in future bidding processes and cannot be assigned any further tasks. In the following theorem, we prove that, under the small bids assumption, the above algorithm is $\left(1-\frac{1}{e}\right)$-competitive.

Theorem 1 *The* SGSupplyAdwords *algorithm solves the* SupplyBudgetUtilization *problem under the small buds assumption and achieves a competitive ratio of $1-\frac{1}{e}$.*

Proof: Let $v \in V$ be an arbitrary node that OPT assigns on node u^*, while SGSupplyAdwords assigns on node u. From the definition of SGSupplyAdwords we know that $\forall v \in V$

$$bid_{u^*v}\left(1-e^{\frac{l_{u^*}-s_{u^*}}{b_{u^*}}-1}\right) \leq bid_{uv}\left(1-e^{\frac{l_u-s_u}{b_u}-1}\right). \tag{2}$$

Since the value of storage s is bounded from below by 0, it is easy to see that $\forall v \in V$

$$bid_{u^*v}\left(1-e^{\frac{l_{u^*}}{b_{u^*}}-1}\right) \leq bid_{u^*v}\left(1-e^{\frac{l_{u^*}-s_{u^*}}{b_{u^*}}-1}\right). \tag{3}$$

Note that we start with zero storage on all nodes, and we need to assign load to (equiv. spend from budgets of) machines in order to generate any storage. Therefore, we can

[2] In the rest of the paper we will use the expression "spend budget" (or simply "spend") to refer to the absolute amount of a node's budget that has been spent, and the expression "spend percentage" to denote the spend budget's relevant value to the total budget of the node.

bound storage s from above by the budget sum $\sum_{u \in U} b_u$ for every node $u \in U$ and write

$$bid_{uv}\left(1 - e^{\frac{l_u - s_u}{b_u} - 1}\right) \leq bid_{uv}\left(1 - e^{\frac{l_u}{b_u} - \frac{\sum_{u \in U} b_u}{b_u} - 1}\right) \Rightarrow$$
$$bid_{uv}\left(1 - e^{\frac{l_u - s_u}{b_u} - 1}\right) \leq bid_{uv}\left(1 - e^{\frac{l_u}{b_u} - 1} A\right), \forall v \in V, \quad (4)$$

where $A = e^{-\frac{\sum_{u \in U} b_u}{b_u}}$. Finally, by combining equations 2, 3 and 4, we get $\forall v \in V$:

$$bid_{u^*v}\left(1 - e^{\frac{l_{u^*}}{b_{u^*}} - 1}\right) \leq bid_{uv}\left(1 - e^{\frac{l_u}{b_u} - 1} A\right). \quad (5)$$

In the remainder of the proof, we are going to utilize a discrete variation of equation 5, by assuming that, at the end of the algorithm's run, each node will belong to one of k discrete levels, depending on how much of their budget was spent, i.e. instead of the continuous spend percentage $\frac{l_u}{b_u}$ of node u we will write the discrete form $\frac{i}{k}$ if node u belongs to the i-th discrete level[3]. Note that we assume a large discretization constant k and due to the small bids assumption we assume every bid to be much smaller than $\frac{1}{k^2}$.

We call nodes of *type* i (for $i \in [1, k]$) all nodes $u \in U$ who have spent between $\frac{i-1}{k} b_u$ and $\frac{i}{k} b_u$ of their budget b_u at the end of the algorithm, and let α_i be the total amount obtained by the optimal allocation from nodes of type i (with $\sum_{i=1}^{k} \alpha_i = \alpha$ being the total spend budget of the OPT algorithm). Let *slab* i be the set of the amount of budget spent by nodes $u \in U$ in the $\left[\frac{i-1}{k} b_u, \frac{i}{k} b_u\right)$ segment of their budget b_u, and let β_i be the amount of this cumulative spend budget. From the definition of α_i and β_i we have:

$$\beta_i = \frac{\alpha - \sum_{j<i} \alpha_j}{k}, \forall i \in [1, k] \quad (6)$$

Let $\hat{i}$ and $\hat{j}$ be the types of nodes u^* and u respectively at the time of node v's arrival. By replacing the expressions $\frac{l_{u^*}}{b_{u^*}}$ and $\frac{l_u}{b_u}$ in equation 5 with the corresponding discrete versions, we get:

$$bid_{u^*v}\left(1 - e^{\frac{\hat{i}}{k} - 1}\right) \leq bid_{uv}\left(1 - e^{\frac{\hat{j}}{k} - 1} A\right), \forall v \in V. \quad (7)$$

Since the expression $\left(1 - e^{\frac{\hat{i}}{k} - 1}\right)$ is monotonically decreasing and for the type i of node u^* at the end of the algorithm it is $\hat{i} < i$, we have

$$bid_{u^*v}\left(1 - e^{\frac{i}{k} - 1}\right) \leq bid_{u^*v}\left(1 - e^{\frac{\hat{i}}{k} - 1}\right), \forall v \in V$$

[3] Obviously the two expressions are equivalent for $k \to \infty$.

and by equation 7 finally

$$bid_{u^*v}\left(1-e^{\frac{i}{k}-1}\right) \leq bid_{uv}\left(1-e^{\frac{\hat{j}}{k}-1}A\right), \forall v \in V. \tag{8}$$

Equation 8 is valid for all $v \in V$, and summing up all equivalent expressions we get

$$\begin{aligned}\sum_{v\in V} bid_{u^*v}\left(1-e^{\frac{i}{k}-1}\right) &\leq \sum_{v\in V} bid_{uv} - A\sum_{v\in V} bid_{uv}\left(e^{\frac{\hat{j}}{k}-1}\right) = \\ &= ALG - A\sum_{v\in V} bid_{uv}\left(e^{\frac{\hat{j}}{k}-1}\right), \end{aligned} \tag{9}$$

where ALG is the total spend of the SGSupplyAdwords algorithm. By grouping elements of the sums of equation 9 according to types i and $\hat{j}$, and using the definitions of α_i and β_i, we get the equivalent expression

$$\sum_{i=1}^{k} \alpha_i \left(1-e^{\frac{i}{k}-1}\right) \leq ALG - A\sum_{i=1}^{k} \beta_i \left(e^{\frac{i}{k}-1}\right). \tag{10}$$

Using equation 6 in order to express β_i in terms of α_i, equation 10 becomes

$$\begin{aligned}\sum_{i=1}^{k} \alpha_i \left(1-e^{\frac{i}{k}-1}\right) &\leq ALG - A\sum_{i=1}^{k} \frac{\alpha - \sum\limits_{j<i}\alpha_j}{k}\left(e^{\frac{i}{k}-1}\right) = \\ &= ALG - A\frac{\alpha}{ek}\sum_{i=1}^{k} e^{\frac{i}{k}} + A\frac{1}{ek}\sum_{i=1}^{k}\sum_{j<i}\alpha_j e^{\frac{i}{k}}. \end{aligned} \tag{11}$$

For the expression $\sum\limits_{i=1}^{k}\sum\limits_{j<i}\alpha_j e^{\frac{i}{k}}$ we have:

$$\sum_{i=1}^{k}\sum_{j<i}\alpha_j e^{\frac{i}{k}} = S_k\sum_{i=1}^{k}\alpha_i - \sum_{i=1}^{k}\alpha_i S_i, \tag{12}$$

where $S_i = \sum\limits_{j=1}^{i} e^{\frac{j}{k}} = \frac{e^{\frac{i+1}{k}} - e^{\frac{1}{k}}}{e^{\frac{1}{k}}-1}$.

By combining equations 11 and 12, together with the fact that $\sum\limits_{i=1}^{k}\alpha_i = \alpha$, we get:

$$\begin{aligned}\alpha - \frac{1}{e}\sum_{i=1}^{k}\alpha_i e^{\frac{i}{k}} &\leq ALG - A\frac{\alpha}{ek}S_k + A\frac{1}{ek}\left(S_k\sum_{i=1}^{k}\alpha_i - \sum_{i=1}^{k}\alpha_i S_i\right) = \\ &= ALG - A\frac{1}{ek}\sum_{i=1}^{k}\alpha_i S_i = \\ &= ALG - A\frac{1}{ek}\frac{e^{\frac{1}{k}}}{e^{\frac{1}{k}}-1}\sum_{i=1}^{k}\alpha_i\left(e^{\frac{i}{k}}-1\right). \end{aligned} \tag{13}$$

By taking the limit of equation 13 for $k \to \infty$ we finally get $\alpha - \frac{1}{e}\alpha \leq ALG$, which completes the proof. □

3.3 Extended modeling and the SGSupplyGreedy algorithm

There are cases where the modeling of renewable energy sources as accumulation of storage on machines cannot capture the intricacies of the underlying problem. For example, quite often there is a marginal cost (associated with renewable energy sources) that cannot be expressed as energy spent to acquire one unit of stored energy but rather in some other form of value, e.g. currency. Such energy sources can be considered as individual suppliers with price and budget information, and they can be modeled using the proposed modeling of section 3.2, with one important difference: since the amount of storage or renewable energy can increase or decrease, depending on e.g. weather conditions, their available budget is necessarily *dynamic*, increasing or decreasing accordingly[4].

The budget dynamicity mentioned above affects the applicability of the small bids assumption in this context. Although valid in cases of large-scale suppliers (i.e. utilities and energy vendors) and relatively small individual load demands, there exist also cases where the small bids assumption no longer applies. This is particularly true when small-scale storage or renewable energy sources (e.g. local photovoltaics) act as suppliers using the proposed modeling of dynamic budgets: assuming their budget starts at zero and gradually increasing, e.g. as more storage accumulates on a battery array, their available capacity can be directly comparable to individual load demands. Therefore, it can no longer be assumed that bids are small when compared to the total budget amounts. Below we present an algorithm that takes this fact into account and uses generic bids.

Definition 6 (Algorithm SGSupplyGreedy) *When the next node $v \in V$ arrives, allocate v to the node u of maximum bid_{uv}.*

Theorem 2 *The* SGSupplyGreedy *algorithm solves the* SupplyBudgetUtilization *problem under generic bids and achieves a competitive ratio of $\frac{1}{2}$*

Proof: We want to bound the loss (compared to the optimal OPT algorithm) in spend budget that our algorithm SGSupplyGreedy achieves. For every $u \in U$, let B_u, S_u be the total budget and the final spend (of the budget) by algorithm SGSupplyGreedy respectively. Let V' be the set of tasks v for which SGSupplyGreedy receives a less than optimal bid. By partitioning the set V' into sets V'_u according to the nodes $u \in U$ that the OPT algorithm would assign each task $v \in V'$ unto, we will bound the sum $\sum_{u \in U} Loss_u$, where

$$Loss_u = \sum_{v \in V'_u} (opt_v - alg_v)$$

and opt_v, alg_v are the values obtained for the assignment of task v by algorithms OPT and SGSupplyGreedy respectively.

[4]Of course we assume that a budget cannot decrease further than the already assigned load to the respective machine at any point. Nevertheless, since tasks are also assigned to machines that correspond to future timeslots, this may happen in practice. In these cases, appropriate mechanisms must be put in place to "complement" the missing amount of energy accordingly.

From the definition of OPT, and since the set $V_u^{'}$ can include all neighbors of node u, we have

$$Loss_u = \sum_{v \in V_u'} (opt_v - alg_v) \leq O_u - \sum_{v \in V_u'} alg_v \tag{14}$$

where O_u is the total spend of the OPT algorithm on node u at the end of the algorithm's run.

Now we look at the assignment of a node $v \in V_u^{'}$ that SGSupplyGreedy "misplaces" to another node instead of node u that OPT assigns it to. Note that the spends of both OPT and SGSupplyGreedy algorithms, as well as the budget of node u have intermediate values, since both algorithms have not finished their run and can spend more, while budgets are dynamic and may change their value until the algorithms finish. Let $\widetilde{S}_{u,v}$, $\widetilde{O}_{u,v}$ and $\widetilde{B}_{u,v}$ be the spend by algorithms SGSupplyGreedy, OPT and the budget of node u, respectively, at the time of the assignment decision of node v. Since algorithm SGSupplyGreedy assigns node $v \in V_u^{'}$ to a node different than u, we know that the spend $\widetilde{S}_{u,v}$ is at least as much as $\widetilde{B}_{u,v} - alg_v$ (otherwise it could be assigned to u)

$$\widetilde{S}_{u,v} \geq \widetilde{B}_{u,v} - alg_v, \forall v \in V_u^{'}$$

and since $\widetilde{O}_{u,v} \leq \widetilde{B}_{u,v}$, we get

$$\widetilde{S}_{u,v} \geq \widetilde{O}_{u,v} - alg_v, \forall v \in V_u^{'}. \tag{15}$$

Defining as δ_v and ε_v the differences $S_u - \widetilde{S}_{u,v}$ and $O_u - \widetilde{O}_{u,v}$ respectively, and applying to equation 15, we get

$$S_u \geq O_u + \delta_v - \varepsilon_v - alg_v, \forall v \in V_u^{'} \tag{16}$$

and finally through equation 14:

$$Loss_u \leq S_u + \varepsilon_v - \delta_v + alg_v - \sum_{v^* \in V_u'} alg_{v^*}, \forall v \in V_u^{'} \tag{17}$$

Let $T_u = \left|V_u^{'}\right|$ be the amount of nodes that OPT assigns on node u while SGSupplyGreedy assigns to different nodes. By summing equation 17 for all $v \in V_u^{'}$ we get:

$$\begin{aligned}
\sum_{v \in V_u'} Loss_u &\leq \sum_{v \in V_u'} S_u + \sum_{v \in V_u'} (\varepsilon_v - \delta_v) - (T_u - 1) \sum_{v \in V_u'} alg_v \Rightarrow \\
T_u \cdot Loss_u &\leq T_u \cdot S_u + \sum_{v \in V_u'} (\varepsilon_v - \delta_v) - (T_u - 1) \sum_{v \in V_u'} alg_v \Rightarrow \\
Loss_u &\leq S_u + \frac{1}{T_u} \sum_{v \in V_u'} (\varepsilon_v - \delta_v - (T_u - 1)\, alg_v).
\end{aligned}$$

We will prove by induction on T_u that $\varepsilon_v - \delta_v - (T_u - 1)\, alg_v \leq 0$ and by extension that $Loss_u \leq S_u$. For the induction step we assume that $\varepsilon_v - \delta_v - (T_u - 1)\, alg_v \leq 0$ and we notice that $\varepsilon_v - \delta_v - T_u alg_v = (\varepsilon_v - \delta_v - (T_u - 1)\, alg_v) - alg_v \leq 0$, which

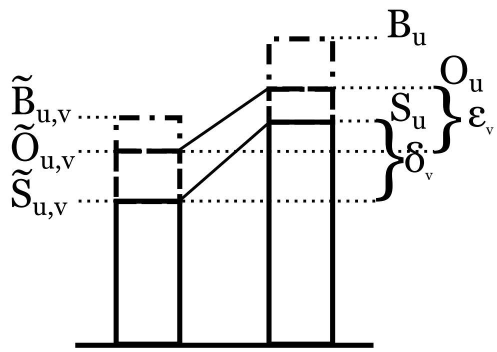

Figure 5: Relationship between budget and spend of OPT and SGSupplyGreedy algorithms of node u at the time of node v's arrival (left), and the corresponding metrics at the end of the algorithms' run (right). The differences $\delta_v = S_u - \widetilde{S}_{u,v}$ and $\varepsilon_v = O_u - \widetilde{O}_{u,v}$ are also shown.

proves trivially the desired result. We now focus on the base case of $T_u = 1$, that is the SGSupplyGreedy algorithm assigns on node u all nodes that OPT assigns there except one, and we will prove that $\varepsilon_v - \delta_v \leq 0$.

Let v be the only node (i.e. $T_u = 1$) from all OPT node assignments to node u that SGSupplyGreedy fails to assign on u (fig. 5). We need to prove that $\varepsilon_v - \delta_v \leq 0$ or equivalently:

$$\begin{aligned}
&\varepsilon_v - \delta_v \leq 0 \Rightarrow \\
&O_u - \widetilde{O}_{u,v} - \left(S_u - \widetilde{S}_{u,v}\right) \leq 0 \Rightarrow \\
&O_u - S_u \leq \widetilde{O}_{u,v} - \widetilde{S}_{u,v}.
\end{aligned} \tag{18}$$

In equation 18, $\widetilde{O}_{u,v} - \widetilde{S}_{u,v}$ expresses the difference between the spend of OPT and SGSupplyGreedy on node u at the time of node v's arrival, and $O_u - S_u$ expresses the same difference at the end of both algorithms' run. Since SGSupplyGreedy assigns to node u at least all nodes other than v that OPT assigns there, the difference between OPT and SGSupplyGreedy cannot have widened more that it was at the time of node v's arrival. In fact, it may have been reduced further, since SGSupplyGreedy has the possibility of assigning additional nodes on node u that OPT does not. Therefore, equation 18 holds, and we have proven that $\varepsilon_v - \delta_v - (T_u - 1)\, alg_v \leq 0$ and by extension

$$Loss_u \leq S_u, \forall u \in U. \tag{19}$$

Using equation 19 it is easy to see that

$$OPT - ALG = \sum_{u \in U} Loss_u \leq \sum_{u \in U} S_u = ALG,$$

where OPT and ALG are the total spends on all nodes achieved by the OPT and SGSupplyGreedy algorithms respectively, leading to competitive ratio $\frac{1}{2}$ for algorithm SGSupplyGreedy.

□

4 Experimental study

While in the previous section we have shown analytically the worst-case guarantees of the algorithms' performance (in the form of competitive ratios), in this section we conduct an experimental study using real world data to look into their performance in practice. Since the Adwords problem is NP-complete in the generic case [4], we cannot compare SGSupplyAdwords and the optimal algorithm in practice, but instead we conduct a comparison with a business-as-usual (BAU) scenario, where demands are being scheduled upon their arrival on the same timeslot (i.e. no flexibility is assumed and no resource allocation technique is being applied). In the rest of this section we describe the experimental setup in detail, followed by the presentation and discussion of the experiments.

4.1 Experiment setup

Source data Regarding consumption, we use detailed data from all household devices (including lighting) of a pilot housing project at the south of Sweden, gathered over a period of approximately 3 months (March to May). The elasticity assumed for each device depends on its type, with some devices being low-elastic (capable of 0 to 2 hours delay - freezer, fridge), some devices being high-elastic (0-4 hours delay - dishwasher, tumble dryer, washing machine, electric car) and many being completely inelastic (lighting, TV/multimedia, general purpose sockets). As a source of storage, we use an array of photovoltaic panels installed at the housing project.

Regarding energy budgets and price information per timeslot, data from the Swedish transmission system operator (Svenska Kraftnät) [17] and the Nordic spot market (Nordpool) [18] have been considered respectively. In both cases, the appropriate data specific for the south of Sweden (zones SN4 and SE4 respectively) for the dates under consideration were used.

Mode of operation For the following experimentations, timeslots of duration of 1 hour have been used along with a timeframe of total 24 hours (24 timeslots), and renewable energy sources are modeled as a form of storage, that is available energy that we can use within each timeslot. Regarding energy carriers, only electricity is relevant for the data considered here and therefore we will use the terms 'timeslots' and 'machines' interchangeably. Note that even though a timeframe of 24 timeslots is considered at any point in time, this is in fact a rolling window of width 24, as the algorithms are running continuously. As a result, at every timeslot one new machine gets created and one is being discarded.

Scenarios We investigate three scenarios:

- BAU: In this business-as-usual (BAU) scenario, no elasticity is assumed and demands are being executed in the order of arrival. This is equivalent to the replay of the data as they were collected originally from the different devices.

<table>
<tr><th>Category name</th><th>Category frequency (% of timeslots)</th><th>Spend budget change in category (range)</th><th>Spend budget totals in BAU (Wh)</th><th colspan="2">AdaptiveFull average budget change</th><th colspan="2">AdaptiveSimple average budget change</th></tr>
<tr><td>High Winners</td><td>18%</td><td>[+100%, +25000%]</td><td>118697</td><td rowspan="2">+220%</td><td rowspan="3">+76%</td><td rowspan="2">+222%</td><td rowspan="3">+77%</td></tr>
<tr><td>Low Winners</td><td>37%</td><td>[0%, +100%)</td><td>235938</td></tr>
<tr><td>Low Losers</td><td>45%</td><td>[-100%, 0%)</td><td>290959</td><td>-57%</td><td>-57%</td></tr>
</table>

Table 1: Results for the FullBudgets set of experiments

- AdaptiveFull: This is a direct implementation of algorithm SGSupplyAdwords.
- AdaptiveSimple: This is a variation of algorithm SGSupplyAdwords, which doesn't include the storage term s_u in equation 1, used here to study the benefits of storage and renewable energy inclusion in the proposed algorithm.

Relation between bids and prices For the experimental part of this work, we define a bid by machine u towards task v as $bid_{uv} = \frac{d_v}{p_u}$, where d_v and p_u are the load demand of task v and price of machine u respectively[5]. Since budgets must be in the same units as the bids, we also normalize the energy budgets of the machines with their respective prices.

Experiments Two different sets of experiments where conducted:

- The FullBudgets set, using full budget and price information: In this set, the full budget and price information available was considered. The aim here is to compare the amount of spend budgets of timeslots, to the corresponding amount in the BAU case. We also look into the unused renewable energy (i.e. storage) that AdaptiveFull and AdaptiveSimple leave in each timeslot, compared to the one left in BAU.
- The SimpleBudgets set, using simplified budget and price information: By keeping budgets and prices the same for all nodes (equal to 30 kWh and 1 respectively), in this set we look into a simplified version of the SupplyBudgetUtilization problem which is equivalent to the peak-reduction problem with a flat pricing scheme. This allows us to focus on the effect of the algorithms on peak demand reduction, as well as look deeper into renewable energy utilization.

4.2 FullBudgets

Results for this set are shown on table 1. We distinguish three categories of timeslots: High Winners (HW), Low Winners (LW) and Low Losers (LL):

[5] We assume machines offer the same price to all tasks and $0 < p_u \leq 1, \forall u \in U$, where U is the set of all machines.

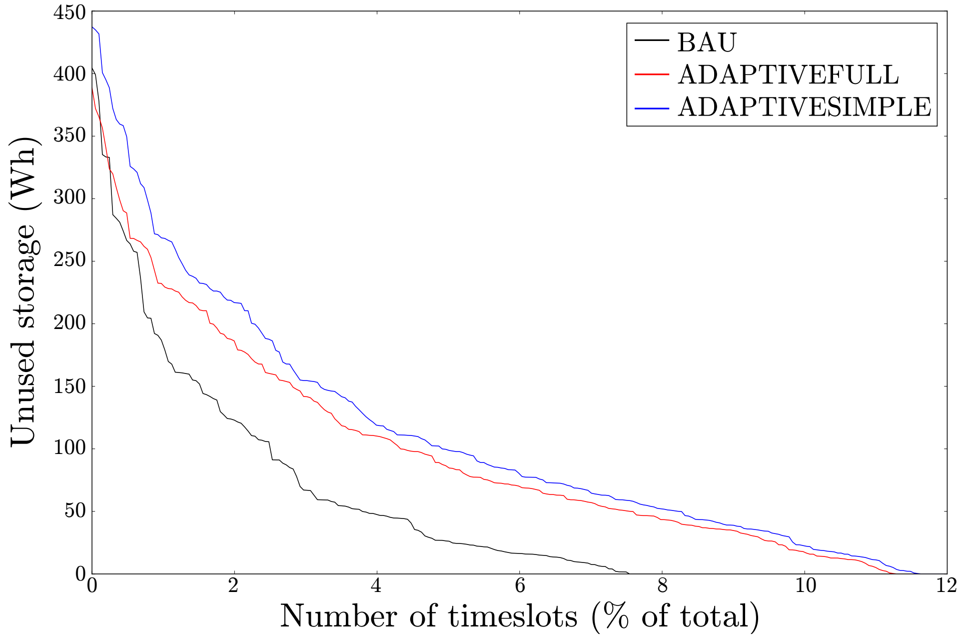

Figure 6: Unused storage per timeslot for the FullBudgets set

- HW are timeslots with initially low spend percentage (of their total budgets), that are experiencing an increase in the range of [100%, 25000%].
- LW are timeslots that achieve modest increase in their spend percentages (range of [0%, 100%]) and cumulatively comprise a modest amount of spend budget (in Wh).
- LL are likewise timeslots that see their spend percentages reduced to a modest degree (range of [-100%,0%]) and define a slightly larger total amount of spend budgets than LW.

The AdaptiveFull algorithm lowers the spend budgets of LL, and raises the HW and LW budgets, in an effort to not let any timeslots spend their whole budget before the others. Note that the small difference between the AdaptiveFull and AdaptiveSimple algorithms (220% and 222% respectively on the HW and LW categories) is due to the inability of AdaptiveSimple to utilize the available storage. Although seemingly counter-intuitive, it is easy to see that since AdaptiveSimple cannot use storage efficiently, it requests energy from the grid to service incoming tasks and ends up increasing spend percentages overall.

On the other hand, the differences between AdaptiveFull and AdaptiveSimple are more clearly seen when looking at the amount of storage they leave unused. In figure 6 we see the unused storage per timeslot, sorted by descending amount for all timeslots.

It is easy to see that AdaptiveFull outperforms AdaptiveSimple by reducing the amount of unused storage particularly on the highest amounts (percentages 0% to approx. 4%). In the same figure 6 the BAU scenario is also given for reference

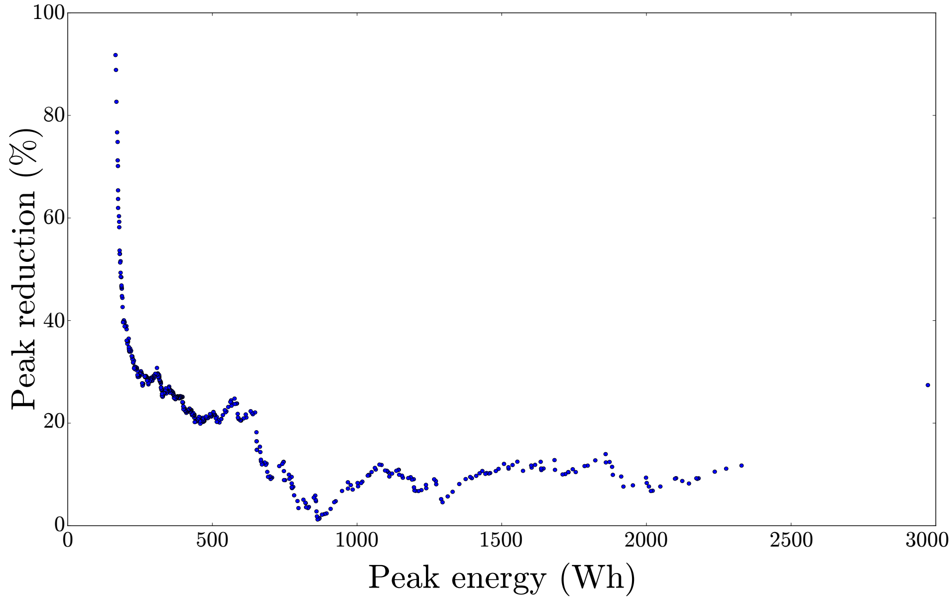

Figure 7: Peak reduction per peak amplitude for the SimpleBudgets set

purposes. Note that BAU manages to perform better than both AdaptiveFull and AdaptiveSimple in the expense of taking budget and price information into account, and without optimizing specifically for storage utilization. In contrast, both AdaptiveFull and AdaptiveSimple try to keep all budgets in a non-exhausted state for as long as possible, sometimes at the expense of readily available storage. Nevertheless, the AdaptiveFull algorithm takes this parameter too into account, leading to an increased storage utilization by approximately 7% on average, compared to the AdaptiveSimple algorithm.

Overall, the Pearson correlation coefficient of the three considered scenarios BAU, AdaptiveSimple and AdaptiveFull (compared to the budget sequence) is respectively 0.0481, 0.062 and 0.0633, showing that the AdaptiveFull algorithm resembles the budget sequence closer (higher Pearson correlation coefficient) than both BAU and AdaptiveSimple.

4.3 SimpleBudgets

As mentioned above, by keeping the parameters of budget and price fixed, we can see how the algorithms behave from the viewpoint of peak demand reduction, as well as storage utilization. The exact peak reduction per peak of varying amplitude can be seen in figure 7 for the AdaptiveFull algorithm (shown alone for reasons of clarity), and in summary form in table 2 for both AdaptiveFull and AdaptiveSimple.

It is easy to see that both AdaptiveFull and AdaptiveSimple perform significantly better than BAU, ranging from 9% to 28% improvement, by managing to lower both peaks of high and low consumption. On the other hand, also in this case the difference between AdaptiveFull and AdaptiveSimple can be seen more clearly

Category name	Category frequency (% of timeslots)	Peak energy range (Wh)	Peak energy totals (Wh)	AdaptiveFull average peak reduction		AdaptiveSimple average peak reduction	
Peaks of high consumption	32%	[650, 2972]	209080	9.18%	22.46%	9.16%	22.54%
Peaks of low consumption	68%	[0, 650)	134820	28.68%		28.74%	

Table 2: Results for the SimpleBudgets set of experiments

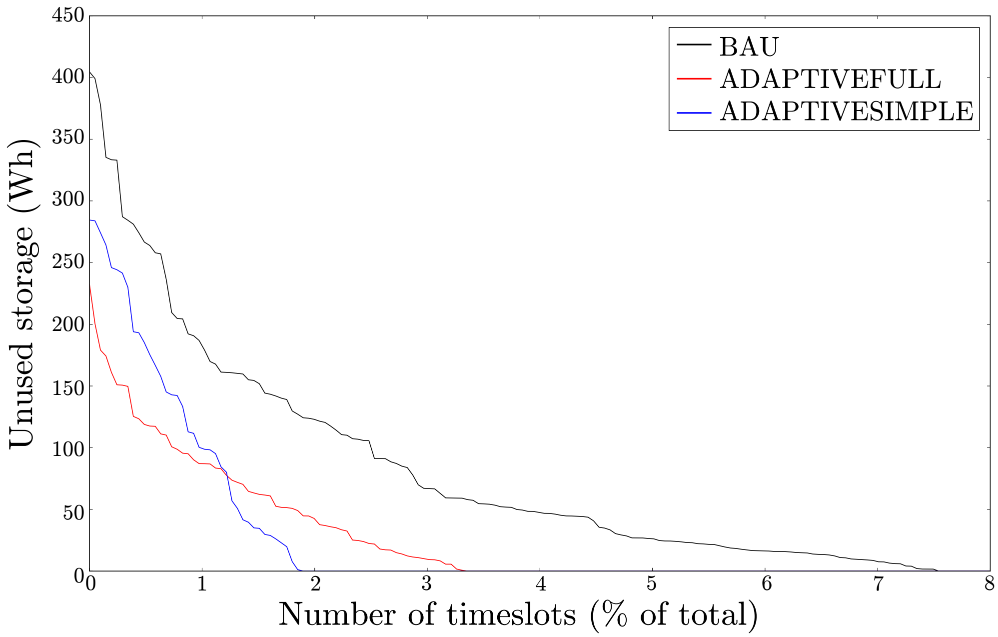

Figure 8: Unused storage per timeslot for the SimpleBudgets set

when looking at the amount of storage they leave unused, and figure 8 shows the unused storage per timeslot, sorted by descending amount for all timeslots. Even though AdaptiveSimple does not prioritize storage utilization, it manages to leave less storage unused compared to the BAU scenario. However, for the same reason, it uses storage blindly, by exhausting storage in timeslots with small offers (percentages 1.2% to 3.3%) but leaving a lot of storage unused in bigger offers (percentages 0% to 1.2%). In contrast, the AdaptiveFull algorithm manages to utilize storage in a smooth way, by ignoring smaller offers when the opportunity to utilize bigger ones arises. Overall, AdaptiveFull utilizes approximately 19% more storage than AdaptiveSimple.

5 Discussion

In the previous section, the AdaptiveFull algorithm (being a direct implementation of algorithm SGSupplyAdwords) is shown to achieve better utilization of available storage compared to the AdaptiveSimple variant, that does not include the storage term s_u in equation 1. This is expected in practice since AdaptiveFull will promote machines with storage, and makes better utilization of storage overall, since machines exhaust their budgets in a slower rate (compared to the AdaptiveSimple case) and are able to service load demands for longer periods. In practice, this means that e.g. utility companies can service a higher amount of consumers using the same amount of supply, leading to higher resource utilization. However, note that theorem 1 applies equally to both variants, giving them the same competitive ratio. This is only natural since competitive ratio is based on the worst case analysis, which is the same for both variants. Note also that budget spending and storage utilization are two distinct benefits of the presented algorithms: even in cases where budget cannot be spent further (i.e. no suitable tasks exist), the AdaptiveFull variant manages to spend less primary (i.e. not stored) energy by utilizing storage more efficiently than AdaptiveSimple.

On the other hand, in the theoretical analysis part of the current work (section 3), we do not take into account price information that might be available on machines. In fact, by assuming non-equal bids bid_{uv} for different tasks v (originating from the same machine u), our algorithms and modeling are as generic as possible in that respect and can take into account i.e. cases where a supplier offers different prices to different customers for the same goods and timeslot, allowing for relevant research on pricing signals (cf [2] and references therein) to connect with the current work.

In addition to the above, a number of benefits, implied in the competitive ratio analysis of theorem 1, are particularly relevant to the domain of Smart Grids considered in this work. As observed by Mehta et al in [16] for a similar algorithm for the Adwords problem, it is possible for machines to have different budgets and also to enter the bidding process at different times. Since machines can also leave at different times, it is possible to continuously run the algorithm in a rolling window of timeslots (as was done in section 4), by allowing new machines to enter and old to leave the bidding process at every timeslot. This fact, together with the dynamic budgets introduced in section 3.3, shows the flexibility and expressiveness of the modeling proposed in section 3, since it can describe many aspects of dynamicity on generation and consumption on Smart Grids.

6 Conclusion

In this paper, we model supply-following demand in Smart Grids in terms of the Adwords problem, in order to provide algorithmic solutions of measurable quality. In doing so, we extend the Adwords problem to incorporate load credit (i.e. storage) and we present and analyze online algorithms that can schedule demand, given availability constraints on supply with guaranteed competitive ratio. In systems where demands are small compared to the individual supply, we prove a $\left(1-\frac{1}{e}\right)$-competitive ratio. For cases where this does not hold, we extend the Adwords problem to utilize dynamic budgets, and present an algorithm with a $\frac{1}{2}$-competitive ratio. Using real data from a pilot housing project, we show the effectiveness of the proposed algorithmic solution both in shaping demand to follow closely the available supply, but also in utilizing available storage and renewable energy options.

Since we approach supply-following demand from a direction orthogonal to research related to pricing mechanisms, our methods can be applied together with any pricing mechanism but also in cases where pricing signals are not available. It would therefore be interesting to look into the combined effects of different pricing mechanisms and the methodology proposed in the current work, in order to study possible synergies between different solutions. On the other hand, since the methodology itself is modular, it can be extended to describe and address additional cases of interest in Smart Grids, such as direct consumer-producer couplings and the inclusion of forecasts.

References

[1] S. D. Ramchurn, P. Vytelingum, A. Rogers, and N. R. Jennings, "Putting the 'smarts' into the smart grid: A grand challenge for artificial intelligence," *Commun. ACM*, vol. 55, no. 4, pp. 86–97, Apr. 2012.

[2] W. Yang, R. Yu, and M. Nambiar, "Quantifying the benefits to consumers for demand response with a statistical elasticity model," *IET Generation, Transmission Distribution*, vol. 8, no. 3, pp. 503–515, Mar. 2014.

[3] R. Katz, D. Culler, S. Sanders, S. Alspaugh, Y. Chen, S. Dawson-Haggerty, P. Dutta, M. He, X. Jiang, L. Keys, A. Krioukov, K. Lutz, J. Ortiz, P. Mohan, E. Reutzel, J. Taneja, J. Hsu, and S. Shankar, "An information-centric energy infrastructure: The berkeley view," *Sustainable Computing: Informatics and Systems*, vol. 1, no. 1, pp. 7–22, 2011.

[4] A. Mehta, "Online matching and ad allocation," *Foundations and Trends in Theoretical Computer Science*, vol. 8, no. 4, pp. 265–368, 2013.

[5] J. K. Kok, C. J. Warmer, and I. G. Kamphuis, "PowerMatcher: Multiagent control in the electricity infrastructure," in *Proceedings of the Fourth International Joint Conference on Autonomous Agents and Multiagent Systems*, ser. AAMAS '05. New York, NY, USA: ACM, 2005, pp. 75–82.

[6] S. Barker, A. Mishra, D. Irwin, P. Shenoy, and J. Albrecht, "SmartCap: Flattening peak electricity demand in smart homes," in *2012 IEEE International Conference on Pervasive Computing and Communications (PerCom)*, Mar. 2012, pp. 67 –75.

[7] L. Lu, J. Tu, C.-K. Chau, M. Chen, and X. Lin, "Online energy generation scheduling for microgrids with intermittent energy sources and co-generation," in *Proceedings of the ACM SIGMETRICS/International Conference on Measurement and Modeling of Computer Systems*, ser. SIGMETRICS '13. New York, NY, USA: ACM, 2013, pp. 53–66.

[8] J. Tu, L. Lu, M. Chen, and R. K. Sitaraman, "Dynamic provisioning in next-generation data centers with on-site power production," in *Proceedings of the Fourth International Conference on Future Energy Systems*, ser. e-Energy '13. New York, NY, USA: ACM, 2013, pp. 137–148.

[9] G. Georgiadis and M. Papatriantafilou, "Dealing with storage without forecasts in smart grids: Problem transformation and online scheduling algorithm," in *Proceedings of the 29th Annual ACM Symposium on Applied Computing*, ser. SAC '14. New York, NY, USA: ACM, 2014, pp. 518–524.

[10] A. Roscoe and G. Ault, "Supporting high penetrations of renewable generation via implementation of real-time electricity pricing and demand response," *IET Renewable Power Generation*, vol. 4, no. 4, pp. 369–382, Jul. 2010.

[11] A.-H. Mohsenian-Rad, V. W. Wong, J. Jatskevich, R. Schober, and A. Leon-Garcia, "Autonomous demand-side management based on game-theoretic energy consumption scheduling for the future smart grid," *Smart Grid, IEEE Transactions on*, vol. 1, no. 3, pp. 320–331, 2010.

[12] S. Caron and G. Kesidis, "Incentive-based energy consumption scheduling algorithms for the smart grid," in *2010 First IEEE International Conference on Smart Grid Communications (SmartGridComm)*, Oct. 2010, pp. 391–396.

[13] T. Wijaya, K. Larson, and K. Aberer, "Matching demand with supply in the smart grid using agent-based multiunit auction," in *2013 5th International Conference on Communication Systems and Networks (COMSNETS)*, Jan. 2013, pp. 1–6.

[14] T. Carpenter, S. Singla, P. Azimzadeh, and S. Keshav, "The impact of electricity pricing schemes on storage adoption in ontario," in *Proceedings of the 3rd International Conference on Future Energy Systems: Where Energy, Computing and Communication Meet*, ser. e-Energy '12. New York, NY, USA: ACM, 2012, pp. 18:1–18:10.

[15] A. Borodin and R. El-Yaniv, *Online Computation and Competitive Analysis*. Cambridge University Press, Apr. 1998.

[16] A. Mehta, A. Saberi, U. Vazirani, and V. Vazirani, "AdWords and generalized online matching," *J. ACM*, vol. 54, no. 5, Oct. 2007.

[17] "Svenska kraftnät, elstatistik per elområde," http://www.svk.se/Drift-och-marknad/Statistik/Elstatistik-per-elomrade/.

[18] "Nord pool spot historical market data," http://www.nordpoolspot.com/historical-market-data/.