Multi-agent Planning Under Uncertainty for Capacity Management

Nijs, Frits de; Weerdt, Mathijs M. de; Spaan, Matthijs T. J.

DOI
10.1007/978-3-030-00057-8_9

Publication date
2019

Document Version
Final published version

Published in
Intelligent Integrated Energy Systems

Citation (APA)
Nijs, F. D., Weerdt, M. M. D., & Spaan, M. T. J. (2019). Multi-agent Planning Under Uncertainty for Capacity Management. In P. Palensky, M. Cvetković, & T. Keviczky (Eds.), *Intelligent Integrated Energy Systems: The PowerWeb Program at TU Delft* (pp. 197-213). Springer. https://doi.org/10.1007/978-3-030-00057-8_9

Chapter 9
Multi-agent Planning Under Uncertainty for Capacity Management

Frits de Nijs, Mathijs M. de Weerdt and Matthijs T. J. Spaan

Abstract Demand response refers to the concept that power consumption should aim to match supply, instead of supply following demand. It is a key technology to enable the successful transition to an electricity system that incorporates more and more intermittent and uncontrollable renewable energy sources. For instance, loads such as heat pumps or charging of electric vehicles are potentially flexible and could be shifted in time to take advantage of renewable generation. Load shifting is most effective, however, when it is performed in a coordinated fashion to avoid merely shifting the peak instead of flattening it. In this chapter, we discuss multi-agent planning algorithms for capacity management to address this issue. Our methods focus in particular on addressing the challenges that result from the need to plan ahead into the future given uncertainty in supply and demand. We demonstrate that by decoupling the interactions of agents with the constraint, the resulting algorithms are able to compute effective demand response policies for hundreds of agents.

9.1 Introduction

The electric power system is the largest man-made system in the world. Its network is designed such that its connected users and devices all can receive the electrical energy they need whenever they demand. However, the energy system is undergoing a transition. Power is not only generated by controllable power plants, but gradually a larger part comes from intermittent and uncontrollable renewable generation from sun and wind. As the power produced needs to be equal to the power consumed at all times and storage of electrical energy is in many places extremely inefficient, this transition to more renewable generation can only be realized by consuming

F. de Nijs (✉) · M. M. de Weerdt · M. T. J. Spaan
Delft University of Technology, Delft, The Netherlands
e-mail: f.denijs@tudelft.nl

M. M. de Weerdt
e-mail: m.m.deweerdt@tudelft.nl

M. T. J. Spaan
e-mail: m.t.j.spaan@tudelft.nl

P. Palensky et al. (eds.), *Intelligent Integrated Energy Systems*,
https://doi.org/10.1007/978-3-030-00057-8_9

the power at the moment it is produced, called *demand response*. Especially new electrical loads, such as heat pumps and electric vehicles, are relatively flexible: they can be shifted in time to match (renewable) generation.

Shifting loads to moments of high renewable generation creates higher correlations of such controllable loads, and this may create new peak loads in the distribution system. At some places therefore costly reinforcements of the network may seem necessary given the goal of designing for peak use. However, in some areas, such an overload of the network will only occur for a few hours, and only for a few times a year. In such cases, it may not be economical to reinforce the network. The alternative is to coordinate some of these loads to prevent the congestion and guarantee that network use stays within the capacity limits of cables and/or converters. An important difficulty here is that on the one hand this requires looking ahead some time into the future to be able to decide which loads to shift to an earlier (or later) time, but on the other hand we do not know exactly how much renewable power is produced, and how much of the network capacity will be used by uncontrollable loads. This chapter discusses scalable algorithmic methods for capacity management that can deal with such uncertainty.

In the next section, we provide a formal model of the computational problem of capacity management. We give sufficient detail such that it could be straightforwardly implemented to be solved by mixed-integer linear solvers. However, this straightforward approach has two shortcomings. First, it scales poorly with the number of controllable loads, so solving this problem takes too long to be of practical use. Second, it does not capture the uncertainty appropriately (e.g. of generation and of the demand from controllable and uncontrollable loads). In the remaining part of this chapter we introduce methods to get around these shortcomings: we use Markov decision processes (MDPs) to include uncertainty explicitly, and describe how to decouple the problem into agents that only interact through the capacity constraints.

9.2 Problem Description

As a running example in this chapter we use the scheduling of heat pumps. The power draw of a heat pump device easily exceeds the entire remaining household demand. Simultaneous use by a large number of heat pumps easily overloads the capacity of the local grid. However, a heat pump has also a significant potential to contribute to both the integration of renewable sources as well as for capacity management by exploiting the available system inertia: for well-insulated buildings, running the heat pump a few hours earlier can obtain the same level of comfort at negligible extra energy loss. The problem we study then is to optimize the temperature trajectory in the buildings by controlling such thermal devices over time, subject to the available capacity of the electricity network.

In principle, given a discrete-time model of the devices in the aggregation, the control problem can be solved using standard centralized optimization techniques. We present a mathematical formulation of such a general optimization framework for

heat pumps here, which is representative of the control problem for any flexible load. Controlling an aggregation of thermal devices subject to a network capacity constraint comes down to choosing an activation schedule per device which ensures the capacities are never exceeded, while simultaneously guaranteeing that every device maintains its desired temperature. This problem can be formulated as a constrained optimization problem, where the temperature goal (the comfort level) is the objective and the capacity is enforced as one of the constraints.

Let $\theta_{i,t}$ be the temperature of a single device i at a specific point in time t, and let $m_{i,t}$ represent the binary (on/off) control decision of the heat-pump. Then, the given temperature transition model f_i specifies how the temperature of the device evolves up to the next decision step $t+1$ to $\theta_{i,t+1}$. In the following part, we assume that all devices can be modeled by the same general transition model f, with device specific parameters a_i. In the following sections we use the thermal model given by Mortensen and Haggerty [1], which is straightforward to optimize due to its linearity, however our algorithms can be applied to more advanced building thermal models such as those described in [2].

The n heat-pump devices should be constrained such that the sum of power draw does not exceed the (remaining) network capacity and power production. To state the multi-device model we use boldface characters to represent vectors of device parameters over all devices, i.e., $\boldsymbol{\theta}_t = \left[\theta_{1,t}\, \theta_{2,t}\, \ldots\, \theta_{n,t}\right]$. Then, using the Hadamard product $\boldsymbol{b} = \boldsymbol{a} \circ \boldsymbol{\theta}_t \implies \forall i : b_i = a_i \times \theta_{i,t}$, we can define a state transition function to compute $\boldsymbol{\theta}_{t+1}$ as

$$\boldsymbol{\theta}_{t+1} = f(\boldsymbol{\theta}_t, \boldsymbol{m}_t, \theta_t^{\text{out}}) = \boldsymbol{a} \circ \boldsymbol{\theta}_t + (1 - \boldsymbol{a}) \circ \left(\theta_t^{\text{out}} + \boldsymbol{m}_t \circ \boldsymbol{\theta}^{\text{pwr}}\right). \tag{9.1}$$

With this function, we can define a planning problem using a given horizon h, the thermal properties of the n thermostatic loads with initial temperatures $\boldsymbol{\theta}_1$, the predicted outdoor temperature θ_t^{out}, and the predicted power constraint L_t. A solution to such a problem is a device activation schedule that never switches on more devices than is allowed while minimizing cost function $c(\boldsymbol{\theta}_t)$. The entire planning problem becomes:

$$\begin{aligned} \underset{[\boldsymbol{m}_1, \boldsymbol{m}_2, \ldots, \boldsymbol{m}_h]}{\text{minimize}} \quad & \sum_{t=1}^{h} c(\boldsymbol{\theta}_t) \\ \text{subject to} \quad & \boldsymbol{\theta}_{t+1} = f(\boldsymbol{\theta}_t, \boldsymbol{m}_t, \theta_t^{\text{out}}) \\ & \sum_{i=1}^{n} m_{i,t} \leq L_t \\ & m_{i,t} \in [0, 1] \qquad\qquad \forall i, t \end{aligned} \tag{9.2}$$

Due to the generality of the model, we can optimize the devices for different objectives expressed through the cost function. Besides typical functions such as the squared error on the deviation from the set-point $c(\boldsymbol{\theta}_t) = \sum_{i=1}^{n} \left(\theta_{i,t} - \theta_{i,t}^{\text{set}}\right)^2$, or the maximum deviation $c(\boldsymbol{\theta}_t) = \max_i \left(\theta_{i,t} - \theta_{i,t}^{\text{set}}\right)$, we might imagine more

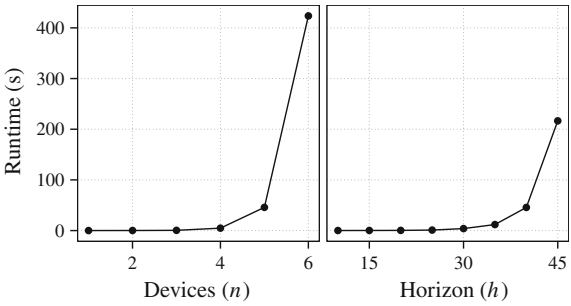

Fig. 9.1 Wall-clock time needed to optimize the mixed-integer linear program (9.2) as a function of the number of devices and the planning horizon

application-specific functions. For example, a refrigerator may only incur high penalties when the temperature gets above a (thawing) threshold.

Unfortunately, this approach suffers from two major drawbacks: limited scalability, and difficulty representing uncertainty. Figure 9.1 demonstrates that solving such a straightforward, centralized model quickly becomes intractable. Furthermore, in the problem description thus far, we ignored that the effect of actions of the heat pump on the (modeled) state of the household is not known exactly, but may vary significantly, because the model does not capture all aspects of reality. For example, the physical building is much more complex, consisting of several different rooms and corridors and windows and walls, radiation from the sun can increase temperatures but is not taken into account, and inhabitants may open and close doors and windows.

In the next section, we demonstrate how to overcome both these weaknesses. First, we show how uncertainty can be incorporated in a principled manner by transforming the proposed model to a multi-agent Markov decision process. Then, we introduce several algorithms to coordinate the agents' demand in a scalable manner, while optimizing their cost functions.

9.3 Multi-agent Planning Under Uncertainty

Here we first introduce an MDP model for the problem of planning the use of heat pumps under uncertainty in the temperature development of the households over time. We show that this can be neatly modeled as a so-called multi-agent MDP (MMDP). In the type of MMDPs that we consider, we identify several agents that take actions more or less independently. In this case, these agents represent the heat pumps of the different houses.

The challenge here is that the size of a straightforward MMDP model increases exponentially with the number of agents, because the set of actions in such a model is the set of *all possible combinations* of actions by individual agents. The domain of study, though, has some specific structure: the only interaction between the agents is because of the limited network capacity. In the remainder of the section we then

discuss a number of different methods to exploit this structure by decoupling the reasoning for each of the agents.

First, we show how an arbitrage mechanism can be used in combination with several iterations of optimal responses by the agents given their likelihood of getting allocated to ensure that these individual policies never cross the resource limit.

Second, we investigate preallocation methods to avoid dependence on an on-line component, allowing policies to be executed without communication. Although scalable methods to compute preallocations only satisfy the constraints in expectation, we demonstrate that constraint violations can be minimized by reducing the available constraint capacity. Further, we show that this can even be extended to settings where these constraints themselves are uncertain. For example, it may be that there is an uncertain amount of uncontrollable use of the network, making the exact amount of remaining capacity uncertain as well.

Finally, we discuss a method for the case where the agents do not necessarily have a good model of the building they are controlling.

9.3.1 Centralized MMDP Model

MDPs form a flexible mathematical framework for optimizing the course of action of an agent that experiences an uncertain response from the environment to its actions [3]. This allows it to cope with uncertain environmental factors such as outdoor temperature but also uncertainty resulting from imperfect models, for instance a lack of information about whether windows are closed or not. As our problem consists of multiple decision makers, we model the problem as a Multi-agent Markov decision process (MMDP) [4]. A key assumption is that agents are fully cooperative, i.e., that they optimize their decision making according to a joint objective function.

In our MMDP model for the optimization problem defined in Eq. 9.2 we have a set of n agents that all have the same actions $\mathcal{A} = \{\text{off}, \text{on}\}$ available to them, and an agent-specific state $\mathcal{S}$. The continuous temperature is discretized into k non-overlapping states s_j each defining a temperature interval $[\theta_{s_j,\min}, \theta_{s_j,\max})$. In addition to this, there are two extrema states $s_{\min}$ and $s_{\max}$ ranging from $(-\infty, \theta_{s_1,\min})$ and $[\theta_{s_k,\max}, \infty)$ respectively, resulting in the following state space of an agent: $\mathcal{S} = \{s_{\min}, s_1, s_2, \ldots, s_k, s_{\max}\}$.

The transition function $T : \mathcal{S}^n \times \mathcal{A}^n \times \mathcal{S}^n \to [0, 1]$ describes for each joint state and joint action pair $(\mathbf{s}, \mathbf{t})$ the probability of attaining joint state $\mathbf{s}'$. It is derived by applying the Markov heat-transfer function $f(\theta, m) = \mathbf{a}\theta + (1 - \mathbf{a})(\theta_{\text{outside}} + m\theta_{\text{heating}})$ to the lower and upper values of the temperature range $[\theta_{s,\min}, \theta_{s,\max})$. This produces a new range $[\theta'_{\min}, \theta'_{\max})$ that may overlap the ranges of multiple discrete states $s_j, s_{j+1}, \ldots$. The degree of overlap determines the (uniform) probability of transitioning to each of these potential future states.

Agents are rewarded for their actions in a certain joint state, through the reward function $R : \mathcal{S}^n \times \mathcal{A}^n \to \mathbb{R}$. The rewards assigned to the agents in each time step are the costs depending on how large the deviation from the setpoint $\theta^{\text{set}}_{i,t}$ is:

$$\sum_{i=1}^{n} - \max\{0, \left|s_i - \theta_{i,t}^{\text{set}}\right| - 0.5\}^2. \tag{9.3}$$

The imposed power constraint L_t which limits the number of activated devices is then encoded in the joint transition function. The joint transition function specifies the cross product of all agents' action spaces $\mathcal{A}^n$. By removing those actions where the number of agents 'on' is more than L_t we obtain the required constraint.

The resulting MMDP can be solved optimally [3], but suffers from scaling exponentially in the number of agents.

9.3.2 Decoupling and Best Response with Arbitrage

A first approach to avoid the exponential blow-up of the centralized model is to model the decision problem for each heat pump as a separate problem, and have a very simple centralized *arbitrage* mechanism to guarantee that never too many devices switch on. The details of this approach can be found in [5]. In this section, we summarize the main conceptual idea behind this method, and discuss the effects on the run time compared to the centralized approaches introduced above.

The problem for each agent representing a heat pump is the MMDP for one heat pump from the previous section, discarding the restriction L_t in the joint transition function. Such so-called *single-agent* MDPs can be solved for each agent separately.

However, to prevent that the agents' plans violate the power limit in a certain time step t, the agent that expects to lose the least utility from switching off is switched off at t, and its plan is re-computed. This arbitrage mechanism is repeated until the conflict is resolved. To determine which agent expects to lose the least utility by going from on to off we look at the difference between the planned utility scores in the value table. However, because this procedure risks getting caught in a local minimum, we use the utility loss as a probability of being selected instead of always selecting the agent that expects to lose the least utility. Moreover, we explicitly model the probability of being forced to switch off by the arbitrage mechanism in the planning model for the single-agent MDPs. This indirectly models the effect of the plans of other agents. We call this approach the arbitrage best-response method (arbitrage-BR).

We use the following artificial instances of the problem to compare the scalability of this approach relative to the optimal solutions. In its simplest form, this instance has 3 agents, a horizon of 20 and $\theta_{i,t}^{\text{set}} = 20, \forall i, t$. In the first 5 time steps, 3 agents are allowed to switch on, in the next 5 time steps only 2 are allowed, followed by 5 time steps where only 1 is allowed. The final 5 time steps are unconstrained. We then evaluate the scalability by varying the number of agents between 1 and 6, and the horizon between 5 and 45, and the number of agents fixed at 3 (10 instances per setting). In addition, we set the MMDP approach to use only 6 temperature states, and we imposed a run-time cut-off of 5 minutes. The arbitrage-BR method scales

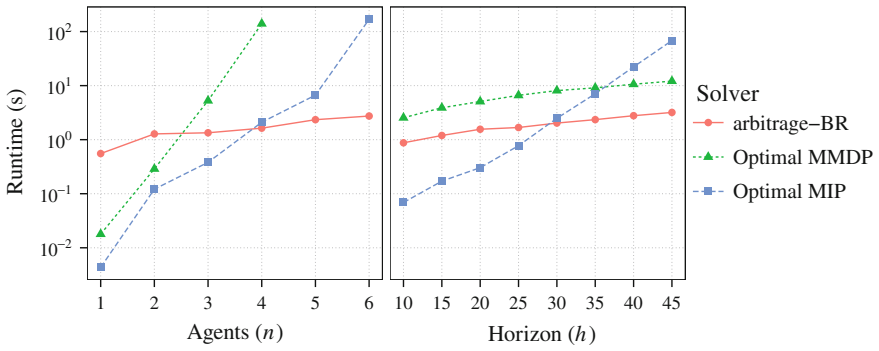

Fig. 9.2 The runtime of both optimal methods does not scale with the number of agents, while the arbitrage-BR method scales reasonably well in both agents and horizon

well in terms of both number of agents and horizon, as shown in the experimental results in Fig. 9.2.

Next, to better understand the potential of arbitrage-BR in practice, we consider a more realistic evaluation in a neighborhood of 182 households equipped with heat pumps. Given runtimes observed earlier, we cannot expect to be able to run the optimal solvers for 182 households. We, therefore, consider a relaxation of the optimal MIP that allows the devices to be switched on only partially. Because in practice these devices cannot be switched on in arbitrary fractions, and because the decision time step granularity of 1 minute is the minimum to prevent short-cycling, it is not possible to implement the outcome of the relaxation. However, this provides a lower bound on the penalty of the optimal solution with binary activations.

For this evaluation, we model a two-day period during which a gradual decrease of the available power occurs starting from hour 6. At hour 20 the minimum capacity is reached and only 10 out of 182 heat pumps are allowed to be switched on. At the start of the second day, all devices can be switched on again. The households would all like to maintain an internal temperature of 21° ($\theta^{\text{set}}_{i,t} = 21, \forall i, t$).

The decision frequency is set to once every minute. While it is unlikely that in a real-world scenario the power limit is known with such accuracy, this granularity allows each unit to switch just in time, and it also serves as a worst-case problem size to demonstrate scalability. Since the agents are now decoupled, we can solve the MMDP with much finer temperature discretization. For arbitrage-BR we discretized temperature from 16 to 24° over 80 states, resulting in bins of 0.1° width.

Figure 9.3 presents the average indoor temperature, the normalized cumulative error and the number of devices switched on for this instance.

First, we observe that all approaches using planning perform significantly better than the non-anticipatory control, which has a rather high cumulative penalty. The cumulative penalty of the arbitrage-BR stays close to the MIP relaxation lower bound, which confirms that it is close to optimal even in this larger instance. Computing the 182 policies for the adaptive decomposition took only 7.5 min, less time than it took

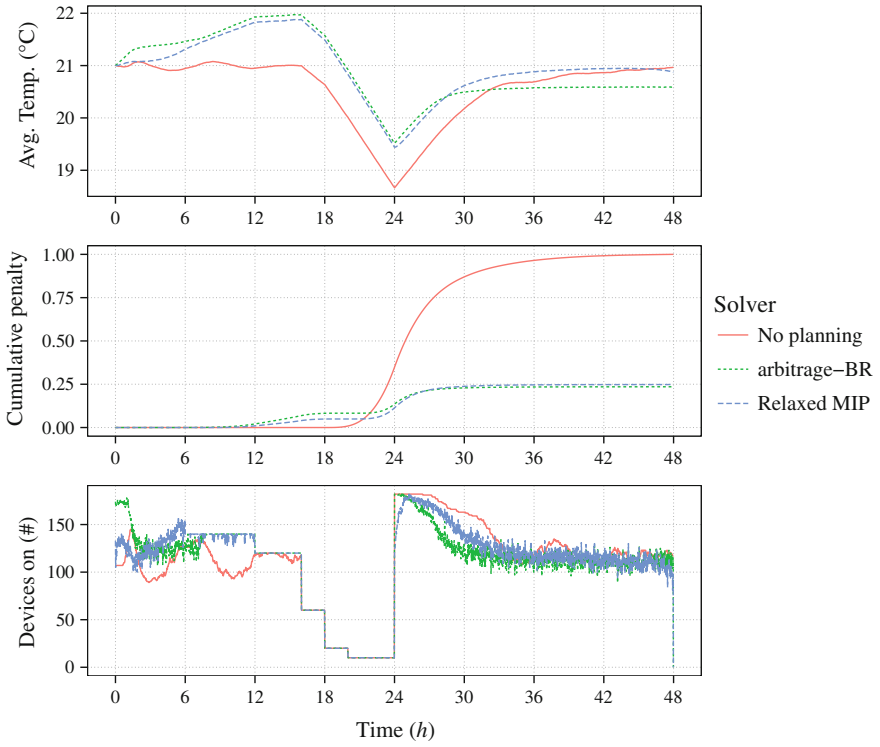

Fig. 9.3 Simulation of the response of a realistic neighborhood of 182 households to a strong curtailment request. Algorithm arbitrage-BR performs on par with the theoretical upper bound given by the relaxed MIP solution

the optimal MMDP solver to compute a solution for the four agent toy example above. This demonstrates that the adaptive decomposition is indeed scalable to real-world instances.

9.3.3 Off-Line Control Through Preallocations

The algorithms discussed in the previous section allow us to safely control an aggregation of heat-pumps through communication with a centralized arbiter. However, because the power grid should be robust to both system failures and malicious attacks, we additionally require that a decentralized fall-back exists. Therefore, in this section, we study algorithms to compute decentralized control policies which satisfy the constraints by adhering to a resource *preallocation*.

A preallocation specifies for each agent in advance at which times it has permission to use resources. Policies computed for a preallocation are communication-

free: because the allocation fully specifies the way the constraint should be shared, an agent never needs to coordinate its consumption with others during execution. Given the imposed power constraints per time step L_t, a preallocation $U_{i,t}$ is computed for each agent i, such that the allocations jointly satisfy

$$\forall t: \sum_{i=1}^{n} U_{i,t} \leq L_t. \tag{9.4}$$

Then, each agent can individually optimize a policy π_i satisfying its own allocation, meaning that the consumption of the policy $C_{\pi_i,t}$ never exceeds the preallocation,

$$\forall i: \max_{\pi_i} V_{\pi_i}, \text{ subject to } \forall t: C_{\pi_i,t} \leq U_{i,t}. \tag{9.5}$$

Existing algorithms to compute preallocations for MDPs can be categorized according to the type of preallocation they compute: (i) A MILP [6] and LDD + GAPS [7] compute preallocations which restrict the *worst-case* resource consumption. (ii) The Constrained MDP LP (CMDP; [8]) and Column Generation [9] compute preallocations which restrict the *expected* resource consumption. Unfortunately, both categories have drawbacks which limit their use in practice. The algorithms which restrict worst-case consumption have exponential worst-case complexity in the number of resources, which makes them intractable for our models. In addition, they may lead to low efficiency: resources may sit unused when the uncertain state trajectory leads agents to a state where resources are not needed (e.g. sufficiently warm in the case of heat-pumps). Restricting the expected consumption is tractable, however, the resulting policies are stochastic and may *violate* the constraints at execution time.

Because the tractability of restricting the expected consumption makes it more promising, we investigate how to limit the risk of policies jointly violating the constraints in the next Sect. 9.3.3.1. Then, because renewable power sources may introduce uncertainty about the constraint itself, an extension to compute preallocations for stochastic constraints is proposed in the following Sect. 9.3.3.2.

9.3.3.1 Bounding Constraint Violation Risk

Stochastic preallocation algorithms like CMDPs and Column Generation compute stochastic policies that only ensure that their *expected* resource consumption does not violate the limits. As such, these methods do not provide any guarantees regarding the probability that a resource limit is violated during execution. In this section we introduce methods which improve these algorithms by ensuring that the probability of violating any individual constraint is upper bounded by a given parameter α. In doing so, we summarize our work on bounding the probability of constraint violations, for details see [10].

Because of the uncertainty present in the transition function of the temperature, the temperature state $s_{i,t}$ of agent i in a future time step t is a random variable. Given

a control policy π_i which switches the heat-pump on below a certain temperature, the future power consumption of an agent also becomes a random variable $C_{i,\pi_i,t}$. When each agent executes their policy unconditionally, and without communication, these random variables are independent. Therefore, by independence, we can compute the total resource consumption at time t as the sum of the agents' consumption, $C_t = \sum_{i=1}^{n} C_{i,\pi_i,t}$. The stochastic allocation algorithms guarantee that

$$\forall t : \mathbb{E}\big[C_t\big] \leq L_t. \tag{9.6}$$

In practical applications, even when constraints are soft, exceeding them typically incurs some cost to the system operator, such increased wear from overheating when exceeding the capacity of a power grid element. Therefore, even when we are using stochastic allocation algorithms, we would additionally like to restrict the probability that a realization of C_t exceeds the limit, or

$$\forall t : \mathbb{P}\big[C_t > L_t\big] \leq \alpha. \tag{9.7}$$

To obtain policies which additionally satisfy the constraint on the tail probability, we propose to impose reduced resource constraints $0 \leq L_t^* \leq L_t$, resulting in more conservative policies. Because the random variables $C_{i,\pi_i,t}$ can be upper bounded by the power consumption of the most-consuming action, we can apply Hoeffding's inequality [11] to determine L_t^*, resulting in

$$L_t^* = L_t - \sqrt{\frac{\ln(\alpha) \cdot \left(\sum_{i=1}^{n} (\max C_{i,t})^2\right)}{-2}}. \tag{9.8}$$

In practice the bound obtained by applying Hoeffding's inequality can be relatively loose (Fig. 9.4, top). Therefore, we also propose a dynamic constraint relaxation technique which adjusts the reduced resource limit L_t^* on the basis of empirical evidence of actual violations during simulation (Fig. 9.4, bottom).

To evaluate the proposed approaches to bound the risk of constraint violations, we compare them to the heat-pump planning problem. Each agent has its possible temperature states discretized over 24 states, and we plan for a time horizon of 24 steps. Figure 9.5 presents the performance of the algorithms as the number of agents grows. We observe that the preallocation algorithms constraining worst-case performance (MILP and LDD + GAPS) indeed exhibit poor scalability, as they exceed 60 minutes of computation time at 4 agents. At the same time, we observe that while the CMDP algorithm is highly scalable, it computes solutions which exceed the available capacity nearly half the time on tight constraints. The results show that this high risk is averted when we virtually reduce the resource capacity available to the planner through application of Hoeffding's inequality. However, the resulting policies are on the conservative side of the risk threshold, staying an order of magnitude below the target tolerance of $\alpha = 0.05$. Our dynamic constraint relaxation algorithm is able to target the lower, observed tolerance of $\alpha = 0.005$ exactly. While

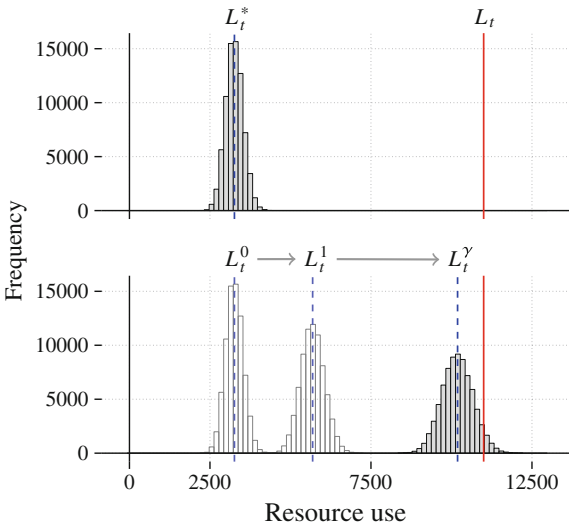

Fig. 9.4 Histograms showing realized resource demands obtained through simulation. Policies to satisfy the constraint (*solid lines*) are computed for reduced limits (*dashed lines*). **Top**: reduced resource limits on the basis of Hoeffding's inequality. **Bottom**: initial and final iterations of our dynamic bound relaxation

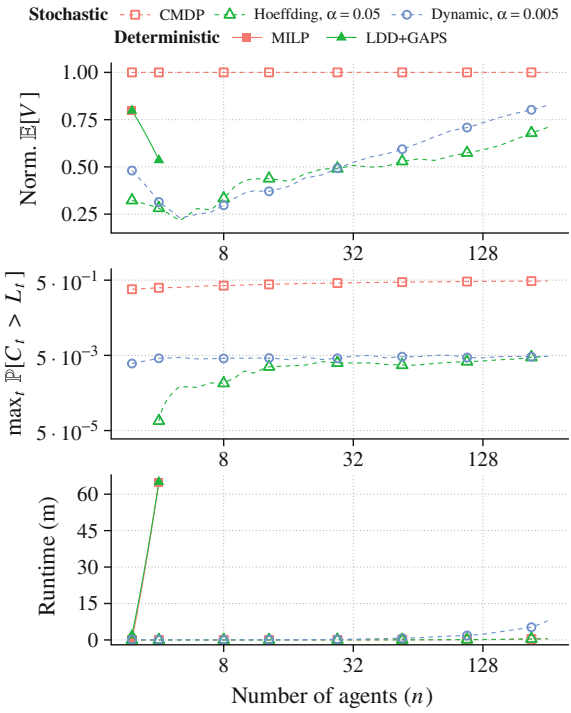

Fig. 9.5 Comparison of the performance of preallocation algorithms on heat-pump planning problems as the number of agents increases, measured on three performance metrics. **Top**: expected value of the solution normalized to CMDP (log x). **Middle**: simulated constraint violation probability of the most-often violated constraint (log-log). **Bottom**: mean wall-clock time required to compute a policy, in minutes (log x)

the dynamic algorithm takes slightly longer to compute, it nevertheless remains tractable, especially when compared to the deterministic allocation algorithms. In addition, the solutions it finds are of higher quality than the Hoeffding-bounded policies for larger numbers of agents, even though the risk level it attains is the same. In conclusion, when comparing stochastic preallocations we observe that for large numbers of agents, the values of the bounded approaches tend towards the CMDP value. When more agents are available to spread the load of the reduced resource limit, their individual rewards are compromised less.

9.3.3.2 Computing Preallocations for Stochastic Constraints

Thus far we have assumed that the planner knows exactly how much power is available in each time step. One major challenge of the integration of renewable energy sources such as wind and solar power generators is that it makes the available power production capacity dependent on the weather, and therefore volatile. Unfortunately, it is not yet possible to predict weather perfectly even on short (day-ahead) timescales. Therefore controllers of such buffers should take into account multiple statistical forecast scenarios [12, 13]. In order to address this requirement, this section investigates how this assumption can be relaxed to deal with stochastic resource constraints when communication is unreliable; for more details, see our work in [14].

A collection of potential power production scenarios can be represented by a Markov chain defined over outcomes. This Markov chain is defined by a state space S_L of power production outcomes, and the transition probabilities $T_L : S_L \times S_L \rightarrow [0, 1]$. Since all agents must adhere to the same constraint, the transition function of the stochastic constraint threatens to couple the agents together. Fortunately, Becker et al. [15] show that independence is retained when shared features only exist in a part of the state space that agents cannot affect themselves. As such, the stochastic constraint problem can also be *decomposed* into n single-agent sub-problems, which we propose to do by augmenting the state space of each agent with the current limit (captured in factored state space $S_L \times S$). Nevertheless, the preallocation algorithms must be modified to handle the fact that agents expected consumption is now correlated with the probability of visiting a power production state s_L.

Alternatively, we could collapse the constraint Markov chain to its expectation in each time step, to obtain a planning problem with a fixed constraint, which the preallocation algorithms can solve directly. However, doing so would result in policies which make two-sided errors: if the realized constraint is less than the expectation, the policy is likely to cause a violation, while if the realized constraint is more than the expectation, the policy will leave resources unused. In addition, knowledge of the current constraint-state may inform which future scenarios are more likely, allowing the planner to anticipate on future constraint realizations. Therefore, we expect that taking into account the model of stochasticity in the preallocation will result in policies which are both significantly safer and which obtain significantly better expected value.

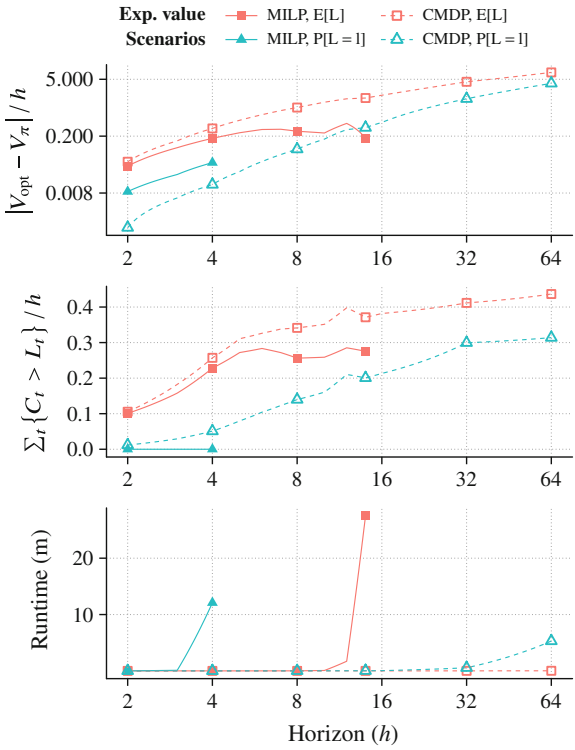

Fig. 9.6 Comparison of two approaches to handle probabilistic forecasts of power production: using the expected value (*squares*) versus using all scenarios in the planning problem (*triangles*). Results for increasing planning horizon on three performance metrics (lower values are better). **Top**: deviation from the optimal expected value, normalized to horizon length (log-log). **Middle**: simulated number of constraint violations, normalized to horizon length (log x). **Bottom**: mean wall-clock time required to compute a policy, in minutes (log x)

To test our expectations, we perform an experiment comparing the modified preallocation algorithms on an instance of the heat-pump planning problem with 10 artificially generated power production scenarios. For these experiments, we discretized the temperature state range into 25 states, and we restrict the number of agents to 3 in order to be able to compute the optimal (on-line) joint policy. This allows us to compare the solution quality of the preallocation algorithms objectively. Figure 9.6 presents the results of an experiment evaluating plan quality as the length of the planning horizon increases. We observe that planning for the stochastic constraint scenarios is more computationally intensive, resulting in increased plan runtime. However, in return, we observe that both our expectations on plan quality hold in practice: planning for a stochastic constraint results in a smaller error *and* fewer constraint violations for both types of preallocation algorithms.

Another conclusion we can draw from this experiment is that the quality of control degrades when agents must operate without communication for long periods of time. While off-line control may be required to satisfy grid robustness requirements, under normal operation we expect agents to be able to operate with regular communication intervals. To determine if there are also benefits to incorporating stochastic constraints in such a rolling horizon re-planning setting, we perform an additional experiment where we let the agents communicate their current state at set intervals. Figure 9.7 presents the results, showing the relationship between the average number

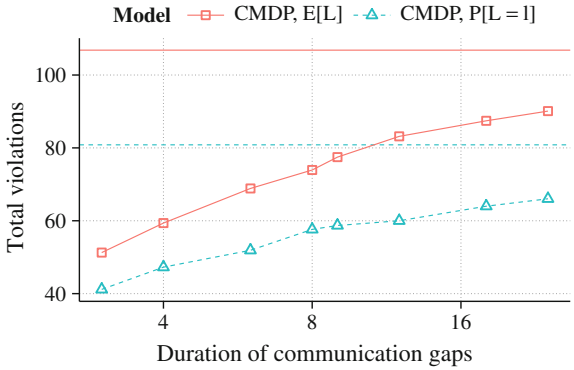

Fig. 9.7 Effect of intermittent communication on the quality of control when: using the expected value of the constraint (*squares*) versus using all constraint scenarios in the planning problem (*triangles*). Plot shows the total number of violations relative to the no-communication upper bound (*horizontal lines*), as the time between successive re-planning increases (log x)

of violations over the horizon (of $h = 216$) and the re-planning frequency, with a gap of three indicating that agents communicate and re-plan every fourth time-step. We observe that re-planning more frequently leads to fewer constraint violations, although this comes at the cost of needing sufficient computational capacity to compute a plan before the next decision point. Further, we see that in this re-planning setting it also makes sense to use the scenario information in the planning problem, as this also significantly reduces the total number of violations. Note that, although we did not do so here for comparison, we can combine stochastic constraints with the bounding technique presented in Sect. 9.3.3.1 to further reduce the risk of constraint violations.

9.3.4 Learning Agent Types

Up to now, we have assumed that all model parameters are fully known to the planning algorithm, in which case a solution can be computed offline (i.e., before execution of the policies). In practice, however, that might not be the case, which requires the planning algorithm to take into account information it can gather online (i.e., while executing the policies). In particular, we consider a setting in which model parameters such as grid constraints are known, but certain characteristics of the individual agents are not. By observing their behavior, however, we can estimate their model parameters. The key challenge is how to optimize the heating of houses given uncertain estimates of their parameters while capacity limits are not to be exceeded.

We identify different *types* of agents, where each type identifies certain key parameters of the system the agent is controlling. Those can be physical characteristics,

such as the insulation level of a house, but can also be related to user preferences, such as the desired temperature setpoint. While types in principle can be dynamic, our focus has been on static types [16]. For instance, the physical properties of a building are unlikely to change quickly, although we likely do not know the thermal response of every building initially; to address this challenge requires the use of a learning agent [17]. However, in this setting, we only need to perform the learning once, as part of the initialization of the device.

To deal with the initially unknown type of each agent, we proposed two novel algorithms [16]: the first algorithm is an extension of Posterior Sampling Reinforcement Learning [18] to the multi-agent, constrained setting. The second algorithm exploits the structural properties of the problem to approximately solve the constrained partially observable problem itself, by bounding the belief space expansion to states where the regret of switching to the best type's MDP policy is low. In particular, we showed how both algorithms can be used as subroutine in the Column Generation stochastic preallocation algorithm described above.

9.4 Conclusions

This chapter discusses how to keep our energy system affordable by making demand responsive to grid limitations, shifting some of the demand to less congested times. In order to perform effective demand response, the controller is required to optimize over future control decisions: in order to decide if we can shift charging an electric vehicle to a later time, we need to know when the car must be charged, and how many more charging opportunities will come. Unfortunately, optimizing a control policy over the future necessarily involves dealing with uncertainty, both in the needs of the device under control (e.g. when the owner of the car returns), as well as in the evolution of the system (e.g. the demand of uncontrolled loads, and the production from renewable sources). We show in Sect. 9.3.1 that optimizing control under uncertainty can be naturally modeled as a Markov decision process. Unfortunately, the resulting constrained, multi-agent Markov decision process model of demand response suffers from intractable scalability in the number of devices. In this chapter we present several novel algorithms to overcome this intractability, as well as innovations that make existing algorithms more effective.

In the first place, we show that the scalability challenge can be overcome if we are able to decouple the control problem of the individual devices from the constraint allocation problem. Section 9.3.2 investigates the use of a centralized resource arbiter to distribute available capacity *on-line*, on the basis of the utility each device expects to receive from its requested allocation. This decouples the agents, as each agent can determine an individual best-response to the probability that the arbiter will award its request. The resulting algorithm is shown to efficiently find solutions with a minimal loss in solution quality. In addition, the arbiter guarantees that the solution satisfies the current system constraints at all times.

Unfortunately, the use of an on-line arbiter requires devices to maintain a connection to the centralized mechanism at all times. In order to compute solutions which are robust to both connection failures and malicious attacks, we investigate resource preallocation algorithms in Sect. 9.3.3. Existing tractable algorithms compute preallocations which satisfy the constraint in expectation, which results in a high risk of constraint violations. While constraint violations can sometimes be absorbed by the inertia of the system, their occurrence should nevertheless be avoided. In Sect. 9.3.3.1 we present an effective approach to bound the probability of constraint violations while retaining the tractability of the preallocation algorithms. Stochastic constraints (as resulting from wind prediction scenarios) are an additional challenge for preallocation algorithms because the agents cannot coordinate on the realized constraint on-line. Nevertheless, we show in Sect. 9.3.3.2 that preallocation algorithms can be modified to incorporate stochastic constraints, resulting in solutions which are both of higher quality and resulting in fewer constraint violations. This result can be combined with occasional re-planning to further improve the coordination.

Finally, we show in Sect. 9.3.4 that our results can be extended to a learning setting, where the devices operate according to one of a set of potential models describing behavior types (for example, insulation levels and preferred set-points). By making use of an optimal learning framework, we are able to identify the correct device model in a minimal number of learning steps.

These results show that our proposed algorithms and extensions are effective at computing high-quality demand-response policies. Nevertheless, there are challenges left to address in future work. Importantly, shifting demand may come at costs for the users. In the current model we assume these costs are known and the algorithms aim to minimize the total costs. However, in many situations, besides total costs, also fairness is an important criterion to take into account. For example, we may not want to have always the same (well-insulated) house be pre-heated throughout the night in order to minimize losses, because the owners will in the end have higher energy costs than without coordination. Related is the issue that the true rewards may not be known to the system, but that the system is informed, e.g. about the desired temperature. In the current proposal, there is no remedy against users who feel that they are left out in the cold and just increase their desired set-point significantly above the real goal in order to increase home temperature. This will probably increase their allocation of the scarce capacity, but at the cost of other users.

Nevertheless, the positive results in simulation make it worthwhile to run a pilot in practice to assess the algorithms performance in resolving bottlenecks in real-world infrastructure. In addition, the methods introduced in this chapter are not just made for heat pumps. They can easily include any other demand that can be modeled as a Markov decision process. In fact, the proposed methods are even more general: we have also applied them to a budget allocation problem in on-line advertising, and in computing capacity-aware recommendations for guiding tourists through crowded cities [16].

Acknowledgements Support of this research by network company Alliander is gratefully acknowledged.

References

1. R.E. Mortensen, K.P. Haggerty, A stochastic computer model for heating and cooling loads. IEEE Trans. Power Syst. **3**(3), 1213–1219 (1988)
2. T.A. Reddy, J.F. Kreider, P.S. Curtiss, A. Rabl, *Heating and Cooling of Buildings*, 3rd edn. (CRC Press, USA, 2017)
3. M.L. Puterman, *Markov Decision Processes-Discrete Stochastic Dynamic Programming* (Wiley, New York, 1994)
4. C. Boutilier, Planning, learning and coordination in multiagent decision processes, in *TARK* (1996), pp. 195–210
5. F. de Nijs, M.T.J. Spaan, M.M. de Weerdt, Best-response planning of thermostatically controlled loads under power constraints, in *Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence* (2015), pp. 615–621
6. J. Wu, E.H. Durfee, Resource-driven mission-phasing techniques for constrained agents in stochastic environments. J. Artif. Intell. Res. **38**, 415–473 (2010)
7. P. Agrawal, P. Varakantham, W. Yeoh, Scalable greedy algorithms for task/resource constrained multi-agent stochastic planning, in *Proceedings of the 25th International Joint Conference on Artificial Intelligence* (2016), pp. 10–16
8. E. Altman, *Constrained Markov Decision Processes*. Stochastic Modeling (Chapman & Hall/CRC, Boca Raton, 1999)
9. K.A. Yost, A.R. Washburn, The LP/POMDP marriage: optimization with imperfect information. Nav. Res. Logist. **47**(8), 607–619 (2000)
10. F. de Nijs, E. Walraven, M.M. de Weerdt, M.T.J. Spaan, Bounding the probability of resource constraint violations in multi-agent MDPs, in *Proceedings of the 31st AAAI Conference on Artificial Intelligence* (2017), pp. 3562–3568
11. W. Hoeffding, Probability inequalities for sums of bounded random variables. J. Am. Stat. Assoc. **58**(301), 13–30 (1963)
12. P. Pinson, H. Madsen, H.A. Nielsen, G. Papaefthymiou, B. Klöckl, From probabilistic forecasts to statistical scenarios of short-term wind power production. Wind Energy **12**(1), 51–62 (2009). https://doi.org/10.1002/we.284
13. A. Staid, J.P. Watson, R.J.B. Wets, D.L. Woodruff, Generating short-term probabilistic wind power scenarios via nonparametric forecast error density estimators. Wind Energy **20**(12), 1911–1925 (2017). https://doi.org/10.1002/we.2129
14. F. de Nijs, M.T.J. Spaan, M.M. de Weerdt, Preallocation and planning under stochastic resource constraints, in *Proceedings of the 32nd AAAI Conference on Artificial Intelligence* (2018), pp. 4662–4669
15. R. Becker, S. Zilberstein, V. Lesser, C.V. Goldman, Solving transition independent decentralized Markov decision processes. J. Artif. Intell. Res. **22**, 423–455 (2004). https://doi.org/10.1145/860575.860583
16. F. de Nijs, G. Theocharous, N. Vlassis, M.M. de Weerdt, M.T.J. Spaan, Capacity-aware sequential recommendations, in *Proceedings of International Conference on Autonomous Agents and Multi Agent Systems* (2018), pp. 416–424
17. F. Ruelens, S. Iacovella, B. Claessens, R. Belmans, Learning agent for a heat-pump thermostat with a set-back strategy using model-free reinforcement learning. Energies **8**(8), 8300–8318 (2015). https://doi.org/10.3390/en8088300
18. M.J.A. Strens, A bayesian framework for reinforcement learning, in *Proceedings of the 17th International Conference on Machine Learning* (2000), pp. 943–950