A trust- and cooperation-based solution of a dynamic resource allocation problem

Gerrit Anders, Jan-Philipp Steghöfer, Florian Siefert, Wolfgang Reif

Angaben zur Veröffentlichung / Publication details:

Anders, Gerrit, Jan-Philipp Steghöfer, Florian Siefert, and Wolfgang Reif. 2013. "A trust- and cooperation-based solution of a dynamic resource allocation problem." In *2013 IEEE 7th International Conference on Self-Adaptive and Self-Organizing Systems, 9-13 September 2013, Philadelphia, PA, USA*, edited by Peppo Valetto, Ozalp Babaoglu, Tom Holvoet, Jeremy Pitt, and Ichiro Sato, 1–10. Piscataway, NJ: IEEE. https://doi.org/10.1109/saso.2013.33.

A Trust- and Cooperation-Based Solution of a Dynamic Resource Allocation Problem

Gerrit Anders, Jan-Philipp Steghöfer, Florian Siefert, and Wolfgang Reif
Institute for Software & Systems Engineering, Augsburg University, Germany
E-Mail: {anders, steghoefer, siefert, reif}@informatik.uni-augsburg.de

***Abstract*—Resource allocation is a common problem in technical systems. For instance, the main task in power management systems is to maintain the balance between energy production and consumption at all times. If such a resource allocation problem has to be solved in a multi-agent system in a decentralized or regionalized manner, agents have to rely on cooperation due to their limited resources and knowledge. In open systems, various uncertainties – introduced by the environment as well as the agents' possibly self-interested or even malicious behavior – have to be taken into account to be able to allocate the resources according to the actual demand. Trust has been proposed as a concept to measure and deal with such uncertainties.**

In this paper, we present a trust- and cooperation-based algorithm that solves a dynamic resource allocation problem in open multi-agent systems. Throughout the paper, the problem of creating power plant schedules in decentralized autonomous power management systems serves as a running example to illustrate our algorithm and results.

***Keywords*—Resource Allocation Problems; Trust; Uncertainty; Open Multi-Agent Systems; Distributed Problem Solving; Electronic Markets; Smart Grids**

I. Resource Allocation in Open Systems

In many technical systems, such as power management systems, the basic task is to solve a *resource allocation problem* in which a number of system components have to provide a certain amount of resources in order to satisfy a specific demand. This is often accomplished in a centralized and even optimal way [1]. However, centralized approaches require detailed knowledge about the behavioral model of all system components and are intractable in large systems because the problem is NP-hard in the number of participants. When giving the system components agency, i.e., the ability to act autonomously in their environment, regionalized or decentralized approaches based on local decisions and coordination can be applied. These approaches are inherently suited for solving problems on the basis of imperfect information about the behavioral model and knowledge of other agents as well as the environment. Since agents only have limited knowledge and capabilities, they have to rely on cooperation to achieve their own and the system's goals.

The willingness to cooperate, however, can not be taken for granted, in particular in open multi-agent systems (MAS) [2]. Such systems consist of a large number of heterogeneous agents that are embedded in a dynamic and potentially hostile environment. Since the agents' behavior and objectives are possibly unknown and not under the control of the system designer or other agents, the benevolence assumption has to be abandoned. As this introduces uncertainty into the decision making process of each individual agent, without further measures, the agents might not be able to fulfill their own or the system's goals.

It is therefore crucial that the agents are able to identify and deal with these uncertainties at runtime. This is especially important when regarding mission-critical systems, such as power management systems. In such cases, the systems' resilience and dependability hinge on the ability to deal with uncertainties introduced by other agents and the environment. *Trust*, more specifically *credibility*, has been proposed as a way to measure and deal with these kinds of uncertainties [3]. Regarding decision processes that use uncertain information, trust allows the agents to quantify the credibility of information sources and to use this data to form expectations of behavior. In incentive-compatible or trust-based mechanism design, trustworthy behavior is incentivized, e.g., by providing compensation based on the trust others put in a participant [4], thereby promoting cooperative and altruistic behavior.

Dynamic resource allocation problems (DRAPs) that have to be solved in open MAS are a domain in which trust data are helpful. The DRAP we address in this paper is dynamic in the sense that it is solved online in a running system and the circumstances under which the optimization takes place change. Some of the information used to determine the future demand for resources stems from stochastic sources. This uncertainty has to be incorporated into the decision processes.

In this paper, we introduce a **t**rust- and **c**ooperation-based **a**lgorithm for **o**pen MA**S**, called *TruCAOS*, that allows to solve the DRAP introduced in Section II. The algorithm is based on the principle of electronic markets and thus solves the DRAP by cooperation between agents instead of centralized instances. To satisfy a fraction of the expected demand, agents can make proposals that contain predicted contributions. Since we regard open MAS, the agents' predicted contribution of resources can (un)intentionally deviate from their actual contribution. An intermediary whose goal is to assign the available resources according to the expected demand decides which proposals to accept or reject. The trust-based principle allows the intermediary to make informed decisions by taking the above-mentioned uncertainties into account, and further serves as a means to sanction misbehavior and incentivize cooperation. Because of the DRAP's dynamic nature, *TruCAOS* does not have to provide optimal solutions but solutions that are "good enough" and robust insofar as – despite the uncertainties – the system satisfies the demand as accurately as possible.

We present our algorithm, *TruCAOS*, in Section IV. In Section V, we show evaluation results and demonstrate that TruCAOS' trust-based principle significantly improves the intermediary's ability to solve the DRAP. Related work is discussed in Section VI before we conclude the paper and give an outlook on future work in Section VII. Throughout the paper, the problem of creating power plant schedules in a decentralized autonomous power management system serves as a running example. This case study and the centralized approaches traditionally used in power management systems are introduced in Section III.

II. A Model of the Dynamic Resource Allocation Problem

We regard a DRAP that has to be solved by a set of possibly self-interested agents $\mathcal{A} = \{a_1, \ldots, a_n\}$, each with limited resources (see Section II-D). The system's goal is to find an allocation so that, in each time step t, the agents provide a *total* ***actual*** *contribution of resources* $A_{\mathcal{A}}[t] \in \mathbb{R}$ that matches an ***actual*** *demand for resources* $A_{env}[t] \in \mathbb{R}$ imposed by the environment env as accurately as possibly. This means that the sum of the absolute deviation $\sum_{t \in \mathcal{T}} |A_{env}[t] - A_{\mathcal{A}}[t]|$ over all time steps $t \in \mathcal{T}$ has to be minimized. While achieving this goal is paramount, the costs of providing these resources should be kept down.

A. Solving the Dynamic Resource Allocation Problem

We assume that the agents can not arbitrarily change their contribution from one time step to another. Instead, their behavior is subject to inertia (see Section II-D) – a property which can be found in many systems that control physical devices, such as power generators or energy storage devices. As the agents' contribution might thus not change quickly enough to reactively adapt to the demand in all situations, it has to be scheduled beforehand. To this end, in each time step t, each agent $a_k \in \mathcal{A}$ autonomously decides how much to contribute in future time steps $t_f \in \mathcal{T}_F$ (with $t < t_f$) by adjusting and announcing its own ***scheduled*** *contribution* $S_{a_k}[t_f]$. This decision is made on the basis of the remaining difference $(P_{env}[t_f] - S_{\mathcal{A}}[t_f])$ between the ***predicted*** *demand* $P_{env}[t_f]$ and the *total* ***scheduled*** *contribution* $S_{\mathcal{A}}[t_f] = \sum_{a_k \in \mathcal{A}} S_{a_k}[t_f]$ in future time steps $t_f \in \mathcal{T}_F$. To achieve the system's goal, the agents adapt their scheduled contribution iteratively until the remaining difference is minimized. The remaining difference is announced by the system in each iteration and time step.

B. Dynamics and Uncertainties

A predicted demand $P_{env}[t_f]$ has to be used because the actual demand $A_{env}[t_f]$ is based on a stochastic process driven by the environment. Predicted and actual demand might thus differ. As we assume that the predictions become more accurate as a point in time t_f approaches, the agents have to adapt their schedules when predictions change. A contribution scheduled for a time step t_f and determined in a time step t is thus not fixed but can be increased or decreased in later time steps t^* (with $t < t^* < t_f$).

The scheduled contribution $S_{a_k}[t_f]$ is the contribution a_k is actually *willing* to provide in t_f. However, e.g., because of technical difficulties, $S_{a_k}[t_f]$ might *unintentionally* differ from a_k's ***actual*** *contribution* $A_{a_k}[t_f]$. In addition, since the agents might be self-interested, they might *intentionally* lie about their scheduled contribution $S_{a_k}[t_f]$. The agents thus do **not** actually announce $S_{a_k}[t_f]$ but a ***pretended*** or ***predicted*** *contribution* $P_{a_k}[t_f]$ which is likely to deviate from $S_{a_k}[t_f]$ or $A_{a_k}[t_f]$. Therefore, the remaining difference is calculated as $(P_{env}[t_f] - P_{\mathcal{A}}[t_f])$ based on the *total* ***predicted*** *contribution* $P_{\mathcal{A}}[t_f] = \sum_{a_k \in \mathcal{A}} P_{a_k}[t_f]$ of all agents. Hence, $S_{\mathcal{A}}[t]$ and the *total* ***actual*** *contribution* $A_{\mathcal{A}}[t] = \sum_{a_k \in \mathcal{A}} A_{a_k}[t]$ might deviate from $P_{\mathcal{A}}[t]$ in a time step t.

C. Handling Uncertainties

Due to uncertainties introduced by inaccurate predictions, additional measures have to be taken. Because we assume that the system is able to observe the deviation between the predicted and the actual contribution or demand, it is possible to derive *expected values* for both. Thus, on the basis of experiences gained in prior time steps, an ***expected*** *demand* $E_{env}[t_f]$ can be derived from the demand $P_{env}[t_f]$ predicted by the environment and a *total* ***expected*** *contribution* $E_{\mathcal{A}}[t_f] = \sum_{a_k \in \mathcal{A}} E_{a_k}[t_f]$ can be derived from predictions $P_{a_k}[t_f]$. The remaining difference is thus redefined as $(E_{env}[t_f] - E_{\mathcal{A}}[t_f])$.

Despite the agents' possibly self-interested behavior, they have to cooperate to solve the system's task. This is due to the fact that each agent only has limited resources and rewards are chosen such that the utility of the individual agent is increased if it makes a higher contribution to achieve the system's goal. To incentivize agents to announce their actual contribution as accurately as possible, they receive higher rewards if the difference between $P_{a_k}[t]$ and $A_{a_k}[t]$ is small.

D. Contribution Constraints

Each agent a_k's actual contribution $A_{a_k}[t]$ is subject to a minimum contribution of $A_{a_k}^{min} \in \mathbb{R}$ and a maximum contribution of $A_{a_k}^{max} \in \mathbb{R}$ (with $A_{a_k}^{min} \leq A_{a_k}^{max}$). In case $0 \in [A_{a_k}^{min}, A_{a_k}^{max}]$, agents can decide against a contribution (i.e., $A_{a_k}[t] = 0$). The property of inertia is regarded in the way that $A_{a_k}[t]$ depends on a_k's contribution $A_{a_k}[t-1]$ in the previous time step $t-1$. This means that a_k can increase or decrease its actual contribution by at most $\Delta A_{a_k}^{max}[t] \in [0, A_{a_k}^{max} - A_{a_k}^{min}]$ from one time step to another. Because of this inertia as well as the minimum and maximum contribution, in each time step t, a_k can actually increase its actual contribution by at most $\Delta A_{a_k}^{+}[t]$ and decrease it by at most $\Delta A_{a_k}^{-}[t]$:

$$\Delta A_{a_k}^{+}[t] = \min\left\{\Delta A_{a_k}^{max}, A_{a_k}^{max} - A_{a_k}[t]\right\}$$
$$\Delta A_{a_k}^{-}[t] = \min\left\{\Delta A_{a_k}^{max}, A_{a_k}[t] - A_{a_k}^{min}\right\}$$

An agent a_k can thus change its contribution $A_{a_k}[t+1] = A_{a_k}[t] + \Delta A_{a_k}[t]$ from time step t to time step $t+1$ only by a value $\Delta A_{a_k}[t] \in [-\Delta A_{a_k}^{-}[t], \Delta A_{a_k}^{+}[t]]$. The agents also have to regard these constraints when deciding about their scheduled contribution $S_{a_k}[t]$. We assume that the agents' properties are attributes of the system and can not be changed from outside.

III. Running Example: Power Plant Scheduling in Autonomous Power Management Systems

In an autonomous power management system, current manual scheduling – i.e., the determination of the output levels of dispatchable prosumers[1] ("prosumer" stands for producers

[1] The physical constraints of dispatchable prosumers are usually a **superset** of those given in Section II-D.

as well as consumers), such as coal, biofuel, gas, or electric vehicles, for future time steps – is replaced by an autonomous mechanism. This requires prosumers to act as agents on their own behalf (see, e.g., [5], [6]) to proactively participate in the creation of schedules and in maintaining the stability of the grid. Ideally, not only large power plants are used in the scheduling process (as is the case today), but small prosumers participate as well. As a result, the load of non-dispatchable consumers can be satisfied locally, alleviating the need for complex distribution networks and allowing to locally deal with complications such as voltage band constraints [7].

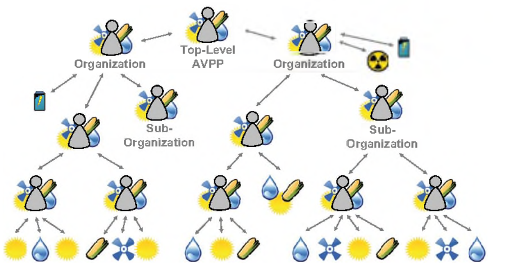

Fig. 1. Hierarchical system structure of a future autonomous power management system: prosumers are structured into systems of systems represented by AVPPs, thereby decreasing the complexity of control and scheduling. AVPPs can be part of other AVPPs and can represent organizations.

Such a scheme, however, puts a lot of strain on the computational system as hundreds or thousands of dispatchable prosumers have to be integrated. Since scheduling is an NP-hard problem, scalability becomes a major concern for systems of this size. The introduction of intermediaries, forming a hierarchy [8] of so-called *Autonomous Virtual Power Plants* (AVPP) [9] as shown in Figure 1, allows the regionalization of scheduling and to scale the system to the number of agents required. Within an AVPP, the schedules can be created in a centralized fashion. For this purpose, the intermediary must know the physical and economic limitations of the prosumers and combine them in a model. This model can then be solved by centralized optimization methods (see, e.g., [10], [11]). While centralized scheduling allows for very good solutions, has been investigated by the research community, and is in active use in the industry for large dispatchable power plants, it suffers from a number of drawbacks. Foremost, scheduling requires a model of each individual prosumer. Small cooperatives or individuals that own dispatchable prosumers, however, have no interest in disclosing all information necessary to create adequate schedules to the organization that performs this task. Cooperation-based approaches, such as TruCAOS (see Section IV), allow prosumers to formulate valid schedules without disclosing their internal models. Of course, such an approach has a need for increased communication and only yields sub-optimal solutions, but, since the scheduling problem is an instance of the DRAP presented in Section II, optimal solutions are not necessary (see Section I).

With respect to the DRAP, the actual demand for resources $A_{env}[t]$ corresponds to the *residual load*, i.e., the load that has to be fulfilled by the dispatchable prosumers. It is calculated from the difference between overall non-dispatchable load minus the output of intermittent power plants, such as wind and solar power plants. The actual contribution $A_{\mathcal{A}}[t]$ is the sum of the actual output of all dispatchable prosumers that participate in the scheduling process.[2] Each prosumer a_k participates with its actual output $A_{a_k}[t]$. The aim of the scheduling is to stipulate the future output $S_{a_k}[t_f]$ for all dispatchable prosumers a_k such that $\sum_{t \in \mathcal{T}} |A_{env}[t] - A_{\mathcal{A}}[t]|$ is minimal (see Section II). In contrast to the general formulation in Section II, in centralized scheduling, $S_{a_k}[t_f]$ is not autonomously defined by the agents but externally by the scheduling mechanism. Even though $A_{a_k}[t]$ might differ from the scheduled output $S_{a_k}[t]$, it is currently usually assumed that prosumers are benevolent and stick to their schedule as closely as possible.

Power management systems are an example of mission-critical systems. Because their failure can have massive consequences for people, industries, and public services, it is of utmost importance that they are stable and available at all times. However, a number of uncertainties, both originating in the system itself as well as in the environment, complicate the stable operation [6]. There are three main sources of uncertainty: the actual non-dispatchable load, the actual output of intermittent power plants, as well as the ability and willingness of dispatchable prosumers to stick to their schedules. Since the former two factors both contribute to the residual load, they are subsumed in this stochastic process. In this paper, we regard the residual load as part of the environment env. The predicted residual load $P_{env}[t_f]$ is calculated from predictions made by non-dispatchable consumers and intermittent power plants.

As mentioned in Section II-B and above, $S_{a_k}[t_f]$ must not equal $P_{a_k}[t_f]$ or $A_{a_k}[t_f]$ if agent autonomy permits the prosumers more freedom in their choices. Consequently, predictions give only an indication of the actual output and residual load. To create robust schedules, i.e., schedules that allow the system to satisfy the residual load despite uncertainties, the accuracy and credibility of the predictions has to be taken into account. As outlined in Section IV-B, we propose to use the concept of trust as a measure of prediction credibility in order to solve the DRAP on the basis of an expected residual load $E_{env}[t_f]$ and expected outputs $E_{a_k}[t_f]$ (see Section II-C).

An additional complication arises by the fact that predictions can change at runtime. Changes in the status of system components, for example, can change the predictions of non-dispatchable consumers and intermittent power plants which in turn changes the prediction of the residual load. Therefore, schedules have to be continuously adjusted.

IV. The Trust- and Cooperation-Based Algorithm

In this section, we propose a market-based algorithm, called *TruCAOS*, that solves the DRAP introduced in Section II in a cooperative and regionalized manner. With regard to our running example (see Section III), each intermediary, i.e., AVPP, satisfies a part of the overall residual load by allocating available resources of directly subordinated dispatchable prosumers. A market-based approach seems to be a natural choice because it inherently allows prosumers to keep their behavioral model private, and electricity infeed as well as consumption is already subject to monetary rewards and costs. In open systems, the former characteristic is of particular importance as we can not assume that the agents are willing to disclose information about their internal state. Even if they

[2]We assume that sealed and secure equipment, similar to smart meters, is available to measure the actual load and output.

did, other agents would not know if this information is correct. In the following, we focus on how a *single* AVPP solves the scheduling problem. The top-down process of allocating resources in the hierarchy of AVPPs is future work.

In its simplest form, TruCAOS does not include any measures of prediction accuracy and does not allow to identify prosumers that make erroneous proposals. To illustrate TruCAOS' basic procedure (see Section IV-A), we therefore initially assume benevolent behavior (i.e., scheduled and predicted contributions or demands are identical). In Section IV-B, we abandon the benevolence assumption and extend TruCAOS by trust-based measures, allowing to find robust solutions in open environments (e.g., by forming expectations of behavior).

A. Basic Procedure

In principle, in each time step, all dispatchable prosumers satisfy a part of the predicted residual load by selling or buying energy to or from their superordinate intermediary, i.e., the AVPP that controls them. This is done in an iterative and incremental process (see Figure 2) that, in its basic form, is reminiscent of an iteratively performed first-price sealed-bid auction (see, e.g., [12]).

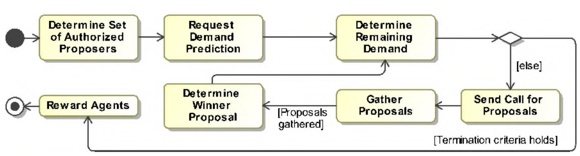

Fig. 2. TruCAOS' basic procedure

1) The Iterative Process: Before this process is started in the current time step t, the AVPP requests residual load predictions $P_{env}[t_f]$ for the next $|\mathcal{T}_F|$ time steps $t_f \in \mathcal{T}_F = \{t_1, ..., t_F\}$ (with $t < t_1 = t+1 \leq t_f \leq t_F$) from the environment. In all iterations $i \in \mathcal{I}$, the AVPP announces an auction in which a part of the *remaining residual load* $R[t_f] = P_{env}[t_f] - P_{\mathcal{A}}[t_f]$ (note that $R[t_f]$ might also be negative in specific time steps) should be satisfied by allocating additional resources of one of the dispatchable prosumers. The corresponding *call for proposals* $CFP = \langle (R[t_1], ..., R[t_f]), \mathcal{T}_F \rangle$ contains a sequence of the remaining residual load values as well as the regarded time frame $\mathcal{T}_F$. The *CFP* is sent to all *authorized proposers* $\mathcal{A}_{auth}$. In TruCAOS' basic form, $\mathcal{A}_{auth} = \mathcal{A}$, i.e., all prosumers are allowed to participate in the auction. Each dispatchable prosumer a_k that receives the call and wants to sell or buy energy to or from the AVPP responds with a proposal, i.e., a ***proposed*** *schedule*, $P^*_{a_k} = \langle (P^*_{a_k}[t_1], ..., P^*_{a_k}[t_f]), P^*_{a_k}[\overline{\kappa}] \rangle$ that includes a sequence of predicted contributions $P^*_{a_k}[t_f]$ as well as the *average costs* $P^*_{a_k}[\overline{\kappa}]$ for providing a contribution.[3] The *total costs* of the contribution specified in $P^*_{a_k}$ are defined as $\kappa(P^*_{a_k}) = P^*_{a_k}[\overline{\kappa}] \cdot \sum_{t_f \in \mathcal{T}_F} |P^*_{a_k}[t_f]|$. The AVPP gathers all proposals received in this iteration in a set $\mathcal{P}^*$ and completes this iteration by identifying and accepting a suitable proposal $P^* \in \mathcal{P}^*$ (see Section IV-A2). All other proposals are rejected.

For each dispatchable prosumer a_k, its *most recently* accepted proposal $P^*_{a_k}$ defines its ***pretended*** *schedule* $P_{a_k} = \langle (P_{a_k}[t_1], ..., P_{a_k}[t_f]), P_{a_k}[\overline{\kappa}] \rangle$ for the next $|\mathcal{T}_F|$ time steps. Note that P_{a_k} is created in the course of multiple time steps if $|\mathcal{T}_F| > 1$. In case the most recently accepted proposal $P^*_{a_k}$ was accepted in a previous time step, the AVPP assumes that, for each future time step $t'_f \in \mathcal{T}_F$ that is not covered by $P^*_{a_k}$, a_k does not contribute, i.e., $P_{a_k}[t'_f] = 0$. Moreover, in case there is no most recently accepted proposal $P^*_{a_k}$, we assume that $P_{a_k}[t_f] = 0$ and $P_{a_k}[\overline{\kappa}] = 0$ for all $t_f \in \mathcal{T}_F$. P_{a_k} is thus a contract between a_k and its superordinate AVPP that can be adjusted over time: a_k has to comply with its prediction in exchange for remuneration (see Section IV-A3). For this reason, whenever a_k wants to change its contribution, it has to make a proposal $P^*_{a_k}$ that is accepted by the AVPP and in turn replaces P_{a_k}. Hence, if $P^*_{a_k}$ is not accepted, it does not change a_k's pretended schedule P_{a_k}. The costs of the contribution specified in a schedule x (e.g., $x = P^*_{a_k}$ or $x = P_{a_k}$) for a specific time step t are defined as $\kappa(x, t) = x[\overline{\kappa}] \cdot |x[t]|$.

Each prosumer a_k creates its proposals with respect to the constraints introduced in Section II-D and its current state. Further, it has to take its scheduled contributions $S_{a_k}[t_f]$ as well as the *CFP* into account. Basically, proposals are generated by solving a constraint satisfaction optimization problem (CSOP) [13] (see Section V), which is composed of variables, their finite domains, constraints that restrict valid assignments, and an optimization function. Such a CSOP is similar to the one solved by a centralized approach (see Section III) but, since it only has to determine the proposal for a single agent, it is not subject to the scalability issues discussed earlier. However, as agents can change their behavior, the optimization problem might have different and more variable objectives.

The iterative process terminates when the remaining residual load is sufficiently satisfied, i.e., when the absolute values of the remaining residual load are below a predefined threshold $R_{max} \in \mathbb{R}^+$ (i.e., $\forall t_f \in \mathcal{T}_F : |R[t_f]| \leq R_{max}$), if the AVPP did not receive any proposals ($|\mathcal{P}^*| = 0$), if there does not exist a suitable proposal, or if a maximum number of iterations $i_{max} \in \mathbb{N}^+$ is exceeded (i.e., $\mathcal{I} = \{1, ..., i_{max}\}$).

Subsequently, the AVPP distributes rewards to the participants according to their schedules P_{a_k} (see Section IV-A3).

2) Determine Winner Proposal: The regarded DRAP is a multi-objective optimization problem: While satisfying the demand as accurately as possibly is the primary goal, the costs of providing the necessary resources should be kept down (see Section II). TruCAOS solves the DRAP by means of a multi-stage decision process that is based on heuristics. It is evident that this procedure does not allow to find optimal solutions. But, as stated in Section I, we only need solutions that are "good enough" and robust in the sense that the system can satisfy the demand.

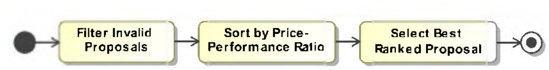

Fig. 3. The steps performed for "Determine Winner Proposal"

The AVPP determines $P^* \in \mathcal{P}^*$ based on a three step process. First, invalid proposals are filtered, then the remaining proposals are sorted, and finally the best one is selected (see Figure 3). Proposals that do not pass the filtering stage or are not selected are rejected.

[3] If $\overline{\kappa} < 0$, the prosumer offers money to the AVPP.

Filter Invalid Proposals: To satisfy the demand, only those proposals $P^*_{a_k} \in \mathcal{P}^*$ that would improve the overall satisfaction of the total remaining residual load by a minimum value of imp_{min} are regarded as valid and pass the filtering stage. This improvement is called the *gain in satisfaction* $G(P^*_{a_k})$ and is defined as follows ($R^*(P^*_{a_k}, t_f) = R[t_f] + P_{a_k}[t_f] - P^*_{a_k}[t_f]$ is the remaining residual load for time step t_f that would result in case $P^*_{a_k}$ was accepted and thus replaced P_{a_k}):

$$G(P^*_{a_k}) = \sum_{t_f \in \mathcal{T}_F} \left(|R[t_f]| - \left| R^*(P^*_{a_k}, t_f) \right| \right)$$

$P^*_{a_k}$ is valid if the predicate $valid(P^*_{a_k})$ evaluates to true:

$$valid(P^*_{a_k}) :\Leftrightarrow G(P^*_{a_k}) \geq imp_{min}$$

Sort Proposals: In the sorting stage, proposals are sorted by their price-performance ratio to keep the costs down. While this measure allows smaller prosumers with very limited resources to compete with bigger prosumers, it might increase the number of iterations needed to satisfy the demand. The price-performance ratio $PP(P^*_{a_k})$ is defined as the ratio of the gain in satisfaction $G(P^*_{a_k})$ to the change in the costs $\Delta\kappa(P^*_{a_k})$:

$$PP(P^*_{a_k}) = \frac{G(P^*_{a_k})}{\Delta\kappa(P^*_{a_k})}$$
$$\text{with } \Delta\kappa(P^*_{a_k}) = \sum_{t_f \in \mathcal{T}_F} \kappa(P^*_{a_k}, t_f) - \kappa(P_{a_k}, t_f)$$

If $\Delta\kappa(P^*_{a_k}) = 0$ and $G(P^*_{a_k}) > 0$, $PP(P^*_{a_k})$ is set to ∞. Otherwise, if $\Delta\kappa(P^*_{a_k}) = 0$ and $G(P^*_{a_k}) \leq 0$, $PP(P^*_{a_k})$ is set to $-\infty$.

Select best proposal: Finally, the best ranked proposal $P^* \in \mathcal{P}^*$ wins the auction of this iteration. For all $t_f \in \mathcal{T}_F$, the total predicted contribution $P_{\mathcal{A}}[t_f]$ is updated according to the predicted contributions $P_{a_k}[t_f]$ in the updated schedule P_{a_k}.

3) Rewarding Agents: In each time step $t+1$, the AVPP distributes a reward r_{a_k} to each agent a_k for its contribution in the previous time step t. The reward depends on a_k's pretended contribution $P_{a_k}[t]$ and the costs $P_{a_k}[\overline{\kappa}]$ stipulated in the schedule P_{a_k} for the *previous* time step:

$$r_{a_k} = \kappa(P_{a_k}, t) = P_{a_k}[\overline{\kappa}] \cdot |P_{a_k}[t]|$$

As agents might not adhere to their predicted contribution, we redefine the reward in Section IV-B2a.

B. Dynamic Resource Allocation under Uncertainty

To be applicable in open systems, TruCAOS must be able to deal with various uncertainties (see Section II and Section III) that originate from *information asymmetry* [14], among others. First, this means that TruCAOS has to identify uncertainties that stem from deviations between predicted and actual contributions or predicted and actual residual load values. Second, TruCAOS has to make informed decisions that allow dealing with these uncertainties at runtime. Third, TruCAOS should provide measures that incentivize the agents to behave beneficially, at least if this is in their power.

We meet these challenges by extending TruCAOS' basic procedure by a social system based on trust. Since trust stems from established contracts and their fulfillment, it allows to quantify uncertainties, identify uncooperative agents, and effectively lower their utility by sanctioning misbehavior (see Section IV-B2). Besides assessing the utility or risk to conclude a contract with a specific agent, trust thus serves as an enforcement mechanism that incentivizes rational agents to comply with their contracts [15].

1) Trust to Identify Uncertainties: With respect to predicted contributions, TruCAOS uses trust values to assess the risk that a contract, i.e., a pretended schedule P_{a_k}, is violated (see Section IV-A3). In each time step t and for each prosumer a_k, the AVPP gains an experience $e_{a_k}[t] = A_{a_k}[t] - P_{a_k}[t]$ that captures the difference between the actual and the predicted contribution.[4] For each a_k, these experiences are gathered in a sequence $\Gamma_{a_k} = (e_{a_k}[t - n_{\mathcal{A}} + 1], ..., e_{a_k}[t])$ that always contains the $n_{\mathcal{A}}$ newest experiences with a_k.

From these experiences, the AVPP can derive two trust values $T^{\pm}_{\mathcal{A}}(\mathcal{E}^c_{a_k})$ and $T^{abs}_{\mathcal{A}}(\mathcal{E})$ that both assess the AVPPs trust in a_k at time step t. $T^{\pm}_{\mathcal{A}}(\mathcal{E}^c_{a_k})$ is the trust that a_k provides a certain predicted contribution c. As $T^{\pm}_{\mathcal{A}}(\mathcal{E}^c_{a_k})$ depends on the predicted contribution, the AVPP can detect situations in which a_k's behavior depends on the amount of resources it pretends to provide in the future (but it also allows to mirror behavior that is independent of a specific contribution). For example, because of technical difficulties, there might be a power plant that makes rather accurate predictions in case its predicted contribution is rather low, but very inaccurate predictions if its predicted contribution is high. $T^{\pm}_{\mathcal{A}}(\mathcal{E}^c_{a_k})$ is therefore derived from the $n'_{\mathcal{A}}$ (with $n'_{\mathcal{A}} \leq n_{\mathcal{A}}$) newest experiences $\mathcal{E}^c_{a_k} \subseteq \Gamma_{a_k}$ that stem from predicted contributions that are most similar to c. We only regard the $n_{\mathcal{A}}$ and $n'_{\mathcal{A}}$ newest experiences because the agents' behavior may change over time, and to allow for *forgiveness* [16]. Further, when calculating trust values, we weight the relevant experiences according to their age. The older an experience $e_{a_k}[t_x]$, the lower its weight. With respect to the current time step t and a maximum weight $w_{max} > 1$, $w(w_{max}, t_x) = w_{max} - (t - t_x)$ (with $w(w_{max}, t_x) \in [1, w_{max}]$ and $t_x \leq t$) is the weight of an experience $e_{a_k}[t_x]$.

Based on a sequence of experiences $\mathcal{E}^c_{a_k}$, the trust $T^{\pm}_{\mathcal{A}}(\mathcal{E}^c_{a_k})$ in an agent a_k to provide a contribution c is defined as the weighted mean of measured differences between actual and predicted contributions, recorded in experiences $e_{a_k}[t_x]$:

$$T^{\pm}_{\mathcal{A}}(\mathcal{E}^c_{a_k}) = \sum_{e_{a_k}[t_x] \in \mathcal{E}^c_{a_k}} \frac{w(n_{\mathcal{A}}, t_x) \cdot \frac{e_{a_k}[t_x]}{\max\{e^{max}_{\mathcal{A}}, 1\}}}{\sum_{e_{a_k}[t_x] \in \mathcal{E}^c_{a_k}} w(n_{\mathcal{A}}, t_x)}$$

To assess the trust in a_k in relation to the other agents in $\mathcal{A}$, we use the maximum deviation $e^{max}_{\mathcal{A}} = \max\{|e_{a_k}[t_x]| \mid e_{a_k}[t_x] \in \Gamma_{a_k} \land a_k \in \mathcal{A}\}$ represented by an experience as a normalizing factor. Note that $T^{\pm}_{\mathcal{A}}(\mathcal{E}^c_{a_k}) \in [-1, 1]$. More precisely, it represents the *expected deviation* $\Delta^{\pm}_{\mathcal{A}}(\mathcal{E}^c_{a_k}) = T^{\pm}_{\mathcal{A}}(\mathcal{E}^c_{a_k}) \cdot \max\{e^{max}_{\mathcal{A}}, 1\}$ from a prediction $c = P^{(*)}_{a_k}[t_f]$ ($P^{(*)}_{a_k}[t_f]$ can either stand for a pretended $P_{a_k}[t_f]$ or a proposed contribution $P^*_{a_k}[t_f]$). The greater $|T^{\pm}_{\mathcal{A}}(\mathcal{E}^c_{a_k})|$, the more untrustworthy a_k. If $T^{\pm}_{\mathcal{A}}(\mathcal{E}^c_{a_k}) < 0$, the AVPP expects that a_k's actual contribution $A_{a_k}[t_f]$ is lower than its predicted contribution $P^{(*)}_{a_k}[t_f]$. Otherwise, if $T^{\pm}_{\mathcal{A}}(\mathcal{E}^c_{a_k}) \geq 0$,

[4]We assume that the AVPP is impartial and that it is able to measure the actual contribution as well as the actual demand (see Section III).

it is expected that $A_{a_k}[t_f]$ is greater than $P_{a_k}^{(*)}[t_f]$ or that a_k complies with its prediction, i.e., $A_{a_k}[t_f] = P_{a_k}^{(*)}[t_f]$. Note that positive and negative experiences can compensate one another, i.e., $T_{\mathcal{A}}^{\pm}(\mathcal{E}_{a_k}^c)$ might be 0 although a_k always deviates from predictions. Such situations can be captured in $T_{\mathcal{A}}^{abs}(\mathcal{E})$.

$T_{\mathcal{A}}^{abs}(\mathcal{E}) \in [0,1]$ represents the *expected **absolute** deviation* $\Delta_{\mathcal{A}}^{abs}(\mathcal{E}) = T_{\mathcal{A}}^{abs}(\mathcal{E}) \cdot \max\{e_{\mathcal{A}}^{max}, 1\}$ of an agent a_k from a prediction $P_{a_k}^{(*)}[t]$. It is derived from a set of experiences $\mathcal{E} \subseteq \Gamma_{a_k}$ (which experiences actually form the set $\mathcal{E}$ depends on the situation in which $T_{\mathcal{A}}^{abs}(\mathcal{E})$ is evaluated):

$$T_{\mathcal{A}}^{abs}(\mathcal{E}) = \sum_{e_{a_k}[t_x] \in \mathcal{E}} \frac{w(n_{\mathcal{A}}, t_x) \cdot \frac{|e_{a_k}[t_x]|}{\max\{e_{\mathcal{A}}^{max}, 1\}}}{\sum_{e_{a_k}[t_x] \in \mathcal{E}} w(n_{\mathcal{A}}, t_x)}$$

The greater $T_{\mathcal{A}}^{abs}(\mathcal{E})$, the more untrustworthy a_k.

The uncertainties associated with residual load predictions are identified in a similar way. However, because there are $|\mathcal{T}_F|$ demand predictions for every time step $t_f \in \mathcal{T}_F$, in each time step t, the AVPP gathers $|\mathcal{T}_F|$ experiences $e_{env}[t, t_m] = A_{env}[t] - P_{env}^{t_m}[t]$ that reflect the difference between the actual and the predicted demand. Here, $t_m \in \{t-1, ..., t-|\mathcal{T}_F|\}$ is the time step in which the demand $P_{env}^{t_m}[t]$ was predicted for time step t. Because the uncertainty of a predicted demand $P_{env}^{t_m}[t]$ depends on the distance $\delta \in \{1, ..., |\mathcal{T}_F|\}$ between the time step t_m in which and the time step t for which the prediction was made (predictions are assumed to improve over time), the AVPP evaluates the credibility of demand predictions with respect to this distance (in the following, we deliberately omit the exponent t_m). To capture this behavior, the AVPP manages $|\mathcal{T}_F|$ sequences of experiences $\Gamma_{env}^{\delta} = (e_{env}[t - n_{env} + 1, t - n_{env} + 1 - \delta], ..., e_{env}[t, t-\delta])$, one for each distance $\delta = \{1, ..., |\mathcal{T}_F|\}$. Each Γ_{env}^{δ} contains the n_{env} newest experiences $e_{env}[t_x, t_y]$ with $t_x - t_y = \delta$. Again, the AVPP assesses the expected accuracy of a demand prediction $d = P_{env}[t_f]$ on the basis of the n'_{env} (with $n'_{env} \leq n_{env}$) newest experiences $\mathcal{E}_{env}^{\delta,d} \subseteq \Gamma_{env}^{\delta}$ that stem from predicted demands that are most similar to d.

The corresponding trust value $T_{env}^{\pm}(\mathcal{E}_{env}^{\delta,d})$ in the context of a distance $\delta = t_f - t$ and a residual load prediction $d = P_{env}[t_f]$ is based on a metric that is analogously defined to $T_{\mathcal{A}}^{\pm}(\mathcal{E}_{a_k}^c)$. Instead of $e_{\mathcal{A}}^{max}$, it uses e_{env}^{max}, the maximum deviation from a demand prediction, to normalize the experiences. $T_{env}^{\pm}(\mathcal{E}_{env}^{\delta,d}) \in [-1, 1]$ thus represents an expected deviation $\Delta_{env}^{\pm}(\mathcal{E}_{env}^{\delta,d}) = T_{env}^{\pm}(\mathcal{E}_{env}^{\delta,d}) \cdot \max\{e_{env}^{max}, 1\}$ (with $\delta = t_f - t$) from a predicted demand $P_{env}[t_f]$.

2) Dealing with Uncertainties: AVPPs use trust values in various situations to identify and deal with uncertainties, e.g., by incentivizing beneficial behavior through indirect, trust-based sanctions that lower the utility of uncooperative agents. We assume that the agents know how misbehavior is sanctioned and the opportunity cost they incur imply that it is in the agents' best interest to behave benevolently.

a) Rewarding Agents: First, as a_k's actual contribution $A_{a_k}[t]$ might differ from the predicted contribution $P_{a_k}[t]$, we redefine the reward r_{a_k} agent a_k receives on the basis of its actual contribution:

$$r_{a_k} = P_{a_k}[\overline{\kappa}] \cdot |P_{a_k}[t]| - |\overline{P_{\mathcal{A}}[\overline{\kappa}]}| \cdot |P_{a_k}[t] - A_{a_k}[t]|$$

Here, $\overline{P_{\mathcal{A}}[\overline{\kappa}]} = \sum_{a_k \in \mathcal{A}} \frac{P_{a_k}[\overline{\kappa}]}{|\mathcal{A}|}$ are the average costs of resources, i.e., an average market price. It is derived from the pretended schedules P_{a_k} of all agents $a_k \in \mathcal{A}$. This means that a_k is charged for the deviation $|P_{a_k}[t] - A_{a_k}[t]|$ from its predicted contribution – even if a_k is not expected to gain a reward at all, i.e., if $P_{a_k}[\overline{\kappa}] = 0$.

b) Expected Demand and Contribution: Having introduced different trust metrics, the AVPP can now form expectations of the future demand and contribution. The expected demand in a time step $t_f \in \mathcal{T}_F$ is defined on the basis of a demand prediction $d = P_{env}[t_f]$ and an expected deviation that results from the corresponding trust value $T_{env}^{\pm}(\mathcal{E}_{env}^{\delta,d})$ as $E_{env}[t_f] = P_{env}[t_f] + \Delta_{env}^{\pm}(\mathcal{E}_{env}^{\delta,d})$ (with $\delta = t_f - t$). Analogously, the expected contribution of an agent a_k is defined on the basis of a predicted contribution $c = P_{a_k}^{(*)}[t_f]$ and the corresponding trust value $T_{\mathcal{A}}^{\pm}(\mathcal{E}_{a_k}^c)$ as $E_{a_k}^{(*)}[t_f] = P_{a_k}^{(*)}[t_f] + \Delta_{\mathcal{A}}^{\pm}(\mathcal{E}_{a_k}^c)$. Similarly to P_{a_k} and $P_{a_k}^*$, we define the ***expected** schedule* $E_{a_k} = \langle (E_{a_k}[t_1], ..., E_{a_k}[t_f]), E_{a_k}[\overline{\kappa}] \rangle$ as well as the *expected schedule* $E_{a_k}^*$ that would result in case a proposal $P_{a_k}^*$ was accepted. Note that the *CFP* now contains the *expected* remaining residual load $R[t_f] = E_{env}[t_f] - E_{\mathcal{A}}[t_f]$.

c) Restricting the Set of Authorized Proposers: The set of authorized proposers $\mathcal{A}_{auth} = \{a_k | a_k \in \mathcal{A} \wedge rdmNb() \leq T_{\mathcal{A}}^{abs}(\mathcal{E}_{a_k}^{auth})\}$ ($\mathcal{E}_{a_k}^{auth}$ corresponds to the $n'_{\mathcal{A}}$ newest experiences in Γ_{a_k} and $rdmNb() \in [0,1]$ generates a uniformly distributed random number) is now determined on the basis of the trustworthiness of the prosumers $a_k \in \mathcal{A}$. The more trustworthy a prosumer, the more likely it is contained in $\mathcal{A}_{auth}$. Here, we use $T_{\mathcal{A}}^{abs}(\mathcal{E}_{a_k}^{auth})$, i.e., an absolute expected deviation that is independent of a specific contribution, to sanction agents that deviate from their predictions in an arbitrary manner by limiting their access to the market. This measure incentivizes the agents to behave beneficially. However, apart from sanctioning misbehavior, the trust-based restriction of $\mathcal{A}_{auth}$ does not allow to reduce uncertainties in general.

d) Redefining Valid Proposals: The remaining residual load $R^*(P_{a_k}^*, t_f) = R[t_f] + E_{a_k}[t_f] - E_{a_k}^*[t_f]$ for time step t_f that would result in case $P_{a_k}^*$ was accepted is now based on the expected change in contribution, which results from $T_{\mathcal{A}}^{\pm}(\mathcal{E}_{a_k}^c)$. The gain in satisfaction $G(P_{a_u}^*)$ is thus an expectation. Although $P_{a_k}^*[t_f]$ might be greater than $P_{a_k}[t_f]$, because of contribution-specific uncertainties associated with an agent a_k, $E_{a_k}^*[t_f]$ might be smaller than $E_{a_k}[t_f]$. For the same reason, regarding two proposals $P_{a_u}^*$ and $P_{a_v}^*$ of two prosumers a_u and a_v, the expected gain in satisfaction $G(P_{a_u}^*)$ might be higher than $G(P_{a_v}^*)$ although the overall contribution predicted in $P_{a_u}^*$ is smaller than in $P_{a_v}^*$.

e) Expected Price-Performance Ratio: When determining the price-performance ratio $PP(P_{a_k}^*)$ of a proposal $P_{a_k}^*$, TruCAOS now relies on the expected gain in satisfaction, which is based on $T_{\mathcal{A}}^{\pm}(\mathcal{E}_{a_k}^c)$. Moreover, because positive and negative deviations can compensate one another in $T_{\mathcal{A}}^{\pm}(\mathcal{E}_{a_k}^c)$, we additionally estimate the uncertainty of a contribution by means of the expected absolute deviation based on the trust value $T_{\mathcal{A}}^{abs}(\mathcal{E}_{a_k}^c)$ (here, $T_{\mathcal{A}}^{abs}(\mathcal{E}_{a_k}^c)$ is derived from the same experience as $T_{\mathcal{A}}^{\pm}(\mathcal{E}_{a_k}^c)$). In a sense, $T_{\mathcal{A}}^{abs}(\mathcal{E}_{a_k}^c)$ estimates the *risk* of a deviation. If this risk decreases in case a proposal is accepted, i.e., if $\Delta_{risk} < 0$, it is assumed that the system's stability increases. By incorporating Δ_{risk} in $PP(P_{a_k}^*)$, we expect the AVPP to be able to make more robust decisions. Therefore, we redefine $PP(P^*_{a_k})$ as follows:

$$PP(P^*_{a_k}) = \frac{G(P^*_{a_k}) - \Delta_{risk}}{\Delta\kappa(P^*_{a_k})}$$
$$\text{with } \Delta_{risk} = \sum_{t_f \in \mathcal{T}_F} \left(\Delta^{abs}_{\mathcal{A}}(\mathcal{E}^{c_1}_{a_k}) - \Delta^{abs}_{\mathcal{A}}(\mathcal{E}^{c_2}_{a_k})\right),$$
$$c_1 = P^*_{a_k}[t_f] \text{ and } c_2 = P_{a_k}[t_f]$$

As before, $PP(P^*_{a_k})$ is set to ∞ if $\Delta\kappa(P^*_{a_k}) = 0$ and $G(P^*_{a_k}) - \Delta_{risk} > 0$, and to $-\infty$ if $\Delta\kappa(P^*_{a_k}) = 0$ and $G(P^*_{a_k}) - \Delta_{risk} \leq 0$. In future work, we want the system to schedule reserves to deal with deviations between demand and contribution at runtime. For this purpose, we need a better estimation of the risk, e.g., by predicting multiple possible future developments by means of trust-based scenarios [3]. In case of deviations, a reactive mechanism (see, e.g., [17]) could then utilize these reserves to compensate for deviations and re-establish the balance without having to recalculate schedules.

Further, because each agent a_k is only rewarded for $min\{A_{a_k}[t], P_{a_k}[t]\}$ (see Section IV-B2a), we redefine $\Delta\kappa(P^*_{a_k})$ such that it represents the expected change in costs. As we do not know $A_{a_k}[t]$ in advance, we make use of the expected contribution based on the trust value $T^{\pm}_{\mathcal{A}}(\mathcal{E}^c_{a_k})$:

$$\Delta\kappa(P^*_{a_k}) = \sum_{t_f \in \mathcal{T}_F} \big(\min\{\kappa(P^*_{a_k}, t_f), \kappa(E^*_{a_k}, t_f)\} - \min\{\kappa(P_{a_k}, t_f), \kappa(E_{a_k}, t_f)\}\big)$$

This trust-based influence on the price-performance ratio, results in price premiums and price discounts [14] because trustworthy agents can charge higher prices for a contribution than untrustworthy agents.

In the following section, we show that TruCAOS' trust-based principle allows to identify and deal with uncertainties.

V. EVALUATION

For evaluation, we compare results we achieved with TruCAOS in two different scenarios. In the first scenario "S-NT", TruCAOS does not make any use of trust values, whereas it uses trust values in the second scenario "S-T". For each scenario, we performed 500 simulation runs. Before we discuss our results in Section V-B, we introduce our test bed.

A. Test Bed

We regarded a setting of a single AVPP consisting of 10 dispatchable power plants of different types (hydro, biofuel, and gas power plants) with different physical properties which are based on real data. To be more precise, all power plants a_k had a minimum contribution $A^{min}_{a_k} = 0$ kW, a maximum contribution $A^{max}_{a_k} \in [40 \text{ kW}, 450 \text{ kW}]$, and a maximum change in contribution of $\Delta A^{max}_{a_k}[t] \in [36 \text{ kW}, 450 \text{ kW}]$ from one time step to another. For the average costs of providing a contribution, we had $\overline{\kappa} \in \{9\frac{\text{EUR cent}}{\text{kWh}}, 10\frac{\text{EUR cent}}{\text{kWh}}\}$.

With regard to the DRAP, the power plants had to satisfy a predefined actual residual load $A_{env}[t] \in [815 \text{ kW}, 1310 \text{ kW}]$, i.e., a demand, over a period of two days, which corresponds to 192 discrete time steps $t \in \mathcal{T} = \{1, ..., 192\}$ with a resolution of 15 minutes per time step (a standard practice in current power management systems). According to the DRAP, the system's goal was to minimize the absolute deviation $DEV = \sum_{t \in \mathcal{T}} |A_{env}[t] - A_{\mathcal{A}}[t]|$ between the actual residual load $A_{env}[t]$ and the total actual contribution of resources $A_{\mathcal{A}}[t]$ over all time steps $t \in \mathcal{T}$. In each time step, the AVPP used TruCAOS to create power plant schedules for the next $4 = |\mathcal{T}_F|$ time steps $t_f \in \mathcal{T}_F$ (i.e., for the next hour). The environment created residual load predictions for these future time steps by adding a random prediction error – generated by using a Gaussian distribution – to $A_{env}[t_f]$. The residual load showed two different behaviors env_1 and env_2. For env_1, we used a mean prediction error of $\mu_{env_1} = 50$ kW and a standard deviation of $\sigma_{env_1} = 5$ kW. The residual load behaved according to env_1 in time steps $t \in \{1, ..., 48\} \cup \{97, ..., 144\}$. In time steps $t \in \{49, ..., 96\} \cup \{145, ..., 192\}$, we used env_2 with $\mu_{env_2} = -100$ kW and $\sigma_{env_2} = 10$ kW. Uncertainty regarding future time steps was implemented by increasing the mean prediction error with the time horizon of predictions by 10 kW for env_1 and 20 kW for env_2.

The evaluation was implemented in a sequential, round-based execution model. Each round corresponds to a specific time step $t \in \mathcal{T}$. At the beginning of each round, the AVPP requested residual load predictions $P_{env}[t_f]$ from the environment for the next 4 time steps t_f, determined the remaining residual load, iteratively sent a call for proposals to the power plants, and accepted suitable proposals until a termination criteria was met ($R_{max} = 1$ kW). In S-T, the AVPP used the trust metrics presented in Section IV-B1 ($n_{\mathcal{A}} = 50$, $n'_{\mathcal{A}} = 5$, $n_{env} = 10$, and $n'_{env} = 5$ proved to be beneficial in preliminary tests) to determine the expected demand $E_{env}[t_f]$ and the expected contributions $E^{(*)}_{a_k}[t_f]$. The evaluation shows that, due to our trust metrics, $E_{env}[t_f]$ is 78% more accurate than $P_{env}[t_f]$.

As stated in Section IV-A1, the power plants generated proposals by solving a CSOP using IBM ILOG CPLEX[5]. The CSOP's objective was to minimize the remaining residual load specified in the CFP. While all power plants had this fixed behavior, we used a "malicious power plant" (MPP) to introduce uncertainty. The MPP predicted future outputs $P_{\text{MPP}}[t_f]$ that deviated from its actual and scheduled output $A_{\text{MPP}}[t_f] = S_{\text{MPP}}[t_f]$ according to two different predefined behaviors mpp_1 and mpp_2. Regarding mpp_1 / mpp_2, the MPP's actual output was 30% / 10% higher than predicted if $P_{\text{MPP}}[t_f] = A^{min}_{\text{MPP}} = 0$ kW, and 30% / 10% lower than predicted if $P_{\text{MPP}}[t_f] = A^{max}_{\text{MPP}} = 450$ kW. Between A^{min}_{MPP} and A^{max}_{MPP}, the prediction error changed linearly with $P_{\text{MPP}}[t_f]$ such that the power plant did not deviate from predictions at its sweet spot $P_{\text{MPP}}[t_f] = 225$ kW. The MPP showed behavior mpp_1 in time steps $t \in \{1, ..., 72\}$ and behavior mpp_2 in time steps $t \in \{73, ..., 192\}$ For all other power plants, the scheduled, predicted, and actual output were always identical.

B. Results

In S-NT, the AVPP allocated resources according to the predicted residual load P_{env}. As depicted in Figure 4, the mean absolute deviation $\Delta(A_{env}, P_{env})$ between the actual A_{env} and the predicted P_{env} residual load varies with the environment's

[5]http://www-01.ibm.com/software/integration/optimization/cplex-optimizer/

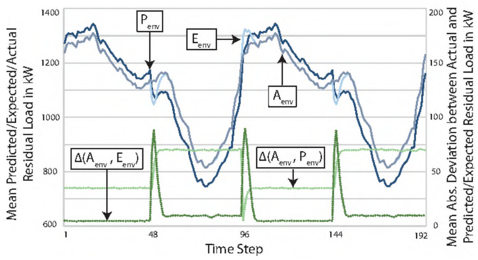

Fig. 4. Mean predicted P_{env}, expected E_{env}, and actual A_{env} residual load. The mean absolute deviation $\Delta(A_{env}, P_{env})$ between A_{env} and P_{env} shows the predefined randomly generated prediction error. The mean absolute deviation $\Delta(A_{env}, E_{env})$ between A_{env} and E_{env} reflects the accuracy of E_{env}, which was calculated by means of the presented trust metrics in S-T.

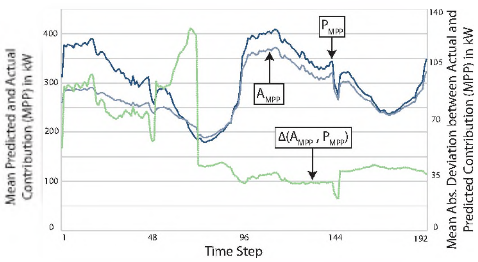

Fig. 5. The predicted P_{MPP} and actual A_{MPP} contribution of the MPP in S-NT. The mean absolute deviation $\Delta(A_{\text{MPP}}, P_{\text{MPP}})$ between A_{MPP} and P_{MPP} mirrors the MPP's varying behavior.

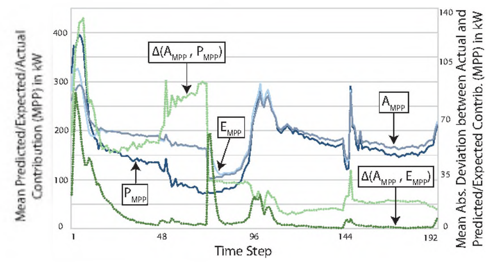

Fig. 6. The predicted P_{MPP}, expected E_{MPP}, and actual A_{MPP} contribution of the MPP in S-T. As in Figure 5, the mean absolute deviation $\Delta(A_{\text{MPP}}, P_{\text{MPP}})$ between A_{MPP} and P_{MPP} mirrors the MPP's varying behavior. The small mean absolute deviation $\Delta(A_{\text{MPP}}, E_{\text{MPP}})$ between A_{MPP} and E_{MPP} shows that TruCAOS' trust-based decisions mitigate uncertainties.

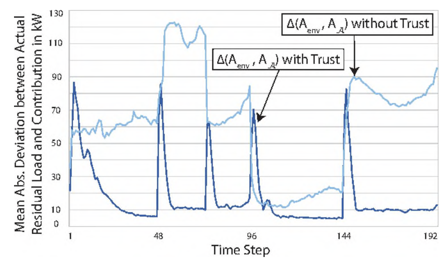

Fig. 7. The mean absolute deviation $\Delta(A_{env}, A_{\mathcal{A}})$ between the actual residual load A_{env} and the total actual contribution $A_{\mathcal{A}}$ in S-NT compared to S-T. TruCAOS' trust-based decisions significantly decrease $\Delta(A_{env}, A_{\mathcal{A}})$.

behavior env_1 and env_2 (the average of $\Delta(A_{env}, P_{env})$ is 51.81 kW, the standard deviation $\sigma = 17.88$ kW). Regarding S-T, the AVPP assessed the quality of predictions by means of the trust metrics and solved the DRAP on the basis of the expected residual load E_{env}. The small mean absolute deviation $\Delta(A_{env}, E_{env})$ between A_{env} and E_{env} indicates that the AVPP is able to anticipate prediction errors very accurately (the average of $\Delta(A_{env}, E_{env})$ is 11.47 kW, $\sigma = 15.78$ kW). Due to the change in the environment's behavior, $\Delta(A_{env}, E_{env})$ is rather high around time steps 48, 96, and 144. While $\Delta(A_{env}, E_{env})$ exceeds $\Delta(A_{env}, P_{env})$ at these time steps, the short time frames in which these deviations are present indicate that the trust values quickly adapt to behavioral changes.

With regard to S-NT, the predicted contribution P_{MPP} of the MPP correlates with P_{env} (see Figure 5), which is also mirrored by the shapes of the two corresponding curves. In time step 72, the change in MPP's behavior from mpp_1 to mpp_2 is clearly visible. As the mean absolute deviation $\Delta(A_{\text{MPP}}, P_{\text{MPP}})$ of actual contributions A_{MPP} from predicted contributions P_{MPP} depends on the MPP's predicted output, $\Delta(A_{\text{MPP}}, P_{\text{MPP}})$ varies with P_{MPP}. The MPP's malicious behavior causes an average of $\Delta(A_{\text{MPP}}, P_{\text{MPP}})$ of 55.15 kW with $\sigma = 27.50$ kW. Over all time steps, the MPP received an average reward of 1074.06 EUR.

With respect to S-T, Figure 6 depicts how the AVPP allocates resources of the MPP in case trust-based decisions are made. The AVPP needed approximately 15 time steps until the trust values accurately reflected the MPP's behavior. Starting at time step 15, the deviation of the expected E_{MPP} from the actual A_{MPP} contribution is almost vanished. The MPP's malicious behavior causes an average of $\Delta(A_{\text{MPP}}, E_{\text{MPP}})$ of only 8.84 kW with $\sigma = 14.77$ kW. At time step 72, the MPP changed its behavior. Again, about 5 time steps were needed to adapt to the new behavior. In the course of these time steps, however, E_{MPP} deviated from A_{MPP} more than P_{MPP} did. Interestingly, the AVPP allocated the resources of the MPP such that P_{MPP} was near its sweet spot of 225 kW (the average of $\Delta(A_{\text{MPP}}, P_{\text{MPP}})$ is 38.20 kW with $\sigma = 29.71$ kW). This shows that TruCAOS' trust-based decisions actively mitigate uncertainties. The mean reward MPP received was 541.13 EUR, and thus 50% less than in S-NT. Moreover, on average, in ≈ 16 of the 192 time steps, the MPP was not contained in the set of authorized proposers $\mathcal{A}_{auth}$. Consequently, TruCAOS' trust-based decisions incentivize agents to behave beneficially.

Figure 7 summarizes our observations: By using trust-based techniques in S-T, the average of the mean absolute deviation $\Delta(A_{env}, A_{\mathcal{A}})$ between A_{env} and the total actual contribution $A_{\mathcal{A}}$ is only 17.19 kW ($\sigma = 17.86$ kW), compared to 61.92 kW ($\sigma = 31.18$ kW) without trust-based measures in S-NT. In S-T, $DEV = 3283.63$ kW. This is 72% lower than in S-NT ($DEV = 11826.00$ kW). The number of iterations needed to allocate the resources was approximately proportional to the residual load that had to be fulfilled. On average, TruCAOS terminated after 8.88 iterations in S-NT and 9.88 iterations in S-T. In S-T / S-NT, the total costs of satisfying the residual load were only 1.4% / 1.2% higher than the optimal total costs (in the worst case, 7% were possible)[6].

VI. Related Work

While resource allocation has been an important topic of research, the incorporation of uncertainties is a relatively recent

[6] To determine comparable values, we used the reward function introduced in Section IV-A3 and compensated for deficits in contribution with the most expensive power plants. The actual total costs were lower.

development. An important aspect of this is the institutionalization of procedural rules in the exploitation of common pool resources (CPR), as considered by Pitt and Schaumeier [18] based on Elinor Ostrom's Principles for Enduring Institutions. To sustainably exploit a CPR, agents need to cooperate as governed by a set of modifiable rules. Uncertainties arise when agents behave selfishly, e.g., by consuming more resources than they were appropriated. In such cases, other agents (i.e., the institution they form) have to sanction them in a way that incentivizes them to behave better in the future. While Pitt et al.'s work is focused on endogenic resources that can be fully exhausted and have to be managed in a sustainable way, we are working with exogenic ones that are supplied constantly. It is also necessary that the full demand is covered by the agents' contributions even if that means that some agents have to contribute more then they would like to.

As a specialization of this kind of incentive-based mechanism design, Dash et al. introduce the notion of trust-based mechanism design [4] which adds the concept of trust to the factors costs and valuations in decision making. The presented algorithm is applied to a task allocation scenario in which agents post tasks to an auctioneer that in turn decides on an allocation on the basis of received proposals. When deciding on this allocation, the auctioneer takes the agents' trustworthiness into account, which is derived from experiences reported by their former interaction partners. The auctioneer's goal is to maximize the sum of the expected utilities of the agents. A trust-based payment and allocation scheme is introduced that chooses the most trustworthy and cheapest agents to fulfill an allocation on the one hand and incentivizes the agents to tell the truth about their costs, valuations, and the experiences with other agents on the other hand. While [4] served as a theoretical foundation for our investigations, our DRAP differs from the task allocation problem regarded there, e.g., in its objective. In contrast to [4], TruCAOS' use of trust-based techniques allows it to identify and cope with agents that unintentionally show non-beneficial behavior.

Based on the principle of trust-based mechanism design, the literature presents various marked-based approaches in the context of supply and demand management in smart grids that prevent strategic misbehavior and gambling through pricing mechanisms (e.g., [19], [20]). In contrast to [20], TruCAOS can proactively identify and cope with agents that unintentionally show non-beneficial behavior. To reactively deal with unintentional prediction errors, [20] proposes an additional online balancing mechanism that compensates for deviations between demand and contributions. The mechanism presented in [19] mitigates these uncertainties by forming coalitions of agents such that prediction errors cancel each other.

In the domain of power management systems, there are multiple approaches that try to solve the scheduling problem on the basis of a definition that is similar to our DRAP. Apart from centralized algorithms that require a detailed model of the agents (see, e.g., [10], [11]), there are several decentralized approaches that are based on the principles of electronic markets (see, e.g., [21], [22], [23]). DEZENT [21] is a market-based approach to balance energy supply and load in a hierarchical system structure in a bottom-up manner. A further approach based on a hierarchical system structure is the PowerMatcher [22] in which the root of a tree balances supply and load by determining an equilibrium price, based on aggregated load, supply, and price predictions, to establish a market equilibrium. The auctioneer is thus a central component of the system. Unlike TruCAOS, DEZENT and PowerMatcher only create schedules for the next instead of multiple future time steps. Stigspace [23] is a coordination mechanism that uses a blackboard, called stigspace, as the medium of communication between distributed energy resources in order to create schedules in an iterative process. Initially, the stigspace is used to announce the load that has to be fulfilled by the distributed energy resources. These in turn revise their schedule (i.e., their contribution for the regarded time frame) in order to minimize the remaining load. The new schedule is then posted to the stigspace, where the remaining load is updated accordingly. This process is repeated until the load is sufficiently satisfied. DEZENT, PowerMatcher, Stigspace, and TruCAOS have in common that the agents do not have to disclose their behavioral model. However, in contrast to TruCAOS, the other approaches do not deal with uncertainties associated with load and supply predictions. With regard to open MAS, we therefore expect that TruCAOS can solve the DRAP in a more robust way.

In [24], Hinrichs et al. present a self-organizing heuristic, called $COHDA_2$, that solves a combinatorial optimization problem. The algorithm is based on a neighborhood structure in which agents iteratively negotiate about their configuration in order to achieve a global as well as individual objectives. The former can be thought of as the system's objective in our DRAP, that is to allocate resources according to a given demand. In $COHDA_2$, every time an agent changes its configuration, it informs its neighbors about its new configuration as well as the configuration of its neighbors and the neighbors' neighbors etc. it received in a prior time step. Although each agent thus only has current information about its direct neighborhood, it builds a complete representation of the configuration of the other agents in the system over time. To decide on suitable own configurations, the agents take this locally perceived global configuration into account. As it is very likely that this configuration differs from the actual configuration of the other agents, the agents keep track of and inform their neighbors about the best configuration they have achieved so far. While $COHDA_2$ solves the combinatorial optimization problem in a truly decentralized manner, the quality of its solutions highly depends on the topology of the overlay network defining the agents' neighborhood. Similarly to TruCAOS, the agents do not disclose information about their behavioral model. However, in contrast to TruCAOS, $COHDA_2$ does not explicitly deal with uncertainties.

The resource allocation problem described in [17], is very similar to our DRAP. However, the algorithms in [17] solve the problem by reacting to deviations between demand and contributions. While they could also be used to proactively allocate resources, they do not take uncertainties into account.

VII. Conclusion and Future Work

In this paper, we formulated a dynamic resource allocation problem (DRAP) whose goal is to find an allocation in which the agents of an open multi-agent system (MAS) provide a total contribution that satisfies a demand for resources that is imposed by the environment. As we assume that the agents' contribution is subject to inertia – a typical property of physical devices, such as power generators – the DRAP has to be solved for a specific future time frame in advance. On this basis, we further presented a **t**rust- and **c**ooperation-based **a**lgorithm for **o**pen MAS, called *TruCAOS*, that solves instances of the DRAP in an iterative and incremental process. Because of TruCAOS' market-based principle, the agents do not have to disclose their behavioral model – a property that is of particular interest in open MAS. Since contribution and demand predictions might (un)intentionally deviate from their actual values, TruCAOS further uses the concept of trust to form expectations of the actual contribution and demand on the basis of experiences gained in the past. Our evaluation shows that TruCAOS' trust-based decisions allow anticipating uncertainties, sanction misbehavior, and incentivize cooperation. Therefore, TruCAOS yields results that are robust in the sense that – despite the uncertainties and the dynamic nature of the DRAP – they satisfy the demand very accurately. We illustrated our investigations on the basis of a self-organizing power management system consisting of a hierarchy of Autonomous Virtual Power Plants (AVPPs) that have to create schedules for dispatchable power plants and consumers to maintain the balance between energy production and consumption.

In future work, we will pursue the integration of TruCAOS in the hierarchical system of AVPPs by means of intermediaries. By decomposing the DRAP for the entire system defined on the macro-level into DRAPs on the meso-level, intermediaries solve the overall DRAP in a regionalized manner. For this purpose, each intermediary creates a model of the directly subordinated agents trough observation. Moreover, we will combine the proactive principle of TruCAOS with a reactive mechanism, such as one of those proposed in [17], to compensate for deviations between actual demand and contributions at runtime without having to recalculate scheduled contributions.

Acknowledgment

This research is partly sponsored by the research unit *OC-Trust* (FOR 1085) of the German Research Foundation.

References

[1] A. Bar-Noy, R. Bar-Yehuda, A. Freund, J. Naor, and B. Schieber, "A Unified Approach to Approximating Resource Allocation and Scheduling," *Journal of the ACM (JACM)*, vol. 48, no. 5, pp. 1069–1090, 2001.

[2] A. Artikis, M. Sergot, and J. Pitt, "Specifying Norm-Governed Computational Societies," *ACM Transactions on Computational Logic (TOCL)*, vol. 10, no. 1, p. 1, 2009.

[3] G. Anders, F. Siefert, J.-P. Steghöfer, and W. Reif, "Trust-Based Scenarios – Predicting Future Agent Behavior in Open Self-Organizing Systems," in *Proc. of the 7th International Workshop on Self-Organizing Systems (IWSOS 2013)*, May 2013.

[4] R. K. Dash, S. D. Ramchurn, and N. R. Jennings, "Trust-Based Mechanism Design," in *Proceedings of the Third International Joint Conference on Autonomous Agents and Multiagent Systems-Volume 2*. IEEE Computer Society, 2004, pp. 748–755.

[5] S. D. McArthur, E. M. Davidson, V. M. Catterson, A. L. Dimeas, N. D. Hatziargyriou, F. Ponci, and T. Funabashi, "Multi-Agent Systems for Power Engineering Applications – Part I: Concepts, Approaches, and Technical Challenges," *IEEE Transactions on Power Systems*, vol. 22, no. 4, pp. 1743–1752, 2007.

[6] S. Ramchurn, P. Vytelingum, A. Rogers, and N. Jennings, "Putting the "Smarts" into the Smart Grid: A Grand Challenge for Artificial Intelligence," *Communications of the ACM*, vol. 55, no. 4, pp. 86–97, 2012.

[7] M. Blank, S. Gerwinn, O. Krause, and S. Lehnhoff, "Support Vector Machines for an Efficient Representation of Voltage Band Constraints," in *2nd IEEE PES International Conference and Exhibition on Innovative Smart Grid Technologies (ISGT Europe)*. IEEE, 2011, pp. 1–8.

[8] J.-P. Steghöfer, P. Behrmann, G. Anders, F. Siefert, and W. Reif, "HiSPADA: Self-Organising Hierarchies for Large-Scale Multi-Agent Systems," in *Proceedings of the Ninth International Conference on Autonomic and Autonomous Systems (ICAS)*. IARIA, 2013.

[9] G. Anders, F. Siefert, J.-P. Steghöfer, H. Seebach, F. Nafz, and W. Reif, "Structuring and Controlling Distributed Power Sources by Autonomous Virtual Power Plants," in *Proc. of the Power & Energy Student Summit 2010 (PESS 2010))*, October 2010, pp. 40–42.

[10] J. Heo, K. Lee, and R. Garduno-Ramirez, "Multiobjective Control of Power Plants Using Particle Swarm Optimization Techniques," *Energy Conversion, IEEE Transactions on*, vol. 21, no. 2, pp. 552–561, 2006.

[11] A. Zafra-Cabeza, M. Ridao, I. Alvarado, and E. Camacho, "Applying Risk Management to Combined Heat and Power Plants," *Power Systems, IEEE Transactions on*, vol. 23, no. 3, pp. 938–945, 2008.

[12] P. Klemperer, "What Really Matters in Auction Design," *The Journal of Economic Perspectives*, vol. 16, no. 1, pp. 169–189, 2002.

[13] E. Tsang, *Foundations of constraint satisfaction*. Academic press London, 1993, vol. 289.

[14] S. Ba and P. Pavlou, "Evidence of the Effect of Trust Building Technology in Electronic Markets: Price Premiums and Buyer Behavior," *MIS Quarterly*, vol. 26, no. 3, pp. 243–268, 2002.

[15] S. Ba, A. Whinston, and H. Zhang, "Building Trust in the Electronic Market through an Economic Incentive Mechanism," in *Proceedings of the 20th international conference on Information Systems*. Association for Information Systems, 1999, pp. 208–213.

[16] A. Vasalou and J. Pitt, "Reinventing Forgiveness: A Formal Investigation of Moral Facilitation," in *Trust Management*, ser. Lecture Notes in Computer Science, P. Herrmann, V. Issarny, and S. Shiu, Eds. Springer Berlin / Heidelberg, 2005, vol. 3477, pp. 39–90.

[17] G. Anders, C. Hinrichs, F. Siefert, P. Behrmann, W. Reif, and M. Sonnenschein, "On the Influence of Inter-Agent Variation on Multi-Agent Algorithms Solving a Dynamic Task Allocation Problem under Uncertainty," in *2012 Sixth IEEE International Conference on Self-Adaptive and Self-Organizing Systems (SASO)*. IEEE Computer Society, Washington, D.C., 2012, pp. 29–38.

[18] J. Pitt, J. Schaumeier, and A. Artikis, "The Axiomatisation of Socio-Economic Principles for Self-Organising Systems," in *Self-Adaptive and Self-Organizing Systems (SASO), 2011 Fifth IEEE International Conference on*, 2011, pp. 138–147.

[19] G. Chalkiadakis, V. Robu, R. Kota, A. Rogers, and N. R. Jennings, "Cooperatives of Distributed Energy Resources for Efficient Virtual Power Plants," in *The 10th International Conference on Autonomous Agents and Multiagent Systems*, ser. AAMAS '11, vol. 2. Richland, SC: International Foundation for Autonomous Agents and Multiagent Systems, 2011, pp. 787–794.

[20] P. Vytelingum, S. Ramchurn, T. Voice, A. Rogers, and N. Jennings, "Trading Agents for the Smart Electricity Grid," in *Proceedings of the 9th International Conference on Autonomous Agents and Multiagent Systems*, vol. 1. International Foundation for Autonomous Agents and Multiagent Systems, 2010, pp. 897–904.

[21] H. Wedde, "DEZENT – A Cyber-Physical Approach for Providing Affordable Regenerative Electric Energy in the Near Future," in *2012 38th EUROMICRO Conf. on Software Engineering and Advanced Applications (SEAA)*, 2012, pp. 241–249.

[22] 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.

[23] J. Li, G. Poulton, and G. James, "Coordination of Distributed Energy Resource Agents," *Applied Artificial Intelligence*, vol. 24, no. 5, pp. 351–380, 2010.

[24] C. Hinrichs, M. Sonnenschein, and S. Lehnhoff, "Evaluation of a Self-Organizing Heuristic for Interdependent Distributed Search Spaces," in *International Conference on Agents and Artificial Intelligence (ICAART 2013)*, vol. Volume 1 – Agents. SciTePress, 2013, pp. 25–34.