Active Learning for Model Abstraction*

Alexander Schiendorfer*, Christoph Lassner†, Gerrit Anders*, Wolfgang Reif*, and Rainer Lienhart†
Institute for Software & Systems Engineering*
Multimedia Computing and Computer Vision Lab†
University of Augsburg, Germany
Email: {alexander.schiendorfer, christoph.lassner, anders, reif, lienhart}@informatik.uni-augsburg.de

***Abstract*—Organizational structures such as hierarchies provide an effective means to deal with the increasing complexity found in large-scale energy systems that results from uncertainties in nature as well as computational efforts in scheduling. In hierarchical systems, abstraction-based methods provide a way to calculate a simpler behavior model to be used in optimization NOTE: , i.e., scheduling, in lieu of a combination of a set of behavior models. In particular, functional dependencies over the combinatorial domain are approximated by repeatedly sampling input-output pairs and substituting the actual function by piecewise linear functions. However, if the selected input-output pairs are weakly informative, the resulting abstracted optimization problem introduces severe errors in quality as well as bad runtime performance. NOTE: In other words, we want to approximate the functional dependencies as good as possible with a preferably low number of input-output pairs. This problem is reminiscent of the task of selecting the next most informative input for supervised learning algorithms in case labeled input is rare. We therefore propose to apply methods from active learning based on decision trees for regression to search for informative input candidates to sample and present preliminary results that motivate further research.**

Keywords*—*Active Learning, Abstraction, Optimization, Holarchies, Hierarchical System, Decision Trees

I. Hierarchical Distributed Energy Management

Future energy systems move from systems of relatively few centrally organized units providing most of the power demanded by consumers to many highly distributed units calling for manageable control mechanisms [1]. To deal with the resulting complexity in scheduling and controlling power plants in the face of uncertainties introduced by nature and technical deficiencies, hierarchical organizations based on virtual power plants that form autonomously can be employed [2], [3]. In our vision of future energy management systems [2], inner nodes of the hierarchy are called *autonomous* virtual power plants (AVPP) and act as intermediaries on behalf of their subordinate agents **TODO:** each prosumer and AVPP is modeled as an agent. Prosumers **TODO:** define "prosumer" are thus structured into systems of systems represented by AVPPs, which can themselves be part of other AVPPs, as shown in Fig. 1. To achieve a reduction of complexity in the optimization problem to be solved by the overall system when creating schedules for dispatchable power plants, techniques are borrowed from model abstraction [4]. In particular, functional dependencies over a combinatorial input domain stemming from the aggregate of underlying agents are approximated by repeatedly sampling input-output pairs and substituting the actual functions by piecewise linear functions [5].

In general, the problem to be solved constitutes a hierarchical resource allocation problem [6], where the resource to be allocated to a set of agents maps to their scheduled contributions in order to meet a predicted demand over a scheduling window $\mathcal{W}$ consisting of finitely many time steps with a fixed resolution of typically 15 minutes. Agents have to act proactively, i.e., create schedules, since they are subject to inertia and cannot be assumed to react fast enough in case of rapidly increasing (or decreasing) demand. We derive the minimal set of constraints from the physical requirements that power plants impose (see [7] for a discussion of the literature): i) a minimal and maximal power boundary, ii) discontinuity given the ability to be switched off, and iii) functions limiting the possible change in production over a certain period of time. The latter function might not only depend on the type of an agent but also on its current contribution. From these physical constraints, we abstract minimal and maximal contributions and switching on and off to a sorted list of *feasible intervals* L_a^t. A power plant a that is capable of being switched off or running between some boundaries $P_{\min}$ and $P_{\max}$ would then, for instance, be represented by $L_a^t = \langle [0,0], [P_{\min}, P_{\max}] \rangle$. To allow planning for inertia in a, we introduce functions $\overrightarrow{A}_a^{\min}$ and $\overrightarrow{A}_a^{\max}$ that return the minimum and maximum contribution in a following time step given the current contribution. In the simplest case—we consider a constant maximal change ΔP—these functions are defined as:

$$\overrightarrow{A}_a^{\min}(x) \stackrel{\text{def}}{=} \max\{P_{\min}, x - \Delta P\}$$
$$\overrightarrow{A}_a^{\max}(x) \stackrel{\text{def}}{=} \min\{P_{\max}, x + \Delta P\}$$

But of course, these functions can model richer systems than that, e.g., consider a hot or cold start-up [7], or depend on the current contribution as well as rates of change that map combinatorially to the underlying agents [5]. In addition to that, cost functions κ_a return the minimal costs incurred for a certain contribution.

We present the scheduling problem for some inner node—called intermediary λ—since the problem is solved top-down, as shown in Fig. 1. Each intermediary in turn redistributes its assigned fraction of the overall demand $S_\lambda[t]$ to its subordinate agents $\mathcal{A}_\lambda$ until all leaf agents, i.e., physical power plants, are assigned schedules. Note that the root node Λ is assigned the

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

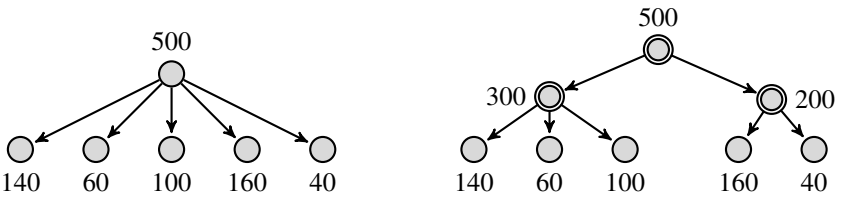


Fig. 1: Resource allocation problems can be solved using a hierarchical decomposition structure. Inner nodes representing intermediaries are marked by double circles.

actual total demand of the environment, i.e., $S_\Lambda[t] = A_{env}[t]$.

$$\begin{aligned} \underset{S_a[t]}{\text{minimize}} \quad & \alpha_\Delta \cdot \Delta + \alpha_\Gamma \cdot \Gamma \qquad (1) \\ \text{subject to} \quad & \forall a \in \mathcal{A}_\lambda, \forall t \in \mathcal{W} : \\ & \exists [x, y] \in L_a^t : x \leq S_a[t] \leq y, \\ & \overrightarrow{A}_a^{\min}(S_a[t-1]) \leq S_a[t] \leq \overrightarrow{A}_a^{\max}(S_a[t-1]) \\ & \text{with } \Delta = \textstyle\sum_{t \in \mathcal{W}} \left| S_{\mathcal{A}_\lambda}[t] - S_\lambda[t] \right|, \\ & \text{and } \Gamma = \textstyle\sum_{t \in \mathcal{W}} \Gamma[t] = \sum_{a \in \mathcal{A}_\lambda} \kappa_a(S_a[t]) \end{aligned}$$

We propose to solve this problem using two approaches based on self-organizing hierarchies for problem decomposition:

- A so-called "regio-central" approach: agents transfer models to their local supervisor who, at meso-level, centrally optimizes the allocation [5], [7]
- An auction-based decentralized approach [2] where agents need not submit their model but only bid on a given demand based on their (possibly private) individual capabilities.

Obtaining a good abstraction of an intermediary's behavior as a compact representation of the underlying subordinate agents' combined behavior is desirable for both algorithms. In the regional-central case, one wants to simplify the resulting optimization problems, in particular by reducing the decision variables and constraints to be considered. In the auction-based algorithm, an intermediary could, in principle, have all agents bid simultaneously to a single auctioneer, i.e., use a super-flat hierarchy. Clearly, this auctioneer imposes a bottleneck with a rising number of agents. When installing a hierarchical organization for scalability, an intermediary ought to be aware of the physical boundaries of its subordinate agents *before* submitting bids in order to avoid inconsistencies that need to be (monetarily) punished by the organization [2]. As a simple illustration, consider that the fact that an intermediary better not bid for a contribution greater than 200 if it is comprised of two underlying physical agents with a maximal contribution of 100 each. Even if the mere summation of maximal boundaries constitutes no computational effort, the possible trajectories resulting from limited rates of change and disconnectability make the problem harder. In fact, an intermediary needs to solve an optimization problem quite similar to Eq. 1 in order to calculate bids for a given demand. Similarly to the "regio-central" approach, abstraction has further to be applied to obtain a simplified model of the intermediary that its own superior may use to compute bids.

II. Obtaining Abstracted Models

As we have now established the need for effective abstraction techniques, we briefly revisit our existing approach to discuss improvements using active learning.

A. Interval Calculations and Sampling Abstraction

A rather straightforward and first essential abstraction in this problem is obtained by considering the possible contributions of an intermediary by combining their lists of feasible intervals. Assume agent a_1 can contribute in $[x_1, y_1]$ and agent 2 in $[x_2, y_2]$. Clearly, their combined contribution must be in $[x_1, y_1] \oplus [x_2, y_2] = [x_1 + x_2, y_1 + y_2]$. The operator $\oplus$ naturally extends to lists of intervals by combining the intervals in a Cartesian fashion and merging the overlapping intervals. That way, one will find that the combined production of two agents a_1, a_2 with $L^{a_1} = \langle [0,0], [1,4] \rangle$ and $L^{a_2} = \langle [0,0], [7,10] \rangle$ will be either of $\langle [0,0], [1,4], [7,14] \rangle$.

But now consider questions such as "What is the minimal cost for an aggregate to contribute x?" or "What is the maximally possible next contribution given the momentary state is y"? Both involve solving an optimization problem since there can naturally be many configurations of subordinate agents leading to a contribution of x with different costs. To address these issues, we acquire an abstract representation of functional relationships by *sampling*, i.e., solving several optimization problems and collecting input and output values (e.g., contribution-cost pairs). Underlying constraints C and decision variables (X, D) of the sampling problems are kept identical of the optimization problem in Eq. 1. Consequently, we can give the algorithm leading to the sampling points:

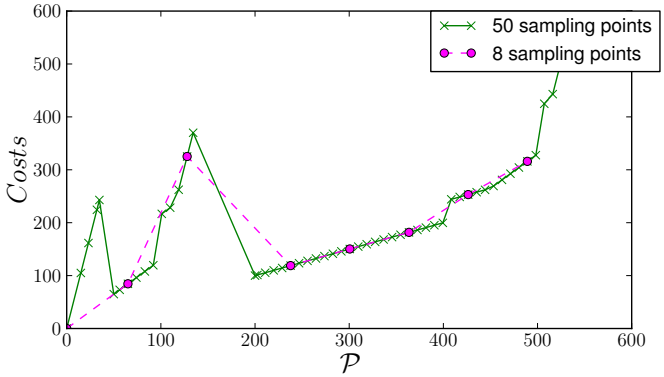
Algorithm 1 Sampling Abstraction for cost function κ_λ

Require: (X, D, C) are the constraints for an intermediary λ
Ensure: γ are contribution-cost pairs

```
I ← s sampling points ∈ L^λ                    ▷ domain of κ_λ
procedure SAMPLING-ABSTRACTION(λ, I)
    for all {i ∈ I} do
        C' ← C ∪ {(S_λ[0] = i)}
        o ← solve ⟨X, D, C'⟩ : minimize Γ[0]
        γ ← γ ∪ {(i, o)}
    return κ_λ ← pwLinear(γ)
```

B. Issues with Sampling Abstraction

However, if the selected input-output pairs are selected in a weakly informative way, the resulting abstracted optimization



(a) Accuracy affected by the number of sampling points selected.

(b) A probabilistic regression model allows to quantify uncertainty at given points in the domain of a learned function.

Fig. 2: Selecting the "right sampling points" is crucial for good accuracy.

problem introduces severe errors in quality as well as bad runtime performance. The former problem is understandable due to unfortunate interpolations, the latter problem is less obvious but stems from the way MIP-solvers resolve piecewise linear functions which might result in many integral decision variables that slow down the optimization. In addition, each sampling point comes at the expense of solving a modified optimization problem. Thus, it is desirable to avoid asking for uninformative points such as those ranging between 200 and 400 in Fig. 2a.

We illustrate the issue with an example of an AVPP λ consisting of three agents a, b, and c:

a: $L^a = \langle [0,0], [50,100] \rangle$, $\kappa^a = 13$
b: $L^b = \langle [0,0], [15,35] \rangle$, $\kappa^b = 70$
c: $L^c = \langle [0,0], [200,400] \rangle$, $\kappa^c = 5$

where κ^i is the price per production unit such that the agents' cost functions are defined by: $\kappa_i(x) = \kappa^i \cdot x$. We derive the feasible contribution ranges for λ and get $L^v = \langle [0,0], [15,35], [50,135], [200,535] \rangle$. Sampling points are selected from this contribution range equidistantly and generally it holds that the more sampling points, the higher the accuracy, as Fig. 2a shows.

III. Improving Sampling Point Selection by Active Learning

Just some very preliminary pointers to literature

Active learning using Gaussian process has been applied to several problems [8], [9], [10].

IV. Evaluation

We investigate the effects of selecting a particular set of sampling points for one group that could have emerged as part of a self-organization process.

References

[1] S. D. Ramchurn, P. Vytelingum, A. Rogers, and N. R. Jennings, "Putting the 'Smarts' Into the Smart Grid: A Grand Challenge for Artificial Intelligence," *Commun. ACM*, vol. 55, no. 4, Apr. 2012.

[2] G. Anders, A. Schiendorfer, F. Siefert, J.-P. Steghöfer, and W. Reif, "Cooperative Resource Allocation in Open Systems of Systems," *ACM Trans. Auton. Adapt. Syst.*, 2015.

[3] A. Nieße, S. Beer, J. Bremer, C. Hinrichs, O. Lunsdorf, and M. Sonnenschein, "Conjoint Dynamic Aggregation and Scheduling Methods for Dynamic Virtual Power Plants," in *Proc. 3rd Int. Wsh. Smart Energy Networks & Multi-Agent Systems (SEN-MAS'14)*, 2014, pp. 1505–1514.

[4] F. Frantz, "A Taxonomy of Model Abstraction Techniques," in *Simulation Conference Proceedings, 1995. Winter*, 1995, pp. 1413–1420.

[5] A. Schiendorfer, J.-P. Steghöfer, and W. Reif, "Synthesis and Abstraction of Constraint Models for Hierarchical Resource Allocation Problems," in *Proc. 6th Int. Conf. Agents and Artificial Intelligence (ICAART'14), Vol. 2*. SciTePress, 2014, pp. 15–27.

[6] T. Van Zandt, "Hierarchical Computation of the Resource Allocation Problem," *European Economic Review*, vol. 39, no. 3-4, pp. 700–708, April 1995.

[7] A. Schiendorfer, J.-P. Steghöfer, and W. Reif, "Synthesised Constraint Models for Distributed Energy Management," in *Proc. 3rd Int. Wsh. Smart Energy Networks & Multi-Agent Systems (SEN-MAS'14)*, 2014, pp. 1529 – 1538.

[8] A. Krause and C. Guestrin, "Nonmyopic Active Learning of Gaussian Processes: An Exploration-Exploitation Approach," in *Proc. 24th Int. Conf. Machine Learning (ICML'07)*. ACM, 2007, pp. 449–456.

[9] A. Krause, A. Singh, and C. Guestrin, "Near-Optimal Sensor Placements in Gaussian Processes: Theory, Efficient Algorithms and Empirical Studies," *J. Mach. Learn. Res.*, vol. 9, pp. 235–284, Jun. 2008.

[10] M. Park, G. Horwitz, and J. W. Pillow, "Active learning of neural response functions with Gaussian processes," in *Advances in Neural Information Processing Systems (NIPS)*, J. Shawe-Taylor, R. Zemel, P. Bartlett, F. Pereira, and K. Weinberger, Eds. Curran Associates, Inc., 2011, pp. 2043–2051.