Noname manuscript No.
(will be inserted by the editor)

Holarchic Structures for Decentralized Deep Learning – A Performance Analysis

Evangelos Pournaras · Srivatsan Yadhunathan · Ada Diaconescu

Received: date / Accepted: date

arXiv:1805.02686v1 [cs.LG] 7 May 2018

Abstract Structure plays a key role in learning performance. In centralized computational systems, hyperparameter optimization and regularization techniques such as dropout are computational means to enhance learning performance by adjusting the deep hierarchical structure. However, in decentralized deep learning by the Internet of Things, the structure is an actual network of autonomous interconnected devices such as smart phones that interact via complex network protocols. Adjustments in the learning structure are a challenge. Uncertainties such as network latency, node and link failures or even bottlenecks by limited processing capacity and energy availability can significantly downgrade learning performance. Network self-organization and self-management is complex, while it requires additional computational and network resources that hinder the feasibility of decentralized deep learning. In contrast, this paper introduces reusable holarchic learning structures for exploring, mitigating and boosting learning performance in distributed environments with uncertainties. A large-scale performance analysis with 864000 experiments fed with synthetic and real-world data from smart grid and smart city pilot projects confirm the cost-effectiveness of holarchic structures for decentralized deep learning.

Keywords deep learning · holarchy · Internet of Things · dropout · multi-agent system · distributed system · network · Smart City · Smart Grid

Evangelos Pournaras · Srivatsan Yadhunathan
Professorship of Computational Social Science
ETH Zurich, Zurich, Switzerland
Tel.: +41446320458
E-mail: {epournaras,ysrivatsan}@ethz.ch

Ada Diaconescu
Telecom ParisTech
Paris-Saclay University, Paris, France
Tel.: +33145818072
E-mail: ada.diaconescu@telecom-paristech.fr

1 Introduction

Smart citizens' devices with increasing processing power and high energy autonomy are becoming pervasive and ubiquitous in everyday life. The Internet of Things empowers a high level of interconnectivity between smart phones, sensors and wearable devices. These technological developments provide unprecedented opportunities to rethink about the future of machine learning and artificial intelligence: Centralized computational intelligence can be often used for privacy-intrusive and discriminatory services that create 'filter bubbles' and undermine citizens' autonomy by nudging [11,27,15]. In contrast, this paper envisions a more socially responsible design for digital society based on decentralized learning and collective intelligence formed by bottom-up planetary-scale networks run by citizens [17,16].

In this context, the structural elements of decentralized deep learning processes play a key role. The effectiveness of several classification and prediction operations often relies heavily on hyperparameter optimization [24,46] and on the learning structure, for instance, the number of layers in a neural network, the interconnectivity of the neurons, the activation or deactivation of certain pathways i.e. dropout regularization [44], can enhance learning performance. Controlling and adjusting a deep hierarchical structure in a centralized computing system is straightforward in the sense that all meta information for model generation is locally available. However, in decentralized learning the challenge of optimizing the learning structure is not anymore exclusively a computational problem. Other challenges such as network latency, node and link failures as well as the overall complexity of building and maintaining an overlay network in a distributed environment perplex the feasibility of decentralized learning.

This paper introduces the concept of *holarchy* in deep hierarchical structures as the means to cope with the aforementioned uncertainties of distributed environments. A learning process can be performed over a holarchic structure in a recursive way without changing the core learning logic and without employing additional mechanisms to reconfigure the network. Forward propagation and backpropagation become recursive in nested levels over the deep hierarchical structure as the means to (i) *explore* improving solutions, (ii) *mitigate* learning performance in case part of the network is disconnected or even (iii) *boost* learning performance after the default learning process completes. These three scenarios are formalized by three holarchic schemes applied to a multi-agent system for decentralized deep learning in combinatorial optimization problems: I-EPOS, the *Iterative Economic Planning and Optimized Selections* [30].

A large-scale performance analysis with 864000 experiments is performed using synthetic and real-world data from pilot projects such as bike sharing, energy demand and electrical vehicles. Several dimensions are studied, for instance, topological properties of the deep hierarchical structure, constraints by the agents' preferences and the scale of the holarchic structures, i.e. number of nested layers. Results confirm the cost-effectiveness of the holarchic learning

schemes for exploration, mitigation and boosting of learning performance in dynamic distributed environments. Nevertheless, in stable environments the localization of the learning process by holarchic structures may result in lower performance compared to a system-wide learning.

In summary, the contributions of this paper are the following:

- The concept of holarchy in decentralized deep learning to manage the uncertainties of distributed environments.
- The introduction of three holarchic schemes as the means to explore, mitigate and boost deep learning performance.
- The applicability and extension of I-EPOS with the three holarchic schemes to perform collective decision-making in decentralized combinatorial optimization problems.
- An empirical performance analysis of 864000 benchmark experiments generated with synthetic and real-world data from Smart Grid and Smart City pilot projects.

This paper is organized as follows: Section 2 introduces the concept of holarchy in decentralized learning as well as three holarchic schemes to explore, mitigate and boost deep learning performance under uncertainties of distributed environments. Section 3 illustrates a case study of a decentralized deep learning system to which the three holarchic schemes are applied: I-EPOS. Section 4 shows the experimental methodology followed to conduct a large-scale performance analysis of the three holarchic schemes. Section 5 illustrates the results of the experimental evaluation. Section 6 illustrates the related work of this paper. Finally, Section 7 concludes this paper and outlines future work.

2 Holarchic Structures for Decentralized Deep Learning

This paper studies *decentralized deep learning* processes in which the deep hierarchical structure is fully distributed and self-organized by remote autonomous (software) agents that interact over a communication network. In other words, the learning process is crowd-sourced to citizens, who participate by contributing their computational resources, for instance their personal computers or smart phones. The agents reside on these devices and collectively run the decentralized learning process. For example, in contrast to a conventional model of a centralized neural network, which performs training by locally accessing all citizens' data, a collective neural network consists of neurons that are remote citizens' devices interacting over the Internet to form an overlay peer-to-peer network [31]. It is this overlay network that represents the hierarchical neural network structure.

Methods for hyperparameter optimization of the hierarchical structure to improve the learning performance of a model are usually designed for centralized systems in which all information, including input data, the network structure and the learning model itself are locally available. This provides a

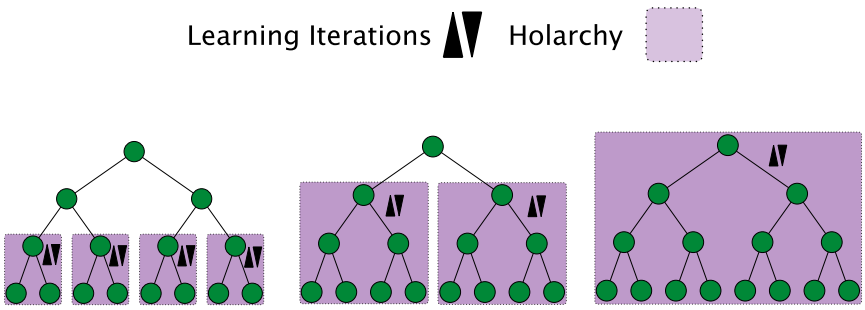
large spectrum of flexibility to change the deep hierarchical structure offline or even online [14,36] and determine via hyperparameter optimization the settings that maximize the learning performance. In contrast to learning based on centralized computational systems, in decentralized deep learning the structure cannot arbitrary change without paying for some computational and communication cost. Network uncertainties such as node and link failures, latency as well as limited resources in terms of bandwidth, computational capacity or even energy in case of sensors and smart phones can limit performance, interrupt the learning process and increase the design complexity of decentralized hyperparameter optimization.

The aforementioned uncertainties of distributed environments introduce endogenous constraints in parts of the deep hierarchical structure: the learning process is interrupted and becomes *localized* within branches of the hierarchical structure. For instance, node and link failures or suspension of the learning processes due to conservation of resources [2,45] in nodes are scenarios under which learning can be localized. On the one hand, it is known that localization and greedy optimization can underperform with search becoming trapped to locally optimum solutions that have a significant divergence from global optimality [3]. On the other hand, limiting the learning units of hierarchical structures can also increase learning performance by preventing overfitting as known by the dropout concept in neural networks [44]. This paper sheds light on the role of localization in decentralized learning.

In this context, the management of the learning performance of an algorithm is not anymore entirely a computational challenge but rather a multifaceted process: Other aspects such as performance *exploration*, *mitigation* and *boosting* come to the analysis foreground. Exploration adapts the learning process and the search space to improve learning performance under localization. Mitigation is the maintenance of a high learning performance under a localization of the learning process. Finally, the feasibility of boosting the learning performance under localization is subject of this study as well.

This paper introduces the concept of *holarchy* in deep learning hierarchical structures to study the performance exploration, mitigation and boosting potential under the aforementioned uncertainties of distributed environments. A holarchy is a recursive hierarchical network of *holons* that represent part of the deep hierarchical structure as well as the whole structure. In the case of a tree topology, every possible branch (part) in the whole tree topology is also a tree topology (whole). When a node (parent) connects two branches, it forms another nested holon that is the next level of the holarchic structure. This recursive process starts from the parents of the leaves in the tree and progresses up to the root as shown in Figure 1. Learning iterations can be independently executed in every holonic branch before the process progresses to the next level of the holarchic structure, in which a new series of learning iterations are executed. The top holonic branch is actually the whole tree topology and therefore the execution of learning iterations at this top level corresponds to the learning iterations without a holarchic structure. In other words, the con-

cept of holarchy introduces multiple localized, nested and incremental learning processes.



(a) Learning iterations between the leaves and their parents.

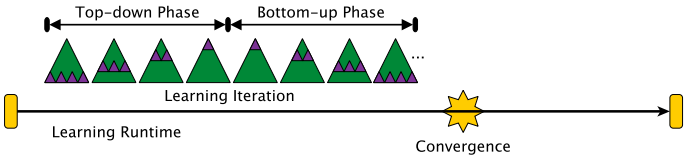
(b) Progress to the next level. Learning iterations are performed within tree branches.

(c) Holarchic learning completes with learning iterations performed over the whole tree structure.

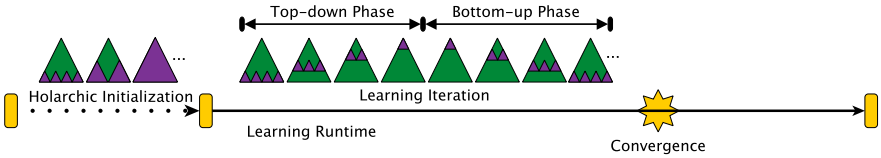
Fig. 1 The concept of holarchic learning. Learning iterations are performed in nested branches, the holons. Figure 1c actually depicts the default baseline learning strategy, while Figure 1a and 1b show the earlier learning iterations performed within the holons.

This paper studies three applicability scenarios of the holarchic concept in decentralized deep learning each designed for performance exploration, mitigation and boosting respectively: (i) *holarchic initialization*, (ii) *holarchic runtime* and (iii) *holarchic termination*. Assume a *baseline* scheme that involves a tree structure with nodes interacting in a (i) *bottom-up phase* and a (ii) *top-down phase* that both complete a *learning iteration*. The former phase may represent a fit forward learning process starting from the leaves and completing to the root while the latter phase a back propagation starting from the root and reaching back the leaves. Without loss of generality, an exact decentralized learning algorithm realizing these concepts is presented in Section 3. Learning iterations repeat to decrease a *cost function*. Learning *converges* when a certain number of iterations is performed or when the cost function cannot be decreased further. Figure 2a illustrates the baseline scheme.

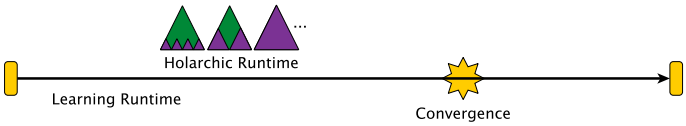
Figure 1c depicts one baseline learning iteration, while within each nested holon formed during the bottom-up phase several learning iterations are performed. This process is common in all holarchic schemes. Holarchic initialization is applied before baseline to perform an exploration of the search space. Several learning iterations can be performed before switching to baseline as illustrated in Figure 2b. In contrast, holarchic runtime applies holarchic learning throughout runtime without switching to baseline as shown in Figure 2c. This scheme is applicable is scenarios of failures or conservation of resources to mitigate losses of learning performance. Finally, holarchic termination is ap-



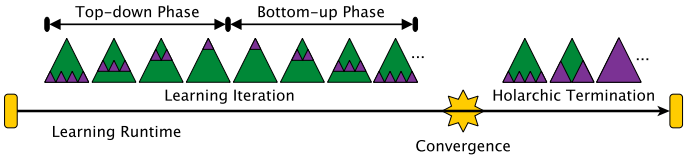
(a) Baseline: Throughout runtime a fixed number of learning iterations is performed during which convergence is potentially achieved. The bottom-up phase starts from the leaves and progresses level-by-level up to the root, while the reverse process (back propagation) is performed in the top-down phase. In both phases, the parent-children interactions always progress to the next level, in contrast to the holarchic strategies in which learning iterations are performed in nested tree branches as shown below.



(b) Holarchic initialization: Before the baseline execution, holarchic learning is performed.



(c) Holarchic runtime: Holarchic learning is performed throughout the runtime. Note that one learning iteration corresponds to multiple holarchic iterations performed in each holon.



(d) Holarchic termination: Holarchic learning is activated after convergence is reached with baseline to potentially discover improved solutions.

Fig. 2 The four learning schemes studied and compared in this paper.

plied after the baseline convergence as the means to further boost the baseline performance. This scheme is shown in Figure 2d.

3 Applicability of Holarchic Schemes

This section illustrates a case study for the applicability of holarchic structures in decentralized deep learning for combinatorial optimization problems: I-EPOS[1], the *Iterative Economic Planning and Optimized Selections* [30]. I-EPOS consists of agents that autonomously plan resources they consume and produce. Planning is a process of *resource scheduling* or *resource allocation*. For instance agents may represent smart phone apps (personal assistants), cyber-physical controllers or smart home information systems with the capability to plan the energy consumption of residential appliances, the charging of electrical vehicles or the choices of bike-sharing stations. Planning serves the local resource requirements of users as well as system-wide objectives, for instance, the decrease of demand oscillations in the whole power grid to prevent blackouts [34], the synchronization of power demand with the availability of renewables [1] or the load-balancing of bike-sharing stations to decrease the operational costs of manual bike relocations [43,30].

I-EPOS introduces the computational model of locally generated *possible plans* that represent users' operational flexibility on how resources can be consumed or produced. For instance, a user may turn on a laundry machine earlier or later in time, can choose among two or more stations to return a shared bike, etc. Computationally, plans are vectors of real values and agents need to collectively choose one and only one of these plans to execute so that the summation of all these selected plans satisfies a system-wide objective measured by a *global cost* function. This paper focuses on the computational problem of the variance minimization that is a quadrtic cost function, which cannot be locally minimized: coordination and collective decision-making is required [37]. Choosing the optimum combination of plans, whose summation minimizes a quadratic cost function is a combinatorial optimization problem known to be NP-hard [35].

Among the global cost that captures system-wide objectives, agents can also assess a *local cost* of their possible plans that may represent a notion of *discomfort or inconvenience* [33]. For instance, the longer the time period a user shifts the energy consumption earlier or later in time to prevent a power peak (global cost reduction) the higher the level of disruption is in the regular residential activities of a user (local cost increase). The agents' choices can be autonomously parameterized to increase or decrease the priority of minimizing the local cost over the global cost. This trade-off is regulated by the λ parameter for each agent. A $\lambda = 0$ results in choices that exclusively minimize the global cost, i.e. variance, and ignores the local cost. In contrast, a $\lambda > 0$ biases the agents' choices to favor plans with a lower local cost.

Learning is performed as follows: agents self-organize [31] in a tree network topology over which collective decision-making is performed – a plan is chosen by taking into account the following aggregate information (element-wise summation of vectors): (i) the aggregate plan choices of the agents in the branch

[1] Available at `epos-net.org` (last accessed: April 2018)

underneath made available during the bottom-up phase and (ii) the aggregate plan choices of all agents at the previous learning iteration made available during the top-down phase. Note that decision-making remains highly localized and decentralized as the planning information of the other agents is always at an aggregate level, i.e. the possible plans of other agents are not explicitly required.

Several of the following factors influence the learning performance defined by the level of the global cost reduction and the number of learning iterations required to minimize variance: (i) the positioning of the agents in the tree that determines the order of the collective choices made, (ii) the λ parameter that regulates the trade-off of global vs. local cost and (iii) the overall topological structure and specifically in this paper the number of children c in balanced trees is studied.

Improving the learning performance by repositioning the agents in the tree or adapting the topology is complex and costly in distributed environments as the aforementioned self-organization methodologies are based on supplementary distributed protocols that consume resources, i.e. exchange of messages, computational power, energy, etc. Moreover, any node or link failure limits the computation of the aggregate plans at a branch level and therefore the collective decision-making cannot be anymore performed over all participating agents. The applicability of the holarchic schemes does not require any change in the logic of I-EPOS. The algorithm is localized and applied within multiple nested and connected branches even when the network becomes disconnected due to node and link failures. The fact that a number of agents is isolated and does not participate in the learning process is the means to traverse the optimization space via alternative pathways with the potential to explore improving solutions, mitigate performance loss compared to a total interruption of I-EPOS, or even boost the reduction of the global cost that cannot be decreased anymore with I-EPOS.

The rest of this paper illustrates an empirical performance analysis of the three holarchic schemes and their applicability on I-EPOS.

4 Experimental Methodology

This section illustrates the varying dimensions in the experiments and the experimental settings. It also illustrates the evaluation metrics used to assess the holarchic learning schemes.

4.1 Varying dimensions and performed experiments

The following system dimensions are studied: (i) *application scenarios*, (ii) *holarchic schemes*, (iii) *scale of holarchy*, (iv) *number of children c in the tree network*, (v) *different agent preferences* λ.

Synthetic and empirical plans are generated for 1000 agents using data from real-world pilot projects. These four scenarios are referred to as follows: (i) *synthetic*, (ii) *bike sharing*, (iii) *energy demand* and (iv) *electrical vehicles*.

The synthetic dataset consists of 16 possible plans of size 100 generated from a standard normal distribution with a mean of 0 and a standard deviation of 1. A random local cost is assigned to the plans.

The bike sharing dataset[2] of the Hubway bike sharing system in Paris is used to generate a varying number of plans of size 98 for each agent based on the unique historic trips performed by each user. Therefore, the plans represent the trip profiles of the users and they contain the number of incoming/outgoing bike changes made by each user in every station [32]. The local cost of each plan is defined by the likelihood of a user to not perform a trip instructed in the plan [30]. For instance, if three plans are chosen 4, 5 and 1 days during the measured time period respectively, the local cost for these plans is 0.6, 0.5 and 0.9 respectively.

The energy demand dataset of the Pacific Northwest Smart Grid Demonstration Project (PNW) by Battelle[3] is used to generate 10 plans of size 144 containing electricity consumption records for every 5 minutes during 01:00-13:00 on 23.07.2014. Planning is performed by shifting the energy demand $5, 10, \ldots, 30$ minutes into the past or into the future. The local cost of each plan is the minutes of the shifted demand compared to the original one [30].

The plans for the electrical vehicles are generated using data[4] from the California Department of Transportation's California Household Travel Survey for 2010–2012. The plans concern the energy consumption of the electrical vehicles by charging from the power grid. Four plans per agent with size 1440 are generated by extracting the vehicle utilization using the historical data and then computing the state of charge by redistributing the charging times over different time slots. The methodology is outlined in detail in earlier work [32]. The local cost of each plan is measured by the likelihood of the vehicle utilization during the selected charging times.

The learning schemes studied are the *baseline* that is I-EPOS and the three holarchic schemes: *Holarchic initialization* is used as an *exploration strategy* to evaluate its likelihood to improve the learning capacity. *Holarchic runtime* is used as a *mitigation strategy* to evaluate the maintenance of learning capacity in distributed environments under uncertainties. Finally, *holarchic termination* is used as a *boosting strategy* to evaluate the likelihood of improving the learn-

[2] The dataset is made available in the context of the Hubway Data Visualization Challenge: `http://hubwaydatachallenge.org/` (last accessed: April 2018). Although this dataset does not contain personalized records, user trips are extracted from user information: zip-code, year of birth and gender. All trips that have common values in these fields are assumed to be made by the same user. The timeslot is chosen from 8:00 am to 10:00 am. All historic unique trips a user did in the defined timeslot of a week day are considered as the possible plans for that day.

[3] Available upon request at `http://www.pnwsmartgrid.org/participants.asp` (last accessed: April 2018)

[4] Available at `www.nrel.gov/tsdc` (last accessed: April 2018). Electrical vehicles equipped with GPS are selected.

ing capacity. In each holon of the holarchic schemes within one main learning iteration, $\tau = 5$ holarchic iterations are executed.

Two holarchic scales are evaluated: (i) *full* and (ii) *partial*. The full scale uses all levels of the baseline tree network to apply a holarchic scheme as also shown in Figure 1. In contrast, partial scale is applied in one branch under the root.

The influence of the topological tree structure and agent preferences on the learning capacity is studied by varying respectively the number of children as $c = 2, .., 5$ and the λ parameter as $\lambda = 0, 0.25, 0.5, 0.75$. Table 1 summarizes the dimensions and their elements in the performed experiments.

Table 1 Dimensions and their elements in the total of 864000 experiments.

Dimension	**Elements**	**Element 1**	**Element 2**	**Element 3**	**Element 4**
Application scenario	4	Synthetic	Bike sharing	Energy demand	Electrical vehicles
Learning scheme	4	Baseline	Holarchic initialization	Holarchic runtime	Holarchic termination
Holarchic scale	2	Full	Partial	-	-
Number of children (c)	4	$c = 2$	$c = 3$	$c = 4$	$c = 5$
Agent preferences (λ)	4	$\lambda = 0$	$\lambda = 0.25$	$\lambda = 0.5$	$\lambda = 0.75$
Total:	**864000**				

The total experiments performed are calculated as follows: For the dimension of application scenarios, 4 elements are counted and 3 elements for the learning schemes given that the baseline and the holarchic termination can be generated within one experiment. The partial scale uses all possible branch combinations: $2 + 3 + 4 + 5 = 14$ elements plus 4 elements for the full scale result in 18 total elements for the two dimensions of holarchic scale and number of children. Finally, 4 elements are counted for the dimension of agent preferences. The total number $4 * 3 * 18 * 4 = 864$ of element combinations is the total number of experiments performed. Each experimental combination is repeated 1000 times by (i) 1000 samples of possible plans in the synthetic scenario and (ii) 1000 random assignments of the agents in the tree network in the other three application scenarios. Therefore, the total number of experiments performed is $864 * 1000 = 864000$.

4.2 Evaluation metrics

Performance is evaluated with the following metrics: (i) *standardized global cost*, (ii) *improvement index* and (iii) *communication cost*.

The standardized global cost is the variance of the global plan at convergence time. The minimization of the variance is the optimization objective and therefore the variance is used as the criterion of the learning capacity. Standardization[5] is applied on the variance so that the learning capacity among different datasets can be compared.

The improvement index I measures the reduction or increase of the global cost at convergence time for the holarchic schemes compared to the baseline. Positive values indicate an improvement of the baseline, while negative values show a deterioration. The improvement index is measured as follows:

$$I = \frac{C_{\mathsf{b}} - C_{\mathsf{h}}}{C_{\mathsf{b}} + C_{\mathsf{h}}} \tag{1}$$

where C_{b} is the global cost for the baseline and C_{h} is the global cost for a holarchic scheme, both at convergence time. The improvement index is calculated based on the principle of the symmetric mean absolute error, which compared to the mean absolute error can handle single zero values, it is bound to $[-1, 1]$ and eliminates very large values originated by low denominators [19].

The communication cost measures the number of messages exchanged to complete a learning iteration and can be distinguished to *total* and *synchronized*. The total communication cost counts all exchanged messages between the nodes during a learning iteration and it is calculated for the baseline M_{b} as follows:

$$M_{\mathsf{b}} = 2(c^0 + ... + c^l - 1) \tag{2}$$

where c is the number of children in a balanced tree. Equation 2 sums up the number of nodes in each level of the tree and substracts 1 to count the number of links. Multiplication by 2 counts both bottom-up and top-down phases. The total communication cost of a holarchic scheme can be measured as follows:

$$M_{\mathsf{t}} = 2\tau \sum_{j=0}^{l} c^{l-j}(c^0 + ... + c^j - 1) \tag{3}$$

where c is here again the number of children in a balanced tree/holarchy and τ is the number of holarchic iterations. The summation starts from leaves (level $j = 0$), and progresses to the root of the holarchy (at level $j = l$). Equation 3 multiplies the number of nodes c^{l-j} at each level $l-j$ with 2τ times (τ bottom-up and τ top-down holarchic iterations) the number of nodes in the branches

[5] Standardization transforms the global cost values to have zero mean and units of variance as follows: $\frac{x-\mu}{\sigma}$

underneath: $c^0 + ... + c^j - 1$. For example, $j = 0$ corresponds to Figure 1a with a communication cost of $2\tau 2^2(2^0 + 2^1 - 1) = 16\tau$. $j = 1$ corresponds to Figure 1b with a communication cost of $2\tau 2^1(2^0 + 2^1 + 2^2 - 1) = 24\tau$ and respectively, for $j = 2$ and Figure 1c communication cost is calculated as $2\tau 2^0(2^0 + 2^1 + 2^2 + 2^3 - 1) = 28\tau$. These nested calculations for the full holarchy sum up to $M_{\mathsf{t}} = (16 + 24 + 28)\tau = 68\tau$ messages.

The synchronized communication cost counts the number of messages exchanged within holons, while counting this number only once for holons at the same level which can exchange messages in parallel. For instance, Figure 1a illustrates four parallel holons with a total communication cost of $2\tau 2^2(2^0 + 2^1 - 1) = 16\tau$ messages. Instead, the synchronized communication cost counts for $2\tau(2^0+2^1-1) = 4\tau$ messages. The synchronized communication cost of a full holarchy can be measured as follows:

$$M_{\mathsf{s}} = 2\tau \sum_{j=0}^{l} (c^0 + ... + c^j - 1) \tag{4}$$

where c is the number of children and τ the number of holarchic iterations. Note that the synchronized communication cost considers holons performing in parallel and not individual nodes, that is why within each holon all messages exchanged are counted. For the same reason, the synchronized communication cost for baseline is not defined for a fairer comparison with holarchic schemes.

4.3 Computational challenge

To better understand the computational challenge and context in which the performance of the holarchic schemes is studied, the performance characteristics of the baseline are illustrated in this section. Figure 3 illustrates the learning curves of the baseline in the four application scenarios.

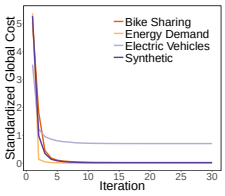
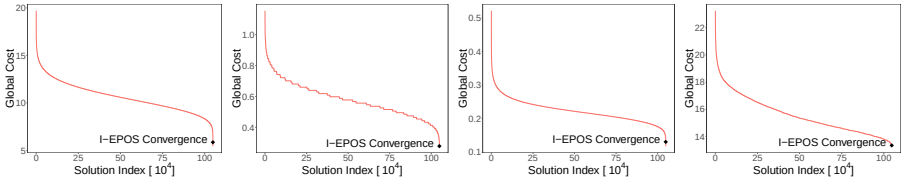


Fig. 3 Learning curves of I-EPOS for the four benchmark application scenarios with 1000 agents generating 4 plans and choosing a plan with $\lambda = 0$.

The learning performance of I-EPOS shows the following behavior: Global cost decreases dramatically in very few iterations in all application scenarios, while the decrease is monotonous. Therefore, I-EPOS has a superior efficiency to learn fast combinations of plans that minimize the variance by executing

10-15 iterations. Convergence does not necessarily mean that the globally optimum solution is found. The evaluation of the global optimality in a system with 1000 agents and more than 4 plans per agents is computationally infeasible given the exponential complexity of the combinatorial space: 4^{1000}. For this reason, the global optimality is evaluated using brute force search in a small-scale system of 10 agents with 4 plans per agent and $\lambda = 0$. Therefore the total number of solutions is 4^{10}. Each experiment is repeated 10 times by shuffling the agents over a binary tree. Figure 4 illustrates the global optimality results for each application scenario.



(a) Synthetic **(b)** Bike sharing **(c)** Energy demand **(d)** Electrical vehicles

Fig. 4 Global optimality of I-EPOS for the four benchmark applicaiton scenarios with 10 agents generating 4 plans and choosing a plan with $\lambda = 0$.

I-EPOS finds the 0.007%, 0%, 0.017% and 0.153% top solution in each of the application scenarios of Figure 4. Designing a new learning scheme to overpass this performance level, without introducing additional complexity and resources is a challenge and potentially not an endeavor worth pursuing. Instead this paper studies holarchic structures as learning strategies to explore, mitigate and boost the cost-effectiveness of I-EPOS in distributed environments and in this sense the notion of holarchy is the means for decentralized learning to tolerate their uncertainties.

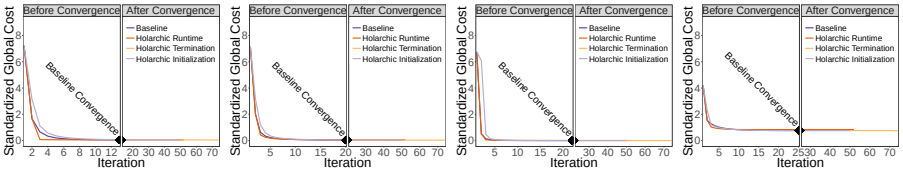
5 Experimental Evaluation

This section illustrates the learning capacity of the three holarchic schemes followed by the trade-offs and cost-effectiveness of the holarchic runtime.

5.1 Learning capacity

Figure 5 illustrates the learning curves for the four application scenarios and learning schemes. The partial scale with $\lambda = 0$ and $c = 2$ is illustrated. The learning curves for the full scale, $\lambda = 0.5$ and $c = 5$ are illustrated in Figure 12 of Appendix A.

The following observations can be made in Figure 5: Holarchic runtime achieves the fastest convergence speed given the several multi-level holarchic

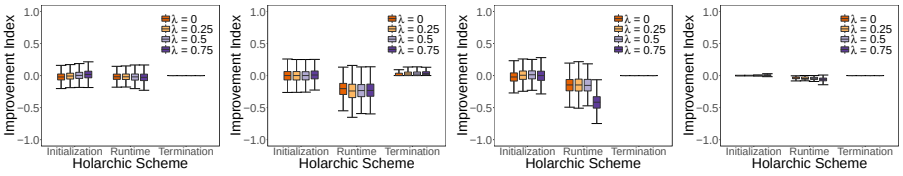


(a) Synthetic **(b)** Bike sharing **(c)** Energy demand **(d)** Electrical vehicles

Fig. 5 Learning curves. *Dimensions*: learning schemes, application scenarios. *Settings*: partial scale, $\lambda = 0$, $c = 2$.

iterations performed within a main learning iteration. However, a sacrifice of global reduction is observed, which is though low and observable for the scenario of electrical vehicles in which the global cost is 10% higher than the baseline. Moreover, within 7-8 iterations all holarchic schemes achieve the global cost reduction of the baseline. The convergence speed of the holarchic initialization is 1-2 iterations slower than the baseline, though this insignificant difference is not anymore observable in the scenario of electrical vehicles.

Figure 6 illustrates the improvement index of the holarchic schemes for different application scenarios and different λ values. The exploration potential of the holarchic initialization is indicated by the error bars above the mean. Moreover, holarchic initialization does not influence significantly the global cost reduction, however, for higher λ values, i.e. $\lambda = 0.75$, an average increase of 0.5% is observed in the improvement index. This means that in more constrained optimization settings, the exploration strategy of the holarchic initialization contributes a low performance improvement. In contrast, the mitigation strategy of the holarchic runtime manages to preserve the baseline performance with an improvement index close to 0 values. An average performance boosting of 1.65% is observed in the bike sharing scenario via the holarchic termination, while in the other scenarios no significant improvement is observed.

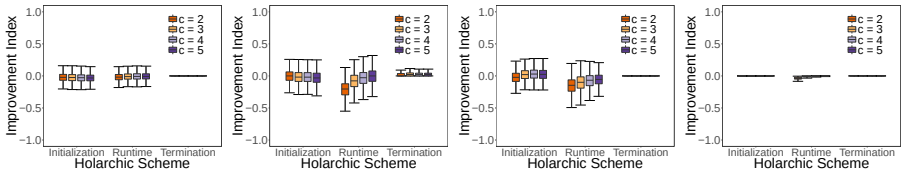


(a) Synthetic **(b)** Bike sharing **(c)** Energy demand **(d)** Electrical vehicles

Fig. 6 Improvement index. *Dimensions*: holarchic schemes, application scenarios, different λ values. *Settings*: partial scale, $c = 2$.

Figure 7 illustrates the improvement index of the holarchic schemes for different application scenarios and different c values. Holarchic initialization

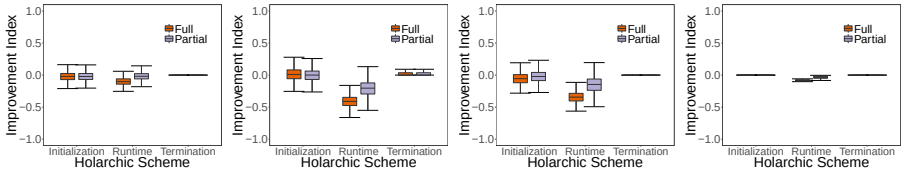
retains an average improvement index of -0.008 while it can scale up the improvement index to values of 0.225 on average. The number of children does not influence the performance of this holarchic scheme. In contrast, the holarchic runtime shows an average increase of 4.1% in the improvement index by increasing c from 2 to 5, while this holarchic scheme serves well its performance mitigation role: an average improvement index of -0.041. Finally, holarchic termination boosts performance by 1.1% in the bike sharing scenario.



(a) Synthetic **(b)** Bike sharing **(c)** Energy demand **(d)** Electrical vehicles

Fig. 7 Improvement index. *Dimensions*: holarchic schemes, application scenarios, varying number of children. *Settings*: partial scale, $\lambda = 0$.

Figure 8 demonstrates the higher performance that the partial scale shows compared to full scale: 0.25% higher improvement index for holarchic initialization and 6.5% for holarchic runtime.



(a) Synthetic **(b)** Bike sharing **(c)** Energy demand **(d)** Electrical vehicles

Fig. 8 Improvement index. *Dimensions*: holarchic schemes, application scenarios, partial vs. full scale. *Settings*: $\lambda = 0$, $c = 2$.

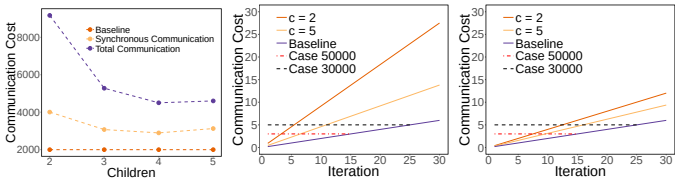
Figure 13, 14 and 15 of Appendix A show the probability density of the improvement index for different λ, c values and holarchic scales respectively. In these figures one can study in more detail the density of the improvement index values behind the error bars of the respective Figure 6, 7 and 8.

In summary, the results on the learning capacity confirm the following insights: (i) The evident computational challenge to contribute improvements in terms of global cost as motivated in Figure 3 and 4. (ii) The capability of the holarchic schemes to perform a performance exploration, mitigation and boosting of decentralized learning designed for distributed environments with uncertainties.

The rest of this section studies the trade-offs and the cost-effectiveness of the holarchic runtime designed for the mitigation of the learning performance.

5.2 Trade-offs and cost-effectiveness

Figure 9a illustrates the communication cost per iteration of the baseline vs. the total and synchronized communication cost of the holarchic runtime. This is a worse case scenario as applying the holarchy to a smaller branch or for a fewer than 5 holarchic iterations can make the communication cost equivalent[6] to the one of the baseline. In Figure 9a, the communication cost of the holarchic runtime decreases as the number of children increases given that the recursion of the holarchy is limited to a lower number of levels in the tree, i.e. fewer holons are formed. The synchronized communication cost is on average 45% lower than the total communication cost.



(a) Total vs. synchronized communication for different c.

(b) Total communication

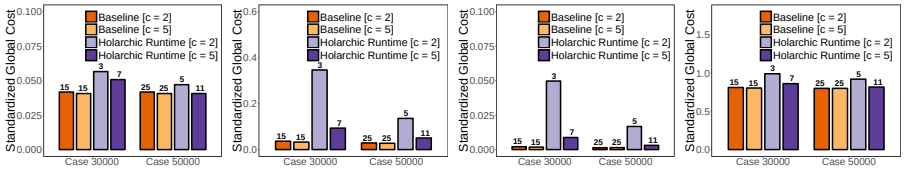
(c) Synchronized communication

Fig. 9 Communication cost. *Settings*: partial scale, $\lambda = 0$ (a) *Dimensions*: varying number of children, total vs. synchronized communication cost, baseline vs. holarchic runtime. (b) and (c) Sampling case 30000 and case 50000 under total vs. synchronized communication cost. *Dimensions*: $c = 2$ vs. $c = 5$, number of iterations, baseline vs. holarchic runtime.

The cost-effectiveness of the holarchic runtime is studied by fixing the communication cost for both baseline and holarching runtime and looking into the global cost reduction achieved for the same number of messages exchanged. Figure 9b and 9c illustrate this process for total and synchronous communication cost. Two cases are determined: (i) *case 30000* and (ii) *case 50000*. Each case runs for a given number of iterations that is determined by the intersection with the horizontal dashed lines for each of the $c = 2$ and $c = 5$. Then, the global costs can be compared for the same number of exchanged messages as shown in Figure 10.

The following observations can be made in Figure 10a to 10d for the total communication cost: When $c = 5$, the holarchic termination achieves a

[6] The communication cost of the holarchic runtime can even become lower than baseline assuming a holarchy at partial scale with the nodes that do not belong to the holarchy being disconnected.

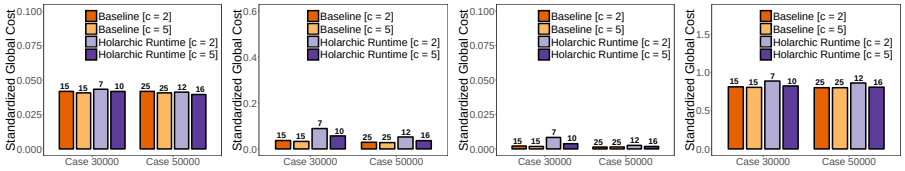


(a) Synthetic, total communication

(b) Bike sharing, total communication

(c) Energy demand, total communication

(d) Electrical vehicles total communication



(e) Synthetic, synchronized communication

(f) Bike sharing, synchronized communication

(g) Energy demand, synchronized communication

(h) Electrical vehicles, synchronized communication

Fig. 10 Global cost. *Dimensions*: baseline vs. holarchic runtime, total vs. synchronized communication cost, application scenarios, $c = 2$ vs. $c = 5$, number of iterations given a communication cost, case 30000 vs. case 50000. *Settings*: partial scale, $\lambda = 0$.

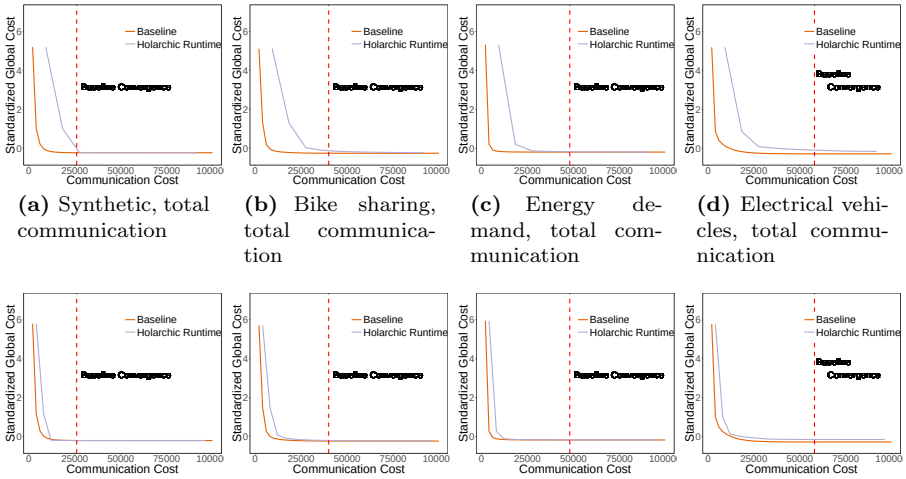
highly equivalent performance with the baseline. This also holds for $c = 2$ in the scenarios of synthetic and electrical vehicles. The performance mitigation becomes 13.64% more significant when the synchronized communication cost is counted in Figure 10e to 10h. This is also shown by the shifted probability densities of the improvement index in Figure 16 of Appendix A.

These findings can be generalized further for the broader range of communication cost as shown in Figure 11 for $c = 2$. The key observation that confirms the mitigation capability of the holarchic termination is the comparable global cost achieved using the communication cost required for baseline to converge. This mitigation potential is also demonstrated by the probability density of the relative global cost between baseline and holarchic runtime in Figure 17 of Appendix A.

Figure 11 can also be compared with Figure 18 of Appendix A that shows the cost-effectiveness under $c = 5$. The performance mitigation is even higher with this setting.

6 Related Work

H. A. Simon [41,42] identifies the key role that hierarchical design plays in dealing with complexity in large-scale, highly dynamic systems. He shows how certain types of hierarchy [8] address the challenge of *limited rationality*, which stems from the combinatorial explosion of alternative system compositions, in



(a) Synthetic, total communication **(b)** Bike sharing, total communication **(c)** Energy demand, total communication **(d)** Electrical vehicles, total communication

(e) Synthetic, synchronized communication **(f)** Bike sharing, synchronized communication **(g)** Energy demand, synchronized communication **(h)** Electrical vehicles, synchronized communication

Fig. 11 Cost-effectiveness. *Dimensions*: baseline vs. holarchic runtime, application scenarios, total vs. synchronized communication cost. *Settings*: partial scale, $\lambda = 0$, $c = 2$.

the context of this paper, the combinatorial explosion of planning alternatives. Koestler [20] labels this particular type of hierarchical design as 'holonic', emphasising its recursive self-encapsulated nature, with each hierarchical level representing both an entire system (whole) and a mere sub-system in a larger supra-system (part). This enables every holarchic level to be managed relatively independently from the higher and lower holarchic levels. It enables a divide-and-conquer approach to complex system management and hence enhances parallelism (increasing speed), reusablity (decreasing costs), and diversity (improving robustness) as each level can be managed independently [7].

From a software engineering perspective, the key properties of a holonic design include [7]: (i) inter-level *abstraction* – representing the state and/or behavior of a sub-system within its supra-system via an *aggregate description* (with loss of information); (ii) *partial isolation* among levels – ensuring that sub-systems only react to higher-level state changes describing their supra-systems, while not being influenced directly by detailed low-level state changes describing other subsystems; (iii) inter-level *time tuning* – ensuring relative execution times at different holarchic levels to avoid the occurrence of cross-level oscillations and/or divergence.

General-purpose holonic designs that integrate the above principles are proposed via Holonic Multi-Agent Systems (HMAS) platforms [39,38]; multi-level modeling and simulation approaches [12]; and hierarchical problem solving [21]. Holonic design has also been applied to various domains, including

hierarchical planning [29], traffic control [9], manufacturing [6,13] and smart grids [22,40,10].

Multi-agent approaches often rely on self-organization to achieve their goals, hence changing the structure of the multi-agent system at runtime. For instance, a holonic multi-agent approach for optimizing facility location problems is earlier proposed [28]: distribution of bus stops, hospitals or schools within a geographical area. The agents react to mutual attraction and repulsion forces to self-organize into a holarchy. Stable solutions represent optimal facility localization distributions. In contrast, the holarchic schemes studied in this paper maintain the agents' organizational structure, while the collective information used for system optimization becomes more localized.

The experimental work of this paper shows how different structural configurations, i.e. agents' positioning and number of children, influence learning performance. The key role that such hyperparameters play in ensuring the effectiveness of learning approaches is also confirmed by related work on the optimization of the hierarchical structure and its configuration variables – for instance grid search, i.e. exhaustive search of all possibilities, (which though suffers from exponential combinatorics), random search, Bayesian optimization, e.g. in neural networks and deep belief systems [4,5] and gradient-based optimization [24]. This is also relevant for deep learning applications via unsupervised pre-training [46] and evolutionary algorithms, e.g. in deep learning neural networks [26,47]. In the context of this work, such approaches can be used to determine the most effective holarchic structures and hyperparameter configurations.

In dynamic environments, rescheduling, or reoptimization [25] becomes a critical function for dealing with unpredicted disturbances. It usually raises the additional constraints of minimizing reoptimization time and minimizing the distance between the initial optimization solution and the re-optimized one – e.g. dynamic rescheduling in manufacturing systems [23] or shift rescheduling [25]. For instance, a two-level holarchy is earlier adopted to combine global optimization scheduling with fast rescheduling when dynamic system disturbances occur [25]. The work of this paper follows a similar principle as the holarchic schemes do not significantly impede on the learning performance while they promise higher resilience, a more localized optimization and re-usability, especially in highly distributed dynamic environments.

The design of I-EPOS [30] used here as baseline already adopts a hierarchical approach featuring inter-level abstraction i.e. via aggregate representations of lower-level scheduling solutions for the higher levels. However, the learning process over the hierarchy is performed level-by-level sequentially and not nested within multiple tree levels.

7 Conclusion and Future Work

This paper concludes that holarchic structures for decentralized deep learning can be a highly cost-effective organizational artifact for managing learning per-

formance under uncertainties of distributed environments. More specifically, an extensive experimental evaluation with more than 864000 experiments fed with synthetic and real-world data from pilot projects confirm the potential to explore, mitigate and boost the learning performance using three respective holarchic schemes applied to the I-EPOS decentralized deep learning system for solving combinatorial optimization problems.

Results show that the exploration of improving solutions is feasible and more likely to happen under stricter agents' constraints, while performance mitigation is effective with a higher likelihood in balanced tree topologies with higher number of children. Boosting the learning performance via holarchic structures is challenging yet consistently observed under computational problems with sparser data, i.e. the bike sharing application scenario. The partial scale of the holarchic structures is more cost-effective than the full scale. Nevertheless, when the uncertainties of distributed environments are not anymore a constraint, holarchic schemes cannot outperform the cost-effectiveness of learning systems that make use of the whole hierarchical structure, which is a finding consistent with earlier work on greedy optimization and suboptimum heuristics trapped in local optima [3].

Mechanisms for an automated activation and deactivation of holarchic schemes as well as the applicability of these schemes in other more complex hierarchical structures than tree topologies are subject of future work. Applying the concept of holarchy in non-hierarchical structures such as the unstructured learning network of COHDA [18] can provide new means to control high communication costs, while preserving a high learning performance.

Acknowledgements This work is supported by the European Communitys H2020 Program under the scheme 'INFRAIA-1-2014-2015: Research Infrastructures, grant agreement #654024 'SoBigData: Social Mining & Big Data Ecosystem (http://www.sobigdata.eu) and the European Communitys H2020 Program under the scheme 'ICT-10-2015 RIA', grant agreement #688364 'ASSET: Instant Gratification for Collective Awareness and Sustainable Consumerism'.

References

1. Aghaei, J., Alizadeh, M.I.: Demand response in smart electricity grids equipped with renewable energy sources: A review. Renewable and Sustainable Energy Reviews **18**, 64–72 (2013)
2. Aziz, A.A., Sekercioglu, Y.A., Fitzpatrick, P., Ivanovich, M.: A survey on distributed topology control techniques for extending the lifetime of battery powered wireless sensor networks. IEEE communications surveys & tutorials **15**(1), 121–144 (2013)
3. Bang-Jensen, J., Gutin, G., Yeo, A.: When the greedy algorithm fails. Discrete Optimization **1**(2), 121–127 (2004)
4. Bergstra, J., Bengio, Y.: Algorithms for hyper-parameter optimization. In: In NIPS, pp. 2546–2554 (2011)
5. Bergstra, J., Bengio, Y.: Random search for hyper-parameter optimization. J. Mach. Learn. Res. **13**, 281–305 (2012)
6. Christensen, J.H.: HMS/FB Architecture and its Implementation, pp. 53–87. Springer Berlin Heidelberg, Berlin, Heidelberg (2003)

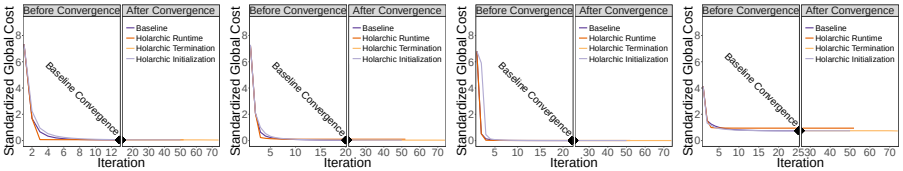
7. Diaconescu, A.: Goal-oriented holonic systems. In: C. Müller-Schloer, S. Tomforde (eds.) Organic Computing – Technical Systems for Survival in the Real World, pp. 209–258. Springer International Publishing (2017)
8. Diaconescu, A.: Organising complexity: Hierarchies and holarchies. In: C. Müller-Schloer, S. Tomforde (eds.) Organic Computing – Technical Systems for Survival in the Real World, pp. 89–105. Springer International Publishing (2017)
9. Fischer, K.: Holonic multiagent systems theory and applications . In: P. Barahona, J. Alferes (eds.) Progress in Artificial Intelligence, *LNCS*, vol. 1695, pp. 34–48. Springer Berlin Heidelberg (1999)
10. Frey, S., Diaconescu, A., Menga, D., Demeure, I.: A generic holonic control architecture for heterogeneous multiscale and multiobjective smart microgrids. ACM Trans. Auton. Adapt. Syst. **10**(2), 9:1–9:21 (2015)
11. Friedman, A., Knijnenburg, B.P., Vanhecke, K., Martens, L., Berkovsky, S.: Privacy aspects of recommender systems. In: Recommender Systems Handbook, pp. 649–688. Springer (2015)
12. Gaud, N.: A.: Towards a multilevel simulation approach based on holonic multiagent systems. In: In: UKSIM 08: Proceedings of the Tenth International Conference on Computer Modeling and Simulation, pp. 180–185 (2008)
13. Giret, A., Botti, V.: Engineering holonic manufacturing systems. Comput. Ind. **60**(6), 428–440 (2009)
14. Grande, R.C., Chowdhary, G., How, J.P.: Nonparametric adaptive control using gaussian processes with online hyperparameter estimation. In: Decision and Control (CDC), 2013 IEEE 52nd Annual Conference on, pp. 861–867. IEEE (2013)
15. Heidelberger, N., Karpinnen, K., D'Acunto, L.: Exposure diversity as a design principle for recommender systems. Information, Communication & Society, 2, 21, 191-207 (2018)
16. Helbing, D., Frey, B.S., Gigerenzer, G., Hafen, E., Hagner, M., Hofstetter, Y., van den Hoven, J., Zicari, R.V., Zwitter, A.: Will democracy survive big data and artificial intelligence. Scientific American **25** (2017)
17. Helbing, D., Pournaras, E.: Society: Build digital democracy. Nature **527**, 33–34 (2015)
18. Hinrichs, C., Lehnhoff, S., Sonnenschein, M.: A decentralized heuristic for multiple-choice combinatorial optimization problems. In: Operations Research Proceedings 2012, pp. 297–302. Springer (2014)
19. Hyndman, R.J., Koehler, A.B.: Another look at measures of forecast accuracy. International journal of forecasting **22**(4), 679–688 (2006)
20. Koestler, A.: The Ghost in the Machine, 1 edn. GATEWAY EDITIONS, Henry Regnery Co. (1967)
21. Landauer, C., Bellman, K.L.: New architectures for constructed complex systems. Appl. Math. Comput. **120**(1-3), 149–163 (2001)
22. Lässig, J., Satzger, B., Kramer, O.: Hierarchically structured energy markets as novel smart grid control approach. In: J. Bach, S. Edelkamp (eds.) KI 2011: Advances in Artificial Intelligence, pp. 179–190. Springer Berlin Heidelberg, Berlin, Heidelberg (2011)
23. Leitão, P., Restivo, F.: A holonic approach to dynamic manufacturing scheduling. Robot. Comput.-Integr. Manuf. **24**(5), 625–634 (2008)
24. Maclaurin, D., Duvenaud, D., Adams, R.P.: Gradient-based hyperparameter optimization through reversible learning. In: Proceedings of the 32Nd International Conference on International Conference on Machine Learning - Volume 37, ICML'15, pp. 2113–2122. JMLR.org (2015)
25. Meignan, D.: A heuristic approach to schedule reoptimization in the context of interactive optimization (2014)
26. Miikkulainen, R., Liang, J.Z., Meyerson, E., Rawal, A., Fink, D., Francon, O., Raju, B., Shahrzad, H., Navruzyan, A., Duffy, N., Hodjat, B.: Evolving deep neural networks. CoRR **abs/1703.00548** (2017). URL `http://arxiv.org/abs/1703.00548`
27. Mik, E.: The erosion of autonomy in online consumer transactions. Law, Innovation and Technology **8**(1), 1–38 (2016)
28. Moujahed, S., Gaud, N., Meignan, D.: A self-organizing and holonic model for optimization in multi-level location problems (2007)
29. Nolle, L., Wong, K., Hopgood, A.: Darbs: A distributed blackboard system. In: M. Bramer, F. Coenen, A. Preece (eds.) Research and Development in Intelligent Systems XVIII, pp. 161–170. Springer London (2002)

30. Pilgerstorfer, P., Pournaras, E.: Self-adaptive learning in decentralized combinatorial optimization: a design paradigm for sharing economies. In: Proceedings of the 12th International Symposium on Software Engineering for Adaptive and Self-Managing Systems, pp. 54–64. IEEE Press (2017)
31. Pournaras, E.: Multi-level reconfigurable self-organization in overlay services. Ph.D. thesis, TU Delft, Delft University of Technology (2013)
32. Pournaras, E., Jung, S., Zhang, H., Fang, X., Sanders, L.: Socio-technical smart grid optimization via decentralized charge control of electric vehicles. arXiv preprint arXiv:1701.06811 (2017)
33. Pournaras, E., Vasirani, M., Kooij, R.E., Aberer, K.: Decentralized planning of energy demand for the management of robustness and discomfort. IEEE Transactions on Industrial Informatics **10**(4), 2280–2289 (2014)
34. Pournaras, E., Yao, M., Helbing, D.: Self-regulating supply–demand systems. Future Generation Computer Systems **76**, 73–91 (2017)
35. Puchinger, J., Raidl, G.R.: Combining metaheuristics and exact algorithms in combinatorial optimization: A survey and classification. In: International Work-Conference on the Interplay Between Natural and Artificial Computation, pp. 41–53. Springer (2005)
36. Reymann, C., Renzaglia, A., Lamraoui, F., Bronz, M., Lacroix, S.: Adaptive sampling of cumulus clouds with uavs. Autonomous robots **42**(2), 491–512 (2018)
37. Rockafellar, R.T., Uryasev, S., et al.: Optimization of conditional value-at-risk. Journal of risk **2**, 21–42 (2000)
38. Rodriguez, S., Gaud, N., Galland, S.: SARL: a general-purpose agent-oriented programming language. In: the 2014 IEEE/WIC/ACM International Conference on Intelligent Agent Technology. IEEE Computer Society Press, Warsaw, Poland (2014)
39. Rodriguez, S., Hilaire, V., Gaud, N., Galland, S., Koukam, A.: Holonic Multi-Agent Systems, first edn., chap. 11, pp. 238–263. Self-Organising Software From Natural to Artificial Adaptation - Natural Computing. Springer (2011)
40. Schiendorfer, A., Steghöfer, J.P., Reif, W.: Synthesis and abstraction of constraint models for hierarchical resource allocation problems. Proc. of the 6th International Conference on Agents and Artificial Intelligence (ICAART) **2** (2014)
41. Simon, H.A.: The architecture of complexity. American Philosophical Society **106** (1962)
42. Simon, H.A.: The Sciences of the Artificial. MIT Press (1996)
43. Singla, A., Santoni, M., Bartók, G., Mukerji, P., Meenen, M., Krause, A.: Incentivizing users for balancing bike sharing systems. In: AAAI, pp. 723–729 (2015)
44. Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., Salakhutdinov, R.: Dropout: A simple way to prevent neural networks from overfitting. The Journal of Machine Learning Research **15**(1), 1929–1958 (2014)
45. Sterbenz, J.P., Hutchison, D., Çetinkaya, E.K., Jabbar, A., Rohrer, J.P., Schöller, M., Smith, P.: Resilience and survivability in communication networks: Strategies, principles, and survey of disciplines. Computer Networks **54**(8), 1245–1265 (2010)
46. Yao, C., Cai, D., Bu, J., Chen, G.: Pre-training the deep generative models with adaptive hyperparameter optimization. Neurocomputing **247**, 144 – 155 (2017)
47. Young, S.R., Rose, D.C., Johnston, T., Heller, W.T., Karnowski, T.P., Potok, T.E., Patton, R.M., Perdue, G., Miller, J.: Evolving deep networks using hpc. In: Proceedings of the Machine Learning on HPC Environments, MLHPC'17, pp. 7:1–7:7. ACM, New York, NY, USA (2017)

A Detailed Experimental Results

Figure 12 compares with Figure 5 by varying partial scale to full (Figure 12a-12d), $\lambda = 0$ to $\lambda = 0.5$ (Figure 12e-12h) and $c = 0$ to $c = 5$ (Figure 12i-12l).

Figure 13 elaborates on the Figure 6. It illustrates the probability density function of the improvement index by fixing the holarchic scale to partial, $c = 2$ and varying all other dimensions.

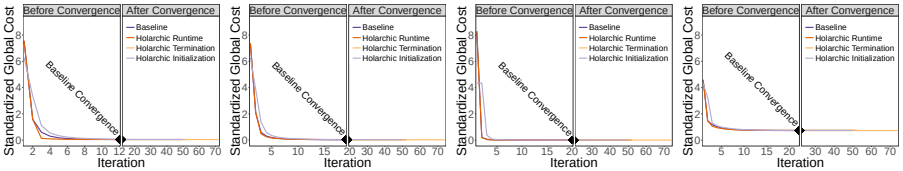


(a) Synthetic, full scale, $\lambda = 0$, $c = 2$

(b) Bike sharing, full scale, $\lambda = 0$, $c = 2$

(c) Energy demand, full scale, $\lambda = 0$, $c = 2$

(d) Electrical vehicles, full scale, $\lambda = 0$, $c = 2$

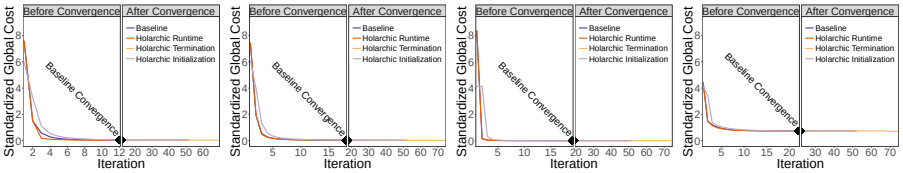


(e) Synthetic, partial scale, $\lambda = 0.5$, $c = 2$

(f) Bike sharing, partial scale, $\lambda = 0.5$, $c = 2$

(g) Energy demand, partial scale, $\lambda = 0.5$, $c = 2$

(h) Electrical vehicles, partial scale, $\lambda = 0.5$, $c = 2$



(i) Synthetic, partial scale, $\lambda = 0$, $c = 5$

(j) Bike sharing, partial scale, $\lambda = 0$, $c = 5$

(k) Energy demand, partial scale, $\lambda = 0$, $c = 5$

(l) Electrical vehicles, partial scale, $\lambda = 0$, $c = 5$

Fig. 12 Learning curves for comparison with Figure 5. *Dimensions*: holarchic schemes, application scenarios, full vs. partial scale, different λ values, varying number of children.

Figure 14 elaborates on the Figure 7. It illustrates the probability density function of the improvement index by fixing the holarchic scale to partial, $\lambda = 0$ and varying all other dimensions.

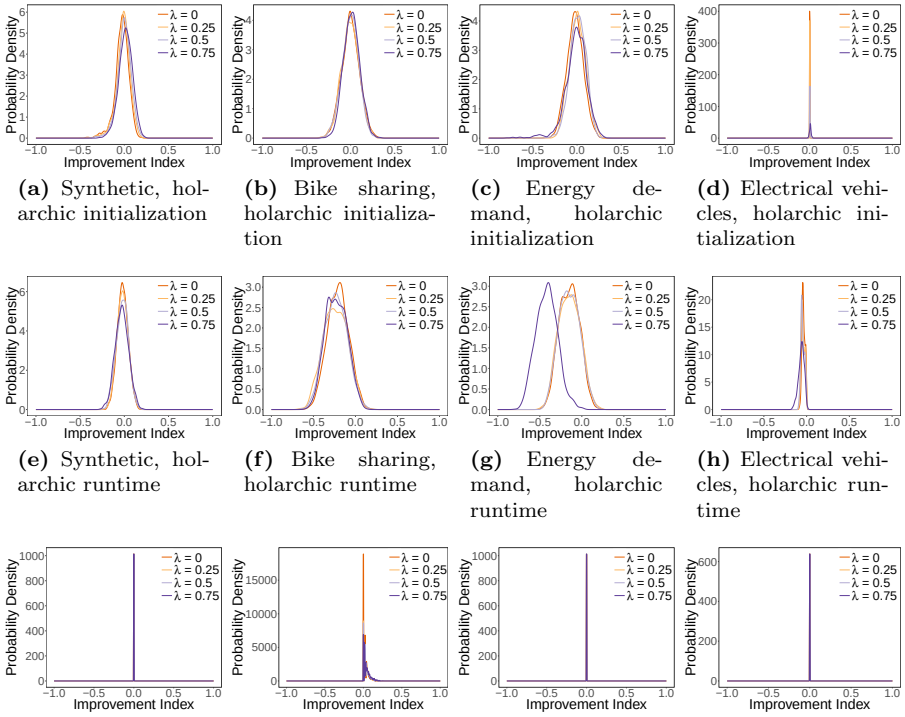
Figure 15 elaborates on the Figure 8. It illustrates the probability density function of the improvement index by fixing $\lambda = 0$, $c = 2$ and varying all other dimensions.

Figure 16 contrasts the improvement index of the holarchic runtime on the basis of total vs. synchronized communication cost. Results show an average increase of the improvement index by 9.5%, indicated by the shift to the right for synchronized communication.

The contrast of total vs. synchronized communication cost is illustrated in Figure 17 via the *relative performance* that is defined as follows:

$$P = \frac{C_{\mathsf{h}}^{(1)} - C_{\mathsf{h}}^{(T)}}{C_{\mathsf{b}}^{(1)} - C_{\mathsf{b}}^{(T)}} \tag{5}$$

where $C_{\mathsf{h}}^{(1)}$, $C_{\mathsf{b}}^{(1)}$ is the global cost at the first iteration $t = 1$ for holarchic runtime and baseline respectively, while $C_{\mathsf{h}}^{(T)}$, $C_{\mathsf{b}}^{(T)}$ is the global cost at convergence $t = T$. This metric encodes the additional information of the improvement extent during the learning iterations. For instance, while Figure 16f shows a density with improvement index values around -0.7,



(a) Synthetic, holarchic initialization

(b) Bike sharing, holarchic initialization

(c) Energy demand, holarchic initialization

(d) Electrical vehicles, holarchic initialization

(e) Synthetic, holarchic runtime

(f) Bike sharing, holarchic runtime

(g) Energy demand, holarchic runtime

(h) Electrical vehicles, holarchic runtime

(i) Synthetic, holarchic termination

(j) Bike sharing, holarchic termination

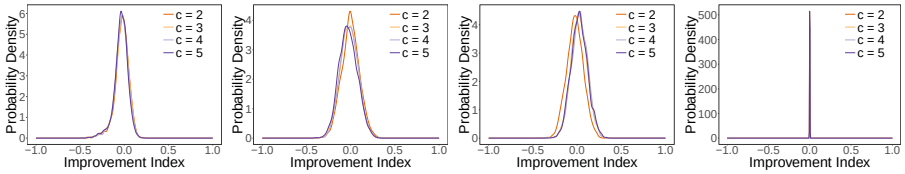
(k) Energy demand, holarchic termination

(l) Electrical vehicles, holarchic termination

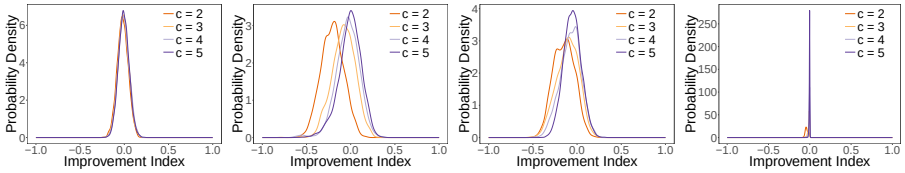
Fig. 13 Probability density of the improvement index that elaborates on Figure 6. *Dimensions*: holarchic schemes, application scenarios, different λ values. *Settings*: partial scale, $c = 2$.

the relative performance values in Figure 17f spread very close to 100%, which means that a performance peak is achieved.

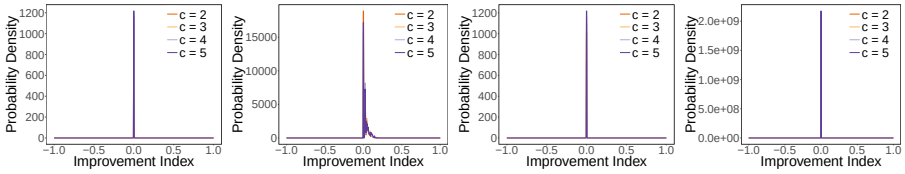
Figure 18 compares with Figure 11 by varying $c = 2$ to $c = 5$. In this case, the global cost of the baseline and holarchic runtime is equivalent for the required communication cost to converge. The global cost of the holarchic runtime requires fewer messages to drop for $c = 5$ compared to $c = 2$ as indicated by the respective line shifted to the left.



(a) Synthetic, holarchic initialization **(b)** Bike sharing, holarchic initialization **(c)** Energy demand, holarchic initialization **(d)** Electrical vehicles, holarchic initialization

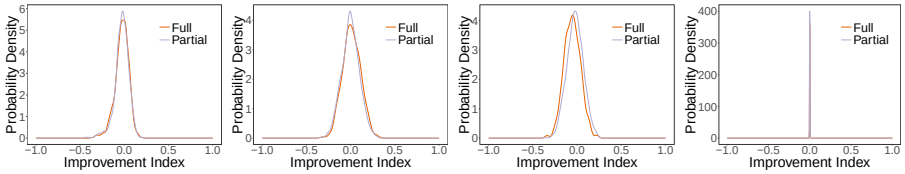


(e) Synthetic, holarchic runtime **(f)** Bike sharing, holarchic runtime **(g)** Energy demand, holarchic runtime **(h)** Electrical vehicles, holarchic runtime

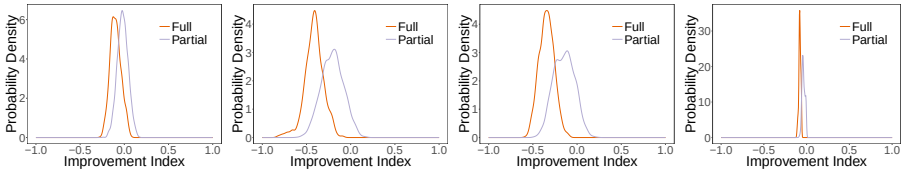


(i) Synthetic, holarchic termination **(j)** Bike sharing, holarchic termination **(k)** Energy demand, holarchic termination **(l)** Electrical vehicles, holarchic termination

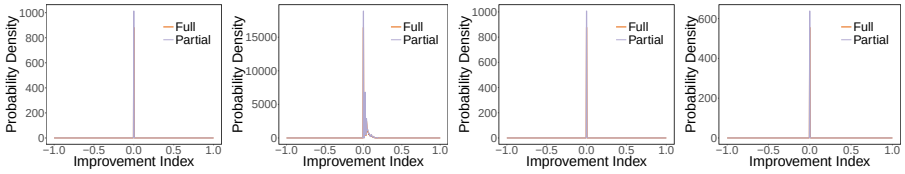
Fig. 14 Probability density of the improvement index that elaborates on Figure 7. *Dimensions*: holarchic schemes, application scenarios, varying number of children. *Settings*: partial scale, $\lambda = 0$.



(a) Synthetic, holarchic initialization **(b)** Bike sharing, holarchic initialization **(c)** Energy demand, holarchic initialization **(d)** Electrical vehicles, holarchic initialization

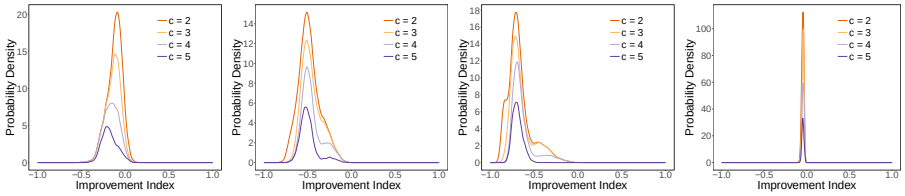


(e) Synthetic, holarchic runtime **(f)** Bike sharing, holarchic runtime **(g)** Energy demand, holarchic runtime **(h)** Electrical vehicles, holarchic runtime



(i) Synthetic, holarchic termination **(j)** Bike sharing, holarchic termination **(k)** Energy demand, holarchic termination **(l)** Electrical vehicles, holarchic termination

Fig. 15 Probability density of the improvement index that elaborates on Figure 8. *Dimensions*: holarchic schemes, application scenarios, partial vs. full scale. *Settings*: $\lambda = 0$, $c = 2$.

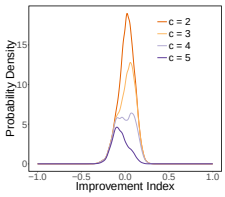


(a) Synthetic, total communication

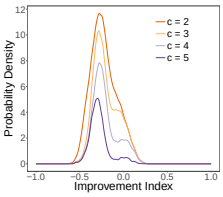
(b) Bike sharing, total communication

(c) Energy demand, total communication

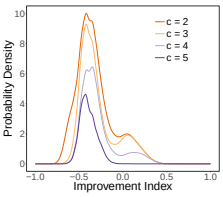
(d) Electrical vehicles, total communication



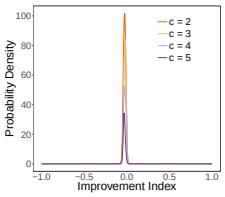
(e) Synthetic, synchronized communication



(f) Bike sharing, synchronized communication

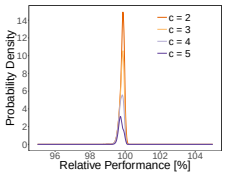


(g) Energy demand, synchronized communication

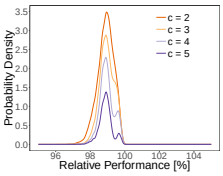


(h) Electrical vehicles, synchronized communication

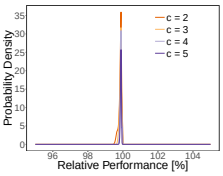
Fig. 16 Probability density of the improvement index. *Dimensions*: total vs. synchronized communication cost, application scenarios, varying number of children. *Settings*: holarchic runtime, partial scale, $\lambda = 0$.



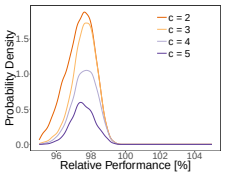
(a) Synthetic, total communication



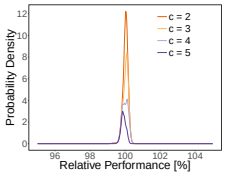
(b) Bike sharing, total communication



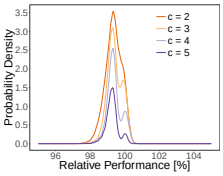
(c) Energy demand, total communication



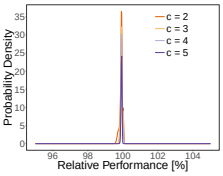
(d) Electrical vehicles, total communication



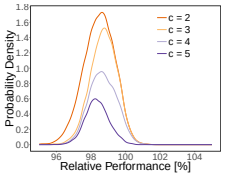
(e) Synthetic, synchronized communication



(f) Bike sharing, synchronized communication

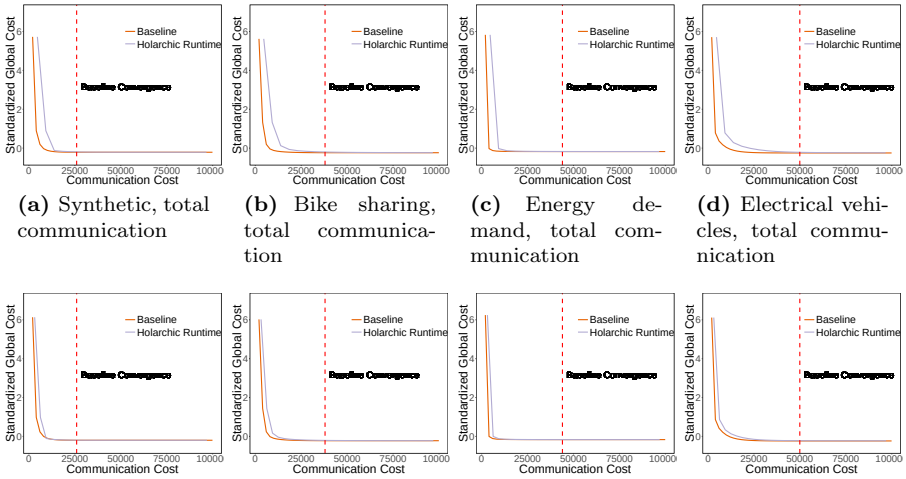


(g) Energy demand, synchronized communication



(h) Electrical vehicles, synchronized communication

Fig. 17 Probability density of the relative global cost reduction. *Dimensions*: total vs. synchronized communication cost, application scenarios, varying number of children. *Settings*: holarchic runtime, partial scale, $\lambda = 0$.



(a) Synthetic, total communication

(b) Bike sharing, total communication

(c) Energy demand, total communication

(d) Electrical vehicles, total communication

(e) Synthetic, synchronized communication

(f) Bike sharing, synchronized communication

(g) Energy demand, synchronized communication

(h) Electrical vehicles, synchronized communication

Fig. 18 Cost effectiveness for comparison with Figure 11. *Dimensions*: baseline vs. holarchic runtime, application scenarios, total vs. synchronized communication cost. *Settings*: partial scale, $\lambda = 0$, $c = 5$.