Local Agent-based Self-stabilisation in Global Resource Utilisation

Evangelos Pournaras*
Martijn Warnier
Frances M.T. Brazier

Section of Systems Engineering
Department of Multi-actor Systems
Faculty of Technology, Policy and Management
Delft University of Technology
The Netherlands
E-mail: {E.Pournaras,M.E.Warnier,F.M.Brazier}@tudelft.nl
*Corresponding author

Abstract Distributed management of complex large-scale infrastructures, such as power distribution systems, is challenging. Sustainability of these systems can be achieved by enabling stabilisation in global resource utilisation. This paper proposes EPOS, the Energy Plan Overlay Self-stabilisation system, for this purpose. EPOS is an agent-based approach that performs self-stabilisation over a tree overlay, as an instance of a hierarchical virtual organisation. The global goal of stabilisation emerges through local knowledge, local decisions and local interactions among software agents organised in a tree. Two fitness functions are proposed to stabilise global resource utilisation. The first proactively keeps deviations minimised and the second reactively reverses deviations. Extensive experimentation reveals that EPOS outperforms a system that utilises resources in a greedy manner. Finally, this paper also investigates and evaluates factors that influence the effectiveness of EPOS.

Keywords: software agent; adaptation; tree overlay; resource utilisation; stabilisation; energy management; thermostatic devices.

Reference to this paper should be made as follows: Pournaras E., Warnier M., Brazier F.M.T. (2009) 'Local Agent-based Self-stabilisation in Global Resource Utilisation', Int. J. Autonomic Computing, Vol. X, Nos. X/X/X, pp.X–X.

Biographical Notes: Evangelos Pournaras is a PhD student in Delft University of Technology, department of Multi-actor Systems, section of Systems Engineering. He moved from the VU University in Amsterdam where he started his PhD research. He holds a MSc with distinction in Internet Computing from University of Surrey, UK. Before this, he finished his BSc in the department of Technology Education and Digital Systems from University of Piraeus, Greece. His research focuses on the self-optimisation and self-organisation of distributed systems using peer-to-peer and software agent technologies.

Martijn Warnier is an assistant professor in Delft University of Technology, in the department of Multi-actor Systems, section of Systems Engineering. He received his PhD in Computer Science at the Radboud University Nijmegen, the Netherlands. He worked as a Postdoctoral Researcher in the IIDS group at the

VU University Amsterdam. His research interests include autonomic computing and computer security.

Frances Brazier is a full professor at Delft University of Technology, department of Multi-actor Systems, section of Systems Engineering. Previously she had a full professorship position at the VU University Amsterdam, where she chaired the IIDS group. Her group's research focus is on design and (self-)management of large scale distributed autonomous systems.

1 Introduction

Distributed resource management is challenging, especially when applied in large-scale virtual organisations such as power grid infrastructures. The state of these systems changes continuously: adaptation is required. This paper addresses how local adaptations can be used for the management of global resource utilisation in large-scale distributed systems.

Adaptivity has been recognised by Wirsing & Holzl (2006) as a means to handle the arising complexity of knowledge and interactions. Software agents are capable of autonomously adapting their behaviour to dynamic environments as illustrated for resource management and scheduling in the work of Chakravarti, Baumgartner & Lauria (2005). Software agents are strongly related to peer-to-peer, grid computing and service oriented architectures (SOA) with many overlapping concepts and applications. This fact is outlined in the work of Moro, Ouksel & Sartori (2002), Tianfield & Unland (2005) and Brazier, Kephart, Parunak & Huhns (2009).

Virtual organisations of agents define communication structures between agents, e.g., hierarchical organisations, see Jennings (2001). Agents benefit from hierarchical organisations as they can choose to cooperate and coordinate their actions between and within them, or compete. Dynamic organised hierarchies, proposed by Luis (1999), can be used to support adaptive, aggregate, nonlinear behaviour, as a means to reduce complexity. In contrast, coordination in unstructured environments entails distributed search and distributed scheduling as shown by the application examples of Theocharopoulou, Partsakoulakis, Vouros & Stergiou (2007) and Johansson, Davidsson & Carlsson (2000).

This paper focuses on the problem of stabilisation in global resource utilisation. In this paper, *resource utilisation* is defined as the amount of resources that are allocated over a period of time. *Global resource utilisation* entails the aggregation of every individual local allocation that is performed for the same time period. Finally, *stabilisation* means that the acquired global resource utilisation maintains the system in a beneficial robust state.

The core question addressed in this paper is the following:

To which extent can stabilisation in global resource utilisation be acquired by local coordination in the resource utilisation of software agents?

This paper proposes *EPOS, the Energy Plan Overlay Self-stabilisation* system. EPOS is a fully decentralised agent-based approach for achieving stabilisation in global resource utilisation. It is based on one of our earlier papers (Pournaras, Warnier & Brazier 2009*b*). The approach of EPOS is threefold and can be outlined as follows:

1. Agents are organised in a tree overlay as an instance of a hierarchical virtual organisation. This organisation scheme structures the interactions and the aggregation of the agents' resource utilisations.

2. Agents are able to exploit alternative resource utilisations that all satisfy their requirements.
3. Agents can make local adaptive decisions on the basis of information they receive from the agents to which they are linked.

The problem and the proposed solution are illustrated in the context of the energy management in the electricity domain. This paper assumes a virtual organisation of consumer devices, specifically thermostatically controlled appliances, such as refrigerators, air conditioners, water heaters etc. These devices consume 25% of the total energy supply in the USA according to Mazza (2002), thus their management can have a significant impact on the sustainability of the whole power system.

In this context, thermostatic devices are controlled by software agents. The type of resource discussed in this context corresponds to the energy consumed by thermostatic devices or the energy provided to these consumer devices by electricity providers. Resource utilisation also refers to the allocation of energy by the agents of thermostatic devices for a period of time. Finally, stabilisation of the global energy utilisation means that the system has fewer deviations (peaks) in its total consumption and is able to shift or 'fade-out' high power peaks. These stabilisation issues are addressed in the related work of Shaw, Attree, Jackson & Kay (2009) and Strbac (2008).

EPOS is extensively evaluated in a simulation environment. Results reveal significant improvement in the stabilisation compared to a system that performs greedy selections. Beyond this comparison, this paper identifies and extensively evaluates the factors that influence the degree of stabilisation achieved.

This paper is outlined as follows: Section 2 illustrates the technical background related to the application context. Section 3 discusses issues that concern the self-organisation and robustness of the tree overlay that EPOS is based on. Section 4 outlines the local tasks and knowledge of the agents. Section 5 presents the core algorithm of EPOS. Section 6 explains the local decision-making process and the two fitness functions on which the stabilisation is based. Section 7 provides an overview of the convergence of the stabilisation process over the tree overlay. Section 8 moves to the experimental part of this paper by illustrating the simulation environment and the results derived from it. Section 9 illustrates related work and Section 10 discusses the findings and open issues of EPOS. Finally, Section 11 concludes this paper and outlines future work.

2 Application

Large-scale power system are highly complex and heterogeneous. They consist of many different participating parties with different goals, such as consumers, providers, and utility companies. Sustainability of power systems and distributed energy management is challenging. One of our related papers (Brazier, Ogston & Warnier 2009) discusses various challenges related to decentralised self-management in energy markets.

EPOS contributes to the sustainability of power infrastructures by proposing two self-stabilisation functions in a network of thermostatic devices controlled by software agents. The functions are the following:

- **Minimum deviations:** The system *proactively* tries to minimise the deviations in the global energy utilisation.

- **Reversed deviations:** The system *reactively* reverses the deviations of a previous global energy utilisation. This means that the average result of the two global energy utilisations should be a 'flat' one over time.

In Section 1, stabilisation is defined and related to a beneficial robust state of the system. The above functions correspond to two such states. The first function assumes that the system does not receive considerable perturbations. It aims to retain this state by keeping the deviations to a minimum. In contrast, the second function assumes that some perturbations are experienced by the system and thus it should compensate them. For example, a sudden peak in the energy consumption results in increasing the energy provided by a power plant. For this reason, reversing the high energy consumption to a low energy consumption in the next utilisation period keeps the cost of the energy provided by this power plant balanced.

As mentioned before, EPOS is based on agents that control thermostatic devices. It is also assumed that these agents can communicate through an infrastructure such as the Internet or the dedicated power grid. An example of the latter communication scheme is illustrated by Stadler, Krause, Sonnenschein & Vogel (2009). These options are realistic according to the work of Guo, Li & James (2005), James, Cohen, Dodier, Platt & Palmer (2006) and Hammerstrom (2002).

EPOS is able to apply the above stabilisation scheme by exploiting local alternative utilisations based on which the thermostatic devices can operate. Thermostatic devices work in a periodic fashion by turning the thermostat 'on' and 'off'. In this way, the temperature of the device remains within the target range of its operation. The temperature for which the thermostat changes from the 'on' state to the 'off' state, and visa versa, is referred to as the *temperature setpoint*. Note that thermostatic devices consume energy during the 'on' state.

The intuition behind EPOS is that software agents can modify the temperature setpoints and change the energy utilisation over time by keeping the target temperature range unmodified. Thus, software agents in EPOS generate alternative *plans* for energy utilisation for a given period of time. Each plan consists of a sequence of discreet values. Each value represents a level of energy utilisation. A set of *possible plans* represents different options for energy utilisation for a single device during a given period of time. Note that this notion of energy plan is used throughout this paper. Related work, such as the one by Lu, Chassin & Widergren (2005), also explores these alternative energy utilisations. Figure 1 depicts the idea of generating different possible plans for a thermostatic device such as a water heater.

Each agent in EPOS should know which possible plan to execute. This means that the agents should choose one of the configurations for the thermostat for the time period defined by the energy plans. The decisions of all agents as to which plan to execute potentially results in a global plan that is more stabilised compared to a greedy system in which agents always use a single plan. Note that the global plan corresponds to the aggregation of every selected local utilisations (plans) of the participating agents in the system.

The aggregation and decision-making scheme of EPOS is based on a tree overlay. Before discussing these processes, the next section outlines how the agents can be self-organised in a robust tree overlay that can support the aggregation and decision-making.

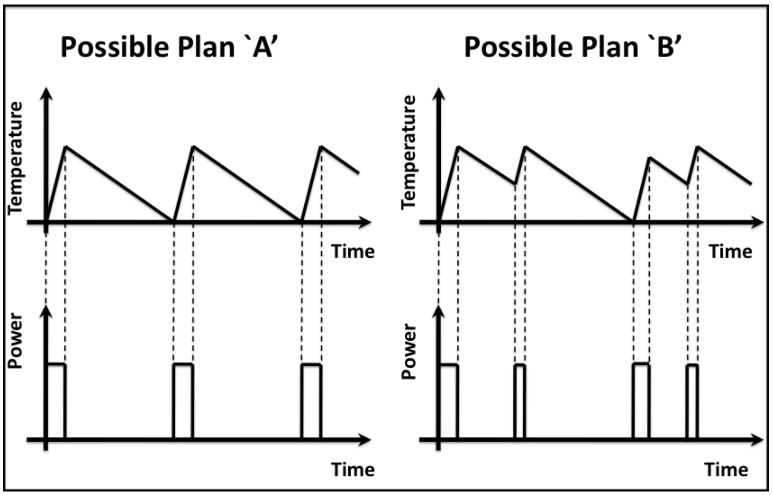

Figure 1 Two possible energy plans. Possible plan ‘A’ is a conventional one that always uses the same temperature setpoints. In contrast, the possible plan ‘B’ is generated by modifying the temperature setpoints. The power distribution of this plan over time is different.

3 Self-organisation and Robustness

Tree overlays can be a very effective distributed communication scheme for performing operations such as search, aggregation and information dissemination. The cost of these operations is usually logarithmic or bounded to the size of the network.

EPOS could perform central coordination to optimally stabilise the global energy utilisation. This approach would include the brute-force calculation of all the aggregated combinations of possible plans that the agents generate. This solution guarantees the discovery of the best global solution, considering and assuming that the sets of possible plans do not change during aggregation. Such central coordination, however, does not scale. The complexity is $O(p^n)$, where p is the number of possible plans per agent and n is the number of agents in the network.

In contrast, EPOS performs fully distributed coordination over a tree overlay. In this case, the complexity is bound to $O(p^c)$, where c is the number of children per node in the tree. The estimations assume a balanced tree structure and fixed p for all agents. Scalability improves significantly and is influenced by the trade-off between processing cost and latency. Similar trade-offs are explored by Tan, Jarvis, Chen & Spooner (2006).

Despite the above benefits of using tree overlays, they are highly sensitive to failures in distributed environments. Similar to other structured overlays, such as Distributed Hash Tables (DHTs) discussed by Rhea, Geels, Roscoe & Kubiatowicz (2004), individual node failures have a high impact on the global topology, especially in trees. A failure in a node disconnects the branch underneath. In addition, trees require optimisation according to different, and sometimes conflicting, performance metrics. For example, in the work of Tan, Jarvis, Chen & Spooner (2006), a bandwidth-ordered and a time-ordered tree are combined to optimise application-level multicast. The work of Frey & Murphy (2008) explores different repair strategies and Fei & Yang (2007) propose a proactive approach to recovery from failures. Finally, Plumtree, proposed by Leitao, Pereira & Rodrigues (2007), combines eager and lazy push gossiping strategies to build and maintain a dynamic tree overlay.

These examples reveal significant effort from different communities, that investigate different applications, to guarantee a certain level of robustness and fault-tolerance in tree overlays. Thus, using trees in a dynamic distributed environment could be feasible in many cases.

Different applications have different notions of robustness. It is difficult to identify the most suitable self-organised tree overlay that could satisfy the requirements of EPOS. Some of our related work (Pournaras, Warnier & Brazier 2009a) focuses on providing a generic self-organisation framework for: (i) proactively optimising a tree overlay according to one or more metrics provided by the application that uses the overlay, (ii) reactively adapting the topology to the changes of the environment. *AETOS, the Adaptive Epidemic Tree Overlay Service*, AETOS is based on the concept of combining highly robust unstructured overlays with structured ones.

Combining EPOS with AETOS could enhance the effectiveness of stabilisation as it guarantees that more nodes dynamically participate in the aggregation and decision-making processes that are described in Section 5 and 6 respectively. However, this paper focuses on EPOS, leaving reliability, fault-tolerance, the integration of the two systems and its evaluation for future work.

4 The Locality of the Agents

Agents use local knowledge to perform local computations and execute their tasks locally, using knowledge acquired from their children and/or their parent. Table 1 outlines the mathematical symbols that are used for the remainder of this paper.

A plan is represented by a set of arithmetic values that correspond to the energy consumed at a specific time period. For example, the plan $\boldsymbol{x} = (5.3, 6.4, 6.1, 8.6, 8.1, 4.2)$ represents the utilisation of energy for 6 consecutive equal time periods. During the third time period, the plan allocates 6.1 units of energy. Note that the symbols for plans are bold as they are sets of energy values over time. In contrast, the set of plans, as supersets, are depicted with bold and capital letter.

4.1 Overview of Agent Tasks

The main tasks of an agent are (plan) generation, (plan) aggregation and (plan) execution:

Generation is composed of two subtasks:

- **Planning:** A set $\boldsymbol{P}$ of possible plans $\boldsymbol{p}$ is generated for the energy consumption of the thermostatic device that an agent represents over a fixed time period.
- **Parent Inform:** The agent sends its possible plans $\boldsymbol{P}$ and one or more aggregate plans $\boldsymbol{a}$ to its parent. The aggregate plans are described in detail in Section 4.2.

Aggregation is composed of four subtasks:

- **Pre-processing:** The agent generates the set of combinations $\boldsymbol{C}$ from the received sets $\boldsymbol{P}$ of possible plans $\boldsymbol{p}$. It also merges the aggregate plans $\boldsymbol{a} \in \boldsymbol{A}$ received from

Table 1 The main mathematical symbols used in this paper.

	Symbol	Description
Plan	$\boldsymbol{p}$	possible plan from $\boldsymbol{P}$
	$\boldsymbol{p}'$	selected possible plan from $\boldsymbol{P}$
	$\boldsymbol{a}$	aggregate plan referring to $\boldsymbol{a}_n$ or $\boldsymbol{a}_h$
	$\boldsymbol{a}_n$	aggregate new plan ($\boldsymbol{a}_n \subseteq \boldsymbol{g}$)
	$\boldsymbol{a}_h$	aggregate history plan ($\boldsymbol{a}_h \subseteq \boldsymbol{g} \in \boldsymbol{H}$)
	$\boldsymbol{c}$	combination plan from $\boldsymbol{C}$
	$\boldsymbol{c}'$	selected combination plan from $\boldsymbol{C}$
	$\boldsymbol{g}$	global plan ($\boldsymbol{g} \supseteq \boldsymbol{a}$)
Set of Plans	$\boldsymbol{P}$	set of possible plans $\boldsymbol{p}$
	$\boldsymbol{C}$	set of combination plans $\boldsymbol{c}$
	$\boldsymbol{A}$	set of aggregate plans $\boldsymbol{a}$
	$\boldsymbol{A}_n$	set of aggregate new plans $\boldsymbol{a}_n$
	$\boldsymbol{A}_h$	set of aggregate history plans $\boldsymbol{a}_h$
	$\boldsymbol{H}$	the set of plans $(\boldsymbol{p}', \boldsymbol{g})$ from a previous aggregation round
Fitness Function	f_{MD}	the minimum deviations fitness function
	f_{RD}	the reversed deviations fitness function

its children. Note that $\boldsymbol{A}$ is the set of aggregate plans received from the children of an agent. Section 4.3 describes the pre-processing task in detail.

- **Selection:** The agent chooses the best plan combination $\boldsymbol{c}' \in \boldsymbol{C}$. The decision-making is based on one of the fitness functions f_{MD} and f_{RD} as explained in Section 6. From the best combination $\boldsymbol{c}'$, each selected plan $\boldsymbol{p}'$ for every child is derived.

- **Update:** Each aggregate plan $\boldsymbol{a}$ of the agent is updated with the respective selected combination $\boldsymbol{c}'$. More information about the update task is given in Section 6.1 and 6.2.

- **Children Inform:** The agent sends the selected plans $\boldsymbol{p}'$ to its children.

Execution concerns the execution of the selected plan $\boldsymbol{p}' \in \boldsymbol{P}$ that is received from the parent. In this case, the agent controls the ‘on’ and ‘off’ states of the thermostat according to the energy utilisation that the selected plan defines.

The root agent executes an additional **broadcast** task that is important for the aggregation process. Section 5 provides more information about the execution of this additional task.

4.2 Aggregate and History Plans

The aggregate plan $\boldsymbol{a}$ represents the plan selections that have been made so far over the tree overlay by a set of agents. Specifically, it refers to the aggregated selections that have been made in the branch underneath an agent. In EPOS, an aggregate plan can be: the *aggregate new plan* $\boldsymbol{a}_n$ and the *aggregate history plan* $\boldsymbol{a}_h$. The former corresponds to the aggregate plan that is calculated by the selections during the current aggregation, whereas the latter refers to the selections in a previous aggregation. The aggregate plan $\boldsymbol{a}$ from leaf agents is zero, whereas the one of the root agent, at the end of each round, is the final converged global plan $\boldsymbol{g}$.

The notion of history $\boldsymbol{H}$, as part of each agent's knowledge, is in fact a set $\boldsymbol{H} = (\boldsymbol{p}', \boldsymbol{g})$ that includes: (i) the selected plan $\boldsymbol{p}' \subset \boldsymbol{g} \in \boldsymbol{H}$ of a previous aggregation performed and (ii) the root's aggregate plan, that is the global plan $\boldsymbol{g}$ as a result of a previous aggregation over the tree overlay. Note that $\boldsymbol{p}' \in \boldsymbol{H} \subseteq \boldsymbol{a}_h \subseteq \boldsymbol{g} \in \boldsymbol{H}$. The global plan is broadcasted by the root at the end of each round. The role of this task in the aggregation process is illustrated in Section 5.

4.3 Aggregation pre-processing

Before an agent can calculate the fitness function, it performs some pre-processing of the information it receives from its children. This pre-processing concerns the computation of all possible combinations $\boldsymbol{C}$ of all possible plans generated by its children. For example, an agent with $c = 2$ children, each with $p = 2$ possible plans generates the following 4 combinations:

$$\boldsymbol{C} = \{(\boldsymbol{P}_1^1 + \boldsymbol{P}_1^2), (\boldsymbol{P}_1^1 + \boldsymbol{P}_2^2), (\boldsymbol{P}_2^1 + \boldsymbol{P}_1^2), (\boldsymbol{P}_2^1 + \boldsymbol{P}_2^2)\}$$

with the expression $\boldsymbol{P}_j^i$ defining the possible plans in each combination for the $i = 1, ..., c$ child and $j = 1, ..., p$ possible plan.

Each agent also merges the aggregate plans received from its children by summing them up. The computation of the aggregate plans is performed as follows:

$$\text{(1)} \qquad \boldsymbol{a}_n = \sum_{i=1,2,\ldots,c} \boldsymbol{a}_n \in \boldsymbol{A}_n^i \qquad \boldsymbol{a}_h = \sum_{i=1,2,\ldots,c} \boldsymbol{a}_h \in \boldsymbol{A}_h^i$$

for c number of children.

5 The Core Algorithm

The *core algorithm* of EPOS, illustrated in Algorithm 1, defines (i) the interactions between the agents over the tree overlay and (ii) the execution sequence of the local tasks. The algorithm is based on the notion of the *aggregation step* and *aggregation round.* These concepts are illustrated in Figure 2.

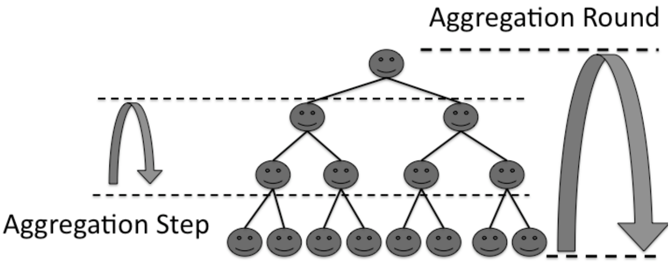

Figure 2 The aggregation step and aggregation round of EPOS

Algorithm 1 The core agent algorithm of EPOS

```
1. if receive parent inform messages
2. then aggregation-pre-processing;
3.      aggregation-selection;
4.      aggregation-update;
5.      aggregation-children inform;
6.      generation-planning;
7.      generation-parent inform;
8.      if is root
9.      then aggregation-selection;
10.          aggregation-update;
11.          broadcast;
12.          execution;
13. if receive children inform message
14. then execution;
15. if receive broadcast message and is leaf
16. then generation-planning;
17.      generation-parent inform;
```

During an aggregation step, the agents that belong in a level of the tree receive plan information from their children. This is the set of possible plans $\boldsymbol{P}$ and the aggregate plans $\boldsymbol{a}$. Upon receiving this information from each of their children , they execute the aggregation and generation tasks respectively (lines 1-7 in Algorithm 1). During these two tasks, the agents select a possible plan for execution for each of their children (line 3 in Algorithm 1), inform them about these selections (line 5 in Algorithm 1) and trigger the next aggregation step by informing, this time, their own parents about their plans (line 7 in Algorithm 1). The children execute the selected plans (line 14 in Algorithm 1). The process is recursive over all levels of the tree.

Note that the root agent selects its own execution plan, without sending the plans to another agent (lines 8-12 in Algorithm 1). In this case, the set of combinations is $\boldsymbol{C} = \boldsymbol{P}$. In other words, the root aggregates twice, one for its children and one for itself. Moreover, the root broadcasts information to all the other agents and, in this way, can initiate the next aggregation round.

An aggregation round is defined by all of the consecutive aggregation steps starting from the leaf agents up to the root agent, ending with the global aggregate plan $\boldsymbol{g}$ being broadcasted to all agents. When the leaf agents receive the broadcast message, they execute the generation task and, in this way, trigger the next aggregation round (lines 15-17 in Algorithm 1).

6 Decision-making

Choosing the best possible plan is related to the stabilisation goal of the system. As outlined in Section 2, EPOS proposes a proactive and reactive stabilisation. Decision-making is based on two fitness functions that achieve stabilisation by minimising deviations or reversing deviations. The next two subsections illustrate these two fitness functions.

6.1 Stabilisation by Minimising Deviations

The minimum deviations fitness function defines the best combination $\boldsymbol{c}'$ to be the one that minimises the standard deviation σ of the aggregate new plan $\boldsymbol{a}_n$. This can be expressed as:

$$f_{MD} = \min_{i=1,2,\ldots,|\boldsymbol{C}|} \sigma(\boldsymbol{a}_n + \boldsymbol{C}_i) \tag{2}$$

This fitness function checks which of the potentially new aggregate plans has the minimum overall standard deviation σ. Section 7 discusses the convergence of the process. The selected plans for every child are extracted from the best combination $\boldsymbol{c}' \in \boldsymbol{C}$. Moreover, the aggregate knowledge is updated. This action concerns the update of the aggregate new plan as $\boldsymbol{a}_n = \boldsymbol{a}_n + \boldsymbol{c}'$. Finally, the selected plans $\boldsymbol{p}$ are sent to the respective children.

6.2 Stabilisation by Reversing Deviations

Given a previously aggregated global plan , the reversed deviation fitness function results in a new global plan that reverses the deviations of the previous one. This previous global plan, called the *global history plan*, is devised by the same agents, representing the same devices. This fitness function specifies that, if resource providers have to supply $\overline{g} + v_t$ amount of resources at time t then reverse of the deviations provides $\overline{g} - v_t$. Where $\overline{g}$ is the average value of plan $\boldsymbol{g}$ and v_t is the variation, that is the difference between the average value $\overline{g}$ and the value of the global plan at time t.

The aggregation process remains exactly the same. The fitness function is calculated as follows:

$$f_{RD} = \min_{i=1,2,\ldots,|\boldsymbol{C}|} \sigma(\overbrace{\boldsymbol{g} - \underline{\boldsymbol{a}_h}}^{history} \underbrace{+ \overbrace{\underline{\boldsymbol{a}_n + \boldsymbol{C}_i}}^{new}}_{replacement}) \tag{3}$$

The aggregate history plan $\boldsymbol{a}_h$ is replaced by the equivalent summation of the aggregate new plan $\boldsymbol{a}_n$ and the plan $\boldsymbol{C}_i$ from the combinations ($\boldsymbol{a}_h \equiv \boldsymbol{a}_n + \boldsymbol{C}_i$). This replacement is adapted (summed) to the global history plan $\boldsymbol{g} \in \boldsymbol{H}$.

The concept behind this reversing operation is the following: The average of the global history plan and global new plan must ideally result in zero deviations. This is because

$\frac{(\overline{g}+v_t)+(\overline{g}-v_t)}{2} = \overline{g}$. Choosing the combination that contributes best on transforming v_t to $-v_t$ results in a global plan with reversed deviations. The transformation is gradually achieved in every aggregation step. Section 7 discusses the convergence of this process.

Finally, as in the previous case, the aggregate plans are updated as $\boldsymbol{a}_n = \boldsymbol{a}_n + \boldsymbol{c}'$ and $\boldsymbol{a}_h = \boldsymbol{a}_h + \boldsymbol{p}' \in \boldsymbol{H}$ respectively. As mentioned in Section 6.1, the selected plans are extracted and sent to the respective children.

7 Emerging Stabilisation Convergence

The two fitness functions, described in Sections 6.1 and 6.2, are based on local knowledge. The decisions each agent makes are based on the aggregate plans it receives from its children, by definition a subset of the global knowledge. Summations are performed over the hierarchical structure during the aggregation process. In each aggregation step, the aggregate values of the aggregate plans increase. As a result, the combinations are adapted to plans that come from the operations of more agents.

The aggregation starts from the leaf agents. In this first level of the tree, the parents provide the best option taking into account only the information from children as the values of the aggregate plans are zero. In the next steps, local decisions have a twofold advantage. Each agent not only chooses the plans that: (i) satisfy the stabilisation goal locally but also (ii) plans that 'fade out' the effect of less optimal decisions in previous aggregation steps. For the first fitness function, the approach based on minimising deviations, the second advantage is achieved by adapting the combinations to the aggregate new plan. For the second fitness function, the approach based on reversed deviations, the aggregate history plan is replaced by the aggregate new plan and adapted to the global history plan. The degree of adaptation increases after each aggregation step as the values of the aggregates plans increase as well.

This step-by-step emerging convergence is depicted in Figure 3 for the second fitness function, namely the function based on reversed deviations. In the first aggregation steps, decisions are mainly based on the aggregate history plan. As aggregation evolves, the global history plan is gradually replaced by the aggregate new plan. Finally, the aggregate new plan converges to the new global plan devised by the root agent.

8 Experiments and Results

This section illustrates the experimental environment and the results from the evaluation of the proposed stabilisation approach. More specifically, a series of experiments evaluate the two fitness functions: the minimum deviations and reversed deviations described in Section 6.1 and 6.2 respectively.

EPOS is compared with a system that performs greedy selections. This means that there are no alternative energy plans. This is implemented by configuring agents to generate a single possible plan. There is also an effort to compare EPOS with the optimum centralised coordination of complexity $O(p^n)$ that is discussed in Section 3.

The experimental study aims to answer the five following questions:

1. *What is the degree of minimum deviations achieved in EPOS compared to greedy selection and centralised coordination?*

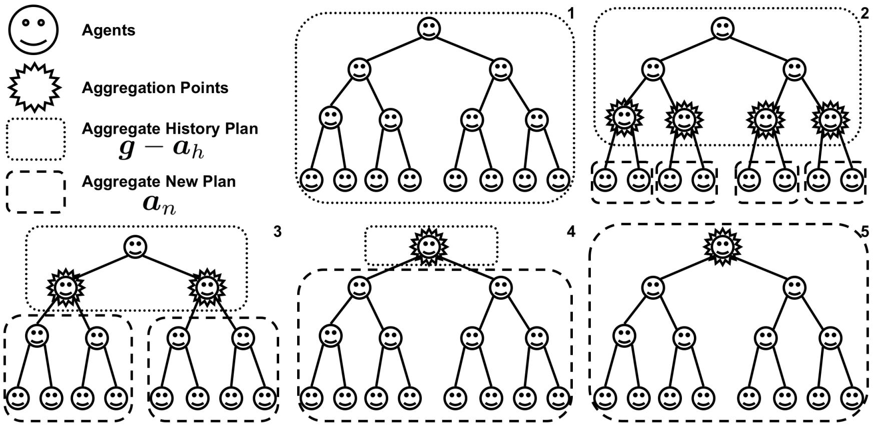

Figure 3 Visualisation of the stabilisation convergence in every aggregation step for the reversed deviations fitness function.

2. *What is the degree of successful reversed deviations?*
3. *How does the number of possible plans influence the stabilisation of EPOS?*
4. *How does the local plan deviations influence the stabilisation of EPOS?*
5. *How does the organisation of the tree overlay influence the stabilisation of EPOS?*

Section 8.1 outlines the simulation environment. Sections 8.2-8.6 illustrate the results for each of the above questions respectively. Section 8.7 summarises the findings.

8.1 Simulation Environment

The simulation scenario assumes a network of interconnected thermostatically controlled appliances. These devices are represented by software agents in a hierarchical virtual structure. The hierarchical structure, that is a tree, is balanced. The heterogeneity of devices over the network is simulated by the top-down approach illustrated in Figure 4.

The simulation configuration starts by considering the following (Level 1 in Figure 4): (i) the average consumption of a generic thermostatically controlled appliance, (ii) the deviation of this average consumption that represents different types of thermostatic devices (refrigerator, water heater etc.) and (iii) the total number of these types. Then a number of average consumption seed values are randomly generated for every type of device (Level 2 in Figure 4). The seed values belong to the range defined by the deviation of average device consumption that represents different types of thermostatic devices. The number of seeds is equal to the number of types of thermostatically controlled appliances in the network. The consumption of a specific type of device also varies in a much smaller proportion compared to the deviation of average consumption among different types (Level 3 in Figure 4).

Moreover, every agent in the network, regardless of which type of device it represents, generates a fixed number of possible plans over a fixed time period (Level 4 in Figure 4).

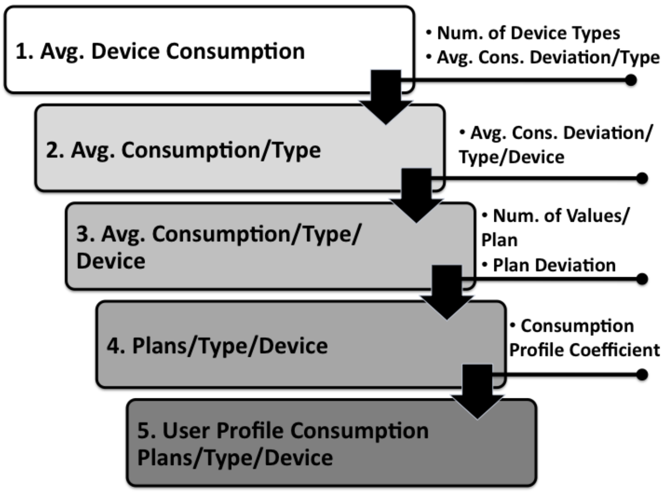
1. Avg. Device Consumption
• Num. of Device Types
• Avg. Cons. Deviation/Type
2. Avg. Consumption/Type
• Avg. Cons. Deviation/ Type/Device
3. Avg. Consumption/Type/ Device
• Num. of Values/ Plan
• Plan Deviation
4. Plans/Type/Device
• Consumption Profile Coefficient
5. User Profile Consumption Plans/Type/Device

Figure 4 The top-down approach of the simulation environment configuration

The generation of the plans is simulated as follows: for every final average consumption of every device, a random sample is generated with size equal to the number of energy values of the plan. The values are estimated between a percentage range of plan deviation from the average consumption value of the plan. For example, a 20% plan deviation from the average consumption of value 10, results in deriving random values from the range [8,12].

Finally, user consumption profiles are simulated (Level 5 in Figure 4). Each device has a high, a medium and a low consumption profile. The consumption profile coefficient multiplies or divides respectively the values of the possible plans. The profiles change cyclically in every round and are initially attributed randomly to devices. This means that every individual device may start with any of the high, medium or low consumption profiles and it follows the same cyclical row in every round.

Four experimental setting are defined for the evaluation of EPOS. Note that in this paper these settings are referred as *Simulation Environments (SimEnv)*. Table 2 outlines the parameters for each of them. Simulation Environment 1 is a small-scale environment of 15 agents for comparing EPOS with the optimum central coordination. Simulation Environment 2 is used as an illustrative environment for the evaluation of the minimum deviations fitness function. It is also used for evaluating the same fitness function under varying the number of possible plans. Simulation Environment 3 does not use consumption profiles and varies the deviation of the generated possible plans. This has the potential to reveal if local deviations affect the global deviations. Finally, Simulation Environment 4 varies the number of children over a series of experiments to evaluate how the tree organisation affects the stabilisation of EPOS. It does not use consumption profiles, similarly to Simulation Environment 3, and keeps the number of possible plans equal to 3.

Note that the units for energy consumption are the same in all the simulation settings. The simulations runs for 100 rounds and the illustrated results are averaged over this running period.

Table 2 Simulation settings used for the experiments and evaluation.

Parameter	SimEnv 1	SimEnv 2	SimEnv 3	SimEnv 4
Num. of Agents	15	3280	3280	3280
Num. of Children	2	3	3	2-5
Num. of Possible Plans	2	2-7	5	3
Num. of Values/Plan	10	10	10	10
Avg. Device Consumption	0.5	0.5	0.5	0.5
Num. of Device Types	3	3	3	3
Avg. Cons. Deviation/Type	0.35	0.35	0.35	0.35
Avg. Cons. Deviation/Type/Device	0.035	0.035	0.035	0.035
Plan Deviation	90%	90%	10%-90%	50%
Num. of Consumption Profiles	3	3	1	1
Consumption Profile Coefficient	2	2	1	1

8.2 Evaluation of Minimising Deviations

The minimum deviations fitness function is evaluated by calculating the standard deviation of the global plans at the end of each round. Figure 5a compares EPOS with the optimum central coordination and greedy selection in matters of the standard deviation in the final global plans. Figure 5b illustrates an example of two groups of global plans with different consumption profiles, one aggregated by the EPOS system and a second aggregated by the greedy system. The results derived by using Simulation Environment 1 and Simulation Environment 2 respectively.

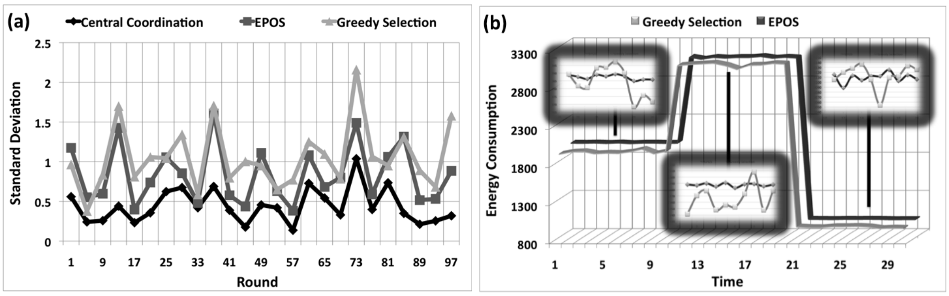

Figure 5 The effect of EPOS in minimising deviations. (a) The standard deviation of the global plans is measured using Simulation Environment 1. EPOS is compared to central coordination and greedy selection. (b) Two groups of global plans with 3 consumption profiles for each one are shown. For illustration purposes, each profile is also depicted with an enlarged figure, indicating the qualitative difference in the stabilisation. The data are collected using Simulation Environment 2.

In Figure 5a, the results collected from the small-scale environment (Simulation Environment 1) indicate that EPOS stabilisation lies between greedy selection and optimal centralised coordination. The average standard deviation for 100 rounds in the centralised coordination is 0.42, whereas, for EPOS and greedy selection the standard deviation is 0.79 and 1.08 respectively.

Figure 5b illustrates the energy consumption of 3 consecutive global plans that consist of 30 time intervals. In every round, or every 10 time intervals, the consumption changes due to the local consumption profiles. The deviations decrease 78.71%, 36.54%

and 73.46% in EPOS compared to the greedy selection. The difference in the global plans is depicted in the three enlarged figures.

Please note that EPOS does not aim to stabilise consumption between different profiles. The global energy consumption changes and this paper focuses on the decrease of positive and negative peak loads in each aggregate plan.

8.3 Evaluation of Reversing Deviations

The reversed deviations fitness function is evaluated by calculating the correlation coefficient r of the global plans that are reversed during simulation runtime. Ideally, optimum reversing corresponds to a correlation coefficient between two global plans equal to '-1'. Figure 6a illustrates the values of the correlation coefficient during the simulation runtime with 70% plan deviation. Figure 6b depicts the reversing effect in two global plans. The data are collected from Simulation Environment 3.

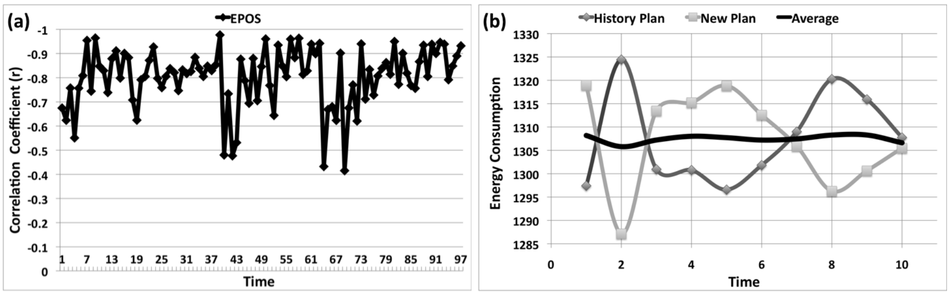

Figure 6 The effect of EPOS in reversing deviations. The data collected by using Simulation Environment 3 (70% plan deviation). (a) The correlation coefficient of the reversed global plans during simulation runtime. (b) An example of reversing deviations between two global plans. Their average is depicted as well.

Figure 6a reveals that there is always a negative correlation between two global plans on which the reversing function is applied. This means that EPOS reacts always positively as is shown in these simulation settings. The average value of the correlation coefficient is '-0.8' in this case.

Figure 6b illustrates the reversing effect in two global plans. The global new plan converges to a mirroring version of the global history plan. The average of these two plans is also drawn to depict the effect of the nearly flat energy consumption.

8.4 Evaluation of Varying the Number of Possible Plans

The number of possible plans is a local parameter of EPOS. The purpose of the following experiments is to examine how the number of local options that the agents have affects the global stabilisation. For this reason, multiple experiments have been run in Simulation Environment 2 and 3 by varying the number (#) of possible plans. Figure 7 illustrates how the effectiveness of the two fitness functions is affected in this case.

Figure 7a illustrates the standard deviation values of the global plans for the minimum deviations fitness function in EPOS and for the system that performs greedy selections. The increase in the number of possible plans influences the stabilisation positively as the

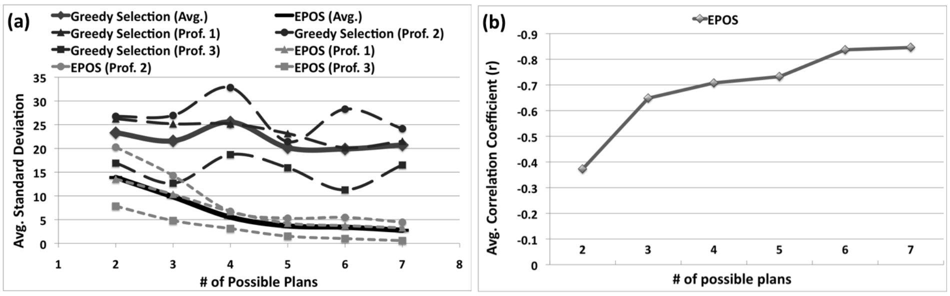

Figure 7 The effect of varying the number (#) of possible plans that the agents generate in EPOS. The data are collected by using Simulation Environment 2 and 3 respectively. (a) The average standard deviation in EPOS and greedy selection for the minimum deviations fitness function. (b) The correlation coefficient in EPOS for the reversed deviations fitness function.

values of the standard deviation decrease. For 2 possible plans the difference in the decrease between the two methods is 9.41, whereas for 7 possible plans it almost doubles to 17.96, denoting almost double improvement. The reason for this positive influence is the increased number of options from which agents can choose, resulting in a higher potential for better stabilisation.

The number of possible plans also influences the reversing effect as Figure 7b reveals. In this case, the similarity between the global history and new global plan is examined by increasing the number of possible plans in each agent. Similarity is measured by calculating the correlation coefficient r of the values of the plans, similarly to the evaluation described in Section 8.3. The values of r decrease from '-0.37' to '-0.85' as the number of possible plans increases in the range of 2-7. In this case, agents have again more options from which they can choose and thus a higher potential to reverse a plan more effectively.

8.5 Evaluation of Varying the Plan Deviation

The simulation environment described in Section 8.1 allows a change in the degree of deviations of the local generated plans. The issue that arises in this case is if local deviations affect global ones. Is higher stabilisation achieved when the local plans vary more from their average or does the opposite relationship hold? In the experiments below, the percentage of deviations from the average is varied between 10% and 90%. Simulation Environment 3 is used for the evaluation in this case. The effect of the two fitness functions in this simulation setting is outlined in Figure 8.

Figure 8a shows a clear linear relationship between the deviations of the local plans from their average and the deviations in the final global plans. The standard deviation increases from 0.76 to 6.63 for EPOS and from 1.50 to 13.42 for greedy selection. Note that the improvement in the stabilisation between EPOS and greedy selection is nearly 50% in this simulation setting.

In contrast, Figure 8b shows some different results. In this case, as the local deviations of the plans from the average increase, the correlation coefficient r of the reversed plans decreases from '-0.75' to '-0.84'. This reveals that more local deviations result in a more successful global reversing effect. These results indicate that without deviations in the local plans, the system is unable to converge. The reversed deviations fitness function requires a search space to reverse the global plan. This search space is the local plan deviations.

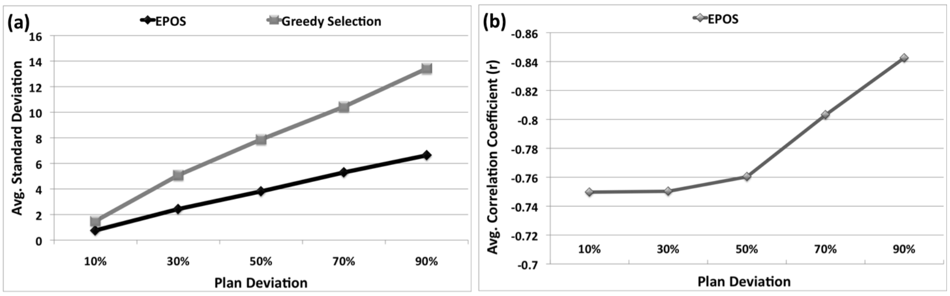

Figure 8 The effect of varying the percentage of plan deviations from the average in EPOS. The data collected by using Simulation Environment 3. (a) The average standard deviation in EPOS and greedy selection for the minimum deviations fitness function. (b) The correlation coefficient in EPOS for the reversed deviations fitness function.

8.6 *Evaluation of Varying the Number of Children*

The stabilisation scheme of EPOS is based on a tree overlay, thus questioning the influence of the tree organisation on its effectiveness is important. Although there are many aspects with this issue, this paper only investigates the variation of the number of children in (static) balanced tree overlays. This parameter is varied from 2 to 5. A higher number of children could not be supported in the current version of the custom simulator that is used as the storing and processing complexity increases exponentially. This is because the number of combinations that are generated and evaluated by the agents increases exponentially as well.

By varying the number of children from 2 to 5, the topology of the tree overlays changes as follows: for 3280 agents, the number of levels in the tree is 12, 8, 7 and 6 respectively. The number of leaves in each topology is 1233, 2187, 1915 and 2499 respectively. These numbers mean that the tree structure changes from being 'long' and 'thin' to 'short' and 'fat'. The intuition behind this experiment is to investigate if a potential convergence of the system to the optimum central coordination exists as the number of children increases. The fact that the tree structure changes to 'short' and 'fat' also indicates that the overlay converges to a star topology. Figure 9 illustrates the stabilisation in the two fitness functions with the results collected using Simulation Environment 4.

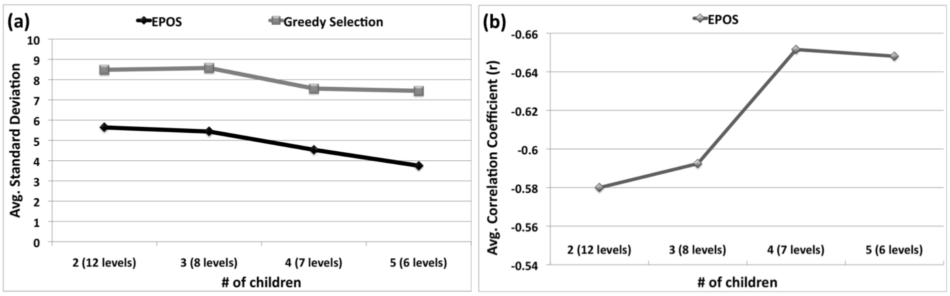

Figure 9 The effect of varying the number (#) of children per agent in EPOS. The data is collected by using Simulation Environment 4. (a) The average standard deviation in EPOS and greedy selection for the minimum deviations fitness function. (b) The correlation coefficient in EPOS for the reversed deviations fitness function.

Figure 9a illustrates the average standard deviation for EPOS and greedy selection. The increase in the number of children indicates a step-by-step decrease of in total 1.89 in the average standard deviation. This value is relatively low, thus definite conclusions can not be reached about how positively the increase in the number of children influences the stabilisation achieved using the minimum deviations function.

Similarly, Figure 9b cannot provide a clear indication if the number of children influences the reversed deviations function in EPOS. In this case, there is again a small improvement, but the difference in the correlation coefficient is 0.07, which is extremely small.

Despite the challenge to explain these results in a deterministic way, our explanation is based on the fact that there is a trade-off between adaptation and optimality in the decision-making. This means that when the number of children increases, the agents generate more combinations and their decisions are much more informed. However, the tree has fewer levels and thus fewer adaptations are performed. Note that in every aggregation step the aggregate plans are used for adapting the combinations. Every aggregation step corresponds to a level in the tree overlay. This explains why adaptations are fewer when the levels of the tree are fewer as well.

8.7 Evaluation Summary

The results from the four simulation environments depicted in Table 2 and their evaluation provide the following answers to the questions set at the beginning of this section:

1. *What is the degree of minimum deviations achieved in EPOS compared to greedy selection and centralised coordination?*

   EPOS always provides higher stabilisation than greedy selection. The results collected from Simulation Environment 2 reveal that deviations decrease in the range of 36.54%-78.71% and the decrease is 50% in Simulation Environment 3. The smaller-scale experimental environment indicates the stabilisation of EPOS (0.79) between central coordination (0.42) and greedy selection (1.08).

2. *What is the degree of successful reversed deviations?*

   EPOS always achieves a negative correlation that ranges from '-0.37' to '-0.85' in the simulation settings of Table 2. The average result of the two global plans corresponds to a 'flat' stabilised plan with its deviations approaching zero.

3. *How does the number of possible plans influence the stabilisation of EPOS?*

   The increase in the number of possible plans increases the stabilisation in both fitness functions. In the minimum deviations, the improvement between EPOS and greedy selection almost doubles. In reversed deviations, the negative correlation also increases reaching the value of '-0.85' in the results collected using Simulation Environment 2.

4. *How does the local plan deviations influence the stabilisation of EPOS?*

   The local plan deviations influence the global ones as the experiments in Simulation Environment 3 reveal. The stabilisation in case of minimum deviations deteriorates linearly as deviations in the local generated plans increases. In contrast, the stabilisation from the reversed deviations fitness function benefits from the local deviations as the correlation coefficient decreases from '-0.75' to '-0.84'.

5. *How does the organisation of the tree overlay influence the stabilisation of EPOS?*

   The experimental settings could not provide a clear answer to this question. There is a positive influence in the effectiveness of stabilisation in EPOS when the number of children increases but it is an extremely small one. This issue must be further investigated in line with our future work that concerns the integration of EPOS with AETOS (Pournaras, Warnier & Brazier 2009*a*).

9 Related Work

Energy management covers a wide range of research areas and problems. Most related work either focuses on the management of providers or consumers. The latter, which is the main focus of this paper and the focus of this related work, is referred as *demand-side management*.

Load-shifting, that appears in most related approaches, is the stabilisation approach that is close to the concept of EPOS. In the work of Stadler, Krause, Sonnenschein & Vogel (2009), cooling devices are assumed to respond to signals from the power grid in order to decrease the energy consumed during peak times or shift their 'on' states to periods with low energy demands. The main concept of this approach is very close to the one of EPOS. However, the whole process is centrally controlled with no interactions among devices. For example, it is not clear what happens in case the devices shift their consumption to another time period resulting in a shift of the peak.

Similar centralised methods are illustrated in other approaches as well, as they guarantee optimal control and optimal load-shifting. For example, Middelberg, Zhang & Xia (2009) propose such an approach that is based on a binary integer programming problem that can be solved with existing methods. The model is applied for the management of a colliery. A similar integer programming model is proposed by Ashok (2006) for the management of steel plants. In contrast to EPOS, these central approaches are mostly suitable for industrial environments rather than a wide large-scale deployment in household consumers.

Load-shifting can be also achieved in the context of energy markets with price-response approaches. In these cases, users actively participate and buy energy for a period of time. Prices change dynamically according to the demand and the load. These load-shifting potentials are explored by Faruqui & George (2005), McDonough & Kraus (2007) and Hopper, Goldman, Bharvirkar & Neenan (2006). These methods require (i) the investigation of the user consumption profiles and (ii) the active participation of the users in the system. These requirements cannot be met in every case. This is why EPOS proposes a fully automated distributed method for stabilising the global resource utilisation, transparent for the user and the other systems of the power infrastructure.

10 Discussion and Open Issues

EPOS introduces a new concept of stabilisation in power systems that goes beyond load-shifting. The *beneficial robust state* introduced in Section 1 enables the system to adapt to various conditions that appear, and react appropriately. EPOS provides two potential solutions towards such beneficial robust states: the proactive minimisation and reactive reverse of the deviations in the global resource utilisation. This reveals that there is space for applying a wide range of energy management policies beyond load-shifting and even beyond the two stabilisation functions that are illustrated in this paper.

Our vision towards autonomic and sustainable power infrastructures is a stabilisation framework based on:

1. the realisation of a generic fitness function.

2. the unification of EPOS with other parts of the power system.

The realisation of a generic fitness function can emerge local selections of energy utilisation to any acquired global utilisation. The second issue means that stabilisation should not only concern thermostatic devices but also other consumption devices. It also means that consumers and providers should not be decoupled but rather both be part of an integrated system. Environmental friendly resource providers, such as wind and solar generators have a very high cost of storing and later distributing their generated energy. Thus, enabling consumers to use this energy when it is generated can lead to significant decrease in the storing costs. These problems are outlined by Middelberg, Zhang & Xia (2009).

Software agents are able to locally control household devices but also participate in higher organisational structures. This is the reason why they have been envisioned as the main computing paradigm for power systems in the book of Rehtanz & Rehtanz (2003). Furthermore, the use of a tree overlay appears to be a very effective communication scheme in this work, allowing stabilisation convergence in the two proposed stabilisation functions. Tree overlays are widely considered in applications that require effective search, decision-making, information dissemination and aggregation. All these operations are crucial in power systems. These benefits motivate the further use of tree overlays in future work. However, guarantying their resilience to failures is crucial when considering large-scale distributed environments. The integration of EPOS with the AETOS (Pournaras, Warnier & Brazier 2009*a*) system, mentioned in Section 3, can potentially bridge this gap.

11 Conclusions and Future Work

This paper describes EPOS, the Energy Plan Overlay Self-stabilisation system. EPOS is based on an agent-based method of stabilisation in the global resource utilisation. The problem and solution are illustrated in the context of the electricity domain and energy management. In EPOS, local software agents (i) control thermostatic devices, (ii) are organised in a tree overlay, (iii) perform aggregation and local decision-making. The global goal of EPOS is to perform self-stabilisation by minimising or reversing the deviations in the global resource utilisation. The main contribution of EPOS is the following:

Hierarchical local coordination achieves emerging convergence of global stabilisation through local knowledge, local decisions and local interactions of local software agents.

Through extensive evaluation in a simulated environment, EPOS appears to be a highly effective stabilisation scheme as it shows significant improvement in the stabilisation of global resource utilisation. This improvement is identified in comparison with greedy selection and to its ability to reverse deviations between two global utilisations. When minimising deviations, EPOS appears 36.54%-78.71% more effective than greedy selection. The reversed deviations function always achieves to correlate two global plans negatively in the range of '-0.37' to '-0.85'. The number of possible plans influences both stabilisation functions positively, as the evaluation reveals. In addition, the local deviations do influence the global ones. When minimising deviations, the increase of the local deviations decreases the stabilisation effectiveness linearly whereas, when reversing deviations, it increases the effectiveness of it. Finally, the tree organisation only indicates a small influence in the collected results, thus further investigation of this issue is part of future work.

These findings reveal that EPOS can be set as a new effective approach for stabilisation in sustainable power systems beyond the load-shifting schemes that have appeared so far. This paper also indicates that there is space for a future generic framework that could allow different stabilisation functions and would unify the systems of consumers and providers towards the sustainability of the whole power infrastructure.

Future work aims in this direction. Some issues that must be further investigated are the integration of AETOS, the Adaptive Epidemic Tree Overlay Service (Pournaras, Warnier & Brazier 2009*a*), with EPOS to increase fault-tolerance and reliability of the tree overlay in dynamic environments. Synchronisation and bootstrapping issues will also be investigated. Finally, evaluation in more realistic contexts is a current focus. EPOS implementation in the AgentScape platform, outlined by Overeinder & Brazier (2004), is in progress. AgentScape is an asynchronous middleware communication system that supports large-scale agent-based systems. Data collected by ZigBee metering devices will be fed into AgentScape for further validation.

Acknowledgements

This project is supported by the NLnet Foundation `http://www.nlnet.nl`.

References

Ashok, S. (2006), 'Peak-load management in steel plants', *Applied Energy* **83**(5), 413 – 424.

Brazier, F. M. T., Kephart, J. O., Parunak, H. V. D. & Huhns, M. N. (2009), 'Agents and service-oriented computing for autonomic computing: A research agenda', *IEEE Internet Computing* **13**(3), 82–87.

Brazier, F. M. T., Ogston, E. & Warnier, M. (2009), The future of energy markets and the challenge of decentralized self-management, *in* 'Proceedings of the Eighth International Workshop on Agents and Peer-to-Peer Computing (AP2PC09)'.

Chakravarti, A. J., Baumgartner, G. & Lauria, M. (2005), 'The organic grid: self-organizing computation on a peer-to-peer network', *IEEE Transactions on Systems, Man, and Cybernetics, Part A* **35**(3), 373–384.

Faruqui, A. & George, S. (2005), 'Quantifying customer response to dynamic pricing', *The Electricity Journal* **18**(4), 53–63.

Fei, Z. & Yang, M. (2007), 'A proactive tree recovery mechanism for resilient overlay multicast', *IEEE/ACM Trans. Netw.* **15**(1), 173–186.

Frey, D. & Murphy, A. L. (2008), Failure-Tolerant Overlay Trees for Large-Scale Dynamic Networks, *in* 'P2P '08: Proceedings of the 2008 Eighth International Conference on

Peer-to-Peer Computing', IEEE Computer Society, Washington, DC, USA, pp. 351–361.

Guo, Y., Li, J. & James, G. (2005), Evolutionary Optimisation of Distributed Energy Resources, *in* 'Australian Conference on Artificial Intelligence', pp. 1086–1091.

Hammerstrom, D. (2002), Part II. Grid Friendly^{TM} Appliance Project, PNNL 17079, Pacific Northwestern National Laboratory.

Hopper, N., Goldman, C., Bharvirkar, R. & Neenan, B. (2006), 'Customer response to day-ahead market hourly pricing: Choices and performance', *Utilities Policy* **14**(2), 126–134.

James, G., Cohen, D., Dodier, R., Platt, G. & Palmer, D. (2006), A deployed multi-agent framework for distributed energy applications, *in* '5th International Joint Conference on Autonomous Agents and Multi-agent Systems (AAMAS 2006),Hakodate, Japan'.

Jennings, N. R. (2001), 'An agent-based approach for building complex software systems', *Commun. ACM* **44**(4), 35–41.

Johansson, S., Davidsson, P. & Carlsson, B. (2000), Coordination Models for Dynamic Resource Allocation, *in* 'Coordination Models and Languages', pp. 182–197.

Leitao, J., Pereira, J. & Rodrigues, L. (2007), Epidemic Broadcast Trees, *in* 'SRDS '07: Proceedings of the 26th IEEE International Symposium on Reliable Distributed Systems', IEEE Computer Society, Washington, DC, USA, pp. 301–310.

Lu, N., Chassin, D. P. & Widergren, S. E. (2005), 'Modeling Uncertainties in Aggregated Thermostatically Controlled Loads Using a State Queueing Model', *IEEE Transactions on Power Systems* **20**(2).

Luis, M. (1999), 'Complex System Modeling: Using Metaphors from Nature in Simulation and Scientific Methods', *BITS: Computer and Communications News, Computing, Information and Communications Divisions* . Available online via: http://www.c3.lanl.gov/ rocha/complex/csm.html (accessed September 30, 2008).

Mazza, P. (2002), 'The Smart Energy Network: Electrical Power for the 21st Century', *Climate Solutions* .

McDonough, C. & Kraus, R. (2007), 'Does dynamic pricing make sense for mass market customers?', *The Electricity Journal* **20**(7), 26–37.

Middelberg, A., Zhang, J. & Xia, X. (2009), 'An optimal control model for load shifting - With application in the energy management of a colliery', *Applied Energy* **86**(7-8), 1266 – 1273.

Moro, G., Ouksel, A. M. & Sartori, C. (2002), Agents and Peer-to-Peer Computing: A Promising Combination of Paradigms, *in* 'AP2PC', pp. 1–14.

Overeinder, B. J. & Brazier, F. M. T. (2004), Scalable Middleware Environment for Agent-Based Internet Applications, *in* 'Proceedings of the Workshop on State-of-the-Art in Scientific Computing (PARA'04)', Vol. 3732 of *Lecture Notes in Computer Science*, Springer, Copenhagen, Denmark, pp. 675–679.

Pournaras, E., Warnier, M. & Brazier, F. M. T. (2009*a*), Adaptive Agent-based Self-organization for Robust Hierarchical Topologies, *in* 'ICAIS '09: Proceedings of the International Conference on Adaptive and Intelligent Systems', IEEE. (to appear).

Pournaras, E., Warnier, M. & Brazier, F. M. T. (2009*b*), A distributed agent-based approach to stabilization of global resource utilization, *in* 'Complex, Intelligent and Software Intensive Systems, International Conference', IEEE Computer Society, Los Alamitos, CA, USA, pp. 185–192.

Rehtanz, C. & Rehtanz, C. (2003), *Autonomous Systems and Intelligent Agents in Power System Control and Operation (Power Systems)*, SpringerVerlag.

Rhea, S., Geels, D., Roscoe, T. & Kubiatowicz, J. (2004), Handling churn in a dht, *in* 'ATEC '04: Proceedings of the annual conference on USENIX Annual Technical Conference', USENIX Association, Berkeley, CA, USA, pp. 10–10.

Shaw, R., Attree, M., Jackson, T. & Kay, M. (2009), 'The value of reducing distribution losses by domestic load-shifting: a network perspective', *Energy Policy* **37**(8), 3159 – 3167.

Stadler, M., Krause, W., Sonnenschein, M. & Vogel, U. (2009), 'Modelling and evaluation of control schemes for enhancing load shift of electricity demand for cooling devices', *Environmental Modelling & Software* **24**(2), 285 – 295.

Strbac, G. (2008), 'Demand side management: Benefits and challenges', *Energy Policy* **36**(12), 4419 – 4426. Foresight Sustainable Energy Management and the Built Environment Project.

Tan, G., Jarvis, S. A., Chen, X. & Spooner, D. P. (2006), 'Performance Analysis and Improvement of Overlay Construction for Peer-to-Peer Live Streaming', *Simulation* **82**(2), 93–106.

Theocharopoulou, C., Partsakoulakis, I., Vouros, G. A. & Stergiou, K. (2007), Overlay networks for task allocation and coordination in dynamic large-scale networks of cooperative agents, *in* 'AAMAS '07: Proceedings of the 6th international joint conference on Autonomous agents and multiagent systems', ACM, New York, NY, USA, pp. 1–8.

Tianfield, H. & Unland, R. (2005), 'Towards self-organization in multi-agent systems and Grid computing', *Multiagent Grid Syst.* **1**(2), 89–95.

Wirsing, M. & Holzl, M. (2006), Software Intensive Systems, Technical report, Beyond the Horizon Thematic Group 6 on Software Intensive Systems.