Distributed Optimization of Energy Costs in Manufacturing using Multi-Agent System Technology

Tobias Küster, Marco Lützenberger, Daniel Freund, and Sahin Albayrak
DAI-Labor, Technische Universität Berlin
Ernst-Reuter-Platz 7, 10587 Berlin, Germany
Telephone: +49 (0)30 - 314 74000, Fax: +49 (0)30 - 314 74003
{*tobias.kuester*|*marco.luetzenberger*|*daniel.freund*|*sahin.albayrak*}*@dai-labor.de*

***Abstract*—While widely endorsed, the increased provision of electricity from renewable sources comes with the concern that energy supply will not be as reliable in the future as it is today, due to variations in the availability of wind and solar power. However, fluctuations in energy supply also give rise to volatility of the price for short-term energy procurement, and therefore bear the opportunity to save costs through shifting energy consumption to periods of low market prices. In a previous work, we presented an evolution-strategy-based optimization of production schedules with respect to day-ahead energy price predictions, yielding good results, but – being a stochastic optimization – not always arriving at the best solution. In this paper, we extend our framework by agent-based mechanisms for distribution and parallelization of the optimization, to increase scalability and reliability of the approach.**

***Keywords*-multi-agent systems; production planning; energy efficiency**

I. Introduction

In recent years, environmental-friendly production of energy and goods has gained more and more importance, with both customers and governments demanding for "green" production and increased provision of renewable energies. However, critics claim that by relying more on variable and volatile energy sources such as wind and solar power, the future energy supply will not be as stable as today, with variations in available energy over time [1].

However, these fluctuations in energy supply also give rise to the volatility of the price for short-term energy procurement, e.g., via the *European Energy Exchange* (EEX) [2], taking into account the predicted feed-in of wind energy, the current oil price and previous EEX results. This bears the opportunity for manufacturing industries to decrease production costs by shifting energy-intense production steps to periods of high wind and solar energy availability – and thus low energy prices – at the same time also fostering the use of environmental-friendly renewable energy sources. Today, demand-response mechanisms like this are only beginning to be implemented in the industry, but are expected to gain currency in some countries, as fluctuation of short term energy price becomes more distinctive [3].

In previous work [4], we presented an approach for optimizing a production schedule with respect to day-ahead energy price predictions, using evolution-strategy. The optimization yields good results most of the time; but being a stochastic algorithm, it does not always arrive at the best solution, getting stuck in local optima instead. Thus, to find the global optimum, the optimization needs to be run more than once on a specific process graph. However, depending on the complexity and granularity of the process graph, the optimization can be a time consuming process, and with the restricted time frame available between receiving energy price forecasts and the end of the bidding period, it becomes necessary to distribute and to parallelize the optimization procedure.

In this paper, the optimization framework is extended by agent-based mechanisms. Using a simple interaction protocol, the optimization can transparently be distributed to multiple servers to increase scalability and reliability, as individual optimization runs are executed in parallel. As we evaluated the results, we found out that even very few runs, or "populations", are sufficient to reliably arrive at a near-optimal solution, without increasing the time to find these results significantly.

We start this paper by outlining the principles of our so far work (Section II) and proceed by describing on how the multi-agent system paradigm can be used for the purpose of manufacturing process optimization with respect to dynamic energy procurement (Section III). We proceed with the evaluation of the optimization framework (Section IV) and subsequently compare our approach to related work (Section V). Finally, we wrap up with a conclusion (Section VI).

II. Previous Work

In the following, we will provide a short recapitulation of our work so far [4]. We start by presenting the production process meta model and proceed with details on the simulation and optimization algorithms. Subsequently, we explain how the system has been implemented.

A. Production Process Meta-Model

In our approach, any production process is modeled as a bipartite graph of *activities* and *resources*, similar to a

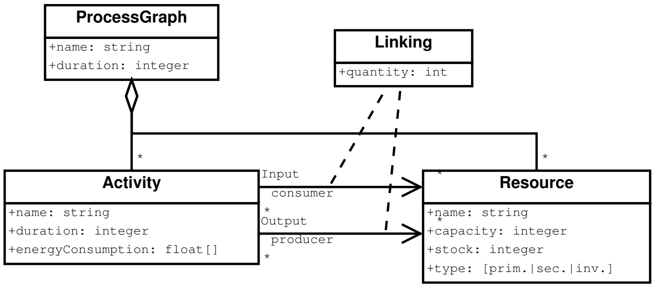

Figure 1. Process meta-model [4]

Petri net [5]. Activities have input and output resources, a duration of multiple atomic time steps, and a variable energy consumption over this duration, which can also be negative for energy sources and storage devices. Resources have a minimum and maximum capacity and a current stock. Resources are subdivided into primary resources (e.g., raw materials, intermediate products), secondary resources (e.g., pressurized air, gas, waste heat) and inventory resources (e.g., machines). The meta-model is shown in Figure 1.

Since electrical energy is the main concern of the optimization, it is not modeled as a resource but treated separately. Unlike other resources, electrical energy is available in (for all practical purposes) unlimited quantity and at a variable price, based on the energy market.

When an activity is executed, its input resources are consumed and its output resources are produced, and it will add to the overall energy consumption of the production process. Primary and inventory resources are consumed/allocated in the first step and produced/de-allocated in the last step of the activity's execution; secondary resources are produced or consumed in every step of the respective activity.

Using this simple meta-model, a wide range of production processes can be modeled. At the same time it facilitates the simulation and optimization of energy consuming activities in other domains, such as e.g., the utilization and charging schedules of electric vehicles.

B. Simulation and Optimization

The purpose of the optimization is to find the best possible *production schedule* for a given process model. In the implementation at hand, cost optimization is conducted mainly on the basis of day-ahead price forecasts, e.g., for the EEX electricity spot market. The optimization consists of three major steps: (1) The simulation of a given production schedule, (2) measuring the quality of that simulated schedule, and (3) finding the schedule with the highest quality.

1) Simulation: The simulation of a production schedule keeps track of the resource stocks and the energy consumption in each step of the simulation for the duration of the process, checking which activities are to be started, which activities are still running, and which activities are to be ended in the current step, producing and consuming resources and energy accordingly.

Concerning energy consumption and cost, two parameters of the simulation can be adjusted to reflect different determining factors: First, an *energy price curve* can be provided, for instance from the day-ahead energy market. Second, a *base energy* level can be specified, being the amount of energy the facility acquires via a flat fee. Energy consumption up to this level has already been paid for, so the *energy price curve* does not apply for it.

2) Quality Measurement: The *quality* of a production schedule p is determined by the inverse of its *defect*, which is the weighted sum of the total energy costs ($p_e \cdot w_e$) and the total over- and undershootings of the several resources' capacities ($p_{r,d} \cdot w_{r,d}$) over all steps of the simulation.

$$defect(p) = p_e \cdot w_e + \sum_{r \in \{p,s,i\}} \sum_{d \in \{l,h\}} p_{r,d} \cdot w_{r,d}$$

Different weights ($w_{r,d}$) can (and should) be used for resource stocks being too low and those being too high ($d \in \{l,h\}$) and for the different kinds of resources ($r \in \{p,s,i\}$).

Production schedules, which exceed the maximum or minimum capacities of a resource are not discarded, but are merely given a lower quality rating. For many optimization algorithms this is necessary in order to overcome local optima. For example, a schedule might be highly improved by swapping two activities. During this swap, there may be a phase in which the activities will both occupy a shared resource, but the benefits of the new schedule may be big enough to compensate for this temporary defect.

3) Optimization: For finding an energy- and cost-efficient production schedule, making best use of a given energy price curve, we use Evolution Strategy [6], which is similar to genetic algorithms.

As the name implies, Evolution Strategy is inspired by natural evolution: Using a $(\mu/\rho + \lambda)$ strategy, an initial "population" of μ individuals is generated. In the system at hand, each individual represents one production schedule. Based on these μ "parents", λ "offspring" are generated by recombining a random selection of ρ parents and slightly "mutating" the result. Finally, the quality of each of the parents and offspring is determined and the μ best individuals are selected to be the parents of the next generation. This process is repeated until a satisfactory production schedule is found.

The initial population is created by a very simple scheduler, aligning production activities as long as and as early as the primary resources permit, or until a desired quantity of products has been produced. In order to mutate an individual, either a random activity is inserted into or removed from the schedule, or one or more activities are moved to another position in the schedule, thus being executed earlier or later.

C. Optimization Framework and Tools

The approach is currently being evaluated using a prototypical implementation, which can be used to design the manufacturing process to be simulated, to configure and to run the actual optimization, and to visualize the results.

The process meta-model and a simple graphical editor for creating and configuring process models have been implemented as an extension to the Eclipse development environment. Following the usual notation for Petri nets, activities are represented by rectangles and resources by circles (see the example in Section IV).

Regarding the optimization, a generic optimization framework has been created, which can be used for optimizing different domains using different optimization algorithms. The actual Evolution Strategy algorithm as well as the process model domain have been implemented as plug-ins for this framework, targeting the optimization of comparably complex and heterogeneous industrial manufacturing processes. With respect to other domains, such as charge optimization of a large numbers of electric vehicles, other algorithms may be expedient.

For the manufacturing domain, the system features a large domain-specific area, providing controls for configuring the simulation and optimization (e.g., the energy price curve to use) and for showing the best production schedule found so far in a Gantt-like diagram. Once the optimization has come to an end, additional charts are available, showing the energy consumption and stocks of individual resources over the course of the simulation, as well as the development of these charts over the course of the entire optimization as a three-dimensional plot. Finally, the optimized process plan can be saved to file.

III. Agent-based Optimization of Manufacturing Schedules

While Evolution Strategy yields good results most of the time, it is also possible, as with other stochastic local search algorithms, that the optimization gets stuck in local optima. To increase the chances of arriving at a solution close to the global optimum, the optimization should be applied on more than one “population”, and since the individual populations are independent of each other, they can easily be parallelized.

As described in the introduction, we combine the optimization framework with agent-oriented technologies to distribute and parallelize the optimization, and find global optima within reasonable time.

In the following we describe a simple interaction protocol, which makes a number of optimization servers (i.e., agents conducting the optimization) available to more than one optimization client. Further, we explain how the protocol was implemented using the JIAC V agent framework.

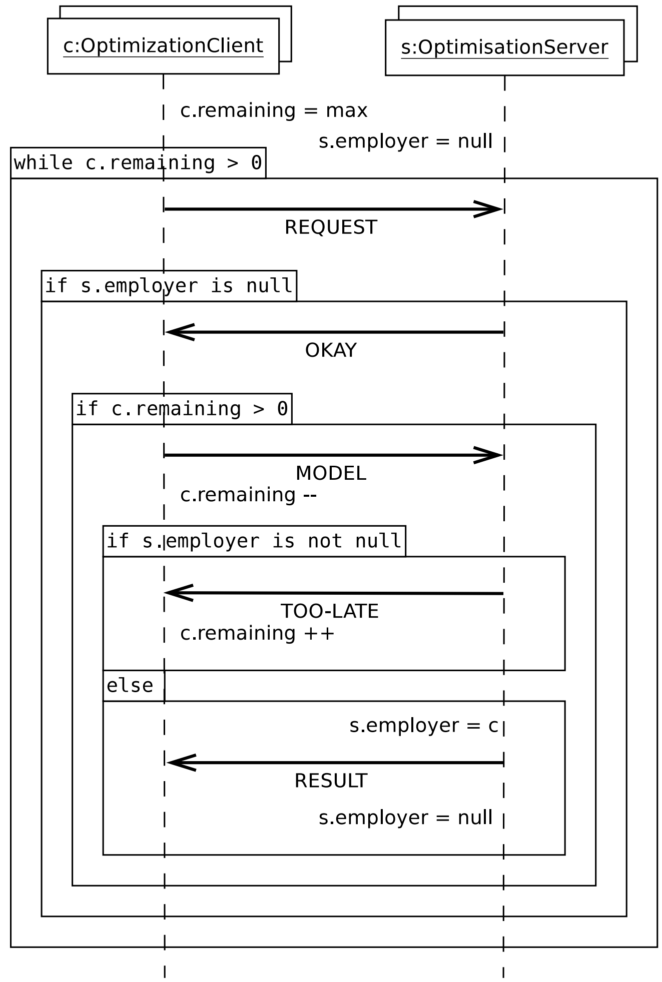

Figure 2. Interaction protocol used in the distributed optimization.

A. Interaction Protocol

Using the simple interaction protocol outlined in this section, each of the populations of a $(\mu/\rho+\lambda)$ optimization can be distributed to another agent. Since each run of the optimization, or each population respectively, is independent from the others, this does not introduce any noteworthy communication overhead. An interaction diagram of the protocol is shown in Figure 2.

The roles in the protocol are:

- *optimization client*, requesting an optimization
- *optimization server*, conducting the optimization

Obviously, there should be more than one optimization server agent for the distribution to provide any benefit at all, and there may be multiple clients, as well, sharing those servers. In the following we will describe the several interactions comprised in the protocol.

1) The protocol starts with a client broadcasting a REQUEST message to all the servers.
2) Each server receiving the message checks whether it

 ISBN: 978-1-61208-189-2

already has an “employer”, i.e., whether it is currently running an optimization. If not, it replies with an OKAY message.

3) The client received the OKAY message, and if it still requires the server (i.e., if there have not been enough replies from other servers yet), it replies by sending the actual MODEL to be optimized to that server. The number of remaining optimization runs is reduced.
4) On receiving the MODEL message, the server will check again whether it already has an employer, as in the case of multiple clients, it might have sent OKAY messages to other clients, which may already have sent their MODEL messages.
   - If so, the server replies with a TOO LATE message. The client received this messages and corrects the number of remaining optimizations.
   - Otherwise, the server accepts the client as its new employer and starts the optimization run, and finally sends a message holding the RESULT back to the client.
   - At any time, the client can send an ABORT message, stopping the optimization.
5) The client continues sending out REQUEST messages until the desired number of optimizations has been conducted.

B. The JIAC V Multi-Agent Framework

JIAC V (Java Intelligent Agent Componentware, Version 5) is a Java-based multi-agent development framework and runtime environment [7]. Among others, JIAC features communication, tuple-space based memory, transparent distribution of agents and services, as well as support for dynamic reconfiguration in distributed environments, such as component exchange at runtime. Individual JIAC agents are situated within Agent Nodes, i.e., runtime containers, which also provide support for strong migration. The agents' behaviors and capabilities are defined in a number of so-called *Agent Beans*, which are controlled by the agent's life cycle.

C. Implementation

The protocol has been implemented by means of two JIAC Agent Beans, namely the *Optimization Client Bean* and *Optimization Server Bean*. Just like the optimization framework introduced in Section II, the Agent Beans were kept generic so that they – and thus the protocol – can just as well be used with domain-models other than the one presented in this work, and even with different optimization algorithms.

Using asynchronous messaging, the implementation with JIAC (or a similar multi-agent framework) has some advantages over traditional approaches using remote procedure calls or web services:

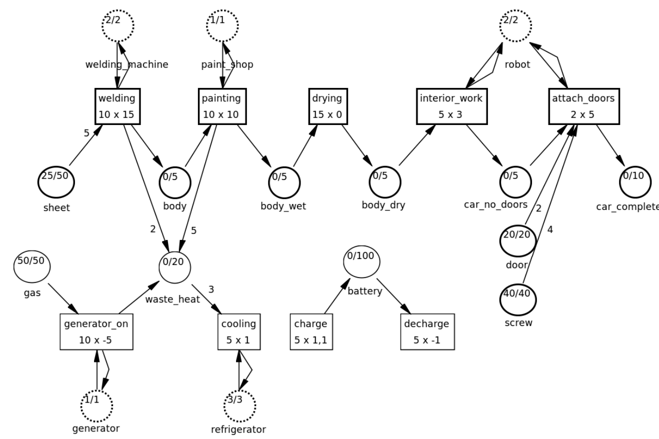

Figure 3. Example process: automobile construction (simplified)

- Both the Client Nodes and the Server Nodes can be distributed to any computer in the local network, with no need to configure IP addresses or ports.
- With each JIAC agent running in a separate thread, a node with multiple agents can be deployed to a multi-core server computer, and will automatically make ideal use of the several CPUs.
- Optimization procedures can be aborted ahead of time by sending the appropriate message. Similarly, the servers can send back intermediate results, to provide a trend for long-running optimizations.

Besides advantages over alternative means of distributed systems, the agent-based approach performs as expected with respect to previous local optimization. It yields good results in reasonable time and the variability of results decreases with an increased number of populations.

IV. EVALUATION

We evaluated our optimization algorithm and the benefits of distribution and parallelization using a fictional process, a production goal of five completed cars, a hill-shaped energy price curve and an evolution strategy with $\mu = 3$ and $\lambda = 8$.

A. Example Process

For the evaluation, a simple, fictional example process inspired by automotive industry was used (Figure 3). The process starts with two energy-intensive activities, which induce lots of waste-heat besides their primary production purpose: welding and painting the car chassis. Once the paint has dried, some interior works are performed, and finally the doors are attached to the chassis. For each of the intermediate products, a specific primary resource is created. The resulting production process graph is supplemented with utility activities and resources such as cooling, on-site electricity storage and a gas-powered cogeneration unit. The latter two elements can be used to temporarily decrease the grid energy consumption, but costs for the

 ISBN: 978-1-61208-189-2

corresponding increase in gas consumption will in turn add to the production schedule's penalty.

While the process surely is very simplified, it comprises most of the aspects that can be realized in the process model, for example

- the modeling of the basic production chain,
- one kind of machinery being used for two activities,
- the use of resources associated with a cost, or
- cooling facilities and other supporting processes.

B. Optimization Results

To asses the benefit of distribution and parallelization, the example process was optimized several times with different numbers of populations. The size of populations ranged from one to thirteen, and ten runs of the optimization were performed in each case. The results are shown in the logarithmic plot in Figure 4.

As can be seen, using only one population, the quality of the optimized process plan varies greatly. While there are some results with near-optimal quality, many populations apparently get stuck in local optima, and obtain a low overall-quality. For up to four populations, results start to look better, but are still noticeably scattered. For five and more populations, the results become reliable, with almost each optimization run resulting in near-optimal quality.

It may be noticed that the maximum quality reached – around 0.05 – is still far from the theoretically possible 1.0. The reason for this is that energy costs, no matter whether they could be improved any further, still add to the defect of the process. Thus, with minimum energy costs of around 20 (in no specific currency), the quality can not be much greater than 0.05.

Also to be noted is the gap in quality between around 0.015 and 0.045. This gap separates results, which still have resource conflicts, and those merely suffering from less-than-optimal energy costs. In the evaluation, the weight of resource conflicts was set to add greatly to the overall result's defect.

Further, we noted that there is little to none correlation between the time an individual optimization run takes, and the resulting quality (see Figure 4), i.e., a quick optimization run can yield a very good result, while a long-running optimization does not guarantee to bring a good result. Thus, one possibility to improve the performance could be to start a large number of optimization in parallel, and to abort the remaining optimization runs once the first few results to choose from have arrived.

V. Related Work

Industry has long since discovered, that process optimization is able to increase revenues significantly. As a result, there are many sophisticated applications available today. In this section we outline the latest optimization tools. Due to the broad range of existing approaches we focus our survey on works, which influenced us the most. We conclude this section by discussing significance of our work against the backdrop of contemporary applications.

Highly interesting for our work is the approach of *Santos et al.* [8], as it puts focus on energy related criteria. Yet, as opposed to our objective, the aim of *Santos et al.* is to reduce energy consumption in general, while we try to adapt our manufacturing schedules to a given objective function. *Bernik et al.* [9] developed a similar approach, although they do not account for energy criteria. The approach is capable to propose manufacturing schedules, which are able to fulfill a given production target. In addition to the manufacturing schedule, resource requirements are calculated and assigned to the production depots. *Schreiber et al.* [10] describe a similar application, which optimizes manufacturing schedules with respect to a specified given production target. As opposed to the approach of *Bernik et al.*, the application is able to calculate so called lot-sizes, which are defined as the number of pieces, which are processed at the same time at one workplace with one-off (time) and at the same costs investment for its set up [10].

In addition to the above mentioned academic works, there are many commercial software packages available.

The *Siemens Plant Simulation Software* [11], for instance, is a commercially available software, which facilitates the optimization of production systems and controlling strategies. Business- and logistic- processes may be supported as well. Processes are captured in compliance with an object oriented domain model. The *SIMUL8* framework [12], *Arena* [13] and *GPSS/H* [14] provide similar features and are able to simulate entire production processes, from warehouse capacities, to equipment utilization, to logistics-, transportation-, military- and mining applications. Beyond that, *SIMUL8* additionally accounts for real life requirements, such as maintenance intervals and shift patterns. Other types of software packages as for instance *Simio* [15] and *ShowFlow* [16] do not explicitly focus on the optimization of production processes, but on their visualization. For this purpose, most of the mentioned applications apply sophisticated 3D engines.

Thus far, the mentioned works are focused on the optimization of production processes. Yet, over the last years, the idea of general purpose frameworks emerged. Instead of focusing on a particular domain or problem, general purpose frameworks are able to optimize processes in general. Foundation to these frameworks is a generic meta-model, which is able to capture process structures.

PACE [17] and *AnyLogic* [18] for instance feature an arbitrary level of detail for process design. While *PACE* uses hierarchically arranged *High-Level-Petri-Nets* for this purpose, *AnyLogic* applies an object-oriented meta-model to capture process structures. *SLX* [19] takes a layered approach to process modeling. Most commonplace processes are handled in *SLX*'s upper layers, while unique

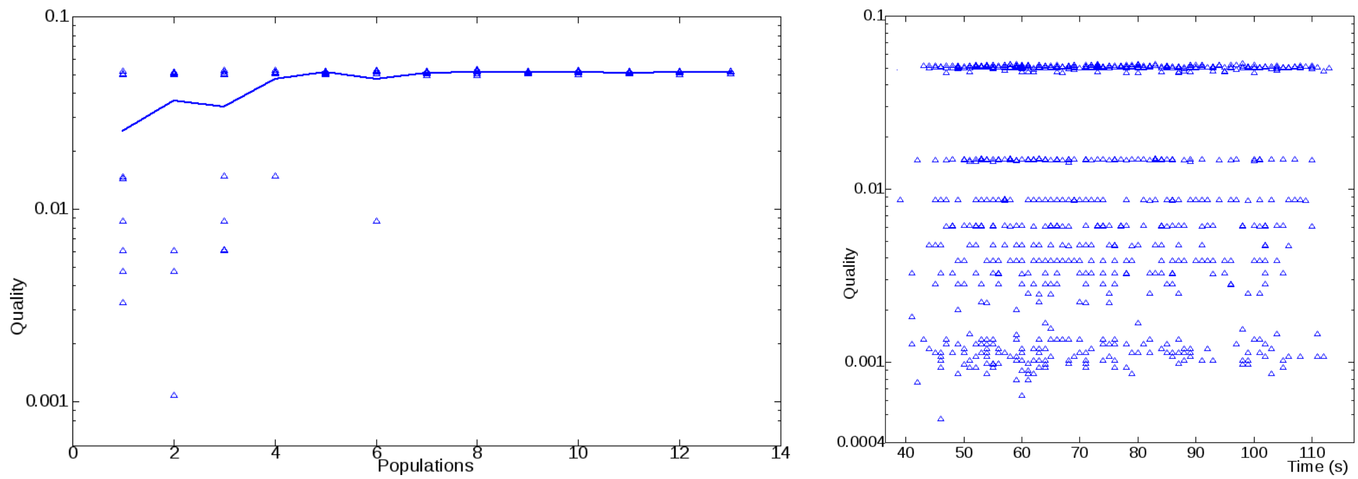

Figure 4. Left: Correlation of number of populations to expected result quality. The graph indicates average quality values. Right: Correlation of time of optimization run to result quality.

and more complex problems can be captured with *SLX*'s lower layers. The *Microsaint* [20] package totally avoids hierarchically structures and facilitates readability as well as easy comprehensibility. The framework entirely relies on flow charts as meta process language. In most analyzed frameworks, process design is usually supported by visual editing tools. The *ADONIS* framework [21] for instance provides an impressive graphical editor for the design and manipulation of the examined process system.

Finally, we analyzed tools which have been developed for similar optimization problems, but for domains different from manufacturing. Business processes for instance have a striking resemblance to manufacturing processes and as there are optimization frameworks for business processes, we want to mention the most prominent members of this realm as well.

To start with, *ProcessModel* [22] is a business process optimization software, which supports optimization from problem analysis to efficiency evaluation. The tool is able to visualize many aspects such as money savings or the efficiency of analyzed processes to serve customers. A similar application is *SIMPROCESS* [23]. In addition to the capabilities of *ProcessModel*, *SIMPROCESS* is able to handle hierarchical process structures and comes along with a set of sophisticated tools for the process design. Both applications apply means of simulation in order to verify optimized processes and to estimate their overall quality.

In this section, we gave a comprehensive overview on state of the art concepts and applications. To sum up; our idea of optimizing production with respect to dynamic energy tariffs is adopted by none of the examined applications. Further, we can state that energy related criteria are currently not comprehensively covered by state-of-the-art solution, as only the approach of *Santos et al.* facilitates such factors. We learned that evolutionary algorithms can be used to increase the performance of optimization algorithms and thus applied such principle [6]. Finally, the *AnyLogic* framework convinced us to apply mechanisms of distributed computing, namely the agent paradigm.

VI. Conclusion and Future Work

This paper proposed the use of a multi-agent system for the distributed process optimization with respect to energy consumption. A set of software agents has been designed and deployed to a physically distributed client-server-architecture, implementing an interaction protocol for the dynamic coordination of optimization processes and result aggregation. Feasibility and performance of the system were verified by using an exemplary manufacturing process. Results show that an increased number of parallel populations significantly decreases the variability of simulation outcomes and the probability of receiving a suboptimal result. Due to parallelization, the duration of the optimization did not vary noticeably with an increased number of populations.

In this study, only the distribution of many equivalent optimization jobs to several agents is evaluated. However, the agent-based optimization system is designed to accomplish variations between the individual jobs, e.g., using different settings for the optimization. Furthermore, diverse optimization strategies besides Evolution Strategy can be introduced as plug-ins to the system. As an example, there may be distinct independent sub-problems in the manufacturing site's overall process graph, such as the optimization of manufacturing processes on the one hand and the charging of forklift trucks on the other hand, where distinct optimization algorithms perform better or worse.

Future work will be dedicated to the evaluation of quality and performance gains through the aforementioned extensions to the system; namely diversification of population parameters (e.g., number of parents and offspring), diversifi-

 ISBN: 978-1-61208-189-2

cation of optimizing algorithms, and breaking down process graphs into sub-problems to distribute them among different agents.

Further, it is our intention to exploit the agent-paradigm stronger. In this work, we focused on the aspect of distribution and neglected other important characteristics of software agency, as for instance autonomy, pro- or reactiveness. The reason for this decision is simple, as we see the contribution of this particular paper in the distributed structure of our formerly centralized solution. For the future, we want to use this distributed structure as a basis for further extensions. Having this objective in mind, we aspire an autonomous energy procurement of additional energy and also an autonomous brokering of energy surpluses, based on predicted energy demands. In addition, we want to enhance our distributed optimization by load balancing capabilities. Optimization clients will be aware of the local load and be able to migrate to machines with free capacity.

VII. Acknowledgments

This work is funded by the *Federal Ministry of Economics and Technology* under the funding reference number 0327843B.

References

[1] H. Holttinen, P. Meibom, A. Orths, F. van Hulle, B. Lange, M. O'Malley, J. Pierik, B. Ummels, J. O. Tande, A. Estanqueiro, M. Matos, E. Gomez, L. Sder, G. Strbac, A. Shakoor, J. Ricardo, J. C. Smith, M. Milligan, and E. Ela, "Impacts of large amounts of wind power on design and operation of power systems, results of iea collaboration," *Wind Energy*, vol. 14, no. 2, pp. 179–192, 2011.

[2] European Energy Exchange AG, "EEX homepage," 2012. [Online]. Available: http://www.eex.com

[3] J. Torriti, M. G. Hassan, and M. Leach, "Demand response experience in europe: policies, programmes and implementation," *Energy*, vol. 35, no. 4, pp. 1575–1583, 2010. [Online]. Available: http://centaur.reading.ac.uk/23573/

[4] T. Küster, M. Lützenberger, and D. Freund, "An evolutionary optimisation for electric price responsive manufacturing," in *Proceedings of the 9th Industrial Simulation Conference, Venice, Italy*, S. Balsamo and A. Marin, Eds. EUROSIS-ITI, June 2011, pp. 97–104.

[5] T. Murata, "Petri nets: Properties, analysis and applications." in *Proceedings of the IEEE*, april 1989, pp. 541–580.

[6] I. Rechenberg, *Evolutionsstrategie : Optimierung technischer Systeme nach Prinzipien der biologischen Evolution*, ser. Problemata. Stuttgart-Bad Cannstatt: Frommann-Holzboog, 1973.

[7] B. Hirsch, T. Konnerth, and A. Heßler, "Merging agents and services – the JIAC agent platform," in *Multi-Agent Programming: Languages, Tools and Applications*, R. H. Bordini, M. Dastani, J. Dix, and A. El Fallah Seghrouchni, Eds. Springer, 2009, pp. 159–185.

[8] A. Santos and A. Dourado, "Global optimization of energy and production in process industries: a genetic algorithm application," *Control Engineering Practice*, vol. 7, no. 4, pp. 549–554, 1999.

[9] I. Bernik and M. Bernik, "Multi-criteria scheduling optimization with genetic algorithms," in *Proceedings of the 8th WSEAS International Conference on Evolutionary Computing*. Stevens Point, Wisconsin, USA: World Scientific and Engineering Academy and Society (WSEAS), 2007, pp. 253–258.

[10] P. Schreiber, P. Vazan, P. Tanuska, and O. Moravcik, "Production optimization by using of genetic algorithms and simulation model," in *DAAM International Scientific Book 2009*, B. Katalinic, Ed. DAAM International, 2009.

[11] Siemens, "Plant simulation — plant, line and process simulation and optimization," Project Broschure, 2010, siemens Product Lifecycle Management Software Inc. [Online]. Available: http://www.plm.automation.siemens.com/en_us/Images/7541_tcm1023-4957.pdf

[12] K. Concannon, M. Elder, K. Hunter, J. Tremble, and S. Tse, *Simulation Modeling with SIMUL8*, 4th ed. Visual Thinking International Ltd., 2003.

[13] M. D. Rossetti, *Simulation Modeling and Arena*, 1st ed. Wiley, 2009.

[14] R. C. Crain, "Simulation using GPSS/H," in *Proceedings of the 29th Winter Simulation Conference*, December 1997, pp. 567–573.

[15] J. A. Joines and S. D. Roberts, *Simulation Modeling with SIMIO: A Workbook*. Simio LLC, 2010.

[16] I. S. Solutions, "The ShowFlow website," 2011. [Online]. Available: http://www.showflow.com/

[17] B. Eichenauer, "Optimizing business processes using attributed petri nets," in *Proceedings of the 9th Symposium about Simulation as Commercial Decision Help*, March 2004, pp. 323–338.

[18] P.-O. Siebers, U. Aickelin, H. Celia, and C. W. Clegg, "Understanding retail productivity by simulating management practices," in *Proceedings of the Eurosim 2007, Ljubljana, Slovenia*, 2007, pp. 1–12.

[19] J. O. Henriksen, "SLX: The x is for extensibility," in *Proceedings of the 32nd Winter Simulation Conference*, 2000, pp. 183–190.

[20] D. W. Schunk, W. K. Bloechle, and K. R. L. Jr., "Micro saint: Micro saint modeling and the human element," in *Proceedings of the 32nd Winter Simulation Conference*, 2002, pp. 187–191.

[21] S. Junginger, H. Kühn, R. Strobl, and D. Karagiannis, "Ein Geschäftsprozessmanagement-Werkzeug der nächsten Generation – ADONIS: Konzeption und Anwendungen," *Wirtschaftsinformatik*, vol. 42, no. 5, pp. 392–401, 2000.

[22] ProcessModel, Inc., "The ProcessModel website," 2011. [Online]. Available: http://www.processmodel.com/

[23] CACI, "The SIMPROCESS website," 2011. [Online]. Available: http://simprocess.com/

