Distributed Evolutionary Optimisation for Electricity Price Responsive 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
{tobias.kuester, marco.luetzenberger, daniel.freund, sahin.albayrak}@dai-labor.de

***Abstract*—With the recent uptake in renewable energies, such as wind and solar, often comes the apprehension of unreliable energy supply due to variations in the availability of those energy sources, also resulting in severe fluctuations in the price of electricity at energy exchange spot markets. However, those fluctuations in energy costs can also be used to stimulate industry players to shift energy intense processes to times when renewable energies are abundant, not only saving money but at the same time also stabilising the power grid. In previous work, we presented a software framework that can be used to simulate and optimise industrial production processes with respect to energy price forecasts, using a highly generic meta-model and making use of evolutionary algorithms for finding the best process plan, and multi-agent technology for distributing and parallelising the optimisation. In this paper, we want to wrap up our work and to aggregate the results and insights drawn from the EnEffCo project, in which the system has been developed.**

***Keywords*—production planning; energy efficiency; evolutionary computing; multiagent systems**

I. Introduction

The transition towards sustainable energy provision must be regarded as one of the most urgent global challenges of the upcoming decades. Regulatory and technological solutions must be developed to pursue the goal of decreasing the environmental impact, and supplying reliable, secure and affordable energy nevertheless. The amount of integrated Renewable Energy Sources (RES) on the one hand and the enhancement of primary energy efficiency on the other are crucial dimensions for a successful transition. However, without the implementation of intelligent technological and regulatory mechanisms, intermittency of regenerative sources will affect primary energy efficiency of global energy networks and markets.

In this paper, building upon previous work [1], we present a system for simulating production processes and for optimising those processes w.r.t. local energy production and variable energy prices.

From an economic point of view, the energy price in particular has sparked fierce debates, and its impact becomes apparent when looking at the most recent incidents in Bulgaria, where increased energy prices caused political disturbances [2]. Fossil and nuclear generation technologies currently appear to bear economic advantages over still emergent photovoltaic and wind generation. In 2009, 5.8% of the globally produced energy was generated by nuclear power plants [3]. In industrialised nations, this share is significantly higher: In the United States and in the European Union, the amount of energy that was produced by nuclear power plants ranged between 10.0% and 14.1% [3].

However, the Fukushima Daiichi nuclear incident has painfully fostered an increasing awareness for the insecurity of nuclear power and convinced many governments to adopt phase-out legislations. The German government, for instance, adopted a similar legislation and intends to shut down all nuclear power plants before the year 2023. As laudable as this endeavour is, the complete nuclear phase-out entails difficulties, not least because the ceasing amount of controllable base load electricity has to be replaced.

Now, looking at already high energy costs, and having in mind that 10 to 14 percent of our today energy production will cease over the next years, it is most likely that energy costs will increase even more in the foreseeable future.

Regarding the intermittency of generation, especially in the electricity grid, primary energy efficiency and affordable electricity can only be provided together with electricity storage or powerful demand response mechanisms.

The industrial sector in particular will be in need of these to remain competitive in the presence of significant energy price increase and price fluctuations. Considering the amount of energy the industry procures, even slight changes in the energy pricing may entail large amounts of additional costs.

A. The Industry

In Germany, the industry requires roughly 42% of the overall energy demand [4]. Industrial players are well aware of the chances and obstacles related to the 'Energiewende'. One approach to counter the dependency from energy providers and energy prices is to install local power generation facilities, such as gas- or coal-fired power plants or block heating stations on site. In most cases, energy still has to be procured from external providers, yet, as opposed to private customers, industrial players have flexible options in doing so. Energy can be either procured in the long-term at fixed prices, or short-term strategies can be applied, procuring energy only hours before it is actually needed. These short-term purchases of energy are done at energy exchanges, such as the *European Energy Exchange AG, EEX* [5]. Following the principle of demand and response, electricity prices at energy exchanges are highly flexible and time-dependent – at times, the price can even become negative. Whenever there is a low demand for energy (e.g., at night times or sometimes right in the middle of the day), and, at the same time, an usually high amount of available energy (e.g., as a result to sunny or stormy weather and energy that is produced by solar panels or by wind turbines, respectively), the resulting price drops. Conversely, when there is a high demand for energy and there is less intermittent energy available, the price increases. The flexibility in purchasing energy allows industrial players to optimise their energy costs by means of complex investment strategies and production planning. Besides, the European legislation allows industries not only to purchase energy, but also to offer surpluses of energy at the energy exchange. This additional option further increases the potential of industries to minimise energy costs, though it aggravates the production planning likewise.

B. Production Planning and Energy Costs

Fluctuating energy prices allow industries to significantly decrease energy costs. To put it simple: In order to utilise periods with low energy costs, energy consuming parts of the process have to be shifted. As simple as this sounds, today production processes comprise a large number of sub-processes, which are also frequently interconnected and codependent. Thus, shifting parts of a process most likely requires other parts of the process to be shifted, as well. As an example, consider the welding of automotive bodies. Welding is considered an energy expensive production step and to shift welding processes to periods with low energy prices may yield significant savings, yet, welding is also one of the first processes in automotive production lines and shifting may require a complete reconfiguration of the entire production schedule, including material delivery and personnel planning. If one now considers the shifting of processes not as the only option, but as one of many options of industrial players to optimise their energy costs (e.g., to use local energy production, to procure and to sell energy at flexible prices, to use intermediate storage, or to reconfigure the production schedule), the complexity of the optimisation problem becomes apparent.

C. The EnEffCo Project

Within the project *EnEffCo* (**En**ergy **Eff**iciency **Co**ntrolling in the automotive industry), we were confronted with this exact problem, namely to optimise primary energy efficiency of industrial production facilities. In a joint project our goal was to develop an optimisation framework for short term energy procurement.

We decided to use stochastic optimisation, or more specifically evolutionary algorithms for this problem. We implemented an optimisation routine based on Evolution Strategy [6], considering production schedules as partially optimised phenotypes, which were continuously measured and mutated until some steady state occurred. The approach yielded good results most of the time; however, as with most stochastic algorithms, it could also get stuck in local optima. To counter this problem, and at the same time to make use of today's distributed computing infrastructure, we extended our approach by means of multi-agent technology [1]. Instead of using one single optimiser, we deployed many optimisation agents simultaneously, and overcame the problem of local optima by using different initial populations.

In this article, we summarise and conclude our work by presenting collected experiences in optimising energy costs of production processes by means of artificial intelligence.

We start with introducing the reader to the concepts used within this work, describing the domain model used for representing production processes and schedules in Section II. Then, in Section III, we describe in detail how the production processes are simulated and how the process schedule is optimised in terms of the prospected energy costs. Afterwards, in Section IV, we present a first evaluation of this optimisation using three different example processes. Subsequently, in Section V, we elaborate our approach in distributing the optimisation process among software agents, and how it fares compared to the centralised optimisation. Finally, we have a look at related approaches in Section VI and conclude our work in Section VII, where we also motivate the application of our framework in other domains.

II. Concept and Process Representation

In our approach, we use evolutionary algorithms to rearrange individual processing steps to make the best use of times of cheap energy, for instance due to variations in the availability of renewable energies, like wind, or solar.

Of course, this approach is only feasible if the production facilities are not used to their full capacity at all times, but only if there is potential for variations. This may also be the case if some machines can be used for multiple tasks, only one of which can be carried out at a time, or in case of variable shifts and break times. Another requirement is the availability of storage area for intermediate products, so that their production can be brought forward, or be deferred, to make use of times of low energy costs. Locally installed energy sources, energy storages and co-generation units can also be taken into account.

In preparation of the optimisation, the first thing to do is to create a model of the production process, including the several activities, the machinery, resources, and (intermediate) products involved. We decided on employing a very simple model, being inspired by Petri nets and adding only a bit of domain-specific information on top of that. Basically, the model consists only of *activities*, representing the individual steps in the production as well as supportive processes, and *resources*, representing all physical entities in the factory, i.e., products and by-products as well as machinery.

This model of the production process – the individual activities and how they are connected – can then be simulated, executing the several activities and consuming and producing resources accordingly. The result of the simulation is used as a quality measure for the actual optimisation algorithm, which will eventually return the process plan with the highest quality, which can then be used to re-schedule the execution of the individual activities in the process.

Besides finding the optimal process plan for a given production process, the simulation and optimisation can also be used for investigating the effect of variations in the process model, e.g., higher storage capacities.

In the following, we will introduce the production process meta model; in the next section, simulation and optimisation are explained.

A. Production Process Meta-Model

The production process is modelled as a bipartite graph of *activities* and *resources*, similar to a Petri net [7]: Activities correspond to transitions, and resources correspond to places. Consequently, activities are "activated", or executable, if both the resources to be consumed by that activity as well as enough capacities for the resources to be produced are available. Other than in a classical Petri net, activities are not executed instantaneously but have a certain duration. Also, there are different types of resources with specific characteristics.

A slightly simplified diagram of the meta-model is shown in Figure 1. In the following, the individual elements of the model are described in detail.

- The *ProcessGraph* represents the process as a whole, made up of activities and resources. The attribute `secPerStep` specifies the number of seconds each atomic time step takes.
- An *Activity* is an individual action in the production process, having the given `duration` (multiples of the atomic time step). Activities can have `input` and `output` resources and an `energyConsumption` (one value per time step), which can also be negative.
- *Resources* represent items involved in the production, e.g., raw materials, products, machinery, or even waste heat. Depending on what they represent, their `type` is either *primary*, *secondary*, or *inventory*. Each resource has an initial `stock`, a maximum `capacity`, and may also have associated `costs`.
- *Linkings* represent the `input/output` relation between activities and resources. The `quantity` specifies the amount to consume or to produce of that resource. Consequently, they indirectly act as a precedence constraint between activities.
- *Constraints* can be used to handle a variety of additional conditions that are difficult to check otherwise, like time windows when activities must (not) be executed, e.g., for break times.

The classification of resources is based on these rules:

1) *Primary Resources* are more or less directly integrated into the final product, e.g., raw materials, pre-fabricated parts, and intermediate products.
2) *Secondary Resources* have a role in the production, without being an actual part of the product, e.g., pressurised air and gasoline for machines, waste heat, or a battery's state of charge.
3) *Inventory Resources* are part of the inventory of the factory, e.g., machines and tools. (Consequently, we think of inventory resources as not being actually consumed or produced, but merely allocated and deallocated.)

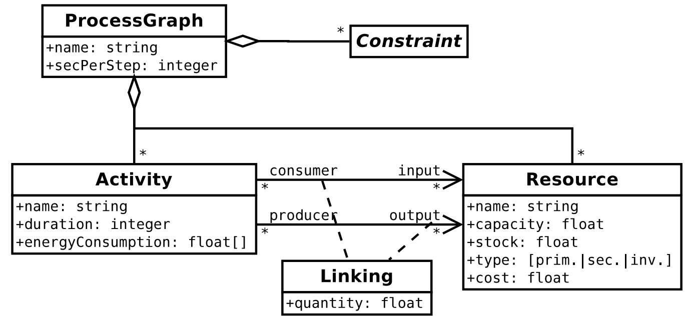

Fig. 1. Process meta-model, slightly simplified.

Conversely, activities that are producing or consuming one or more primary resources, and are thus directly involved in the production, are considered *primary activities*; otherwise, we will speak of *secondary activities*.

Electrical energy, being the main concern of the optimisation, is not regarded as a resource, but treated separately. Unlike resources, which have to be produced by activities of the process itself, electrical energy can be retrieved in (for all practical purposes) unlimited quantities and at any time. Moreover, the price for electrical energy can vary over the course of the day, based on the energy market. Surplus energy can be sold, as well.

When an activity is executed, its input resources are consumed and its output resources are produced, and it adds to the overall energy consumption of the production process. Primary and inventory resources are consumed in the first step and produced in the last step of the activity's execution; both secondary resources and energy are consumed and/or produced in *each* step of the activity.

Using this simple meta-model, a wide range of production processes can be modelled. At the same time its generality also allows for the simulation and optimisation of energy-related processes in other domains, such as creating charging schedules for electric vehicle fleets [8].

B. Implementation of Process Model and Modelling Tool

The process meta-model and a simple graphical editor for creating and configuring process models have been implemented as extensions to the Eclipse development environment. Following the usual notation for Petri nets, activities are represented by rectangles and resources by circles, using line style and colour to distinguish the different types of activities and resources (Figure 2).

Besides the basic modelling capabilities, the editor provides means for validating the process graph, for browsing and importing energy consumption data from a data base, and for passing the process graph to the optimisation system.

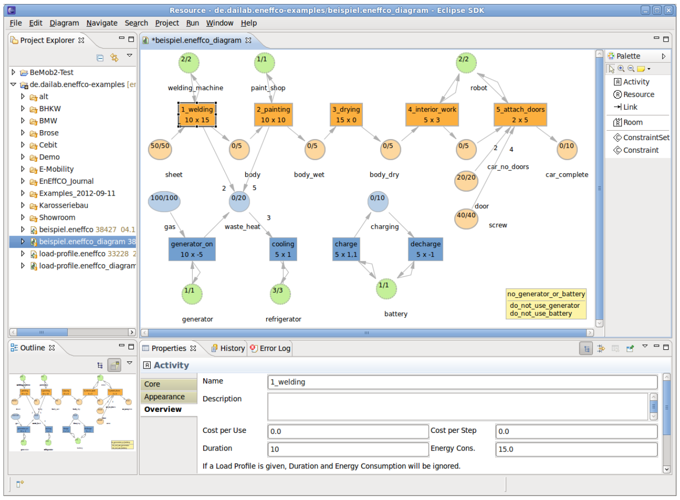

Fig. 2. Graphical process editor showing an example process.

C. Acquisition of Energy Consumption and Price Prognosis Data

One important prerequisite for optimising production processes with respect to their energy consumption is, of course, the measurement of the energy consumption of the several activities making up the production process. To this end, a number of sensors have to be installed in the production facilities to measure and record the energy consumption of the individual machines.

However, while this yields the energy consumption of e.g., a certain industrial robot, this does not correspond directly to any of the activities. For instance, an activity can require the work of different machines, and one machine can serve different activities. Instead, an activity's energy consumption is usually a section of the combined energy consumption of multiple machines.

For this purpose, a special data base client has been integrated into the process modelling tool. Using this client, the user can choose energy consumption profiles for one or more machines from the data base and select the section corresponding to a certain activity from a diagram. The combined energy consumption data for that period is then set to be the energy consumption of that activity.

III. Simulation and Optimisation

The purpose of the optimisation is to find the best possible *production schedule* for a given process model [6]. That schedule is defined by the times the individual activities are executed.

The optimisation process consists of three major aspects:

1) the simulation of a given production schedule,
2) measuring the quality of that schedule, based on the result of the simulation, and
3) finding the schedule with the highest quality.

In the following, we will look at each of these aspects in detail.

A. Simulation

The simulation of a production schedule keeps track of the resource stocks and the energy consumption in each step 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:

- For each activity to be started, the given quantity of *primary* and *inventory* input resources are consumed.
- For each activity that is currently running, the given quantities of both *secondary* resources (input and output) and *energy* are consumed and/or produced.
- For each activity to be finished, the given quantity of *primary* and *inventory* output resources are produced.

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 based on the prognosis given by the day-ahead energy market – in the implementation at hand, cost optimisation is conducted based on day-ahead price forecasts, e.g., for the EEX electricity spot 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 that.

Once the simulation has terminated, it yields a record of the energy consumption and the resource stocks for each individual step in the execution of the process. These numbers, combined with the resources' capacities, the energy price curve, and other constraints, can now be used to determine the *quality* of that production schedule.

B. Quality Measurement

The *quality* of a production schedule p is determined by a sigmoid function of its *defect*, such that a high defect results in a quality close to -1, and a defect close to zero gives a quality close to zero (see Equation 1). A negative defect will result in a positive quality (this is possible in some situations, e.g., in case of negative energy prices, or energy-producing activities).

$$quality(p) = \frac{-defect(p)}{\sqrt{1 + defect(p)^2}} \tag{1}$$

The *defect* of p is the weighted sum of the energy costs $(e(p,i) \cdot w_e)$ and the defects (over- and under-shootings) of the several resources' stocks $(s_r(p,i) \cdot w_r i)$ over all steps i of the simulation (see Equation 2).

$$defect(p) = \sum_{i \in steps} [e(p,i)\ w_e + \sum_{r \in res.} s_r(p,i)\ w_r(i)] \tag{2}$$

In this equation, the energy consumption, stocks and weights are represented as functions. Different weights w can (and should) be used for resource stocks being too low and those being too high and for the different kinds of resources.

Production schedules that exceed the maximum or minimum capacities of a resource are not discarded immediately, but are merely given a lower quality rating. This is beneficial in overcoming local optima.

C. Optimisation

Finding an energy- and cost-efficient arrangement of the several activities in the process for a given energy price curve is both a constraint-satisfactory problem and an optimisation problem: on the one hand, there must be no violations of the resources' capacities; on the other hand, the production schedule has to be as cost-efficient as possible.

Due to the large number of degrees of freedom in the process plans – with many different activities that can be started or stopped in each step of the process – the search space is much too big for exhaustive search to be applicable.

In our approach, we make use of *Evolution Strategy* (ES), a stochastic optimisation method originally introduced by Rechenberg [9], which is similar to Genetic Algorithms [10]. Besides Evolution Strategy, both *Simulated Annealing* and *Ant Colony Optimisation* have been tried, as well. However, of the three algorithms ES yielded by far the best results.

1) The ES Algorithm: 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. Based on these μ "parents", λ "offspring" are created by recombining and mutating a random selection of ρ parents. 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 the quality of the best individual does not improve for a certain number of generations.

Algorithm 1 EVOLUTION STRATEGY(μ, ρ, λ)

```
current ← INITPOPULATION(μ)
repeat
  next ← ∅
  for i ∈ {1..λ} do
    parents ← rand. select ρ indiv. from current
    offspring ← MUTATE(RECOMBINE(parents))
    next ← next ∪ {offspring}
  end for
  current ← select μ best from current ∪ next
until quality stagnates
return  best individual from current
```

2) Applying ES to Manufacturing Schedules: In the system at hand, each individual represents a possible production schedule. To this end, three functions have to be implemented for production schedules, next to the quality measurement: (i) How to create the initial population of individuals, (ii) how to mutate an individual, and optionally (iii) how to recombine individuals.

The initial population is created by a very simple scheduler, chaining primary activities as long as and as early as the primary resources and inventory resources permit, or until a desired quantity of products has been produced. Thus, the initial production schedule already constitutes a valid (but naive) schedule for all the primary activities, but without taking secondary resources or energy costs into account.

There are several possibilities for mutating an individual, one of which is chosen at random: (a) a randomly chosen secondary activity can be inserted into or removed from the schedule, (b) an activity or a group of activities (primary or secondary) can be moved to another place in the process plan, i.e., being executed earlier or later, or (c) the execution times of two activities can be swapped.

For recombination, one can randomly select activities from one of the two parents, or take the activities up to some specific step from one parent, and the rest from another – of course always taking care that the right number of primary activities is selected to complete the task at hand. However, due to the many dependencies among the individual activities of a production schedule – the ordering of primary activities as well as secondary activities being executed at times relatively to some other activities – recombination does not yet work well for this domain. Thus, in practice, the parameter ρ was always assumed to be 1.

D. Implementation of the Optimisation Framework

A generic optimisation framework was created that can be used for optimising different domains using different optimisation algorithms. The actual Evolution Strategy algorithm as well as the process model domain have been implemented as plug-ins for this framework [6].

The optimisation is controlled via a simple graphical user interface (GUI, Figure 3). Like the rest of the optimisation framework, the GUI has both generic and domain- or algorithm dependent parts. For the manufacturing domain, the optimisation GUI features a large domain-specific area, providing controls for configuring the simulation and optimisation (e.g., the energy price curve to use) and for showing the best production schedule found so far in a Gantt chart-like diagram. The process chart is continuously updated as the optimisation proceeds, and also allows to 'rewind' to previous steps in the optimisation.

Once the optimisation has come to an end, additional charts are available, showing the energy consumption and resource stocks for each step in the final production schedule, as well as the development of these numbers over the course of the entire optimisation as a three-dimensional plot. Finally, the optimised production schedule can be saved to file.

IV. EVALUATION

In this section, we will discuss a number of application examples of the process optimisation algorithm. Our first example describes the ideal manufacturing process, providing enough capacities – in both time and space – to shift primary activities so that parts of the production can be handled at times of cheap energy. The scenario shown in the second example may be more realistic w.r.t. today's manufacturing processes: Here, the production activities can not be changed, but only secondary activities (such as cooling, co-generation units and buffer batteries) may be used for shifting energy consumption. Finally, the third example demonstrates both the flexibility of

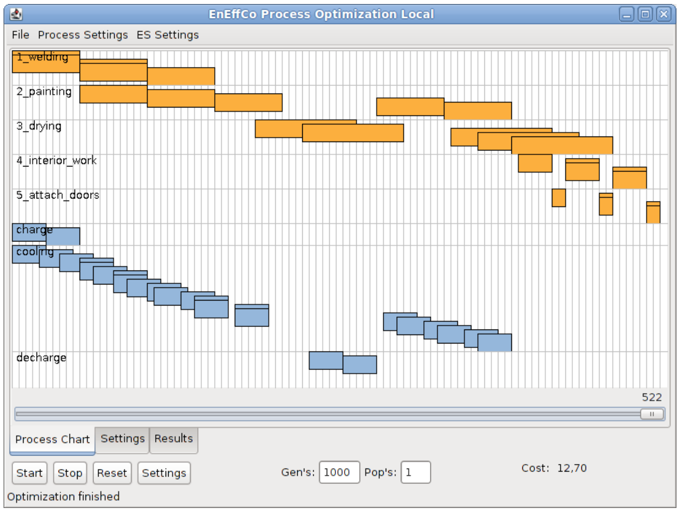

Fig. 3. Prototypical user interface for controlling the optimisation and viewing the results.

the process model and the prospects of our approach, as we use the optimisation for creating optimal charging schedules for large electric vehicle fleets.

A. Example 1: Use of Functional Storage

In this example, a simple, fictional car manufacturing process is pictured (see Figure 2).

1) Example Process: The process starts with two energy-intensive activities that induce high amounts of waste-heat: 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 was created. The resulting production process graph is supplemented with utility activities and resources such as cooling, on-site electricity storage and a gas-powered co-generation unit. The latter two elements can be used to temporarily decrease the energy consumption, but costs for the consumed gas will in turn add to the production schedule's defect.

While the process surely is not too realistic, it demonstrates many of the aspects that can be realised in the process model, for example

- modelling the basic production chain,
- inventory resources used by multiple activities,
- resources associated with a cost, and
- cooling facilities and other supporting processes.

2) Optimisation Results: The resulting schedule can be seen in Figure 3. Here, the process has been optimised against a hypothetical hill-shaped energy price curve, i.e., with highly-priced energy in the mid of the day and low-priced energy in the morning and evening.

As can be seen, most of the energy-intensive activities (*welding* and *painting*) are taken care of in the morning, with the exception of one instance of the *painting* activity, which has been deferred to the afternoon. The high-price period is spend entirely with the *drying* activity, which consumes no electric energy at all. The remaining activities are positioned as late as possible, to get the lowest possible price for the required energy. Note also, that among the several instances of the *cooling* activity in the morning there are also two instances of the charging activity, *charging* the aforementioned in-house energy storage when energy is cheap, and *discharging* it again when the energy price is highest.

B. Example 2: Shifting Secondary Activities

The second example deals with a more realistic setting: Here, the core manufacturing process is fixed in time; no primary activities can be shifted. The straightforward motivation for this scenario is that in most industries, energy consumption is not the key cost driver. Hence, the goal for energy cost minimisation is to optimise energy consumption, given a specific production schedule. In this scenario, no primary activities are shifted. Instead, secondary activities, such as ventilation or even the generation of electricity and heat through combined heat and power stations (CHP) are viable means to approximate an optimum energy load curve. In fact, secondary processes may contribute significantly to the overall energy consumption of industrial sites.

1) Example Process: The example chosen describes a site configuration, where wind energy will be provided on site and a 24 hour wind generation forecast is incorporated into the calculations. Additionally, local energy generation comprises a combined heat and power station, which can be either idle or operate with half or full generation capacity. All primary production processes are combined into a single, day-long activity with a specific load curve, since, as mentioned before, modifications are not eligible for them. As a shiftable secondary activity, ventilation is modelled for load shifting purposes. It can be operated on standard capacity or can alternatively be increased or decreased to adjust its load level. However, a sufficient amount of fresh air must be provided at all times.

The schedule is now optimised according to a 24 hour day-ahead electricity price forecast. Local energy production is assumed to be very cost efficient. Wind generation is merely characterised by maintenance costs for the turbines, CHP related costs are calculated from maintenance and gas expenses and are attributed to heat and electricity proportionally. The costs for ventilation load shifting are assumed to be higher than wind generation costs but lower than CHP expenses in this example.

2) Optimisation Results: Figure 4 shows the resulting load curves for this optimisation example. The abscissa shows the time of the optimisation period. The first ordinate on the left shows electrical load in megawatts. The second ordinate on the right shows electricity market price forecasts in euro per kWh. Six different graphs describe the optimisation results. The area graph shows the wind generation forecast for the site. The dotted line, which is related to the second y axis on the right, depicts the external electricity price forecast. The dashed line and the solid line show the external electricity procurement

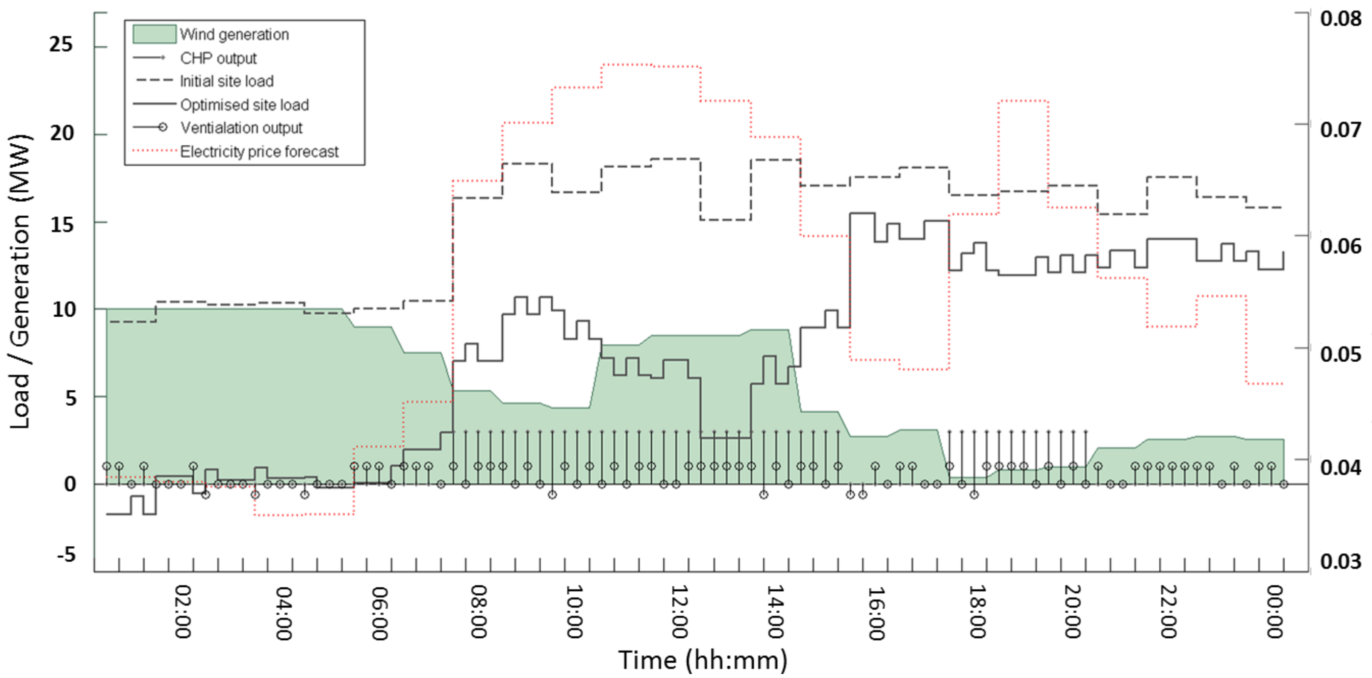

Fig. 4. Load optimisation from shifting secondary activities on an industry site.

of the site before and after the optimisation respectively. Energy consumption before the automation (dashed line) does not incorporate any local wind generation and is assumed to be covered by the external grid only. Shiftable loads are presented as stem graphs in this figure. Stem graphs with a solid point marker present the operation of the combined heat and power station, stem graphs with circle markers present ventilation. Negative ventilation load occurs when an increase of operation capacity is enforced to maintain the amount of fresh air within the predefined boundaries. Positive ventilation load occurs when operation capacity is reduced to decrease external electricity procurement.

Since wind forecast, site consumption from primary activities, and electricity market price are assumed to be non-variable, the attention should be drawn to the shiftable activities. It can be seen that the combined heat and power are strictly dependent on the market price and generation is curtailed when the price drops below about 0.059 Euros per kWh. Load shedding from ventilation control also depends upon the electricity market price but is less costly than CHP, so it is only excluded when the electricity price is at its lowest point during the high price interval between 8:00 and 21:00. To keep the amount of fresh air within the necessary boundaries, additional ventilation is injected from time to time. This occurs primarily when the electricity price is low.

To sum up these results, the optimisation tool clearly adjusts load profiles to external electricity price forecast and internal electricity generation costs to decrease the overall energy procurement costs.

C. *Example 3: EV Fleet Charging Schedules*

In the third and final example, the meta-model is applied to a different domain: creating charging schedules for electric vehicle (EV) sharing fleets in a micro smart grid (MSG) [8]. Here, the challenge is to schedule long-running charging activities so that no bookings are at risk while at the same time making use of locally produced energy and times of low energy prices. Further, the EVs can be used as temporary energy storages for load balancing.

1) Example Process: Here, the process graph is not created by hand, using the graphical editor, but instead is derived automatically from another model, describing the setup of the MSG, including among others the various *electric vehicles* and their current state of charge, a number of *bookings* for those EVs, and different *prosumers* representing both locally installed regenerative energy sources as well as the facility's own prospected power consumption.

For each prosumer, a day-long primary activity with an energy consumption curve reflecting the prognosis is created. Each EV is represented by a small subgraph featuring an inventory resource for the storage (the EV's battery), a secondary resource for its current capacity, and one or more charging activities reflecting the different possibilities for charging and discharging the battery, dependent on the charging station. Finally, each booking is represented by another primary activity, linked to the respective storage resource, and fixed in time at the booking's starting time. Figure 5 depicts one of those segments.

2) Optimisation Results: This optimisation was carried out two times: Once with the above described process model using

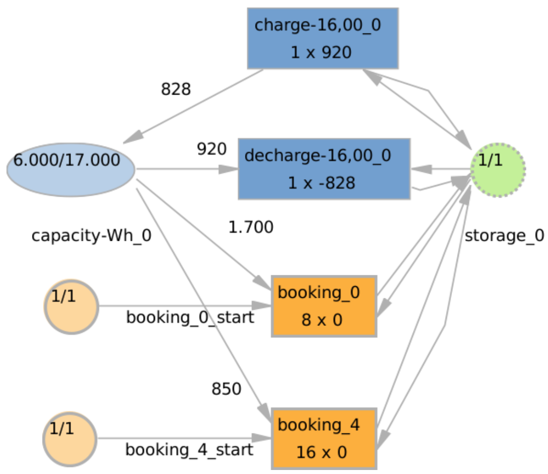

Fig. 5. Segment of process graph showing activities and resources for a single electric vehicle and associated bookings.

the meta-model and optimisation described in this paper, and once using a domain-specific meta-model developed specifically for this task (still using the same optimisation algorithm).

The results of the optimisation are promising: Charging activities are scheduled such that all bookings can be fulfilled (i.e., the storage of the respective electric vehicle is sufficiently charged) while at the same take making best use of locally produced energy and periods of low energy prices. The resulting charging schedules closely resemble those created using the domain-specific model.

However, we also found some limitations in our meta-model. For instance, using inventory resources to "lock" electric vehicles while charging or being rented does not allow for bookings that are not bound to a specific vehicle (e.g., bookings for *any* vehicle).

Nevertheless, first results convinced us to extend our approach to the domains of electro mobility and smart grids.

V. Agent-based Optimisation and Distribution

While evolutionary algorithms yield good results most of the time, it is also possible, as with other stochastic optimisation algorithms, that the optimisation gets stuck in local optima. To increase the chances of arriving at a solution close to the global optimum, the optimisation should be applied to more than one "population", and since the individual populations are optimised independently from each other, they can easily be parallelised and distributed.

To this end, the optimisation framework has been embedded into a distributed multi-agent system, allowing for the transparent and dynamic distribution of an arbitrary number of optimisation clients and servers.

Admittedly, the strength of the agent paradigm is less the transparent distribution but rather the comprehensive support that facilitates the development of autonomous, reactive, proactive and social competent entities, namely agents. As mentioned above, our current implementation is focused on distribution rather than on exploiting the latter attributes of agency. Yet, we justify the use of agent technology with our future intentions. The presented optimisation framework was well planned and its development was subdivided into different stages. In the first stage it was our intention to implement a distributed system and to ensure reliable and robust communication between the system's entities. Right now we find ourselves at this very point. For the future, however, we plan to exploit agent capabilities more comprehensively. Based on the robust and reliable distribution we want to allow agents to exchange partially optimised process plans and to recombine these plans for a more effective mutation mechanism. The recombination process, however, challenges agency far beyond distributional aspects and for this exact reason we decided to make use of agent technology right from the beginning. We consider the current application as first step towards a far more efficient and complex multi-agent based optimisation software. For a complete overview of our future intention, however, the reader is referred to Section VII.

In the following, we describe the interaction protocol, which makes a number of optimisation servers ("agents" conducting the optimisation) available to one or more optimisation clients [1]. Afterwards, we explain how the protocol and the surrounding multi-agent system have been implemented using the JIAC V agent framework.

A. Interaction Protocol

Two roles are involved in the protocol:

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

Obviously, there should be more than one optimisation server agent for the distribution to provide any benefit at all, but there may be multiple clients, as well, sharing those servers. An interaction diagram of the protocol is shown in Figure 6. It is composed of the following steps:

1) The protocol starts with a client broadcasting a REQUEST message to all the servers.
2) Each server receiving the message checks whether it already has an "employer", i.e., whether it is currently running an optimisation. If not, it replies with an OKAY message.
3) The client receives 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 optimised to that server. The number of remaining optimisation runs is reduced. (The full model, including energy consumption curves, price curves, etc., is not sent until now, to reduce network traffic.)
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 optimisations.

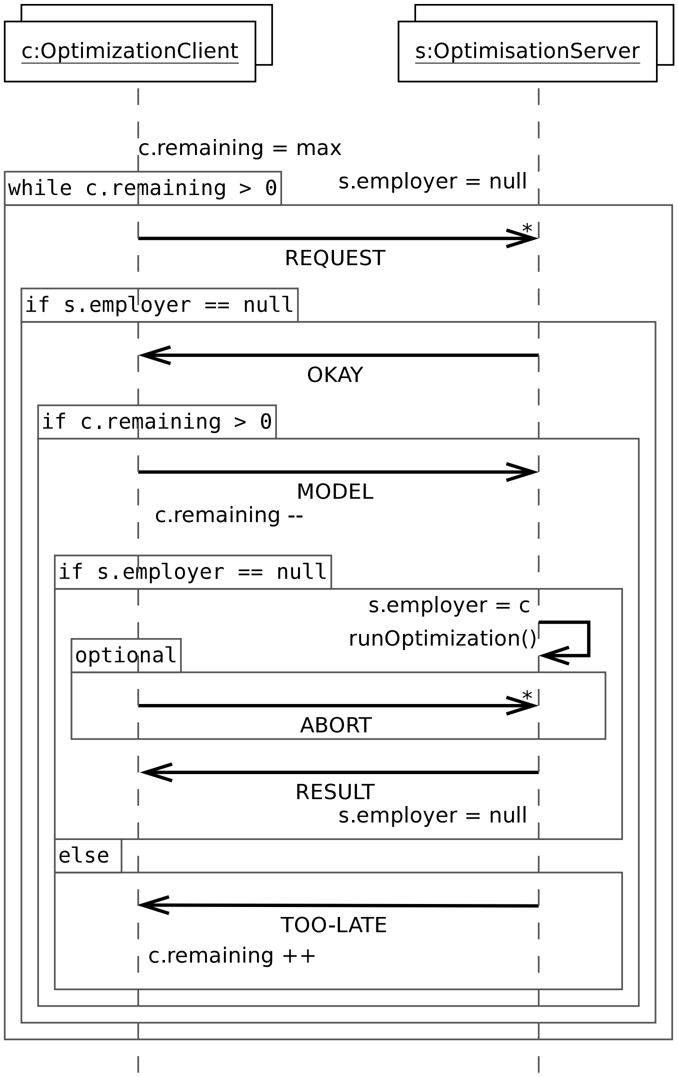

Fig. 6. Interaction protocol used in the distributed optimisation. Messages marked with an '*' are sent to all servers.

- Otherwise, the server accepts the client as its new employer and starts the optimisation 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 optimisation.

5) The client continues sending out REQUEST messages until the desired number of optimisations has been conducted.

Using this interaction protocol, each of the populations of a $(\mu/\rho + \lambda)$ optimisation can be distributed to another agent. Since each run of the optimisation, or each population respectively, is independent from the others, this does not introduce any noteworthy communication overhead.

B. Implementation using JIAC V

JIAC V (Java Intelligent Agent Componentware, Version 5) is a Java-based multi-agent development framework and runtime environment [11], [12]. 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' behaviours and capabilities are defined in a number of so-called *Agent Beans*, which are controlled by the agent's life cycle.

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

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

- 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. Consequently, if one of the server agents drops out, it can seamlessly be replaced by another one.
- With each JIAC agent running in a separate thread, a node with multiple agents being deployed to a multi-core server computer will automatically make best use of the several CPUs.
- Using asynchronous messaging, optimisations can be aborted ahead of time. Also, servers can send back intermediate results, to provide a trend for long-running optimisations.

Besides agents holding the Optimisation Client and Server Beans, a number of additional agents have been added to the system to represent and to connect the different components, as shown in Figure 7.

- A *DB Agent* provides an interface to the data base holding the measured energy consumption values, making them available to the other agents.
- Integrated into the Eclipse IDE is a *Plugin Agent*, which connects to the *DB Agent* to acquire energy consumption data to be imported into the current process graph. Further, this agent can send the process graph created in the editor to an *Optimisation Client Agent*.
- The *Optimisation Client Agents* carries out the distributed optimisation, sending individual optimisation jobs to different *Optimisation Server Agents*.
- The result of the optimisation can then be sent to the *Web GUI Agent*, showing the resulting process plan and its properties in a number of diagrams and graphs.

Using the same optimisation algorithms, the distributed system performs as well as the local system. It yields good results in reasonable time and the variability of results quickly decreases with an increased number of populations.

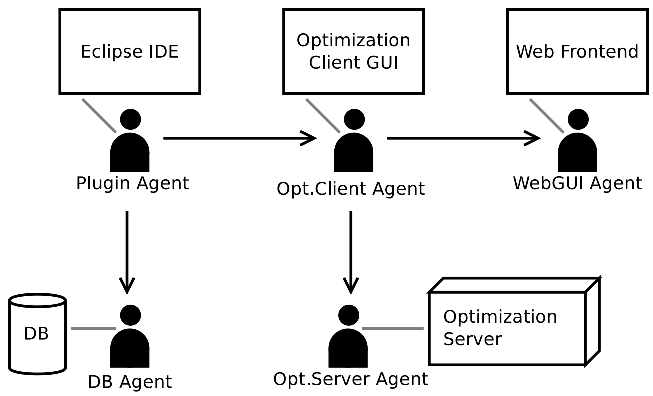

Fig. 7. Agent, components, and interactions in the distributed system.

C. Evaluation

Complementary to the evaluation of the optimisation approach in general, as discussed in Section IV, in this section we want to evaluate specifically the benefits of the distributed and parallelised version of the algorithm using the agent-based setup [1].

We evaluated the benefits of distribution and parallelisation using the first example process from Section IV, a production goal of five completed cars, a hill-shaped energy price curve and an evolution strategy with $\mu = 3$ and $\lambda = 8$.

The example process was optimised several times with different numbers of populations. The number of populations ranged from one to thirteen, and ten runs of the optimisation were performed in each case. The results are shown in the logarithmic plot in Figure 8. Note that at the time this evaluation was conducted [1], the quality function was $quality(p) = \frac{1}{1+defect(p)}$, resulting in a different range in quality values. As can easily be seen, the results of the evaluation are still valid using the new quality function.

As can be seen in Figure 8 (left), using only one population, the quality of the optimised 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 optimisation 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. 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 schedule. 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, making the quality look almost discrete.

Further, we discovered that there is little to none correlation between the time an individual optimisation run takes, and the resulting quality (see Figure 8, right): the result of a quick optimisation run can be just as good (or bad) as that of a longer running optimisation, and vice versa. Thus, one possibility to improve the performance could be to start a large number of optimisations in parallel, and to abort the remaining optimisation runs once the first few results to choose from have arrived.

VI. Related Work

Industry has long since discovered, that the optimisation of manufacturing processes is able to significantly increase revenues. As a result to the continuous demand for optimisation frameworks, there are many sophisticated applications available today. In this section we outline the current spearhead of optimisation tools and concepts; yet, due to the broad range of existing approaches it is difficult to present a comprehensive survey and for this reason we decided to put emphasis on approaches and concepts that influenced our own work the most. We open this section with an analysis of academic approaches that apply evolutionary algorithms for the optimisation of manufacturing processes and proceed by presenting commercially distributed optimisation software. Here we distinguish between general purpose frameworks, visual approaches, manufacturing- and business process optimisation tools. Finally, we discuss the significance of our work against the backdrop of contemporary applications.

A. Evolutinary Algorithms and Process Optimisation

The idea to use evolutionary algorithm for the optimisation of manufacturing processes is not entirely new, as the complexity of many optimisation problems has strongly promoted their use.

Highly interesting for our work is the approach of *Santos et al.* [13], 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 energy price curve. *Bernik et al.* [14] developed a similar approach, although they do not account for energy criteria. The approach is capable to propose manufacturing schedules that are able to satisfy a given production target. In addition to the manufacturing schedule, resource requirements are calculated and assigned to the production depots. *Schreiber et al.* [15] describe a similar application, which optimises manufacturing schedules towards a given production target. As opposed to the work of *Bernik et al.*, the application is able to calculate so called lot-sizes, which are defined as the number of pieces that are processed at the same time at one workplace with one-off (time) and at the same costs investment for its set up [15].

To summarise, while there are some approaches that account for energy related factors, dynamic pricing is currently not covered although the markets offer such possibility.

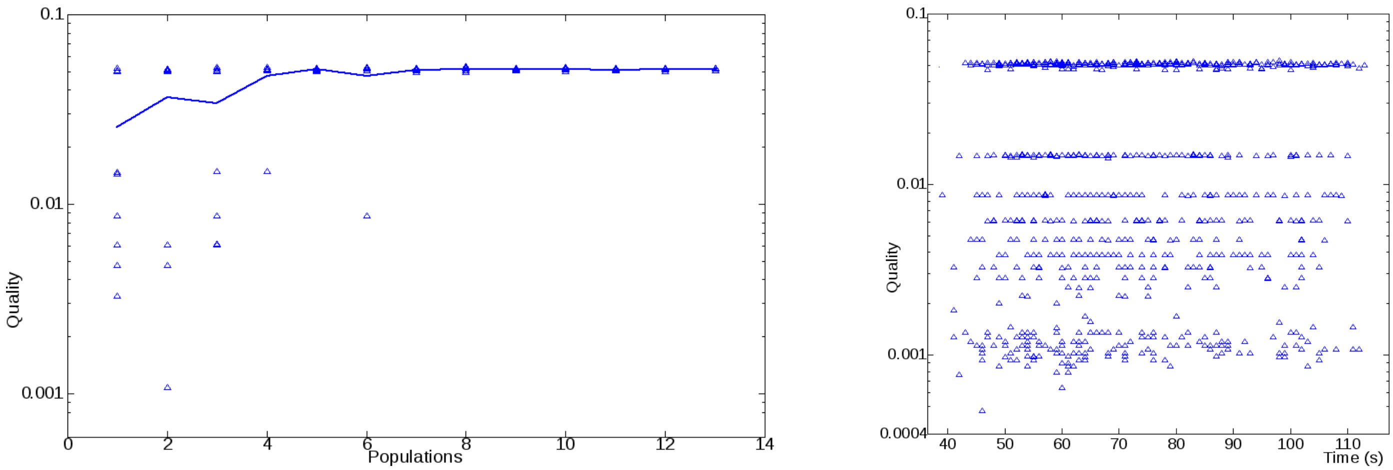

Fig. 8. Left: Correlation of number of populations to expected result quality. The graph indicates average quality values. Right: Correlation of time of optimisation run to result quality. [1]

B. Optimisation Frameworks for Production Processes

Next to frameworks that apply evolutionary algorithms in order to optimise manufacturing processes, there are of course applications that apply other methods for the same objective. The *Siemens Plant Simulation Software* [16] for instance facilitates the simulation based optimisation of production systems and controlling strategies, while business- and logistic-processes may be supported as well. The *SIMUL8* framework [17] (Figure 9), *Arena* [18] and *GPSS/H* [19] provide similar features and are able to simulate entire production processes, from warehouse capacities and equipment utilisation to logistics, transportation, military and mining applications. *SIMUL8* additionally accounts for real life requirements, such as maintenance intervals and shift patterns. Further, *SIMUL8* uses an agent-based simulation for the optimisation of production processes.

Other types of software packages as for instance *Simio* [20] and *ShowFlow* [21] do not explicitly focus on the optimisation of production processes, but on their visualisation. For this purpose, most of the mentioned applications apply sophisticated 3D engines.

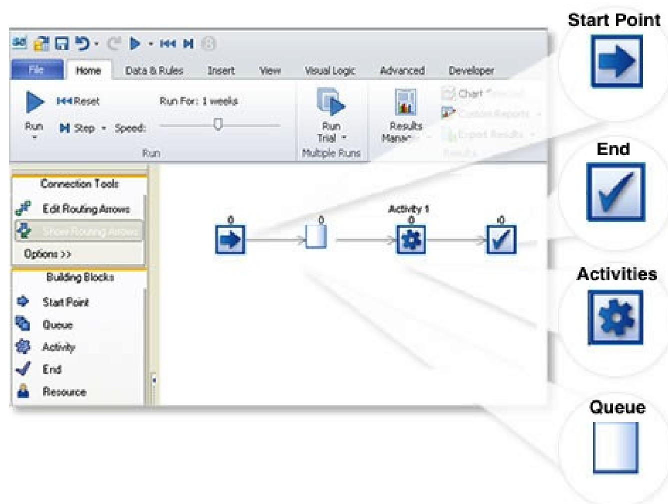

Fig. 9. The *SIMUL8* production process configuration tool, showing a palette (bottom left) and a canvas (bottom right) element as well as an exemplary process structure, including a start-, a queue-, an activity- and an end object. [17]

C. General Purpose Frameworks

Thus far, we have exclusively analysed approaches that have been developed for the optimisation of manufacturing 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 optimise general processes, such as monetary flow, quality- and organisation management, allocation scenarios, logistics, transports and many more. Foundation to these frameworks is a generic meta-model, which is able to capture process structures, and which is usually based on established concepts.

The *PACE* framework of Eichenauer [22] and the work of Siebers et al. [23] for instance feature an arbitrary level of detail for process design. While *PACE* uses hierarchically arranged *High-Level-Petri-Nets* for this purpose, Siebers et al., using *AnyLogic*, apply an object-oriented meta-model for its processes and uses a multi-agent based model for the simulation of process configurations. *AnyLogic* further comprises a graphical user frontend, which provides information on simulated processes similar to the representation that we use for our own process configuration tool. However, *AnyLogic* integrates information on the current simulation procedure and allows for the real-time adjustment of simulation parameters, such as throughput rates or storage capacities. As an example, this feature can be used in order to simulate and observe the impact of sudden machine failures. An illustration of the visual representation of the simulated processes is given in Figure 10.

SLX [24] takes a layered approach to process modelling.

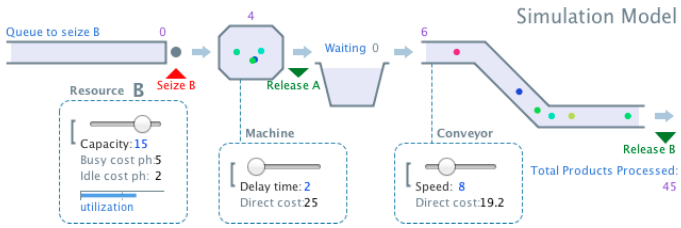

Fig. 10. The *AnyLogic* visualisation of simulated processes ([23], cut-out). Not only displays the tool the current state of simulation processes, but also allows for the adjustment of simulation parameters, such as throughput rates and capacities.

Most commonplace processes are handled in *SLX*'s upper layers, while more complex problems can be captured with *SLX*'s lower layers. The *Microsaint* [25] package avoids hierarchical structures and facilitates readability as well as easy comprehensibility. The framework entirely relies on flow charts as meta process language. In most analysed frameworks, process design is usually supported by visual editing tools. The *ADONIS* framework [26] for instance provides an impressive graphical editor for the design and manipulation of the examined process system.

In summary, we can state that general purpose tools apply a generic meta-model in order to facilitate a broad range of problems. For this meta-model, established concepts such as Petri nets [7] or flow charts are used. The analysed frameworks facilitate process design by graphical editing tools.

D. Optimisation Frameworks for Business Processes

In addition to applications that explicitly account for the optimisation of production processes, we want to attend focus on those that have been developed for other reasons. Business processes for instance have a striking resemblance to manufacturing processes and as there are optimisation frameworks for business processes, we want to mention the most prominent members of this realm as well.

To start with, *ProcessModel* [27] is a business process optimisation software, which supports optimisation from problem analysis to efficiency evaluation. The tool is able to visualise many aspects such as money savings or the efficiency of analysed processes to serve customers. A similar application is *SIMPROCESS* [28]. In addition to the capabilities of *ProcessModel*, *SIMPROCESS* is able to handle hierarchical process structures and comes with a set of sophisticated tools for the process design. Both applications apply means of simulation in order to verify optimised processes and to estimate their overall quality.

E. Lessons Learned

In this section, we gave a comprehensive overview on state of the art concepts and applications. We have already mentioned, that there are many sophisticated applications available today. Some of them have been explicitly developed in order to optimise production processes, others were designed in a generic fashion and yet feature similar capabilities.

The idea of optimising production with respect to dynamic energy tariffs is adopted by none of the examined applications, and energy related criteria in general are currently not comprehensively covered by state-of-the-art solutions, as only the approach of *Santos et al.* supports such factors.

We learned that evolutionary algorithms can be used to increase the performance of optimisation algorithms and thus selected such principle for our own application. For this purpose, we applied an established concept [9] whose performance we further enhanced by distributing our computation units.

Also, our survey did not indicate distributed computing to be widely used in process optimisation frameworks. Only the *AnyLogic* framework provides an according feature, for which the developers make use of the agent paradigm.

For our implementation we use the exact same view, only that we apply a rather comprehensive agent model as we use our agents as autonomous problem solvers while *AnyLogic* agents can be understood as simulated autonomous entities, such as persons or vehicles.

The analysis of general purpose frameworks inspired us to use a very simple and generic domain model in order to support a large number of process structures and also to provide functionality beyond the scope of optimising manufacturing processes.

To sum up, we can say that currently there are neither concepts nor frameworks, which account for the optimisation of manufacturing processes with respect to variable energy prices.

VII. Conclusion and Future Work

In this paper, we presented an optimisation framework that was developed within the government- and industry funded project EnEffCo. The main objective of the EnEffCo project was to develop software that facilitates to increase the primary energy efficiency in production and to evaluate the software with the involved industry partners.

The optimisation framework exploits the fact that industrial users are able to purchase energy with a short lead and at highly flexible prices, e.g., at the European Energy Exchange, EEX. The energy prices at the energy exchange comply with the principle of demand and response. As such, time periods with surpluses of energy (e.g., caused by an increase in wind or sunshine and the resulting energy from wind engines or solar collectors) and low grid demand (e.g., right in the middle of the day or during night times) result in low and possibly negative energy prices, while periods with only little energy from renewable energy sources and an increased grid demand result in high energy prices.

In order to capture the arrangement of production lines, we have developed a suitable domain model. The model is similar to a Petri net and comprises only two main types, namely activities and resources. A link type is used to indicate a connection between activities and resources. The generic design of the domain model allowed us to consider scenarios beyond the originally intended scope of the project. As an example, we were able to optimise charging procedures of electric vehicles.

An instance of this model, representing the different machines and tasks found in a production process, is then passed to the optimisation framework, together with an energy price forecast obtained from the energy exchange.

Due to the many options in optimising production processes (e.g., randomly shifting, adding- or removing individual activities within extensive timeframes), we decided to use stochastic optimisation. Besides other approaches, such as Simulated Annealing and Ant Colony Optimisation, the best results were achieved using Evolution Strategies, where a population of individuals (process plans) is gradually mutated and (optionally) recombined until a satisfying quality is reached.

While the quality of the results may vary, the optimisation process generally produces reliable results in a timely manner, allowing industries to quickly act even on short-term energy price fluctuations. Making use of today's distributed computing architectures, the optimisation can be distributed to multiple clients and servers, using the JIAC V multi-agent framework. This way, the reliability of the outcomes increases further, while on average taking no longer than a single run of the optimisation. Additional JIAC agents are used to integrate the optimisation with other components of the system, e.g., the process modelling tool and the user frontend.

The EnEffCo Project officially ended in December 2012. Nevertheless, we intend to further refine our approach and to extend the capabilities to other domains.

Currently, we are transferring our findings from the EnEffCo project to ongoing projects, for instance for load-balancing in micro smart grids and for optimising the charging schedules of large electric vehicle car sharing fleets [8]. While some extensions to the domain model and the optimisation framework had to be introduced, so far, the results look promising.

Furthermore, it is our intention to enhance the optimisation process by allowing agents to exchange and recombine partially optimised production schedules. This requires agents to autonomously query intermediate results from other optimisation agents and to select suitable parts from these results for the mutation process. Where our current implementation is focused on distributional aspects only, this extension will exploit the agent paradigm more comprehensively and pave the way for a high-performance agent-based optimisation framework.

ACKNOWLEDGMENT

This work was developed in the project "Energieeffizienzcontrolling am Beispiel der Automobilindustrie" (*EnEffCo*) (Eng.: Energy Efficiency Controlling in the automotive industry), funded by the *Federal Ministry of Economics and Technology* under the funding reference number 0327843B.

REFERENCES

[1] T. Küster, M. Lützenberger, and D. Freund, "Distributed optimization of energy costs in manufacturing using multi-agent system technology," in *Proc. of 2nd Int. Conf. on Smart Grids, Green Communications and IT Energy-aware Technologies (ENERGY 2013), St. Maarten, St. Maarten*, P. Lorenz and K. Nygard, Eds. IARIA, March 2012, pp. 53–59.

[2] "Bulgaria government to resign, pm boiko borisov says," BBC NEWS Europe, March 2013, accessed: 21.5.2013. [Online]. Available: http://www.bbc.co.uk/news/world-europe-21516658

[3] "Energymix EU–USA–China," Bundeszentrale für politische Bildung, March 2013, accessed: 21.5.2013. [Online]. Available: http://www.bpb.de/nachschlagen/zahlen-und-fakten/europa/75143/energiemix

[4] "Auswertungstabellen zur Energiebilanz für die Bundesrepublik Deutschland 1990 bis 2011 – Berechnungen auf Basis des Wirkungsgradansatzes," Arbeitsgemeinschaft für Energiebilanzen e.V., September 2012.

[5] "EEX Website," European Energy Exchange AG, accessed: 21.5.2013. [Online]. Available: http://www.eex.com

[6] 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.

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

[8] D. Freund, A. F. Raab, T. Küster, S. Albayrak, and K. Strunz, "Agent-based integration of an electric car sharing fleet into a smart distribution feeder," in *3rd IEEE PES Innovative Smart Grid Technologies Europe (ISGT Europe)*, 2012.

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

[10] A. Eiben and J. Smith, *Introduction to Evolutionary Computing*, ser. Natural Computing. Springer-Verlag, Berlin, 2003.

[11] 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.

[12] M. Lützenberger, T. Küster, T. Konnerth, A. Thiele, N. Masuch, A. Heßler, M. Burkhardt, J. Tonn, S. Kaiser, J. Keiser, and S. Albayrak, "JIAC V – a MAS framework for industrial applications," in *Proc. of the 12th Int. Conf. on Autonomous Agents and Multiagent Systems (AAMAS 2013), Saint Paul, MN, United States of America*, T. Ito, C. Jonker, M. Gini, and O. Shehory, Eds., 2013.

[13] 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.

[14] 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.

[15] 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.

[16] Siemens, "Plant simulation – plant, line and process simulation and optimization," Project Broschure, 2010, accessed: 21.5.2013. [Online]. Available: http://www.plm.automation.siemens.com/en_us/Images/7541_tcm1023-4957.pdf

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

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

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

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

[21] I. S. Solutions, "The ShowFlow website," 2011, accessed: 21.5.2013. [Online]. Available: http://www.showflow.com/

[22] 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.

[23] 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.

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

[25] 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.

[26] 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.

[27] ProcessModel, Inc., "The ProcessModel website," 2011, accessed: 21.5.2013. [Online]. Available: http://www.processmodel.com/

[28] CACI, "The SIMPROCESS website," 2011, accessed: 21.5.2013. [Online]. Available: http://simprocess.com/