

# Metaheuristic Simulation-based Production Planning for Energy Efficiency: A Case Study

Bernhard Heinzl<sup>\*</sup>, Wolfgang Kastner

Research Unit of Automation Systems E191-03, TU Wien, Treitlstraße 1–3, 1040 Vienna, Austria;

<sup>\*</sup>[bernhard.heinzl@tuwien.ac.at](mailto:bernhard.heinzl@tuwien.ac.at)

SNE 30(3), 2020, 105-116, DOI: 10.11128/sne.30.tn.10523  
Received: July 30, 2020; Revised: August 12, 2020;  
Accepted: August 15, 2020  
SNE - Simulation Notes Europe, ARGESIM Publisher Vienna,  
ISSN Print 2305-9974, Online 2306-0271, [www.sne-journal.org](http://www.sne-journal.org)

**Abstract.** Modern industrial production planning and control (PPC) systems are responsible for supporting planning decisions on how to optimally produce a given set of products while minimizing costs and retaining production constraints, such as delivery tardiness or off-times. In recent years, more and more attention has also been paid on energy efficiency as part of production optimization, resulting in competing optimization targets. In order to solve such complex multi-objective scheduling problems in practice, metaheuristic methods are used because of their ability to deliver acceptable solutions in feasible time. In this paper, we demonstrate the application of a General Variable Neighborhood Search (GVNS) metaheuristic on a case study of flow shop scheduling in an industrial bakery in different scenarios and study the effect of different energy prices on the planning result. The case study features a simple production line with thermal processes for baking and freezing and also incorporates the energy supply system as well as a model of the thermal building hull. The metaheuristic is combined with a hybrid discrete/continuous simulation model to evaluate the energy efficiency of different production scenarios. The hybrid simulation enables to accurately capture material and energy flow within the production in an integrated and dynamic manner. Overall, this simulation-based optimization method is intended to support energy-aware production scheduling in practical applications.

## Introduction

Energy efficiency in industrial production has become an important topic in recent years because of the substantial potential for energy savings in the industrial sector [1]. Energy-aware Production Planning and Control (PPC) strategies can be used to influence energy demand and energy costs during operation, for example by shifting the production of energy-intensive products to the night hours, where energy is often cheaper. However, it is not sufficient to only consider energy as an optimization goal. Instead, energy efficiency must be seen as part of a multi-objective system of production targets together with production variables such as storage costs, throughput times or delivery delays. Such multi-objective problems with complex, sometimes time-dependent constraints are hard to solve for real-world problems. Modern solutions often rely on heuristic or metaheuristic methods [2].

For evaluating the fitness of solution candidates during metaheuristic search, simulation-based methods are gaining interest because they enable to capture complexity of real-world problems including difficult dynamic interactions without the limiting assumptions many other approaches have. However, with regard to energy optimization, interdisciplinary holistic simulation models are required which include dynamic interactions across engineering domains in order to get an accurate prediction of the overall energy demand, that not only includes production machinery, but also technical building services. For example, heating a production oven generates waste heat that is dissipated into the room and affects heating and cooling energy demand for the building. Similarly, the actual setup time for preheating the oven depends on different conditions, including which products have been produced before, and the setup time affects production throughput and scheduling. Incorporating energy considerations in production logistics simulations with their time-dependent interactions in an accurate manner requires advanced

modeling and simulation approaches that combine discrete (product flow) and continuous dynamics (energy flow). Describing such hybrid discrete/continuous simulation models in a consistent manner with formally sound semantics in complex real-world applications is still challenging [3, 2].

In our work, we aim to support modern PPC tools in incorporating energy considerations into the planning process using simulation-based optimization techniques. We present a metaheuristic procedure for energy-aware optimization of production scheduling, which employs a dynamic discrete/continuous simulation model for evaluating overall energy demand as well as material flow, while taking into account dynamic interdependencies across domains. To this end, we have developed a component-based hybrid simulation tool based on a formal model description, called *hyPDEVs*. The simulation tool includes reusable model components that can be used to simulate interdisciplinary production systems [4]. The optimization method aims at sequencing and time scheduling a given list of production jobs while minimizing energy demand together with other production goals. The procedure combines a Variable Neighborhood Search (VNS) metaheuristic with Variable Neighborhood Descent and Simulated Annealing (SA) for local search and diversification. We demonstrate the feasibility of this method on a flow shop scheduling problem of an industrial bakery. The simulation model includes a production line with alternative paths as well as technical building services for energy supply and a thermal building model. We compare different scenarios, investigate the effect of variable energy prices and on the planning result and highlight the potential benefit of considering energy as an optimization target.

## 1 Related Work

Since optimization in a complex solution space with simultaneous objectives is a difficult problem to solve, practical multi-objective optimization has spawned a wealth of approaches and solution methods [5, 6]. A general overview is given in [7, 8, 9], while [10] focuses on optimization in the food manufacturing industry. Optimization methods can be categorized in exact methods, such as gradient methods, dynamic programming, etc., and approximate/heuristic methods like greedy heuristics, Simulated Annealing, Evolutionary Algorithms or Variable Neighborhood Search [11].

According to [10], metaheuristics and customized multi-objective heuristic approaches are well-suited for applications in real-life industrial production planning problems (which typically are NP-hard), in contrast to exact approaches that require simplified models. Metaheuristics allow to explore the search space more efficiently and effectively, especially if they are tailored to the individual problem [12]. Different metaheuristic algorithms, such as Evolutionary Computation, Tabu Search, Particle Swarm Optimization (PSO) or Simulated Annealing (SA) have been successfully applied to various logistics optimization problems [13]. In previous work, we have also investigated applying a Genetic Algorithm on a similar case study [14, 15]. A set of tuning and customization measures was applied to significantly improve performance, including adapting operators for a guided search, and hybridization with Tabu Search and Pattern Search.

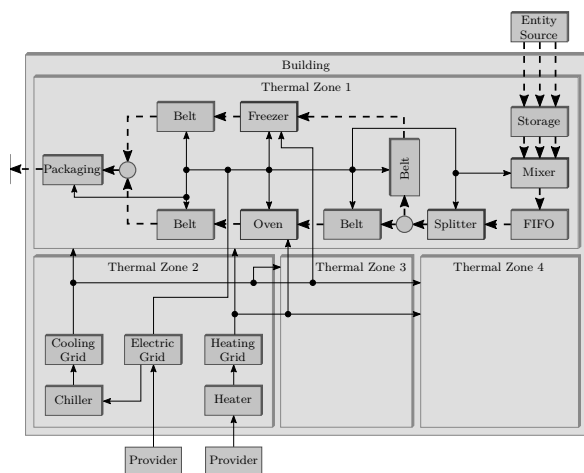
Besides population-based metaheuristics, like Genetic Algorithms, which work with a population of candidate solutions to concurrently sample different regions of the solution space, single-solution-based methods, also called trajectory methods, iterate over a single solution are more exploitation-oriented and usually need fewer simulation evaluations, which improves overall computation time. Among these trajectory methods, Variable Neighborhood Search (VNS) algorithms have shown excellent capability for solving scheduling problems [16]. This is in accordance with other publications, e.g. [17, 18, 16], which have successfully applied VNS for job scheduling problems in the production domain. In [19], the authors compare different optimization methods for simulation-based optimization of production plans, in which VNS also leads to the best results.

For simulation-based optimization strategies in particular, several authors employ discrete-event simulations (DES). In [20], the authors combine a DES with a Genetic Algorithm for energy-oriented machine allocation planning. The DES, however, uses deterministic energy profiles and only allows very simplified consideration of dynamic energetic interdependencies. It also does not include production periphery or building facilities in the energy consumption. Other publications describe hybrid discrete/continuous simulation for energy efficiency in production, however, most of these use them in a scenario-based manner without systematic metaheuristic optimization. In [21], a multilevel simulation is presented to model the material flow to-

gether with production equipment and energy system. This includes dynamic coupling across domains, however, optimizations are conducted manually.

## 2 Case Study Overview

To evaluate the described method on a real-life example, we devised a case study of a production plant. The case study is a simplified model of a real production plant of an industrial bakery that produces baked goods [4]. The model features a typical production line with machines, storage and conveyor belts, an energy supply system with heater and cooler, and a building model with thermal zones.



**Figure 1:** Production facility consisting of production machines (blue), logistics components (purple), technical building services for energy supply (red) and thermal building zones (green).

The model has been simplified compared to the original plant by omitting unnecessary production steps and other components, in order to reduce problem complexity and simplify parametrization. However, this does not change the structure or topology of the general archetype and therefore does not undermine the goal of demonstrating the feasibility of the method. The main critical elements of the production still remain in the model, specifically: intertwined continuous and discrete dynamics, cross-domain modeling with dynamic interdependencies, and complex product flow with splitting, merging and batching. More details on the simulation are given in Section 4.

The production and logistics components form a production line for two product variants: baked and frozen. Baked products pass a oven for baking while frozen products are frozen directly (in a freezer) without being baked. Both are designed as conveyor belts, meaning that new entities continuously enter the stations and leave on the other side. Since these products share all other stations, only one type of product can be produced at any time. For both products, respective ingredients are being pulled from the storage, after which they are mixed into a dough, divided into portions (splitting) and continue on different conveyor belts. After baking/freezing, the finished products are packaged in different quantities.

The building is modelled as a simple thermal compartment model with four thermal zones, each representing a distinct part of the facility: production hall, plant rooms and cold storage. These zones all have independent conditioning (for example, the cold storage is kept at 4 °C) and exchange thermal energy with one another according to the defined wall topology. They also exchange thermal energy with the environment, for which a variable ambient temperature may be specified.

The energy system provides necessary technical building services, mainly supplying energy for the production machines as well as for heating and cooling the thermal building zones. The energy system is comprised of a heater (powered by natural gas) that supplies heat to the oven and building, a chiller (powered by electricity) that supplies cold to the freezer and building, as well as respective energy grids responsible for distributing the energy. The heating and cooling grids also include thermal energy storage that models the storage capacity of the grid. All other production stations receive electric energy from the grid.

The production orders are executed according to a production schedule (Pplan), which is the main input vector to specify the production scenario and is the subject of optimization, which constitutes a flow shop scheduling problem. In addition to the order starting times, the Pplan also specifies the start of the setup processes for the oven and freezer. During these setup processes, the oven (or freezer) is preheated (or cooled) to the defined operating temperature before the products arrive at the station. This also implies that, in contrast to traditional discrete-event material flow simulations, the setup time is not a fixed parameter, but may change dynamically during the optimization process.

The case study further includes a work plan (Aplan) that specified the production steps and process parameters, such as temperature set points, baking time or batching size. It serves as a look-up table for the individual stations and the process parameters may be different for different product types.

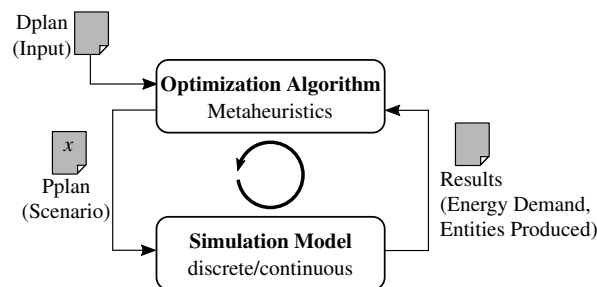
### 3 Optimization Method

Production optimization deals with finding an optimal combination of production resources, such as equipment, utilities or energy, to achieve a given production target in the best possible way. Operative Production Planning and Control (PPC) methods consider production resources and cost factors to find optimal production schedules. To support PPC in practice, modern APS (Advanced Planning and Scheduling) software systems offer integrated resource planning [22]. The complexity of the underlying optimization problem with multiple competing objectives and complex constraint conditions suggests using a simulation-based strategy [23]. Hereby, a simulation model of the system under consideration is part of the objective function and serves as a prescriptive tool to predict and evaluate the performance of a given scenario. Dynamic simulation allows to consider more complex systems than with conventional analytical models, especially for highly time-dependent problems, while offering more accurate predictions and overall improving planning quality [2]. Recent advances in computational techniques have also led to an increased interest in simulation-based methodologies to solve optimization problems [23].

However, the fact that these methods in general do not provide closed analytical representations of the objective function (or its derivatives) prevents straightforward deployment of many standard optimization algorithms. Instead, many practical simulation-based optimization solutions employ metaheuristics that rely solely on the evaluation of the objective function itself. These algorithms modify a candidate solution in an iterative manner to find a near-optimal solution until termination criteria are met. The final solution may not be the global optimum, but is often good enough in practice.

The overall computation cycle of using the simulation-based optimization methodology for production scheduling in industry is illustrated in Figure 2. The starting point is a given demand plan (Dplan) specifying how many of which entities (i.e. products) need to be delivered when. This Dplan serves as a basis

to generate an initial solution of a production schedule (Pplan), which is then evaluated using a dynamic simulation model. The results of the simulation are fed back to the optimization to be evaluated for its fitness using a specified cost function  $f(x)$  (objective function). Based on this evaluation, the optimization algorithm iteratively adapts the solution in order to find the minimum of  $f(x)$  subject to constraints. The cycle continues until certain termination criteria (e.g. fixed number of iterations, computation time threshold, etc.) are met [2].



**Figure 2:** Simulation optimization cycle. Input is a demand plan (Dplan) of entities to be produced. The optimization algorithm generates a production schedule (Pplan), which is adapted iteratively based on an evaluation of simulation results [2].

We employ a single-solution metaheuristic method based on a General Variable Neighborhood Search (VNS) procedure. The VNS [16] is an effective method guiding a local search by switching between increasingly larger neighborhoods to efficiently explore the solution space. The VNS consists of (1) a shaking phase that generates randomized perturbations of the solution for diversifying the search to escape local optima, and (2) an intensification phase that searches for improvements in the local neighborhood, typically using a local search procedure. For improving the performance of the search, the Generalized VNS (GVNS) replaces the local search by a Variable Neighborhood Descent (VND) procedure [24, 25]. The VND is similar to the VNS, but does not include shaking and thus limits the search to strict improvements (i.e. descent). It does not use the same neighborhood structures as the VNS, which offers more flexibility in tailoring the search procedure to the problem instance [2].

Algorithm 1 presents an overview of the optimization method as pseudocode. After generating a feasible initial solution  $x$ , the VNS component (described in more detail in Section 3.3) guides a search for im-

improvements over a fixed number of iterations  $n_{\text{iter}}$  by generating in each iteration a random neighboring solution  $x'$  (by means of the neighborhood structure  $\mathcal{N}_k(x)$ ) that serves as starting point for the VND that tries to find a local minimum  $x''$ . If the solution is accepted, the VNS restarts with neighborhood  $\mathcal{N}_1$  and the new starting solution  $x''$ . Otherwise, the neighborhood is enlarged. The acceptance criterion is based on Simulated Annealing (SA) [24] for further diversifying the search by allowing to accept potentially worse solutions during the early search phase. Infeasible solutions are also allowed during the search and are evaluated based on a penalizing cost function, which is described in Section 3.2. For the particular case study, the optimization vector  $x$  contains the job starting times and the setup time duration for oven and freezer, which allows to optimize job sequencing and scheduling as well as setup processes, which also have an influence on the energy demand.

---

**Algorithm 1** GVNS/SA algorithm
 

---

```

1:  $\mathcal{N}_k \leftarrow$  set of VNS neighborhood structures
2:  $x \leftarrow \text{InitialSolution}()$ 
3:  $k \leftarrow 1$ 
4:  $i \leftarrow 0$ 
5:  $T \leftarrow T_0$ 
6: while  $i \leq n_{\text{iter}}$  do
7:    $x' \leftarrow \text{shaking}(\mathcal{N}_k(x))$   $\triangleright$  random perturbation
8:    $x'' \leftarrow \text{VND}(x', \mathcal{N}_k)$   $\triangleright$  local improvement search
9:   if  $\text{acceptSA}(x'', x, T)$  then  $\triangleright$  SA criterion
10:     $x \leftarrow x''$ 
11:     $k \leftarrow 1$ 
12:   else
13:     $k \leftarrow (k \bmod k_{\text{max}}) + 1$   $\triangleright$  next neighborhood
14:   end if
15:    $i \leftarrow i + 1$ 
16:   update  $T$ 
17: end while
    
```

---

### 3.1 Initial Solution

An initial solution (function  $\text{InitialSolution}()$ ) is generated by employing a custom construction heuristic, which takes the jobs from the provided input list (Dplan), sorts them in increasing order of due date and then schedules the jobs in this order as soon as possible (forward scheduling) while considering safety gaps between jobs as well as setup times. This method en-

sures that no collisions occur, and the initial solution is feasible to be a valid starting point for the subsequent improvement search. However, further constraint violations, such as delivery delay, or energy considerations, are not checked during this phase.

### 3.2 Generalized Cost Function

The algorithm allows infeasible solution during the search process. Constraints are penalized by means of the generalized cost function

$$f(x) = \omega_1 \cdot f_{\text{dev}}(x) + \omega_2 \cdot f_{\text{en}}(x) + \omega_3 \cdot f_{\text{st}}(x) + \omega_4 \cdot f_{\text{del}}(x) + \omega_5 \cdot f_{\text{sep}}(x) \quad (1)$$

where  $f_{\text{dev}}$  denotes a penalization for deviating number of entities produced,  $f_{\text{en}}$  the energy costs,  $f_{\text{st}}$  the storage costs,  $f_{\text{del}}$  is a penalization for any potential delivery delay, and  $f_{\text{sep}}$  penalizes job separation violations. These part goals are evaluated based on the simulation.

The energy costs  $f_{\text{en}}$  are calculated by taking the simulated power supply  $P_s$  from each of the external providers  $s \in \text{Prov}$ , rate them with time-dependent energy prices  $c_s$  and accumulate them over time to obtain the overall energy costs:

$$f_{\text{en}}(x) = \sum_{s \in \text{Prov}} \sum_{t_i} P_{s,t_i}(x) \cdot (t_{i+1} - t_i) \cdot c_s(t_i). \quad (2)$$

This method of using variable energy prices allows to take into account different effects that have an influence on the energy price and therefore may potentially influence the production planning result, like for example lower energy prices during nighttime, or an additional photovoltaic system that provides solar energy during the day. The storage costs  $f_{\text{st}}$  are determined based on the time difference between job completion and delivery due date (given in the demand plan). For more details, we also refer to [26]. The coefficients  $\omega_i > 0$  are weighting factors, which may be adapted by the user to balance their preferences for individual part goals and trade-offs in a transparent manner. This weighted sum method is common for multi-objective optimization problems in practice as it is easy to implement and intuitive for the user. However, the results are often highly dependent on the weights [9, 2].

### 3.3 Neighborhood Structures

The general operation of the VNS has been described in Algorithm 1. It is responsible for diversifying the

search in a structured way during the shaking phase by successively changing the neighborhood structures  $\mathcal{N}_k$ . These neighborhood structures are usually defined implicitly by means of operators that modify the solution. For our case, we have defined four different operators [2]:

1. *OpSwitch*: This operator changes the order of jobs by taking a random number  $r$  of successive jobs (starting from a random position) and moving them to a different position. Hereby,  $r$  is chosen in the interval  $r \in [1, \min\{r_{\max}, n\}]$ , where  $n$  is the overall number of jobs and  $r_{\max}$  changes depending on the neighborhood  $k$ , see Table 1.
2. *OpShift*: The shifting operator takes a random position and moves all subsequent jobs by a specified time  $t_{\text{shift}}$ , where  $t_{\text{shift}}$  depends on  $k$ .
3. *OpChangeSetuptime*: Here, the setup time is changed by a specified amount  $t_{\text{setup}}$ .
4. *OpMerge*: The merging operator takes two random jobs, which are removed by a distance of  $d$ , and which have the same product type and combines them into one job. Merging jobs has the advantage that it reduces the number of setup processes and avoids gaps between jobs, thereby increasing production and energy efficiency.

The VND, which is used in place of a local search procedure inside the VNS to improve the generated solution in the intensification phase [25], uses the same basic operations, albeit in different neighborhoods depending on the current VNS neighborhood  $\mathcal{N}_k(x)$ . For the switching operator, the VND neighborhood containing all pairs of successive jobs is explored exhaustively. The same is done with respect to merging. For *OpShift*, the VND checks shifting groups of jobs in a binary search pattern by a set of different times  $t_{\text{shift}}$ . Similarly for the setup times, which are being reduced iteratively by a set of different  $t_{\text{setup}}$ . The complete neighborhood structures used for the case study are presented in Table 1.

### 3.4 Acceptance Criterion

Instead of accepting only improving solutions, a modified acceptance criterion (*acceptSA()*) is used that is based on a Simulated Annealing (SA) method [24, 2]. It enables to further diversify the search and better escape local optima by accepting potentially worse solutions, albeit with decreasing probability as the search

| k | Operator | Shaking                            | VND  |
|---|----------|------------------------------------|--|
| 1 | 1        | $r_{\max} = 2$                     | $r_{\max} = 1$                                     |
| 2 | 1        | $r_{\max} = 4$                     | $r_{\max} = 1$                                     |
| 3 | 2        | $t_{\text{shift}} = 8 \text{ h}$   | $t_{\text{shift}} \in \{4, 2, -1, 0.5\} \text{ h}$ |
| 4 | 2        | $t_{\text{shift}} = 12 \text{ h}$  | $t_{\text{shift}} \in \{4, 2, -1, 0.5\} \text{ h}$ |
| 5 | 3        | $t_{\text{setup}} = 0.5 \text{ h}$ | $t_{\text{setup}} \in \{0.5, 0.25\} \text{ h}$     |
| 6 | 4        | $d = 2$                            | $d = 1$  |

**Table 1:** Neighborhood structures used in the GVNS.

progresses. To be more precise, while an improving solution is always accepted, a deteriorating solution  $x''$  is accepted with the probability

$$p_{\text{SA}} = \exp(-(f(x'') - f(x))/T), \quad (3)$$

where  $f$  is the generalized cost function (see Section 3.2) and  $T$  is the temperature that decreases linearly after every VNS iteration in such a way that  $T < 10^{-3}$  during the last 10% of iterations. This effectively tightens the acceptance criterion as the search progresses until, finally, only improving solutions are accepted at the end. The temperature  $T$  is initialized with  $T_0$  according to  $T_0 = -\Delta_{\text{SA}}/\log(0.5)$ , meaning that, initially, a solution being  $\Delta_{\text{SA}}$  worse than  $f(x)$  is accepted with a probability of 50%.

## 4 Hybrid Simulation

Hybrid discrete/continuous simulation in the context of interdisciplinary assessment of energy efficiency in production enables both the material flow to be accurately modelled as Discrete-Event system and the energy flow by means of differential equations, while also taking into account dynamic interactions between these domains. Continuous representation of energy flow, as opposed to discrete energy profiles, enables to accurately incorporate transient dynamics, for example the heat-up process of an oven or the thermal heat capacity of the building. For hybrid modeling and simulation, we employ a formal model description, called hypDEVs, which is based on the Discrete Event System Specification (DEVs) [27].

The hypDEVs formalism extension aims to incorporate continuous model aspects into a discrete-event model description [28]. It is similar to the DEV&DESS formalism introduced by Prähofer [27], with the difference being that it allows improved handling of parallel and concurrent events (since it is based on the Parallel-DEVs extension [27]).

DEVS formalisms follow a component-based paradigm where the overall model is comprised of individual components being coupled together. Coupled systems can be arranged hierarchically, meaning they can be incorporated just like an atomic into a larger coupled system. This property allows to construct modular hierarchical (tree-like) models in a component-based manner [29, 27].

Component-based modeling as a paradigm facilitates modularity and separation of concerns for managing the complexity of large-scale simulation models. Components can be implemented as object classes in an object-oriented programming language [30]. These classes can then be organized in a library of model components, to be instantiated by the user in different contexts, thereby facilitating component reuse. Reusing models is crucial in an attempt to reduce the effort necessary to develop new application models [31, 32]. In order to retain modularity and reusability, it is strictly necessary to encapsulate all aspects (in particular discrete and continuous aspects) of a model component within uniform boundaries. This may become a challenge when using e.g. co-simulation [33], as this approach forces the user to split the model into discrete and continuous sub-models to be implemented by respective simulators [34]. The hypPDEVS specification, on the other hand, allows to integrate hybrid aspects already on the atomic level, thereby removing the restrictions on hybrid composability.

Based on the described hypPDEVS formalism, a simulation tool was implemented in C++ [4] together with a library of model components for modeling production systems, which are also used in the case study, see Figure 1. The production stations, especially oven and freezer, are modelled as hybrid discrete/continuous models, where the discrete model is responsible for entity flow and control logic, and the continuous model handles energy input and conversion. In particular, the energy conversion follows simple energy balance equations, including thermal heat capacity of the station, and generates diffuse waste heat that is dissipated into the respective thermal zone in the room model. Energy demand for oven and freezer is controlled by a PI controller located inside the component.

The energy model also interacts with the material flow by means of state events that indicate e.g. when the oven has reached its target temperature and is ready to accept entities. More details are described in [29].

The building and energy system components on the

other hand are mainly continuous models (as they do not directly interact with entities), except for simple control logics. The building model contains four thermal zones with homogeneous temperature distribution and calculates heat transfer across the walls between the zones as well as with the environment. For this, a temperature profile for the ambient temperature can be specified as simulation input, which allows to compare different weather conditions, e.g. summer vs. winter. For this case study, the ambient temperature is set to 20 °C. A more sophisticated thermal building model has also been developed for the hypPDEVS simulator [35], which is, however, not part of this case study. Outside energy providers supply the energy across the system boundary to the energy system, which is billed to the customer using time-dependent energy pricing.

## 5 Case Study Experiments

To demonstrate the application of the proposed GVN-S/SA method on the flow shop scheduling case study, the optimization was implemented as a prototype in MATLAB and coupled with the standalone hypPDEVS simulator [36]. Table 2 presents a simplified scenario of demand (Dplan) needing to be scheduled over the course of two days (i.e. 48 h simulation time). The table lists different orders (coming from customers) with quantities and delivery due dates (measured from the start of the simulation). Although realistic demand plans are usually more complex, simplifying the scenario allows to manually verify the results and check the algorithm for plausibility.

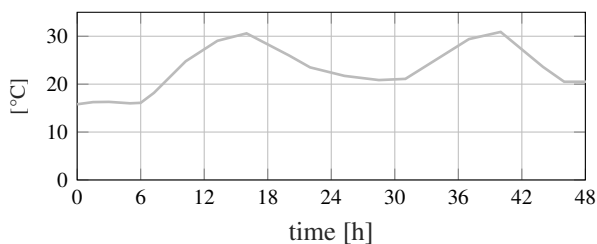
For the ambient temperature, we consider a real-world temperature curve presented in Figure 3 and we compare different energy price profiles  $c_s(t)$  (see Equation (2)). The price profiles are depicted in Figure 4. One electricity price profile  $c_{1,\text{real}}(t)$  was taken from real historical electricity spot market data of the Austrian Energy Exchange (EXAA)<sup>1</sup>, the other one  $c_{1,\text{const}}(t)$  represents its mean value (0.0437 €/kWh) kept constant over the entire period. The gas price  $c_2(t)$  is also constant with 0.04 €/kWh.

Figure 5 depicts the Oven and Freezer allocations (i.e. number of entities and temperature over time) of the final optimization result when using the constant energy price profile. The optimization ran for  $n_{\text{iter}} = 50$  iterations and the partgoal weights were cho-

<sup>1</sup>Source: <https://www.exaa.at/en/marketdata/historical-data>

| Type   | Quantity | Due Time   |            |
|--------|----------|------------|------------|
|        |          | Scenario 1 | Scenario 2 |
| baked  | 32       | 48 h       | 48 h       |
| baked  | 8        | 24 h       | —          |
| frozen | 20       | 48 h       | —          |
| baked  | 4        | 48 h       | —          |

**Table 2:** Demand plan for the simulated scenario.

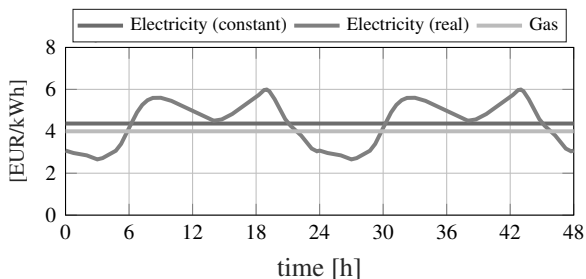


**Figure 3:** Ambient temperature curve used for the simulation, based on real data.

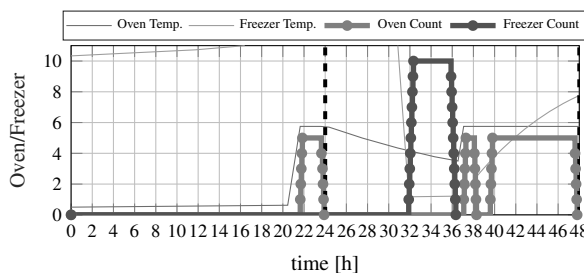
sen as  $\omega = (\omega_i)_{i=1..5} = (1, 2, 2, 1, 1)$ , while the deterioration percentage of the Simulated Annealing (SA) acceptance criterion was lowered to  $\Delta_{SA} = 0.3$ .

The result is by and large as expected, with all jobs being produced as late as possible (due to the storage costs) and the one job (Job 2 in Table 2) adhering to the earlier delivery due date. While the overall goal was improved by 28%, not much energy was saved in this case, only about 6%, as there is not much leeway for shifting or merging jobs. This is also visible in the cost function plot in Figure 6. While the energy costs make up the majority of the target value, most of the improvements are achieved through the storage costs, which have been lowered by 77%.

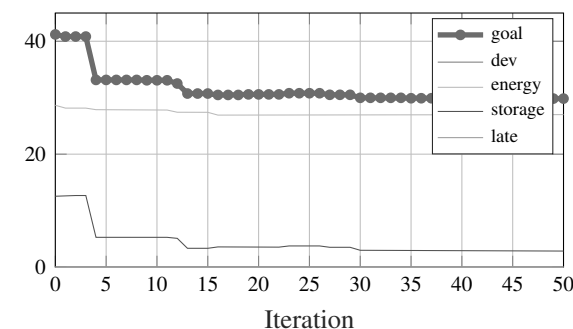
Also, the variable ambient temperature itself does



**Figure 4:** Energy price profiles, constant and real.



**Figure 5:** Optimization results for Scenario 1. The dashed vertical lines indicate job due times.



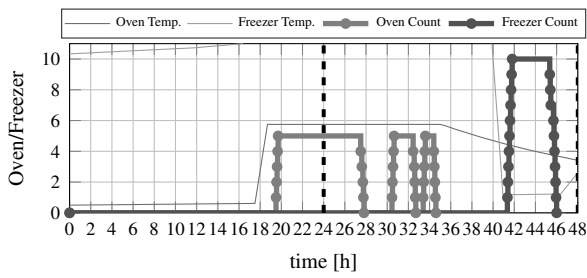
**Figure 6:** Cost function and part goals for Scenario 1.

not have much influence on the optimization result either. As further analyses have shown, this is partly due to the temperature control in the building. On the one hand, this suppresses any direct influence of the ambient temperature on the thermal behavior of the production stations, on the other hand, a (more or less) constant total amount of energy has to be supplied to heat the building over the observation period, which is only reduced by the waste heat of the stations. However, it is irrelevant when exactly this waste heat is supplied.

Next, we want to take a deeper look at the influence the energy has on the planning result. For this, we omit the storage costs in the target system by setting the respective weight to  $\omega_3 = 0$ . This has the effect that the oven lots can be produced earlier, together with Job 2 that is due earlier, thereby saving a second setup process. Switching the sequence of the oven lots does not make much of a difference energetically. The result, which we will call  $x_c$  for further reference, is shown in Figure 7.

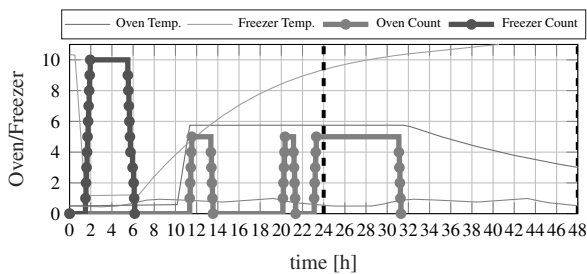
If we compare this result with the planning result  $x_r$  obtained with the real energy price profile  $c_{1,real}(t)$ , shown in Figure 8, a different picture emerges. Here, the jobs are scheduled earlier, especially in the periods





**Figure 7:** Oven and Freezer allocation  $x_c$  for Scenario 1 without storage costs and constant energy price.

with low energy prices, while the periods of high energy prices are avoided. Especially the oven lots exploit these valleys, even though the oven continues to run during these gaps.



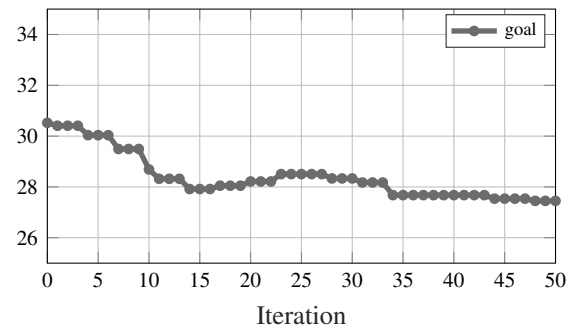
**Figure 8:** Results  $x_r$  for Scenario 1 without storage costs and real energy price (red line).

If we take  $x_c$  (Figure 7), which only considers a constant energy price, and evaluate it with the realistic variable price profile  $c_{1,real}(t)$  (as it would be done in reality), we see that the goal would be about 5% worse compared to  $x_r$ , which considers the real energy price in the optimization, see Table 3.

|        | var. Price<br>$x_r$ | const. Price<br>$x_c$ | Difference |
|--------|---------------------|-----------------------|------------|
| $f(x)$ | 27.45               | 28.82                 | 5%         |

**Table 3:** Comparison of Scenario 1 between  $x_r$  and  $x_c$ , showing the cost value evaluated using the real energy price profile  $c_{1,real}(t)$ .

The progression of the cost function, depicted in Figure 9, shows that, compared to the results without storage costs, it takes more iterations to reach a near-optimal value, and the overall values are slightly higher. This will become even more apparent later in Figure 10.



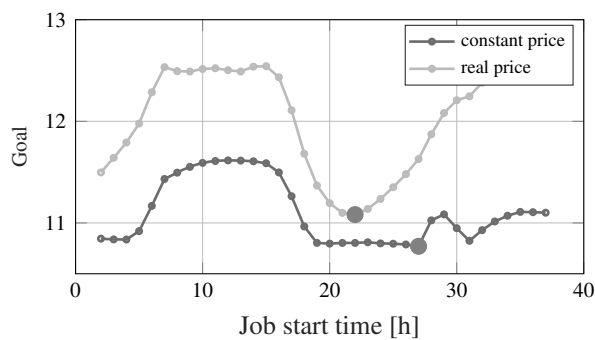
**Figure 9:** Cost function, i.e. energy costs, for Scenario 1 without storage costs and real energy price.

The effect of job scheduling depending on the energy price can be examined more closely by looking at a single job as defined in Scenario 2 in Table 2. By sweeping a single job across the planning horizon (starting time  $x_1 = 2 \text{ h} \dots 37 \text{ h}$ ), we can determine how the target function changes depending on the starting time. The result is presented in Figure 10. Both with constant energy price and real price profile, the target function has clearly pronounced hills and valleys that correlate approximately with the ambient temperature. However, the optimum values do not align, namely  $x_1 = 27 \text{ h}$  compared to  $x_1 = 22 \text{ h}$  with constant energy price.

What is also noticeable is that the target value is on average about 9% higher when using the real energy profile, compared to the constant value, even though the mean price over the course of the day is the same, see Figure 4. These higher energy costs have to do with the fact that the energy demand is unevenly distributed throughout the day, not only due to production, but also, for example, due to intermittent filling of the heat and cold storage (that are part of the energy system infrastructure), which often occurs in times of above-average energy prices. Even if one could have suspected that this would balance out over the day, this does not seem to be the case here [36].

## 6 Conclusion

The case study results show that including energy considerations and realistic energy pricing into the planning optimization can potentially reduce overall costs and provide a better planning result. Although the case study has been simplified to highlight the essential characteristics, it is easy to imagine applying the described method to more complex production systems with more



**Figure 10:** Target value plotted over the start time sweep, both for constant energy price and real energy price profile. The red dots indicate the respective minima.

diverse material flow and product types, as part of a practical PPC tool. This makes the optimization problem even more interesting because of the job completion times being more difficult to predict due to potential bottlenecks etc. From the current point of view, the presented simulation-based approach should also be applicable for these cases.

Many factors potentially influence the overall energy demand, which have to be included into the simulation. The building facilities contribute a significant portion to that, but also setup times of thermal processes can be optimized in order to not only save energy, but also improve production tardiness and reduce storage costs. These factors, however, have to be considered simultaneously, as part of a multi-objective optimization, instead of individually.

One advantage of the VNS metaheuristic is that it allows to tailor the operators and neighborhood structures to the individual problem instance, thereby allowing more efficient search and decreasing computation time. Neighborhoods (i.e. switching, shifting, etc.) are explored repeatedly and iteratively instead of sequentially, and the trade-off between exploration and exploitation can be controlled by the user. In contrast to population-based approaches, like Genetic Algorithms, VNS as a single-solution-based methods needs fewer function evaluations (i.e. simulation runs), which is a topic of importance for simulation-based methods. In the future, more extensive comparisons to population-based methods are planned.

Experiments on the case study have also shown that the Simulated Annealing (SA) acceptance criterion is

important especially for the merging operator, since, although merging two jobs might be beneficial in the end, there might still be a short-term increase in the cost function (e.g. if the products from the second job are being produced sooner and therefore increase storage costs). In the future, VNS parameter and neighborhood calibration might still be improved and even be automated to directly adjust to the problem instance without the need for user intervention.

The hyPDEVS-based simulation itself, despite its hybrid nature, delivers sufficient performance to be feasible for simulation-based optimization tasks with a large number of iterations. This has also been tested with larger real-world case studies, see e.g. [37].

### Acknowledgement

This work is funded by the Austrian Research Promotion Agency (FFG) as part of the project AS-PeCT (Adaptive Smoothed Production, project number: 858655) within the research program "Production of the Future". The authors would like to thank all project partners for their contributions.

### References

- [1] Chan Y, Kantamaneni R, Allington M. Study on Energy Efficiency and Energy Saving Potential in Industry and on Possible Policy Mechanisms. *Tech. rep.*, ICF Cons. Ltd. 2015.
- [2] Heinzl B, Kastner W. A General Variable Neighborhood Search for Simulation-Based Energy-Aware Flow Shop Scheduling. In: *Proceedings of the 2020 Summer Simulation Conference*. Virtual Event: SCS. 2020; .
- [3] Brailsford SC, Eldabi T, Kunc M, Mustafee N, Osorio AF. Hybrid Simulation Modelling in Operational Research: A State-of-the-Art Review. *European Journal of Operational Research*. 2019; 278(3):721–737.
- [4] Heinzl B, Kastner W. Platform-Independent Modeling for Simulation-Based Energy Optimization in Industrial Production. *International Journal of Simulation: Systems, Science and Technology*. 2019; 20(6):10.1–10.10.
- [5] Hwang CL, Masud ASM. *Multiple Objective Decision Making — Methods and Applications: A State-of-the-Art Survey*. Springer Science & Business Media. 2012.
- [6] Ehrgott M, Gandibleux X. *Multiobjective Combinatorial Optimization — Theory, Methodology,*

- and Applications. In: *Multiple Criteria Optimization: State of the Art Annotated Bibliographic Surveys*, edited by Ehrgott M, Gandibleux X, International Series in Operations Research & Management Science, pp. 369–444. Boston, MA: Springer US. 2002;.
- [7] Boussaïd I, Lepagnot J, Siarry P. A Survey on Optimization Metaheuristics. *Information Sciences*. 2013;237:82–117.
- [8] Xiujuan L, Zhongke S. Overview of Multi-Objective Optimization Methods. *Journal of Systems Engineering and Electronics*. 2004;15(2):142–146.
- [9] Freitas AA. A Critical Review of Multi-Objective Optimization in Data Mining: A Position Paper. *ACM SIGKDD Explorations Newsletter*. 2004;6(2):77–86.
- [10] Wari E, Zhu W. A Survey on Metaheuristics for Optimization in Food Manufacturing Industry. *Applied Soft Computing*. 2016;46:328–343.
- [11] Karimi-Nasab M, Aryanezhad M. A Multi-Objective Production Smoothing Model with Compressible Operating Times. *Applied Mathematical Modelling*. 2011;35(7):3596–3610.
- [12] Güller M, Uygun Y, Noche B. Simulation-Based Optimization for a Capacitated Multi-Echelon Production-Inventory System. *Journal of Simulation*. 2015;9(4):325–336.
- [13] Silva CA, Runkler TA, Sousa JM, da Costa JMS. Optimization of Logistic Processes in Supply-Chains Using Meta-Heuristics. In: *Progress in Artificial Intelligence*, edited by Goos G, Hartmanis J, van Leeuwen J, Pires FM, Abreu S, vol. 2902 of *Lecture Notes in Computer Science (LNCS, Volume 2902)*, pp. 9–23. Berlin, Heidelberg: Springer Berlin Heidelberg. 2003;.
- [14] Sihni W, Sobottka T, Heinzl B, Kamhuber F. Interdisciplinary Multi-Criteria Optimization Using Hybrid Simulation to Pursue Energy Efficiency through Production Planning. *CIRP Annals – Manufacturing Technology*. 2018;67(1):447–450.
- [15] Sobottka T, Kamhuber F, Sihni W. Increasing Energy Efficiency in Production Environments through an Optimized, Hybrid Simulation-Based Planning of Production and Its Periphery. In: *Proceedings of the 24th CIRP Conference on Life Cycle Engineering*. Kamakura, Japan. 2017; pp. 440–445.
- [16] Roshanaei V, Naderi B, Jolai F, Khalili M. A Variable Neighborhood Search for Job Shop Scheduling with Set-up Times to Minimize Makespan. *Future Generation Computer Systems*. 2009;25(6):654–661.
- [17] Yazdani M, Amiri M, Zandieh M. Flexible Job-Shop Scheduling with Parallel Variable Neighborhood Search Algorithm. *Expert Systems with Applications*. 2010; 37(1):678–687.
- [18] Adibi MA, Zandieh M, Amiri M. Multi-Objective Scheduling of Dynamic Job Shop Using Variable Neighborhood Search. *Expert Systems with Applications*. 2010;37(1):282–287.
- [19] Gansterer M, Almeder C, Hartl RF. Simulation-Based Optimization Methods for Setting Production Planning Parameters. *International Journal of Production Economics*. 2014;151.
- [20] Rager M. *Energieorientierte Produktionsplanung: Analyse, Konzeption und Umsetzung*. Springer-Verlag. 2008.
- [21] Thiede S, Schönemann M, Kurle D, Herrmann C. Multi-Level Simulation in Manufacturing Companies: The Water-Energy Nexus Case. *Journal of Cleaner Production*. 2016;139:1118–1127.
- [22] Kilger C, Meyr H, Stadler H. *Supply Chain Management and Advanced Planning: Concepts, Models, Software, and Case Studies*. Springer. 2015.
- [23] Swisher J, Hyden P, Jacobson S, Schruben L. A Survey of Simulation Optimization Techniques and Procedures. In: *Proceedings of the 2000 Winter Simulation Conference*, vol. 1. Orlando, FL, USA: IEEE. 2000; pp. 119–128.
- [24] Siarry P. *Metaheuristics*. Springer. 2016.
- [25] Hansen P, Mladenović N. Variable Neighborhood Search Methods. In: *Encyclopedia of Optimization*, edited by Floudas CA, Pardalos PM, pp. 3975–3989. Boston, MA: Springer US. 2009;.
- [26] Kamhuber F, Sobottka T, Heinzl B, Sihni W. An Efficient Multi-Objective Hybrid Simheuristic Approach for Advanced Rolling Horizon Production Planning. In: *Proceedings of the 2019 Winter Simulation Conference*, vol. 1. Maryland, USA: IEEE. 2019; pp. 1–11.
- [27] Zeigler BP, Prähofer H, Kim TG. *Theory of Modeling and Simulation: Integrating Discrete Event and Continuous Complex Dynamic Systems*. USA: Academic Press. 2000.
- [28] Deatcu C, Pawletta T. A Qualitative Comparison of Two Hybrid DEVS Approaches. *SNE - Simulation Notes Europe*. 2012;22(1):15–24.
- [29] Heinzl B. Hybrid Modeling of Production Systems: Co-Simulation and DEVS-Based Approach. Diploma Thesis, TU Wien, Vienna, Austria. 2016.
- [30] Leobner I. Modeling of Energy Systems for Complex Simulations. Dissertation, TU Wien, Wien. 2016.

- [31] Crnkovic I, Chaudron M, Larsson S. Component-Based Development Process and Component Lifecycle. In: *International Conference on Software Engineering Advances*. 2006; .
- [32] Cetinkaya D, Verbraeck A, Seck MD. Applying a Model Driven Approach to Component Based Modeling and Simulation. In: *Proceedings of the Winter Simulation Conference, WSC '10*. Baltimore, Maryland. 2010; pp. 546–553.
- [33] Awais MU. Distributed Hybrid Co-Simulation. Dissertation, TU Wien, Wien. 2015.
- [34] Heinzl B, Raich P, Preyser F, Kastner W. Simulation-Based Assessment of Energy Efficiency in Industry: Comparison of Hybrid Simulation Approaches. In: *Proceedings of the 9th Vienna International Conference on Mathematical Modelling (MATHMOD 2018)*. Wien, Austria. 2018; .
- [35] Smolek P, Leobner I, Gourlis G, Mörzinger B, Heinzl B, Ponweiser K. Hybrid Building Performance Simulation Models for Industrial Energy Efficiency Applications. *Journal of Sustainable Development of Energy, Water and Environment Systems*. 2018;.
- [36] Heinzl B. Methods for Hybrid Modeling and Simulation-Based Optimization in Energy-Aware Production Planning. PhD Thesis, TU Wien, Wien. 2020.
- [37] Sobottka T, Kamhuber F, Faezirad M, Sihn W. Potential for Machine Learning in Optimized Production Planning with Hybrid Simulation. *Procedia Manufacturing*. 2019;39:1844–1853.