Proceedings of the 2012 Winter Simulation Conference
C. Laroque, J. Himmelspach, R. Pasupathy, O. Rose, and A. M. Uhrmacher, eds.

SIMULATION AND OPTIMIZATION OF ROBOT DRIVEN PRODUCTION SYSTEMS FOR PEAK-LOAD REDUCTION

Sören Lorenz
Matthias Hesse

Fraunhofer Institute for Machine Tools
and Forming Technology IWU
Reichenhainer Straße 88
D-09126 Chemnitz, GERMANY

Anja Fischer

Department of Mathematics
Chemnitz University of Technology
Reichenhainer Straße 39
D-09126 Chemnitz, GERMANY

ABSTRACT

One way to improve the energy efficiency in manufacturing is the use of energy-sensitive methods in production planning. So far, the energy consumption behavior of production facilities has not been investigated in great detail. Estimates are typically obtained by connected wattage values and concurrency factors. We present a new methodology to simulate and optimize complex robot driven production systems with special emphasis on energy aspects. In particular, we show how to translate the process descriptions and energy consumption profiles into a discrete-event-based simulation model and illustrate this with an example of a car body shop facility. In order to minimize the peak-load we set up an optimization model that is based on periodic time-expanded networks. A solution of this model corresponds to a process sequence for the robots that prescribes relative starting times via additional wait intervals. This sequence is then reinserted into the simulation model to validate the improvement.

1 INTRODUCTION

Raising energy costs and the social movement to more sustainability require adjustments in the industry. Therefore dedication to the energy efficiency of production has received attention in academia as well as in business practice for some time. In the automobile production and in car body shops in particular, highly automated equipment is used which is hard to optimize once put in operation. Although control procedures still offer potential for energy savings (Putz et al. 2012), the most important parameters of the energy requirements are already determined with start of production. All the more important is the planning phase of these systems, where the subsequent behavior is determined.

In the past low energy consumption of new production systems was often only a subordinate goal. Priority was given to cost, throughput and quality. Hence methods and tools to plan for energy efficiency are mostly missing. Heilala et al. (2008) and Moeller et al. (2009) discuss this in the broader sense of *life cycle assessment* and stress the high efforts for the acquisition of the needed data. The difficulty is to increase the expenditures during planning only moderately, but on the other hand to reach a certain accuracy in order to actually influence planning decisions. In Figure 1 the options of taking energy efficiency into account at the planning stage of a production system are depicted. Our objective is to shift the efforts for energy efficiency to the level of material flow simulation. This is the most abstract level that still allows the investigation of dynamic properties of prospective production systems.

Material flow simulation is an already established method for exploring and verifying the dynamic behavior of production systems. In earlier work we investigated how energy consumption can be embedded in such a simulation (Putz et al. 2011). This allowed the identification of load peaks and their causes. We refined this approach and developed a mathematical optimization procedure that yields far better results. In the first part of this paper we describe the production scenario and a corresponding simulation model.

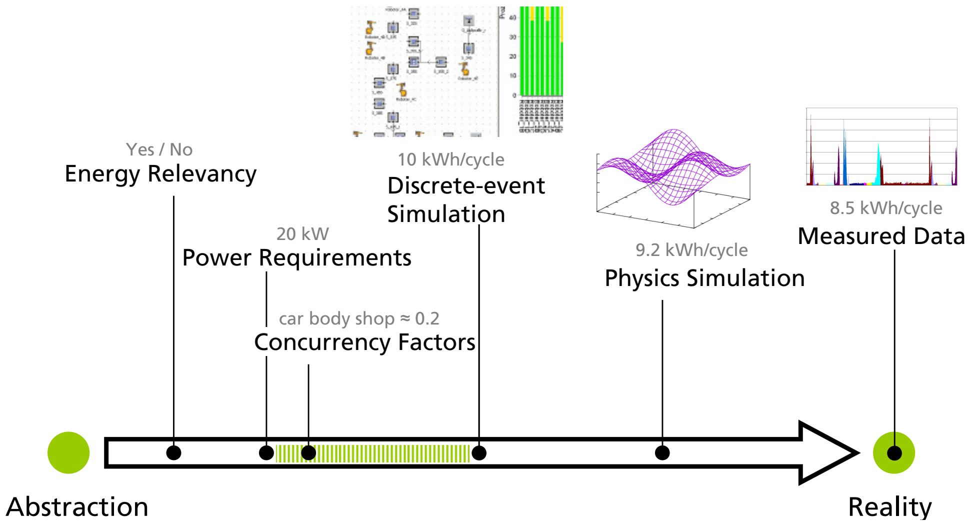

Figure 1: Towards more detailed planning models.

In the second part we introduce an optimization procedure and show the improved load behavior of the production system using a revised process sequence.

2 ENERGY-ORIENTED MATERIAL FLOW SIMULATION

2.1 Simulation Goals

The production system considered is used for joining several sheet metal parts to a car body subassembly. Except from material feeding the process is fully automated. About thirty robots handle parts and intermediate assemblies or carry joining tools. Furthermore a variety of additional energy-consuming equipment is employed. The spectrum ranges from continuous loads such as local exhaust ventilations to spot welding guns, which cause intermittent peak-loads. We focus on the robots because their correctly timed operation is the driving factor in the design of this kind of production system. Hence the time aspect of robot applications is already planned in detail and opens up some potential for optimization.

In order to map the real behavior of the robots into a model, it must be abstracted in an appropriate way. For this purpose the goals and the scope of the simulation have to be defined with respect to available data and feasible efforts. The energy-oriented material flow simulation pursues the following objectives:

1. estimation of the total energy demand and peak-loads of industrial robot applications in early planning stages to obtain a realistic base for the dimensioning of the infrastructure;
2. estimation of effects of changes in the process sequences on total energy demand and peak-loads for existing production lines;
3. providing a basis for an analysis of the energy supply flexibility of a production system and the impact on productivity if only a limited supply of energy is available.

In the simulation model, all production-related processes necessary to represent the timing correctly should be taken into account, but from the energy point of view we consider only those where robots are involved. These processes are to be assigned with characteristic energy consumption profiles, in order to determine the consumption at chosen time points and represent the history for the whole system or individual robot groups. Besides it will be clarified whether material flow simulation is useful even at this unusually detailed modeling level.

2.2 Energy Consumption Profiles for Industrial Robots

The mapping of real energy consumption behavior of technical facilities to energy consumption profiles is a frequently used method. Most often the aim is to predict the energy demand of technical systems. Further motives might be to build a basis for the formulation of legal requirements or for a comparison of different technical solutions. In general two steps are necessary: First, one has to normalize the value the technical equipment produces and then one has to determine the corresponding energetic behavior. For instance Weinert et al. (2011) propose a detailed procedure for the planning of job sequences with energy consumption profiles. Concatenation of so called energy blocks, which are characterized by operation states, process parameters and load profiles, allows a forecast of energy demand and peak-loads of the production system. The authors state that load profiles have to be generated via measurements, simulations, estimations or by comparing to reference processes showing similar behavior. For our purposes we derive such a characterization for industrial robots from measurements.

The observed industrial robots are complex mechatronic devices. Being programmable universal manipulators they perform a large variety of processes, which differ in a number of parameters. Consequently, modeling their load profiles is very difficult. While payload or travel speed can be covered by scalar quantities, the trajectories of the movements cannot be mapped to simple formal descriptions. In the planning phase it is also neither necessary nor possible to exactly determine every single process parameter.

The planning data for the production system at hand included process descriptions and time durations, from which the following process types were derived:

wait The robot is powered on and ready to immediately resume its program. In the simulation model this state is generated if a robot has to interrupt the execution of its processes because the necessary conditions are not met. In Section 3 we analyze the effect of purposefully inserted wait periods.

pick up The robot picks up a part with a mounted gripper. This movement is performed with high precision and low velocity. At the end of it the actuators of the gripper cause energy consumption, for example compressed air consumption in pneumatic clamps.

put down The robot puts down a part on a welding jig or a conveyor. The energetic behavior is identical to the one of process type *pick up*.

hold The robot holds a part precisely in a position so that it can be processed with a stationary tool. In comparison to *wait*, the energy demand is increased because of the position control.

work The robot performs a machining process using a mounted tool, *e. g.*, spot welding. The energy demand in this state is equal to the one in *hold* because we only consider the energy demand of the robot and not of the tool. Nevertheless a second type is useful, because it allows to arrange the energy demands of the tool within the timing of the robot processes. In *hold* it is assumed that there are no such energy demands or it is intended to assign them to other production equipment.

move The robot moves from its current to a target pose. This is characterized by the three phases acceleration, a phase with approximately constant path velocity and deceleration. Static friction and mass inertia cause energy peak-loads at the beginning and the end of the process.

All process types except for *move* have a constant energy demand in the simulation. For *move* processes a reduction to an average value would be a very rough simplification because measurements indicated an M-shaped power curve due to the three phases mentioned above. The energetic behavior also depends on the position of the trajectory relative to the robot console. But such effects are neglected here since the exact trajectories are not known in early stages of planning anyway. Therefore the M-shaped power curve is approximated by a piecewise linear function in the simulation. Two possible power curves are shown in Figure 2 for process durations of four seconds (solid line) and of two seconds (dashed line).

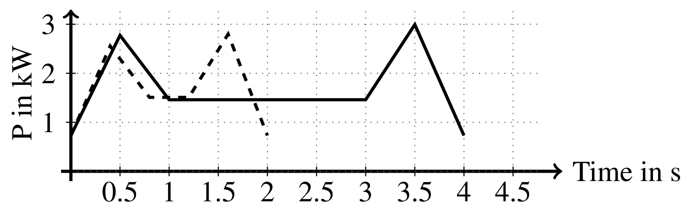

Figure 2: M-shaped power curves for different durations of a move process.

2.3 Structure of the Simulation Model

The model has been designed for an discrete-event-oriented simulation system, suitable for analyzing especially the material flows of complex production systems. These systems usually provide a set of standard components in a library, *e. g.*, for stations, conveyors and buffers. Complex networks can be built by a combination of these components. Networks can also be nested to provide for an object-oriented modeling paradigm. Various predefined report and graphing functionalities help to find bottlenecks, low utilization and other issues of production systems.

While the workstations can be mapped to the standard component *single station*, we have to create a new network for the robots. The main reasons are the requirements for tracking the state of the robot and considering its process sequence as the simulation progresses. This network is duplicated for all robots needed in the model. The inserted robot networks differ only in their underlying process table. Table 1 shows an example of some typical process steps. In addition to the entries shown, the simulation model contains consumption profiles for the respective process types. For visualization and analysis of the electric load, a group of *single stations* is included in the robot networks to model the different states of the robot. This group is used to represent the states working, production-ready, blocked, paused and disordered. Within this group, exactly one station can be occupied by a movable material flow object. This allows to model the exclusivity of states. Furthermore each robot network contains a *single station* for the modeling of disruptions. Using this station, service intervals as well as planned or random downtimes can be covered. Regarding the grippers, a further group of *single stations* represents the gripper states open and closed. Because some of the grippers have to hold more than one part the gripper contents are modeled as *parallel stations*.

Table 1: Process sequence for a single robot. Process steps set in italics are results of the optimization.

No.	Process name	Process type	Duration [s]	Start station	Target station
0	*peak-load optimization offset*	optimize	152.17		
1	approach S230	move	3.20	S235	S230
2	*peak-load optimization*	optimize	12.95		
3	laser welding	work	1.80	S230	S230
⋮					
6	approach home position	move	4.50	S230	S230
7	approach S235	move	3.30	S230	S235
8	*peak-load optimization*	optimize	10.27		
9	laser welding	work	1.80	S235	S235
⋮					

The manual feeding of parts in the real production line is represented in the model by *sources*. Under the assumption that the production is not delayed by interruptions of the supply of material, the flow objects are inserted without a creating time duration. A special case in the modeling of material flow is the occupation of a workstation by a robot holding a flow object. An example for this is a stationary glue gun where the part is slowly moved by a robot to apply the glue. Both components must be occupied with a single flow object. In the model the component is placed on the station, but the robot remains in working state and is waiting for the workstation to finish.

This and the state representation may not be straightforward uses of the offered functionalities in a discrete-event-oriented simulation system, but are a convenient way to model this extra behavior. In general, it is still a material flow simulation and most features of this paradigm stay intact.

2.4 Procedure of Simulation

The robot networks have to be initialized at the start of the simulation. Mobile flow objects are generated and placed on the corresponding *single stations* to generate the initial state of the robot. At the start of the simulation all robots are in operational-state. For the first process of each robot, a start event is added to the event queue of the simulation engine. The start events are logged with a processing status. This is one of four integer values, describing the process as planned but not started (0), started and currently active (1), started but currently stopped (2) or finished (3).

During the simulation, the event queue is processed and based on the conditions it is decided, if a process can start, resume, finish or has to stop. When a process finishes, the following process of the respective robot is inserted into the list with state 0. As soon as the event queue is traversed without modifications, the next occurring event is determined. Then the simulation time proceeds until this new event. If no random events are triggered in the model, the simulation will reach a steady state after the transient oscillation phase. In this state the entire system runs a fixed repeating process sequence.

We assign a consumption profile to each process of the robots. When a process ends in the simulation, an entry is generated in a table to log start time, duration and the associated consumption profile. This table provides the basis for the following analysis of the energy consumption. Any waiting-event of a robot due to material flow also generates an entry in the table.

For the creation of the simulation model no data about availabilities or cycle times of specific workstations were used. We compared the results of our simulation with the objectives concerning output quantity and cycle times given by the planning engineers. In the model these are attained sufficiently accurate which can be regarded as a passed plausibility check.

2.5 Simulation Results

The discrete-event approach is abandoned in the analysis, because no events are created for peak-loads and changes in the energy demand during the time period of a process. Instead, the table mentioned in Section 2.4 provides information about the starting time of each process, the executing resource, the used process type and the duration of each process. This table provides the base for the calculation and assignment of consumption profiles. The *move* processes of the robots have a duration time between one and several seconds. It is not plausible that very short processes cause full M-shaped power loads. Therefore, when the process duration is smaller than 2.5 s, the consumption profile will be adjusted by lowering peak-loads and raising the plateau in addition to pure time scaling. The power value is calculated on a scale with increments of hundreds a second. Thus, the energy-consumption can be determined for every point in time for single robots, groups of robots and for the whole production line.

The result can be visualized in form of power-time-diagrams. Figure 3 shows such a diagram for a part of the production line with 12 robots. The time interval spans a production cycle of about a minute in steady state of the simulation. During normal operation the overall electric power oscillates between about 11 to 18 kW. The value of peak-loads and the points in time when they occur as well as the involved robots can easily be identified. Figure 4 shows the total energy demand for a group of four robots. Here the superposition of the M-profiles can be easily recognized. If two or more robots start their *move*-processes simultaneously, especially high peak-loads arise. Furthermore occasionally *wait*-processes follow directly

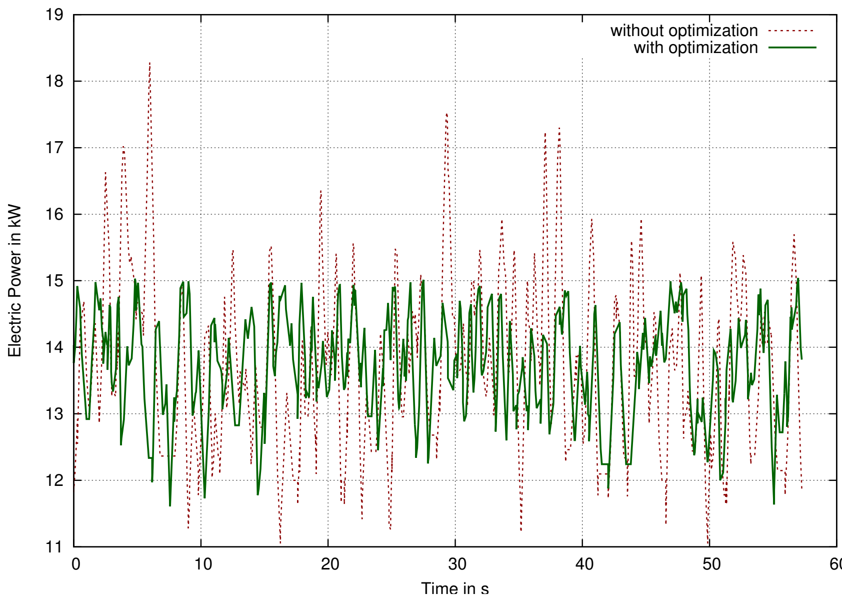

Figure 3: Comparison of the total electric power of the production line for the simulation without (red) and with additional wait period. The cycle time of both variants is the same but the peak-loads differ considerably.

after peak-loads. This is because the original sequence depends only on conditions for executing a process. As soon as the conditions for executing a process are fulfilled, the corresponding robot performs its tasks. Otherwise the robot executes a wait-process until all conditions are fulfilled. These wait-processes converge to constant time blocks in the steady-state of the simulation. Hence the question arises if an adjustment of the starting times of the move-processes of robots can reduce the peak-load. In the next section we present an optimization model for determining precise starting times for each process and appropriate, though possibly not minimal, additional wait periods so that the peak-load is minimized. In order to distinguish these additional wait periods from the original ones we introduce a *peak-load optimization* process type in the simulation model, which is only used for the insertion of time variable slices in the production process sequence.

3 PEAK-LOAD OPTIMIZATION

The simulation shows that there are potentials to reduce the peak-load of the robots because the simulated cycle time of the production line is greater than the duration of the processes each robot has to perform and so waiting times occur. As described above in the simulation each robot performs its task as early as possible. But in general there is the possibility to suspend the execution of the processes at a station. This allows to shift the peak-loads of the robots in time slightly. A mathematical programming model to reduce the maximal workload of the machines is also used in Klemmt et al. (2010). Fang et al. (2012) included an upper bound on the maximal peak-load in models for a flow shop problem.

For choosing the forced waiting times so as to minimize the maximal total load for the production line we present an optimization model. The optimal solution of this assigns a starting time to each process of the robots whenever a waiting period is allowed before a robot starts its given sequence at a station. For a given cycle time T_C we ask for a schedule of the processes performed by the robots. During the cycle time a given number of objects should be produced, in our case car body subassemblies. This schedule can be periodically repeated for, *e. g.*, a shift. The main modeling idea is to construct a graph for each robot $r \in R$, with set of robots R, and for each station $s \in S$, with set of stations S. A robot graph represents the sequence of the stations the robot passes and a station graph represents the order of the robots at the

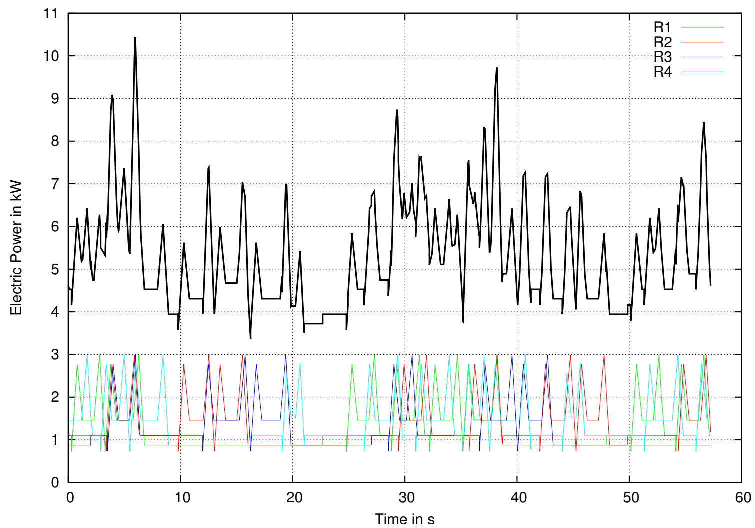

Figure 4: Power-time-diagram for a group of four robots.

station. For the convenience of the reader we assume in the following that each robot visits each station at most once and so that in a given order in which the robots have to perform their sequences at a station a robot appears at most once. Let $o_r = (s_0^r, s_1^r, \ldots, s_{k_r-1}^r)$, $s_0^r, s_1^r, \ldots, s_{k_r-1}^r \in S$, denote the sequence of the k_r stations robot $r \in R$ has to pass and $o_s = (r_0^s, r_1^s, \ldots, r_{k_s-1}^s)$, $r_0^s, r_1^s, \ldots, r_{k_s-1}^s \in R$ the sequence of k_s robots that work at a station $s \in S$. Based on these sequences we introduce the following graphs that contain exactly one cycle induced by o_r resp. o_s.

For a graph $G_r = (V_r, A_r)$ of robot $r \in R$ the node set V_r consists of all stations $s \in S$ with $s = s_i^r$ for an $i \in \{0, 1, \ldots, k_r - 1\}$ and the set of arcs A_r contains the arcs $(s_i^r, s_{(i+1) \bmod k_r}^r) \in A_r, i = 0, \ldots, k_r - 1$. Similarly, in the station graph $G_s = (V_s, A_s)$ of $s \in S$ the node set V_s consists of all $r \in R$ with $r = r_j^s$ for a $j \in \{0, 1, \ldots, k_s - 1\}$ and the set of arcs contains all arcs $(r_j^s, r_{(j+1) \bmod k_s}^s) \in A_s, j = 0, \ldots, k_s - 1$. As examples the graphs $G_{r'}, G_r, r', r \in R$, and $G_s, s \in S$, are illustrated in Figure 5.

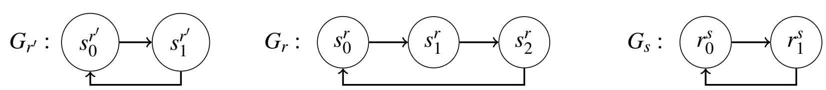

Figure 5: Graphs $G_{r'}, G_r$ for robots $r', r \in R$ with $o_{r'} = (s_0^{r'}, s_1^{r'}), o_r = (s_0^r, s_1^r, s_2^r)$ and graph G_s for a station $s \in S$ with $o_s = (r_0^s, r_1^s)$.

The graphs $G_r, r \in R$, and $G_s, s \in S$, are time-expanded to allow the assignment of certain time steps to the processes. For an overview over the theory on and the algorithms for *dynamic networks flows* we refer the reader to Skutella (2009) and the references therein. We introduce the discretization width T_W, *e. g.* 0.25 seconds, that is the duration of the period that is subsumed in one time step. The number of time steps is $n_T := \lceil T_C / T_W \rceil$ and the set of time steps is denoted by $T = \{0, 1, \ldots, n_T - 1\}$. For the time-expansion of the graphs we need the number of time steps d_i^r reflecting the duration of the subprocesses at station s_i^r and the number of time steps u_i^r for the transfer from s_i^r to the next station in order, *i. e.*, $s_{(i+1) \bmod k_r}^r, r \in R, i \in \{0, 1, \ldots, k_r - 1\}$. In order to preserve the property that a solution of the optimization model can be realized in the simulation using the original durations the quotient of duration and discretization width should be rounded up. This allows to set up the time-expanded graphs $G_r^T = (V_r^T, A_r^T), r \in R$, and $G_s^T = (V_s^T, A_s^T), s \in S$, with node sets $V_r^T := \{(s_i^r, t) \colon s_i^r \in V_r, t \in T\}$, $V_s^T := \{(r_i^s, t) \colon r_i^s \in V_s, t \in T\}$ and sets of arcs to be defined next. For $r \in R$ the set of arcs consists of $A^T_{r,1} \cup A^T_{r,2} = A^T_r$ with set of wait arcs

$$A^T_{r,1} = \{((s^r_i,t),(s^r_i,(t+1) \bmod n_T)) \colon i = 0,1,\dots,k_r-1, t \in T\}$$

and set of arcs corresponding to the subprocesses performed at a station and the transfer to the next station

$$A^T_{r,2} = \{((s^r_i,t),(s^r_{(i+1) \bmod k_r},(t+d^r_i+u^r_i) \bmod n_T)) \colon i = 0,1,\dots,k_r-1, t \in T\}.$$

Note that the rule that a robot goes to its next station directly after processing its task at the current station, *i. e.*, waiting is only allowed before the execution of tasks at a station and not after it, allows to subsume processing and moving in one arc each. The station graphs contain the arcs

$$A^T_{s,1} = \{((r^s_i,t),(r^s_i,(t+1) \bmod n_T)) \colon i = 0,1,\dots,k_s-1, t \in T\}$$

that represent the possibility that no robot is working at the station at time step t, *i. e.*, $r^s_i, i \in \{0,1,\dots,k_s-1\}$, has performed its tasks at station s but the following robot has not started its procedure, and the arcs

$$A^T_{s,2} = \{((r^s_i,t),(r^s_{(i+1) \bmod k_s},(t+d^{r^s_i}_k) \bmod n_T)) \colon i = 0,1,\dots,k_s-1, t \in T, s^{r^s_i}_k = s\}$$

that correspond to the robots performing their tasks at $s \in S$ with $A^T_s = A^T_{s,1} \cup A^T_{s,2}$. The time-expansion of the graphs $G_{r'}, G_r, G_s$ displayed in Figure 5 with $s^{r'}_0 = s, s^r_0 = s, d^{r'}_0 = 2, u^{r'}_0 = 1, d^{r'}_1 = 1, u^{r'}_1 = 1, d^r_0 = 1, u^r_0 = 2, d^r_1 = 1, u^r_1 = 0, d^r_2 = 1, u^r_2 = 1, r^s_0 = r', r^s_1 = r$ and $n_T = 9$ is illustrated in Figure 6.

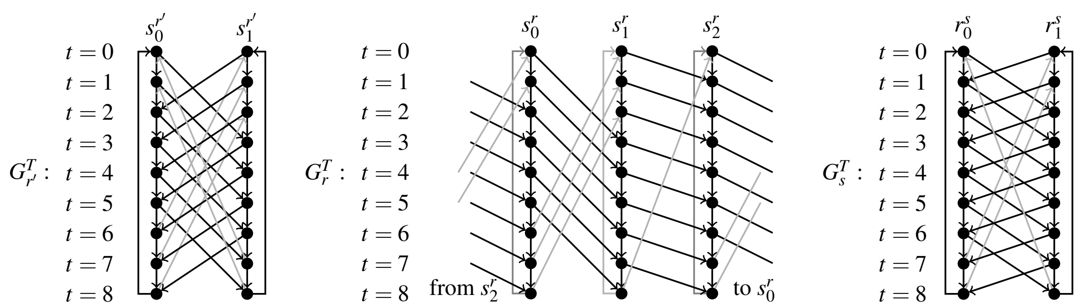

Figure 6: Time-expansion of graphs $G_{r'}, G_r, G_s$ displayed in Figure 5 with $s^{r'}_0 = s, s^r_0 = s, d^{r'}_0 = 2, u^{r'}_0 = 1, d^{r'}_1 = 1, u^{r'}_1 = 1, d^r_0 = 1, u^r_0 = 2, d^r_1 = 1, u^r_1 = 0, d^r_2 = 1, u^r_2 = 1, r^s_0 = r', r^s_1 = r$ and $n_T = 9$.

Based on the time-expanded graphs we set up an optimization model. The idea is that the control of a robot corresponds to a directed cycle in each time-expanded network. If in a robot graph G^T_r the cycle uses a wait arc the robot waits at the corresponding station and if a work and transfer arc is used the robot performs its task at the respective station and moves on to the next station. Figure 7 displays three possible cycles in the graphs $G^T_{r'}, G^T_r, G^T_s$ that are shown in Figure 6. Robot r' is waiting in time steps $t = 0,1$, starts its processes at $s^{r'}_0$ at time step $t = 2$ and arrives at station $s^{r'}_1$ at $t = 5$, after waiting a further time step it performs its tasks at $s^{r'}_1$ and moves back to $s^{r'}_0$. This procedure repeats all nine time steps. In time steps $t = 0,1$ robot r waits at s^r_2, starts its processes there at $t = 2$ and arrives at s^r_0 at $t = 4$. Immediately after this movement r performs its task at s^r_0 and moves to s^r_1 where it waits one time step before the execution of its processes and the movement back to s^r_2. Similarly, robot $r^s_0 = r$ works at s starting at $t = 2$ and exactly when r^s_0 moves to its next station r^s_1 performs its task at s ($t = 4$). After that no robot is working at s for five time steps, then the schedule is repeated.

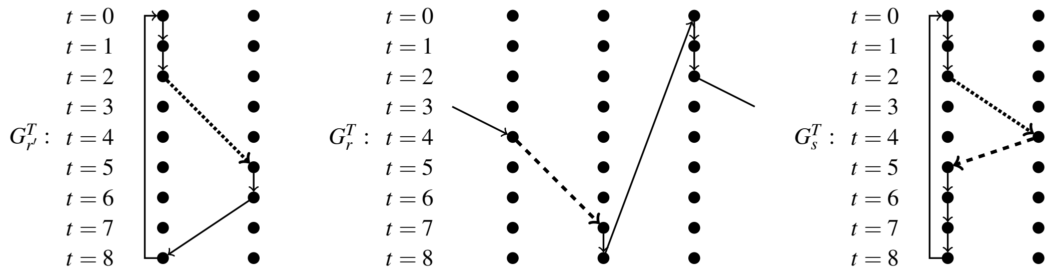

Figure 7: Possible directed cycles in the graphs $G^T_{r'}, G^T_r, G^T_s$ displayed in Figure 6. The variables corresponding to the dashed arcs as well as the ones to the dotted arcs fulfill the coupling constraints (6).

Each cycle has to fulfill certain properties to ensure that each robot $r \in R$ resp. each station $s \in S$ is in exactly one state in all time steps. We assign a binary variable x^r_a to each arc $a \in A^T_r, r \in R$, and y^s_a to each arc $a \in A^T_s, s \in S$, with the interpretation

$$x^r_a = \begin{cases} 1 & \text{if } r \in R \text{ is in the state associated with } a = ((s^r_i,t),(s^r_j,t')) \in A^T_r, \text{ beginning at time step } t, \\ 0 & \text{otherwise.} \end{cases}$$

Possible states for $r \in R$ are waiting at a station s_i and performing some processes at station s_i with subsequent movement from station s_i to the next in order. Analogously for $a \in A^T_s$ the variable

$$y^s_a = \begin{cases} 1 & \text{if } s \in S \text{ is in the state associated with } a = ((r^s_i,t),(r^s_j,t')) \in A^T_s, \text{ beginning at time step } t, \\ 0 & \text{otherwise.} \end{cases}$$

Here possible states are that a robot performs its subprocesses at s or that the station is waiting because no robot is working on s.

In the following we present constraints that ensure that each time-expanded graph G^T_r, G^T_s contains exactly one directed cycle that visits all corresponding stations resp. robots regarding the arcs whose variables are one. The *flow conservation constraints*

$$\sum_{a=((j_1,t_1),(j_2,t_2))\in A^T_r} x^r_a - \sum_{b=((j_0,t_0),(j_1,t_1))\in A^T_r} x^r_b = 0, \qquad r \in R, (j_1,t_1) \in V^T_r, \tag{1}$$

ensure that the total flow that enters a node $(j_1,t_1) \in V^T_r, r \in R$, has to leave it again and vice versa. Besides, in total a robot $r \in R$ has to wait exactly $n_T - \sum_{i=0}^{k_r-1}(d^r_i + u^r_i)$ time steps, that is the number of time steps in total reduced by the number of time steps that are needed for performing the tasks at the stations and for the movement processes between the stations by r,

$$\sum_{a\in A^T_{r,1}} x^r_a = n_T - \sum_{i=0}^{k_r-1}(d^r_i + u^r_i), \qquad r \in R. \tag{2}$$

Furthermore a robot $r \in R$ has to perform its subprocesses exactly once at each of its k_r visited stations,

$$\sum_{a\in A^T_{r,2}} x^r_a = k_r, \qquad r \in R. \tag{3}$$

Similar constraints force a correct flow on the station graphs $G^T_s, s \in S$,

$$\sum_{a=((j_1,t_1),(j_2,t_2))\in A^T_s} y^s_a - \sum_{b=((j_0,t_0),(j_1,t_1))\in A^T_s} y^s_b = 0, \qquad s \in S, (j_1,t_1) \in V^T_s, \tag{4}$$

$$\sum_{a\in A^T_{s,1}} y^s_a = n_T - \sum_{r\in R}\sum_{i\in\{0,1,\dots,k_r-1\}:\ s^r_i = s} d^r_i, \qquad s \in S. \tag{5}$$

The total flow entering a node $(j_1, t_1) \in V_s^T$ equals the total flow leaving that node and the number of waiting arcs is known in advance. Note that in contrast to (2) for the robot graphs the transfer times are not counted. This allows that the next robot in order can work at this station while the last robot is moving to its next station.

The *coupling constraints*

$$x^r_{((s^r_i,t),(s^r_{(i+1)\,\mathrm{mod}\,k_r},(t+d^r_i+u^r_i)\,\mathrm{mod}\,n_T))} = y^{s^r_i}_{((r^{s^r_i}_k,t),(r^{s^r_i}_{(k+1)\,\mathrm{mod}\,k_{s^r_i}},(t+d^r_i)\,\mathrm{mod}\,n_T))}, \quad r = r^{s^r_i}_k \in R, i \in \{0,\dots,k_r-1\}, t \in T, \quad (6)$$

relate the robot and the station graphs. A robot $r \in R$ has to perform its tasks at station $s^r_i, i \in \{0,1,\dots,k_r-1\}$, at the same time as the associated robot works at station s^r_i. With the constraints above and the requirements $x^r_a \in \{0,1\}, r \in R, a \in A^T_r$, and $y^s_a \in \{0,1\}, s \in S, a \in A^T_s$, it is also implied that at most one single robot is allowed to work at a station, so parallel execution of processes at the same station is forbidden.

For the objective function an extra variable $z \geq 0$ is introduced. It allows to determine the maximal total power over all time steps $t \in T$. Let $e^r_{(s^r_i,k)}$ denote the electric power of robot $r \in R$ at station $s^r_i \in S$ at time steps $k \in \{0,1,\dots,d^r_i+u^r_i-1\}$, *i. e.*, in the time period $[k \cdot T_W, (k+1) \cdot T_W]$ while it performs its tasks at s^r_i and moves to the next station, and let $e^r_{(s^r_i,w)}$ denote the electric power if robot $r \in R$ waits at station $s^r_i, i \in \{0,1,\dots,k_r-1\}$. It is not clear in advance how the electric power value at a time step should be chosen regarding the time discretization. If one wants an upper bound on total maximal load of all robots one should use the maximal load in the period for the time step. Alternatively, one could use the average load.

With

$$z \geq \sum_{r\in R}\sum_{i=0}^{k_r-1}\sum_{l=0}^{d^r_i+u^r_i-1} \left(x^r_{((s^r_i,(t-l)\,\mathrm{mod}\,n_T),(s^r_{(i+1)\,\mathrm{mod}\,k_r},(t-l+d^r_i+u^r_i)\,\mathrm{mod}\,n_T))} \cdot e^r_{(s^r_i,l)}\right) + \sum_{r\in R}\sum_{i=0}^{k_r-1} \left(x^r_{((s^r_i,t),(s^r_i,(t+1)\,\mathrm{mod}\,n_T))} \cdot e^r_{(s^r_i,w)}\right), \qquad t \in T, \quad (7)$$

the variable z is an upper bound on the total electric load at each time step $t \in T$. Indeed, in (7) we sum up the electric load of all processes that have an effect on that time step, *i. e.*, all processes that start at this time step or are still running at this time step because their starting time $\tilde{t}$ is close to t. Now the objective function simply reads

$$\min z. \quad (8)$$

The model (1)–(8) presented above can be solved using a solver for mixed integer linear programs. We used the modeling language Zimpl 3.2 to create an lp-file and solved this using Gurobi 4.6 (for further information we refer the reader to Koch (2004) and the webpages http://zimpl.zib.de/ and http://www.gurobi.com/). Because the problem is periodic the starting time of one process of one robot should be fixed to break the symmetry, *e. g.*, $x^{\bar{r}}_{((s^{\bar{r}}_0,0)(s^{\bar{r}}_1,d^{\bar{r}}_0+u^{\bar{r}}_0))} = 1$ for a robot $\bar{r} \in R$. This reduces the size of the solution space that has to be explored in a branch-and-cut algorithm. A problem of time-expanded networks is that the number of variables explodes if the discretization width is small. In fact, the solvers might even have problems in computing the value of the root relaxation in reasonable time. So a two-stage approach seems to help to get a good solution fast. In a first stage one solves the problem with a large discretization width and uses this solution as a starting point for a finer discretization limiting the solution space by fixing all variables of robot graphs to zero that correspond to time steps that are not near, *e. g.*, in a corridor of two seconds, to the starting times in the solution of the model with larger discretization. The result of the optimization is displayed in Figure 3. Note, that a calculation of start offsets for the robots is needed, compare Table 1, to integrate the optimized solutions in the simulation.

4 CONCLUSION AND OUTLOOK

We developed and demonstrated a new approach for reducing the peak-loads in production lines with several energy consumers without reducing the throughput by shifting execution periods appropriately. First we presented how energy consumption profiles can be included into a material flow simulation to model the dynamic load behavior of the production line. The simulation showed that the robots need to wait between the execution of their given process sequences. This inspired the introduction of predetermined wait periods into the process sequences in order to minimize the peak-loads in the second part of this paper.

For the purpose of finding optimal wait periods we developed a mathematical model, that can be solved by general purpose integer programming software. This resulted in an optimized arrangement of the individual process steps and thus in a beneficial superposition of the energy profiles. Given our use-case and the assumptions we made, the peak-load dropped by nearly 20 percent. If needed the M-shaped energy profiles can easily be scaled or otherwise adjusted without changes in the simulation and optimization procedures.

Better knowledge of the power demand allows to improve the dimensioning of infrastructure components. There are several benefits of smaller infrastructure components like cables and transformers. Usually they are cheaper and bind less valuable resources. Many electric devices reach their best degree of efficiency only at specific load levels. A well balanced power demand curve allows to operate them more often at these load levels and saves energy.

For further development not only other types of consumers should be integrated in the simulation but also the optimization goals might require adaptations. For instance the technical capability of brake energy recuperation within groups of robots would ask for different process plans. Meike and Ribickis (2011) discuss so called *energy teams* of robots and estimate the saving potential. The authors rely on peak-load extinction by random effects between several spot welding robots. Respective modifications of the process sequences by mathematical optimization would allow an increased exploitation and make the principle available for handling applications, where random effects are less likely.

Instead of using waiting periods as a buffer for delaying process steps one could also consider a temporal stretching of processes. The robot would move with lower velocity, which would lead to lower peak-loads and save energy. Such a *controlled speed scaling* needs further investigation of robot behavior and corresponding modeling options.

Accuracy of production planning is inherently limited by the viable efforts. The power consumption profiles may be refined with results of extensive measuring programs, but when some key factors simply cannot be determined in certain planning stages, they will remain approximations. In mathematical optimization there are approaches that allow to handle these uncertainties concerning the real energy demand of the robots and the resulting total load of the production. In the field of *robust optimization* (Ben-Tal, Ghaoui, and Nemirovski 2009) solutions are preferred that are good even if the used information, *e. g.*, the energy demand, vary slightly. This would help to evolve mathematical load-peak optimization towards a standard item in planning engineers toolboxes.

ACKNOWLEDGMENTS

The presented work summarizes results of the research projects InnoCaT® and eniPROD®. The pre-competitive joint research project *Innovation Alliance Green Car Body Technologies* is funded by the *Bundesministerium für Bildung und Forschung (BMBF)* and supervised by *Projektträger Karlsruhe (PTKA)*. The Cluster of Excellence *Energy-Efficient Product and Process Innovation in Production Engineering* is funded by the European Union (European Regional Development Fund) and the Free State of Saxony.

REFERENCES

Ben-Tal, A., L. E. Ghaoui, and A. Nemirovski. 2009. *Robust Optimization*. Princeton University Press.

Fang, K., N. A. Uhan, F. Zhao, and J. W. Sutherland. 2012. "Flow shop scheduling with peak power consumption constraints". Technical report, Pursue University. Available online on http://www.optimization-online.org/DB_HTML/2012/03/3411.html.

Heilala, J., S. Vatanen, H. Tonteri, J. Montonen, S. Lind, B. Johansson, and J. Stahre. 2008, December. "Simulation-based sustainable manufacturing system design". In *Proceedings of the 2008 Winter Simulation Conference*, edited by S. J. Mason, R. R. Hill, L. Moench, O. Rose, T. Jefferson, and J. W. Fowler, 1922–1930. Piscataway, New Jersey: Institute of Electrical and Electronics Engineers, Inc.

Klemmt, A., J. Lange, G. Weigert, F. Lehmann, and J. Seyfert. 2010, December. "A multistage mathematical programming based scheduling approach for the photolithography area in semiconductor manufacturing.". In *Proceedings of the 2010 Winter Simulation Conference*, edited by B. Johansson, S. Jain, J. Montoya-Torres, J. Hugan, and E. Yücesan, 2474–2485. Piscataway, New Jersey: Institute of Electrical and Electronics Engineers, Inc.

Koch, T. 2004. *Rapid Mathematical Prototyping*. Ph. D. thesis, Technische Universität Berlin.

Meike, D., and L. Ribickis. 2011. "Energy efficient use of robotics in the automobile industry". In *15th International Conference on Advanced Robotics (ICAR)*, edited by M. Kruusmaa, X. Tan, R. Madhavan, and P. Fiorini, 507–511.

Moeller, A., M. Prox, M. Schmidt, and H. Lambrecht. 2009, December. "Simulation and optimization of material and energy flow systems". In *Proceedings of the 2009 Winter Simulation Conference*, edited by M. D. Rossetti, R. R. Hill, B. Johansson, A. Dunkin, and R. G. Ingalls, 1444–1455. Piscataway, New Jersey: Institute of Electrical and Electronics Engineers, Inc.

Putz, M., R. Neugebauer, A. Schlegel, T. Langer, E. Franz, and S. Lorenz. 2012. "Energy-Sensitive Production Control in Mixed Model Manufacturing Processes". In *Leveraging Technology for a Sustainable World*, edited by D. A. Dornfeld and B. S. Linke, 399–404. Berlin: Springer.

Putz, M., A. Schlegel, S. Lorenz, E. Franz, and S. Schulz. 2011. "Gekoppelte Simulation von Material- und Energieflüssen in der Automobilfertigung". In *Tagungsband der 14. Tage des Betriebs- und Systemingenieurs*, edited by E. Müller and B. Spanner-Ulmer, 135–144. Chemnitz.

Skutella, M. 2009. "An Introduction to Network Flows over Time". In *Research Trends in Combinatorial Optimization*, 451–482. Springer-Verlag.

Weinert, N., S. Chiotellis, and G. Seliger. 2011. "Methodology for planning and operating energy-efficient production systems". *CIRP Annals - Manufacturing Technology* 60 (1): 41–44.

AUTHOR BIOGRAPHIES

SÖREN LORENZ is a researcher at the Fraunhofer Institute for Machine Tools and Forming Technology. He holds degrees in automotive engineering and production planning from the University of Applied Sciences Zwickau and Chemnitz University of Technology. After several years working for an automobile manufacturer his current research interest includes new methods for factory planning with special consideration of resource efficiency. His email address is soeren.lorenz@iwu.fraunhofer.de.

MATTHIAS HESSE is a researcher at the Fraunhofer Institute for Machine Tools and Forming Technology. He received an Master of Science in Engineering and Computing from University of Technology Bergakademie Freiberg, Germany. His areas of research includes discrete event plant simulation and physical effects system simulation. His email address is matthias.hesse@iwu.fraunhofer.de.

ANJA FISCHER is a research assistant at the Chemnitz University of Technology (CUT) at the Department of Mathematics. She received a diploma in Mathematics from CUT, Germany. Her areas of research include mathematical methods of operations research, in particular integer programming and convex relaxations. Her email address is anja.fischer@mathematik.tu-chemnitz.de.