Proceedings of the 2013 Winter Simulation Conference
R. Pasupathy, S.-H. Kim, A. Tolk, R. Hill, and M. E. Kuhl, eds

SIMULATION-BASED OVERHEAD-CRANE SCHEDULING FOR A MANUFACTURING PLANT

Tao Zhang
Oliver Rose

Universität der Bundeswehr München
Department of Computer Science
D-85577 Neubiberg, GERMANY

ABSTRACT

The overhead-crane scheduling problem with spatial constraints has attracted extensive attention and lots of approaches are introduced to solve the problem. As we all know, in the manufacturing plant the crane scheduling is one part of the production scheduling. However, most of approaches concern the crane scheduling in isolation. In this paper, we include the crane scheduling problem into the production scheduling environment and combine them together to obtain an integrated schedule. A simulation-based optimization solves this integrated scheduling problem. A genetic algorithm is introduced to determine the allocation of machines and cranes. A simulation model referring to a queuing network is used to evaluate the crane and machine allocation results and provides the fitness value for the genetic algorithm. The sequences of operations (processing and transporting) on each machine and each crane are determined by using the dispatching rule LPT. A heuristic deals with crane collision events.

1 INTRODUCTION

Overhead cranes, commonly called bridge cranes, are a type of crane widely used in manufacturing plants to transport, load and unload in-process products or raw materials. An overhead crane consists of three major components (shown in Figure 1): a bridge which traverses along parallel overhead runways; a hoist & trolley which traverses along the bridge and lifts up and down; parallel runways which are fixed on the top of the building structure. In the manufacturing plant, generally more than one crane run on the same runways and machines are usually arranged beside the runways or between the parallel runways below the cranes. The cranes transport in-process products from one machine to another. Because the cranes share the same runways, they cannot move past one another. Crane interference is a main factor affecting the utilization of cranes.

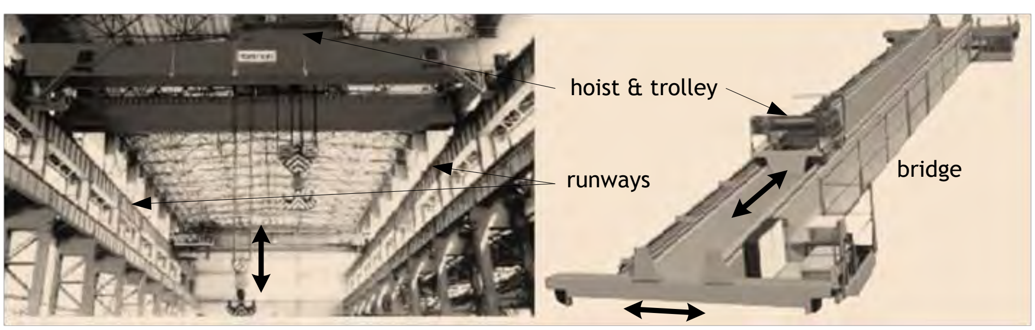

Figure 1: Structure of the overhead crane

The crane scheduling is a scheduling problem with spatial constraints, i.e. crane interference, which makes the crane scheduling problem more complex than the general job-shop problem. An abundant amount of literature focuses on quay crane scheduling in container terminals. The bridge of the quay crane is rigidly supported on two or more legs running on a fixed rail at ground level. Compared with overhead cranes in manufacturing plants, the quay cranes load (or unload) containers into (or from) ships rather than transporting items from one location on the runway to another. Therefore, it is difficult to apply the achieved results in the quay crane scheduling problem to the crane scheduling problem in manufacturing plants.

There are a few researchers focusing on overhead crane scheduling in manufacturing plants. Zhou and Li (2012) studied cyclic single crane scheduling problems with two parallel train oilcan repairing lines. A crane is used to move jobs between the workstations in two parallel lines. The objective is to schedule the moves to minimize the production cycle. A mixed integer linear programming model is developed to solve the problem. Aron et al. (2008) studied the problem of finding optimal space-time trajectories for two factory cranes or hoists that move along a single overhead track. The objective is generally to follow a production schedule as closely as possible. A specialized dynamic programming algorithm is used to solve the problem, which just needs to consider certain types of trajectories. Tang et al. (2009) studied a single crane scheduling problem motivated by batch annealing process in the iron and steel industry. A two-phase algorithm is constructed for the problem. In the first phase, a fully polynomial time approximation scheme (FPTAS) is developed for the assignment problem. In the second phase, a heuristic is proposed for the scheduling problem. Lieberman and Turksen (1981) investigated a crane scheduling problem with one operation per job when arrival patterns are static or dynamic and when the processing times are arbitrary.

These approaches generally assume that a production schedule is given. They make a crane schedule according to the production schedule. If no feasible crane schedule exists, the production schedule is revised and the crane schedule is generated again. For the production scheduling, crane capacity is considered to be infinite and the crane scheduling is not concerned in the production scheduling. The production scheduling and the crane scheduling are combined in a hierarchical manner. However, in this manner the case of jobs waiting for cranes still occurs frequently in practice and highly impacts the productivity. Therefore, it is necessary to involve the crane scheduling into the production scheduling to prevent the jobs from waiting for cranes.

The production scheduling is typically a flexible job shop problem. Two main types of solution procedures can be found in practice: heuristic procedures and meta-heuristic procedures have been applied to solve the problem and a near optimal schedule can be found within a reasonable time. The heuristic procedures include dispatching rules (Tay and Ho 2008), beam search (Wang and Yu 2010) and so on. There are many meta-heuristics procedures, such as local search (Yazdani et al. 2010), tabu search (Li et al. 2010), simulated annealing (Xia and Wu 2005), genetic algorithms (De Giovanni and Pezzella 2010; Zhang et al. 2011; Teekeng and Thammano 2012), ant colony algorithms (Rossi and Dini 2007) and particle swarm algorithms (Moslehi and Mahnam 2011), and so on.

In this paper, we include the crane scheduling problem into the production scheduling environment and combine them together to obtain an integrated schedule in one step. A simulation-based optimization algorithm is used to solve this integrated scheduling problem. The paper is structured as follows. Section 2 describes the problem in detail. The simulation-based optimization algorithm is stated in Section 3, including a simulation model and a genetic algorithm. A test is outlined in Section 4. The paper is concluded in the last section.

2 PROBLEM DESCRIPTION

Figure 2 shows one section of a manufacturing plant with crane transportation. We use it to describe the problem. As Figure 2 illustrates, some machines are laid beside the runways while some are assigned between the runways. For each machine there are two locations (stops) for loading and unloading in-process products. Sometimes these two locations overlap. A trolley is usually used to carry the in-process products (jobs) between the machine and these two stops. Jobs arrive at an arrival stop and depart from a departure stop by train or trolley. Jobs may also arrive at a machine and depart from another machine. Each job includes several operations in a specified sequence and each operation is performed on one of appropriate identical machines. Cranes transport the jobs from one machine to another. One crane may yield to another and move passively. Here we assume that,

- Jobs arrive at fixed intervals,
- The operations of a job and their sequence are given,
- A machine set for each operation of the jobs is given,
- Cranes are identical and can transport any jobs,
- The runways are straight,
- The velocity of cranes is constant as the cranes are moving, and
- The hoist & trolley has enough time to make the necessary lateral and vertical movements as the crane moves from one location to another. So we account only for the longitudinal movements of the crane along the runways.

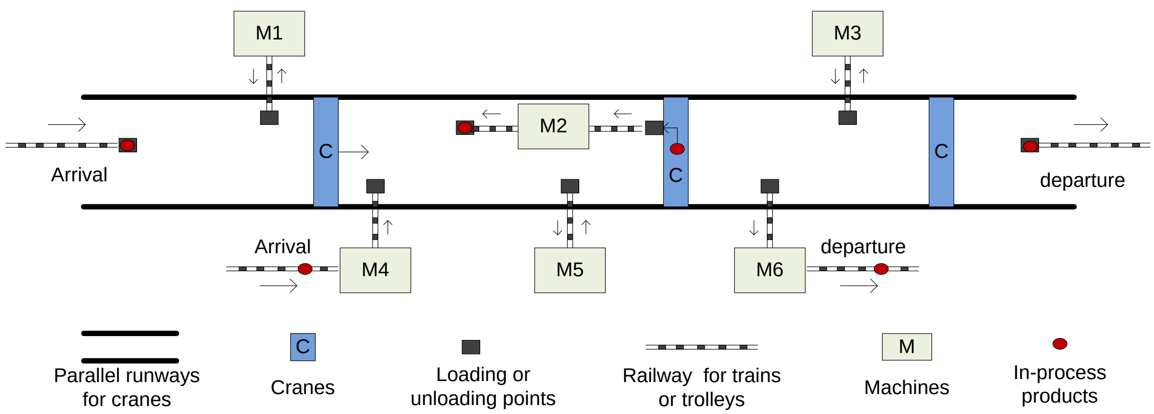

Figure 2: An example of a manufacturing plant with crane transportation

A scheduling problem is defined to determine:

- The allocation of each operation of a job to a machine ,
- The sequence of operations on each machine,
- The allocation of each transportation task of a jobs to a crane , and
- The sequence of transportation tasks on each crane.

And the computations are made under the following constraints:

- Each operation is assigned to exactly one machine,
- One job's operation starts only if the job's previous operation is completed and the job has arrived at the appropriate machine,
- A machine can process at most one job at a time and cannot be interrupted,
- No machine is free if there are operations waiting before them,
- The spatial constraints, i.e., cranes cannot move past one another
- Cranes cannot move out of the runways,
- Each transportation task is assigned to exactly one crane,

- A transportation task starts only if the related operation is completed,
- A crane can transport at most one job at a time and cannot be interrupted.

The objective is to minimize the makespan.

3 SIMULATION-BASED OPTIMIZATION

We combine a genetic algorithm with a simulation approach to solve the scheduling problem. The genetic algorithm is introduced to determine the allocation each operation of a job to a machine and the allocation of each transportation task of a job to a crane. A simulation model is used to evaluate the allocation results (just runs once following the allocation schedule represented by the related individual) and provides the fitness value for the evolution of the genetic algorithm. The simulation model is a queuing network with fixed and movable servers (machines and cranes). By using dispatching rule LPT (longest processing time) and LTT (longest transporting time) in the queuing network, the sequence of operations on each machine and the sequence of transportation tasks on each crane are determined. All constraints except the spatial constraints are met naturally in the queuing network. The collision between two cranes is allowed in the simulation model and a heuristic deals with the collision and decides about the movement of the collided cranes after a collision occurs.

3.1 Simulation of the Manufacturing Plant

The manufacturing processes are modeled as a queuing network. We use two different queuing networks with emphasis respectively on machines and cranes, shown in Figures 3 and 4, to illustrate the simulation model. Regardless of the crane transportation, the model is a normal queuing network composed of many single servers (machines). Each server has only one queue. If we only focus on the cranes, the model is a special queuing network. The servers (cranes) share all queues in different locations and the customers (jobs) can be served by anyone. The servers are movable and move customers from queues to their destinations. The customers arrive at a location, but depart from another location after being served. A one-dimensional coordinate system is used to determine the position of cranes and stops. The origin is set at one of the runways' end points and the direction of the sequence is towards the other end point.

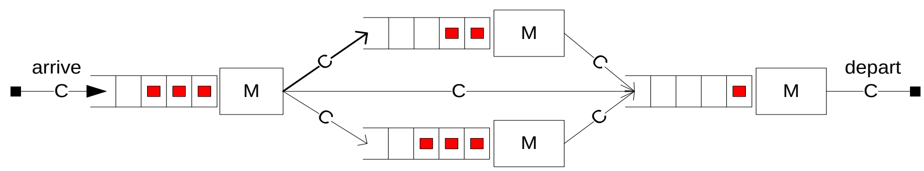

Figure 3: A queuing network with emphasis on machines

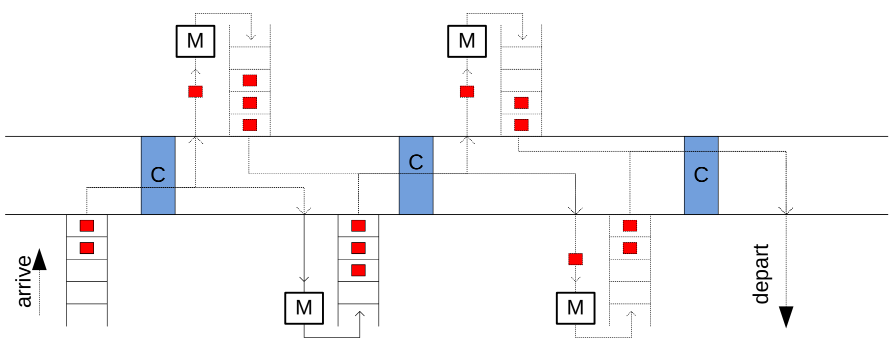

Figure 4: A queuing network with emphasis on cranes

3.1.1 Event-driven Simulation

For the simulation of the machine queuing network, two types of events, job arriving and operation finished, are considered. For the simulation of crane movements, five types of events are involved, including transportation task arriving, loading finished, unloading finished, crane colliding and crane crossing the stops. The events of crane colliding and crossing the stops are space-related and generated by special processes.

A crossing event is generated if the following conditions are met:

- A crane is moving towards a stop,
- There are no other cranes between the crane and the stop, and
- The crane will not collide with any other cranes while moving to the stop.

The time of the crossing event is $\tau_1 = \tau_{cur} + \left|x_{crane} - x_{stop}\right| / v$, where τ_{cur} is the current time, x_{crane} is the current position of the crane, x_{stop} is the stop's position, v is the velocity of the crane.

A collision event is generated if the following conditions are met:

- Two cranes are moving face to face or one crane is moving towards another crane while another crane stops, and
- There are no other cranes and loading/unloading stops between two given cranes.

The time of the collision event is $\tau_2 = \tau_{cur} + \left|x_{crane1} - x_{crane2}\right| / v$, where x_{crane1} and x_{crane2} are the two cranes' positions.

3.1.2 Heuristic Algorithm to Deal with Collisions

A heuristic algorithm is introduced to deal with the collisions. We use 4 states to describe the movement of cranes. They are ready (stopped and empty), loading/unloading, moving actively, moving passively. A list is created for each crane to store the names of cranes which are forced to move by this crane. We call the list the crane's passive-crane-list. The collision occurs in three situations,

- A moving crane (actively and passively) collides with a ready crane,
- A moving crane (actively and passively) collides with a loading/unloading crane,
- Two moving cranes collide with each other.

In the first situation, after collision the ready crane starts to move passively in the moving crane's direction and the name of the ready crane is added to the passive-crane-list of the moving crane. For the second situation, the moving crane stops and the name of the moving crane is added to the passive-crane-list of the loading/unloading crane. For the third situation, the collisions are handled according to Table 1.

If crane X moves actively, the operation X>>Y denotes the procedure that crane X turns back and moves in the same direction as crane Y and the name of crane X is added into the passive-crane-list of crane Y. If crane X moves passively, X0 represents the original crane which forces crane X to turn back and move in its direction; X>>Y denotes a procedure that crane X0 and the cranes in its passive-crane-list all turn back and move in crane Y's direction; the names of all these cranes are added to crane Y's passive-crane-list; Crane X0's passive-crane-list is cleared. To read Table 1, the reader has to replace X and

Y with A or B. The nearest destination rule (NDR) is used to compare the distances from cranes A and B to their destinations. If the distance from crane A to its destination is shorter than the distance from crane B to its destination, the operation A>>B will be carried out; otherwise, B>>A will be performed.

Table 1: Procedures responding to the collision events occurring between two moving cranes

Sub-situation		Procedure
State of crane A	State of crane B	
moving actively empty	moving actively, empty	Nearest destination rule (NDR)
	moving actively, loaded	B>>A
	moving passively, empty	If B0 is loaded, A->B; otherwise, NDR
	moving passively, loaded	A>>B
moving actively loaded	moving actively, loaded	NDR
	moving passively, empty	If B0 is loaded, NDR; otherwise, B>>A
	moving passively, loaded	NDR
moving passively empty	moving passively, empty	If A0,B0 are empty, NDR; If A0,B0 are loaded, NDR; If A0 is empty and B0 is loaded, A>>B; If A0 is loaded and B0 is empty, B>>A.
	moving passively, loaded	If A0 is empty, A>>B; otherwise, NDR
moving passively loaded	moving passively, loaded	NDR

3.2 Genetic Algorithm

The genetic algorithm solves the crane allocation and machine allocation problems. An individual is made up by two types of chromosomes. One type is related to the machine allocation. Another type is related to the crane allocation. The operations with the same type have the same set of optional machines. Operations of all jobs are grouped by the types of operation. The indices of machines which are selected to perform the operations in each group make up one machine allocation chromosome. The number of chromosomes related to machine allocations equals to the number of operation types. The indices of cranes which are selected to carry out the transportation tasks of all jobs make up one crane allocation chromosome. There is only one crane allocation chromosome. The fitness function is $f = makespan$. The makespan is obtained from the simulation model mentioned above. The simulation runs following the allocation results represented by the concerned individual. The time when the last job leaves the system is the makespan.

The crossover takes place between two matching chromosomes, as shown in Figure 4. The numbers and positions of genes for crossing over are not limited. Mutation is restricted to the sets of optional machines and cranes, and alters the gene value to one of other optional gene values. Figure 5 shows the mutation operation.

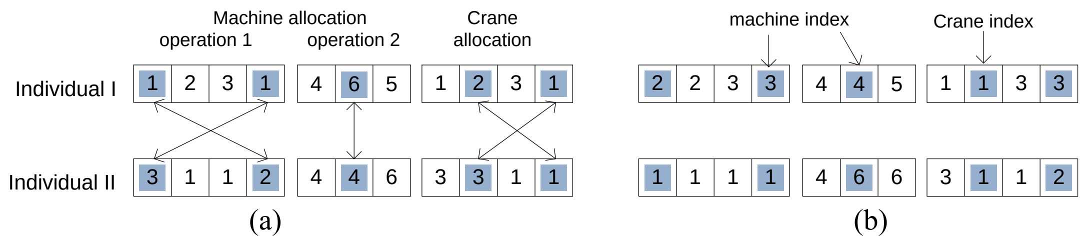

Figure 4: Crossover operations (a) before crossover (b) after crossover

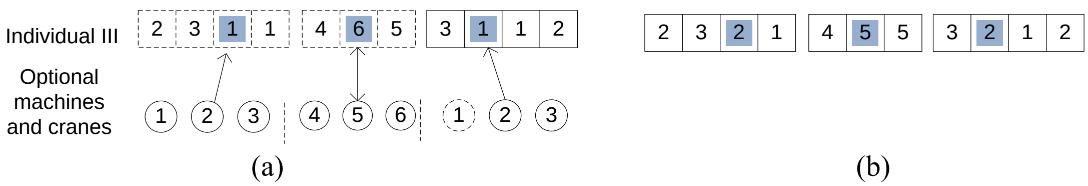

Figure 5: Mutation operations (a) before mutation (b) after mutation

The fitness proportionate selection, also known as roulette wheel selection, is used to select potentially useful solutions for reproduction. The probability of selection of individuals is proportionate to the fitness values. A random number $\varepsilon \in [0,1]$ is chosen. If the k-th individual meets the following condition, the individual will be selected.

$$\sum_{i=1}^{k-1} f_i / \sum_{i=1}^{N} f_i < \varepsilon \le \sum_{i=1}^{k} f_i / \sum_{i=1}^{N} f_i ,$$

where f_i is the fitness value of the i-th individual and N is the population size.

4 APPLICATION

We report about computational tests of a representative problem that is based on an actual industry scheduling situation. Five machines are arranged besides the runways. Three cranes travel on the runways. Two types of products are produced. The velocity of cranes is 1m/s. A schedule is created based on the one day production plan shown in Table 2.

Table 2: One day production plan

Product	Number	Release Interval (minutes)	Operations and appropriate machines
A	29	50.0	O1[M11,M12]->O2[M2]->O3[M31,M32]
B	44	33.3	O1[M11,M12] ->O3[M31,M32]

The parameters of the genetic algorithm are set as follows – generation number: 30, population size: 20, crossover probability: 0.8, mutation probability: 0.1. An individual consists of three chromosomes in this application. The indices of machines allocated to operations (O1, O3) of jobs make up two chromosomes. For operation O2, there is only one optional machine, so no allocation problem exists. The indices of cranes which are selected to carry out the transportation tasks of all jobs make up one chromosome related to crane allocations. Jobs depart the system from the machine M31 or M32, so there is no need to use cranes to carry jobs when the jobs depart. When the algorithm ends we can obtain a near optimal allocation schedule (as shown in Table 3) from the best individual. The fitness value of the best individual is 27.1 hours.

Table 3: The near optimal allocation schedule

Job	Machine allocation		Crane allocation			
	O1	O3	→O1	O1→O2	O1→O3	O2→O3
A01	M11	M32	C3	C2	-	C2
A02	M12	M31	C3	C3	-	C1
A03	M11	M31	C3	C1	-	C1
…	…	…	…	…	…	…
B01	M12	M31	C3	-	C3	-
B02	M11	M31	C2	-	C1	-
B03	M12	M32	C2	-	C2	-
…	…	…	…	…	…	…

The simulation runs one more time according to the near optimal allocation schedule to obtain the movement tracks of the cranes. Figure 6 shows a segment of the movement tracks. We can see that no interferences occur. The start times and the end times of jobs' operations and transportation tasks can also be obtained from the simulation. Figure 7 is a Gantt chart of the machines. It shows that there are no time interferences and machine interferences. We can also see that machines M11 and M12 are bottlenecks which work efficiently.

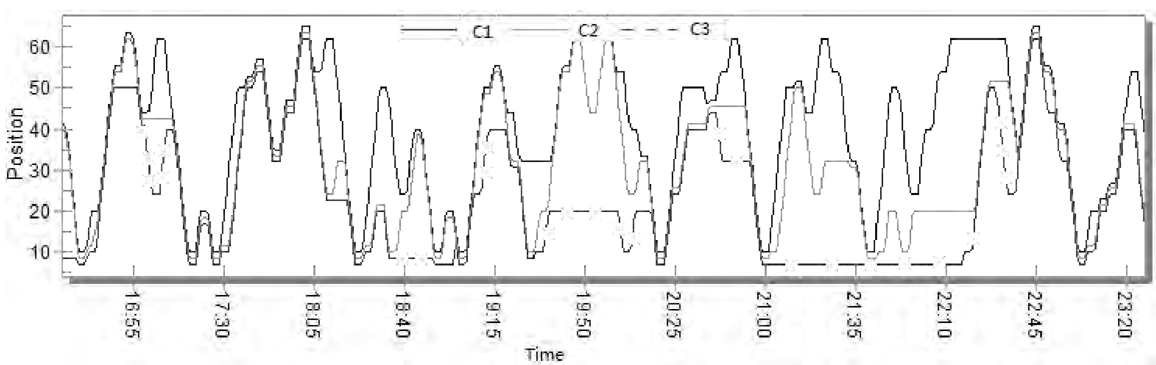

Figure 6: Movement tracks of the Cranes

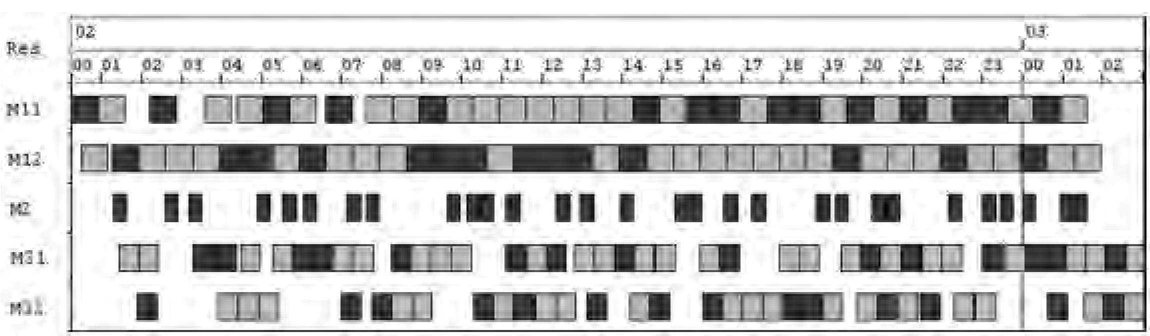

Figure 7: Gantt chart of the machines

In manufacturing plants, it makes no sense to schedule cranes without concern about production scheduling. The crane scheduling must be included in the production scheduling. Comparing to the hierarchical approach, the integrated approach we proposed is more concise and the schedule we made is closer to the actual condition.

5 CONCLUSION

The integrated scheduling problem combining the crane scheduling with the production scheduling is solved by our simulation-based optimization algorithm. An integrated schedule is made in one step, so the proposed approach avoids revising the production schedule according to the crane schedule. The simulation-based optimization algorithm simplifies the solution of the optimization problem. The simulation model referring to a queuing network can meet most of constraints naturally and uses the dispatching rules to make sequencing decisions. The crane collision is allowed in the simulation and a heuristic algorithm deals with the collision. The spatial constraints can be met easily in this way. The genetic algorithm only solves the allocation problem and the simulation provides the fitness values for the evolution. An application to a real manufacturing plant is outlined and the results show the validity of the proposed approach.

REFERENCE

Aron, I., L. Genç-Kaya, I. Harjunkoski, S. Hoda and J. N. Hooker 2008. "Optimal Movement of Factory Cranes." Tepper School of Business. Accessed July 15. 2013, http://repository.cmu.edu/tepper/143/.

De Giovanni, L. and F. Pezzella. 2010. "An Improved Genetic Algorithm for the Distributed and Flexible Job-Shop Scheduling Problem." *European Journal of Operational Research*. 200: 395-408.

Li, J.-q., Q.-k. Pan and Y.-C. Liang. 2010. "An Effective Hybrid Tabu Search Algorithm for Multi-Objective Flexible Job-Shop Scheduling Problems." *Computers & Industrial Engineering*. 59: 647-662.

Lieberman, R. W. and I. B. Turksen. 1981. "Crane Scheduling Problems." *A I I E Transactions*. 13: 304-311.

Moslehi, G. and M. Mahnam. 2011. "A Pareto Approach to Multi-Objective Flexible Job-Shop Scheduling Problem Using Particle Swarm Optimization and Local Search." *International Journal of Production Economics*. 129: 14-22.

Rossi, A. and G. Dini. 2007. "Flexible Job-Shop Scheduling with Routing Flexibility and Separable Setup Times Using Ant Colony Optimisation Method." *Robotics and Computer-Integrated Manufacturing*. 23: 503-516.

Tang, L., X. Xie and J. Liu. 2009. "Scheduling of a Single Crane in Batch Annealing Process." *Computers & Operations Research*. 36: 2853-2865.

Tay, J. C. and N. B. Ho. 2008. "Evolving Dispatching Rules Using Genetic Programming for Solving Multi-Objective Flexible Job-Shop Problems." *Computers & Industrial Engineering*. 54: 453-473.

Teekeng, W. and A. Thammano. 2012. "Modified Genetic Algorithm for Flexible Job-Shop Scheduling Problems." *Procedia Computer Science*. 12: 122-128.

Wang, S. and J. Yu. 2010. "An Effective Heuristic for Flexible Job-Shop Scheduling Problem with Maintenance Activities." *Computers & Industrial Engineering*. 59: 436-447.

Xia, W. and Z. Wu. 2005. "An Effective Hybrid Optimization Approach for Multi-Objective Flexible Job-Shop Scheduling Problems." *Computers & Industrial Engineering*. 48: 409-425.

Yazdani, M., M. Amiri and M. Zandieh. 2010. "Flexible Job-Shop Scheduling with Parallel Variable Neighborhood Search Algorithm." *Expert Systems with Applications*. 37: 678-687.

Zhang, G., L. Gao and Y. Shi. 2011. "An Effective Genetic Algorithm for the Flexible Job-Shop Scheduling Problem." *Expert Systems with Applications*. 38: 3563-3573.

Zhou, Z. and L. Li. 2012. "Optimal Cyclic Single Crane Scheduling for Two Parallel Train Oilcan Repairing Lines." *Computers & Operations Research*. 39: 1850-1856.

AUTHOR BIOGRAPHIES

TAO ZHANG is a Ph.D. student working on production planning and scheduling at the Department of Computer Science of the Universität der Bundeswehr München, Germany. From 2007 to 2009 he received his Master in metallurgical engineering with the subject of production planning and scheduling in iron and steel industry from Chongqing University, China. He is involved in modeling and simulation of complex system and intelligent optimization algorithms. His email address is tao.zhang@unibw.de.

OLIVER ROSE holds the Chair for Modeling and Simulation at the Department of Computer Science of the Universität der Bundeswehr, Germany. He received a M.S. degree in applied mathematics and a Ph.D. degree in computer science from Würzburg University, Germany. His research focuses on the operational modeling, analysis and material flow control of complex manufacturing facilities, in particular, semiconductor factories. He is a member of IEEE, INFORMS Simulation Society, ASIM, and GI, and has been the General Chair of WSC 2012. His email address is oliver.rose@unibw.de.