Proceedings of the 2008 Winter Simulation Conference
S. J. Mason, R. R. Hill, L. Mönch, O. Rose, T. Jefferson, J. W. Fowler eds.

SIMULATION OPTIMIZATION FOR INDUSTRIAL SCHEDULING USING HYBRID GENETIC REPRESENTATION

Marcus Andersson
Amos H. C. Ng
Henrik Grimm

Centre for Intelligent Automation
University of Skövde
Box 408, 541 28, SWEDEN

ABSTRACT

Simulation modeling has the capability to represent complex real-world systems in details and therefore it is suitable to develop simulation models for generating detailed operation plans to control the shop floor. In the literature, there are two major approaches for tackling the simulation-based scheduling problems, namely dispatching rules and using meta-heuristic search algorithms. The purpose of this paper is to illustrate that there are advantages when these two approaches are combined. More precisely, this paper introduces a novel hybrid genetic representation as a combination of both a partially completed schedule (direct) and the optimal dispatching rules (indirect), for setting the schedules for some critical stages (e.g. bottlenecks) and other non-critical stages respectively. When applied to an industrial case study, this hybrid method has been found to outperform the two common approaches, in terms of finding reasonably good solutions within a shorter time period for most of the complex scheduling scenarios.

1 INTRODUCTION

In the last decade, there have been extensive research in the field of production scheduling using simulation. Simulation modeling has the capability to represent complex real-world systems in details. In general, a simulation model built for scheduling can be quite different from an ordinary simulation model developed for the design and analysis of an existing or a proposed new system, especially if the model is used for generating detailed operation plans to control the shopfloor. Generally speaking, simulation-based scheduling approaches are derived from dispatching rule based approaches (Kiran 1998). In simulation-based scheduling, several dispatching rules can be used at different stages for making decisions on what parts to select for the next scheduling period. Basically, a dispatching rule is a rule of thumb that gives priority to a job among other jobs at a specific stage. This is why dispatching rules can also be called priority dispatching rules (PDRs). The most common PDRs are for example, shortest processing time (SPT), longest processing time (LPT), earliest due date (EDD), first come first served (FCFS), to name but a few. The number of rules can be infinite because it is possible to define new scheduling rules as the combinations of several other dispatching rules (Panwalkar and Wafik 1977)(Holtaus 1997). Generally speaking, a PDR-based simulation scheduling approach does not attempt to find an "optimal" schedule, but rely on knowing that one rule, or a combination of rules, performs better than another one. In comparison, using a meta-heuristic optimizer, such as a Genetic Algorithm (GA), to generate the optimal schedules directly, which is referred as a direct approach in this paper, may be advantageous if searching for "optimal" solutions is desired. Nevertheless, to generate a complete schedule using a GA-based simulation optimization may require very long computing time. This is usually impractical or even unacceptable if the result is needed for controlling the system in "real-time".

There are many studies that compare these two approaches and some of them provide results showing the use of GAs to generate detailed schedules can obtain better solutions than those ones obtained by using PDRs. For example, Sankar et al. (2003) use a GA for the scheduling of a Flexible Manufacturing System (FMS). Several objectives like customer satisfaction, machine utilization and total elapsed time are combined into a single combined objective function. A GA is coded in such way that the chromosomes represent the job sequences of which the manufacturing system has to follow in order to achieve the best schedule. The results obtained with the GA are then compared with the results obtained using six different dispatching rules including SPT, LPT, EDD, largest batch quantity (LBT), smallest batch quantity (SBQ) and highest penalty (HP). It has been found that the solutions generated by GA outperform the solutions obtained by using PDRs for this specific FMS.

978-1-4244-2708-6/08/$25.00 ©2008 IEEE

Kim et. al. (2007) made a comparison between the use of PDRs and GAs for solving the scheduling problem in a real manufacturing factory of standard hydraulic cylinders. Different dispatching rules have been used in this study, namely SPT, LPT, most work remaining (MWKR) and least work remaining (LWKR). When using GAs, different jobs to be performed by different machines are codified into a individual chromosome, then the different individual are selected following the "natural selection" in order to obtain the most advantageous objectives (minimize makespan and total tardiness). Again in this study, the researchers found that the GA-based approach outperforms the PDR-based one. At the same time, it is stated in their conclusion that "production manager could provide more effective and timely support … and obtain better solutions for the scheduling of orders using the two techniques in combination". A method that combines GA and PDR can be found in (Tanev et. al. 2004), where a hybrid evolutionary algorithm for the scheduling of a factory of plastic injection machine was developed. In their approach, they proposed a hybrid GA combined with the use of PDRs; a GA was used to evolve the different combination of dispatching rules and finally find which one provides the best schedule. The solutions were then evaluated by means of a fitness function conformed by the different parameters to be optimized. Similar approach can be found recently in (Tay and Ho 2008). In this kind of approach in which a GA chromosome is used to represent different combination of PDRs, it is referred as the indirect approach in this paper.

Through an industrial case study, a machining line in an automotive factory, the purpose of this paper is to illustrate that a hybrid representation of the direct and indirect approach may provide the combined strengths, i.e. able to find good solutions quickly (time-efficiently) for complex scheduling problems. It should be noted that this approach is different from a hybrid GA that combines GA with other local search algorithms like hill climbing (Candido 1998); in the current paper, hybrid means a combination of the direct and indirect genetic representation. The rest of the paper is as follows: Section 2 introduces the industrial case used in this study; Section 3 describes the direct, indirect and hybrid genetic representation used in the optimizations. The weight-based objective function of the optimization problem to handle multiple objective parameters of the industrial problem is revealed in Section 4. Presentation and discussion of the optimization results can be found in Section 5 and 6 respectively followed by the conclusion of the paper in Section 7.

2 AN INDUSTRIAL CASE STUDY

The real-world problem considered is a camshaft machining line at Volvo Cars in Sweden. The line is responsible for producing about 15 different camshaft variants. The machining line consists of 14 different machine groups with one to seven parallel machines in each group; totally 34 machines. Unlike an ordinary flow shop with parallel machines problem, each machine has its own processing time, physical capability and limitations, as well as variability in terms of failures and set-ups. There are many different types of products and each type of products has its own necessary operation steps in the line. The machining line is semi-automated with robots that feed machines inside the cells, but the loading, unloading and the decision of when or where to process different types of parts is decided by operators at each work area. Currently, batches of different product variants are scheduled to start in a specific order, in order to keep the variants in the finished goods stock above a security level over time, and to reach predetermined targets as early as possible each week. The security levels are important because these make it possible for the line to feed camshafts downstream even if a major breakdown of a machine occurs. Even though different batches are prioritized in a specific order, the operators usually re-schedule them to minimize the number of set-ups. The consequence of these manual decisions in the machining line is that some machines might be optimized, but not the overall performance of the line, especially when unexpected events, like urgent orders, machine failures occur.

A complex simulation model was therefore developed to model the existing line. The simulation model is primarily built for short-period scheduling. At the start of the simulation, the model is updated with the transient status in order to get a start condition that matches the real production line. In the Microsoft Excel user interface of the simulation model it is possible to set the scheduling scenario, as each production day/week is somehow deviate from another. A scheduling scenario might differ in many ways, such as material deficits, machine failures, quality problems which may generate critical deadlines, etc., all these have to be handled properly in the simulation model.

The simulation-based scheduling application utilizes OPTIMISE to run experiments and optimizations with the simulation model. OPTIMISE is a Web-based parallel and distributed computing platform that supports multiple users to run experiments and optimizations with different simulation systems (Ng et al. 2008). The camshaft machining line has started to use the simulation-based scheduling application (Andersson et al. 2007) for evaluation purposes. The results so far shows that the application is successful in finding good solutions in a reasonably short time.

3 HYBRID GENETIC REPRESENTATION

The GA encodes possible solutions as genomes and each genome instance represents a single solution to the problem – in this case an operation schedule. In many applications, the efficiency of GAs is determined mainly on how the domain problem is encoded in the genome and the representation has therefore been considered carefully in this study. The schedule can be represented both directly and indirectly for this problem, see Figure 1. In the indirect approach priority dispatching rules (PDR) are used.

	Representation
OP Group 1	Direct
OP Group 2	Direct
OP Group 3	PDR
OP Group n	PDR

Figure 1: Direct and indirect representation

The genome for this problem is designed to represent the schedule of the direct approach presented in Figure 2. The genome is implemented as a matrix in which each row corresponds to a specific batch and each column represents a machine. Each batch is scheduled over machines in those operation groups belonging to the product type's process flow.

The sequence consists of a list of batches where each batch consists of a specific product variant type, a deadline, a start position, and the number of parts. The adjacent batches of the same variant type forms a batch group, which will be used in the sequence mutation operator. Figure 2 below shows an example of a simplified genome with six batches scheduled on two operation groups.

	OP Group 1				OP Group n		
B1	50						50
B2		100			100		
B3			100	50			150
B4				100	100		
B5			50				50
B6	50					50	

Figure 2: Direct representation of an operation schedule

The task of the GA is to change the sequence and to fill in the solid light cells of the matrix while considering some of the constraints, such as batch size. However, most of the difficult constraints, such as a product variants required operation steps and allowed operations are handled in the input data interface (Andersson et al. 2007) and results in non-solid and solid cells in the matrix. The cart size is a user-defined constant set to be 50. In the indirect approach, when using PDRs, the genome is designed to represent the schedule presented in Figure 3.

	OP Group 1	OP Group n
B1	SPT	EDD
B2		
B3		
B4		
B5		
B6		

Figure 3: Indirect representation of an operation schedule

The genome has the same base as the direct representation, in which each row corresponds to a specific batch, but the batch allocation over machines is not needed. Instead the GA will change dispatching rules on the operation groups. Accordingly, in the current implementation all machines within one operation group will use the same dispatching rule.

It is also possible to use a hybrid between the direct and the indirect representation, see figure 4. When using the hybrid approach some of the machines will follow an exact schedule and some machines uses PDRs to determine the sequence order. Once an operation group has been chosen to use PDRs, following operation group also need to use PDRs in order to avoid deadlocks, such as buffer capacity restrictions.

	OP Group 1				OP Group n
B1	50				EDD
B2		100			
B3			100	50	
B4				100	
B5			50		
B6	50				

Figure 4: A hybrid of direct and indirect representation

3.1 Initialization

A first population, currently set to 50 candidate solutions, is generated satisfying all the constraints. All the solutions in first population have the same sequence order, sorted on position in line, deadline, and variant number, as this order is a good base further change. Batches carts are randomly allocated for the allowed operations and the dispatching rules are randomly set for each operation group.

3.2 Crossover

During each successive generation of the GA, a proportion of the existing population is selected to breed a new genera-

tion. Individual solutions are chosen for mating through tournament selection, i.e. taking the better of two randomly chosen solutions. Thus, solutions with higher fitness values are more likely to be selected, but a small number of solutions with less fitness values have some probability of being selected as well in order to keep a large diversity of the population.

From the pool of selected solutions, two solutions are chosen as parents and through crossover two new solutions are formed. The crossover operator applies to batches cart allocation over operations and operation groups dispatching rules, but not sequence order, which are simply inherited from the parents. The first child solution inherits sequence order from the first parent and the second child solution inherits sequence order from the second parent.

The crossover procedure consists of two steps. First, the batch cart allocation crossover is carried out using a uniform crossover operator by taking each batch's cart allocation randomly from one of the two parents, and then crossover is applied to the dispatching rules for operation groups using a uniform crossover operator by taking each operation group's dispatching rule settings randomly from one of the two parents.

3.3 Mutation

To maintain the genetic diversity from one generation to the next, the offspring solutions are mutated. The number of mutations is determined using a geometric distribution, where at least one mutation is always made. There are three different possible mutation types: Batch allocation mutation, dispatching rule mutation, and sequence mutation. One of these three different mutations can be chosen for each number of mutations depending on the probability of each mutation type.

3.3.1 Batch Allocation Mutation Operator

The batch allocation mutation operator is only applied to machines using the direct representation. In the mutation procedure, a batch is first selected for allocation mutation within an operation group. Then one cart of this batch is re-allocated on another parallel machine, provided that some other available machines exist within the chosen operation group.

3.3.2 Dispatching Rule Mutation Operator

The dispatching rule mutation operator is only applied to machines using the indirect representation. In the dispatching rule mutation procedure, an operation group is randomly chosen and a new dispatching rule is randomized for the operation group.

3.3.3 Sequence Mutation Operator

The same sequence mutation operator is used both for direct and indirect representation. The sequence of batches is grouped into batch groups, where each group consists of one to several batches of the same product variant.

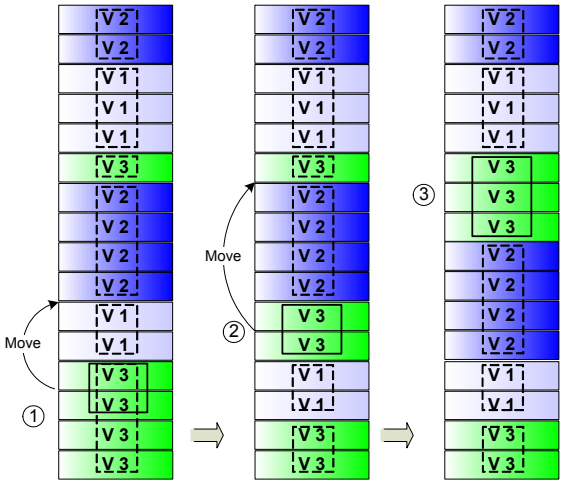

Figure 5: Sequence mutation operator

The figure above shows an example of the mutation procedure of moving one batch group:

1. First, one batch group is chosen randomly - in this case a group of four batches of V3 (Variant 3). The next step determines if the batch group is to be moved up or down, which in this case is up. Then there is a possibility of batch group splitting. In this case the first two batches in the group becomes a new group and will be the ones being moved upwards one step.
2. From the new obtained position of the batch group there is a chance of moving it even further one step by a probability of 0.8 (predefined sequence mutation rate) .
3. If the batch group reaches a batch group of the same variant type the moving procedure will stop and the two groups will be merged into one.

The advantage of grouping batches is that the mutation operator can do more in one mutation step, and thereby avoid some of the really poor solutions. For this problem it's generally better to move more than one batch at a time as moving only one batch could result in more set-ups. It will also make the schedules easier for the workers to follow as it indirectly will generate schedules with less variant shifts at the machines. Batch splitting will still make it possible to create whatever sequence desired. Stopping when reaching the same product variant (step 3 in figure 5) will automatically help keeping the internal order of batches within the same variant type.

4 WEIGHT-BASED OBJECTIVE FUNCTION FOR MULTI-OBJECTIVE SCHEDULING

All results, fitness values, simulation data, different Gantt charts etc. are presented for the user in OPTIMISE Browser (Andersson et al. 2007). A weight-based objective function has been used because the production planner needs to obtain the result after quite short period of time, and the user has no time to study separate sub-targets. Fitness is the sum of all the sub-targets fitness values.

$$F = f_{th} + f_{sh} + f_{tl} + f_{sa} + f_{st} \tag{1}$$

where, F is fitness, f_{th} is the fitness of Throughput, f_{sh} is the fitness of Shortage, f_{tl} is the fitness of the Target Levels, f_{sa} is the fitness of Stopped in Advance and f_{st} is the fitness value of the Setup Time.

Throughput is the number of products produced per hour measured on the whole simulation period. It is important to increase the overall throughput of the system. Not only is the performance mean important because a high variation could cause major losses. Therefore the objective function for throughput (f_{th}), Equation (2), was created according to the values of experts of the system.

$$f_{th} = w\left(k\sigma + \left(\tau - \mu\right)\right) \tag{2}$$

where, w is the objective weight for Throughput, μ is the mean value of Throughput, τ is the target level of Throughput, σ is the standard deviation and k is the weight of the standard deviation.

An essential objective for the camshaft machining line is keeping the security levels of the variants in the finished goods stock. If a variant's FGI level goes under the security level it results in a momentary shortage in equation 3.

$$ms = \left(\frac{cl - sl}{sl} \times 100\right)^2 \tag{3}$$

where, ms is a momentary shortage, cl is the current stock level and sl is the security level of one product variant. All momentary shortage is summed and at the end of the simulation the variant shortage is calculated.

$$vs = \sqrt{\frac{\sum_{i=1}^{n} ms_i}{n}} \tag{4}$$

where, vs is variant shortage and n is the total number of measures carried out. A robust schedule is requested, so based on a quadratic loss function (Sanchez 2000), the fitness of shortage in Equation (5), is a good approximation in the representation of this objective as it penalizes small values of deviation to target little, but penalizes large ones much.

$$f_{sh} = \sum_{i=1}^{n} w_i \left(\sigma_i^2 + \mu_i^2\right) \tag{5}$$

where, μ is the performance mean and σ is the standard deviation, w is the product variant's weight, and n is the total number of product variants.

The fitness of target levels measures the ability to reach necessary targets for different product variant groups. Target levels are measured in FGI once a week and the best result is when a variant is equivalent to the target level. A measured level that is less than the target level gets a higher penalty than one that is over the target level.

$$f_{tl} = \sum_{i=1}^{n} w_i \left(\sigma_i^2 + \mu_i^2\right) \tag{6}$$

where, μ is the performance mean, σ is the standard deviation, n is the number of target levels, and w is penalty weight, which depends on whether the measured value is above or below the actual target level.

Stopped in advance is based on when the production is shut down each week. Machines can shut down when the corresponding shortage and target levels that the machine produces is reached for the week.

$$f_{sa} = -w\mu \tag{7}$$

where, w is the objective weight and μ is the performance mean of ´Stopped in Advance´ of all machines.

The fitness of setup time is included as it is important to reduce the total setup time because it will reduce the time a worker needs to work at the machine.

$$f_{st} = w\left(\frac{\mu_{st}}{\mu_{st} + \mu_w}\right) \tag{8}$$

where, w is the objective weight, μ_{st} is the mean value of the total setup time of all machines and μ_w is the mean value of the total processing time of all machines.

5 OPTIMIZATION RESULTS

In this section the optimization results of two scheduling scenarios using different representations are presented. The first scenario represents one production week starting at a Monday, were the production line is stable and no machine break downs or critical deadlines need to be considered.

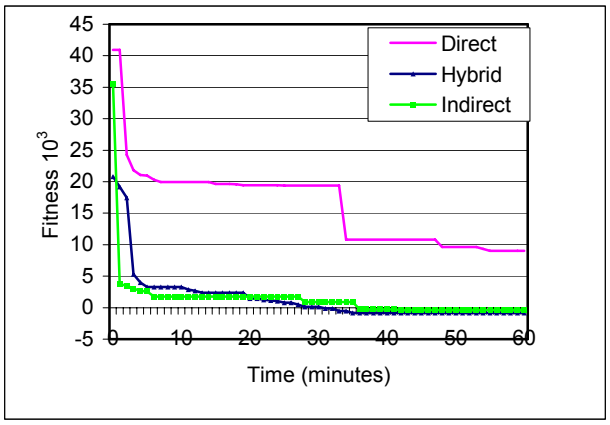

Figure 6: Optimization results of scenario one

Three graphs can be seen in the figure above: direct approach, indirect approach and hybrid approach. All three types eventually find good solutions in this scenario within one hour. The user does not need to wait an entire hour if the result is acceptable before that time. In the scenario above, both the hybrid and indirect approach generate good schedules within 20 minutes, but the direct approach takes longer time and does not find as good solutions, as the other approaches. The second scenario represents one production week starting on a Monday, when the production line has no machine break downs, but critical deadlines need to be considered for some products.

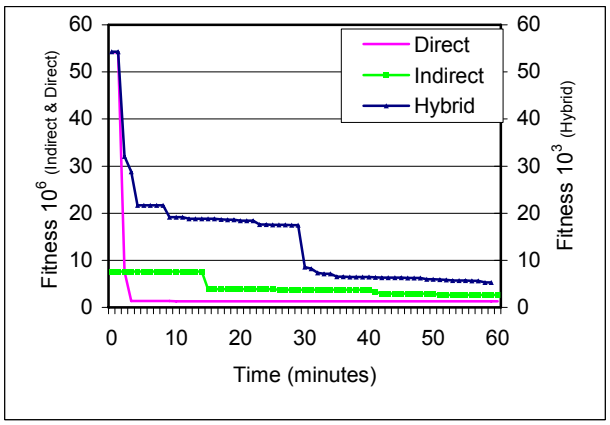

Figure 7: Optimization results of scenario two

Three graphs can be seen in the figure above: direct approach, indirect approach and hybrid approach. The direct and indirect approach, fitness values on the left y-axis, fails to find really good solutions in this scenario within one hour. Both the direct and indirect approach also fails to find solutions that avoids deficit of some of the product variants that had critical deadlines, resulting in bad fitness. However, the hybrid approach is successful in avoiding a deficit of the product variants that had critical deadlines, thereby generates good solutions.

5.1 Direct Approach

In the direct approach all machines follow machine schedules generated by the optimization. Each machine needs to follow the exact sequence that the optimization generated. In the first scenario the direct approach finds good solutions even though it takes a quite long time compared to the hybrid approach. In the second scenario, the direct approach fails in finding good solutions.

5.2 Indirect Approach

In the indirect approach the machines uses dispatching rules that are determined by the optimization. When a machine is ready to produce something, the dispatching rule at the machine determines which product to produce next. The dispatching rules used in the optimization are:

- First Come First Served (FCFS),
- Shortest Processing Time (SPT),
- Longest Processing Time (LPT),
- Earliest Due Date (EDD),
- Same Variant, lowest Setup time, earliest Due date (VSD)

The VSD dispatching rule is user defined rule that first tries to choose a product variant as the previous variant type, discard the others, but if that's is not possible the product variant with the lowest set-up time will be chosen. Secondary it's sorted on EDD. In the first optimization scenario the indirect approach is the best one of the three approaches. However, in the second scenario the indirect approach fails to find an acceptable solution.

5.3 Hybrid Approach

In the hybrid approach the first two thirds of the machines follow machine schedules generated by the optimization and the rest use dispatching rules as mentioned above. In the first scenario the hybrid approach is successful in finding good solutions. Even though the PDR is faster in the first scenario, the hybrid approach still finds good solutions fast. In the second scenario, with some product variants having critical deadlines, the hybrid is still successful in finding good solutions.

6 DISCUSSIONS

The problem with the direct approach is usually that it is very time consuming in finding good solutions, as the search space is huge. In order to find good solutions fast the representation of each machine sequence is generated based on sequence order and batch allocation over machines. Accordingly, each machine sequence is not changed directly by the optimization, and thereby some solutions are impossible to find. However, compared to the indirect approach the search space is still huge. The strength of the direct approach is absolute control and lets the optimization act in a more global sense, compared to a dispatching rule that usually acts more locally at a machine. In difficult scenarios where there are long sequence dependent setup times, many constraints, and critical products that need to be produced at once, the direct approach is believed to be suitable because of its broad search space. However, from the optimization results of scenario two, the direct approach has failed to find good solutions, which might depend on an "indirect" representation of machine sequence. This might be able to be solved by representing each machine sequence by itself or by reprioritizing line sequence order at different points in the line, but this would lengthen the optimization process.

In the indirect approach, when using PDRs, a reasonably good solution can be found very fast as the search space is limited to a number of existing rules. A good thing with dispatching rules is that in some cases it is possible to find a set of rules that can be used for all scenarios, based on knowing that these set of rules statistically generates good schedules. When it comes to complex real-world scheduling problems it might be difficult in finding a set of "static" rules to generate good schedules, because the scenarios might be really different from day to day. Furthermore, most of the basic dispatching rules, such as SPT, EDD, might not be adequate to reflect the necessary decisions. Therefore specialized dispatching rules for the scheduling are usually needed. Even though some rules are specialized, it might be a problem to create a rule that considers, not only in local and global sense, but also avoids different deadlocks, such as cart limitations. Various optimization scenarios have shown that the user-defined dispatching rule, VSD, seems to be the most frequent chosen dispatching rule by the optimization. Even though the indirect approach doesn't find any good solutions in scenario two, it is promising because it can generate good schedules relatively fast due to the limited search space. To add more well known dispatching rules and to develop more specialized dispatching rules might be a future option.

The hybrid approach combines the strengths and weaknesses of both approaches. In the hybrid approach the break point between direct and indirect approach has been chosen at a point after an operation group of multiple parallel machines, followed by several shared machines for the different product variants at the end of the line. The reason for choosing this point is that many of the machines upstream is shared between different product variant groups, some with long sequence dependent setup times, and after this point it is difficult to obtain really good schedules only using the direct approach, without any further changes to the algorithm. Once a machine group has been chosen for using PDRs, the succeeding operations also need to use PDRs in order to avoid deadlocks, such as cart limitations and capacity constraints in buffers. The result for the hybrid approach is very promising in this scenario as it outperforms both the direct and the indirect approach. In the beginning of the line, when machines use the direct approach, the optimization will lay out a schedule that avoids long set-up times and still prioritize batches with critical deadlines. At the end of the line the product variants will not have the same order as the line sequence order because some product variants have a longer lead time through the system and because of different batch allocation, and therefore the dispatching rules helps to utilize the machines more effectively at the end of the line.

7 CONCLUSIONS

A dispatching rule based approach can be a useful scheduling approach as it is time-efficient, but when there are highly complex scheduling problems or scenarios a direct approach may be advantageous when searching for good solutions time-effectively. However, a hybrid representation of these two approaches may offer their combined strengths - finding good solutions quickly for complex scheduling problems. In the first scenario carried out in this study all three approaches can find quite good solutions within an acceptable time. In the second scenario only the hybrid approach is able to find good solutions. In a complex real-life situation, as each production day is different from the other, more scenarios need to be tested so that reliable solutions that can solve both easy and difficult scenarios within an acceptable time have to be sought. This will be the future work of this research.

ACKNOWLEDGMENTS

This work is partially financed by the Knowledge Foundation (KK Stiftelsen), Sweden. The authors gratefully acknowledge them for the provision of research funding and also the industrial partner companies including Volvo Cars Engines, Skövde, for their support in this study.

REFERENCES

Andersson, M., A. Persson, H. Grimm, and A. Ng. 2007. A Web-based simulation optimization system for industrial scheduling. In *Proceedings of the 2007 Winter Simulation Conference*, 1844–1852. Washington D.C., USA.

Candido, M. A. B., S. K. Khator, and R. M. Barcia. 1998. A genetic algorithm based procedure for more realistic job shop scheduling problems. *International Journal of Production Research* 36 (12):3437–3457.

Holtaus, O. 1997. Design of efficient job shop scheduling rules. *Computers and Industrial Engineering* 33 (1–2): 2492–252.

Kim, I., J. Watada, and T. Shigaki. 2007. A comparison of dispatching rules and genetic algorithms for job shop schedules of standard hydraulic cylinders. *Soft Computing* 12:121–128.

Kiran, A. S. 1998. Simulation and scheduling. In *Handbook of Simulation*, ed. J. Banks, 677–717, New York: John Wiley and Sons, Inc.

Koh, K-H., R. Souza, and N-C. Ho. 1996. Database driven simulation/simulation-based scheduling of a job-shop, *Simulation Practice and Theory* 4:31–45. Elsevier.

Ng, A., H. Grimm, T. Lezama, A. Persson, M. Andersson, and M. Jägstam. 2008. OPTIMISE: An Internet-Based Platform for Metamodel-Assisted Simulation Optimization. *Recent Advances in Communication Systems and Electrical Engineering*, ed. X. Huang, Y-S. Chen and S-L. Ao, 281–296. Springer.

Panwalkar, S. and I. Wafik. 1977. A survey of scheduling rule. *Operations Research* 25 (1):45–61.

Sanchez, S. M. 2000. Robust design: Seeking the best of all possible worlds. In *Proceedings of the 2000 Winter Simulation Conference*, ed. J. A. Joines, R. R. Barton, K. Kang, and P. A. Fishwick, 69–76.

Sankar, S. S., S. G. Ponnanbalam, and C. Rajendram. 2003. A multiobjective genetic algorithm for scheduling a flexible manufacturing system. *International Journal in Advanced Manufacturing Technologies* 22:229–236.

Tanev, I. T., T. Uozumi, and Y. Morotome. 2004. Hybrid evolutionary algorithm-based real-world flexible job shop scheduling problem: application service provider approach. *Applied Soft Computing* 5:87–100.

Tay, J. C. and N. B. Ho. 2008. Evolving dispatching rules using genetic programming for solving multi-objective flexible job-shop problems. *Computers and Industrial Engineering* 54:453–473.

AUTHOR BIOGRAPHIES

MARCUS ANDERSSON s a Ph.D. candidate at University of Skövde, Sweden. He received his BSc and MSc degrees in Manufacturing Engineering from University of Skövde and Loughborough University, UK, respectively. His research is within simulation-based optimization, scheduling and production system analysis. His e-mail address is <marcus.andersson@his.se>.

AMOS H. C. NG is a Senior Lecturer at the University of Skövde, Sweden. He holds a B.Eng. degree and a M.Phil. degree, both in Manufacturing Engineering from the City University of Hong Kong and a Ph.D. degree in Computing Sciences and Engineering from De Montfort University, Leicester, U.K. He is a member of the IET and a Chartered Engineer in the U.K. His research interests include simulation-based optimization and production systems analysis. His e-mail address is <amos.ng@his.se>.

HENRIK GRIMM is a Systems Developer at the University of Skövde, Sweden. He received his BSc and MSc degrees in Computer Science from University of Skövde. His research interests include computer simulation, artificial intelligence, and distributed systems. His e-mail address is <henrik.grimm@his.se>.