Mathematical optimization of a flexible job shop problem including preventive maintenance and availability of fixtures

Karin Thörnblad (karin.thornblad@volvo.com)
Chalmers University of Technology and Volvo Aero Corporation

Torgny Almgren
Chalmers University of Technology and Volvo Aero Corporation

Michael Patriksson
Chalmers University of Technology

Ann-Brith Strömberg
Chalmers University of Technology

Abstract

The *multitask cell* at Volvo Aero Corporation is a flexible job shop containing ten resources aimed at being flexible with regard to product mix and processing types. Computing schedules for this type of job shop is an NP-hard problem. The computation times will therefore always be an issue, especially as the scheduling problem includes limited availability of fixtures and preventive maintenance planning. Computational results show that with the method developed we are able to produce optimal, or near-optimal, schedules for real data instances within an acceptable time frame. The scheduling principle proposed shortens lead times and minimizes tardiness.

Keywords: Production planning, Flexible job shop scheduling problem, Mathematical optimization

Introduction

The *multitask cell* at Volvo Aero Corporation is a flexible job shop containing ten resources aimed at being flexible with regard to both product mix and processing types. The production cell is intended to carry out a large variety of jobs, five of its resources being multi-purpose machines that are able to process three types of operations: turning, milling, and drilling. The production cell was built to increase the degree of machine utilization and to reduce product lead times, compared with the ordinary job shops at the production site. The capital tied up in the investment in the multitask cell is substantial; high machine utilization is therefore crucial. This is a demanding requirement as the scheduling of the cell is a highly complex combinatorial problem, recognized as the *flexible job shop scheduling problem (FJSP)* in the literature of operations research (Baykasoglu and Özbakir, 2010).

The purpose of the work presented in this paper is to contribute to the goal of enabling the creation of optimal, or near-optimal, schedules for multi-purpose production cells similar to the one at Volvo Aero. The scheduling procedure must be fast and produce reliable and robust schedules, since the conditions are unceasingly changing with new jobs continuously arriving at the queue. The optimization objective is to minimize the mean throughput time and the total tardiness of the jobs. With a certain periodicity, a number of preventive maintenance activities need to be carried out in some of the resources of the cell, and should hence also be scheduled simultaneously, and optimally, with the production tasks.

Related work

A few decades ago it was not possible to employ exact mathematical optimization methods to instances of sizes relevant for real applications of the flexible job shop problem, since the computation times required for the solution process were too long. Therefore, a lot of research was concentrated on obtaining approximate solutions to job shop problems through the application of heuristic methods; see (Jain and Meeran, 1999) for a historical overview. However, during the past decades the development of theory and practice of mathematical optimization modeling and methods, together with the development of computer hardware, have decreased computation times by several orders of magnitude. Computing job shop schedules is an NP-hard problem; see (Brucker et al., 1997), where it is shown that the *multi-purpose machine (MPM) job shop problem* (another name for the FJSP) is NP-hard for problems with more than three jobs and two machines. This means that computation times will always be an issue for complex problems such as that of scheduling the multitask cell including preventive maintenance planning and a limited availability of fixtures.

The methods proposed for obtaining feasible solutions to job shop problems are still dominated by various meta-heuristics, see e.g. (Bülbül, 2011), and (Beck et al., 2011) who propose a hybrid between the shifting bottleneck heuristic and tabu search, and a combination of constraint programming and local search, respectively. Some heuristic methods for tackling the FJSP are found in (De Giovanni and Pezzella, 2010), (Mati et al., 2011) and (Hmida et al., 2010). The only article dealing with a FJSP with fixture constraints we have come across is (Rahimifard and Newman, 1997), who describe a simulation based scheduling approach. Articles considering FJSP including the scheduling of maintenance activities are for example (Wang and Yu, 2010) and (Golmakani and Namazi, 2012) who propose a so called filtered beam search algorithm and an artificial immune algorithm, respectively. In the latter a nonlinear mathematical model is presented, which is solved for small instances only (6 jobs and 6 machines).

There are few *mixed integer linear programming (MILP)* models proposed for the flexible job shop problem in the literature; examples are (Fattahi et al., 2007), (Özgüven et al., 2010) and (Mati and Xie, 2011). All models presented in these references are based on variables commonly used for job shop problems in text books in the field of operations research, i.e. variables similar to the ones first employed in (Manne, 1960); see, e.g., p. 365 in (Taha, 2007). In previous work (Thörnblad, 2011), we have presented three MILP models of a sub-problem to the problem of scheduling the multitask cell, namely the problem of scheduling the jobs on the five multi-purpose machines. One of the models was based on Manne variables and this model was by far outperformed by our newly developed time-indexed model with respect to both computation times and the sizes of instances that can be solved using standard optimization software. In this paper we present a time-indexed model of the problem of scheduling the whole multitask cell including fixture availability constraints.

In the next section, the problem is described in detail and the notation needed for presenting the mathematical model is given. This section is followed by the presentation of the mathematical model. The article ends with a section of computational results based on real data, and conclusions.

Problem description

In addition to the ten resources in the multitask cell, denoted $\mathcal{K}$, there is an input/output conveyor, being the entrance for the parts arriving at the cell, a stocker crane for transporting the parts inside the cell, and a central tool storage to furnish the processing machines with the appropriate tools. Each *job* in the set $\mathcal{J}$ of jobs to be scheduled consists of a set of $\mathcal{N}_j = \{1, \ldots, n_j\}$ so-called *route operations*, that are to be processed in a specific order. Since the multipurpose machines are similar but not identical, some route operations are allowed to be processed only by a subset of the resources. Hence, there are several possible routes for the completion of one job; see an example in Fig. 1. The processing time of route operation i of job j is denoted by p_{ij}.

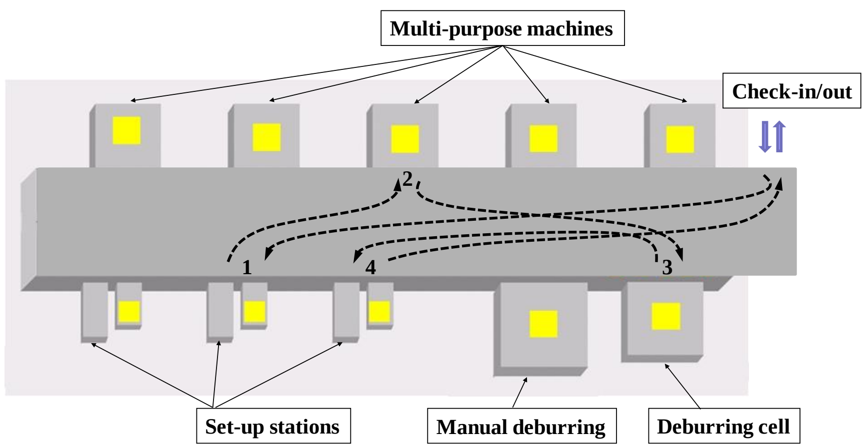

Figure 1 –A schematic overview of the multitask cell. A possible route for a part is indicated with dashed lines.

These are areas inside the cell dedicated to the storage of parts before and between processing. These storage areas have never been used to their full capacity since production started in the multitask cell, and are therefore assumed to be sufficiently large. The parts that are ready to be processed in the cell are those that are checked-in at the input conveyor but not yet put into a fixture at a set-up station; the release dates, r_j, of the corresponding jobs are set to 0. The release dates of the remaining jobs to be scheduled are set to the corresponding part's estimated time of arrival at the cell. All jobs have a due date, d_j, which is the time when they are due to be completed.

Each part to be processed in the multitask cell is mounted into a fixture in one of the cell's three setup stations. The fixtures are specially designed and manufactured for each type of processing operation to be performed in one of the multi-purpose machines. The number of fixtures of each type is limited, and is thus a constraint to be considered when solving the scheduling problem.

The planning horizon of the schedule is divided into $T + 1$ intervals, each of length ℓ hours. Since the resources are often occupied by the processing of previous jobs at time 0, we use the parameter a_k to denote the first time a resource k is available. The value of the parameter T has to be large enough such that the time horizon $[0,(T+1)\ell]$ contains an optimal schedule. A small value of T is, however desirable, since this means that the computation times become shorter. This is due to the fact that the number of variables and constraints in the time-indexed formulation is a multiple of the number of intervals. We determine a suitable value of T using a heuristic; see (Thörnblad, 2011).

The problem is thus to construct a feasible schedule for all route operations of all jobs considered within the planning horizon which minimizes (or maximizes) the objective function. The minimization of tardiness is considered the main objective, but since this model should work for all possible scenarios, there is a high probability that there will be some scenarios with no tardy jobs, and in order to produce good optimal schedules also for these scenarios, the minimization of flow time is a secondary objective. The objective function is thus to minimize

$$\sum_{j\in\mathcal{J}}(a_j C_j + b_j T_j - \varepsilon^{\mathrm{s}} t_{1j}), \tag{1}$$

where C_j and T_j are the completion time and tardiness of job j, respectively, and t_{1j} is the starting time of job j. It is important that the objective weight ε^{s} is much smaller than the other weights a_j and b_j, since this term strives to schedule the jobs as late as possible. It is included in the objective function in order to reduce the time each fixture is occupied. This objective function contributes to the generation of schedules that will satisfy the goals of low tardiness, short product lead times as well as high machine utilization.

In the next section we will present our time-indexed formulation of the problem of scheduling the multitask cell.

The mathematical model

The time-indexed model formulated in this section is expressed solely in the variables x_{ijku}, which are valued 1 if operation i of job j is scheduled to start processing in resource k in the beginning of time interval u, 0 otherwise. Since the starting time of an operation can be expressed in terms of these variables by

$$t_{ij} = \sum_{k\in\mathcal{K}}\sum_{u\in\mathcal{T}} u x_{ijku},$$

the objective function (1) given in the previous section can be rewritten as

$$\sum_{j\in\mathcal{J}}\sum_{k\in\mathcal{K}}\sum_{u\in\mathcal{T}}((a_j(u+p_{n_j j}) + b_j(u+p_{n_j j}-d_j)_+)x_{n_j jku} - \varepsilon^{\mathrm{s}} u x_{1jku}). \tag{2}$$

Throughout the article we define $(z)_+ := \max\{z;0\}$. The objective function (2) is linear since the max expression is used solely on parameters and not on the variables.

First we will present and explain the constraints for the so-called *base model*, before we move on to introduce some additional constraints. The base model is to minimize the function (2) with respect to the variables x_{ijku} subject to

$$\sum_{k\in\mathcal{K}}\sum_{u\in\mathcal{T}} x_{ijku} = 1,\ i\in\mathcal{N}_j,\ \ j\in\mathcal{J}, \tag{3}$$

$$\sum_{u\in\mathcal{T}} x_{ijku} \le \lambda_{ijk},\ i\in\mathcal{N}_j,\ \ j\in\mathcal{J},\ k\in\mathcal{K}, \tag{4}$$

$$\sum_{i\in\mathcal{N}_j}\sum_{j\in\mathcal{J}}\sum_{\mu=(u-p_{ij}+1)_+}^{u} x_{ijk\mu} \le 1,\ k\in\mathcal{K}\setminus\{k_s\},\ u\in\mathcal{T}, \tag{5}$$

$$\sum_{i\in\mathcal{N}_j}\sum_{j\in\mathcal{J}}\sum_{\mu=(u-p_{ij}+1)_+}^{u} x_{ijk_s\mu} \le 3,\ u\in\mathcal{T}, \tag{6}$$

$$\sum_{k\in\mathcal{K}}\left(\sum_{\mu=0}^{u} x_{ijk\mu} - \sum_{\nu=0}^{u+p_{ij}} x_{i+1,jk\nu}\right) \ge 0,\ i=1,\ldots,n_j-1,\ \ j\in\mathcal{J},\ u=0,\ldots,T-p_{ij}, \tag{7}$$

$$x_{1jku} = 0,\ \ j\in\mathcal{J},\ k\in\mathcal{K},\ u=0,\ldots,\max\{r_j;a_k\}, \tag{8}$$

$$x_{ijku} = 0,\ j\in\mathcal{J},\ k\in\mathcal{K},\ u=T-\delta_{ij},\ldots,T. \tag{9}$$

The constraints (3) ensure that each operation *i* of job *j* is scheduled to be processed exactly once. The constraints (4) regulate that all operations are processed in an allowed resource *k*, since the parameter λ_{ijk} is valued 1 if operation *i* of job *j* can be processed in resource *k*, and 0 otherwise. The ten processing resources are treated as eight when solving the problem, since the three set-up stations are identical and therefore treated as one resource, k_s, with the capacity of processing three operations simultaneously, in order to avoid problems with symmetry. The constraints (5) and (6) ensure that one operation at a time is scheduled in each resource *k*, since the constraints consist of sums of all the decision variables for a specific resource *k* over all time periods of the same length as the processing time p_{ij} of an operation. The constraints (7) make sure that no operation *i* can start being processed before the previous operation for the same job has been completed. These are the so-called precedence constraints for the operations within a job *j*. The constraints (8) ensure that the first operation of a job is scheduled after the release date of the job in an available resource. Finally, the constraints (9) make sure that an operation is not scheduled to start at a time that would implicate that the operation in question or a succeeding operation would not be completed at the end of time interval *T*.

Some of the jobs considered are to be processed on the same physical part, and are hence subject to another type of precedence relationships than those described in (7). These jobs have to be separated by a time lag, v_{jq}, which is the planned lead time between the completion of job *j* and the start of job *q*; this time period might solely include the internal transportation time within the multitask cell, but it might also be a longer time including several operations that are to be performed on the part in other workshops of the factory. These precedence constraints may be formulated as $t_{1q} \ge C_j + v_{jq}$,where job *j* that has to precede job *q* by a time lag v_{jq}, but their equivalent formulation (10) given below typically yields better LP bounds, enabling a faster resolution of the problem:

$$\sum_{k\in\mathcal{K}}\left(\sum_{\mu=0}^{u} x_{n_j jk\mu} - \sum_{\nu=0}^{u+v_{jq}} x_{1qk\nu}\right) \geq 0,\ (j,q)\in\mathcal{Q},\ u=0,\ldots,T-v_{jq}. \tag{10}$$

In (10) the set $\mathcal{Q}$ consists of all pairs of adjacent jobs in the routing that are to be performed on the same physical part.

Each job occupies a fixture during the whole visit in the multitask cell. Each fixture type $f\in\mathcal{F}$ is specially designed and can only be used for a subset $\mathcal{S}_f$ of jobs. Since they are very expensive, only α_f fixtures of each type are available. The capacity constraints on the number of fixtures occupied at each time interval are formulated as

$$\sum_{j\in\mathcal{S}_f}\sum_{k\in\mathcal{K}}\left(\sum_{\mu=0}^{u} x_{1jk\mu} - \sum_{\nu=0}^{u-p_{n_j j}} x_{n_j jk\nu}\right) \leq \alpha_f,\ f\in\mathcal{F},\ u\in\mathcal{T}, \tag{11}$$

where the sum over the resources of the expression within parentheses is 1 if job *j* is in process during time interval *u*, and 0 otherwise.

The scheduling of preventive maintenance activities

There are three types of preventive maintenance activities performed regularly: cleaning of the tool magazines, machine verification, and probe verification. A tool magazine is cleaned and a probe verification is carried out in each multipurpose machine on a regular basis during a time window of length Δ^{M}. Regarding the machine verification, there is a rolling schedule over two weeks where two of the multi-purpose machines are maintained during the first week, and three during the second week.

In order to schedule a maintenance activity $m\in\mathcal{M}^{\mathrm{M}}$ of duration d_{mk} in resource *k*, we define the variable χ_{mku}, which equals 1 if activity *m* is scheduled to start in resource *k* at the beginning of time interval *u*, and 0 otherwise. The constraints that β_{mk} activities (β_{mk} is integer-valued) is scheduled during a time window that starts at the beginning of time interval τ_{mk} are formulated as

$$\sum_{\mu=(\tau_{mk})_+}^{\min\{\tau_{mk}+\Delta^{\mathrm{M}}-1;T+1-d_{mk}\}} \chi_{mk\mu} = \beta_{mk},\ m\in\mathcal{M}^{\mathrm{M}},\ k\in\mathcal{K}\setminus\{k_s\}, \tag{12}$$

where the equality sign can be replaced by a $\geq$ (which will then be referred to as ($12\geq$)), if we add a positive term with the sum of the maintenance decision variables with a small weight to the objective function:

$$\sum_{j\in\mathcal{J}}\sum_{k\in\mathcal{K}}\sum_{u\in\mathcal{T}}((a_j(u+p_{n_j j})+b_j(u+p_{n_j j}-d_j)_+)x_{n_j jku} - \varepsilon^{\mathrm{S}} u x_{1jku}) + \varepsilon^{\mathrm{M}}\sum_{m\in\mathcal{M}}\sum_{k\in\mathcal{K}\setminus\{k_s\}}\sum_{u\in\mathcal{T}}\chi_{mku}. \tag{13}$$

The capacity constraints (5) of the base model also need to be altered when adding the scheduling of maintenance activities, since no job can be scheduled in the resource during maintenance. Preemption is not allowed in any operation or maintenance activity, i.e. the operations and maintenance activities must be completed before the resource can process the next task. Hence, the constraints (5) need to be reformulated as

$$\sum_{i\in\mathcal{N}_j}\sum_{j\in\mathcal{J}}\sum_{\mu=(u-p_{ij}+1)_+}^{u} x_{ijk\mu}+\sum_{m\in\mathcal{M}}\sum_{\nu=(u-d_{mk}+1)_+}^{u}\chi_{mk\nu}\leq 1,\ k\in\mathcal{K}\setminus\{k_s\},\ u\in\mathcal{T}. \tag{14}$$

Instead of using time windows that begin at a certain time τ_{mk} as in the constraints (12), another version of these constraints can be formulated where the time window is replaced by a rolling horizon, so that at any time two consecutive maintenance activities will never be separated by more than $\Delta_{\mathrm{R}}\ell$ hours. These constraints are well suited for situations when the maintenance of the resource is driven by restricted life lengths of crucial parts of the resource:

$$\sum_{\mu=u}^{\min\{u+\Delta^{\mathrm{R}}-1;T+1-d_{mk}\}}\chi_{mk\mu}\geq\beta_{mk},\ m\in\mathcal{M}^{\mathrm{R}},\ k\in\mathcal{K}\setminus\{k_s\},\ u\in a_k,\ldots,\max\{a_k;T+1-\Delta^{\mathrm{R}}\}. \tag{15}$$

Differentiated objective weights

As pointed out in the problem description section, the scheduling algorithm should work for all possible scenarios, and among these are scenarios where some parts are late already when they arrive at the multitask cell, i.e., their respective jobs have negative due dates. This means that the job will have a positive tardiness, $T_j=C_j+|d_j|$, since the definition of tardiness is $T_j=(C_j-d_j)_+$. Hence, if all jobs have negative due dates, then the objective to minimize the total tardiness will yield the same solutions as the objective to minimize the total flow time (+ the constant term $\sum_{j\in\mathcal{J}}|d_j|$).

Since the solutions to the optimization model do not depend on the value of the negative due dates, the same solutions will be found if all negative due dates are set to 0. However, in the schedule one might wish that the jobs that are the most late are prioritized before not so late jobs. One way to accomplish this is to determine the tardiness objective weights, b_j, for a job j so that a late job gets a higher weight than a job that is less late. Let B be the weight for the jobs with due date of 0. Then, the weight b_j for a job j is defined as

$$b_j=B-B\,d_j/|d_q|, \tag{16}$$

where $|d_q|$ is the due date of the job with the largest absolute due date that is not an outlier. In the computational tests below, we defined an outlier to be a job p for which 20% of the job's absolute due date exceed the median of $|d_j|$, i.e., a job p such that $0.2|d_p|\geq\tilde{d}$, $\tilde{d}$ being the median of $|d_j|$. Instead of this simple definition we intend to use the definition that an outlier "is a point which falls more than 1.5 times the inter-quartile range above the third quartile or below the first quartile" (Renze, 2012) in future research.

Computational results

Real data, from the site's Enterprise Resource Planning (ERP) system, have been used to test and validate the model. Nine scenarios were collected from the multitask cell during a period of three months in the spring of 2012, and from these instances were created with $15\leq n\leq 35$, where n is the number of jobs. In Table 1, all the models used in the computational tests are defined. The computations were carried out using AMPL-CPLEX12 on a computer with two 2.66GHz Intel Xeon 5650, each with six cores (24 threads), and its total memory was 48 Gbyte RAM.

Table 1 – Definition and notation of the model tested.

Notation	Description	Model
P	Base model with precedence constraints	Min (2) s.t.(3)-(10)
PF	Base model + fixture constraints	Min (2) s.t. (3)-(11)
PFMeq	Base model + fixture constraints + maintenance constraints with time window	Min (13) s.t. (3), (4), (6)-(11), (12), (14)
PFMgeq	Base model + fixture constraints + maintenance constraints with time window with $\geq$	Min (13) s.t. (3), (4), (6)-(11), (12$\geq$), (14)
PFR	Base model + fixture constraints + maintenance constraints with "rolling time window"	Min (13) s.t. (3), (4), (6)-(11), (14), (15)
PFMgeqW	As PFMgeq, but with differentiated objective tardiness weights	Min (13) s.t. (3), (4), (6)-(11), (12$\geq$), (14)

The model PFMeq was not tested for all instances, since the time required to solve this model to optimum was longer than that required to solve PFMgeq to optimum. When comparing the two models for an instance with 30 jobs, PFMeq quickly found a very good solution, although it took longer to reach an optimum; see Fig. 2. Both models find a solution with objective values within 3% from the optimal objective value within the first 5 minutes of computation time.

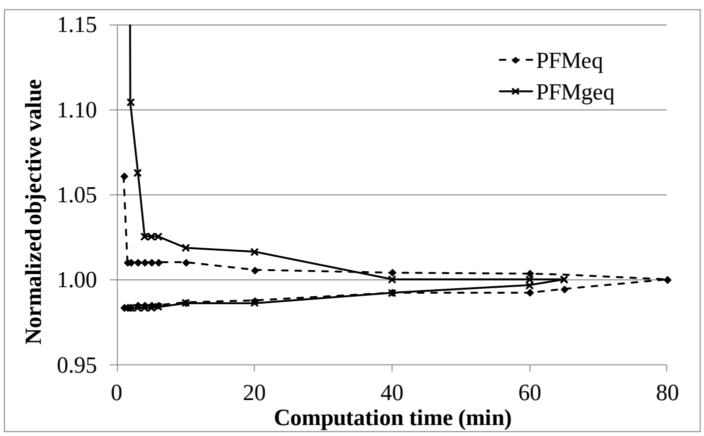

Figure 2 – Lower and upper bound of the objective function for one instance with 30 jobs. The upper bound is the objective value of the best found feasible solution at this time point of the algorithm. The values are normalized with respect to the optimal objective value.

In Fig. 3 the computation time required to solve the models P, PF, PFMgeq, PFR, and PFMgeqW to optimality is plotted for all instances. A time limit of 2 h was set for the computation time (clocktime). The base model P was solved in just a few seconds for some instances but had trouble solving some other instances: it is the only model that reached the time limit of 7200 s for one instance of 20 jobs and two instances of 25 jobs. The time to compute the instances vary more between the different instances than between the models, hence the added fixture and maintenance constraints do not seem to complicate the solution algorithm too much, at least not for this sample of real instances.

The objective weights used for all models but PFMgeqW were a_j =1, b_j=10, ε^{s} = 0.001, and ε^{M} =0.0001, and the differentiated tardiness weights were calculated using the formula in (16), with B=10.

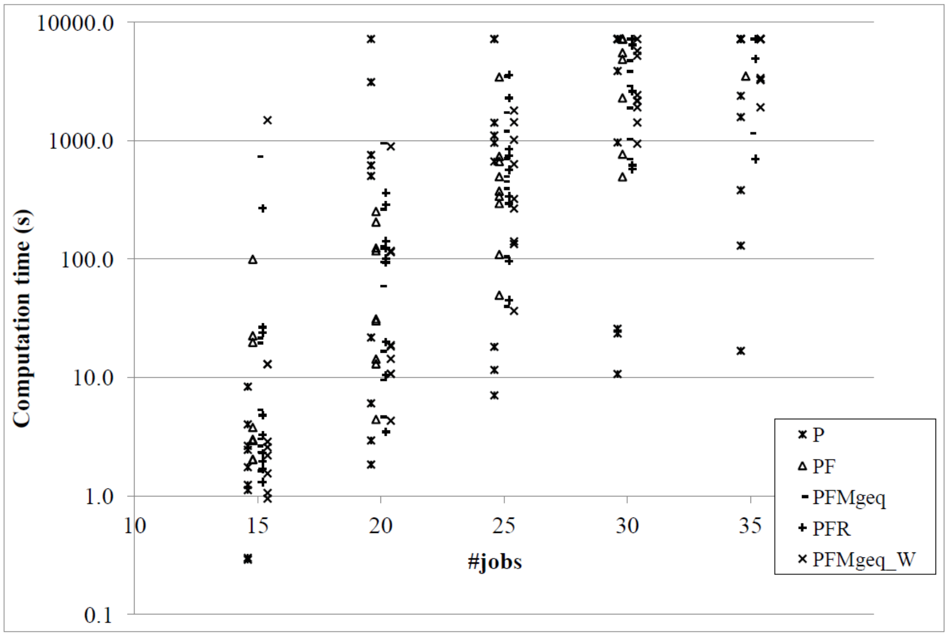

Figure 3 – The computation times (clock time) required to solve the models P, PF, PFMgeq, PFR and PFMgeqW to optimality versus different instance sizes. The markers at 7200s indicate the instances when the time limit was reached before optimality was proven.

The instance sizes needed to schedule the coming shift for the test scenarios varies between $21 \le n \le 33$, i.e., all jobs with release dates $r_j \le 8$h. In Table 2, the computation times required for the model PFMeqW to find a solution with a relative optimality gap of at most 1 and 5%, respectively, are listed. The average gap is also listed, since the algorithm often jumps to a solution much better than the limit set.

Table 2 – Computation times in seconds required for the model PFMeqW to find a solution with a relative optimality gap of 1 and 5%, respectively. One of the instances with 35 jobs was terminated due to the time limit before the gap had decreased below 1%.

	Gap 5%				Gap 1%			
#jobs (n)	**min**	**mean**	**max**	**Avg. gap**	**min**	**mean**	**Max**	**Avg. gap**
15	0.8	4.9	23.9	0.28%	0.8	5.0	23.8	0.14%
20	4.0	29.3	149.1	1.40%	4.0	49.0	268.3	0.25%
25	22.1	37.1	71.9	1.76%	22.1	128.6	529.1	0.69%
30	70.9	192.5	737.9	2.37%	73.8	853.0	2773.7	0.80%
35	67.8	477.3	1293.9	2.53%	144.0	–	7200.0	1.04%

The mean computation time for finding a schedule with 35 jobs is about 8 min (477.3 s), and the longest time required is a bit more than 20 minutes. We consider this a reasonable computation time to find a schedule for the coming shift, provided one can accept that the objective value of the solution found is at most 5% from the optimal objective value. The algorithm found an optimal solution in 12 (19) out of the 45 test runs when the termination criteria was an optimality gap of 5% (1%).

Conclusions

We have shown that by employing the time-indexed model presented in this paper, we are able to produce optimal schedules for a real application within a reasonable amount of time, a task which is impossible when employing a model formulated with variables of the kind most frequently used in the literature for job shop problems. Employing the proposed scheduling algorithm will shorten lead times and minimize tardiness, and provide a more efficient use of the resources of the multitask cell than that of today. To our knowledge, this is the first time-indexed model presented for a flexible job shop problem including preventive maintenance and fixture availability.

References

Baykasoglu, A. and Özbakir, L. (2010), "Analyzing the effect of dispatching rules in the scheduling performance through grammar based flexible scheduling system", *International Journal of Production Economics*, Vol. 124, pp. 269-381.

Beck, J.C., Feng, T.K., and Watson, J.-P. (2011) "Combining constraint programming and local search for job-shop scheduling", *INFORMS Journal on Computing*, Vol. 23, No. 1, pp. 1-14.

Brucker, P., Jurisch, B., and Krämer, A. (1997), "Complexity of scheduling problems with multi-purpose machines", *Annals of operations research*, Vol. 70, pp. 57-73.

Bülbül, K. (2011), "A hybrid shifting bottleneck-tabu search heuristic for the job shop total weighted tardiness problem", *Computers & Operations Research*, Vol. 38, No. 6, pp. 967-983.

De Giovanni, L. and Pezzella, F. (2010), "An improved genetic algorithm for the distributed and flexible job-shop scheduling problem", *European Journal of Operational Research*, Vol. 200, No. 2, pp. 395-408.

Fattahi, P., Mehrabad, M. S., and Jolai, F. (2007), "Mathematical modeling and heuristic approaches to flexible job shop scheduling problems", *Journal of Intelligent Manufacturing*, Vol. 18, pp. 331-342.

Golmakani, H. R. and Namazi, A. (2012), "Multiple-route job shop scheduling with fixed periodic and age-dependent preventive maintenance to minimize makespan", *Journal of Quality in Maintenance Engineering*, Vol. 18, No. 1, pp. 60-78.

Hmida, A. B., Haouari, M., Huguet, M.-J., and Lopez, P. (2010), "Discrepancy search for the flexible job shop scheduling problem", *Computers & Operations Research*, Vol. 37, No. 12, pp. 2192-2201.

Jain, A. and Meeran, S. (1999), "Deterministic job-shop scheduling: Past, present and future", *European Journal of Operational Research*, Vol. 113, pp. 390-434.

Manne, A. S. (1960), "On the job-shop scheduling problem", *Operations Research*, Vol. 8, pp. 219-223.

Mati, Y. and Xie, X. (2011), "Multiresource shop scheduling with resource flexibility and blocking", *IEEE transactions on automation science and engineering*, Vol. 8, No. 1, pp. 175-189.

Mati, Y., Lahlou, C., and Dauzère-Pérès, S. (2011), "Modelling and solving a practical flexible job-shop scheduling problem with blocking constraints", *International Journal of Production Research*, Vol. 49, No. 8, pp. 2169-2182.

Rahimifard, S. and Newman, S.T. (1997), "Simultaneous scheduling of workpieces, fixtures and cutting tools within flexible machining cells", *International Journal of Production Research*, Vol. 35, No. 9, pp. 2379-2396.

Renze, J. (2012) "Outlier", *From MathWorld--A Wolfram Web Resource*, created by Weisstein, E. W., http://mathworld.wolfram.com/Outlier.html (2012-05-15).

Taha, H.A. (2007) *Operations Research: An Introduction*, 8th ed., Pearson Prentice Hall, Upper Saddle River, NJ, USA.

Thörnblad, K. (2011), *On the Optimization of Schedules of a Multitask Production Cell*, Licentiate thesis, Department of Mathematical Sciences, Chalmers University of Technology and the University of Gothenburg, September 2011.

Özgüven, C., Özbakır, L., and Yavuz, Y. (2010), "Mathematical models for job-shop scheduling problems with routing and process plan flexibility", *Applied Mathematical Modelling*, Vol. 34, No. 6, pp. 1539-1548.