LOWER AND UPPER BOUNDS FOR THE JOB-SHOP SCHEDULING PROBLEM WITH SEQUENCE-DEPENDENT SETUP TIMES

Christian Artigues (Corresponding author)
Laboratoire d'Informatique, University of Avignon
CNRS FRE 2487, BP 1228, Avignon Cedex 9, France
christian.artigues@univ-avignon.fr

Fabrice Buscaylet
Athelys
215 rue Jean-Jacques Rousseau, 92 136 Issy-Les-Moulineaux Cedex, France
fabrice.buscaylet@artelys.com

Dominique Feillet
Laboratoire d'Informatique, University of Avignon
CNRS FRE 2487, BP 1228, Avignon Cedex 9, France
dominique.feillet@univ-avignon.fr

Abstract We present several methods to minimize the makespan in a job-shop problem with sequence-dependent setup times. We first present active, semi-active and non-delay schedule generation scheme, from which priority-rule based sampling heuristics are derived. Focusing on the disjunctive graph representation of the problem, we discuss the loss of dominance properties of efficient neighborhoods for the standard job-shop problem. We define an alternative interchange neighborhood and we propose a simple tabu search method. We use the traveling salesman problem with time windows relaxation of the problem to derive new lower bounds that we embed in a dichotomizing exact tree search method. All together, the proposed methods significantly improve the best known results on all the 10 previously unsolved instances of Brucker and Thiele, 1996.

Keywords: job-shop scheduling, sequence dependent setup times, priority-rule based sampling, tabu search, TSPTW relaxation, branch-and-bound

1. Problem definition

In the job-shop scheduling problem with sequence dependent setup times, denoted $J|s_{ij}|C_{\max}$, a set of n jobs $\mathcal{J} = \{J_i, \ldots, J_n\}$ and a set of m machines $\mathcal{M} = \{M_1, \ldots, M_m\}$ are considered. Each job $J_i \in \mathcal{J}$ is made of a chain of m operations $O_{i1}, \ldots, O_{im}$. Each operation O_{ij} of J_i requires a single machine $m_{ij} \in \mathcal{M}$, all operations of the same job requiring distinct machines. Each operation O_{ij} has a non zero duration p_{ij}. Each operation O_{ij} must precede operation $O_{i,j+1}$ for all $i = 1, \ldots, n$ and for all $j = 1, \ldots, m-1$. A sequence dependent setup time, denoted s_{ijk}, is defined for each couple of distinct jobs (J_i, J_j) and for each machine M_k. An initial setup time s_{0ik} is defined for each job J_i and for each machine M_k. It is assumed that the triangular inequality holds: for each machine M_k and for each triple of distinct jobs (J_i, J_j, J_x), we have $s_{ixk} \leq s_{ijk} + s_{jxk}$. For each machine M_k and for each couple of distinct jobs (J_i, J_j), we have $s_{0jk} \leq s_{0ik} + s_{ijk}$.

A non-negative starting time S_{ij} has to be assigned to each operation O_{ij} such that the makespan $C_{\max} = \max_{i=1,\ldots,n} S_{im} + p_{im}$ is minimized, subject to precedence constraints ($S_{ij} + p_{ij} \leq S_{i,j+1}$, for each $i = 1, \ldots, n$ and for each $j = 1, \ldots, m-1$) and machine constraints ($S_{ij} + p_{ij} + s_{ixk} \leq S_{xy}$ or $S_{xy} + p_{xy} + s_{xik} \leq S_{ij}$ for each couple of distinct operations (O_{ij}, O_{xy}) such that $m_{ij} = m_{xy} = M_k$).

The job-shop scheduling problem with sequence dependent setup times is a NP-hard problem, as an extension of the standard job-shop scheduling problem, denoted $J||C_{\max}$, where all setup times are zero.

2. Literature review

Problem $J|s_{ij}|C_{\max}$ received only little attention in the literature, albeit being a natural extension of the job-shop problem with many applications in manufacturing (see Allahverdi et al., 1999). Brucker and Thiele, 1996 proposed a branch-and-bound method for more general shop problems and generated a set of 15 $J|s_{ij}|C_{\max}$ instances, denoted by BT. Focacci et al., 2000 proposed a branch-and bound method in a constraint programming framework for another variant with alternative resources. Choi and Choi, 2002 proposed a local search method, also for an extension involving alternative operations. A simple tabu search heuristic has been tested on the BT instances by Artigues and Buscaylet, 2003. A branch-and-bound procedure is proposed by Artigues et al., 2004 for the $J|s_{ij}|C_{\max}$ which improve the results obtained by Brucker and Thiele, 1996 and Focacci et al., 2000 on the BT instances thanks to the upper bounds obtained by a priority rule-based multipass heuristic proposed in Artigues et al., 2005. In the present paper we make a synthesis of the methods described in Artigues et al., 2004, Artigues and Buscaylet, 2003 and Artigues et al., 2005. Furthermore, we obtain further significant improvements of our results on the BT instances by integrating tabu search and multi-pass sampling heuristics.

3. Multi-pass priority-rule based methods

We carry out an experimental study of several priority-rule based heuristics for the $J|s_{ij}|C_{\max}$, focusing especially on semi-active, active and non-delay schedule generation schemes (SGS). We show easily from a counterexample that a simple extension of the Giffler and Thompson, 1960 heuristic can be unable to compute the optimal solution. We then exhibit an alternative active SGS able to generate a set of dominant active schedules. We use the 3 proposed SGS as priority-rule based heuristics for the $J|s_{ij}|C_{\max}$. We test both single- and multi-pass methods. In the single pass method a unique priority rule is used to break the ties so as to obtain a single solution. Our conclusion, at least on the 15 $J|s_{ij}|C_{\max}$ instances and for the tested priority rules, is that the non-delay SGS obtains better results than the other ones. In the multi-pass framework, we select a priority rule and a random factor which gives the probability to select another operation than the one elected by the rule. Hence we make a sampling of the solutions reachable by a given SGS. As the number of iterations increases, it appears that the active and semi-active SGS with the MINSTART priority rule[1] are superior to the non-delay SGS. Since a single run of the active or semi-active SGS with the MINSTART rule (and another rule R to break the remaining ties) is equivalent to a run of the non-delay SGS with rule R, we conclude that the non-delay policy is a good one in the presence of sequence-dependent setup times and that it is profitable to search solutions in the neighborhood of non-delay schedules.

4. Tabu search

We study the extension of the disjunctive-graph based local search methods for the standard job-shop problem to the $J|s_{ij}|C_{\max}$. We restrict our attention the the block-based neighborhood proposed by Nowicki and Smutnicki, 1996, denoted as INTB (Interchange Near The Borderline). It aims at selecting a block of consecutive critical operations (located on a critical path of the disjunctive graph) and at allowing the interchange of 2 adjacent operations in a block, except for 2 internal operations. Furthermore, INTB is empty when there is a single block. Again, from a counterexample, we show that a local search method based on INTB may be unable to reach the optimal solution. Hence we propose an alternative neighborhood switching to a larger neighborhood (the interchange of 2 internal operations is allowed, even if there is only one block) when INTB becomes empty.

5. Lower bound computation and exact method

Removing the precedence constraints between the operations in the $J|s_{ij}|C_{\max}$ gives m independent traveling salesman problems. We study such a relaxation, introducing operation time windows to obtain a strengthened relaxation made of m independent traveling salesman problems with time windows (TSPTW). The time windows are initialized using an upper bound of the makespan computed by the above-described heuristics. Then, the time windows are reduced by computing longest path computations in the precedence graph associated with basic resource constraint propagation techniques (namely disjunctive constraint propagation and edge-finding ignoring setup times, see Baptiste et al., 2001). The TSPTW relaxation is computed to obtain an initial lower bound. The optimal solution is then searched between the lower and upper bounds by a dichotomizing tree-search procedure. Based on a basic branching scheme, the procedure uses the decision variant of the TSPTW relaxation to prune the nodes of the search tree. Its originality lies in solving exactly the TSPTW relaxation by branch-and-bound, memorizing a limited number of found solutions in a dictionary. For each node (of the $J|s_{ij}|C_{\max}$ tree), the dictionary is searched to check whether a previously computed TSPTW solution is still feasible. In the positive case, no solving of the relaxation is needed.

6. Computational experiments

We have tested the proposed methods on the 15 $J|s_{ij}|C_{\max}$ instances proposed by Brucker and Thiele, 1996. These instances have the particularity that each operation is associated with a setup type and the sequence-dependent setup time matrix is indexed by the setup types. In these instances however, all the operations of the same job have the same setup type[2]. The instances are made of 3 series of $5 \times 10 \times 5$, $5 \times 15 \times 5$ and $5 \times 20 \times 10$ machines $\times$ jobs $\times$ setup types, respectively.

The priority-rule based sampling method has been coded in C++, whereas the tabu search method have been coded in CLAIRE. The exact method has been coded in C++ using ILOG Scheduler. The machine used is a Pentium 4 with 2.4 Ghz and 512 MB RAM. The makespan values obtained by the methods are given in table 6 for the 15 BT instances. The maximal time used by the sampling method with 10000 iterations (column H1) is always lower than 3 seconds. The maximal time used by the tabu search method (column H2) does not exceed 41 seconds. The CPU times of the exact method (column H3 for the best upper bound and column LB for the best lower bound) are given in column CPU. When the CPU tile is not indicated, the optimality could not be proven in a limit of 5 hours. The exact method is unable at the present time to solve the 5 largest problems. The results are compared with the previous best known solutions obtained by Brucker and Thiele, 1996 and Focacci et al., 2000 (column prev).

inst	prev	H1	H2	H3	LB	CPU
1	798	818	798	798	798	522
2	784	829	784	784	784	7.7
3	749	783	771	749	749	47.8
4	730	745	743	730	730	34.3
5	691	693	693	692	692	30.2
6	1056	1026	1026	1026	996	-
7	1087	1039	1022	970	970	16650
8	1096	1019	994	1002	923	-
9	1119	1060	1060	1060	1037	-
10	1058	1036	1018	1018	1018	498
11	1658	1536	1509	NA	NA	NA
12	1448	1319	1305	NA	NA	NA
13	1549	1439	1439	NA	NA	NA
14	1592	1492	1492	NA	NA	NA
15	1744	1556	1556	NA	NA	NA

Table 1. Results on the 15 BT $J|s_{ij}|C_{\max}$ instances

The table shows that the exact method is able to solve 2 previously unsolved instances (7 and 10). However instance 7 is solved at the expense of large computational time and instance 10 is solved by initial upper and lower bound computations. The best known results are improved on all the 10 medium and large instances. The benefit of using the tabu search method appears on some instances, with an additional CPU time requirement. The medium quality results obtained by tabu search underline the need of testing more efficient neighborhoods. The quality of the initial TSPTW lower bound is good in comparison to the initial lower bounds computed for the branch-and-bound method of Brucker and Thiele, 1996 but rather costly in CPU time.

7. Conclusion and future work

We have studied the dominance properties of standard SGS and neighborhoods proposed so-far for the problem. With the heuristics issued from this study, we have significantly improved the best known solutions on a set of $J|s_{ij}|C_{\max}$ benchmark instances. The good upper bounds provided by these heuristics associated to a TSPTW based lower bound allowed us to close 2 medium size instances, with the help of a branch-and-bound method.

The proposed lower bounds could be further improved by a closer look of the TSPTW relaxations. We also intend to test new insertion-based neighborhoods.

Notes

1. MINSTART: select the operation with the smallest starting time

2. In this case, the model is equivalent to the one presented in Section 1, which is not true in general.

References

Allahverdi, A., Gupta, J. N. D., and Aldowaisan, T. (1999). A review of scheduling research involving setup considerations. *Omega*, 27:219–239.

Artigues, C., Belmokhtar, S., and Feillet, D. (2004). A new exact solution algorithm for the job shop problem with sequence-dependent setup times. In J.C.-Régin and Rueher, M., editors, *1st International Conference on Integration of AI and OR Techniques in Constraint Programming for Combinatorial Optimization Problems*, Lecture Note in Computer Science 3011, pages 37–49. Springer.

Artigues, C. and Buscaylet, F. (2003). A fast tabu search method for the job-shop problem with sequence-dependent setup times. In *Metaheuristic International Conference MIC'2003*.

Artigues, C., Lopez, P., and Ayache, P. (2005). Schedule generation schemes and priority rules for the job-shop problem with sequence-dependent setup times: Dominance properties and computational analysis. *Annals of Operations Research*. to appear.

Baptiste, P., Le Pape, C., and Nuijten, W. (2001). *Constraint-based scheduling: applying constraint programming to scheduling problems*. Kluwer Academic Publishers.

Brucker, P. and Thiele, O. (1996). A branch and bound method for the general-shop problem with sequence-dependent setup times. *Operations Research Spektrum*, 18:145–161.

Choi, I.-N. and Choi, D.-S. (2002). A local search algorithm for jobshop scheduling problems with alternative operations and sequence-dependent setups. *Computers and Industrial Engineering*, 42:43–58.

Focacci, F., Laborie, P., and Nuijten, W. (2000). Solving scheduling problems with setup times and alternative resources. In *Fifth International Conference on Artificial Intelligence Planning and Scheduling*, pages 92–101.

Giffler, B. and Thompson, G. (1960). Algorithms for solving production scheduling problems. *Operations Research*, 8(4):487–503.

Nowicki, E. and Smutnicki, C. (1996). A fast taboo search algorithm for the job shop problem. *Management Science*, 42:797–813.