CONCLUSION AND FURTHER RESEARCH DEVELOPMENTS

CONCLUSION

We have carried out an in-depth study of the hybrid flowshop (HFS) scheduling problem. First, we have presented a general study of the production planning and scheduling problem in the HFS facility in order to emphasize the importance and the role of scheduling. A general decomposition-based heuristic to deal with the HFS production planning and scheduling problem has been proposed. The main issue was the effective use of the available capacity. We then tackled the scheduling problem in the HFS facility through the study of branch and bound algorithms. Several lower and upper bounds as well as branching schemes were developed. Based on this study an effective new branch and bound algorithm has been designed.

Upper bounds

We studied five classes of heuristics using different scheduling approaches. Numerical tests on several shop configurations have been carried out. Statistical analysis was used to select the best set of heuristics. Only a small set of heuristics, out of the 31 which had been evaluated, was selected as robust heuristics (the best ones to be used without significant loss of good solutions with respect to the tested configurations). The selected heuristics belong to the first and the second classes of heuristics only, i.e. the single stage based heuristics and the two-stage flowshop based heuristics. These two classes of heuristics are complementary in the sense that most of the time (for 85% of instances) the best value is given by either the first or the second class of heuristics.

Priority rule based heuristics are outperformed despite the large number of tested variants and do not seem to be effective for dealing with the HFS scheduling problem. The same remark is valid for randomly scheduling heuristics which gave the worse solutions. Local search methods (a simple local search and simulated annealing procedures), given the same amount of running time as other heuristics, are outperformed by the single stage based heuristics. The two-stage flowshop based heuristics as well as those scheduling the single stage subproblem using Jackson’s heuristic are selected as the best and the least time consuming heuristics.

Lower bounds

All developed lower bounds are based upon the single stage HFS relaxation (parallel machine problem with release dates and tails). At the beginning of our research, the best known lower bounds were the Preemptive Bound (PB) computed in $O(p_{max}n^3)$ and subset bound (SSB) which can be computed in $O(n^2)$ where n is the number of jobs and p_{max} is the maximum processing time. Three new lower bounds have been introduced. The first lower bound is a Dual Heuristic based Bound (DHB) induced by a time indexed Linear

Program (LP) formulation where the objective is to build a non-preemptive schedule. The second lower bound builds a partially preemptive schedule (PPB) obtained by constraining the preemptive assignment of jobs. PB and PPB rely on the maximum flow algorithm. We introduced an improved version of the “Preflow Push” algorithm for bipartite graph. The third lower bound is a Heuristic for SSB (HSSB) which can be calculated in a time complexity of O(nM) where M is the number of machines.

All existing lower bounds have been tested on several classes of configurations. For the “*Preflow push”* algorithm, our improved version runs 25-30% faster than the original one although the theoretical complexity has not changed. Also, based on numerical tests, DHB was retained as the best lower bound giving the largest percentage of the best-obtained values, i.e. 99%. Apart from DHB, PPB dominates all the other tested lower bounds including PB. DHB dominates PPB and runs much faster. Unfortunately, DHB has a pseudo-polynomial running time complexity. For all tested instances HSSB gives the same values as PB and SSB and runs faster. Actually, HSSB gives the best value-running time complexity ratio. Only three of all tested lower bounds are retained, i.e. the Set Bound (SB), HSSB and DHB. According to our tests none dominates the other ones. These lower bounds are eventually selected for developing a branch and bound algorithm.

Branch and bound methods

We have studied the branch and bound techniques for solving the HFS scheduling problem. Two branch and bound algorithms using two different branching schemes were designed. We proposed several improvements to the sequence enumeration algorithm introduced by Brah and Hunsucker (1991). We have shown, by numerical tests, that the new obtained algorithm outperforms the original one. The second branch and bound algorithm is obtained by generalizing the so-called interval method to the HFS case. Based on the studied upper and lower bounds, several bounding strategies have been proposed. Both algorithms along with the different bounding strategies were tested on different HFS configurations with reasonable sizes (15 to 30 jobs). The following conclusions have been drawn.

The sequence enumeration method is largely outperformed by the interval method. The interval method is more robust than the sequence enumeration method. It gives a large number of optimal solutions even for HFS problems with large sizes (15 to 30 jobs and 5 stages). After several minutes of running time (i.e. 10 minutes) the interval method has reduced the gap between the root upper bound and the best-known lower bound by more than 60%. For the interval method, the use of tighter lower bound (HSSB), though more time consuming, provides better solutions.

Object Model for Scheduling Algorithm Implementations

Extensive study and design of several scheduling algorithms have led us to implement a large variety of scheduling algorithms. For efficiency’s sake and in order to reduce the complexity as well as the time needed for computer implementation, object oriented modeling has been used so as to design a global Object Model for Scheduling Algorithm Implementations (OMSAI). The model is presented (see appendix) as a set of definitions of object classes. The main objective is to present the general framework within which the proposed scheduling algorithms have been developed. One can then either use or improve the proposed model for future implementation purposes, or simply draw one's inspiration for developing a specific algorithm.

FURTHER RESEARCH DEVELOPMENTS

On the basis of the study conducted in this thesis, we hereafter outline some topics for further research developments.

By using the proposed scheduling algorithms in this project, one can carry out numerical implementations of the decomposition-based heuristic (proposed in Chapter I) for the HFS production planning and scheduling problem.

In this thesis, we have tackled the HFS scheduling problem where stages are made up of *identical* parallel machines and where jobs do not have sequence dependent setup times. As far as the upper bounds are concerned, one can adapt the proposed heuristics to the case where the machines are non-identical and/or to the case where the jobs have sequence dependent setup times.

The tested lower bounding strategies have shown that a tighter and slightly more time-consuming lower bound provides better solutions. Based on this observation, we do believe that tighter lower bounds with improved running time are likely to improve the proposed branch and bound algorithms.

The heuristic for the subset bound (HSSB) runs very fast and, in all tested problems, gives the same results as the preemptive bound. One should then develop lower bounds which are tighter than the preemptive bound. Also, one can enhance the time-indexed formulation proposed for developing the lower bounds (i.e. PPB and DHB). Similarly, one could improve the running time of the dual heuristic based lower bound.

Based on numerical tests, we have stated that the sequence enumeration method is not an effective branching scheme for the HFS case. The branch and bound algorithm using the interval method gives better results which, in turn, might be improved. For example, one could attempt at improving the interval method and/or designing other lower bounding strategies.

We also suggest that the scheduling research community, studying the HFS case, develops a set of standard test problems as the hard ones to be solved. The latter will allow to better appreciate the newly proposed algorithms.

As the final conclusion to this thesis, we would like to point out that this field of investigation is riddled with different research orientations affecting a great many deal of topics and which have many further possible studies in store.