Scheduling a single resource using a B&B approach to limit the total tardiness under uncertainty

Urgo Marcello[1]

Manufacturing and Production Systems Laboratory, Mechanical Engineering Department
Politecnico di Milano, Milano, Italy
marcello.urgo@polimi.it

Keywords: Stochastic Scheduling, Total Tardiness, Value at Risk, Aircraft Production.

1 Introduction and problem statement

In production, tardiness-related indicators are widely used to assess the performance of planning against unexpected events stemming from a wide range of sources, both internal and external. Robust scheduling approaches aim at protecting the performance of the schedule by anticipating to a certain degree the occurrence of uncertain events and, thus, avoiding or mitigating the impact of extremely unfavorable events with low probability. In this paper we address the scheduling of a set of jobs A on a single resource. Each job j has a stochastic processing time p_j and stochastic release date r_j following a general distribution. The schedule must take into consideration a set of deterministic due dates d_j. The aim is at minimizing the value-at-risk of the total tardiness using a branch-and-bound algorithm. The value-at-risk is one of the most common risk indicators used in the financial literature to consider the impact of uncertain events both in terms of their effect and of their occurrence probability but, in the scheduling area, risk-based approaches are not so popular (Cai *et al.* 2007, Tolio *et al.* 2011). A vast research effort has been devoted to the total tardiness problem with equal release dates (Lawler 1977) and its NP-hardness (Du and Leung 1990). The stochastic version of has been addressed in (Gutjahr *et al.* 2011) to minimize the expected total weighted tardiness through a stochastic branch-and-bound technique and a sampling approach. Alternatively, scenario-based approaches can be used to optimize the worst-case performance (Yang and Yu 2002, de Farias *et al.* 2010, Lu *et al.* 2011). All these papers define specific robustness measures (e.g. maximum or maximum relative deviation from optimality, the maximum value over the whole set of scenarios), while (Aloulou and della Croce 2008) also provide several complexity results in the domain of robust single-machine scheduling.

2 Description of the approach

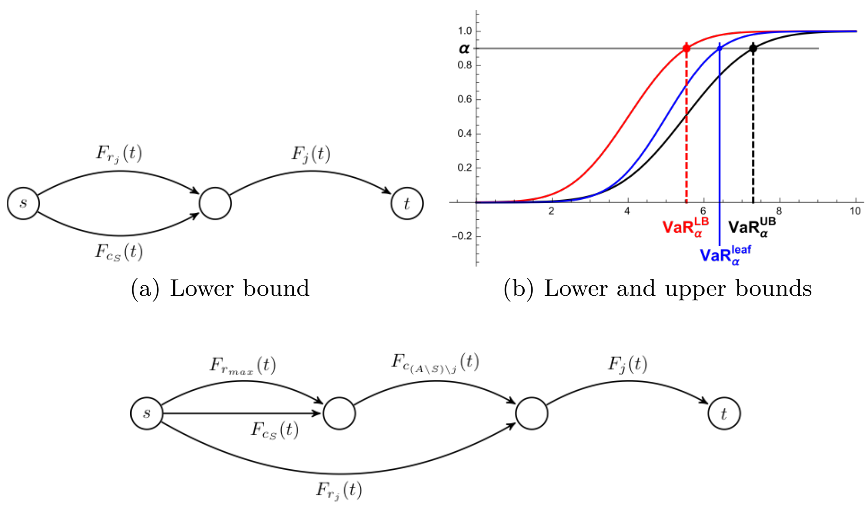
The proposed approach grounds on a a sequential decision scheme where, at each step, the job to be scheduled next on the considered single resource is selected among the un-scheduled ones. The formalization of this scheme can be provided in terms of a tree, with a root node with as many successors as the number of the jobs to be scheduled and , similarly, at each level h of the tree, h jobs have already been scheduled while the remaining $n-h$ defines just as many departing branches. The contribution to the objective function of the set of the already scheduled jobs, S can be easily calculated. On the contrary, the impact of set of the set of not yet scheduled jobs $A \backslash S$ cannot be determined. To this aim, it is necessary to define proper bounds (a lower and an upper one) allowing the selection of the most promising branches and the pruning of the dominated ones. The lower bound distribution of the completion time of a not scheduled job j, $F^{LB}_{C_j}(t)$,can be estimated according to

Figure 1(a), supposing that it is going to be scheduled immediately after the already scheduled jobs and, consequently, its tardiness can be defined as $F^{LB}_{T_j}(t|t \geqslant 0) = F^{LB}_{C_j}(t + d_j)$. Grounding on this, the lower bound of the considered node can be calculated hypothesizing that all the unscheduled jobs are sequenced immediately after the already defined sequence, hence:

$$F^{LB}_{TT}(t) = \circledast_i F_{T_j}(t) \ * \ \circledast_j F^{LB}_{T_j}(t) \qquad \forall i \in A, j \in A \backslash S \tag{1}$$

where $*$ is the convolution operator.
On the contrary, the upper bound distribution of a not scheduled job j can be estimated supposing that it will be scheduled as the last in the sequence as described in Figure 1(c) where $F_{r_{max}}$ is the maximum release data among the not scheduled jobs. Notice that, in theory, it could be more accurate to calculate $F_{r_{max}}$ over all the unscheduled jobs excluded j. Nevertheless the proposed approach allows a much more fast calculation during the exploration of the branching tree since the maximum release date does not need to be recalculated for each job to be scheduled in the last position. Given $F^{UB}_{C_j}(t)$ the upper bound distribution of the completion time of $j \in A \backslash S$, its tardiness can be calculated as $F^{UB}_{T_j}(t|t \geqslant 0) = F^{UB}_{C_j}(t + d_j)$ and the upper bound distribution for the total tardiness is:

$$F^{UB}_{TT}(t) = \circledast_i F_{T_j}(t) \ * \ \circledast_j F^{UB}_{T_j}(t) \qquad \forall i \in A, j \in A \backslash S \tag{2}$$



(a) Lower bound

(b) Lower and upper bounds

(c) Upper bound

Fig. 1. Definition of the bounds.

Figure 1(b) shows how the red and black distributions represents a lower and upper bound respectively for the blue distribution that is the one associated to a leaf departing from the considered node. Since the total tardiness is a regular objective function, the values of the cdf of a node must be greater or equal to the values of any of the successor nodes. Hence, at each node in the branching tree, the lower bound cdf effectively provides a bound on the lower bound cdf of all the successor nodes. In addition, the lower bound cdf stochastically dominates the lower bound cdfs of all the successor nodes. Moreover, since at the leaves of the tree the upper and lower bound cdfs collapse into a single distribution, then this distribution is also stochastically dominated by the lower bound cdfs of all its ancestor nodes. The figure also shows how to obtain bounds for the value at risk.

3 Testing

The proposed branch-and-bound approach has been implemented in C++ using the BoB++ library. The computational experiments have been executed on 16 parallel threads on a workstation equipped with an Intel Eight-Core Xeon Processor E5-2650v2 running at 2.6 GHz and 64 GB of RAM. The performance of the algorithm has been analyzed in terms of the time to find an optimal solution and the fraction of nodes in the tree explored on a test set of twenty 10-job scheduling instances generated using the following steps:

1. the processing times of the jobs is distributed according to a discrete triangular distribution whose mean is sampled from a discrete uniform distribution between 50 and 150 (50% of the instances) or between 25 and 75, the coefficient of variation is between 0.4 and 0.6 (50% of the instances) or between 1.4 and 1.6 and the skewness is randomly assigned the value -0.5, 0 or 0.5;
2. the release times follow a discrete uniform distribution whose mean is sampled from a discrete uniform distribution between 1 and 10 times the mean of the processing time of the job while the half-width is sampled from a discrete uniform distribution between 40 and 60 (50% of the instances) or between 120 and 160;
3. the due dates are deterministic and are sampled from a discrete uniform distribution between 0 and 50 (50% of the instances) or between 150 and 200;

The instances are used to run the optimization algorithm with different risk levels (1% and 5%) for a total of 80 experiments. The proposed algorithm was able to find the optimal solution in 79.89 seconds on average, ranging from a minimum of 9.07 to a maximum of 285.39 seconds. To find the optimal solution, the branch-and-bound approach evaluates 1315 nodes on average over a total of 6 235 300. Nevertheless some difference in the behavior has been experienced according to the different characteristics of the instances. This is shown in Figure 2 (left) reporting the solution time and the number of nodes evaluated for the different experiments according to the average processing times of the jobs, showing a difference among the two groups.

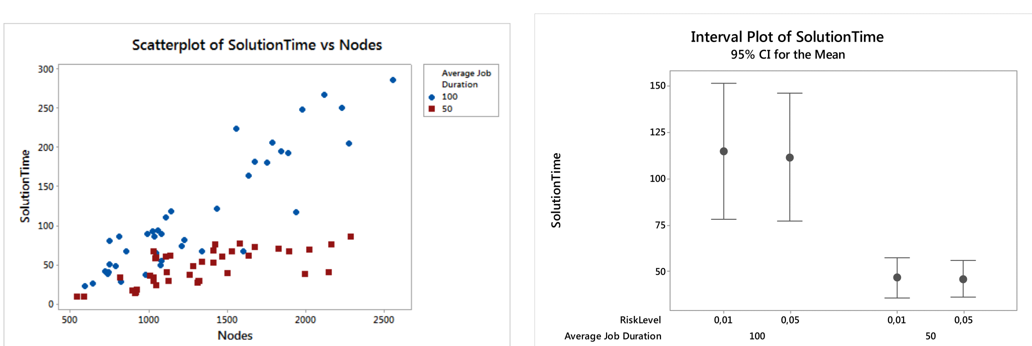


Fig. 2. Solution time: scatter plot against the number of node (left) and interval plot in relation to the average duration and the risk level (right).

This is further supported in Figure 2 (right) demonstrating that the instances with higher average processing time of the jobs require more time to be solved while the impact of the considered level of risk seems negligible. The difference is caused by the time needed to solve a node of the tree. In particular, the amplitude of the support of the distributions influences the time needed to perform the convolution operations, since the combinations of values to be considered is bigger. Furthermore, the time needed to perform the convolution calculations also depends on the support of the distribution and, hence, on the length of the duration of the activities.

The performance of the proposed branch-and-bound approach is reasonably fast in terms of the time needed to find the optimal solution. Needles to say that the dimension of the solved instances (10 jobs) is rather small and the performances will be significantly decreasing as the number of jobs to be scheduled increases. However, the parallel capabilities of the implementation allow to easily exploit the benefits of new multi core architectures or the execution on high performance calculation environments. Moreover, the proposed exact solution approach finds its most favorable application with a small number of activities since, as the number of jobs increases, the approximation of the distribution of the total tardiness with a normal one is a good option.

4 Industrial application and acknowledgements

The proposed approach has been inspired and foresees a potential application in the production of aircrafts where different components are produced in different plants and then assembled together. Due to the extremely large dimension of the components to be moved (typically the fuselage and the wings) special ships or aircrafts are booked to operate the transportation. Due to this, the total tardiness is an objective function suitable to identify the time these special vehicles are waiting for the parts to be moved and, hence, the extra cost associated. This research was supported by the EU projects "ProRegio - Customer-driven design of product-services and production networks to adapt to regional market requirements", Grant 636966.

References

Cai, X., Wang, L., Zhou, X., 2007. Single-machine scheduling to stochastically minimize maximum lateness, *Journal of Scheduling*, 10 (4), pp. 293–301.

Tolio, T. , Urgo, M. and Váncza, J., 2011. Robust production control against propagation of disruptions, *CIRP Annals - Manufacturing Technology*, 60 (1), pp. 489–492.

Emmons, H., 1969. One-machine sequencing to minimize certain functions of job tardiness, *Operations Research*, 17, pp. 701–715.

Lawler, E.L., 1977. A pseudo-polynomial algorithm for sequencing jobs to minimize total tardiness. *Annals of Discrete Mathematics*, 1, pp. 331–342.

Du J., Leung, J.Y.T., 1990. Minimizing total tardiness on one processor is NP-hard. *Mathematics of Operations Research*, 15, pp. 483–495.

Gutjahr, W. J., Hellmayr, A., Pflug, G. C., 1999. Optimal stochastic single-machine-tardiness scheduling by stochastic branch-and-bound. *EJOR*, 117 (2), pp. 396–413.

Yang, J., Yu, G., 2002. On the robust single machine scheduling problem. *Journal of Combinatorial Optimization*, 6 (1), pp. 17–33.

Lu, C.-C., Lin, S.-W., Ying, K.-C., 2012. Robust scheduling on a single machine to minimize total flow time. *Computers and Operations Research*, 39 (7), pp. 1682–1691.

de Farias, JR, I. R., Zhao, H., Zhao, M., 2010. A family of inequalities valid for the robust single machine scheduling polyhedron. *Computers and Operations Research*, 37 (9), pp. 1610–1614

Aloulou, M. A., della Croce, F., 2008. Complexity of single machine scheduling problems under scenario-based uncertainty. *Operations Research Letters*, 36 (3), pp. 338–342.