A hybrid algorithm combining heuristics with Monte Carlo simulation to solve the Stochastic Flow Shop Problem

Esteban Peruyero * Angel A. Juan * Daniel Riera *

* Open University of Catalonia
Barcelona, 08018, SPAIN
ajuanp@gmail.com

ABSTRACT

In this paper a hybrid simulation-based algorithm is proposed for the Stochastic Flow Shop Problem. The main idea of the methodology is to transform the stochastic problem into a deterministic problem and then apply simulation. To achieve this goal we use Monte Carlo simulation and a modified version of the well-known NEH heuristic. This approach aims to provide flexibility and simplicity due to the fact that it is not constrained by any previous assumption and relies in well-tested heuristics.

Keywords: Scheduling, Monte-Carlo simulation, Heuristics, Randomized algorithm

1. INTRODUCTION

The Flow Shop Problem (FSP) is a well-known scheduling problem in which a set of independent jobs have to be sequentially executed (processed) by a set of machines. In this scenario, the processing time of each job in each machine is a known constant value. The classical FSP goal is to determine a sequence of jobs minimizing the total makespan, which is the time difference between the start and finish of processing all the jobs in all the machines (Figure 1).

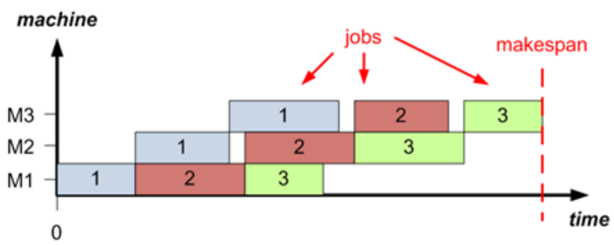

Figure 1: A graphical representation of the FSP

The Stochastic Flow Shop Problem (SFSP) can be seen as a generalization of the FSP. In this non-deterministic version of the Flow Shop Problem, the processing time of each job in each machine is not a constant value, but instead it is a random variable which follows a given probability distribution. Therefore, in this scenario the goal uses to be minimizing the expected makespan, which is not the same as the expected total processing time. The study of the SFSP is within the current popularity of introducing randomness into combinatorial optimization problems. It allows to describe new problems in more realistic scenarios where uncertainty is present.

It is important to remark the FSP as a relevant topic for current research. As it happened with other combinatorial optimization problems, a large number of different approaches and methodologies have been developed to deal with the FSP. These approaches range from pure optimization methods (such as linear and integer programming), which allow to solve small-sized problems, to approximate methods such as heuristics and metaheuristics, which can find near-optimal solutions for medium- and large-sized problems. Although the usual goal is to minimize the makespan, other goals could also be considered, e.g. to minimize the total processing time. Moreover, some of these methodologies are able to provide a set of near-optimal solutions from which the decision-maker can choose according to his/her specific utility function. The situation is quite different in the case of the SFSP: to the best of our knowledge, there is a lack of efficient and flexible methodologies able to provide near-optimal solutions to the stochastic version of the FSP. Moreover, most of the existing approaches are quite theoretical and make use of restrictive assumptions on the probability distributions that model job processing times.

2. BASIC NOTATION AND ASSUMPTIONS

The Stochastic Flow Shop Problem (SFSP) is a scheduling problem that can be formally described as follows: a set J of n independent jobs have to be processed by a set M of m independent machines. Each job $i \in J$ requires a stochastic processing time, p_{ij}, in every machine $j \in M$. This stochastic processing time is a random variable following a certain distribution, e.g. log-normal, exponential, weibull, etc. The goal is to find a sequence for processing the jobs so that a given criterion is optimized. The most commonly used criterion is the minimization of the expected completion time or expected makespan, denoted by $E[C_{max}]$. In addition, it is also assumed that:

- All jobs are processed by all machines in the same order.
- There is unlimited storage between the machines, and non-preemption.
- Machines are always available for processing jobs, but each machine can process only one job at a time.
- A job cannot be processed more than once for each machine.
- Job processing times are independent random variables.

At this point, it is interesting to notice that our approach does not require to assume any particular distribution for the random variables that model processing times. In a practical situation, the specific distributions to be employed will have to be fitted from historical data (observations) using a statistical software. In most existing approaches, however, it is frequently assumed that these processing times will follow a normal or exponential distribution. This assumption is, in our opinion, quite unrealistic and restrictive. For instance, it is unlikely that positive processing times can be conveniently modeled throughout a normal distribution, since any normal distribution includes negative values .

3. STATE OF THE ART AND RELATED WORK

The FSP is a NP-complete problem [1]. Many heuristics and metaheuristics have been proposed in order to solve the FSP due to the impossibility of finding, in reasonable times, exact solutions for most medium- and large-sized instances. Some of the first publications on FSP are those of Johnson [2] and Makino[3]. These authors presented approaches for solving small problems, e.g. problems with only two machines and two jobs. Campbell et al. [4] built a heuristic for the FSP with more than two machines. The NEH algorithm is considered by most researchers as one of the best performing heuristics for solving the FSP. It was introduced by Nawaz et al. [5]. Later, Tailard [6] reduced the NEH complexity by introducing a data structure to avoid the calculation of the makespan. Ruiz and Stützle [7] proposed the Iterated Greedy (IG) algorithm for the FSP built on a two-step methodology. In our opinion, this is one of the best algorithms developed so far to solve the FSP, since it combines simplicity with an outstanding performance.

Many works have focused on the importance of considering uncertainty in real-world problems, particularly in those related to scheduling issues. Thus, Al-Fawzan[8]analyzes the Resource Constrained Project Scheduling Problem (RCPSP) by focusing on makespan reduction and robustness. Jensen[9] also introduces the concepts of neighborhood-based robustness and tardiness minimization. Ke [10] proposes a mathematical model for achieving a formal specification of the Project Scheduling Problem. Allaoui [11] studied makespan minimization and robustness related to the SFSP, suggesting how to measure the robustness. Proactive and reactive scheduling are also characterized in his work. On the one hand, an example of reactive scheduling can be found on Honkomp et al. [12], where performance is evaluated using several methodologies. On the other hand, robustness in proactive scheduling is analyzed in Ghezail et al. [13], who propose a graphical representation of the solution in order to evaluate obtained schedules. As the concept of minimum makespan from FSP is not representative for the stochastic problem, Dodin [14] proposes an optimality index to study the efficiency of the SFSP solutions. The boundaries of the expected makespan are also analyzed mathematically. A theoretical analysis of performance evaluation based on markovian models is performed in Gourgand et al. [15], where a method to compute expected time for a sequence using performance evaluation is proposed. A study of the impact of introducing different types of buffering among jobs is also provided in this work. On the other hand, Integer and linear programming have been employed together with probability distributions to represent the problem in Janak et al. [16].

Simulation has been applied in Juan et al. [17] to solve the FSP. In this work, the NEH algorithm is randomized using a biased probability distribution. Thus, their approach is somewhat similar to a GRASP-like methodology. Simulation-based approaches for the SFSP have mainly focused on performance evaluation, as in Gougard et al. [18]. Similarly, Dodin [14] performs simulations as a way to validate his empirical analysis on the makespan boundaries. Finally, Honkomp et al. [12] also make use of simulation techniques in their approach for reactive scheduling.

In a recent work, Juan et al. [19] describe the application of simulation techniques to solve the Vehicle Routing Problem with Stochastic Demands (VRPSD). The VRPSD is a variation of the classical Vehicle Routing Problem where customer demands are not known in advance. These demands are random variables following some probability distributions. The authors propose to transform the original stochastic problem into a set of related deterministic problems, which are then solved using an efficient algorithm introduced in a previous work [20]. As it will be discussed in more detail next, this paper proposes a similar approach for solving the SFSP.

4. PROPOSED METHODOLOGY

The main idea behind our simulation-based approach is to transform the initial SFSP instance into a FSP instance and then to obtain a set of near-optimal solutions for the deterministic problem by using an efficient FSP algorithm. Notice that, by construction, these FSP solutions are also feasible solutions of the original SFSP instance. Then, simulation is used to determine which solution, among the best-found deterministic ones, shows a lower expected makespan when considering stochastic times. This strategy assumes that a strong correlationship exists between near-optimal solutions for the FSP and near-optimal solutions for the SFSP. Put in other words, good solutions for the FSP are likely to represent good solutions for the SFSP. Notice, however, that not necessarily the best-found FSP solution will become the best-found SFSP solution, since its resulting makespan might be quite sensitive to variations in the processing times. The transformation step is achieved by simply considering the expected value of each processing time as a constant value. Since any FSP solution will be also a feasible SFSP solution, it is possible to use Monte Carlo simulation to obtain estimates for the expected makespan. That is, we obtain these estimates by iteratively reproducing the stochastic behaviour of the processing times in the sequence of jobs given by the FSP solution. Of course, this simulation process will take as many iterations as necessary to obtain accurate estimates. If necessary, variance reduction techniques could be employed in order to reduce the number of iterations to run. Figure 2 shows the flow chart diagram of our approach, which is described next in detail:

1. Consider a SFSP instance defined by a set J of jobs and a set M of machines with random processing times, p_{ij}, for each job $i \in J$ in each machine $j \in M$.
2. For each random processing time p_{ij}, consider its expected or mean value $p^*_{ij} = E\left[p_{ij}\right]$.
3. Let FSP* be the non-stochastic problem associated with the processing times p^*_{ij}, $\forall i \in J, j \in M$.
4. Using any efficient algorithm (e.g. [7, 17]), obtain a set S of n near-optimal solutions for the FSP*.
5. For each $s_k \in S$, $k = 1, 2, \ldots n$, consider the sequence of jobs in s_k and then start a Monte Carlo simulation in order to estimate the expected makespan associated with this sequence of jobs. Notice that for each s_k, random observations from each p_{ij} ($i \in J, j \in M$) are iteratively generated while maintaining the sequence of jobs provided by s_k.
6. Return the sequence of jobs (solution) which provides the lowest expected makespan.

5. CONTRIBUTION OF OUR APPROACH

The idea of solving a stochastic combinatorial optimization problem through solving one related deterministic problem and then applying simulation is not new (see [19]). However, to the best of our knowledge, this is the first time this approach has been used to solve the SFSP. In fact, most of the SFSP research to date has focused on theoretical aspects of stochastic scheduling. By contrast, the proposed method provides a relatively simple and flexible approach to the SFSP, which in our opinion offers some valuable benefits. In particular, our approach suggests a more practical perspective which is able to deal with more realistic scenarios: by integrating Monte Carlo simulation in our methodology, it is possible to naturally consider any probabilistic distribution for modeling the random job processing times.

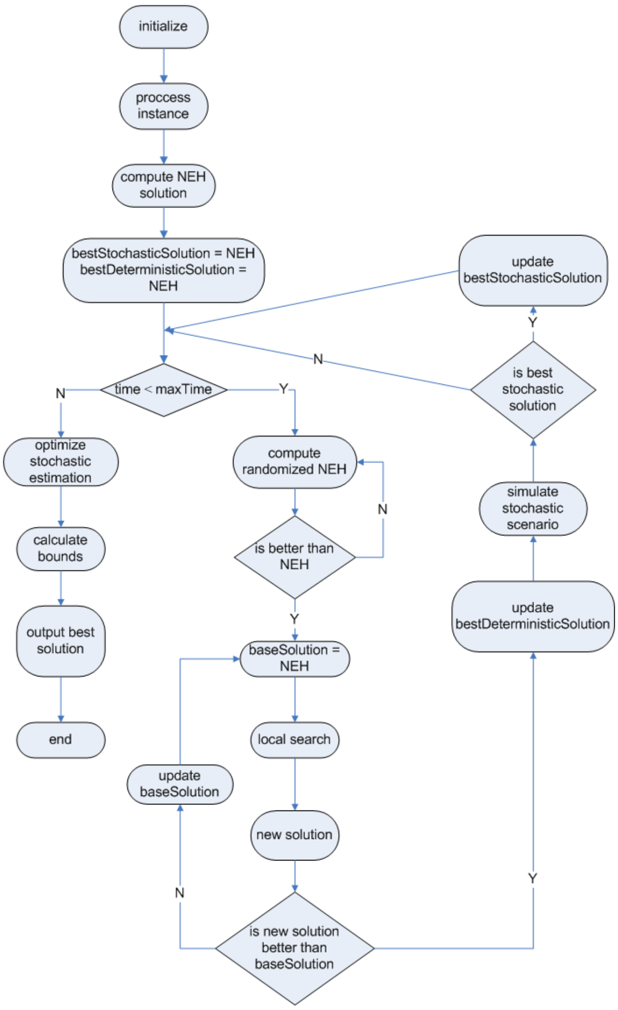

Figure 2: Flow chart of the proposed algorithm

Thus, as far as we know, the presented methodology offers some unique advantages over other existing SFSP approaches. To be specific: (a) the methodology is valid for any statistical distribution with a known mean, both theoretical -e.g. Normal, Log-normal, Weibull, Gamma, etc.- or experimental; and (b) the methodology reduces the complexity of solving the SFSP -where no efficient methods are known yet- to solving the FSP, where mature and extensively tested algorithm have been developed already. All in all, the credibility and utility of the provided solution is increased. Notice also that, being based on simulation, the methodology can be easily extended to consider a different distribution for each job-machine processing time, possible dependencies among these times, etc. Moreover, the methodology can be applied to SFSP instances of any size as far as there exists efficient FSP metaheuristics able to solve those instances. In summary, the benefits provided by our methodology can be summarized in two propierties: simplicity and flexibility.

6. CONCLUSIONS

In this paper we have presented a hybrid approach for solving the Stochastic Flow Shop Problem. The methodology combines Monte Carlo simulation with well tested algorithms for the Flow Shop Problem. The basic idea of our approach is to transform the initial stochastic problem into a related deterministic problem, then obtain a set of alternative solutions for this latter problem using any efficient algorithm, and finally use simulation to verify which of these solutions offers the lowest expected makespan. This approach does not require any previous assumption and is valid for any probability distribution.

7. ACKNOWLEDGEMENTS

This work has been partially supported by the Spanish Ministry of Science and Innovation (TRA2010-21644-C03). This work has been developed in the context of the CYTED-IN3-HAROSA Network (`http://dpcs.uoc.edu`).

8. REFERENCES

[1] A. H. R. Kan, *Machine scheduling problems: Classification, complexity and computations*. Nijhoff (The Hague), 1996.

[2] S. M. Johnson, "Optimal two- and three-stage production schedules with setup times included. naval research logistics," *Naval Research Logistics Quarterly*, no. 1, pp. 61–68, 1954.

[3] T. Makino, "On a scheduling problem," *Operations Research Society of Japan*, vol. 8, pp. 32–44, 1965.

[4] H. G. Campbell, R. A. Dudek, and M. L. Smith, "A heuristic algorithm for the n job, m machine sequencing problem," *Management Science*, vol. 23, no. 16, pp. B630–B637, 1973.

[5] M. Nawaz, E. Enscore, and I. Ham, "A heuristic algorithm for the m-machine, n-job flow-shop sequencing problem," *Omega*, vol. 11, no. 1, pp. 91–95, 1983.

[6] E. Taillard, "Some efficient heuristic methods for the flow shop sequencing problem," *European Journal of Operational Research*, vol. 47, no. 1, pp. 65–74, 1990.

[7] R. Ruiz and T. Stützle, "A simple and effective iterated greedy algorithm for the permutation flowshop scheduling problem," *European Journal of Operational Research*, vol. 177, pp. 2033–2049, 2007.

[8] M. A. Al-Fawzan and M. Haouari, "A bi-objective model for robust resource-constrained project scheduling," *International Journal of Production Economics*, vol. 96, no. 2, pp. 175–187, 2005.

[9] M. T. Jensen, "Improving robustness and flexibility of tardiness and total flow-time job shops using robustness measures," *Applied Soft Computing*, vol. 1, no. 1, pp. 35–52, 2001.

[10] H. Ke and B. Liu, "Project scheduling problem with stochastic activity duration times," *Applied Mathematics and Computation*, vol. 168, no. 1, pp. 342–353, 2005.

[11] H. Allaoui, S. Lamouri, and M. Lebbar, "A robustness framework for a stochastic hybrid flow shop to minimize the makespan," in *International Conference on Service Systems and Service Management*, 2006, pp. 1097–1102.

[12] S. Honkomp, L. Mockus, and G. Reklaitis, "Robust scheduling with processing time uncertainty," *Computers & Chemical Engineering*, vol. 21, no. Supplement 1, pp. S1055–S1060, 1997.

[13] F. Ghezail, H. Pierreval, and S. Hajri-Gabouj, "Analysis of robustness in proactive scheduling: A graphical approach," *Computers & Industrial Engineering*, vol. 58, no. 2, pp. 193–198, 2010.

[14] B. Dodin, "Determining the optimal sequences and the distributional properties of their completion times in stochastic flow shops," *Computers & Operations Research*, vol. 23, no. 9, pp. 829–843, 1996.

[15] M. Gourgand, N. Grangeon, and S. Norre, "Markovian analysis for performance evaluation and scheduling in m machine stochastic flow-shop with buffers of any capacity," *European Journal of Operational Research*, vol. 161, no. 1, pp. 126–147, 2005.

[16] S. L. Janak, X. Lin, and C. A. Floudas, "A new robust optimization approach for scheduling under uncertainty: Uncertainty with known probability distribution," *Computers & Chemical Engineering*, vol. 31, no. 3, pp. 171–195, 2007.

[17] A. Juan, R. Ruiz, H. Lourenço, M. Mateo, and D. Ionescu, "A simulation-based approach for solving the flowshop problem," in *Proceedings of the 2010 Winter Simulation Conference. Baltimore, Maryland, USA.*, 2010, pp. 3384–3395.

[18] M. Gourgand, N. Grangeon, and S. Norre, "A contribution to the stochastic flow shop scheduling problem," *European Journal of Operational Research*, vol. 151, no. 2, pp. 415–433, 2003.

[19] A. Juan, J. Faulin, S. Grasman, D. Riera, J. Marull, and C. Mendez, "Using safety stocks and simulation to solve the vehicle routing problem with stochastic demands," *Transportation Research Part C*, 2010, doi:10.1016/j.trc.2010.09.007.

[20] A. Juan, J. Faulin, J. Jorba, D. Riera, D. Masip, and B. Barrios, "On the use of monte carlo simulation, cache and splitting techniques to improve the clarke and wright savings heuristics," *Journal of the Operational Research Society*, 2010, doi:10.1057/jors.2010.29.