*Proceedings of the 2018 Winter Simulation Conference*
*M. Rabe, A. A. Juan, N. Mustafee, A. Skoogh, S. Jain, and B. Johansson, eds.*

# A SIMHEURISTIC ALGORITHM FOR SOLVING AN INTEGRATED RESOURCE ALLOCATION AND SCHEDULING PROBLEM

Ludovica Maccarrone
Tommaso Giovannelli

ACTOR s.r.l.
DIAG - Sapienza, University of Rome
1-00185 Rome, ITALY

Daniele Ferone
Javier Panadero
Angel A. Juan

Universitat Oberta de Catalunya
Euncet Business School
08018 Barcelona, SPAIN

## ABSTRACT

Modern companies have to face challenging configuration issues in their manufacturing chains. One of these challenges is related to the integrated allocation and scheduling of resources such as machines, workers, energy, etc. These integrated optimization problems are difficult to solve, but they can be even more challenging when real-life uncertainty is considered. In this paper we study an integrated allocation and scheduling optimization problem with stochastic processing times. A simheuristic algorithm is proposed in order to effectively solve this integrated and stochastic problem. Our approach relies on the hybridization of simulation with a metaheuristic to deal with the stochastic version of the allocation-scheduling problem. A series of numerical experiments contribute to illustrate the efficiency of our methodology as well as their potential applications in real-life enterprise settings.

## 1 INTRODUCTION

All enterprises face the problem of managing the resources necessary for their business. This is true, in particular, both for long-term objectives –which involve a proper estimate and planning of the needs– as well as for short-term decisions –such as the optimal proportioning and allocation of resources to operations. In many cases, these resource-management problems have a strong impact on the execution of the activities, and vice versa. Indeed, in most real-life environments it is reasonable to assume that the higher the number of allocated resources is, the lower the time needed to complete the planned tasks will be. Thus, in order to address the resource-management problem in an efficient way it is necessary to balance two conflicting goals: *(i)* minimization of the cost associated with resource utilization; and *(ii)* minimization of the time needed to finish the assigned tasks.

In addition to this trade-off between conflicting optimization goals, real-life is also characterized by uncertainty. In particular, decision makers have to take into account that processing times are usually stochastic in nature. In this paper we consider the integrated allocation and scheduling of resources under stochastic processing times. In particular, the stochastic optimization problem analyzed can be described as a bi-level decision making process (Figure 1):

- **Stage 1 (allocation map):** heterogeneous resources (machines, humans, energy, etc.) have to be allocated to different activities so they can be completed at a reasonably low cost.
- **Stage 2 (scheduling plan):** the activities, that are subject to stochastic processing times, have to be properly scheduled in order to be completed in a reasonably low makespan (i.e., the time required to finish the processing of all jobs on all the machines).

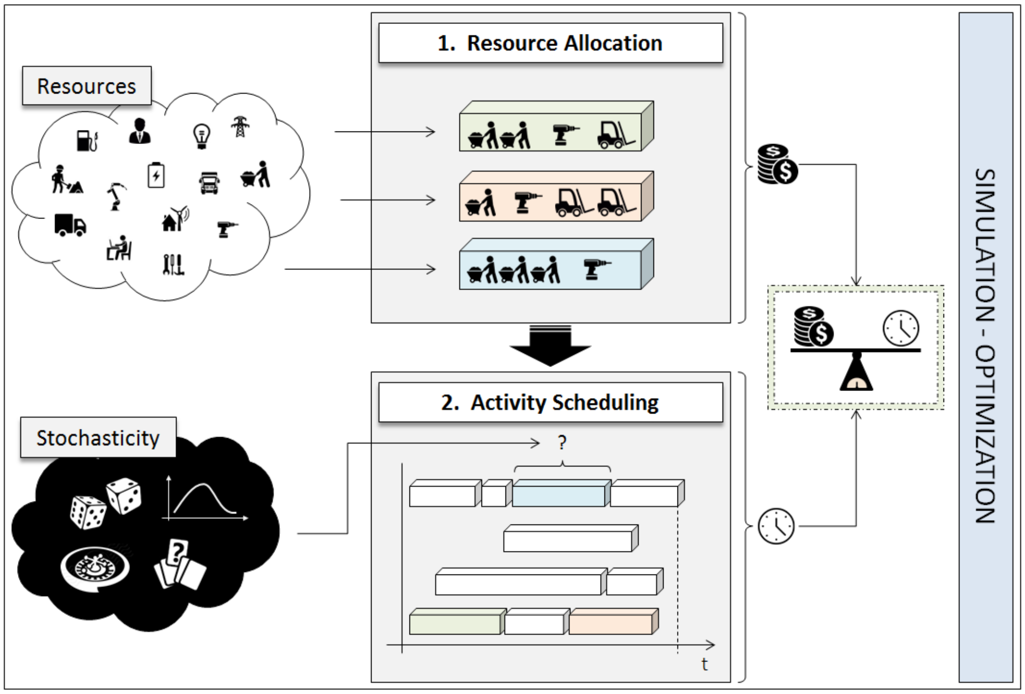

Figure 1: Resource allocation and scheduling under uncertainty.

In the first stage, resources are tentatively allocated to activities (resource allocation map) and sent to the second stage. Here, the activities are scheduled considering the combined effect that resource allocation and stochasticity have on the processing times. During the first stage it is possible to compute the cost associated with the allocated resources. During the second stage an evaluation of the solution, in terms of makespan, is obtained. These two values are then weighted in an overall objective function, which must be able to take into account both financial and time goals (this is achieved by transforming time measures in monetary values). Since the scheduling plan in the second stage depends on the resource allocation map defined in the first stage, different combinations of 'promising' resource allocation maps need to be tested in order to generate high-quality global solutions in terms of total allocation plus scheduling costs.

The main contribution of this paper is the introduction of a simulation-optimization methodology that is able to deal with this integrated resource allocation and stochastic scheduling problem. Our simulation-optimization approach is based on the concept of simheuristics, which hybridize simulation with metaheuristics in order to solve complex combinatorial optimization problems with stochastic components (Juan et al. 2015). Simheuristic algorithms can be seen –at least to some extend– as a specialized case of simulation-based optimization where: *(i)* only metaheuristic algorithms are used in the optimization component; and *(ii)* the simulation component is not only employed to assess the value of a certain objective function or the satisfaction of a given constraint, but also to provide feedback that can be used by the metaheuristic component to improve the solution searching process (Grasas et al. 2016). A Greedy Randomized Adaptive Search Procedure (GRASP) has been proposed for the metaheuristic component (Feo and Resende 1995). GRASP can be easily integrated into a simheuristic framework and offers a good trade-off between quality of the solutions and ease of implementation.

The rest of the paper is structured as follows: Section 2 reviews close solving approaches previously developed in the literature. Section 3 describes in detail the characteristics of the integrated resource allocation and activity scheduling problem considered in this paper. The main ideas behind our simheuristic algorithm are outlined in Section 4. An implementation of the proposed method is described and tested in Section 5. Further possible applications are discussed in Section 6. Finally, Section 7 derives some conclusions.

## 2 RELATED WORK

Resource allocation and scheduling of activities are some of the most well-studied problems in the literature. The integrated version of these problems involves finding the right assignment of resources to each activity, so that their duration is reduced and a balance between the cost of additional resources and the makespan is reached. This topic has been addressed in different research areas. In particular, several works extend previous applications in job scheduling and project management. One of the first works introducing the idea of simultaneously planning resource requirements and scheduling of activities is due to Kelley (1961). The problem they described is known in the literature with the name of Time-Cost Trade-off Problem (TCTP). In the area of project management, the same topic is also referred to as Activity Crashing or Project Compression problem (Elmaghraby 1977). Existing approaches differ in the way they represent the non-increasing pattern between allocated resources and activity duration. On the one hand, some continuous approaches assume a linear function (Baker 1997, Kimms 2001), while others make use of non-linear functions (Moder et al. 1983, Deckro et al. 1992). On the other hand, the discrete version of the TCTP assumes that a discrete set of possible activity durations is given (Kolisch and Padman 2001, Demeulemeester and Herroelen 2002).

While the deterministic version of the TCTP has been widely studied during the last 60 years, the works considering the stochastic counterpart of the same problem are relatively rare (Herroelen and Leus 2005). The existing literature can be classified using different criteria. In particular, different taxonomies can be generated by considering:

- *The type of objective:* Some works take into account conflicting objective functions in a multi-objective approach, while others deal with a single objective function. This function either includes both resources costs and total completion time or only considers one of them. In the latter case, the second objective is generally modeled as a constraint (Godinho and Branco 2012). Azaron et al. (2006) and Azaron and Tavakkoli-Moghaddam (2007) have proposed a multi-objective problem which uses an interactive procedure with four objective functions related to the total cost and total completion time of the project. Mokhtari et al. (2010) and Klerides and Hadjiconstantinou (2010) address the minimization of only one of the two measures and force the other objective under a given threshold introducing an additional constraint. Finally, examples of approaches including time and cost goals in a single weighted objective function can be found in Elmaghraby and Morgan (2007).
- *The type of relation between resource allocation and activity processing times:* As for the deterministic version of the TCTP, a basic distinction can be made considering the nature of patterns between costs and durations. Laslo (2003) and Cohen et al. (2007) assume the existence of a continuous relationship, while Klerides and Hadjiconstantinou (2010) or Godinho and Branco (2012) propose different approaches to solve the stochastic version of the TCTP in the discrete case.
- *The type of assignment (scheduling policy):* In general, the assignment procedure used to allocate resources can be either static or adaptive. The solution is static if the assignment of resources is fixed and does not change as the activities are executed (Bregman 2009, Mokhtari et al. 2010). Otherwise, the policy is adaptive and decisions on assignment can be adjusted over time (Goh and Hall 2013, Kang and Choi 2015).
- *The method for considering stochasticity:* Several approaches have been applied to involve uncertainty. According to Kang and Choi (2015), Monte Carlo simulation was used for the first time by Van Slyke (1963); then, it has been adopted along with heuristic methods by many authors. Other common techniques include robust optimization (Cohen et al. 2007), and stochastic programming (Klerides and Hadjiconstantinou 2010).

In this paper we consider the minimization of a single weighted objective function assuming a continuous relationship between the allocated resources and the stochastic processing times. Our approach integrates Monte Carlo simulation into a metaheuristic framework to simultaneously provide a static resource assignment policy and a feasible activity scheduling.

## 3 THE INTEGRATED RESOURCE ALLOCATION AND SCHEDULING PROBLEM

In the area of scheduling, numerous problems can be distinguished on the basis of the type of configuration considered. In the following, we provide a description of the integrated resource-allocation and scheduling problem as an extension of the well-known Permutation Flow Shop Problem (PFSP). The classical version of PFSP can be described as follows: a set $J$ of $n$ jobs has to be processed by a set $M$ of $m$ machines. Every job $j \in J$ is composed by a set of $m$ operations, each one indicated by $O_{ij}$. The operations must be sequentially performed by the $m$ machines (one operation per machine). Moreover, the processing order of operations in machines is the same for all jobs –i.e., all jobs are processed by all machines in the same order. A processing time, $p_{ij}$, is associated to each operation $O_{ij}$, and is assumed to be known in advance. The goal is to find a sequence (permutation) of jobs so that a given criterion is optimized (Juan et al. 2014). The most commonly and studied criterion is the minimization of the makespan. Adding uncertainty to the PFSP results in the PFSP with stochastic processing times (PFSPST), in which the processing time of each job $j$ in each machine $i$ is not a constant value but a non-negative random variable, $P_{ij}$. Common approaches use for this purpose known probability distributions, such as the Log-Normal, Weibull, etc.

Integrating the resource allocation into the PFSPST involves the definition of additional variables to represent the amount of resources assigned to the machines. In the following, we consider a simple case in which resources are homogeneous and we define the vector $r$, with $r_i$ indicating the assignment of resources to machine $i$. According to the nature of the trade-off problem, we assume that the processing times of the operations change with the value of $r_i$. In general, assigning more resources to a task decreases the time required, but the relation is usually sub-linear (Elmaghraby and Morgan 2007). Without loss of generality, in our numerical experiments we will consider the 'accelerated' processing times, $\hat{P}_{ij}$, as the following function of the single-resource processing times, $P_{ij}$, and the number of resources employed, $r_i$:

$$\hat{P}_{ij} = \frac{P_{ij}}{\sqrt{r_i}} \tag{1}$$

An example with three machines and three jobs is reported in Figure 2. In the first Gantt chart, a single resource is assigned to each machine and the overall makespan is 26. In the second chart, adding a second resource to machine M2 results in lower processing times and a better makespan.

On the one hand, adding more resources will reduce the time needed to process all the operations. On the other hand, these resources will increase the corresponding cost component in the overall objective function. Thus, apart from the cost directly linked to the production volume (e.g., row materials) and the fixed cost, the typical factors influencing the total cost of a manufacturing enterprise are linked to the acquisition of resources (e.g., costs for buying machines, employing people, etc.) and the time spent to carry on activities (renting cost, electricity cost, salaries, etc.). To take into account these aspects in a single minimization goal, we use the following objective function:

$$\lambda \cdot \sum_{i=1}^{m} r_i + \mu \cdot E[C_k] \tag{2}$$

where $E[C_k]$ is the expected makespan of the solution, while $\lambda$ and $\mu$ are two scaling factors for which the conditions $\lambda + \mu = 1$ and $\lambda, \mu \geq 0$ hold. Here, we are assuming homogeneous resources, i.e.: that each resource contributes in the same way to increase the value of Function 2.

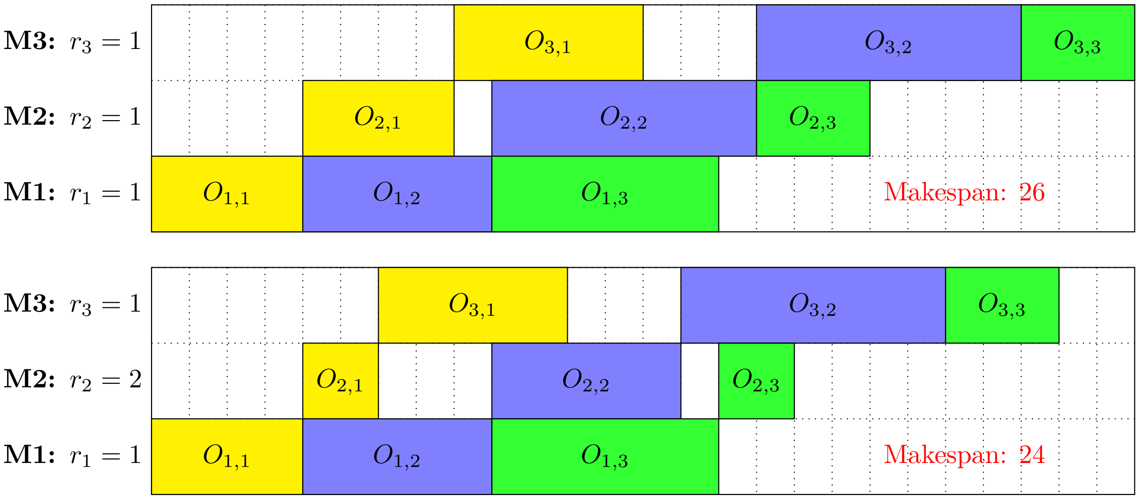

Figure 2: Difference in makespan with more resources assigned to machines.

## 4 OUR SIMULATION-OPTIMIZATION APPROACH

In order to deal with the integrated resource allocation and stochastic scheduling problem described in the previous section, a simheuristic algorithm is proposed. It combines simulation techniques with a GRASP. This is a multi-start algorithm based on a randomized construction process, which is combined with a local search. During the construction phase, a feasible solution is iteratively constructed by adding a new element to the current solution. At each step, the next element to add to the current solution is determined by ordering all candidate elements (that is, all elements that can be feasibly added to the solution) according to a greedy function $g\colon C \to \mathbb{R}$. This function measures the benefit of including an element in the emerging solution. The element is randomly chosen among the most promising candidates, but not necessarily the top one. Notice that this technique tends to generate different solutions every time the multi-start procedure is run, which helps to avoid getting trapped into a local minimum. Once a feasible solution has been constructed, it is locally improved with respect to a neighborhood definition to find a local minimum. This phase is needed, since the initial solution is unlikely to be optimal after the construction phase. We have chosen the GRASP metaheuristic since it is relatively easy-to-implement, does not contain a large number of parameters requiring time-consuming setting processes, and has been successfully applied to a wide range of different optimizations problems (Festa and Resende 2009a, Festa and Resende 2009b, Gonzalez-Neira et al. 2017).

Figure 3 depicts the main characteristics of our simheuristic algorithm, which is composed of three stages. In the first stage, a feasible initial solution is constructed. Then, during the second stage a GRASP algorithm enhances the initial feasible solution by iteratively exploring the search space and conducting a short number of simulation runs. During this stage, a reduced set of promising solutions is obtained. In the third stage, a large number of simulation runs are performed to refine the promising solutions with the objective to obtain the best-possible solution in stochastic terms.

Notice that the first stage is composed of two different phases. First of all, the algorithm carries out a resource assignment. For each machine, the algorithm randomly assigns to it a number of resources between 1 (minimum) and $q$ (maximum). The number of resources assigned to a machine establishes its processing time, which is given by Equation (1). Once the assignment has been made, an initial solution is generated with the help of the well-known NEH constructive heuristic (Nawaz et al. 1983). During the second step of our methodology, the initial solution (*initSol*) is improved using the GRASP algorithm. At the beginning of the algorithm, *initSol* is copied into *bestSol*. Next, an iterative procedure starts in order to create new solutions. Resources are randomly assigned. Then, in order to generate a new solution (*newSol*), the NEH heuristic is run using a biased-randomization strategy as proposed in Grasas et al. (2017). This

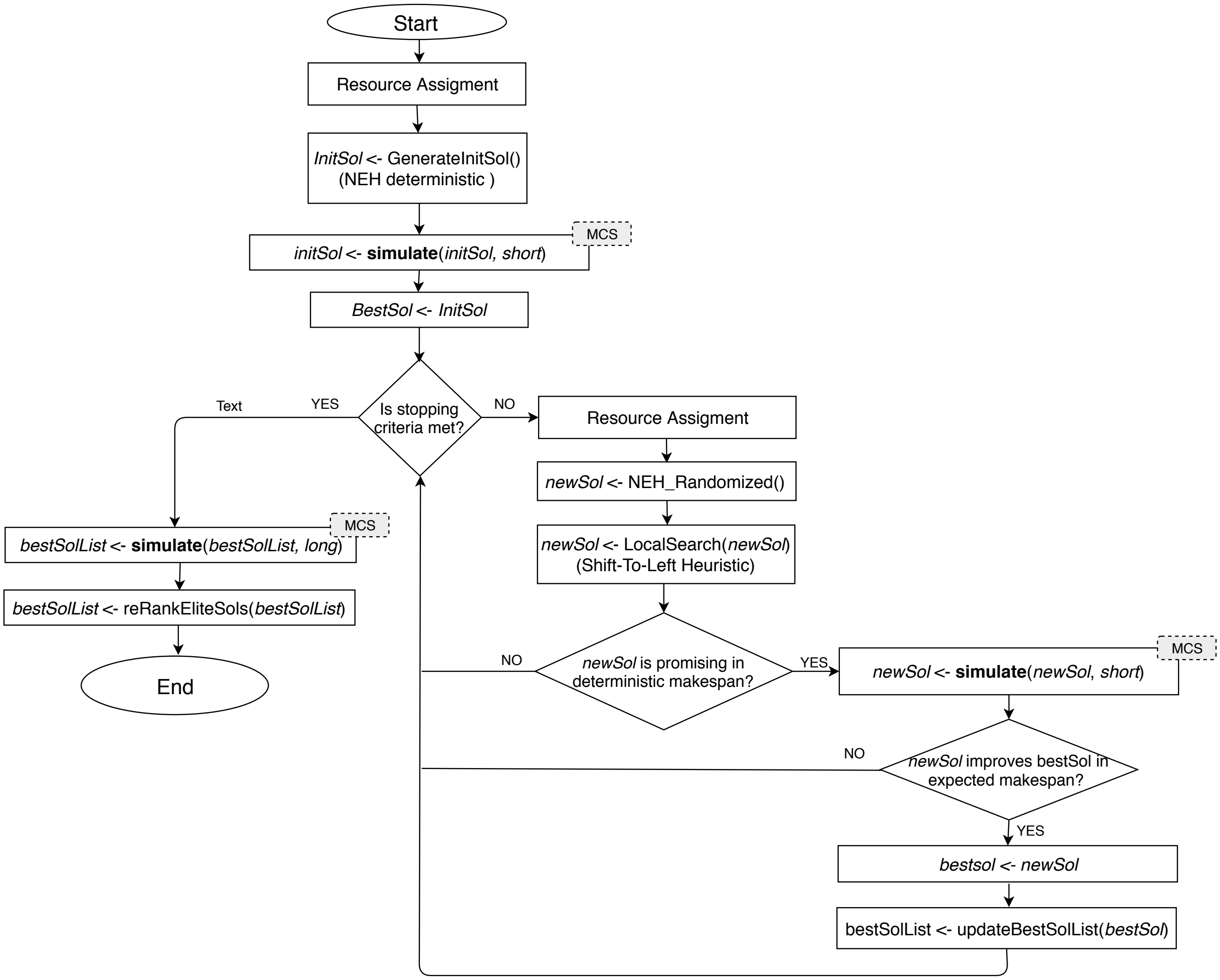
Start
Resource Assigment
InitSol <- GenerateInitSol()
(NEH deterministic )
initSol <- simulate(initSol, short)
MCS
BestSol <- InitSol
Is stopping criteria met?
Text
YES
NO
Resource Assigment
newSol <- NEH_Randomized()
newSol <- LocalSearch(newSol)
(Shift-To-Left Heuristic)
bestSolList <- simulate(bestSolList, long)
MCS
bestSolList <- reRankEliteSols(bestSolList)
End
NO
newSol is promising in deterministic makespan?
YES
newSol <- simulate(newSol, short)
MCS
NO
newSol improves bestSol in expected makespan?
YES
bestsol <- newSol
bestSolList <- updateBestSolList(bestSol)

Figure 3: The proposed simheuristic algorithm.

strategy relies on a skewed probability distribution to assign decreasing probabilities to each possible move of the constructive heuristic. In this case, a Geometric probability distribution with a single parameter $\beta$ $(0 < \beta < 1)$ is employed. A detailed explanation of this strategy, together with successful application examples, can be found in Dominguez et al. (2014), Juan et al. (2015) or Dominguez et al. (2016).

Next, *newSol* undergoes a local search phase in order to find a local minimum. In the local search, the *shift-to-left* operator proposed in Juan et al. (2014) is utilized. The main idea behind this operator is to iteratively examine all the jobs and try to shift them to the left. If the job movement improves the makespan, the movement is accepted, otherwise the solution is not modified. Once the local search returns a *newSol*, if this new solution is promising in terms of deterministic makespan, then it is processed by a 'fast' (200 runs) Monte Carlo simulation in order to estimate the expected makespan. We apply a reduced number of runs to avoid the simulation jeopardizes the metaheuristic time. Whenever the stochastic solution outperforms the *bestSol*, this solution is updated to *newSol*, the new *bestSol* is saved in the pool of the best solutions, and the process continues until it reaches the maximum execution time. In our experiments, this maximum execution time has been defined as follows:

$$maxTime = n \cdot m \cdot t \tag{3}$$

where $n$ is the number of jobs, $m$ is the number of machines, and $t$ is a time factor (initially set to 0.03 seconds). When the GRASP stage ends, the algorithm returns a selected list with the best-found solutions (in our experiments, we have selected the best five ones). For each of these solutions, we perform a more intensive simulation, consisting of $1,000$ runs. This longer simulation provides a more accurate estimate of the expected makespan. Notice that the outcomes of this simulation can also be used to complete a risk analysis on each proposed solution, as well as to obtain other relevant statistics, as discussed in Juan et al. (2014).

## 5 COMPUTATIONAL EXPERIMENTS

Our algorithm was implemented in Java and tested on a personal computer with an i7 Quad core working at 2.4 GHz and with 6 GB of RAM. For our computational experiments, the classical PFSP benchmark set proposed in Taillard (1993) was employed and extended to the integrated problem. The original set consists of 12 sets of 10 instances each one, ranging from 20 to 500 jobs to be completed on 5 to 20 machines. As these instances are deterministic, we extended them considering Log-Normal distributions with $E[P_{ij}] = p_{ij}$. In order to compare our results, we selected the 10 instances used in Ferone et al. (2016) and adapted their costs assuming that one resource is allocated to each machine. Each instance was executed 10 times using different seeds. The Log-Normal variance used in our experiments was given by $Var[P_{ij}] = k \cdot E[P_{ij}]$, being $k$ an experimental design parameter that influences the level of variability. In our case, we used three different variance levels: $k \in \{5, 10, 20\}$. Notice that, in a real-life application, the specific value of this parameter would be adjusted based on historical observations, but the solving methodology would remain the same.

Table 1 shows the best solutions found over the 10 different runs of the algorithm, one for each different seed. The first column reports the instance name. The second column shows the cost summarized in Ferone et al. (2016) plus the resource cost assuming that each machine uses one resource. These solutions are used as a reference to compare them with our best solutions. The next two columns (third and fourth), report the best deterministic value found by our algorithm and the percentage gap with respect to the previous results, respectively. Finally, the last three columns report the best stochastic cost for each variance level.

First, the results produced by our algorithm for the deterministic case have been compared with the solutions provided in Ferone et al. (2016) (Column 2). Looking at Column 4, it can be observed that adding more resources to machines tends to reduce the total cost of the best solution. However, for other instances, the best solution is not improved, obtaining the same solutions as in Column 1 (where a single resource to each machine is assigned). This behavior seems to point out that in the latter instances the makespan savings associated with increasing resources do not compensate their marginal cost.

Table 1: Results for 10 Taillard instances with different variability levels

| **Instance** | **Ferone et al. (2016)** | **Deterministic value** | **%-Gap(2-3)** | **Stochastic values** | | |
|---:|---:|---:|---:|---:|---:|---:|
| **[1]** | **[2]** | **[3]** | **[4]** | **[5](k=5)** | **[6](k=10)** | **[7](k=20)** |
| *tai097* | 12882.00 | 12061.06 | -6.37 | 12181.46 | 12108.56 | 12476.26 |
| *tai102* | 15425.00 | 15425.00 | 0.00 | 17321.55 | 17370.55 | 17465.48 |
| *tai103* | 15513.00 | 15513.00 | 0.00 | 17394.32 | 17468.20 | 17215.68 |
| *tai104* | 15470.00 | 15470.00 | 0.00 | 17252.67 | 17270.40 | 16603.26 |
| *tai105* | 15380.00 | 15380.00 | 0.00 | 17036.50 | 17215.58 | 17047.88 |
| *tai107* | 15517.00 | 15517.00 | 0.00 | 17360.07 | 17341.70 | 17290.16 |
| *tai108* | 15536.00 | 15536.00 | 0.00 | 16763.63 | 17029.10 | 16903.45 |
| *tai112* | 30810.00 | 29122.46 | -5.48 | 29169.32 | 32738.78 | 32588.69 |
| *tai113* | 30613.00 | 28852.54 | -5.75 | 32164.19 | 29117.31 | 32715.56 |
| *tai118* | 30767.00 | 29165.26 | -5.21 | 29227.71 | 29250.82 | 29334.08 |
| *Average* | 19791.30 | 19204.23 | -2.28 | 20587.14 | 20691.10 | 20964.05 |

In a similar way, our simheuristic approach is used in the stochastic scenario. On the average, the stochastic cost of the solutions increases with the variance level $k$. Notice that, for most instances (all except tai102 and tai107), the makespan can decrease as the variance level increases. To better study this behavior, we conducted another set of experiments on instances tai104 and tai113, where the makespan decrease was noticeable. In this new experiment we increased the time factor $t$ (from 0.03 to 3 seconds), as well as the number of runs (they have been increased to 500 and 2000 for short and long simulations, respectively). The results are reported in Table 2. The makespan is still decreasing for instance tai104, but now the magnitude of this reduction is small. For instance tai113 this decreasing effect has disappeared. Therefore, results in Table 1 are probably due to the small number of simulation runs performed in that initial experiment.

Table 2: Results for 2 Taillard instances with different variability levels and more computational times

| **Instance** | **Stochastic values** | | | **% gap** | |
|---:|---:|---:|---:|---:|---:|
| **[1]** | **[2](k=5)** | **[3](k=10)** | **[4](k=20)** | **[3-2]** | **[4-3]** |
| tai104 | 16403.94 | 16342.74 | 16344.01 | -0.37 | 0.01 |
| tai113 | 27779.78 | 27832.76 | 27835.63 | 0.19 | 0.01 |

Figure 4 reports the boxplot of the percentage gaps between the best stochastic solution and the best deterministic solution found at each run. It seems clear that the expected cost increases as the variance raises, inducing larger gaps. This effect is what one should expect, which adds credibility to the results obtained by the algorithm. It also illustrates that using the best deterministic solution in a stochastic environment might be a bad decision, since it usually provides sub-optimal values.

## 6 FURTHER POSSIBLE APPLICATIONS

The simheuristic algorithm introduced here can be easily extended to address a range of similar integrated problems combining allocation with scheduling under uncertainty scenarios. The results reported in this paper promise an easy-to-implement simulation-optimization approach with only a few number of parameters, which is able to efficiently consider different types of variables under uncertainty in low computing times.

Thus, further developments involve applications to other integrated resource-allocation and activities-scheduling problems. For resource allocation, two possible extensions are envisaged: *(i)* considering a

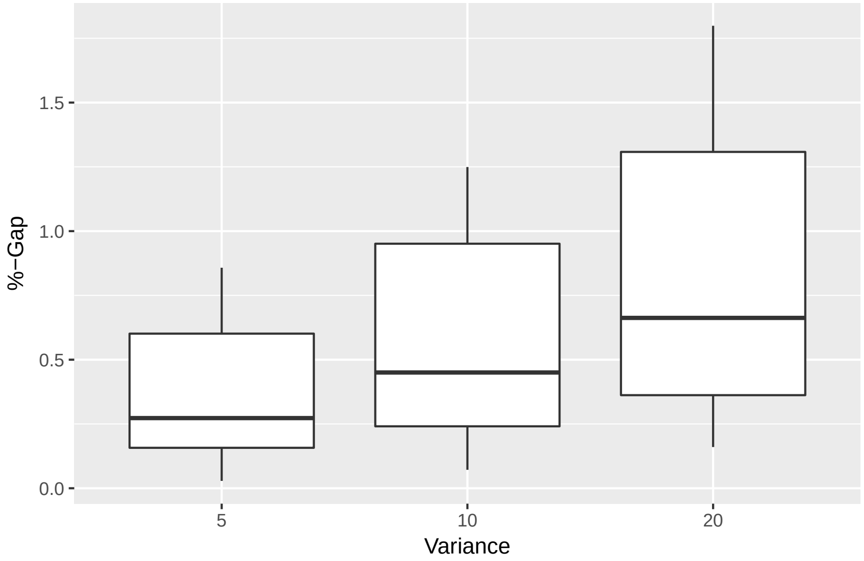

Figure 4: Comparison of gaps between stochastic and deterministic solutions.

heterogeneous set of assignable resources; and *(ii)* admitting the planning of continuous capacities of resources, as it happens in the case of energy consumption. Regarding the scheduling of activities, further experiments could apply procedures similar to those presented in this paper to different production systems, such as flow-shop and job-shop environments.

## 7 CONCLUSIONS

In this paper, we have discussed the importance of considering uncertainty in realistic combinatorial optimization problems, and have proposed the use of simheuristics (combination of simulation with metaheuristics) as one of the most natural ways to deal with complex, large-scale, and stochastic combinatorial optimization problems that are frequently encountered in real-life applications of manufacturing logistics activities.

In particular, the paper proposes the use of a simulation-optimization approach to solve a complex integrated allocation and scheduling problem with stochastic processing times. The algorithm has been tested over a set of stochastic instances based on the classical ones for a deterministic and simplified version of the flow-shop problem. The results show that our algorithm is able to provide good solutions in very short computing times. Also, the computational experiments show how the expected makespan associated with each instance grows as the variability of the random processing times is increased.

Notice that our application provides a simple example of how allocation of resources and scheduling could be integrated in a unique decision process to find a better solution in terms of system performance. Of course, this must be interpreted as a first step before a complete extension to real-life applications. Indeed, the main scope of our work is not only to demonstrate the effectiveness of our approach, but also to give an idea of how managers can adopt similar methods to address their specific problems in an integrated way and how that can improve operations efficiency. Other research lines that we plan to explore in the future include the addition of new stochastic variables, such as machine reliability or availability of human resources.

## ACKNOWLEDGEMENTS

This work has been partially supported by the Spanish Ministry of Economy and Competitiveness (TRA2015-71883-REDT), the Erasmus+ program (2018-1-ES01-KA103-049767), and ACT Operations Research, a spin-off company at the Sapienza University of Rome.

## REFERENCES

Azaron, A., H. Katagiri, M. Sakawa, K. Kato, and A. Memariani. 2006. "A multi-objective resource allocation problem in PERT networks". *European Journal of Operational Research* 172 (3): 838 – 854.

Azaron, A., and R. Tavakkoli-Moghaddam. 2007. "Multi-objective timecost trade-off in dynamic PERT networks using an interactive approach". *European Journal of Operational Research* 180 (3): 1186 – 1200.

Baker, B. M. 1997. "Cost/time trade-off analysis for the critical path method: a derivation of the network flow approach". *Journal of the Operational Research Society* 48 (12): 1241–1244.

Bregman, R. L. 2009. "A heuristic procedure for solving the dynamic probabilistic project expediting problem". *European Journal of Operational Research* 192 (1): 125 – 137.

Cohen, I., B. Golany, and A. Shtub. 2007. "The stochastic timecost tradeoff problem: A robust optimization approach". *Networks* 49 (2): 175–188.

Deckro, R., J. Hebert, and W. Verdini. 1992. "Project scheduling with work packages". *Omega* 20 (2): 169 – 182.

Demeulemeester, E. L., and W. Herroelen. 2002. *Project Scheduling : a Research Handbook*. Boston, USA: Kluwer Academic Publishers.

Dominguez, O., A. A. Juan, B. Barrios, J. Faulin, and A. Agustin. 2016. "Using biased randomization for solving the two-dimensional loading vehicle routing problem with heterogeneous fleet". *Annals of Operations Research* 236 (2): 383–404.

Dominguez, O., A. A. Juan, and J. Faulin. 2014. "A biased-randomized algorithm for the two-dimensional vehicle routing problem with and without item rotations". *International Transactions in Operational Research* 21 (3): 375–398.

Elmaghraby, S. 1977. *Activity Networks: Project Planning and Control by Network Models*. A Wiley-Interscience publication. Wiley.

Elmaghraby, S., and C. Morgan. 2007. "Resource Allocation in Activity Networks under Stochastic Conditions. A Geometric Programming-Sample Path Optimization Approach". *Review of Business and Economic Literature* 0 (3): 367–390.

Feo, T. A., and M. G. Resende. 1995. "Greedy randomized adaptive search procedures". *Journal of global optimization* 6 (2): 109–133.

Ferone, D., P. Festa, A. Gruler, and A. A. Juan. 2016. "Combining simulation with a GRASP metaheuristic for solving the permutation flow-shop problem with stochastic processing times". In *Proceedings of the 2016 Winter Simulation Conference*, edited by T. M. K. Roeder et al., 2205–2215. Piscataway, New Jersey: IEEE Press.

Festa, P., and M. G. C. Resende. 2009a. "An annotated bibliography of GRASP - Part I: Algorithms". *International Transactions in Operational Research* 16 (1): 1–24.

Festa, P., and M. G. C. Resende. 2009b. "An annotated bibliography of GRASP - Part II: Applications". *International Transactions in Operational Research* 16 (2): 131–172.

Godinho, P., and F. G. Branco. 2012. "Adaptive policies for multi-mode project scheduling under uncertainty". *European Journal of Operational Research* 216 (3): 553 – 562.

Goh, J., and N. G. Hall. 2013. "Total Cost Control in Project Management via Satisficing". *Management Science* 59 (6): 1354–1372.

Gonzalez-Neira, E. M., D. Ferone, S. Hatami, and A. A. Juan. 2017. “A biased-randomized simheuristic for the distributed assembly permutation flowshop problem with stochastic processing times”. *Simulation Modelling Practice and Theory* 79:23–36.

Grasas, A., A. A. Juan, J. Faulin, J. de Armas, and H. Ramalhinho. 2017. “Biased Randomization of Heuristics using Skewed Probability Distributions: a survey and some applications”. *Computers & Industrial Engineering* 110:216–228.

Grasas, A., A. A. Juan, and H. R. Lourenço. 2016. “SimILS: a simulation-based extension of the iterated local search metaheuristic for stochastic combinatorial optimization”. *Journal of Simulation* 10 (1): 69–77.

Herroelen, W., and R. Leus. 2005. “Project scheduling under uncertainty: Survey and research potentials”. *European Journal of Operational Research* 165 (2): 289 – 306.

Juan, A. A., B. B. Barrios, E. Vallada, D. Riera, and J. Jorba. 2014. “A simheuristic algorithm for solving the permutation flow shop problem with stochastic processing times”. *Simulation Modelling Practice and Theory* 46:101–117.

Juan, A. A., J. Faulin, S. E. Grasman, M. Rabe, and G. Figueira. 2015. “A review of simheuristics: Extending metaheuristics to deal with stochastic combinatorial optimization problems”. *Operations Research Perspectives* 2:62–72.

Juan, A. A., H. R. Lourenço, M. Mateo, R. Luo, and Q. Castella. 2014. “Using iterated local search for solving the flow-shop problem: Parallelization, parametrization, and randomization issues”. *International Transactions in Operational Research* 21 (1): 103–126.

Juan, A. A., I. Pascual, D. Guimarans, and B. Barrios. 2015. “Combining biased randomization with iterated local search for solving the multidepot vehicle routing problem”. *International Transactions in Operational Research* 22 (4): 647–667.

Kang, C., and B.-C. Choi. 2015. “An adaptive crashing policy for stochastic time-cost tradeoff problems”. *Computers & Operations Research* 63:1–6.

Kelley, J. E. 1961. “Critical-path planning and scheduling: Mathematical basis”. *Operations research* 9 (3): 296–320.

Kimms, A. 2001. *Mathematical Programming and Financial Objectives for Scheduling Projects*. Boston, USA: Kluwer Academic Publishers.

Klerides, E., and E. Hadjiconstantinou. 2010. “A decomposition-based stochastic programming approach for the project scheduling problem under time/cost trade-off settings and uncertain durations”. *Computers & Operations Research* 37 (12): 2131–2140.

Kolisch, R., and R. Padman. 2001. “An integrated survey of deterministic project scheduling”. *Omega* 29 (3): 249–272.

Laslo, Z. 2003. “Activity time–cost tradeoffs under time and cost chance constraints”. *Computers & Industrial Engineering* 44 (3): 365–384.

Moder, J., C. Phillips, and E. Davis. 1983. *Project Management with CPM, PERT, and Precedence Diagramming*. Blitz Publishing Company.

Mokhtari, H., A. Aghaie, J. Rahimi, and A. Mozdgir. 2010. “Project time–cost trade-off scheduling: a hybrid optimization approach”. *The International Journal of Advanced Manufacturing Technology* 50 (5): 811–822.

Nawaz, M., E. E. Enscore Jr, and I. Ham. 1983. “A heuristic algorithm for the m-machine, n-job flow-shop sequencing problem”. *Omega* 11 (1): 91–95.

Taillard, E. 1993. “Benchmarks for basic scheduling problems”. *European Journal of Operational Research* 64:278–285.

Van Slyke, R. M. 1963. “Monte Carlo Methods and the PERT Problem”. *Operations Research* 11 (5): 839–860.

## AUTHOR BIOGRAPHIES

**LUDOVICA MACCARRONE** is a PhD student in Operations Research at the Department of Computer, Control and Management Engineering of Sapienza University of Rome. She holds a M.S. in Management Engineering. Her reaserch area involves simulation, black-box optimization and scheduling. Her email address is ludovica.maccarrone@uniroma1.it.

**DANIELE FERONE** is a post-doctoral researcher in the ICSO group at the IN3 – Open University of Catalonia. His main research interests include network optimization, constrained shortest path problems, and logistics optimization. His email address is danieleferone@gmail.com.

**TOMMASO GIOVANNELLI** is a Ph.D. student in Operations Research at the Department of Computer, Control and Management Engineering of Sapienza University of Rome. He holds a M.S. in Management Engineering. His current research interests include derivative-free optimization and stochastic optimization. His e-mail address is tommaso.giovannelli@uniroma1.it.

**JAVIER PANADERO** is an Assistant Professor of Simulation and High Performance Computing in the Computer Science, Multimedia and Telecommunication Department at the Open University of Catalonia. He is also a post-doctoral researcher at the ICSO@IN3 group. He holds a Ph.D. and a M.S. in Computer Science. His major research areas are: high performance computing, modeling and analysis of parallel applications, and simheuristics. He has co-authored more than 15 articles published in journals and conference proceedings. His e-mail address is jpanaderom@uoc.edu.

**ANGEL A. JUAN** is Professor of Operations Research & Industrial Engineering in the Computer Science Dept. at the Universitat Oberta de Catalunya (Spain). He is also the coordinator of the ICSO research group at the IN3. Dr. Juan holds a Ph.D. in Industrial Engineering and an M.S. in Mathematics. He completed a predoctoral internship at Harvard University and postdoctoral internships at Massachusetts Institute of Technology and Georgia Institute of Technology. His main research interests include applied optimization and simulation (metaheuristics and simheuristics) in computational transportation & logistics, Internet computing, and computational finance. He has published 65+ articles in JCR-indexed journals and 165+ documents indexed in Scopus. His website address is http://ajuanp.wordpress.com and his email address is ajuanp@uoc.edu.