Proceedings of the 2016 Winter Simulation Conference
T. M. K. Roeder, P. I. Frazier, R. Szechtman, E. Zhou, T. Huschka, and S. E. Chick, eds.

COMBINING SIMULATION WITH A GRASP METAHEURISTIC FOR SOLVING THE PERMUTATION FLOW-SHOP PROBLEM WITH STOCHASTIC PROCESSING TIMES

Daniele Ferone
Paola Festa

Dept. of Mathematics and Applications
University of Napoli FEDERICO II
Via Cintia
80126 Napoli, ITALY

Aljoscha Gruler
Angel A. Juan

Dept. of Computer Science
Open University of Catalonia – IN3
Av. Carl Friedrich Gauss 5
08860 Castelldefels, SPAIN

ABSTRACT

Greedy Randomized Adaptive Search Procedures (GRASP) are among the most popular metaheuristics for the solution of combinatorial optimization problems. While GRASP is a relatively simple and efficient framework to deal with deterministic problem settings, many real-life applications experience a high level of uncertainty concerning their input variables or even their optimization constraints. When properly combined with the right metaheuristic, simulation (in any of its variants) can be an effective way to cope with this uncertainty. In this paper, we present a simheuristic algorithm that integrates Monte Carlo simulation into a GRASP framework to solve the permutation flow shop problem (PFSP) with random processing times. The PFSP is a well-known problem in the supply chain management literature, but most of the existing work considers that processing times of tasks in machines are deterministic and known in advance, which in some real-life applications (e.g., project management) is an unrealistic assumption.

1 INTRODUCTION

A large number of decision-making problems in logistics, transportation, and supply chain management can be modeled as combinatorial optimization problems (COPs). Due to the complexity of modern systems and processes, most of these COPs are NP-hard in nature, which means that they need to be addressed with the help of metaheuristics, especially for medium- and large-scale instances. One of the most popular metaheuristics for solving COPs is the Greedy Randomized Adaptive Search Procedure (GRASP).

GRASP is a multi-start algorithm based on a randomized construction process combined with a local search (Resende and Ribeiro 2003). A feasible solution is constructed at each algorithm-iteration, before the neighborhood is investigated to find a local minimum. As a well-known optimization algorithm, the metaheuristic has been successfully applied to a wide range of optimization problems (Festa and Resende 2009). Like other similar metaheuristics, GRASP constitutes a relatively simple and efficient algorithmic framework to deal with deterministic problem settings. However, many real-life applications experience a high level of uncertainty and, therefore, solving deterministic optimization models –as it is usually done in the optimization literature– is an unrealistic simplification of the real-life system. As depicted in Figure 1, when properly combined with the right metaheuristic, simulation (in any of its variants) can be an effective way to cope with this uncertainty. Therefore, simheuristics –hybridization of simulation with metaheuristics– establishes a natural way of addressing realistic and large-scale COPs under uncertainty scenarios (Juan et al. 2015).

In this paper, we present a simheuristic algorithm that integrates Monte Carlo simulation into a GRASP framework to solve the permutation flow shop problem (PFSP) with random processing times. The choice of GRASP is related to its important feature of efficiently allowing the balance between randomness and

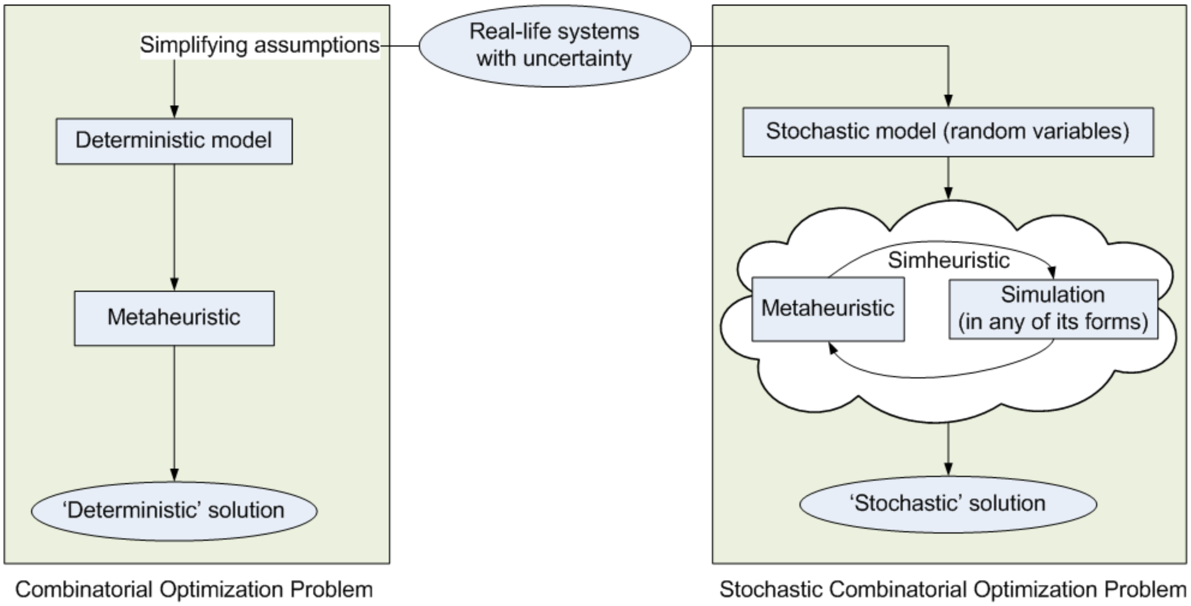

Figure 1: Hybridizing metaheuristics with simulation to solve stochastic COPs.

greediness during the construction phase of a feasible solution. The PFSP is a well-known problem in the supply chain management literature, but most of the existing work considers that processing times of tasks in machines are deterministic and known in advance, which in some real-life applications (e.g., project management) is an unrealistic assumption. Thus, by integrating simulation into the general GRASP framework, our approach allows the consideration of the stochastic components in realistic variants of the PFSP.

This paper is structured as follows: Section 2 reviews solution approaches (with special focus on simheuristics) to stochastic COPs. Section 3 describes the permutation flow shop problem with stochastic processing times. The main ideas behind our simulation-optimization approach are outlined in Section 4. A Java implementation of the proposed algorithm is tested in Section 5 using some benchmarks. Further possible application areas related to transportation and logistics are discussed in Section 6. Finally, Section 7 concludes this work.

2 SOLVING STOCHASTIC COMBINATORIAL OPTIMIZATION PROBLEMS

New solution methods, enhanced computational possibilities, and the high practical relevance of many optimization problems has led to an increased interest in solving real-life problem settings. Many practical problem settings are becoming more and more complex and dynamic, as different input variables experience some kind of uncertainty. Usually this is represented by including uncertain, stochastic, and dynamic information in the mathematical model that describes the real-life system (Bianchi et al. 2009). This paper focuses on *a priori* stochastic optimization. Unlike dynamic or reactive problem settings –in which input data is revealed while a established plan is executed and then changes are made on the initial plan–, *a priori* optimization assumes that some information (e.g., based on historical data) about the stochastic variable is already available during the planning phase. That is, random variables have been modeled via a probability distribution which can be used during the optimization process to establish solutions with a good expected performance (e.g., solutions aimed at minimizing total expected cost, etc.). Additionally, it might be desirable that these solutions show some 'robustness' or low risk level, i.e., that they are not likely to require significant or expensive adjustments during the execution process (Ritzinger, Puchinger, and Hartl 2016).

Different stochastic COPs have been addressed in the literature. Applications include routing problems with stochastic demands and/or travel times (Juan et al. 2011), stochastic scheduling problems (Juan et al. 2014a), and other problem settings in which some kind of uncertainty has to be represented, e.g., in the knapsack- or set-covering problem (Grasas, Juan, and Lourenço 2016, Sahinidis 2004).

2.1 Exact and metaheuristic methods to solve stochastic COPs

In the case of NP-hard COPs (as the one considered in this paper), exact solution approaches can only be applied to solve small- and medium sized instances. Christiansen and Lysgaards (2007) solve the vehicle routing problem (VRP) with stochastic demands by using a branch-and-price algorithm. A similar approach combined with column generation is used by Tas et al. (2014) to address the VRP with stochastic travel times. The arc routing problem (ARP) with stochastic demands is formulated as a set partitioning problem and solved with a branch-and-price algorithm in Christiansen et al (2009). The main advantage of applying exact methods to optimization problems is that they can guarantee the optimality of the obtained solution. However, these methods tend to be computationally expensive, making them mainly applicable to small sized problems or as building blocks for other approximate methods (Dumitrescu and Stützle 2010).

For larger problem instances, optimal solutions cannot be calculated efficiently in practice. Therefor optimality is usually traded for efficiency by applying metaheuristic methods, which can obtain near-optimal (but not necessarily optimal) solutions in reasonable computing times. Bianchi et al. (2006) employs and compares five different metaheuristics (simulated annealing, tabu search, iterated local search, ant colony optimization, and evolutionary algorithms) for solving the VRP with stochastic demands. A more general overview over different metaheuristics to solve stochastic COPs is given by Bianchi et al. (2009) and van Hentenryck and Bent (2010).

2.2 Solving stochastic COPs through simheuristics

Simheuristics allow the consideration of stochastic objective functions and constraints by including simulation in a metaheuristic-based framework (Juan et al. 2015). In its most basic form, it works as follows: Given a stochastic problem setting, the random variables are transformed into their deterministic counterpart by considering expected values, which can be solved using efficient metaheuristics for deterministic COPs. In the following, the constructed *a priori* solution is evaluated in a stochastic scenario by running several simulation runs in which the stochastic variable is depicted from a given (theoretical or empirical) probability distribution. The results from these simulation runs provide feedback to the metaheuristic itself, so that the search process is better guided.

Some successful implementations of simheuristic approaches to different COPs under uncertainty have been presented in recent years. The effect of using safety stocks in VRPs with stochastic demands is discussed by Juan et al. (2011). Gonzalez et al. (2012) address the ARP with stochastic demands by combining a randomized routing algorithm with Monte Carlo simulation. Cabrera et al. (2014) combine discrete-event simulation with metaheuristics to address the problem of minimizing service deployment cost over non-dedicated computer resources to deploy Internet services in large scale systems. Furthermore, simheuristics have been applied to tackle the inventory routing problem with stochastic demands (Juan et al. 2014b), scheduling problems with random processing times (Juan et al. 2014a), and dynamic home-service routing problems with synchronized ride-sharing (Fikar et al. 2016).

3 THE STOCHASTIC PERMUTATION FLOW SHOP PROBLEM

The classical PFSP is a well-known COP that can be described as follows: a set J of n jobs has to be processed by a set M of m machines. Each job $i \in J$ is composed by an ordered set of m operations, O_{ij}, that must be sequentially performed by the m machines (one operation per machine). 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. Each operation O_{ij} requires a processing time, p_{ij}, which is assumed to be known in advance. The goal is to find a sequence (permutation) of jobs so that a given criterion is optimized (Ruiz and Maroto 2005). The most commonly and studied criterion is the minimization of the completion time or makespan, i.e., the time it requires to process all the jobs throughout all the machines (Juan et al. 2014c). Other goals can also be considered, e.g., the minimization of the total processing time –i.e., the sum of the individual processing times of each job in each machine–, or the total tardiness of jobs with respect to their scheduled deadlines (Vallada, Ruiz, and Minella 2008).

Figure 2 illustrates a simple PFSP with three jobs and three machines, where O_{ij} represents the operation of job i in machine j $(1 \leq i \leq 3, 1 \leq j \leq 3)$. Notice that, for a given permutation of jobs, even a single change in the processing time of one job in one machine (O_{21} and O_{22} in the Figure, respectively) can have a noticeable impact on the value of the final makespan. Notice also that, since processing times are random variables, the makespan associated with a given solution will be also a random variable.

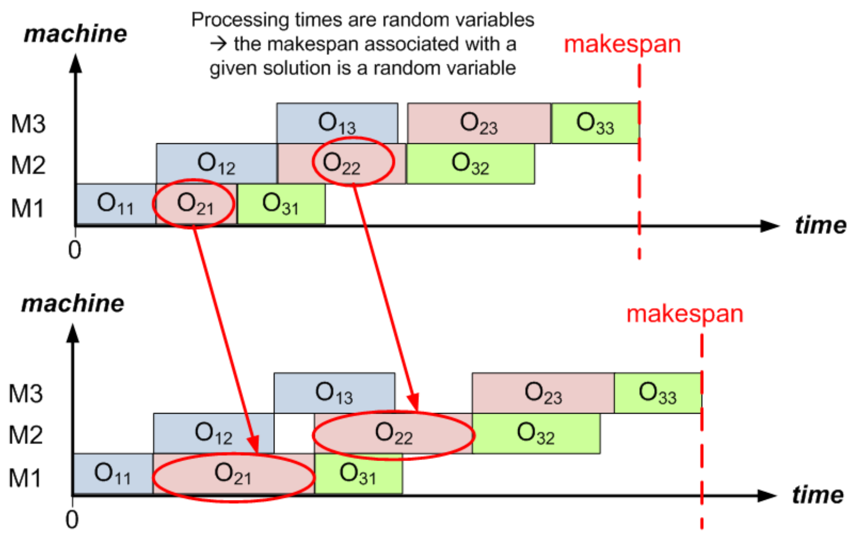

Figure 2: The permutation flow shop problem with random processing times.

The PFSP with stochastic processing times (PFSPST) can be seen as a generalization of the PFSP in which the processing time of each job i in each machine j is not a constant value. Instead, it is a random variable, P_{ij}, following a non-negative probability distribution such as the log-normal, Weibull, etc. Since uncertainty is present in most real-life processes and systems, considering random processing times represents a more realistic scenario than simply considering deterministic times, especially whenever the so-called 'human factor' is involved in the process, e.g., project management. Therefore, one goal that can be considered when dealing with the PFSPST is to determine a sequence (permutation) of jobs that minimizes the expected makespan or mean time to completion of all jobs.

As with other COPs, a number of different approaches and methodologies have been developed to deal with the (deterministic) PFSP. These approaches range from exact optimization methods to heuristics and metaheuristics. However, the situation with the PFSPST is different. To the best of our knowledge, there is a lack of methods able to provide high-quality solutions to the stochastic version of the PFSP. Most of the existing approaches are quite theoretical and require many assumptions on the probability distributions that model job processing times, while other approaches seem to be valid only for small-size instances. Thus, for instance, in the articles by Gourgand et al. (2005), and Baker and Altheimer (2012), simulation-based techniques have been used to get results for the PFSPST. However, simulation is mainly used as a backup method to validate the results generated by other analytical methods. Consequently, only normal or exponential probability distributions were employed to model processing times. Moreover, these papers made strong assumptions on the size of the instances being analyzed. More recently, Juan et al. (2014a) proposed a simheuristic approach, based on the combination of simulation with an iterated local search metaheuristic, to efficiently deal with the PFSPST. Our work extends their paper by hybridizing simulation with a GRASP metaheuristic framework and then compares both approaches using a common set of benchmarks.

4 OVERVIEW OF OUR SIMHEURISTIC APPROACH

GRASP is based on a multi-start process with two main steps at each iteration: the solution construction and the following local search procedure (Figure 1, left). During the construction phase, a feasible solution is iteratively constructed by adding solution elements one at a time. 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 $g : C \rightarrow \mathbb{R}$. This function measures the (myopic) benefit of including an element in the currently constructed solution. Hereby, the benefits are typically recalculated at each iteration of the construction phase. The probabilistic component of GRASP consists of randomly choosing one of the most promising candidates, but not necessarily the top one. That is, the most promising elements (according to the associated benefits of including it in the current solution) are stored in a Restricted Candidate List (RCL), from which the next element to be added to the solution is chosen. Notice that this technique tends to generate different solutions every time the multi-start procedure is run, which helps the algorithm to avoid getting trapped into a local minimum.

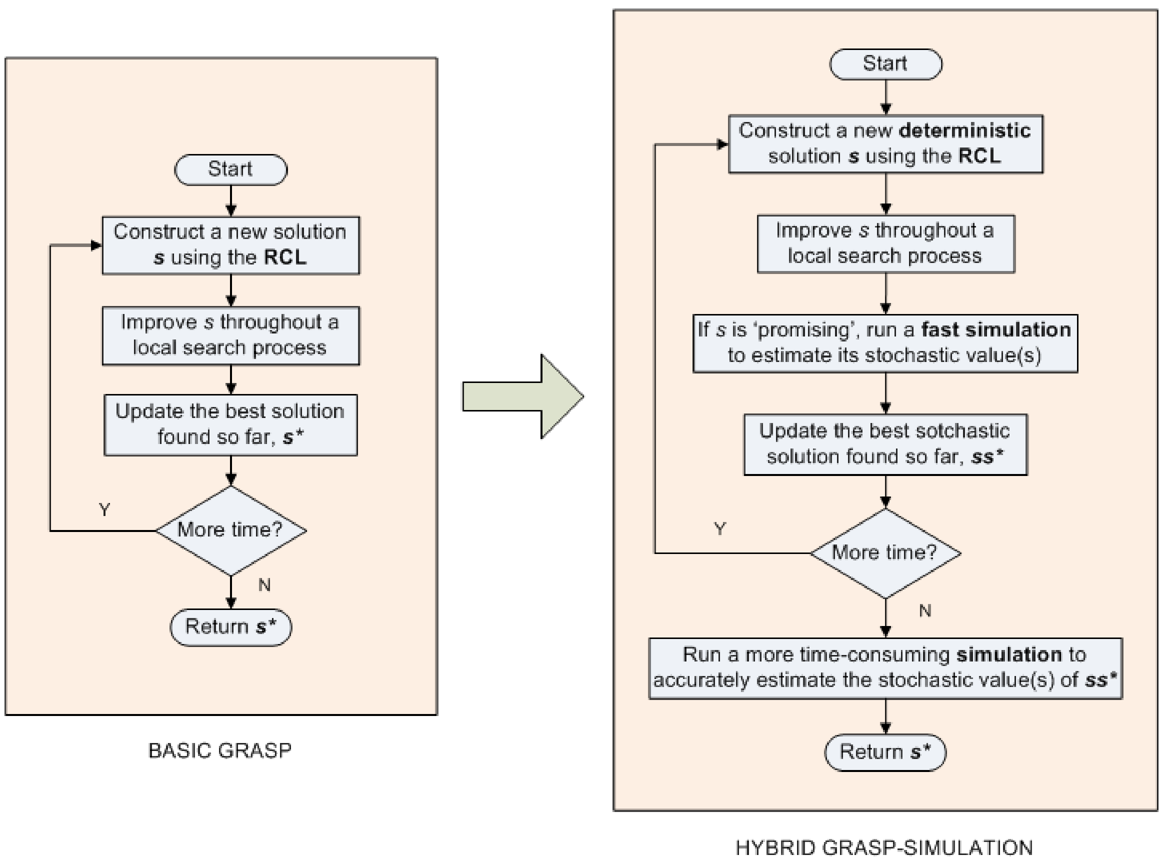

Figure 3: Basic GRASP vs. hybrid GRASP-simulation approach.

Once a feasible solution s has been constructed, it is locally improved with respect to a neighborhood definition. As the initial solution is not guaranteed to be optimal after the construction phase, a local neighborhood search typically allows for further solution improvements. The neighborhood structure N of problem P consists of a subset of solutions $N(s)$. If there is no better solution within $N(s)$, s is said to be locally optimal.

In its most basic form, the general framework of the hybrid GRASP-simulation approach is also outlined in Figure 3 (right). A stochastic COP with random variables is hereby transformed into its deterministic counterpart by considering expected values. The multi-start GRASP metaheuristic is started to create new feasible deterministic solutions. If the deterministic costs of new solutions improve the current best, the solution is considered promising and is simulated to evaluate its behavior with random variables. Accordingly, the current best deterministic and stochastic solutions are updated when necessary. Once the GRASP termination criterion (e.g., iterations or execution time) is met, the most promising solutions undergo a long simulation run to get a more detailed estimate of the expected objective function considering stochastic variables.

During the metaheuristic stage, we apply a low number of simulation iterations to avoid the simulation procedure jeopardizing the exploration of the solution space. While these initial simulations provide only rough estimates of the stochastic solutions, they are sufficient to identify 'promising' results under uncertainty. Once a reduced number of promising solutions has been identified, they are more closely explored through a more extensive simulation, giving a more accurate estimation.

5 COMPUTATIONAL EXPERIMENTS

5.1 Experimental settings

To test our approach, we base our experiments in a randomly selected subset of the benchmark instances provided by Taillard (1993). This benchmark consists of 12 sets of 10 instances each, ranging from 20 to 500 jobs to be completed on 5 to 20 machines. In particular, we randomly chose 30 instances of these benchmarks. The original processing times of the deterministic instances were used as expected values of the random processing times in the generalized instances, i.e., $E[P_{ij}] = p_{ij}$. Also, we assumed these times to follow a log-normal distribution. The constructive and local search procedures used during the generation of the deterministic solutions are described in Juan et al. (2014c).

In order to allow a comparison with the results of Juan et al. (2014a), we have used the same experimental settings. As termination criteria for the multi-start procedure, *maxTime* seconds is defined by the product of the number of jobs, n, the number of machines, m, and a time factor $t = 0.03$, such that: $maxTime = n * m * 0.03$. This leads to very short computation times, ranging from only a few seconds to a maximum of 5 minutes for the largest instances. Similar to the experiments proposed in Juan et al. (2014a), for each log-normal distribution the variance level is given by: $Var[P_{ij}] = k * E[P_{ij}]$, where k is a positive parameter representing different variance levels. In our case, several variance levels are considered: $k = 0.0$ (deterministic scenario), $k = 0.1$ (low-level variability), $k = 0.5$, $k = 2$, and $k = 5$ (high-level variability). For the 'fast' simulation we run $shortSimIter = 200$ iterations, while we use $longSimIter = 1,000$ iterations for the in-debt estimation of stochastic makespans. The low number of iterations for the fast simulation enables the evaluation of a large number of solutions in the stochastic scenario without too much computational effort. This has proven to be a good trade-off between solution quality and computational times.

The algorithm has been implemented as a Java application. It was run on Eclipse on an Ubuntu operating system with a personal computer with a Intel i7 Quad core, 2.67 GHz clock and 6 GB RAM. For each instance, 10 different seeds (for the pseudo-random number generator) have been used.

5.2 Analysis of results

The best solutions found after running our algorithm with 10 different random number seeds are listed in Table 1. Next to the instance name showing the considered number of jobs and machines, our best solution (BS) with a deterministic makespan (DM) and their stochastic counterparts (SM) with difference variability levels can be seen. As expected, the total makespan increases with increasing values of k. This effect is visualized in the boxplot shown in Figure 4, in which the percentage gaps between the best solutions found for each of stochastic scenario and the best solution found for the deterministic scenario over all instances are compared. It can be concluded that expected makespans increase with higher variances of processing times, leading to larger gaps between the lower-bound makespan (the one for the deterministic scenario) and the expected makespan in any of the stochastic scenarios. One important consequence can be derived from this result: the best solutions found for the deterministic scenario might perform poorly in stochastic scenarios with a high variability in the processing times, thus making the use of approaches such as the one presented in this paper necessary.

Table 1: Results for 30 Taillard instances with different variability levels.

Instance	**DM-BS** ($k=0$)	**SM-BS** ($k=0.1$)	**SM-BS** ($k=0.5$)	**SM-BS** ($k=2$)	**SM-BS** ($k=5$)
tai007_20_5	1234	1246.74	1248.49	1251.22	1258.7
tai009_20_5	1230	1247.65	1252.67	1260.9	1268.51
tai010_20_5	1108	1126.18	1128.91	1135.86	1142.21
tai011_20_10	1582	1600.33	1601.06	1618.14	1622.42
tai013_20_10	1496	1512.19	1518.8	1525.13	1537.46
tai027_20_20	2273	2291.72	2291.72	2302.74	2304.57
tai036_50_5	2829	2836.93	2845.71	2836.67	2855.87
tai040_50_5	2782	2782.62	2791.41	2797.03	2810.9
tai044_50_10	3064	3081.22	3088.16	3097.24	3106.82
tai045_50_10	3008	3041.1	3041.1	3072.05	3066.98
tai046_50_10	3023	3059.37	3059.37	3086.1	3083.53
tai047_50_10	3124	3164.59	3164.63	3188.04	3188.76
tai052_50_20	3764	3779.45	3779.45	3797.38	3798.52
tai055_50_20	3669	3693.03	3709.93	3711.72	3723.03
tai062_100_5	5268	5274.83	5277.37	5312.29	5314.92
tai067_100_5	5246	5283.76	5298.62	5309.86	5325.27
tai078_100_10	5640	5699.97	5699.66	5744.75	5741.79
tai082_100_20	6292	6316.63	6329.4	6335.38	6353.5
tai087_100_20	6389	6437.8	6441.48	6453.65	6453.65
tai094_200_10	10893	10904.24	10924.09	10946.27	10981.14
tai097_200_10	10882	10922.13	10915.82	10948.44	10997.43
tai102_200_20	11425	11505.69	11515.97	11565.6	11581.26
tai103_200_20	11513	11559.49	11560.82	11602.2	11636.95
tai104_200_20	11470	11486.6	11481.3	11514.21	11514.21
tai105_200_20	11380	11494.34	11486.91	11493.4	11493.4
tai107_200_20	11517	11591.14	11583.61	11656.62	11654.64
tai108_200_20	11536	11606.51	11611.28	11666.63	11643.94
tai112_500_20	26810	26930.58	26930.58	27022.15	27003.24
tai113_500_20	26613	26667.02	26702.72	26744	26748.28
tai118_500_20	26767	26892.89	26885.86	26998.93	26995.51

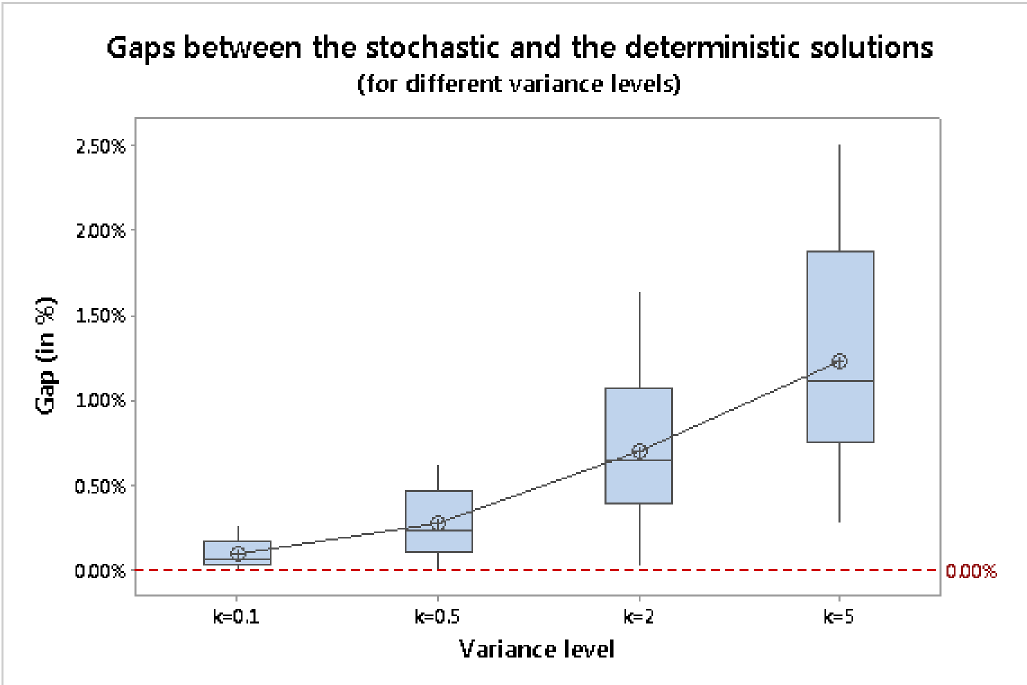

Figure 4: Comparison of gaps between stochastic and deterministic solutions.

6 FURTHER POSSIBLE APPLICATION AREAS IN TRANSPORTATION AND LOGISTICS

Our simheuristic approach is not limited to solve the PFSPST. Indeed, it is flexible enough to address a range of COPs under uncertainty in the field of transportation and logistics, often characterized by some level of randomness concerning customers, processing and/or travel times, as well as customer demands. As the stochastic solution quality obtained by the presented algorithm is directly related to the quality of the underlying metaheuristic, especially the application areas in which GRASP has been successfully implemented to deterministic COPs as outlined by Festa and Resende (2009) is thinkable. In particular scheduling and routing problems can be highlighted in this context. The results reported in this paper promise an easy-to-implement simheuristic approach with only few parameters, which is able to efficiently consider different types of variables under uncertainty in low computational times.

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 way to deal with complex, large-scale, and stochastic combinatorial optimization problems that are frequently encountered in real-life applications of logistics, transportation, and supply chain management activities.

In particular, the paper proposes the use of GRASP as the base metaheuristic, since it is a relatively-easy to implement, efficient, and well-tested framework with a number of applications in the aforementioned fields. To illustrate these ideas, we have chosen the permutation flow-shop problem with stochastic processing times and have developed a hybrid simulation-optimization algorithm to solve it. The algorithm has been tested over a set of stochastic instances based on the classical ones for the deterministic version of the 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 to each instance grows as the variability of the random processing times is increased.

ACKNOWLEDGEMENTS

This work has been partially supported by the Spanish Ministry of Economy and Competitiveness (TRA2013-48180-C3-P and TRA2015-71883-REDT), and FEDER. Likewise, we want to acknowledge the support received by the Department of Universities, Research & Information Society of the Catalan Government (2014-CTP-00001) and the doctoral grant of the UOC.

REFERENCES

Baker, K., and D. Altheimer. 2012. "Heuristic Procedures for the Stochastic Flow Shop Problem". *European Journal of Operational Research* 216:172–177.

Bianchi, L., M. Birattari, M. Chiarandini, M. Manfrin, M. Mastrolilli, L. Paquete, O. Rossi-Doria, and T. Schiavinotto. 2006. "Hybrid Metaheuristics for the Vehicle Routing Problem with Stochastic Demands". *Journal of Mathematical Modelling and Algorithms* 5 (1): 91–110.

Bianchi, L., M. Dorigo, L. M. Gambardella, and W. J. Gutjahr. 2009. "A Survey on Metaheuristics for Stochastic Combinatorial Optimization". *Natural Computing* 8 (2): 239–287.

Cabrera, G., A. A. Juan, D. Lázaro, J. M. Marquès, and I. Proskurnia. 2014. "A Simulation-Optimization Approach to Deploy Internet Services in Large-scale Systems with User-Provided Resources". *Simulation* 90 (6): 644–659.

Christiansen, C., and J. Lysgaard. 2007. "A Branch-and-Price Algorithm for the Capacitated Vehicle Routing Problem with Stochastic Demands". *Operations Research Letters* 35 (6): 773–781.

Christiansen, C., J. Lysgaard, and S. Wohlk. 2009. "A Branch-and-Price Algorithm For the Capacitated Arc Routing Problem with Stochastic Demands". *Operations Research Letters* 37 (6): 392–398.

Dumitrescu, I., and T. Stützle. 2010. "Usage of Exact Algorithms to Enhance Stochastic Local Search Algorithms". In *Matheuristics - Hybridizing Metaheuristics and Mathematical Programming*, edited by V. Maniezzo, T. Stützle, and S. Voß, 103–134. Springer US.

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

Fikar, C., A. A. Juan, E. Martinez, and P. Hirsch. 2016. "A Discrete-Event Metaheuristic for Dynamic Home-Service Routing with Synchronized Ride-Sharing". *European Journal of Industrial Engineering*.

González, S., D. Riera, A. Juan, M. Elizondo, and P. Fonseca. 2012. "Sim-RandSHARP: A Hybrid Algorithm for Solving the Arc Routing Problem with Stochastic Demands". In *Proceedings of the 2012 Winter Simulation Conference*, edited by C. Laroque, J. Himmelspach, R. Pasupathy, O. Rose, and A. Uhrmacher, 3123–3133. Piscataway, New Jersey: Institute of Electrical and Electronics Engineers, Inc.

Gourgand, M., N. Grangeon, and S. Norre.. 2005. "Markovian Analysis for Performance Evaluation and Scheduling in m Machine Stochastic Flow-shop with Buffers of any Capacity". *European Journal of Operational Research* 161:126–147.

Grasas, 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.

Juan, A. A., B. B. Barrios, E. Vallada, D. Riera, and J. Jorba. 2014a. "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. Grasman, D. Riera, J. Marull, and C. Mendez. 2011. "Using Safety Stocks and Simulation to Solve the Vehicle Routing Problem with Stochastic Demands". *Transportation Research Part C: Emerging Technologies* 19 (5): 751–765.

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., S. E. Grasman, J. Caceres-Cruz, and T. Bektaş. 2014b. "A Simheuristic Algorithm for the Single-Period Stochastic Inventory-Routing Problem with Stock-Outs". *Simulation Modelling Practice and Theory* 46:40–52.

Juan, A. A., H. Lourenço, M. Mateo, R. Luo, and Q. Castella. 2014c. "Using Iterated Local Search for Solving the Flow-Shop Problem: Parallelization, Parametrization, and Randomization Issues". *International Transactions in Operational Research* 21 (1): 103–126.

Resende, M. G. C., and C. C. Ribeiro. 2003. "Greedy Randomized Adaptive Search Procedures: Advances, Hybridizations, and Applications". In *Handbook of Metaheuristics*, edited by F. Glover and G. Kochenberger, 219–249. New York, USA: Springer.

Ritzinger, U., J. Puchinger, and R. F. Hartl. 2016. "A Survey on Dynamic and Stochastic Vehicle Routing Problems". *International Journal of Production Research* 54 (1): 215–231.

Ruiz, R., and C. Maroto. 2005. "A Comprehensive Review and Evaluation of Permutation Flowshop Heuristics". *European Journal of Operational Research* 165 (2): 479–494.

Sahinidis, N. V. 2004. "Optimization under Uncertainty: State-of-the-Art and Opportunities". *Computers and Chemical Engineering* 28:971–983.

Taş, D., M. Gendreau, N. Dellaert, T. van Woensel, and A. de Kok. 2014. "Vehicle Routing with Soft Time Windows and Stochastic Travel Times: A Column Generation and Branch-and-Price Solution Approach". *European Journal of Operational Research* 236 (3): 789–799.

Taillard, E. 1993. "Benchmarks for Basic Scheduling Problems". *European Journal of Operational Research* 64:278–285.

Vallada, E., R. Ruiz, and G. Minella. 2008. "Minimising Total Tardiness in the M-Machine Flowshop Problem: a Review and Evaluation of Heuristics and Metaheuristics". *Comput. Oper. Res.* 35:1350–1373.

van Hentenryck, P., and R. Bent. 2010. *Online Stochastic Combinatorial Optimization*. Boston, USA: The MIT Press.

AUTHOR BIOGRAPHIES

DANIELE FERONE is Ph.D. student in Operations Research at the University of Napoli FEDERICO II, Italy. He holds a Master Degree in Computer Science. His main research interests include network optimization and constrained shortest path problems. His email address is daniele.ferone@unina.it.

ALJOSCHA GRULER is Ph.D. candidate of Network & Information Technologies in the Internet Computing & Systems Optimization research group at the Open University of Catalonia (UOC) in Barcelona, Spain. Mr. Gruler holds a M.Eng. in Logistics and Supply Chain Management. His research interests include the application of simheuristic and randomized algorithms in the area of logistics & transportation. His email address is agruler@uoc.edu.

PAOLA FESTA obtained her Ph.D. degree in Operations Research in 2000 and is currently Associate Professor in Operations Research at the University of Napoli FEDERICO II (UNFII). She is in the scientific committee of the Ph.D. Programs in Operations Research (University of Calabria) and in Mathematics & Computer Science (UNFII). Since 1999, she has been frequently and regularly Research Scholar at several national and international research institutions, including CNRs, MIT Lab. for Information and Decision Systems (USA), AT&T Labs Research (USA), and Department of Industrial and Systems Engineering of the University of Florida (USA). She is Associate Editor of several international journals. She has been chair or member of the scientific committee of more than 50 international conferences and workshops. She is author/co-author of more than 100 publications. Her website address is http://www.dma.unina.it/festa.

ANGEL A. JUAN is Associate Professor of Optimization & Simulation in the Computer Science Dept. at the Open University of Catalonia (UOC). Dr. Juan holds a Ph.D. in Industrial Engineering and a

M.Sc. in Applied Mathematics. He completed a predoctoral internship at Harvard University and a postdoctoral internship at the MIT Center for Transportation & Logistics. He has been invited researcher at the University of Southampton (UK), at LAAS-CNRS (France), at the University of Natural Resources and Life Sciences (Austria), and at the University of Portsmouth (UK). His research interests include applications of Randomized Algorithms and Simheuristics in Logistics & Transportation, Production, and Internet Computing. He has published over 150 peer-reviewed papers regarding these fields. His website address is http://ajuanp.wordpress.com.