Why Simheuristics? Benefits, Limitations, and Best Practices when Combining Metaheuristics with Simulation

Manuel Chica[1,2], Angel A. Juan[3], Christopher Bayliss[3],

Oscar Cordon[1], W. David Kelton[4,5]

Abstract

Many decision-making processes in our society involve *NP-hard* optimization problems. The large-scale, dynamism, and uncertainty of these problems constraint the potential use of stand-alone optimization methods. The same applies for isolated simulation models, which do not have the potential to find optimal solutions in a combinatorial environment. This paper discusses the utilization of modeling and solving approaches based on the integration of simulation with metaheuristics. These *simheuristic* algorithms, which constitute a natural extension of both metaheuristics and simulation techniques, should be used as a 'first-resort' method when addressing large-scale and *NP-hard* optimization problems under uncertainty –which is a frequent case in real-life applications. We outline the benefits and limitations of simheuristic algorithms, provide numerical experiments that validate our arguments, review some recent publications, and outline the best practices to consider during their design and implementation stages.

[1]Andalusian Research Institute DaSCI, University of Granada, Granada, Spain

[2]School of Electrical Eng. and Computing, The University of Newcastle, Callaghan NSW, Australia

[3]IN3 – Computer Science Department, Universitat Oberta de Catalunya, Barcelona, Spain

[4]Department of Operations, BA, and Inf. Systems, Univ. of Cincinnati, Cincinnati, OH, USA

[5]Department of Operations Research, Naval Postgraduate School, Monterey, CA, USA

MSC: 68U20, 68T20, 90C59, 90C27.

Keywords: Simulation; Metaheuristics; Combinatorial Optimization; Simheuristics.

1. Introduction

Decision makers in areas such as transportation, logistics, supply-chain management, health care, production, telecommunication systems, and finance have to face complex challenges when tackling optimization problems in real-world applications. Most of these optimization problems are *NP-hard*, while others have a lack of complete information that makes their exact definition or formulation quite challenging if not impossible. These facts limit the use of exact optimization methods to small- and medium-sized instances, in which the optimal values can be obtained in reasonable computing times. Moreover, traditional optimization methods might require the use of simplifying assumptions, which do not always reflect the actual system characteristics in a proper manner. Driven by economic and technological factors, real-world systems are becoming increasingly large and complex. Among these factors, we could include trends such as globalization, increased computing power, information technologies, as well as the availability of vast amounts of data (Xu et al., 2015).

Metaheuristic algorithms have gained popularity as a predominant approach for solving real-world optimization problems (Dokeroglu et al., 2019). These algorithms are able to deal with non-trivial objective functions (e.g., multi-objective, non-convex, non-smooth, and noisy functions), soft constraints, and decision variables of different nature. Metaheuristics allow decision makers to obtain near-optimal solutions to large and complex problems in reasonably low computing times, sometimes even in 'real time' (e.g., a few seconds). Therefore, they have become effective methodologies in application areas where optimization of system resources is needed. In addition, approaches hybridizing exact methods with metaheuristics are also widely used. For instance, matheuristics

(Boschetti et al., 2009) combine both approaches to get the best from each of them. Typically, they employ the metaheuristic component to deal with the large global problem, while the exact component is used to cope with specific parts of it (Fischetti and Fischetti, 2018). Nonetheless, both exact optimization methods and metaheuristics frequently assume that the problem inputs, the underlying objective functions, and the set of optimization constraints are deterministic or follow simple probabilistic rules. These are strong assumptions and, as a consequence, many deterministic models are oversimplified versions of real-world systems. Coping with the inherent uncertainty of the systems to optimize during problem solving has recently gained relevance (Keith and Ahner, 2019). For instance, robust approaches for metaheuristics have been proposed to handle such uncertainty (Beyer and Sendhoff, 2007). Most of these approaches are extensions of exact optimization models, and they can be classified as deterministic (i.e., based on a set of plausible scenarios), probabilistic (i.e., assuming a given probabilistic function), or possibilistic (i.e., fuzzy-interval measures).

Simulation can be understood as the process of model 'execution' that takes a model through its evolution over time. This evolution can produce changes in the system state or not (stationary system). In addition, these changes can occur discretely or continuously through time. In discrete simulation, the *event-oriented* view works with the logic occurring at the instantaneous discrete events themselves, rather than with entities and resources (Wainer, 2017). However, the *process-oriented* world-view describes how entities move through various *processes*, where each process may require one or more *resources* and takes a certain (usually stochastic) amount of time (Couture et al., 2018). Simulation allows us to represent the real system in detail and can maintain better control over experimental conditions than by experimenting with the real system itself. A simulation model can be defined as a set of rules (e.g., equations, flowcharts, or state machines) that define how the system evolve in the future and how uncertain the system is at its present state. A valid simulation model might be able to capture the existing complex reality in a realistic and precise way. A well validated simulation should be one of the preferred approaches to employ when modeling uncertainty in real-world complex optimization problems. As Lucas et al. (2015) noted, "simulation is now an option that should be, in many ways, regarded as the method of choice for analyzing complex systems in the face of astounding advances in affordable processing power, modeling paradigms and tools, and supporting analysis capabilities". Still, stand-alone simulation methods show limitations when dealing with optimization problems of combinatorial nature, since a classical simulation approach does not incorporate efficient search methods to explore vast solution spaces.

Hence, simulation-optimization (Fu, 2015) and simulation-based optimization (Gosavi et al., 2015) methods can provide practitioners with a flexible and rich tool when dealing with optimization problems in uncertain domains. In particular, we focus here in a subset of these methods that uses metaheuristics for the optimization component. When properly designed, these *simheuristics* are capable of solving *NP-hard* and stochastic optimization problems where the simulation component copes with the uncertainty of the system and interacts with the metaheuristic component (Juan et al., 2018). The latter component, in turn, searches the solution space for a near-optimal result. In the past, some optimization problems have been solved by using simulation to evaluate the quality of solutions in engineering. Notice, however, that simheuristic algorithms go one step beyond in the sense that: *(i)* the feedback from the simulation should also be used to guide the metaheuristic search process itself; and *(ii)* all the information provided by the simulation component for a solution to the stochastic optimization problem (stochastic solution) allows considering a risk / reliability analysis; then, this analysis can be used to assess alternative stochastic solutions to the stochastic optimization problem. All these characteristics, plus the fact that integration of simulation techniques with metaheuristic algorithms is relatively simple, make simheuristics a 'first-resort' method when dealing with real-world optimization problems under uncertainty conditions. In this paper, we analyze some of the advantages of using simheuristics over traditional methods, as well as some of their limitations. Advantages range from a better understanding of the system behavior to the use of the generated information through the different simheuristic stages. For example, visualization, machine learning, and sensitivity analysis can be easily used to obtain richer information about the optimization process. We also describe how this combination of metaheuristics and simulation can be carried out to build a successful simheuristic. Several construction guidelines are given to help researchers and practitioners reach their goals. Thus, for instance, validation and stakeholders' discussion of the simulation model used within the simheuristic design and testing stages are encouraged. As simulation can tolerate far less restrictive modeling assumptions, even simple simulations must be correctly validated (Chica et al., 2017) and agreed to by as many decision makers as possible in order to lead to better decisions (Voinov and Bousquet, 2010). These guidelines promote the use of different stages to avoid jeopardizing the optimization process itself, thus obtain the best possible results with reduced computing times. The paper also includes some computational experiments that contribute to support our claims, as well as a number of references to recent publications with additional numerical results. These 'auxiliary' references show applications of simheuristics to different fields.

The rest of the paper is structured as follows: Section 2 provides a short overview of metaheuristic algorithms. Section 3 discusses how uncertainty has been traditionally addressed in optimization problems. Section 4 analyzes the basic concepts behind a simheuristic approach. Section 5 reviews previous simheuristic applications in terms of their constituent components and general results. Section 6 lists the most important advantages of using simheuristics, while Section 7 studies their main limitations and how they can be partially overcome. Section 8 provides some guidelines that can be useful during the design and implementation stages of a simheuristic algorithm. Finally, concluding remarks are provided in Section 9.

2. An Overview on Metaheuristic Optimization

According to Glover and Kochenberger (2006), metaheuristics can be defined as “an iterative process that guides the operation of one or more subordinate heuristics (which may be from a local search process to a constructive process of random solutions) to efficiently produce quality solutions for a problem”. Metaheuristics are a family of approximate non-linear optimization techniques that provide *acceptable* solutions (typically near-optimal ones), in a reasonable amount of time, for solving computationally hard and complex problems in science, engineering, and other fields. Unlike exact optimization algorithms, metaheuristics do not guarantee provably optimal solutions. However, for many large-scale real-world problems, metaheuristics might be preferred over gradient-based methods or mathematical programming (Singh and Jana, 2017). The same is true in the case of optimization problems with non-smooth objective functions (Juan et al., 2020). There are also effective gradient-based methods, like the simultaneous perturbation stochastic approximation one (Spall, 2005). These methods are suitable for adaptive modeling and optimization under uncertainty (Bhatnagar et al., 2003) and control optimization (Li et al., 2013). However, these methods show limitations in the presence of non-smooth objective functions (like the ones due to the existence of realistic soft constraints), where gradients cannot be easily computed. Metaheuristics, on the other hand, are derivative-free optimization methods.

Metaheuristics can be classified according to various characteristics (Talbi, 2009): nature-inspired vs. not nature-inspired, deterministic vs. stochastic, population-based vs. single-solution, iterative vs. greedy, etc. Another issue to be taken into account when selecting a metaheuristic is its exploration *versus* exploitation capabilities. This concept is usually linked to different sub-families. Thus, while single-solution-based algorithms manipulate and transform a single solution during the search (high intensification), population-based algorithms evolve a whole population of solutions (high diversification). Single-solution-based metaheuristics could be viewed as 'walks' through neighborhoods or search trajectories across the search space of the problem at hand. They are performed by iterative procedures that move from the current solution to another one based on local search methods. Among others, some of the most prominent metaheuristics of this sub-family are: tabu search (Glover and Laguna, 2013), simulated annealing (Kirkpatrick et al., 1983), variable neighborhood search (Hansen et al., 2010), GRASP (Feo and Resende, 1995), and iterated local search (Lourenço et al., 2010). Within the set of population-based metaheuristics, evolutionary algorithms and, in particular, genetic algorithms are frequently used in many engineering and production problems (Lee, 2018). There are many other algorithms that are based on handling a set of solutions at every iteration. These are ant-colony optimization (Dorigo and Stützle, 2004), particle-swarm optimization (Kennedy, 2010), scatter search (Laguna and Marti, 2012), and estimation of distribution algorithms (Larranaga and Lozano, 2002), among others. Finally, memetic algorithms (Moscato and Mathieson, 2019) can be seen as a marriage between population-based metaheuristics and single-solution metaheuristics. A recent and complete review on metaheuristics can be found in Hussain et al. (2019).

3. Handling with Uncertainty in Optimization Problems

The traditional formulation of optimization problems is inherently static and deterministic. However, reality is dynamic and uncertain: environmental parameters fluctuate, materials wear down, processing or transportation times vary, clients change their demands, etc. (Beyer and Sendhoff, 2007). When uncertainty is absent from the optimization formulation, the optimized solutions for those systems may be unstable and sensitive to small changes in the input parameters. A traditional way to tackle this uncertainty in optimization is by providing a high degree of robustness in the solutions. In optimization problems, robust solutions are those that remain relatively unchanged when exposed to uncertainty. Thus, a robust solution can be seen as one which is less sensitive to the perturbation of their environmental or operating conditions, uncertainties in the model outputs, and / or imprecision when measuring the decision variables. Strictly speaking, robust solutions are guaranteed to remain insensitive to changes in the system –at least within a certain range. Recoverable robustness requires that a solution is recoverable in all outcomes. Beyond these definitions, there are more relaxed and attainable degrees of robustness. In general, a robust solution possesses some specified minimum level of reliability or performance level over all outcomes and eventualities (Faulin et al., 2008). Taguchi (1989) envisioned a three-stage design methodology for robust optimization: the system, parameters, and tolerance designs. In Taguchi's method, there are two main classes of optimization parameters: *(i)* controllable parameters x that are to be tuned; and *(ii)* uncontrollable noise factors ξ, such as environmental conditions or production tolerances. In a real-world system, an optimal design has to face different types of robustness depending on the source of uncertainties on the latter parameters: changing environmental and operating conditions, production tolerances and actuator imprecision, uncertainties in the system output, and feasibility uncertainty. These types of uncertainties are usually handled by optimization methods in three different ways (Beyer and Sendhoff, 2007): deterministic, probabilistic, and possibilistic. A common approach followed in robust optimization is to consider the worst-case scenario. However, this is a conservative approach since it can result in poor optimization performance, and even in a solution that is useless in reality. Another methodology is to consider a predefined set of deterministic scenarios, where some of the parameters of the problem are uncertain or depend upon future actions (Chica et al., 2016). As an extension of this approach, an associated probability distribution could be assigned to each of these potential scenarios. Also, the search for optimal robust designs often appears as a multi-criteria decision problem, e.g.: while optimizing a conditional expectation and a large dispersion or variance. In all these cases there is a trade-off between maximal expected performance and variance. For example, one proposal along these lines is the multi-objective six sigma of Shimoyama et al. (2005), who define robustness as “stability of the system against uncertainty”.

Simulation-optimization methods in general (Fu, 2002), and simulation-based optimization in particular (Gosavi et al., 2015) constitute an excellent choice to deal with optimization problems with stochastic components. Modern computing hardware, modeling paradigms, and advanced simulation software have together made these approaches the methods of choice that can produce results to complex stochastic problems, which cannot be easily and efficiently addressed using more traditional methodologies. Simulation optimization has benefited from the development of both general computing, metaheuristics, stochastic programming, and simulation-specific modeling paradigms. Thus, simulation-optimization methods –which include simulation-based optimization and simheuristics, among others– might be an excellent choice when solving complex problems where time dynamics and uncertainty are important. Simheuristics (Juan et al., 2018) can be seen as a particular type of simulation-based optimization. Combining metaheuristics with simulation models is becoming popular as an effective procedure to deal with complex combinatorial optimization problems. To the best of our knowledge, it was with the work of Glover et al. (1996, 1999) and April et al. (2003) where this combination was popularized. These authors were the promoters of OptQuest, a ‘black-box’ optimum-seeking software product that is currently integrated into several commercial simulation-modeling packages. By using this commercial software in concert with simulation-modeling packages, a stochastic simulation model is developed for a given system. Then, the input parameters of interest are changed in an attempt to optimize a designated output performance metric (Kleijnen and Wan, 2007).

To end this Section, one should mention other approaches that are also used to deal with stochastic optimization problems. One of the most popular is stochastic programming (Prékopa, 2013). Stochastic programming integrates uncertainty consideration in mathematical programming models. This approach might be highly efficient when considering multi-stage decision processes with a reduced number of possible scenarios at each stage. However, it might also have scalability issues as the number of scenarios and stages grows. The literature on stochastic programming is quite huge, so the interested reader is referred to Ruszczyński and Shapiro (2003) for a nice overview of stochastic programming models. Similarly, stochastic Petri nets (Tigane et al., 2017) provide a powerful set of building blocks for specifying the state-transition mechanism and event-scheduling mechanism of a discrete-event stochastic system. These nets are well suited to represent concurrency, synchronization, precedence, and priority phenomena. As such, they have been used in optimization problems under uncertainty scenarios (Melani et al., 2019). Finally, chaos theory allows to analyze patterns of outcomes over time that evolve according to a deterministic equation, with these outcomes being extremely sensitive to the initial conditions. This paradigm allows for the modeling of events that are unexpected, i.e.: 'black swam' events (Taleb and Swan, 2008). Chaos theory can be combined with optimization techniques to address stochastic optimization problems (Anter and Ali, 2020).

4. The Simheuristic Approach

As discussed in Hubscher-Younger et al. (2012), it is not always possible to apply a simulation-optimization software directly out of the box. Instead, it needs to be adapted to the specific characteristics of the problem. Thus, researchers in the optimization community proposed more flexible and 'white-box' approaches. Basically, simheuristics make use of a simulation paradigm to extend existing and efficient metaheuristics. As metaheuristics are primarily designed to cope with deterministic problems, simheuristics can be seen as a metaheuristic extension to be employed when solving optimization problems under uncertainty. This simheuristics approach can be considered a subset of the simulation-for-optimization paradigm. For example, Andradóttir (2006) elaborates on the subject of simulation-based optimization methods, providing a survey on optimization add-ons for discrete-event simulation software. As pointed out by Figueira and Almada-Lobo (2014), simulation-optimization methods are designed to combine the best of both approaches in order to deal with: *(i)* optimization problems with stochastic components; and *(ii)* simulation models with optimization requirements. Among these simulation-optimization methods, the combination of simulation with metaheuristics is a promising approach for solving stochastic optimization problems that are frequently encountered by decision makers in the aforementioned industrial sectors (Glover et al., 1996, 1999). A discussion on how random search can be incorporated in simulation-optimization approaches is provided in Andradóttir (2006), while reviews and tutorials on simulation-optimization can be found in Chau et al. (2014) and Jian and Henderson (2015). Likewise, simheuristics can be seen as a specialized case of simulation-based optimization (April et al., 2003). Hybridization of simulation techniques with metaheuristics allows us to consider stochastic variables in the objective function of the optimization problem, as well as probabilistic constraints in its mathematical formulation. Hence, a simheuristic algorithm contains a particular simulation for an optimization approach, and it is oriented efficiently to tackle an optimization problem involving stochastic components. These stochastic components can be either located in the objective function (e.g., random customers' demands, random processing times, etc.) or in the set of constraints (e.g., customers' demands that must be satisfied with a given probability, deadlines that must be met with a given probability, etc.). Therefore, most of the metaheuristic frameworks can be easily extended to simheuristics, as discussed in Ferone et al. (2019) for the GRASP. For this reason, when dealing with large-scale *NP-hard* optimization problems –where uncertainty is present–, researchers should consider simheuristics as a 'first-resort' method, since they empower metaheuristic approaches to cope with more realistic stochastic models.

While exact and analytical methods offer superior performance in the *optimality* dimension (i.e., the capacity to reach optimal values), they have severe limitations in other relevant dimensions such as *scalability* (i.e., ability to deal with large-scale problems), *modeling* (i.e., capacity to develop models that accurately represent the real-life system), *uncertainty* (i.e., ability to cope with non-deterministic scenarios), or *computing times* (especially for large-scale instances of complex optimization problems). Being an 'offspring' of metaheuristics and simulation, simheuristics 'inherits' the best properties of both methodologies, thus extending metaheuristics so they can deal with uncertainty. At the same time, by adding a metaheuristic optimization component, they also extend simulation methods with the capability of coping with optimization problems successfully. Seminal research on these concepts showed applications of this methodology to different fields. Thus, for instance, April et al. (2006) constructed a simheuristic based on a discrete-event simulation model of a hospital emergency room. Their goal was to determine the optimal configuration of resources that results in the shortest average length of stay for patients. These authors also developed a simulation-optimization algorithm to minimize staffing levels for personal claims processing in an insurance company. Juan et al. (2011) employed a basic simheuristic to deal with the vehicle routing problem with stochastic demands. An enhanced and extended version of their approach was developed by Calvet et al. (2019) to solve the multi-depot stochastic vehicle routing problem. Juan et al. (2014) used a simheuristic to solve the single-period inventory routing problem with stochastic demands and stock outs, while Gruler et al. (2020) extended the previous approach to the stochastic multi-period inventory routing problem. Gonzalez-Neira et al. (2017) and Hatami et al. (2018) presented simheuristic approaches for solving different permutation flow-shop problems with stochastic processing times. An example of simheuristic applications to distributed computer networks can be found in Cabrera et al. (2014), where discrete-event simulation is combined with a simple metaheuristic framework to optimize a very large, dynamic network of non-dedicated computers offering online services over the Internet. Gruler et al. (2017, 2020) developed simheuristic approaches for supporting stochastic waste-collection management in urban areas. In De Armas et al. (2017), the authors extended a metaheuristic approach into a simheuristic one in order to cope with a stochastic version of the facility location problem. Gruler et al. (2019) propose the use of simheuristics to model human network behavior. Finally, Reyes-Rubiano et al. (2019) introduce a simheuristic algorithm for solving the electric vehicle routing problem with stochastic travel times. Most of the aforementioned applications refer to the integration of Monte Carlo (MC) simulation with a metaheuristic framework. However, other simulation paradigms are also possible (Rabe et al., 2020). Overall, we distinguish four main simulation paradigms to be used within a simheuristic. Apart from MC simulation, discrete event simulation (Heath et al., 2011), system dynamics (Sterman, 2001), and agent-based modeling (Kasaie and Kelton, 2015) are specially suitable depending on the optimization-problem characteristics and available resources.

5. Analysis of Existing Work and Some Numerical Results

In this section, we reflect on the previous implementations of simheuristics and begin by analyzing the structure of simheuristics when they are applied to different problem domains. We also consider possible future simheuristic developments, and focus on the general results that emerge from simheuristic algorithms applied to different fields. Likewise, similarities that exist among simheuristic applications are also discussed, as well as the evolution of the simheuristic framework. Firstly, each simheuristic has the following common steps: *(i)* an input deterministic equivalent model of the stochastic combinatorial optimization problem; *(ii)* an iterative search stage that integrates information from simulation testing of candidate solutions; and *(iii)* one or several best stochastic solutions (i.e., solutions for the stochastic version of the problem), which are returned as the output at the end of the algorithm. Regarding the variety of cases that arise when considering different problem domains, the following can be said: in some cases, the simulation component –which is typically a Monte Carlo simulation or a discrete-event simulation– is used only in a parameter initialization phase, where the expected costs of some predefined policies are approximated. This is the case, for instance, in which the fixed costs are a function of the decision variables but the stochastic / variable costs are not. In most of the applications considered so far, demand has been the stochastic element. Other applications consider processing time uncertainty, service costs, node availability, and cash flows. The simheuristic framework is easily extensible for multiple stochastic elements. Most simheuristic implementations employ a distinct initial solution procedure. In some cases this is required because the metaheuristic component is, by itself, only capable of considering perturbations of a current base solution. In other applications the initialization procedure is used because it had been found that the quality of the initial solution had a significant impact on the quality of the final solution. In more recent applications, it has become increasingly common to use biased-randomized greedy constructive algorithms (Quintero-Araujo et al., 2017) to generate initial solutions. One of the advantages of such an approach is that it facilitates the use of multi-start metaheuristics, which guarantee a more comprehensive exploration of the search space in question. A similar trend can be seen in the choice of the metaheuristic algorithm. Early applications tended to consider relatively simple but efficient heuristics. Thus, for example, in Gonzalez-Martin et al. (2018) a randomized savings heuristic for the arc routing problem is utilized. Some simheuristics used a local search algorithm as the metaheuristic component. Others, such as the one in Pagès-Bernaus et al. (2019), used iterated local search. Yet, more recent applications use the more advanced variable neighborhood search metaheuristic (VNS) framework (Panadero et al., 2020). One of the advantages of VNS algorithms is that they use multiple neighborhood structures, which improve both the exploration and intensification properties of the search trajectory. Given these considerations, the combination of biased-randomization and a VNS search is a very strong approach for ensuring the quality of the optimization component of a simheuristic. In general, the choice of the specific metaheuristic framework should account for the complexity of the simulation component of the problem, as longer simulation times extend the required run times. In other words, simpler simulation models enable the use of more complex metaheuristic algorithms and vice-versa.

Another recurrent theme in simheuristic algorithms is that of using the deterministic value of a candidate solution as a criteria for determining whether that solution should be tested in the integrated simulation component –i.e., as a potential candidate stochastic solution. In applications where simulation runs are not computationally expensive, all candidate solutions can be tested in the integrated simulation model. In different simheuristic applications, the role of the integrated simulation component varies. In some cases the simulation is used to check whether a candidate solution adheres to a number of arbitrary constraints, such as a minimum reliability level (Cabrera et al., 2014). However, by far the most common purpose of the simulation component is that of estimating the stochastic value of a candidate solution. One of the advantages of the simheuristic framework is that both multiple objectives and arbitrary constraints can be handled easily, so future applications could use more of the information output from simulation runs. On the whole, the simulation component of a simheuristic can be utilized during an initial parameter-estimation stage, an optimization stage, and a reliability-analysis stage. The output of simheuristics takes the form of a best stochastic solution or a pool of elite stochastic solutions. Having a pool of elite solutions can be useful for three reasons: *(i)* for storing promising stochastic solutions and complete a risk / reliability analysis over them; *(ii)* for storing a Pareto front of non-dominated solutions –in cases where multiple goals are considered, as in Gruler et al. (2017); and *(iii)* for providing decision makers with a range of alternative solutions, so that they might be able to select a solution that satisfies a number of other arbitrary constraints. In general, it can be seen that a simheuristic is built from a number of relatively fixed steps, including the choice of simulation paradigm, metaheuristic methodology, and output type. In addition, simheuristics have seen an increasing number of optional steps, including: using simulation to provide initial parameter estimates, the use of a distinct initial solution method, and a final detailed reliability analysis. Recent applications tend to include previously introduced steps whilst introducing new ones.

Having discussed the evolving simheuristic framework in some detail, we now consider their possible future evolution. For instance, the input problem that the metaheuristic component searches directly is always the deterministic equivalent model of the stochastic model, where the stochastic variables are replaced by their means. Another approach that could be tested in future applications is to periodically change the deterministic equivalent model by generating random realizations, according to the respective distributions, of some or all of the stochastic variables. Such an approach provides an additional escape mechanism from local stochastic optima. It could also help to improve the diversity of the final elite solution set. Additionally, this represents an alternative method of integrating simulation within the metaheuristic search process. Another possible extension would be to dynamically adjust the number of simulation runs used in the integrated simulation component. For example, the integrated simulation could be terminated as soon as the confidence interval of its stochastic value falls entirely below that of the current best stochastic solution. Such an approach will benefit the run-time of a simheuristic. Yet another possibility would be to generalize the structure of simheuristic algorithms to the extent that it becomes a decision variable. For example, the structure of a simheuristic could be encoded as an integer string. The first integer could correspond to the choice of the initial solution generation method, the second to the choice of the metaheuristic, and so on. Such an approach adds an additional layer to the search, and would thus be most useful for cases where sufficient time is available for generating a solution. In such an investigation, fair testing can be ensured by setting a simulation budget for each instance of a simheuristic algorithm.

Figure 1 displays the gaps of the best deterministic solutions (those associated with the deterministic version of the problem when they are used in a stochastic environment) and the best stochastic solutions (those associated with the stochastic version) found by different simheuristic algorithms. These gaps are computed with respect to the best-known solution for the deterministic version of the problem when it is assessed in a scenario without uncertainty. From this Figure, one can conclude that optimal / near-optimal deterministic solutions might have a poor performance in stochastic scenarios. Notice that this result holds in a wide variety of problem domains. In the following, deterministic scenarios / solutions are denoted as det, while stochastic scenarios / solutions are denoted as $stoch$. For example $OBS_{det,stoch}$ refers to the objective value of our best deterministic solution when evaluated in a stochastic scenario. Then, the Figure also supports the following general result for a minimization problem: $BKS_{det,det} \leq E\left[OBS_{stoch,stoch}\right] \leq E\left[OBS_{det,stoch}\right]$, i.e.: the deterministic value of the best-known deterministic solution ($BKS_{det,det}$) is a lower bound for the stochastic value of the

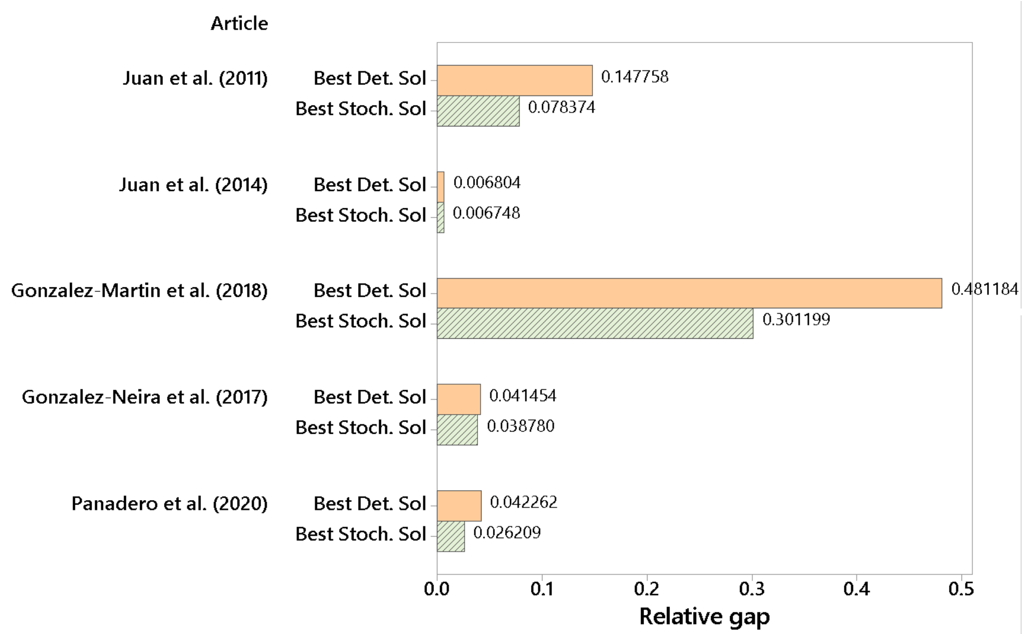

Figure 1. *Relative gaps of the best stochastic and deterministic solutions found by simheuristics compared to the deterministic value of the best-known solutions.*

best stochastic solution ($OBS_{stoch,stoch}$). At the same time, the latter has the stochastic value of the best-known deterministic solution ($OBS_{det,stoch}$) as an upper bound. Figure 1 also highlights the potential benefits of employing a simheuristic in problems that feature uncertainty.

Figure 2 displays optimality gaps and solution times relative to those of several exact methods –for the cases where such experimental results are available. This Figure shows that simheuristics are very competitive in terms of the trade-off between solution quality and solution time. Hence, simheuristics are able to generate solutions that are very close to optimality, and can do so in a small fraction of the time required by exact solution approaches.

Likewise, Figure 3 illustrates the effect that the level of variance in the stochastic instance has on the value of the simheuristic solution –as compared with the deterministic value of the best-known solution for the deterministic version of the problem. This

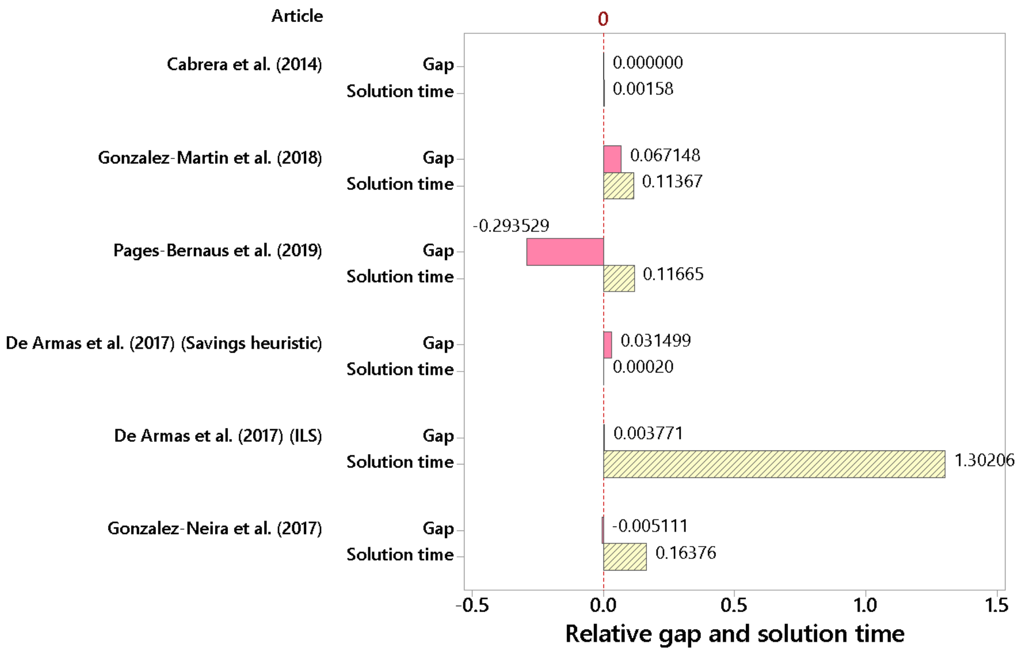

Figure 2. *Optimality gaps and relative solutions times of simheuristics compared to exact formulations over a range of simheuristic applications and problem domains.*

Figure shows that, in approximately 50% of the cases, increasing the variance of the stochastic parameters of an instance also raises the gap of the stochastic solution relative to the deterministic value of the best-known deterministic solution. In the remaining 50% of the cases, increasing the variance of the stochastic parameters of a problem instance has little or no effect.

6. Advantages of Using Simheuristics in Optimization

This section highlights the main advantages of employing simheuristics, which justify why we propose this methodology as a 'first-resort' method for dealing with optimization problems under uncertainty:

- *Embracing reality by a validated simheuristic:* As opposed to the use of standalone analytical models, integrating simulation within a metaheuristic / matheuris-

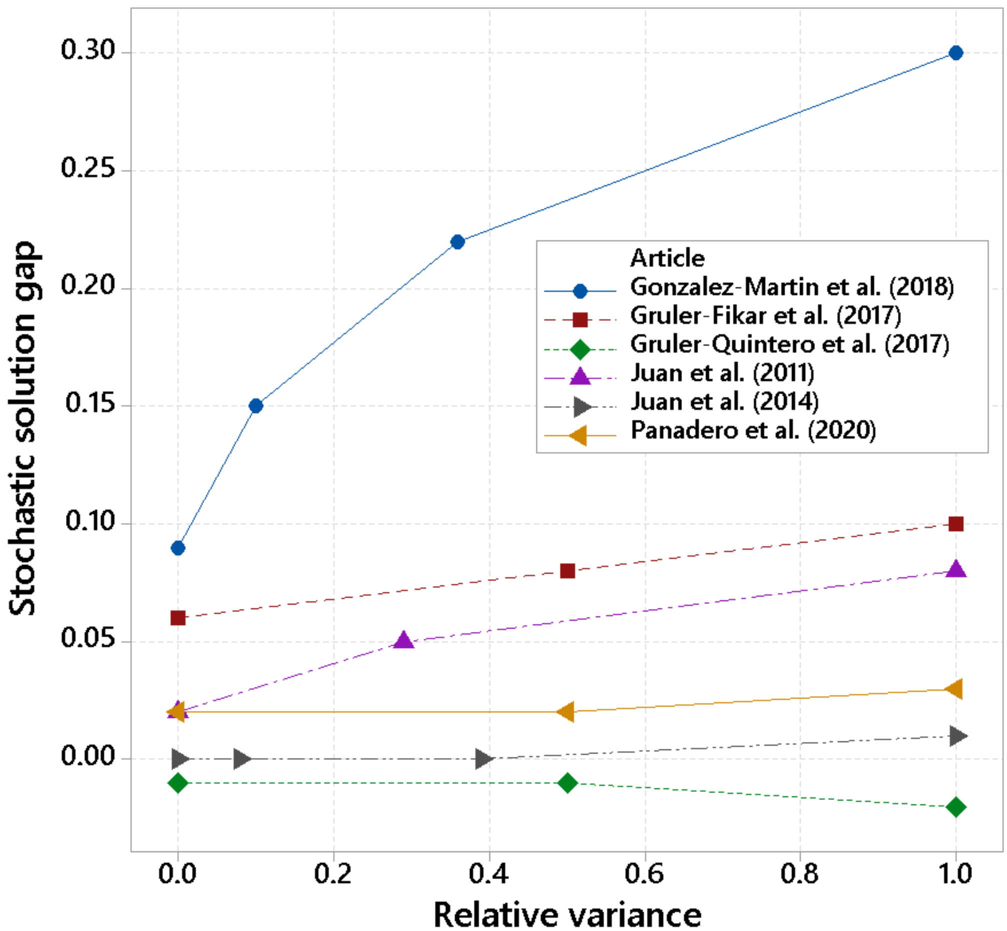

Figure 3. *The effect of increasing the variance of the stochastic variables on the relative gap between the value of our best stochastic solutions and the deterministic value of the best-known deterministic solution.*

tic approach allows researchers and decision makers to construct and study valid models of complex systems. Most recent simulation paradigms also allow for analysis of optimization problems under uncertainty with a low number of assumptions. These paradigms also facilitate involvement of stakeholders, who are not directly the modelers of the simheuristic, i.e., participatory modeling (Voinov and Bousquet, 2010). There are new simulation-optimization paradigms that can better represent complex reality, and powerful computational resources to run demanding simulation models. Model validation is a central pillar within the simulation community, as evidenced by its ubiquity in the leading texts over the years (Kelton et al., 2015). But validation should be applied to all modeling, including analytical, so this is not a disadvantage –but a requirement– when using simulation-optimization.

- *Risk assessment of alternative solutions and sensitivity analysis:* Once a simulation is built and validated, finding robust policies and comparing the merits of various policies are two of the main goals (Kleijnen et al., 2005). Joint use of simulation and metaheuristics / matheuristics within a simheuristic framework can help attain these two goals and has advantages compared to other stand-alone methodologies. The results of the simulations can be used to obtain additional information about the probability distribution of the quality of each stochastic solution. This information is then used to introduce a risk / reliability analysis within the decision-making process. The risk-analysis capability of simheuristics is one of its major advantages. This is due to the ability of metaheuritics to generate a set of different solutions, as well as to the ability of the simulation model to provide an observational sampling of the system. Thus, for instance, stochastic solutions with similar expected cost might show different variance, or even different reliability levels –i.e., some routing plans might have a high probability of failure when put into practice, while others might be more reliable. Running a sensitivity analysis (Saltelli et al., 2008) is another advantage of using a simulation together with a metaheuristic method. Sensitivity analysis reveals those input parameters that are most critical in determining the value of key output performance metrics. Usually, this is achieved by exploring the model sensitivity to a particular parameter configuration and input-value options. Sensitivity analysis is typically carried out to gain insights into existing or prospective systems, and this should lead to better decisions and to improved managerial outcomes. This sensitivity analysis can be directly run by studying the output of the different simulation runs. Although a complete sensitivity analysis requires more advanced methods and specific tools to this end (Chica et al., 2017), the simheuristic learning process can give the modeler a first approach to a deeper sensitivity analysis of the system whose optimization is sought.

- *System understanding and output analysis:* When the simheuristic finishes, we can collect the output-data results and analyze them through machine-learning algorithms to discover hidden properties or relationships. The goal is to enable researchers to identify system patterns interactively, run high-dimensional explorations, or even check the veracity of the approximately-optimized simulation system (Lucas et al., 2015). This is also called the *innovization process* in evolutionary-computation research (Deb et al., 2014). It means that a set of trade-off optimal or near-optimal solutions, found using metaheuristics, are analyzed to decipher useful relationships among problem entities. It provides a better understanding of the problem to a designer or a practitioner. We extend here this concept by adding the simulation face of the simheuristic to enrich the *innovization process*. Additionally, visualization methods (e.g., histograms, box plots, or scatter plots) can be directly used to visualize post-run simulation outputs that go beyond the traditional analysis of the results. There is an increasing number of studies demonstrating that visualization combined with optimization can promote design innovations and provide decision makers with an improved understanding of the problem (Bonissone et al., 2009). A good visualization enables decision makers to enhance insight into the problem and the different solutions to identify differences and similarities before coming to the final decision (Miettinen, 2014). Exploratory analysis of the input / output variables space of a model is also employed to strengthen confidence in the model realism and to improve understanding of the behavior of the optimization and simulation models. By analyzing the distribution of the model variables and parameters, the modeler can move forward to a simpler and easier-to-understand setting. Use of this exploration, together with sensitivity analysis, provides information on influential factors that significantly affect the variability of the model results, and allow modelers to reach a deeper understanding of the complexity of the model, its uncertainties, interrelationships, and its potential future scenarios (Ligmann-Zielinska et al., 2014).

7. Limitations of Simheuristics

As with any methodology, there are also limitations when using simheuristics. In this section, we highlight some of these limitations as well as some positive aspects that ameliorate their negative impact on the optimum-seeking process.

- *Results are not expected to be truly provably optimal:* Metaheuristics do not ensure an optimal solution to an optimization problem, but rather an acceptable solution in a reasonable amount of time. This fact is amplified when using a simulation to be optimized. Even more, this simulation is a non-linear complex stochastic system that cannot be analytically treated. Therefore, simheuristics are an interesting alternative for practical cases requiring simple and flexible methods that do not need to be globally optimal –although they are usually near-optimal.

- *Additional stakeholders' effort is demanded to define the system*: The set of advantages and 'white-box' paradigms used in a simheuristic also requires additional effort when defining the simulation system and analyzing the results provided by the simheuristic. However, we think this design and validation effort is justified as modelers and decision makers can better understand their system from the results of the simheuristics and can adopt the final optimum-seeking results with higher confidence.

- *More computational resources are required compared to traditional methods*: The integration of a simulation engine within a metaheuristic requires high computational effort and also depends on the selected type of simulation paradigm. As will be discussed in Section 8, different strategies can be applied in order to alleviate this effort, such as: *(i)* 'filtering' the solutions generated by the metaheuristic engine, so that only the 'promising' ones are actually sent to the simulation component; and *(ii)* using a small number of simulation runs in a first stage, and then analyzing in more detail only those that can be classified as 'very promising' solutions.

8. Best Design and Implementation Practices

In this section we outline a set of guidelines or best practices to build a simheuristic algorithm appropriately.

- *Do not overload simheuristics with long simulations:* In general, the modeler has to be careful not to let the simulation jeopardize the computing time given to the entire simulation-metaheuristic process. Otherwise, the metaheuristic would not have time to converge to a good solution if the dimension of the search space is high. Therefore, we recommend decomposing the simheuristic into various stages. For instance, a three-stage approach could be considered. During the first stage, only fast simulations are included in the simheuristic framework. This can be achieved by running the simulation only a limited number of times to obtain rough estimates, or by running the simulation for only those new solutions of the metaheuristic that can be considered as 'promising' ones (e.g., solutions with good deterministic performance). During this stage, the simulation component of the simheuristic is used not only as a natural way to model the real system, but it also can provide valuable information to the metaheuristic component (i.e., the search process is simulation-driven). For example, it can be used to filter low quality solutions quickly. In a second stage, the best solutions identified in the previous stage are sent throughout a new simulation process with a larger number of iterations to obtain more precise estimates of the uncertain values of the model. The specific number of iterations might be given by error measures such as confidence intervals of the parameters with high uncertainty. Finally, a third and final stage can be used to complete a risk / reliability analysis on the best solutions selected by the decision maker. Dimensions other than the expected value of the solution need to be considered in a high-uncertainty environment, since a solution with a low expected value could also show more variability than other alternative solutions. For example, in a flow-shop scheduling problem with stochastic processing times there might be several solutions (job permutations) that offer a similar expected makespan; however, some of these solutions might show a higher variability than others, or a lower probability of finishing before a given deadline. Similarly, in a vehicle routing problem with stochastic demands, several solutions might offer similar expected costs, but some of these solutions might also show a higher variability than others. Consequently, the decision maker would need more information to decide which solution to choose based on her / his utility function and aversion to risk, or would even need more advanced optimization methods –such as multi-objective optimization– to have a set of solutions with different trade-offs between expected cost value and robust behavior in the environment.

- *Choose a simulation paradigm that is understandable to decision makers:* Three main goals must be accomplished when developing and selecting the simulation model (Kleijnen et al., 2005): *(i)* develop a basic understanding of the simulation model and the system it emulates; *(ii)* find robust policies and decisions; and *(iii)* compare the merits of various policies or decisions. As mentioned, there is a wide set of available simulation paradigms and within each variant, many variations and possible designs arise. Our guideline here is to use, as much as possible, a participatory simulation-modeling process to increase and share the knowledge and understanding of the system between all the actors involved in the optimization action (Voinov and Bousquet, 2010). This involvement would also clarify and identify the impacts of solutions to a given problem, usually related to the final decision-making support.

- *Choose an appropriate simulation paradigm for each stage of a simheuristic:* Different simulation paradigms can be used for each of the stages of the simheuristic. Then, a more enriched and computationally-intensive simulation model (e.g., agent-based modeling) can be used for the last stages of the simheuristic and applied only to a reduced set of the solutions provided by the metaheuristic. In contrast, lighter computational simulation models (e.g., a simple Monte Carlo simulation over the stochastic simulation model) might be required in the first stage of the simheuristic. Each individual modeling paradigm has a rich history and exemplar cases in which the strengths of the respective methodology make it a good choice for a particular modeling situation. There also possibilities for combining each pair of approaches to develop hybrid models where each paradigm exploits its strengths (Heath et al., 2011). For instance, Djanatliev and German (2013) present different multi-paradigm simulation methods.

- *Validate the simulation model before running the simheuristic:* A decisive phase when modeling a real-world system is model validation (Oliva, 2003). In our view, this is also a main guideline when designing the simheuristic, as it applies to the simheuristic itself and specifically to its simulation component. The validation requires testing a set of hypotheses, the significance of their behavioral components (by assuming that the behavior is a consequence of the system structure), and the historical model fitting. Validation is also measured in terms of degrees of confidence or quality, which is usually difficult to obtain for most non-linear simulation models in use (Forrester, 2007). The validation and testing of any model or decision-support system is a decisive step for ensuring its managerial adoption. Decision makers are all rightly concerned about whether results of each model are *correct* (Sargent, 2005). However, the validation of non-linear models and their effectiveness for real-world problems is not straightforward. The validation stage can be seen as a learning process where the modeler's understanding is enhanced through her / his interaction with the formal and mental model (Morecroft, 2007). As this process evolves, both the formal and mental perceptions of the modelers change, leading to a successive approximation of the formal model to reality. Additionally, the utility and effectiveness of many non-linear models and their outputs are often judged by stakeholders and decision makers (Voinov and Bousquet, 2010). Therefore, it is highly recommended to perform the validation of the models correctly. A set of validation techniques such as calibration (Sargent, 2005), sensitivity analysis (Saltelli et al., 2008), boundary adequacy, and extreme cases tests (Qudrat-Ullah and Seong, 2010) should be carried out for the corresponding simheuristic component in order to guarantee that the simulation model is a valid representation of the underlying system.

9. Concluding Remarks

The motivation of this paper is to advocate that a combination of simulation models and metaheuristics / matheuristics should be considered as a first-resort method when dealing with large-scale *NP-hard* optimization problems with stochastic components, which is a quite common case when considering real-world challenges. In effect, many real-life optimization problems in areas such as logistics, transportation, scheduling, etc., are complex, large-scale, and involve uncertainties regarding their constraints, input values, and objective functions. Although there are metaheuristic applications that add probabilistic and robustness capabilities to analytical models, they are extensions to the original deterministic model formulation. As we have discussed, integration of simulation methods with metaheuristics and matheuristics is a natural way to cope with these problems. Although prohibitive and unaffordable in the past, advanced simulation methods are now commonly used in research and practice due to widespread and affordable availability of high-performance computing resources and much-improved software for simulation modeling and analysis. The same is true for metaheuristics and matheuristics. As it has been shown in a number of recent publications containing extensive computational experiments, the simheuristics methodology can better face complex reality when seeking optima in uncertain environments.

In this paper we highlighted three main advantages of using simheuristics. First, it is a better way to embrace the reality of the systems we are seeking to optimize. There is no need to include many strong and over-simplifying assumptions to render a tractable model. Second, a simheuristic can easily provide a risk assessment of the optimization-problem solutions. Third, simheuristics facilitate the understanding of the system's behavior. *A posteriori* analysis applied to the output provided by the simheuristic can help modelers to understand the system dynamics. For instance, one can observe the most sensitive parameters, or even apply statistical analysis to the returned set of optimization solutions to find relationships between them. Visualization techniques are also useful to generate insights about the system, based on the output of the simheuristic method.

Additionally, we have presented the main simulation paradigms to be used within a simheuristic, and a list of guidelines to take into account when designing a simheuristic. We suggested the use of a multi-stage approach to alleviate the required computation effort of the simulation, and the utilization of different simulation paradigms within the simheuristic. Likewise, the need for using a validated simulation model was affirmed. Finally, we encourage the use of a simheuristic paradigm that can be aligned with the 'white-box' paradigm: being understandable and enhancing the decision makers' participation.

Acknowledgments

This work has been partially supported by the Spanish Ministry of Science, Innovation, and Universities (PID2019-111100RB-C21, RED2018-102642-T, EXASOCO PGC2018-101216-B-I00). Likewise, we want to acknowledge the support received by the Erasmus+ programme (2019-I-ES01-KA103-062602). M. Chica is supported by the Ramón y Cajal program (RYC-2016-19800) and SIMARK (PY18-4475), granted by the Andalusian Government.

References

Andradóttir, S. (2006). An overview of simulation optimization via random search. *Handbooks in Operations Research and Management Science 13*, 617–631.

Anter, A. M. and M. Ali (2020). Feature selection strategy based on hybrid crow search optimization algorithm integrated with chaos theory and fuzzy c-means algorithm for medical diagnosis problems. *Soft Computing 24*(3), 1565–1584.

April, J., M. Better, F. Glover, J. Kelly, and M. Laguna (2006). Enhancing business process management with simulation optimization. In *Proceedings of the Winter Simulation Conference*, Piscataway, New Jersey, pp. 642–649. IEEE.

April, J., F. Glover, J. P. Kelly, and M. Laguna (2003). Simulation-based optimization: practical introduction to simulation optimization. In *Proceedings of the Winter Simulation Conference*, Piscataway, New Jersey, pp. 71–78. IEEE.

Beyer, H.-G. and B. Sendhoff (2007). Robust optimization–a comprehensive survey. *Computer Methods in Applied Mechanics and Engineering 196*(33-34), 3190–3218.

Bhatnagar, S., M. C. Fu, S. I. Marcus, and I.-J. Wang (2003). Two-timescale simultaneous perturbation stochastic approximation using deterministic perturbation sequences. *ACM Transactions on Modeling and Computer Simulation 13*(2), 180–209.

Bonissone, P. P., R. Subbu, and J. Lizzi (2009). Multicriteria decision making: a framework for research and applications. *IEEE Computational Intelligence Magazine 4*(3), 48–61.

Boschetti, M. A., V. Maniezzo, M. Roffilli, and A. B. Röhler (2009). Matheuristics: optimization, simulation and control. In *Hybrid Metaheuristics*, pp. 171–177. Springer.

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.

Calvet, L., D. Wang, A. A. Juan, and L. Bové (2019). Solving the multidepot vehicle routing problem with limited depot capacity and stochastic demands. *International Transactions in Operational Research 26*(2), 458–484.

Chau, M., M. C. Fu, H. Qu, and I. O. Ryzhov (2014). Simulation optimization: a tutorial overview and recent developments in gradient-based methods. In *Proceedings of the Winter Simulation Conference*, Piscataway, New Jersey, pp. 21–35. IEEE.

Chica, M., J. Barranquero, T. Kajdanowicz, O. Cordón, and S. Damas (2017). Multimodal optimization: an effective framework for model calibration. *Information Sciences 375*, 79–97.

Chica, M., J. Bautista, Ó. Cordón, and S. Damas (2016). A multiobjective model and evolutionary algorithms for robust time and space assembly line balancing under uncertain demand. *Omega 58*, 55–68.

Couture, R.-M., S. J. Moe, Y. Lin, Ø. Kaste, S. Haande, and A. L. Solheim (2018). Simulating water quality and ecological status of Lake Vansjø, Norway, under land-use and climate change by linking process-oriented models with a Bayesian network. *Science of the Total Environment 621*, 713–724.

De Armas, J., A. A. Juan, J. M. Marquès, and J. P. Pedroso (2017). Solving the deterministic and stochastic uncapacitated facility location problem: from a heuristic to a simheuristic. *Journal of the Operational Research Society 68*(10), 1161–1176.

Deb, K., S. Bandaru, D. Greiner, A. Gaspar-Cunha, and C. C. Tutum (2014). An integrated approach to automated innovization for discovering useful design principles: case studies from engineering. *Applied Soft Computing 15*, 42–56.

Djanatliev, A. and R. German (2013). Prospective healthcare decision-making by combined system dynamics, discrete-event and agent-based simulation. In *Proceedings of the Winter Simulation Conference*, Piscataway, New Jersey, pp. 270–281. IEEE.

Dokeroglu, T., E. Sevinc, T. Kucukyilmaz, and A. Cosar (2019). A survey on new generation metaheuristic algorithms. *Computers & Industrial Engineering 137*, 106040.

Dorigo, M. and T. Stützle (2004). *Ant Colony Optimization*. MIT Press, Cambridge.

Faulin, J., A. A. Juan, C. Serrat, and V. Bargueno (2008). Predicting availability functions in time-dependent complex systems with saedes simulation algorithms. *Reliability Engineering & System Safety 93*(11), 1761–1771.

Feo, T. A. and M. G. C. Resende (1995). Greedy randomized adaptive search procedures. *Journal of Global Optimization 6*(2), 109–133.

Ferone, D., A. Gruler, P. Festa, and A. A. Juan (2019). Enhancing and extending the classical GRASP framework with biased randomisation and simulation. *Journal of the Operational Research Society 70*(8), 1362–1375.

Figueira, G. and B. Almada-Lobo (2014). Hybrid simulation–optimization methods: a taxonomy and discussion. *Simulation Modelling Practice and Theory 46*, 118–134.

Fischetti, M. and M. Fischetti (2018). Matheuristics. In *Handbook of Heuristics*, pp. 121–153. Springer.

Forrester, J. W. (2007). System dynamics: the next fifty years. *System Dynamics Review 23*(2-3), 359–370.

Fu, M. C. (2002). Optimization for simulation: theory vs. practice. *INFORMS Journal on Computing 14*(3), 192–215.

Fu, M. C. (2015). *Handbook of Simulation Optimization*, Volume 216. Springer.

Glover, F., J. P. Kelly, and M. Laguna (1996). New advances and applications of combining simulation and optimization. In *Proceedings of the Winter Simulation Conference*, Piscataway, New Jersey, pp. 144–152. IEEE.

Glover, F., J. P. Kelly, and M. Laguna (1999). New advances for wedding optimization and simulation. In *Proceedings of the Winter Simulation Conference*, Volume 1, Piscataway, New Jersey, pp. 255–260. IEEE.

Glover, F. and M. Laguna (2013). Tabu search. In *Handbook of Combinatorial Optimization*, pp. 3261–3362. Springer.

Glover, F. W. and G. A. Kochenberger (2006). *Handbook of Metaheuristics*, Volume 57. Springer Science & Business Media.

Gonzalez-Martin, S., A. A. Juan, D. Riera, M. G. Elizondo, and J. J. Ramos (2018). A simheuristic algorithm for solving the arc routing problem with stochastic demands. *Journal of Simulation 12*(1), 53–66.

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.

Gosavi, A. et al. (2015). *Simulation-Based Optimization*. Springer.

Gruler, A., J. d. Armas, A. A. Juan, and D. Goldsman (2019). Modelling human network behaviour using simulation and optimization tools: the need for hybridization. *SORT-Statistics and Operations Research Transactions 43*(2), 0193–222.

Gruler, A., C. Fikar, A. A. Juan, P. Hirsch, and C. Contreras-Bolton (2017). Supporting multi-depot and stochastic waste collection management in clustered urban areas via simulation–optimization. *Journal of simulation 11*(1), 11–19.

Gruler, A., J. Panadero, J. de Armas, J. A. Moreno, and A. A. Juan (2020). A variable neighborhood search simheuristic for the multiperiod inventory routing problem with stochastic demands. *International Transactions in Operational Research 27*(1), 314–335.

Gruler, A., A. Perez-Navarro, L. Calvet, and A. A. Juan (2020). A simheuristic algorithm for time-dependent waste collection management with stochastic travel times. *SORT-Statistics and Operations Research Transactions 44*(2), 1–29.

Gruler, A., C. L. Quintero, L. Calvet, and A. A. Juan (2017). Waste collection under uncertainty: a simheuristic based on variable neighbourhood search. *European Journal of Industrial Engineering 11*(2), 228–255.

Hansen, P., N. Mladenović, and J. A. Moreno (2010). Variable neighbourhood search: methods and applications. *Annals of Operations Research 175*(1), 367–407.

Hatami, S., L. Calvet, V. Fernández-Viagas, J. M. Framiñán, and A. A. Juan (2018). A simheuristic algorithm to set up starting times in the stochastic parallel flowshop problem. *Simulation Modelling Practice and Theory 86*, 55–71.

Heath, S. K., A. Buss, S. C. Brailsford, and C. M. Macal (2011). Cross-paradigm simulation modeling: challenges and successes. In *Proceedings of the Winter Simulation Conference*, Piscataway, New Jersey, pp. 2788–2802. IEEE.

Hubscher-Younger, T., P. J. Mosterman, S. DeLand, O. Orqueda, and D. Eastman (2012). Integrating discrete-event and time-based models with optimization for resource allocation. In *Proceedings of the Winter Simulation Conference*, Piscataway, New Jersey, pp. 1–15. IEEE.

Hussain, K., M. N. M. Salleh, S. Cheng, and Y. Shi (2019). Metaheuristic research: a comprehensive survey. *Artificial Intelligence Review 52*(4), 2191–2233.

Jian, N. and S. G. Henderson (2015). An introduction to simulation optimization. In *Proceedings of the Winter Simulation Conference*, Piscataway, New Jersey, pp. 1780–1794. IEEE.

Juan, 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., C. G. Corlu, R. D. Tordecilla, R. de la Torre, and A. Ferrer (2020). On the use of biased-randomized algorithms for solving non-smooth optimization problems. *Algorithms 13*(1), 8.

Juan, A. A., S. E. Grasman, J. Caceres-Cruz, and T. Bektaş (2014). 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., W. D. Kelton, C. S. Currie, and J. Faulin (2018). Simheuristics applications: dealing with uncertainty in logistics, transportation, and other supply chain areas. In *Proceedings of the Winter Simulation Conference*, Piscataway, New Jersey, pp. 3048–3059. IEEE.

Kasaie, P. and W. D. Kelton (2015). Guidelines for design and analysis in agent-based simulation studies. In *Proceedings of the Winter Simulation Conference*, Piscataway, New Jersey, pp. 183–193. IEEE.

Keith, A. J. and D. K. Ahner (2019). A survey of decision making and optimization under uncertainty. *Annals of Operations Research SI*, 1–35.

Kelton, W. D., R. Sadowski, and N. B. Zupick (2015). *Simulation with Arena (6th Edition)*. McGraw-Hill Education.

Kennedy, J. (2010). Particle swarm optimization. In *Encyclopedia of Machine Learning*, pp. 760–766. Springer.

Kirkpatrick, S., J. C. D. Gelatt, and M. P. Vecchi (1983). Optimization by simulated annealing. *Science 220*(4598), 671–680.

Kleijnen, J. P., S. M. Sanchez, T. W. Lucas, and T. M. Cioppa (2005). State-of-the-art review: a user's guide to the brave new world of designing simulation experiments. *INFORMS Journal on Computing 17*(3), 263–289.

Kleijnen, J. P. and J. Wan (2007). Optimization of simulated systems: OptQuest and alternatives. *Simulation Modelling Practice and Theory 15*(3), 354–362.

Laguna, M. and R. Marti (2012). *Scatter Search: Methodology and Implementations in C*, Volume 24. Springer Science & Business Media.

Larranaga, P. and J. A. Lozano (2002). *Estimation of Distribution Algorithms: a New Tool for Evolutionary Computation*, Volume 2. Springer Science & Business Media.

Lee, C. K. H. (2018). A review of applications of genetic algorithms in operations management. *Engineering Applications of Artificial Intelligence 76*, 1–12.

Li, L., B. Jafarpour, and M. R. Mohammad-Khaninezhad (2013). A simultaneous perturbation stochastic approximation algorithm for coupled well placement and control optimization under geologic uncertainty. *Computational Geosciences 17*(1), 167–188.

Ligmann-Zielinska, A., D. B. Kramer, K. S. Cheruvelil, and P. A. Soranno (2014). Using uncertainty and sensitivity analyses in socioecological agent-based models to improve their analytical performance and policy relevance. *PloS one 9*(10), e109779.

Lourenço, H. R., O. C. Martin, and T. Stützle (2010). Iterated local search: framework and applications. In *Handbook of Metaheuristics*, pp. 363–397. Springer.

Lucas, T. W., W. D. Kelton, P. J. Sánchez, S. M. Sanchez, and B. L. Anderson (2015). Changing the paradigm: simulation, now a method of first resort. *Naval Research Logistics 62*(4), 293–303.

Melani, A. H., C. A. Murad, A. Caminada Netto, G. F. Souza, and S. I. Nabeta (2019). Maintenance strategy optimization of a coal-fired power plant cooling tower through generalized stochastic Petri nets. *Energies 12*(10), 1951.

Miettinen, K. (2014). Survey of methods to visualize alternatives in multiple criteria decision making problems. *OR Spectrum 36*(1), 3–37.

Morecroft, J. (2007). *Strategic Modelling and Business Dynamics: A Feedback Systems Approach*. John Wiley & Sons.

Moscato, P. and L. Mathieson (2019). Memetic algorithms for business analytics and data science: a brief survey. In *Business and Consumer Analytics: New Ideas*, pp. 545–608. Springer.

Oliva, R. (2003). Model calibration as a testing strategy for system dynamics models. *European Journal of Operational Research 151*(3), 552–568.

Pagès-Bernaus, A., H. Ramalhinho, A. A. Juan, and L. Calvet (2019). Designing e-commerce supply chains: a stochastic facility–location approach. *International Transactions in Operational Research 26*(2), 507–528.

Panadero, J., J. Doering, R. Kizys, A. A. Juan, and A. Fito (2020). A variable neighborhood search simheuristic for project portfolio selection under uncertainty. *Journal of Heuristics 26*(3), 353–375.

Prékopa, A. (2013). *Stochastic Programming*, Volume 324. Springer Science & Business Media.

Qudrat-Ullah, H. and B. S. Seong (2010). How to do structural validity of a system dynamics type simulation model: the case of an energy policy model. *Energy Policy 38*(5), 2216–2224.

Quintero-Araujo, C. L., J. P. Caballero-Villalobos, A. A. Juan, and J. R. Montoya-Torres (2017). A biased-randomized metaheuristic for the capacitated location routing problem. *International Transactions in Operational Research 24*(5), 1079–1098.

Rabe, M., M. Deininger, and A. A. Juan (2020). Speeding up computational times in simheuristics combining genetic algorithms with discrete-event simulation. *Simulation Modelling Practice and Theory 103*, 102089.

Reyes-Rubiano, L., D. Ferone, A. A. Juan, and J. Faulin (2019). A simheuristic for routing electric vehicles with limited driving ranges and stochastic travel times. *SORT-Statistics and Operations Research Transactions 1*(1), 3–24.

Ruszczyński, A. and A. Shapiro (2003). Stochastic programming models. *Handbooks in Operations Research and Management Science 10*, 1–64.

Saltelli, A., M. Ratto, T. Andres, F. Campolongo, J. Cariboni, D. Gatelli, M. Saisana, and S. Tarantola (2008). *Global Sensitivity Analysis: the Primer*. John Wiley & Sons.

Sargent, R. G. (2005). Verification and validation of simulation models. In *Proceedings of the Winter simulation Conference*, Piscataway, New Jersey, pp. 130–143. IEEE.

Shimoyama, K., A. Oyama, and K. Fujii (2005). A new efficient and useful robust optimization approach - design for multi-objective six sigma. In *IEEE Congress on Evolutionary Computation*, Volume 1, Piscataway, New Jersey, pp. 950–957.

Singh, A. and N. D. Jana (2017). A survey on metaheuristics for solving large scale optimization problems. *International Journal of Computer Applications 170*(5), 1–7.

Spall, J. C. (2005). *Introduction to Stochastic Search and Optimization: Estimation, Simulation, and Control*, Volume 65. John Wiley & Sons.

Sterman, J. D. (2001). System dynamics modeling: tools for learning in a complex world. *California Management Review 43*(4), 8–25.

Taguchi, G. (1989). *Introduction to Quality Engineering*. American Supplier Institute.

Talbi, E.-G. (2009). *Metaheuristics: from Design to Implementation*. John Wiley & Sons.

Taleb, N. N. and B. Swan (2008). *The Impact of the Highly Improbable*. Penguin Books Limited.

Tigane, S., L. Kahloul, and S. Bourekkache (2017). Reconfigurable stochastic petri nets: A new formalism for reconfigurable discrete event systems. In *International Conference on Mathematics and Information Technology*, pp. 301–308. IEEE.

Voinov, A. and F. Bousquet (2010). Modelling with stakeholders. *Environmental Modelling & Software 25*(11), 1268–1281.

Wainer, G. A. (2017). *Discrete-Event Modeling and Simulation: a Practitioner's Approach*. CRC press.

Xu, J., E. Huang, C.-H. Chen, and L. H. Lee (2015). Simulation optimization: A review and exploration in the new era of cloud computing and big data. *Asia-Pacific Journal of Operational Research 32*(03), 1550019.