

Hybrid Simulation-Optimization Methods: A Taxonomy and Discussion

Gonalo Figueira^a, Bernardo Almada-Lobo^a

^aINESC TEC, Faculdade de Engenharia, Universidade do Porto, Porto, Portugal

Abstract

The possibilities of combining simulation and optimization are vast and the appropriate design highly depends on the problem characteristics. Therefore, it is very important to have a good overview of the different approaches. The taxonomies and classifications proposed in the literature do not cover the complete range of methods and overlook some important criteria. We provide a taxonomy that aims at giving an overview of the full spectrum of current simulation-optimization approaches. Our study may guide researchers who want to use one of the existing methods, give insights into the cross-fertilization of the ideas applied in those methods and create a standard for a better communication in the scientific community. Future reviews can use the taxonomy here described to classify both general approaches and methods for specific application fields.

Keywords: Simulation-Optimization, Taxonomy, Classification, Review, Hybrid Methods

1. Introduction

Simulation and Optimization were traditionally considered separate (or alternative) approaches in the operational research field. However, tremendous leaps in computational power promoted the appearance of methods that combined both. Simulation-based approaches started involving the optimization of the model inputs (also called controllable parameter settings). On the other hand, optimization-based approaches started using simulation for the computation of parameters (e.g. in queuing systems) or the sampling of scenarios for mathematical programming models. Nevertheless, this dichotomy is gradually vanishing, as other approaches are applying a balanced use of simulation and optimization (e.g. ROSA – see Subsection 2.3 – for a complete list of acronyms see Appendix A). The idea is to explore simultaneously the great detail provided by simulation and the ability of optimization techniques to find good or optimal solutions.

One of the main challenges hybrid simulation-optimization tries to answer is uncertainty. This aspect is addressed by a variety of (more conventional) approaches, such as stochastic programming, fuzzy programming and stochastic dynamic programming. The accuracy and detail of these models are however much lower when compared to simulation approaches. Furthermore, the difficulty in dealing with pure mathematical models leads in most cases towards the use of simulation for some computations. For instance, stochastic programming is most common in the form of scenarios (which may apply Monte Carlo simulation to perform the sampling), since the mathematical manipulation of probability distributions easily becomes intractable. Stochastic dynamic programming makes also use of simulation, when solving large complex models with the so-called reinforcement learning algorithms. A good overview of stochastic, fuzzy and stochastic dynamic programming is given by Sahinidis [1].

Another major challenge is the consideration of nonlinear relationships, qualitative aspects or even processes hardly modelled by analytical expressions. The problem tackled in [2] is an example of a completely deterministic problem that requires simulation. Indeed, the core advantage of simulation is its ability to deal with complex processes, either deterministic or stochastic, with no mathematical sophistication.

Still, combining simulation and optimization typically results in highly demanding methods in terms of computational effort, even for today's standards. Hence, the design of a good interaction is crucial. For that reason, and because the possibilities of combining them are so vast, it is very important to have a good overview of the different approaches. There is thus the need for a taxonomy which covers the full spectrum of these hybrid approaches and launches the discussion on the different strategies (their advantages and limitations).

March 17, 2014

A number of taxonomies and classifications have been proposed in the literature using different criteria. Some distinguished simulation-optimization methods by the applied techniques (e.g. statistical procedures, gradient approaches, heuristics, etc.) or their properties of convergence, optimality and correct selection [3, 4, 5]. Other frameworks focused on the optimization problem, i.e. the solution space and objective function [6, 7, 8, 9]. Shanthikumar and Sargent [10] suggested a classification schema, according to the hierarchical structure of both simulation and optimization models. The authors have further distinguished between “hybrid models” (which they classify) and “hybrid modelling” (previously classified in [11]). In the former both analytic and simulation models are combined into one single model, whereas in the latter each model is able to generate a complete solution, but the final solution results from information exchanges between their executions.

While helpful for understanding the extent of research and practice in the field, these classifications have focused only on particular streams of methods. The last two considered only the cases where an analytical model exists *a priori* (which does not happen in several simulation-optimization approaches, such as stochastic approximation). The remaining papers addressed only the optimization of simulation model inputs, commonly known as “simulation optimization” (SO). Here, we refer to “hybrid simulation-optimization” (or simply “simulation-optimization” – S-O) as any combination of these two major OR approaches.

Another aspect of those classifications that is subject to improvement is the possibility of combining different criteria. In fact, relating different dimensions and perspectives in a single classification can be critical when trying to grasp the essence of S-O methods and discover new opportunities for the cross-fertilization of ideas or the exploration of new approaches.

Finally, important criteria were overlooked. The first concerns the purpose of the simulation component in the overall design. This is the criterion that distinguishes the main streams of research in S-O. Fu [12] outlined this dimension in two main categories (“optimization for simulation” and “simulation for optimization”), but the author has not developed it further. Another key dimension is the search scheme with respect to the series of solutions and realizations considered for evaluation. We refer here to “realization” as a short sample path (or simulation run) or part of a long path. The search scheme not only separates methods that tackle deterministic problems from those that address stochastic settings, but also discriminates the different strategies for dealing with the latter.

Two other papers [13, 14] sought to create taxonomies for simulation optimization problems and methods, in order to facilitate numerical comparisons and code reuse. Nevertheless, the interaction between simulation and optimization was not discussed and their studies were confined solely to SO methods.

In light of the above discussion, we propose a comprehensive taxonomy for S-O methods. Our classifying framework comprises four key dimensions: Simulation Purpose, Hierarchical Structure, Search Method and Search Scheme. The first two are related to the interaction between simulation and optimization, whereas the other two concern the search algorithm design. Considering these four dimensions (and their full spectrum), we are able to cover the complete range of S-O methods and distinguish virtually all of them in at least one dimension. The categories of each dimension had to be created from scratch, even for those already considered in the literature, since the confrontation of multiple criteria so required. The range of S-O methods includes: “simulation optimization” (already mentioned); “simulation for optimization”, where simulation helps enhancing an analytical model; and “optimization-based simulation”, where simulation generates the solution based on the optimization output (optimization does not need any simulation feedback).

One may question whether a so ambitious taxonomy is reasonable, or if it would make more sense studying and discussing those main streams of methods separately. The issue is that in many applications, even the choice of the main approach is not straightforward and consequently requires the consideration of methods that are entirely different in spirit. Moreover, some of these methods are more similar than it might appear at first sight.

This paper makes a clear distinction between the characteristics of the problem and those of the method and suggests connections between both. Our work has therefore a threefold contribution:

- give an overview of the full spectrum of simulation-optimization approaches, providing some guidance for researchers who want to use one of the existing techniques;
- explore the characteristics of these methods, giving insights into the cross-fertilization of their ideas and showing gaps that may result in new approaches;

- create a standard for a better communication in the scientific community, either when comparing existing S-O methods or when proposing a new one.

As opposed to other papers, we start by reviewing a variety of well-known methods (in Section 2) and only then propose our taxonomy (in Section 3). We do not intend to do an extensive review. Our aim is just to provide an overview of the main methods in the literature, referring to specific examples. Nevertheless, we seek to cover their full spectrum, in order to create a comprehensive taxonomy. Section 4 presents a discussion on the different S-O strategies and their relationship with the characteristics of the problem. Finally, Section 5 summarizes the conclusions and future lines of research. Since the review and taxonomy comprise a wide range of methods and categories, we provide a list of acronyms in appendix.

2. S-O methods

We divide this section in three main parts, each corresponding to one of the three major streams of simulation-optimization research: Solution Evaluation (SE), Analytical Model Enhancement (AME) and Solution Generation (SG) approaches, plus a fourth part for related methods. The first approach corresponds to SO and is more popular in the simulation community. Here, simulation is used to evaluate solutions and hence perceiving the landscape (or response surface). The other two combine simulation with analytical models, thus being classified in the literature as “hybrid simulation-analytic models/modelling”. These are mostly adopted within the optimization community.

S-O methods can be applied to a wide range of problems in areas such as manufacturing, warehousing, transportation, logistics, services, finance, among others. In this paper we present an illustrative example of a manufacturing system, which will be used to give context to the explanation of the S-O methods.

Consider the job shop consisting of four machines and three buffers (or queues) exhibited in Figure 1. Three different products can be produced in this system and each has its own sequence of machines to visit. Each machine can only process one product at a time. Whenever a machine finishes one job, the product moves to the buffer immediately before the following machine, waiting for being processed. However, the machine cannot discharge it if the succeeding buffer is full. The size of a buffer b is determined by its number of positions S_b , each having a certain capacity a (the overall buffer capacity is then $S_b \cdot a$). The total number of positions is limited. Processing times are stochastic and follow given probability distributions. Each product has a certain demand in each day and backlog is not allowed. The aim is then to determine the (continuous) production amounts X_{it} , for each product i in each day t , and appropriate (discrete) buffer sizes S_b (with $b \in \{2, 3, 4\}$) in order to optimize the expected aggregated costs, which include production, inventory and shortage (unmet demand) costs. Two variants (simplifications) of this problem will also be considered:

- lot sizing (determine \mathbf{X}) assuming infinite buffer capacities (infinite \mathbf{S});
- buffer space allocation (determine \mathbf{S}) assuming fixed production lots (fixed \mathbf{X}).

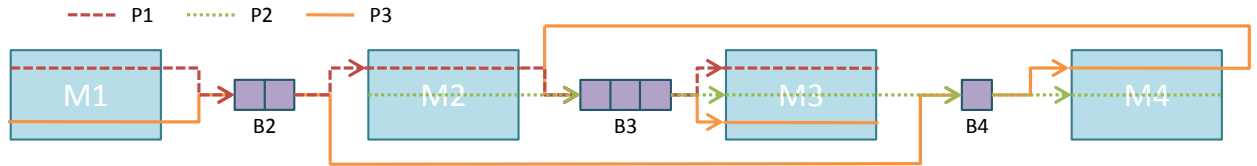


Figure 1: Production system: a job shop with three products (P1, P2 and P3), four machines (M1, M2, M3 and M4) and three buffers (B2, B3 and B4).

2.1. Solution Evaluation approaches

The first major S-O approach consists of developing a comprehensive simulation model to represent the system (such as the manufacturing system presented above) and use that model to evaluate the performance of various solutions. For instance, we could configure the system to produce given lot sizes and include certain buffers. Alternative

solutions, with different lots and buffer spaces would then be tested and compared, in order to find a good (or optimal) solution. This experiment is thus the optimization of a simulation model, which in the literature is called “simulation optimization” (SO). Here, we refer to them as Solution Evaluation (SE) methods, since the purpose of simulation is to evaluate the performance of solutions.

The general SO problem setting is as follows:

$$\min f(\theta) \quad (1)$$

$$\text{s.t. } g(\theta) \geq 0, \quad (2)$$

$$\theta \in \Theta, \quad (3)$$

where θ represents the vector of input variables, Θ its potentially feasible domain (before running any simulation), $f(\theta) = Y[F(\theta, \omega)]$ and $g(\theta) = Z[G(\theta, \omega)]$ the objective and constraint functions (respectively) determined by a simulation model, ω a sample path, F and G the direct performance measures of the simulation output (depending on both the controllable vector and stochastic effects) and Y and Z the corresponding statistics (typically the expected value).

In our problem example:

- θ includes both **X** and **S** (in each variant however it includes either one or the other);
- $f(\theta) = E[F(\theta, \omega)]$, which can be represented by

$$E \left[\sum_{t=1}^T \sum_{i=1}^3 (c_{it} \cdot RX_{it}(\theta, \omega) + h_{it} \cdot RI_{it}^+(\theta, \omega) + q_{it} \cdot RI_{it}^-(\theta, \omega)) \right] \quad (4)$$

where RX_{it} , RI_{it}^+ and RI_{it}^- are the observed production, inventory and shortage amounts, and c_{it} , h_{it} and q_{it} are the production, inventory and shortage costs, respectively;

- $g(\theta) = G(\theta, \omega)$ would include constraints such as the inventory balance equations

$$RX_{it}(\theta, \omega) + RI_{i(t-1)}^+(\theta, \omega) + RI_{it}^-(\theta, \omega) - RI_{it}^+(\theta, \omega) = d_{it} \quad i = 1...3 \quad t = 1...T \quad (5)$$

where d_{it} is the demand;

- Θ incorporates non-negativity and other *a priori* constraints, such as the total available buffer positions

$$\sum_{b=2}^4 S_b \leq N_{pos} \quad (6)$$

- ω is the set of values obtained for the (stochastic) processing times in a given realization.

Note the difference between X_{it} and RX_{it} . The former is the predefined amount, whereas the latter is that observed in the simulation (which will also depend on the buffer sizes **S** and the stochastic effects ω). There is a wide variety of methods approaching this problem. Some of the most important are following presented.

Statistical Selection Methods (SSM)

Consider the buffer space allocation problem stated above (second variant). If the number of possible solutions is not very high (let say each buffer can have one, two or three spaces, resulting in 27 combinations), then all those combinations can be evaluated in order to determine the optimal solution. However, the solution space is not the only to be explored. In fact, the problem is stochastic and thus one replication should not be enough to accurately evaluate the performance of each solution. Therefore, the number of replications for each solution (i.e. the way the probability space is explored) is also to be determined. Statistical Selection Methods, which include the well-known Ranking and Selection (R&S) and Multiple Comparison Procedures (MCP), focus on this aspect. These methods compare and select solutions applying statistical analysis, so that a given confidence is achieved in that process. The solution search typically consists of an exhaustive enumeration and is thus limited to a relatively small set of solutions (Θ) with

discrete input variables, such as in this buffer space allocation problem. At the end, R&S provides the best solution, while MCP quantify the differences in performance ($f(\theta)$) among the solutions. Traditional procedures (here referred to as SSM^a) can be enhanced (to SSM^b) by the method of common random numbers (CRN) which induces a positive correlation among the solutions [15]. In practical terms CRN favours a fairer comparison between solutions, resulting in a reduction in the necessary number of replications.

Metaheuristics (MH)

To explore large or infinite solution spaces (such as larger instances of the buffer space allocation or the lot sizing problem), “true” search algorithms are needed. Metaheuristics are high level frameworks that combine basic heuristics in order to efficiently and effectively explore the search space [16]. These algorithms may be single-solution based (e.g. simulated annealing [17]), population-based (e.g. genetic algorithms [18]) or set-based (e.g. nested partitions [19]) and can tackle both combinatorial (MH^a) and continuous optimization (MH^b). They were originally developed to address deterministic problems, even though the algorithm itself may be stochastic. Nevertheless, metaheuristics have also been applied to stochastic simulation optimization, where the evaluation of solutions is performed (multiple times for each) by the simulation model [20]. Indeed, given their flexibility to tackle any type of solution space and their ability to quickly achieve good quality solutions, these methods dominate the optimization routines of the discrete-event simulation software.

Memory-based Metaheuristics (MMH)

The difference to the previous is that while moving in the solution space, these methods construct and memorize a perception of the landscape. This memory structure is typically a probability distribution on the space of solutions which provides an estimate of where the best solutions are located. The generation of the following solutions then uses this distribution to perform a biased selection towards promising regions. Some examples include Swarm Intelligence (e.g. ant colony optimization [21]), Estimation of Distribution Algorithms [22], Cross-Entropy Method [23] and Model Reference Adaptive Search [24]. These methods are also known as *model-based* methods, as opposed to the previous methods, which are *instance-based* [25], and may work either with discrete (MMH^a) or continuous variables (MMH^b).

Random Search (RS)

RS is very close to MH (and may even be seen as the same method), where a neighbourhood can be defined for each incumbent solution. However, the next move is probabilistically chosen, based on a given probability distribution. These methods have also been originally designed for conventional deterministic problems, but have been extended afterwards to the stochastic setting. In the latter, an additional feature has to be defined: how the best solution is selected. Possible alternatives include the incumbent solution at the end of the procedure, the most visited solution and the solution with the best sample mean. The choice is closely related to the approach for dealing with noise in the stochastic setting, which can range from expending a significant amount of computer effort on each point visited (RS^a) to deciding on how to proceed based only on limited information (RS^b). Further details on RS are given in [26] and recent work is reported by [27] and [28].

Stochastic Approximation (SA)

In contrast to the previous method, SA is a gradient-based procedure and is therefore targeted at continuous variables (such as the lot sizing decisions of the problem described before). Under appropriate conditions, convergence towards local optima can be guaranteed and its rate is dramatically enhanced with the availability of direct gradients, which are problem-specific. Their absence is solved by computing naive estimations, such as finite differences or simultaneous perturbations. SA typically uses a short simulation run in each iteration, but can also perform gradient steps at intervals during a (single) long run (Single-Run Optimization – see [29]). For a comprehensive discussion on SA see [30].

Sample-Path Optimization (SPO)

SPO [31, 32] is based on the idea of going out far enough along the sample path (to have a good estimate of the limit function), fixing it for every solution and solving the resulting problem applying deterministic optimization. In

our illustrative problem a large set of processing times would be sampled and used in a long simulation run. Applying the same set of processing times to every solution (the CRN method extended to the whole domain) makes it a deterministic problem, which should be easier to solve. Indeed, the existence of effective methods for constrained deterministic optimization can be exploited here. In spite of the typical implementations (here denoted by SPO^a) being gradient-step methods, SPO is a general procedure and can be applied to many SE and AME methods.

Metamodel-based Methods

Instead of iterating through different solutions and running simulations to evaluate each of them, another approach is to first perceive the landscape in multiple points (using simulation) and then approximate a metamodel for those points. Deterministic optimization methods can then be applied to the obtained objective function, which is inexpensive to compute. This is the idea behind Metamodel-based Methods. A significant advantage of these methods, as Barton and Meckesheimer [8] advocate, is the “reduction in prediction variance by extending the effect of the law of large numbers over all points in the fitting design”. However, “this advantage comes at a cost: bias that is introduced when the metamodel fails to capture the true nature of the response surface”.

Two main strategies exist: Global Metamodel-based Methods (GMM) and Local Metamodel-based Methods (LMM). The former are typically plain SO sequential procedures and require a global optimization strategy. The latter, which include the well-known Response Surface Methodology (RSM) – see [33], alternate between model construction (from simulation responses) and deterministic optimization. For both, solutions may be evaluated with single (GMM^a and LMM^a) or multiple realizations/replications (GMM^b and LMM^b).

Gradient Surface Methods (GSM)

Combining techniques is frequently a promising avenue of research. One of these fruitful results is the GSM, which combines RSM and SA. Proposed in [34], this method implicitly utilizes a second-order design for the fitting surface (in RSM), by considering the gradient surface modelled by a first-order design. However, only a single replicate is used to obtain each estimate. The procedure switches to a pure SA when the optimum is approached [6].

Surrogate Management Framework (SMF)

There are approaches that make also use of metamodels (or surrogate models), but do not apply a deterministic search directly on them. Instead, the surrogate model is used just to guide the search, screening out poor solutions, which do not get to be evaluated by simulation, or even selecting only high-quality solutions for examination. The latter is the case of SMF. This methodology was proposed in [35] and is built on top of pattern search methods – see [36]. SMF creates a grid of points to serve as a basis for the recalibration of a surrogate model. The points are chosen to be spatially disperse, in order to improve the accuracy of the approximation, but at the same time be promising solutions. The surrogate model then predicts points at which improvements are expected. The recalibration procedure may use single (SMF^a) or multiple evaluations (SMF^b) in each point. The overall method does not require the existence of derivatives and still converges towards local optima.

Reverse Simulation Technique (RST)

The idea of RST is to specify in advance desired target values or ranges of values for particular simulation variables and run the model with expert systems guiding those variables towards their corresponding targets [37]. For instance, in our lot sizing problem we could specify that the time each machine m spends on each product i in each day t could not be higher than a given percentage of the total available time of that day. If this share was reached, then the machine would shift to another product. The adjustments thus occur during the simulation execution. RST appears to be an interesting approach to generate an initial solution of reasonable quality. Both continuous (RST^a) and discrete variables (RST^b) can be addressed by this method. In [38] the authors report an RST algorithm that finds the steady state of a system and an optimal state.

Approximate Dynamic Programming (ADP)

The parametric optimization problem defined in the beginning of this section can be extended to a control optimization problem (often called dynamic optimization), which may be stated as: $\min_{U(\cdot)} E \left[\sum_{t=1}^T C(V(t, \omega), U(V, t)) \right]$, where $V(t, \omega)$ is the state of the system and $U(V, t)$ is the corresponding control action. The solution to this problem

is not a finite dimensional vector (like θ), but a function ($U(V, t)$). The optimal control may be solved via dynamic programming (DP). However, according to Gosavi [39], stochastic DP “suffers from the curses of modelling and dimensionality”. The author states also that “these two factors have inspired researchers to devise methods that generate optimal or near-optimal solutions without having to compute or store transition probabilities”. ADP (also known as reinforcement learning or neuro-dynamic programming, in the artificial intelligence and control theory communities, respectively) is one such method. Combining DP and simulation, ADP is based on a learning agent which selects actions according to its knowledge of the environment. The latter responds by an immediate reward (the reinforcement signal). This procedure is similar to RST (in the sense that there is an agent controlling the system over the simulation), but here the agent learns and improves its actions over the process. For a brief review of ADP see [40].

Retrospective Simulation Response Optimization (RSRO)

All the aforementioned methods (with the exception of RST) deal with stochastic problems performing either one or multiple realizations for each solution. A different approach, which is the essence of RSRO [41], is to consider one common realization for every solution. Note the difference to SPO, where the latter fixes a long sample path (i.e. multiple realizations for each solution). In the buffer space allocation problem, RSRO would simulate the system with an infinite (or very large) number of spaces until the total number of used spaces reached the real availability. This would be actually the optimal solution for that realization if the objective was to minimize the time to buffer exhaustion, but not necessarily a good solution for other scenarios. Hence, the method is repeated for a number of realizations. Each realization results in a distinct solution. Therefore, the expected value of the relevant performance measure cannot be evaluated. Instead, it provides for instance the solution with the maximum likelihood of being optimal (the poor behaviour when it is not optimal is not assessed though). Like RST, RSRO seems to be appropriate for generating good solutions for further evaluation and refinement. The most natural use of RSRO applies exact methods (RSRO^a), although heuristics can also be used. The RSRO approach should not be mistaken for the more recent retrospective optimization [42], the latter being closer to SPO.

2.2. Solution Generation approaches

Using simulation to evaluate different solutions (as in every method of the previous section) can be very computationally intensive. For some particular problems, the feedback from simulation may not even be important to the choice of the solution. In those cases analytical models can be formulated and solved and their solutions simulated (optimization-based simulation), in order to compute all the variables of interest. The purpose of simulation here is not to verify the advantage of one solution over another, but simply to compute some variables and hence be part of the whole solution generation (SG).

The lot sizing problem described in the beginning of this section could be formulated as:

$$\min \sum_{t=1}^T \sum_{i=1}^3 (c_{it} \cdot X_{it} + h_{it} \cdot I_{it}^+ + q_{it} \cdot I_{it}^-) \quad (7)$$

$$\text{s.t. } X_{it} + I_{i(t-1)}^+ + I_{it}^- - I_{it}^+ = d_{it} \quad i = 1 \dots 3 \quad t = 1 \dots T, \quad (8)$$

$$\sum_{i=1}^3 p_{im} \cdot X_{it} \leq \text{cap}_{mt} \quad m = 1 \dots 4 \quad t = 1 \dots T, \quad (9)$$

$$X_{it}, I_{it}^+, I_{it}^- \geq 0 \quad i = 1 \dots 3 \quad t = 1 \dots T, \quad (10)$$

where p_{im} is the processing time (here assumed deterministic) of product i on machine m , cap_{mt} its capacity in period t and the remaining terminology is the same as that used in the previous subsection. Buffer spaces are assumed to be infinite. Note that production, inventory and shortage amounts are all decision variables, not observations of simulation. Simulation is then used to compute more realistic values for these variables.

In order to close the gap between predefined and simulated values, it would be important to reduce the production capacities (cap_{mt}) in the analytical model, since there will be waiting times between operations which are only considered in simulation. If defining these capacity reductions appears to be difficult and critical to the overall optimization, this SG approach might not be the best choice. However, if the analytical model is able to generate good solutions, this should be the best approach, especially when simulation is very expensive, since it only needs to run once. The

optimization process can be performed either before or during that simulation run. These two schemes are described below.

Solution Completion by Simulation (SCS)

In this first method simulation is used to compute some variables, which will complete or correct the solution generated by optimization. In our example the model stated by (7)–(10) is solved for the entire horizon, resulting in certain production (X_{it}), inventory (I_{it}^+) and shortage (I_{it}^-) amounts. This solution is fed to simulation, which gives more accurate values for these variables. The literature applies SCS to different types of problems. Briggs [43] approached the problem of mass tactical airborne operations. The analytical model generates a solution under ideal conditions. Simulation then considers the inherent variability and provides the expected, best and worst outcomes. Other applications of SCS somehow hybridize it with SE methods. For instance, Lim et al. [44] determine a production-distribution plan with an analytical model and simulate that plan to obtain more realistic values. However, if the plan is not acceptable, the replenishment policy is changed and simulation is run again. Truong and Azadivar [18] embed an SCS into an MH to solve a supply chain design problem. Their genetic algorithm determines some qualitative decisions. Each chromosome is decoded by an SCS method, where first an analytical model is solved to determine other decisions and then simulation is used to evaluate the complete solution. The popular commercial optimization module OptQuest also allows applying this type of hybridization.

Iterative Optimization-based Simulation (IOS)

Instead of a simply sequential procedure, optimization may be called during simulation execution. For instance the above analytical model can be solved in a rolling-horizon basis, simulating one period (t) at a time, updating the inventories and shortages of that period and solving the analytical model from that period until the end of the horizon. The new solution is then re-primed in simulation, which advances another period (to $t + 1$). In this way the output of optimization should be closer to that of simulation, since the former keeps track of the latter over the horizon. This iterative process is applied by Subramanian et al. [45]. However, in their case optimization is not called in a periodic scheme, but in an event-driven basis. Whenever the simulation module encounters a need for a control action, it momentarily suspends itself and communicates the state of the system to the optimization module, which resolves it. This approach may resemble RST. However, here the optimization phase is based on an analytical model and does not need simulation to evaluate its actions. In fact, RST obtains feedback from simulation, whereas IOS only receives feedforward. Furthermore the need for adjustments does not result from the specification of targets.

2.3. Analytical Model Enhancement approaches

Solution Generation methods can be very efficient, but if the feedback from simulation proves to be important, they may not be effective. Therefore, as mentioned above, they often end up being hybridized with Solution Evaluation methods. Another possibility is to enhance the analytical model using the simulation results. This analytical model enhancement (AME) can be conducted in different ways.

Stochastic Programming Deterministic Equivalent (SPDE)

Consider the lot sizing problem previously stated, but with only one machine. The waiting times issue is now simple to address. The only difficult aspect that remains is the uncertainty of processing times. In this situation, stochastic programming appears to be appropriate. Still, since mathematically manipulating probability distributions can be difficult, one may resort to Monte Carlo simulation to perform the sampling of scenarios, which are later embedded in a single analytical model. The final model is then the same as before, but where constraints (9) are replaced by:

$$\sum_{i=1}^3 p_{iw} \cdot X_{it} \leq cap_t \quad t = 1 \dots T \quad w = 1 \dots W, \quad (11)$$

where w represents a given scenario, for which all processing times p_{iw} are sampled from their corresponding distributions. This model is called a large-scale deterministic equivalent, since introducing the scenario index may increase the number and size of constraints by multiples. The general procedure is known as sample average approximation

(SAA – see [46]) and includes SPO (which fixes a long sample path), but also the aggregation of realizations of multiple paths. The method starts with a given sample size, which increases until required confidence and variance levels are achieved. An example is given in [47]. These SPDE approaches typically require some assumptions / simplifications to be made, when compared to S-O using discrete-event simulation. Nevertheless, the work by Schruben [48] may change this scenario (see Subsection 2.4).

Recursive Optimization-Simulation Approach (ROSA)

If the SPDE method was applied to the original system, consisting of four machines, it would probably fail to provide a good solution, since it would not consider waiting times. In those cases the output of a discrete-event simulation model should be helpful to assess those parameters and thus correctly estimate a proper capacity reduction for the analytical model. That is the idea behind ROSA. This approach was firstly proposed in [49] and consists of running alternately a (typically linear) deterministic analytical model, such as (7)–(10), and a (typically stochastic discrete-event) simulation model. Simulation uses the solution generated by optimization and computes particular performance measures (in our case, throughput times). The values of these measures are then introduced again into the analytical model, refining its parameters (e.g. capacity reductions, processing times). The iterative process ends after a stopping criterion is met (e.g. convergence of the solution, parameters, objective). It should be noted that even if the processing times were deterministic, simulation would still be needed, due to the waiting times.

There are several implementations of this approach, applying different stopping criteria and refinement strategies. The variants in [2], [50] and [51] are some interesting examples. The first two tackle pure deterministic problems (ROSA^a), whereas the last one approaches a stochastic environment (ROSA^b). In the latter case results from multiple simulation runs have to be aggregated (typically by quantiles, which may reflect different risk-averse levels). Some authors have criticized these iterative procedures, claiming that convergence can be problematic [52]. Still, most of the studies have found convergence in practice after a few iterations. Acar et al. [53] propose a ROSA framework close to SE methods. The authors define an analytical model with an optimistic objective function. Simulation evaluates each solution generated (under more realistic conditions) and corrects the objective function for that particular solution. This ROSA variant is one of the few S-O methods that is optimal. Nevertheless, it is geared to discrete solution spaces.

Function Estimation based Approach (FEA)

An alternative to the iterative procedure of ROSA is the initial estimation of (often nonlinear) functions that describe the relationship between particular input and output variables. In our lot sizing problem the relationship between lot sizes X_t and production lead-times $lt(X_t)$, which include processing and waiting times, is not trivial and needs simulation to be estimated. By performing a set of simulations, this dependency can be characterized and included in the analytical model, specifically in the capacity constraints:

$$lt(X_t) \leq cap_{mt} \quad m = 1...4 \quad t = 1...T, \quad (12)$$

All the other constraints and the objective function remain the same. In this way, the method is not dependent on the convergence between simulation and optimization, like ROSA. However, this comes at the cost of a more difficult analytical model to be solved. An example of FEA is implemented in [54]. The authors conduct an initial simulation study to specify the nonlinear relationship between the expected work-in-progress and the expected throughput. This relationship is reproduced in what are called “clearing functions”. The final analytical formulation incorporates an approximation of these concave functions by outer linearization.

Optimization-based Simulation with Iterative Refinement (OSIR)

ROSA assumes that once the decisions are made, they cannot be revised. However, in practice they often can. If that is to be considered, SPDE for instance has to be extended to a multi-stage stochastic program. This increases considerably the scale of the model to be solved, which easily becomes intractable. An alternative is proposed by Jung et al. [55], who approach a supply chain management problem under demand uncertainty. The authors have devised a framework similar to IOS, but that performs refinements to the analytical model. The latter is thus solved in rolling-horizon within the simulation model (optimization-based simulation). This procedure is repeated multiple times, performing (every n iterations) the appropriate refinements to the safety stock levels, in order to accommodate

the uncertainty of demand. The refinement process, however, is not straightforward in their case. Indeed, unlike the deterministic problems where capacity reductions directly result from simulated waiting times, in this stochastic problem the right adjustments are to be determined. For this purpose, the authors apply a gradient-based search. Their method can thus be seen as an IOS embedded in an SA method. The input variables of the latter are the refinement parameters in the former (the safety stocks).

2.4. *Methods related to S-O*

Before concluding this section, we would like to make a reference to methods that are closely related to S-O, but may not be exactly a hybridization between simulation and optimization. Schruben [48] proposed modelling sample paths from event relationship graphs, which are used to simulate discrete event systems (DES) as the solutions to mathematical programming (MP) models. In [56] the authors provide a procedure that generates these optimization models directly from the explicit event relationships and examine several examples and potential applications. The objective of the MP models, as in the simulation, is simply to execute events as early as possible. Hence, they are seen as simulation models. The solution of each MP model represents a single simulation run. This procedure does not fit in the S-O definition given in Section 1, since it does not combine simulation and optimization. Instead, it replaces the former by the latter. Thus, it can be called “simulation by optimization” (SbO).

SbO has two main advantages over conventional simulation. First, linear programming duality can be used to perform sensitivity analysis, which is simpler than the traditional way of computing sample path derivatives. Moreover, as Chan and Schruben [56] explain, the solution obtained from linear programming may provide more information for a single simulation run because other perturbed sample paths can be reached from the current sample path by some additional computation (pivots), which may be easier than running a new simulation.

The other key advantage is related to the application of this approach to S-O. SbO promotes a deeper integration between simulation and optimization. Indeed, the constraints that describe the system dynamics can be embedded within an optimization model that also determines decisions (in addition to simulating the system). If the concept of SAA is also applied, a single optimization model can be enough to perform the whole S-O. This can be seen as an SPDE, but less restricted by assumptions that oversimplify the problem.

Nevertheless, SbO has a major issue to be addressed: computational time. If one SbO run can provide more information than one conventional simulation run, it is also true that it may consume significantly more time. The MP model may have integer variables which highly increase the computational burden. To overcome this difficulty, Alfieri and Matta [57] propose approximate representations for simulation, based on time buffers. In [58], the authors apply a time-based decomposition algorithm to further reduce the computational effort.

3. **S-O taxonomy: a new unified framework**

The proposed framework for classifying simulation-optimization methods is composed of four dimensions:

1. Simulation Purpose,
2. Hierarchical Structure,
3. Search Method and
4. Search Scheme.

The first two are related to the interaction between simulation and optimization, whereas the other two concern the search algorithm design. In the following subsections we describe all the categories of each dimension and combine the four dimensions, two by two. The two matrices that result are then used to classify all the previously reviewed methods.

3.1. *Interaction between simulation and optimization*

The categories for Simulation Purpose correspond to the main streams of research in S-O:

- Evaluation Function (EF) – iterative procedures that use simulation to evaluate solutions and hence guide the search, validating its moves;

- Surrogate Model Construction (SMC) – methods which apply simulation for the construction of a surrogate model, which is either used to guide the search or is directly searched;
- Analytical Model Enhancement (AME) – approaches making use of simulation to enhance a given analytical model, either by refining its parameters or by extending it (e.g. for different scenarios);
- Solution Generation (SG) – methods where a simulation model generates the solution, using optimization for some computations.

The first two categories compose the Solution Evaluation (SE) approaches. The large variety of these methods and the core difference of having (or not) a surrogate model leads to this distinction.

Some of these streams of methods may in some situations appear quite similar. In both SMC and AME an analytical model is improved with the output of simulation. However, the key difference is that the former uses only evaluations of the objective function to update a problem-independent model, whereas the latter starts from a problem-specific model which is modified according to the simulation output. Similarly, SG may be very close to SE approaches (e.g. IOS vs. RST – see Subsection 2.2), although belonging to different streams of research. The importance of integrating all S-O approaches into one single classification is therefore evident.

For the Hierarchical Structure dimension, four other categories were defined:

- Optimization with Simulation-based Iterations (OSI) – in all (or part) of the iterations of an optimization procedure, one or multiple complete simulation runs are performed;
- Alternate Simulation-Optimization (ASO) – both modules run alternately and in each iteration, either both run completely or both run incompletely;
- Sequential Simulation-Optimization (SSO) – both modules run sequentially (either optimization following simulation or the opposite);
- Simulation with Optimization-based Iterations (SOI) – in all (or part) of the iterations of a simulation process, a complete optimization procedure is performed.

Figure 2 exhibits the combination of these two dimensions and the corresponding classification of each method. According to the categories previously defined, there is a trend along both axes of the matrix: the autonomy of the optimization procedure tends to grow and the total number of simulations, compared to the number of optimization iterations, tends to decrease. There are however cells that are not-applicable, particularly at two corners:

- top right corner – calling the simulation process multiple times (either in OSI or ASO), when only one solution is to be generated and no feedback is necessary;
- bottom left corner – simulation being the evaluation function of the optimization procedure and not being (either completely or partially) in the iterations of the latter.

The number of categories in both dimensions could be larger. For instance, SMC methods could be further divided into those that base all their search in the surrogate model (e.g. LMM, GMM) and those which use it as a supporting tool to guide the search (e.g. MMH, SMF). Likewise, AME and SG methods could be divided according to the extent of the analytical and simulation models (which may result in a hybrid model or hybrid modelling – see [10]). That is the essential difference between SPDE and ROSA, for instance. Also, the ASO and SSO structures include more than one possibility, as described above. Nevertheless, this level of aggregation in both dimensions was kept in order to simplify the analysis and favor a global view over the whole spectrum of methods.

Finally, another aspect worth mentioning is the fact that GSM appears in more than one cell. Indeed, hybrid methods, such as this, may embed different S-O interaction schemes and hence, fill more than one cell in the matrix. Subsection 3.3 explains how these methods can be fully classified.


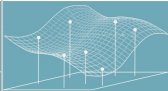
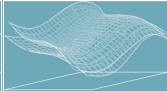

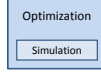
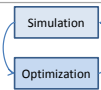
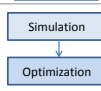
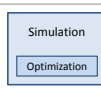
		Simulation Purpose			
		Solution Evaluation (SE)		Analytical Model Enhancement (AME)	Solution Generation (SG)
		Evaluation Function (EF)	Surrogate Model Construction (SMC)		
					
Hierarchical Structure	Optimization with Simulation-based Iterations (OSI)		SSM, MH, RS, SA, SPO ^a , GSM	MMH	Not-applicable
	Alternate Simulation-Optimization (ASO)		RST, RSRO	LMM, GSM, SMF, ADP	
	Sequential Simulation-Optimization (SSO)		Not-applicable	GMM	FEA
	Simulation with Optimization-based Iterations (SOI)			OSIR	IOS

Figure 2: Classification according to the interaction between simulation and optimization: simulation purpose and hierarchical structure.

3.2. Search design

Four types of methods are distinguished:

- Exact (E) – methods which achieve and prove global optimality (e.g. exhaustive and implicit enumeration, Dantzig's simplex algorithm, branch-and-bound, column generation, etc.);
- Derivative-Based Heuristic (DBH) – heuristic methods (i.e. procedures that cannot guarantee an optimal solution) which are based on derivative information, thus targeting continuous variables, to make progress in their search (e.g. gradient-step methods, Hessian-based methods, etc.);
- Other Continuous Heuristic (OCH) – heuristic methods that tackle continuous variables without requiring derivative information (e.g. pattern search, Nelder-Mead's simplex method, particle swarm optimization, etc.);
- Discrete Heuristic (DH) – heuristic methods that tackle discrete variables (e.g. neighbourhood-based meta-heuristics, greedy heuristics, etc.).

When defining these four categories, we have looked at the major dichotomies in Optimization: exact vs. heuristic; and continuous vs. discrete (or combinatorial). Continuous can be further divided into derivative-based or other. Given the difficulty of most problems approached by S-O, the exact methods are not so frequently applied and were thus aggregated into one single category.

The search scheme concerns the sequence of solutions and realizations considered for computation. Four schemes exist:

- One realization for each solution (1R1S) – moving in the solution and probability spaces simultaneously (i.e. changing the realization whenever a new solution is generated);
- Different realizations for each solution (DR1S) – successively considering multiple realizations for each solution that is generated;
- Common realizations for each solution (CR1S) – successively considering multiple realizations for each solution that is generated (the realizations are the same for all solutions);

- One realization for multiple solutions (1RMS) – repeatedly applying the same realization to multiple solutions (the former may change after a series of solutions).

The last scheme is the framework of RSRO and includes methods that approach deterministic problems. If that is the case, the realization does not change during the entire procedure, as there is one single scenario. The second and third schemes (the latter being the essence of SAA and SPO) involve two variants each: the consideration of a constant or a variable (typically increasing) number of realizations. Fully sequential procedures in SSM, where in each iteration a single observation is added to each alternative solution that is still in the running, are examples of the second variant.

Figure 3 presents the matrix that combines these two dimensions and the corresponding classification of each method. An important observation to make is the appearance of many S-O methods in multiple cells each. Indeed, some of the methods proposed in the literature have a clearly specified procedure (e.g. SA), while other methods are more general approaches (e.g. ROSA). Nevertheless, the different variants identified in the literature were properly discriminated with different labels in the previous section. The identified variants are not exhaustive though. For instance, SPDE may use metaheuristics to find solutions, which should not be confused with MH methods (in the latter the metaheuristic is the S-O method, whereas in the former it concerns just the optimization part).

		Search Method			
		Exact (E)	Continuous-space Heuristic (CH)		Discrete-space Heuristic (DH)
			Derivative-Based Heuristic (DBH)	Other Continuous-space Heuristic (OCH)	
Search Scheme	One realization for each solution (1R1S)	IOS, OSIR	SA, GSM, LMM ^a , GMM ^a	SMF ^a	RS ^b , ADP
	Different realizations for each solution (DR1S)	SSM ^a , SCS, ROSA ^b	LMM ^b , GMM ^b	MH ^b , MMH ^b , SMF ^b	MH ^a , MMH ^a , RS ^a
	Common realizations for each solution (CR1S)	SSM ^b , SPDE, FEA	SPO ^a		
	One realization for multiple solutions (1RMS)	RSRO ^a , ROSA ^a		RST ^a	RST ^b

Figure 3: Classification according to the search design: method and scheme.

3.3. Complete classification

Our taxonomy was used to categorize well-known methods proposed in the literature, as well as some of their specific variants. However, the purpose is that it may be used to classify any S-O framework that appears, either being an adaptation of the existing methods or a new approach. The complete classification would then connect the four dimensions, resulting in [SimulationPurpose]-[HierarchicalStructure]/[SearchMethod]-[SearchScheme]. For instance, a genetic algorithm (GA) that evaluates individuals (represented by discrete solutions) by running simulations is a MH method and can be classified as EF-OSI/DH-DR1S. Some approaches apply however a hybridization of interactions and methods. An example is the GSM, which combine SA (EF-OSI) and RSM (SMC-ASO). GSM can then be classified as <EF-OSI,SMC-ASO>/DBH-1R1S. Some hybridizations are less evident. For instance, a GA that tackles both discrete and continuous variables is also considered a hybrid method. In this case the classification would be EF-OSI/<DH,OCH>-DR1S.

4. S-O strategies discussion

In this section we examine and discuss different S-O strategies in each of the dimensions of the proposed taxonomy. The strategies depend on the characteristics of the problem that is approached. Therefore, we try to understand these relationships. The following subsections explore every classifying dimension. Figure 4 summarizes some evident

connections that result from the analysis of the taxonomy. Other connections are possible though, since there is a wide range of aspects that can influence the choice of an S-O design. Moreover, the diversity of S-O methods and the possibilities of modifying and adjusting them are vast, which makes it difficult to provide definite statements in this regard. Nevertheless, we believe this discussion can contribute to a better understanding of this field, similarly to the classifications proposed in the literature that relate S-O methods to problem characteristics (see Section 1). Indeed, the schema in Figure 4 can be seen as an extension of those classifications (considering additional dimensions and categories).

4.1. Simulation purpose

Looking at the first dimension, the major streams of research in S-O can be distinguished. The SE approach assumes that the landscape of solutions is not known (and has to be perceived), whereas in AME it is partially known or known with uncertainty (and has to be improved) and in SG it is completely known (it naturally results from the problem and possibly some simulation). The first is applied in the optimization modules included in commercial simulation software, since it considers the simulation model as a black-box (see [59] for further details). Nevertheless, when designing an S-O algorithm those assumptions do not need to reflect the real context.

It is evident that SE approaches, relying their search only on the output from a (detailed) simulation model, should perform a better selection of candidate solutions. That is longer true for EF than for SMC, since the latter may compute interpolations / extrapolations. Therefore, one may opt for SE approaches, rather than AME, even if the landscape is partially known. The downside of this strategy is that a great amount of simulation runs may be needed, hence increasing computational time. The issue of selecting the appropriate approach is thus related to a trade-off between solution quality and efficiency and how the consideration of an initial analytical model responds to that. This is what we mean by “useful analytical model” in the first question of Figure 4.

Typically, when a linear relationship between most input and output variables exists and is known, but there are particular aspects difficult or undesirable to be included in the analytical model, AME can be very efficient and effective (see ROSA and OSIR implementations). If no relationship is known and simulation is costly, either because of the need to perform many replications or because each replication is expensive, then SMC may be adequate (see for instance [60], where the authors reported a GMM implementation that outperformed OptQuest, in terms of number of replications and final solution quality). In the case of simulation being relatively modest, EF may fit well. The selection of SG approaches is more clear: they are suitable when the analytical model does not depend on any simulation feedback (although it may depend on simulation feedforward – see Subsection 2.2). The trend (along the axis of this dimension) of decreasing number of simulation runs validates these observations.

4.2. Hierarchical structure

The hierarchical structure between simulation and optimization concerns the dependence of one component on the other. This dependence is related to the previous dimension, since for each category of the latter, there are some structures that are typical and others which may not make sense (see Subsection 3.1). For instance, EF methods are usually associated with OSI structures, but not with SSO or SOI, since simulation works as an evaluation function and hence needs to be called in the iterations of optimization. AME approaches, on the other hand, are more flexible and may involve any structure. The choice will typically regard the dependence of the analytical model on the refinements performed by simulation. For example, FEA fully enhances the analytical model just in the beginning, while ROSA does it recurrently, thus being suitable for applications where the parameters depend on the solution. The former is also able to deal with that issue, but they typically complicate the model introducing nonlinear relationships. OSIR requires the optimization problem to be solved multiple times during simulation. This analysis suggests that along this axis the optimization problem should be of decreasing complexity, since it may be further complicated or called more times. Still, it is difficult to make a global conclusion and hence, we chose to leave it as an open (and more conservative) question in Figure 4, which does not directly relate to problem characteristics. Indeed, it only questions about the dependence of one component on the other.

4.3. Search method

The search method focuses on the optimization problem (but not on the interaction with simulation). The categories defined for this dimension are aligned with the characteristics of the target optimization problem. Therefore,

the correspondence is straightforward. Exact methods are appropriate when the problem is relatively easy (e.g. simple combinatorial, linear, quadratic). Large complex problems naturally require the use of heuristics. If the entire solution space is discrete, DH methods are suitable. For continuous optimization, either DBH or OCH methods can be used, although the former are more effective if derivatives exist. In mixed integer-continuous problems a hybrid solution method, combining DH with DBH/OCH methods, can be necessary. An example is given in [61], where the authors propose an algorithm that hybridizes a GA with a pattern search method. Alternatively, heuristics may be combined with exact methods: DH+E, E+DBH or E+OCH. The first is suitable when the combinatorial part is difficult and the continuous part is linear (or simple nonlinear), which typically occurs in AME or SG approaches. The remaining are more appropriate when the combinatorial part is simple and the continuous part is complex nonlinear, most common in SE. The distinction between simple and difficult combinatorial problems typically concerns the existence of polynomial-time optimal algorithms and the problem size.

In the discrete-space case another distinction could be made. The decision variables can be integer or binary. The latter are typically related to qualitative decisions, which may activate sub-options that also need to be optimized. These “simulation configuration” problems [62] can be seen as an extension of SO, since not only the simulation input parameters need to be optimized, but also its configuration (entities or layout of the simulation model). For such problems, optimization becomes more difficult because mathematical programming, hill climbing and stochastic approximation methods are not usually applicable. In addition, there is the need for automatic generation of simulation models according to a systematic process [63]. To tackle these problems, Azadivar [63], Truong [18] and Pierrel and Paris [62] suggest evolutionary / genetic algorithms (a particular type of DH).

4.4. Search scheme

The selection of the search scheme is mainly related to the relative difficulty in travelling in both solution and probability spaces and to the trade-off between convergence and diversification. Therefore, if it is harder to move in the solution space, either DRIS or CRIS should be the correct choice. This is the case of approaches like ROSA, where each iteration requires the entire resolution of an analytical model. The difference between DRIS and CRIS is that the former focuses more on diversification whereas the latter provide faster convergence. On the other hand, if moving in the probability space appears to be more difficult (or non-existent, in the case of deterministic problems), then IRMS would be more appropriate. This is the foundation of the RSRO approach: for problems such as the well-known Newsvendor problem (see [64]), it can be more convenient to consider all the solutions before changing the realization. Finally, if moving in both spaces is similar in terms of effort, then IRIS may be the best option. This is the approach of some methods which are very focused on diversification issues in stochastic settings, such as SA and RS^b. In SMC methods (such as LMM or GMM) greater diversification is achieved with DRIS and IRIS. Both obtain a greater variety of realizations and the latter allows simulating more solutions. There is thus a trend of decreasing diversification (or increasing convergence) along the axis of this dimension, as suggested in Figure 4. Nevertheless, there is another aspect that should be taken into consideration when choosing the search scheme: the use of variance reduction techniques (VRTs) or SSM ideas, which require DRIS/CRIS schemes.

VRTs are methods that aim at reducing the variance of results for a given number of replications. This means that for the same confidence level, the application of VRTs may allow reducing the number of replications and hence, the computational effort. Several VRTs exist, such as the common random numbers [65], antithetic variates [66], control variates [67], importance sampling [68] and stratified sampling [69]. The first is the most simple and widely used VRT and consists on applying the same sets of random number streams to every solution. If this is applied to the complete optimization procedure, we are in the presence of CRIS. Still, it can be applied to each iteration separately, within a DRIS scheme. For instance, in a single-solution based MH all neighbours in a given iteration would be evaluated with the same realizations, but these would change in the following iterations.

In addition to VRTs, SSM principles may be applied in SE approaches. The idea of discarding simulation replications for solutions which quickly prove (statistically) to be of poor quality is very attractive. It is not necessary though that the statistical proof involves a good confidence level (such as 90% or higher). Indeed, it may even be desirable working with a low level, in order to promote diversification in the search procedure. Still, solutions of too low quality would be quickly rejected, saving computational time.

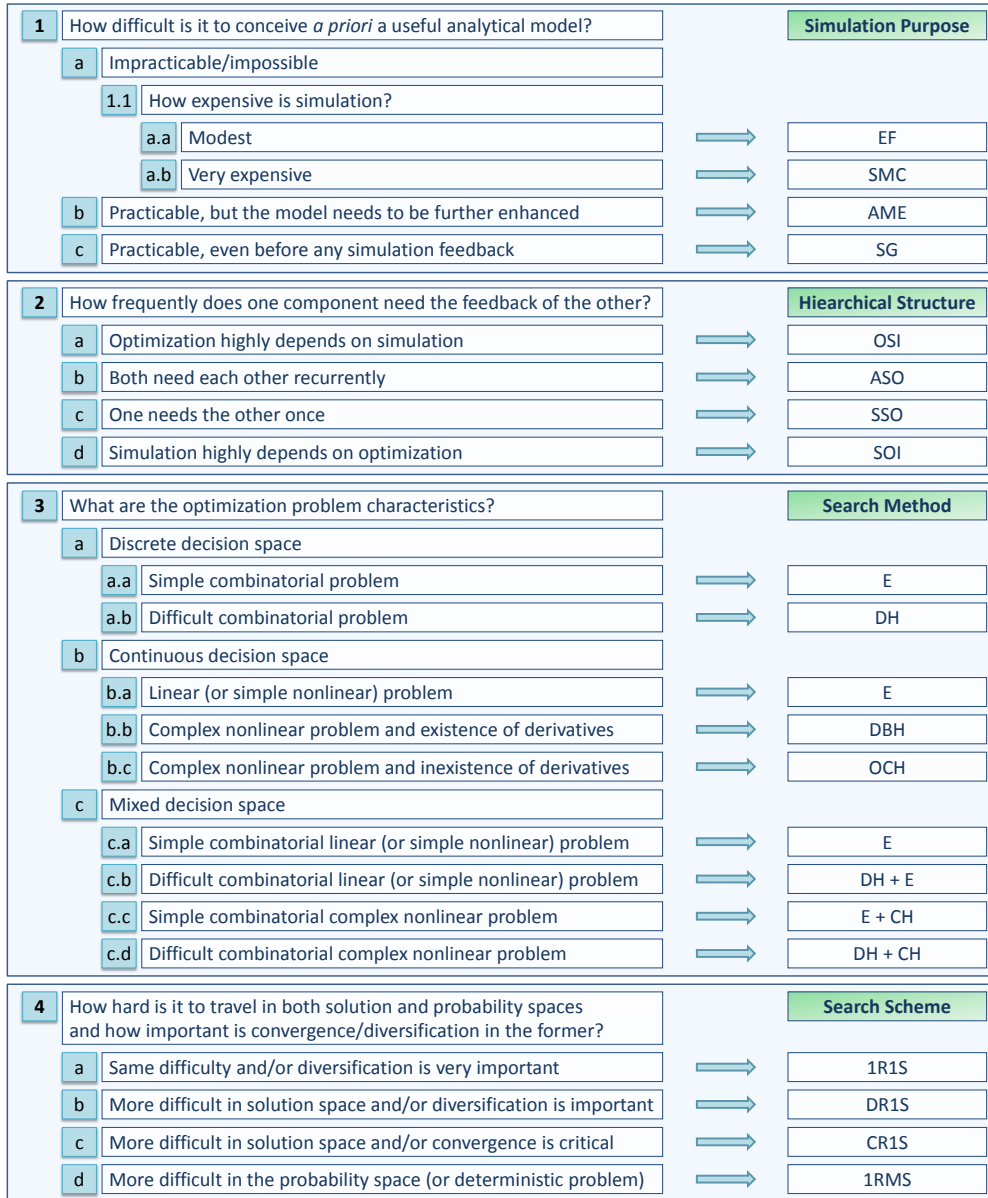


Figure 4: Evident connections between the problem characteristics and the classifying dimensions and categories.

5. Conclusion and future lines of research

In the S-O research field the dichotomy between “simulation-based” and “optimization-based” approaches is vanishing. Some of the main streams of methods may in some situations appear quite similar, as previously shown. Thus, the choice of the main approach may not be straightforward and require the consideration of methods that are different in spirit. Indeed, the boundaries of these methods can be fuzzy. Some categorization is important though. It is also crucial that it covers all S-O approaches, so that a comprehensive view is obtained. That was the aim of our taxonomy.

The classifications used so far in the literature focused on particular streams of methods and typically utilized one single criterion. We propose a classification which relates multiple dimensions and perspectives in an attempt to grasp the essence of S-O methods and discover new opportunities for the cross-fertilization of ideas and the exploration of

new approaches. Two new key criteria were considered, which greatly contributed to the discussion. The organization of criteria in two matrices allowed the examination of both the interaction between simulation and optimization and the search design. We were also able to explore the characteristics of each method and distinguish virtually all of them in at least one dimension. Additionally, common trends along different axes were identified. The discussion of each classifying dimension has resulted in some guidelines which can be used when selecting appropriate S-O designs for given problems.

The variety of methods reviewed in Section 2 reveals an immense range of possibilities for combining simulation and optimization. Some methods have a clearly specified procedure, while others are more general approaches. Nevertheless, research activities in this promising area are likely to increase significantly in the near future. We provide here some directions for further research:

- Embedding ideas from RS^b into MH frameworks. The former are very focused on the stochasticity issues, moving simultaneously in both solution and probability spaces. The latter contain a large variety of diversification and intensification strategies. An interesting framework is proposed in [70], where exploration, exploitation and estimation are carefully balanced. Another example is the work of Xu et al. [71] whose method consists of a global-search phase, followed by a local search phase, and ending with a “clean-up” (selection of the best) phase. Nevertheless, these approaches are still in their infancy.
- Applying VRTs in DRIS/CRIS schemes. As previously discussed, these techniques may allow reducing computational effort when performing simulation replications. Contrary to what might appear, VRTs can be applied not only in SE approaches, but also in AME. Indeed, even when the aim of simulation is not evaluating solutions, the feedback of the former may take advantage of a reduced variance to better enhance the analytical model.
- Using SSM principles in other SE approaches. As opposed to the previous point, this idea can only be employed in SE approaches. The increase of efficiency that result from SSM may improve the performance of DH methods, such as RS and MH. This promising combination is being explored both in the literature (e.g. [72]) and in practice (e.g. by OptQuest). Nevertheless, there are still some questions to be answered. For instance, Almeder and Hartl [73] show that in their case a statistical selection with 95% of confidence performs worse than a sample average of 10 replications, since diversification is spoiled in the former. Therefore, the choice of a good confidence level is still an open question.
- Filling gaps in the matrices. Two obvious gaps in the Interaction matrix exist: SMC-SOI and AME-OSI. The former could be some kind of hybridization between GMM and ADP, where during simulation a global meta-model would be constructed (instead of simply reinforcing a function). The latter might result in a progressive refinement strategy, where an analytical model would be refined during its resolution. Regarding the Search Design matrix blanks, CRIS schemes could be an interesting approach for metaheuristics with difficult convergence (which is more likely in the stochastic setting).

To conclude, our taxonomy fully classifies any S-O method, including hybridizations. Therefore, it can contribute to a better communication in the scientific community. Future reviews can use the dimensions here described to classify additional S-O methods or extend them for particular streams of research. For instance, in ROSA approaches it may be important to distinguish different refinement strategies and stopping criteria. More in-depth studies may also allow better characterizing the relationship between S-O problems and methods (or strategies). There is a wide avenue of research in simulation-optimization, particularly regarding the study of specific application fields.

Appendix A. List of acronyms

General:

OR	Operations Research
SO	Simulation Optimization
S-O	Simulation-Optimization
SbO	Simulation by Optimization
DES	Discrete Event System
MP	Mathematical Programming

Simulation purposes:

SE	Solution Evaluation
EF	Evaluation Function
SMC	Surrogate Model Construction
AME	Analytical Model Enhancement
SG	Solution Generation

Hierarchical structures:

OSI	Optimization with Simulation-based Iterations
ASO	Alternate Simulation-Optimization
SSO	Sequential Simulation-Optimization
SOI	Simulation with Optimization-based Iterations

Search methods:

E	Exact
CH	Continuous-space Heuristic
DBH	Derivative-Based Heuristic
OCH	Other Continuous-space Heuristic
DH	Discrete-space Heuristic

Search schemes:

IRIS	One realization for each solution
DRIS	Different realizations for each solution
CRIS	Common realizations for each solution
IRMS	One realization for multiple solutions

Simulation-optimization methods:

SSM	Statistical Selection Methods
R&S	Ranking and Selection
MCP	Multiple Comparison Procedures
MH	Metaheuristics
GA	Genetic Algorithm
MMH	Memory-based Metaheuristics
RS	Random Search
SA	Stochastic Approximation
SPO	Sample Path Optimization
GMM	Global Metamodel-based Methods
LMM	Local Metamodel-based Methods
RSM	Response Surface Methodology
GSM	Gradient Surface Methods
SMF	Surrogate Management Framework
ADP	Approximate Dynamic Programming
RST	Reverse Simulation Technique
RSRO	Retrospective Simulation Response Optimization
SPDE	Stochastic Programming Deterministic Equivalent
SAA	Sample Average Approximation
ROSA	Recursive Optimization-Simulation Approach
FEA	Function Estimation based Approach
OSIR	Optimization-based Simulation Iterative Refinement
SCS	Solution Completion by Simulation
IOS	Iterative Optimization-based Simulation

Stochastic techniques:

VRT	Variance Reduction Technique
CRN	Common Random Numbers

References

- [1] N. V. Sahinidis, Optimization under uncertainty: state-of-the-art and opportunities, *Computers & Chemical Engineering* 28 (2004) 971–983.
- [2] E. Albey, U. Bilge, A hierarchical approach to fms planning and control with simulation-based capacity anticipation, *International Journal of Production Research* 49 (11) (2011) 3319–3342.
- [3] Y. Carson, A. Maria, Simulation optimization: methods and applications, in: *Proceedings of the 1997 Winter Simulation Conference*, 1997, pp. 118–126.
- [4] J. Banks, J. S. Carson, B. L. Nelson, D. M. Nicol, *Discrete-Event System Simulation*, 3rd Edition, Prentice Hall, 2000.
- [5] M. C. Fu, Simulation optimization, in: *Proceedings of the 2001 Winter Simulation Conference*, Vol. 1, 2001, pp. 53–61.
- [6] M. C. Fu, Optimization via simulation: A review, *Annals of Operations Research* 53 (1994) 199–247.
- [7] E. Tekin, I. Sabuncuoglu, Simulation optimization: A comprehensive review on theory and applications, *IIE Transactions* 36 (11) (2004) 1067–1081.
- [8] R. R. Barton, M. Meckesheimer, Chapter 18 metamodel-based simulation optimization, in: S. G. Henderson, B. L. Nelson (Eds.), *Simulation*, Vol. 13 of *Handbooks in Operations Research and Management Science*, Elsevier, 2006, pp. 535–574.
- [9] A. Ammeri, W. Hachicha, H. Chabchoub, F. Masmoudi, A comprehensive literature review of mono-objective simulation optimization methods, *Advances in Production Engineering & Management* 6 (2011) 291–302.
- [10] J. G. Shanthikumar, R. G. Sargent, A unifying view of hybrid simulation/analytic models and modeling, *Operations Research* 31 (6) (1983) 1030–1052.
- [11] F. Hanssmann, G. Diruf, W. Fischer, S. Ramer, Analytical search models for optimum seeking in simulations, *Operations Research Spektrum* 2 (2) (1980) 91–97.
- [12] M. C. Fu, Feature article: Optimization for simulation: Theory vs. practice, *INFORMS J. on Computing* 14 (3) (2002) 192–215.
- [13] R. Pasupathy, S. G. Henderson, A testbed of simulation-optimization problems, in: *Proceedings of the 2006 Winter Simulation Conference*, 2006, pp. 255–263.
- [14] J. Han, J. A. Miller, G. A. Silver, Sopt: Ontology for simulation optimization for scientific experiments, in: *Proceedings of the 2011 Winter Simulation Conference*, 2011, pp. 2909–2920.
- [15] J. R. Swisher, S. H. Jacobson, E. Yücesan, Discrete-event simulation optimization using ranking, selection, and multiple comparison procedures: A survey, *ACM Trans. Model. Comput. Simul.* 13 (2) (2003) 134–154.
- [16] C. Blum, A. Roli, Metaheuristics in combinatorial optimization: Overview and conceptual comparison, *ACM Comput. Surv.* 35 (2003) 268–308.
- [17] T. M. Alkhamis, M. A. Ahmed, V. K. Tuan, Simulated annealing for discrete optimization with estimation, *European Journal of Operational Research* 116 (3) (1999) 530–544.
- [18] T. H. Truong, F. Azadivar, Simulation optimization in manufacturing analysis: simulation based optimization for supply chain configuration design, in: *Proceedings of the 2003 Winter Simulation Conference*, 2003, pp. 1268–1275.

- [19] L. Shi, S. Ólafsson, Nested partitions method for stochastic optimization, *Methodology and Computing in Applied Probability* 2 (3) (2000) 271–291.
- [20] S. Ólafsson, Chapter 21 metaheuristics, in: S. G. Henderson, B. L. Nelson (Eds.), *Simulation*, Vol. 13 of *Handbooks in Operations Research and Management Science*, Elsevier, 2006, pp. 633–654.
- [21] M. Dorigo, C. Blum, Ant colony optimization theory: A survey, *Theoretical Computer Science* 344 (2) (2005) 243–278.
- [22] P. Larrañaga, J. A. Lozano, Estimation of distribution algorithms: A new tool for evolutionary computation, Vol. 2, Springer, 2002.
- [23] R. Y. Rubinstein, D. P. Kroese, The cross-entropy method: a unified approach to combinatorial optimization, Monte-Carlo simulation and machine learning, Springer, 2004.
- [24] J. Hu, M. C. Fu, S. I. Marcus, A model reference adaptive search method for stochastic global optimization, *Communications in Information & Systems* 8 (3) (2008) 245–276.
- [25] M. C. Fu, F. W. Glover, J. April, Simulation optimization: a review, new developments, and applications, in: *Proceedings of the 2005 Winter Simulation Conference*, 2005, pp. 83–95.
- [26] S. Andradóttir, Chapter 20 an overview of simulation optimization via random search, in: S. G. Henderson, B. L. Nelson (Eds.), *Simulation*, Vol. 13 of *Handbooks in Operations Research and Management Science*, Elsevier, 2006, pp. 617–631.
- [27] L. J. Hong, B. L. Nelson, J. Xu, Speeding up compass for high-dimensional discrete optimization via simulation, *Operations Research Letters* 38 (6) (2010) 550–555.
- [28] J. Xu, B. L. Nelson, L. J. Hong, An adaptive hyperbox algorithm for high-dimensional discrete optimization via simulation problems, *INFORMS Journal on Computing* 25 (1) (2013) 133–146.
- [29] R. Suri, Y. T. Leung, Single run optimization of a siman model for closed loop flexible assembly systems, in: *Proceedings of the 1987 Winter Simulation Conference*, 1987, pp. 738–748.
- [30] H. J. Kushner, G. Yin, *Stochastic approximation and recursive algorithms and applications*, Vol. 35, Springer, 2003.
- [31] G. Gurkan, A. Y. Ozge, T. M. Robinson, Sample-path optimization in simulation, in: *Proceedings of the 1994 Winter Simulation Conference*, 1994, pp. 247–254.
- [32] E. L. Plambeck, B.-R. Fu, S. M. Robinson, R. Suri, Sample-path optimization of convex stochastic performance functions, *Mathematical Programming* 75 (1996) 137–176.
- [33] A. Khuri, S. Mukhopadhyay, Response surface methodology, *Wiley Interdisciplinary Reviews: Computational Statistics* 2 (2) (2010) 128–149.
- [34] Y. Ho, L. Shi, L. Dai, W.-B. Gong, Optimizing discrete event dynamic systems via the gradient surface method, *Discrete Event Dynamic Systems* 2 (1992) 99–120.
- [35] A. J. Booker, J. E. Dennis, P. D. Frank, D. B. Serafini, V. Torczon, M. W. Trosset, A rigorous framework for optimization of expensive functions by surrogates, *Structural and Multidisciplinary Optimization* 17 (1999) 1–13.
- [36] V. Torczon, On the convergence of pattern search algorithms, *SIAM J. on Optimization* 7 (1) (1997) 1–25.
- [37] R. H. Wild, J. J. Pignatiello, Jr., Finding stable system designs: a reverse simulation technique, *Commun. ACM* 37 (10) (1994) 87–98.
- [38] Y. H. Lee, K. J. Park, Y. B. Kim, Single run optimization using the reverse-simulation method, in: *Proceedings of the 1997 Winter Simulation Conference*, 1997, pp. 187–193.
- [39] A. Gosavi, *Simulation-Based Optimization: Parametric Optimization Techniques and Reinforcement Learning*, Kluwer Academic Publishers, 2003.
- [40] W. B. Powell, What you should know about approximate dynamic programming, *Naval Research Logistics (NRL)* 56 (3) (2009) 239–249.
- [41] K. Healy, L. W. Schruben, Retrospective simulation response optimization, in: *Proceedings of the 1991 Winter Simulation Conference*, 1991, pp. 901–906.
- [42] R. Pasupathy, On choosing parameters in retrospective-approximation algorithms for stochastic root finding and simulation optimization, *Operations Research* 58 (4-Part-1) (2010) 889–901.
- [43] D. D. Briggs, A hybrid analytical/simulation modeling approach for planning and optimizing mass tactical airborne operations, Tech. rep., DTIC Document (1995).
- [44] S. J. Lim, S. J. Jeong, K. S. Kim, M. W. Park, A simulation approach for production-distribution planning with consideration given to replenishment policies, *The International Journal of Advanced Manufacturing Technology* 27 (2006) 593–603.
- [45] D. Subramanian, J. F. Pekny, G. V. Reklaitis, A simulation-optimization framework for addressing combinatorial and stochastic aspects of an r&d pipeline management problem, *Computers & Chemical Engineering* 24 (2) (2000) 1005–1011.
- [46] A. Kleywegt, A. Shapiro, T. Homem-De-Mello, The sample average approximation method for stochastic discrete optimization, *SIAM Journal on Optimization* 12 (2) (2002) 479–502.
- [47] T. Santos, S. Ahmed, M. Goetschalckx, A. Shapiro, A stochastic programming approach for supply chain network design under uncertainty, *European Journal of Operational Research* 167 (1) (2005) 96–115.
- [48] L. W. Schruben, Mathematical programming models of discrete event system dynamics, in: *Proceedings of the 32nd conference on Winter simulation*, 2000, pp. 381–385.
- [49] R. L. Nolan, M. G. Sovereign, A recursive optimization and simulation approach to analysis with an application to transportation systems, *Management Science* 18 (12) (1972) B676–B690.
- [50] J.-Y. Bang, Y.-D. Kim, Hierarchical production planning for semiconductor wafer fabrication based on linear programming and discrete-event simulation, *IEEE Transactions on Automation Science and Engineering* 7 (2) (2010) 326–336.
- [51] C. Almeder, M. Preusser, R. Hartl, Simulation and optimization of supply chains: alternative or complementary approaches?, *OR Spectrum* 31 (2009) 95–119.
- [52] D. Irdem, N. Kacar, R. Uzsoy, An experimental study of an iterative simulation-optimization algorithm for production planning, in: *Proceedings of the 2008 Winter Simulation Conference*, 2008, pp. 2176–2184.
- [53] Y. Acar, S. N. Kadipasaoglu, J. M. Day, Incorporating uncertainty in optimal decision making: Integrating mixed integer programming and simulation to solve combinatorial problems, *Computers & Industrial Engineering* 56 (1) (2009) 106–112.
- [54] J. Asmundsson, R. Rardin, R. Uzsoy, Tractable nonlinear production planning models for semiconductor wafer fabrication facilities, *Semi-*

- conductor Manufacturing, *IEEE Transactions on* 19 (1) (2006) 95–111.
- [55] J. Y. Jung, G. Blau, J. F. Pekny, G. V. Reklaitis, D. Eversdyk, A simulation based optimization approach to supply chain management under demand uncertainty, *Computers & Chemical Engineering* 28 (10) (2004) 2087–2106.
 - [56] W. K. V. Chan, L. Schruben, Optimization models of discrete-event system dynamics, *Operations Research* 56 (5) (2008) 1218–1237.
 - [57] A. Alfieri, A. Matta, Mathematical programming formulations for approximate simulation of multistage production systems, *European Journal of Operational Research* 219 (3) (2012) 773–783.
 - [58] A. Alfieri, A. Matta, Mathematical programming time-based decomposition algorithm for discrete event simulation, *European Journal of Operational Research* 231 (3) (2013) 557–566.
 - [59] J. April, F. Glover, J. P. Kelly, M. Laguna, Simulation-based optimization: practical introduction to simulation optimization, in: *Proceedings of the 2003 Winter Simulation Conference*, 2003, pp. 71–78.
 - [60] J. P. Kleijnen, W. van Beers, I. van Nieuwenhuysse, Constrained optimization in expensive simulation: Novel approach, *European Journal of Operational Research* 202 (1) (2010) 164–174.
 - [61] G. Gray, K. Fowler, J. Griffin, Hybrid optimization schemes for simulation-based problems, *Procedia Computer Science* 1 (1) (2010) 1349–1357.
 - [62] H. Pierrevale, J. L. Paris, From ‘simulation optimization’ to ‘simulation configuration’ of systems, *Simulation Modelling Practice and Theory* 11 (1) (2003) 5–19.
 - [63] F. Azadivar, Simulation optimization methodologies, in: *Proceedings of the 31st conference on Winter simulation: Simulation—a bridge to the future—Volume 1*, 1999, pp. 93–100.
 - [64] A. Ruszczyński, A. Shapiro, Stochastic programming models, in: A. Ruszczyński, A. Shapiro (Eds.), *Stochastic Programming*, Vol. 10 of *Handbooks in Operations Research and Management Science*, Elsevier, 2003, pp. 1–64.
 - [65] L. W. Schruben, *Common Random Numbers*, John Wiley & Sons, Inc., 2010.
 - [66] J. M. Hammersley, K. W. Morton, A new monte carlo technique: antithetic variates, *Mathematical Proceedings of the Cambridge Philosophical Society* 52 (03) (1956) 449–475.
 - [67] P. W. Glynn, R. Szechtman, Some new perspectives on the method of control variates, in: K. Fang, F. Hickernell, H. Niederreiter (Eds.), *Monte Carlo and Quasi-Monte Carlo Methods 2000*, Springer-Verlag, 2000, p. 27–49.
 - [68] P. W. Glynn, D. L. Iglehart, Importance sampling for stochastic simulations, *Management Science* 35 (11) (1989) 1367–1392.
 - [69] M. D. McKay, R. J. Beckman, W. J. Conover, A comparison of three methods for selecting values of input variables in the analysis of output from a computer code, *Technometrics* 42 (1) (2000) 55–61.
 - [70] S. Andradóttir, A. A. Prudius, Balanced explorative and exploitative search with estimation for simulation optimization, *INFORMS J. on Computing* 21 (2) (2009) 193–208.
 - [71] J. Xu, B. L. Nelson, J. Hong, Industrial strength compass: A comprehensive algorithm and software for optimization via simulation, *ACM Transactions on Modeling and Computer Simulation (TOMACS)* 20 (1) (2010) 1–29.
 - [72] S. Ólafsson, Two-stage nested partitions method for stochastic optimization, *Methodology And Computing In Applied Probability* 6 (1) (2004) 5–27.
 - [73] C. Almeder, R. F. Hartl, A metaheuristic optimization approach for a real-world stochastic flexible flow shop problem with limited buffer, *International Journal of Production Economics* 145 (1) (2013) 88–95.