Journal of Simulation (2014) 1–9 1747-7778/14

www.palgrave-journals.com/jos/

SimILS: a simulation-based extension of the iterated local search metaheuristic for stochastic combinatorial optimization

Alex Grasas[1]*, Angel A Juan[2] and Helena R Lourenço[3]

[1]EADA Business School, Barcelona, Spain; [2]IN3—Open University of Catalonia, Barcelona, Spain; and [3]Universitat Pompeu Fabra, Barcelona, Spain

Iterated Local Search (ILS) is one of the most popular single-solution-based metaheuristics. ILS is recognized by many authors as a relatively simple yet efficient framework able to deal with complex combinatorial optimization problems (COPs). ILS-based algorithms have been successfully applied to provide near-optimal solutions to different COPs in logistics, transportation, production, etc. However, ILS is designed to solve COPs under deterministic scenarios. In some real-life applications where uncertainty is present, the deterministic assumption makes the model less accurate since it does not reflect the real stochastic nature of the system. This paper presents the SimILS framework that extends ILS by integrating simulation to be able to cope with Stochastic COPs in a natural way. The paper also describes several tested applications that illustrate the main concepts behind SimILS and give rise to a new brand of ILS-based algorithms.

Journal of Simulation advance online publication, 3 October 2014; doi:10.1057/jos.2014.25

Keywords: iterated local search; simulation; stochastic combinatorial optimization; simheuristics

1. Introduction

This paper proposes and discusses the SimILS framework, a simulation-based extension of the well-known Iterated Local Search (ILS) metaheuristic. The SimILS framework integrates simulation methods inside the classical ILS framework in order to naturally deal with random components in the mathematical model of the combinatorial optimization problem. As described in Lourenço *et al* (2003), ILS is a conceptually simple yet powerful metaheuristic that has proven to be very efficient in solving complex Combinatorial Optimization Problems (COPs)-—a COP is a problem in which the best solution needs to be obtained from a finite or countably infinite set of objects (integers, permutations, graphs, etc). The underlying idea behind ILS is to narrow the search for candidate local optimal solutions returned by some embedded algorithm, typically a local search heuristic. Burke *et al* (2010) show that ILS obtains the best average performance among a set of selected metaheuristic approaches in three classical COPs: bin packing, permutation flow shop, and personnel scheduling. The authors also emphasize two main factors for its success: (i) an excellent balance between exploration and exploitation by 'systematically combining a perturbation followed by local search', and (ii) the reduced number of parameters required.

Numerous ILS applications in a myriad of contexts have been studied in the literature. Lourenço *et al* (2010) review some of them up to 2010. In fact, the use of ILS has been steadily increasing in the last years as shown in Figure 1. More recent implementations include Vehicle Routing Problems (VRP) (Penna *et al*, 2011; Nguyen *et al*, 2012; Vansteenwegen and Mateo, 2014), scheduling problems (Dong *et al*, 2013; Subramanian *et al*, 2014; Juan *et al*, 2014c), or travelling salesman problems (Delévacq *et al*, 2012; Subramanian and Battarra, 2012), just to name a few.

Despite its popularity, and as it happens with most other metaheuristics, ILS assumes that all input data is deterministic, there are no probabilistic constraints to meet, and there is no other source of randomness in the system being analysed. In most real-life problems, however, uncertainty is present; therefore, the deterministic assumption only allows simplifying the mathematical model to become tractable. Of course, this turns into a less accurate model that does not reflect the stochastic nature of the real-life system. In order to overcome this divergence, this paper conceptualizes the SimILS framework, an extension of the ILS metaheuristic that integrates simulation into its architecture (Figure 2). The result is an easy-to-implement simheuristic able to deal with stochastic COPs in a natural way. In stochastic COPs, some problem information may be unknown due to uncertainty, and in some cases, a decision must be made before knowing the realization of random variables (Kall and Wallace, 1994). Obviously, the integration between simulation and the ILS architecture must be done carefully in order to avoid incurring in prohibitive computational times due to the simulation component.

The remaining of this paper is structured as follows: Section 2 provides an overview of common stochastic COPs and different approaches used to address them; Section 3 presents the SimILS

**Correspondence: A Grasas, Department of Marketing, Operations and Supply, EADA Business School, Aragó 204, Barcelona 08011, Spain.*
E-mail: agrasas@eada.edu

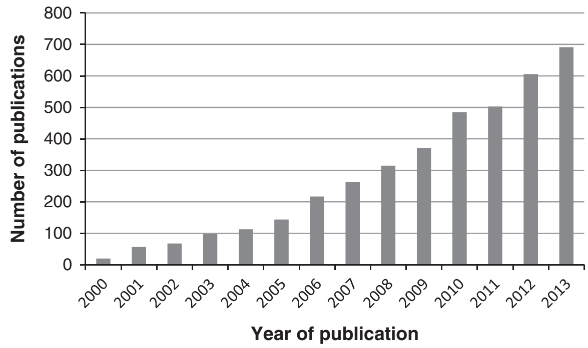

Figure 1 Google scholar's number of publications for 'iterated local search'.

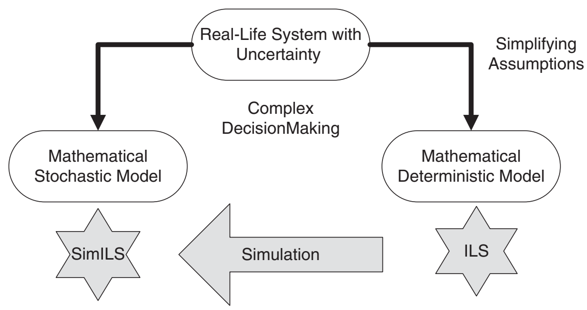

Figure 2 SimILS extends ILS for solving stochastic models.

framework from an algorithmic standpoint; Section 4 illustrates specific SimILS implementations already used to solve stochastic COPs in different fields; and, finally, Section 5 summarizes the main contributions of this article.

2. Solving stochastic combinatorial optimization problems

This section reviews some of the most popular stochastic COPs analysed in the scientific literature and then provides an overview of the different approaches that have been used to cope with them. The following stochastic COPs have received special attention during the last decades:

- *Stochastic VRP* (Gendreau *et al*, 1996b): The most common variant is the VRP with Stochastic Demands (Bertsimas, 1992), where customers' demands are unknown *a priori*, which might cause the planned routes to become infeasible during actual delivery (eg, due to vehicle capacity constraint). An extension of this problem also considers visiting customers only with a given probability (Gendreau *et al*, 1995).
- *Probabilistic Travelling Salesman Problem* (Jaillet, 1985): Customers need to be visited with a prescribed probability. Consequently, an *aprioristic* tour must be planned before knowing which customers will be visited.
- *Inventory Routing Problems with Stochastic Demands* (Federgruen and Zipkin, 1984): In this extension of the VRP, the delivery of products must be planned jointly with the inventory management at each retail centre, which is subjected to the effect of final users' random demands.
- *Stochastic Arc Routing Problem* (Fleury *et al*, 2002; Fleury *et al*, 2005): At least one of the parameters or structural variables is random.
- *Stochastic Scheduling Problems* (Rothkopf, 1966): Job processing times are random variables with known probability distributions. Some popular cases include the Permutation Flow Shop Problem with Stochastic Times (Banerjee, 1965; Makino, 1965) and the Stochastic Job Shop Scheduling Problem (Kise *et al*, 1982).
- *Probabilistic Set-Covering Problem* (Beraldi and Ruszczyński, 2002): The right-hand-side constraints constitute a random binary vector and the covering constraint has to be satisfied with a given probability.
- *Stochastic Time-cost Trade-off Problem* (Gutjahr *et al*, 2000): There is an uncertain cost associated with the reduction of the random duration of activities.
- *Stochastic Knapsack Problem* (Ross and Tsang, 1989): Objects arrive randomly to the knapsack. Other stochastic variants include random item sizes (Kleinberg *et al*, 2000) or random rewards (Steinberg and Parks, 1979).

2.1. Solving stochastic COPs through exact methods

Exact methods can not only provide optimal solutions to small- and medium-size problems, but they can also be used as building blocks for other approximation methods when facing large-size problems. Gendreau *et al* (1995) and Christiansen and Lysgaard (2007) use, respectively, an integer L-shaped method and a branch-and-price algorithm for the VRP with Stochastic Demands. Laporte *et al* (1994) solve the Probabilistic Travelling Salesman Problem with a branch-and-cut algorithm. Federgruen and Zipkin (1984) adapt generalized Bender's decomposition for the Inventory Routing Problem with Stochastic Demands. Christiansen *et al* (2009) formulate the Arc Routing Problem with Stochastic Demands as a Set Partitioning Problem, and develop a branch-and-price algorithm for solving it. Pinedo (1984) surveys optimal policies in Stochastic Open Shop, Flow Shop and Job Shop Scheduling Problems. Beraldi and Ruszczyński (2002) propose a branch-and-bound method for the Probabilistic Set-Covering Problem based on partial and complete enumeration. Despite the theoretical interest of the described exact approaches, and due to the complexity of the stochastic COPs, most of their practical applications are restricted to small-size instances and to a reduced set of probability distributions—for example, Normal or Exponential distributions—to model the randomness in the system. Therefore, they are not suitable for solving large-size instances or instances in which system randomness is modelled with general distributions, as it frequently occurs in real life.

2.2. Solving stochastic COPs through metaheuristics

As the complexity of the problem grows, approximate methods such as heuristics and metaheuristics are more appropriate than exact methods to deal with stochastic COPs, especially when 'high-quality' solutions are needed in reasonable computing times. Different types of metaheuristics have been used to solve a wide range of stochastic COPs. The following examples illustrate the use of different metaheuristics for solving stochastic VRPs: Ant Colony Optimization (Dorigo, 1992; Bianchi *et al*, 2006; Zhang, 2007); Evolutionary Computation and Genetic Algorithms (Holland, 1973; Bianchi *et al*, 2006; Tan *et al*, 2007; Ismail, 2008; Shanmugam *et al*, 2011); Iterated Local Search (Lourenço *et al*, 2003; Bianchi *et al*, 2006); Large Neighbourhood Search (Shaw, 1998; Lei *et al*, 2011); Particle Swarm Optimization (Kennedy and Eberhart, 1995; Shanmugam *et al*, 2011; Moghaddam *et al*, 2012; Marinakis *et al*, 2013); Scatter Search (Glover, 1977; Zhang *et al*, 2012); Simulated Annealing (Kirkpatrick *et al*, 1983; Teodorovic and Pavkovic, 1992; Bianchi *et al*, 2006); and Tabu Search (Glover, 1986; Gendreau *et al*, 1996a, b; Bianchi *et al*, 2006; Ismail, 2008; Shen *et al*, 2009). The reader is referred to Bianchi *et al* (2009) for a thorough survey of metaheuristics applied to other stochastic COPs.

2.3. Solving stochastic COPs through simheuristics

In the last decades, there has been a growing interest for combining simulation and optimization techniques. This allows tackling stochastic optimization problems using the virtues of both realms. Fu *et al* (2000) categorize the four major approaches to simulation optimization, namely: (i) gradient-based and random search algorithms; (ii) evolutionary algorithms and metaheuristics; (iii) mathematical programming-based approaches; and (iv) statistical search techniques. Glover *et al* (1996), after a review of the classical simulation-optimization approach, highlight the emerging role of the integration of simulation and metaheuristics, that is, simheuristics. They unveil how a combined Scatter Search and Tabu Search metaheuristic can effectively guide a series of simulations to uncover near-optimal solutions to stochastic COPs with large solution spaces. In their words: 'The importance of integrating the complementary realms of optimization [metaheuristics] and simulation assures that future advances will have a high impact on real world applications'. It is worth mentioning that this integration is the most widely used approach in commercial simulation-optimization software (Fu *et al*, 2000; April *et al*, 2003), and is becoming more popular among the scientific community (Figure 3).

Several authors have combined simulation and metaheuristics to solve different stochastic COPs. In the area of distribution, Gutjahr (2004) compares the performance of an Ant Colony Optimization stochastic algorithm to that of a stochastic

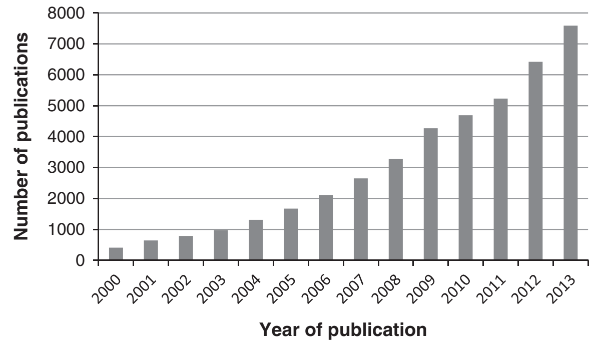

Figure 3 Google scholar's number of publications for 'simulation' and 'metaheuristics'.

Simulated Annealing algorithm for the Probabilistic Travelling Salesman Problem. Subramaniam and Gosavi (2007) employ a combination of simultaneous perturbation and Simulated Annealing to the inventory allocation problem in a distribution network. Chiang *et al* (2009) use Tabu Search to an integrated newspaper production and distribution supply chain management problem. Tripathi *et al* (2009) develop an improved Ant Colony Optimization approach for the VRP with Stochastic Demands. González *et al* (2012) apply a multi-start search procedure that uses Monte Carlo simulation inside a biased-randomized version of a classical heuristic for solving the Stochastic Arc Routing Problem.

In production environments, Dengiz and Alabas (2000) use a Tabu Search algorithm with a simulation model of a Just-In-Time system to find the optimal number of kanbans. Altiparmak *et al* (2002) develop an artificial neural network together with a Simulated Annealing algorithm to optimize the buffer size in an asynchronous assembly system. Arreola-Risa *et al* (2011) design a heuristic based on simulation and regression analysis for stochastic production-inventory systems. Laroque *et al* (2012) present a combination of Particle Swap Optimization and Genetic Algorithms procedures within a material flow simulation problem. For scheduling problems, Legato *et al* (2010) investigate a Simulated Annealing procedure to the quay crane scheduling problem with stochastic discharge/loading processes. Baker and Altheimer (2012) tackle the Permutation Flow-Shop with Stochastic Times through the use of hybrid algorithms that combine simulation with the CDS and NEH heuristics.

In other contexts, Konak and Kulturel-Konak (2005) implement a simulation-optimization method using Tabu Search for the Stochastic Knapsack Problem, and Balasubramanian *et al* (2007) propose a multi-period Genetic Algorithm approach for the assignment problem of panel design in primary care.

3. The SimILS framework

ILS extends a problem-specific local search method by introducing a perturbation at each new local optimal solution before restarting the search for a new local optimal solution. ILS is based on four procedures: (i) Generation of an Initial Solution, (ii) Local

Search, (iii) Perturbation, and (iv) Acceptance Criterion (Algorithm 1).

Algorithm 1 Iterated Local Search framework

Procedure *Iterated Local Search*
  $s_0 = GenerateInitialSolution$
  $s^* = LocalSearch\ (s_0)$
  Repeat
    $s' = Perturbation\ (s^*, history)$
    $s'^* = LocalSearch\ (s')$
    $s^* = AcceptanceCriterion\ (s^*, s'^*, \text{history})$
  Until termination condition met
End

ILS aims at avoiding the disadvantages of random restarts by exploring the region of feasible solutions using a walk that steps from one local optimal solution s^* to a 'nearby' one. Given the current solution s^*, a change or perturbation is first applied leading to an intermediate feasible solution, s'. Then, a Local Search is applied to s' to obtain a new local optimal solution, s'^*. If s'^* passes an acceptance test, it becomes the new current solution; otherwise, one returns to the previous one, s^*. Given the structure in Algorithm 1, a basic version of ILS can be easily developed for most COPs: (i) it is possible to start with a random solution or with one obtained by a simple constructive heuristic; (ii) for most problems, a local search algorithm is readily available; (iii) for the perturbation, a random move in a neighbourhood of higher order than the one used by the local search algorithm can be surprisingly effective; and (iv) a reasonable first guess for the acceptance criterion is to force the cost to decrease, corresponding to a first-improvement descent. Basic ILS implementations of this type usually lead to much better performance than random restart approaches and, in many cases, even better than sophisticated metaheuristics with a large number of parameters to configure. The developer can then run this basic ILS to build her intuition and try to improve the overall algorithm performance by improving each of the four modules and tuning their interaction. Two important aspects to consider are the perturbation phase and the interaction among the different elements of ILS. On the one hand, the perturbation applied to the current local solution should permit to escape from this local optimum. On the other, the interaction among the elements should lead to a large search on the space of feasible solutions. Thus, for example, the perturbation should not be easily undone by the local search or the acceptance criterion should avoid early convergence to an initial local optimal solution.

A well-designed ILS algorithm has all the important attributes of a metaheuristic according to the desirable set of properties described in Cordeau *et al* (2002): accuracy, speed, simplicity, and flexibility. ILS requires few parameters, and they can be easily adapted to different variants or extensions of a given COP. Given its attributes, we consider that ILS is an excellent candidate, as a metaheuristic framework, to be combined with simulation in order to solve stochastic COPs following a natural and easy-to-implement approach.

The general SimILS framework, described in Algorithm 2, integrates simulation at some specific steps, resulting in a simulation-optimization procedure capable of dealing with stochastic COPs. In particular, simulations are inserted after applying the Local Search to evaluate the current local optimal solution, s^* or s'^*. These simulations take this solution and a parameter indicating whether the simulation should be run for a long or a short time, and then obtain the corresponding simulated objective function, $sf(\cdot)$, along with other relevant statistics that can be used later to improve the search (eg, satisfaction degree of the probability constraints). A simulation component is also inserted at the end of the ILS process.

Notice that, whenever possible, the simulation component inside the main loop should be applied just over a selected and reduced subset of newly generated solutions. Otherwise, the execution of simulation runs for every newly generated solution will consume most of the algorithm's computational time, thus avoiding the ILS framework to converge to pseudo-optimal solutions in reasonable computing times. One way to do this is by performing short-run simulations only when a newly generated solution (almost) improves the deterministic value of the base solution, that is, whenever a newly generated solution is a 'promising' solution. Also, since the accuracy of a simulation depends on the number of runs executed, it might be worthy to keep in memory a selected list of top solutions obtained during the ILS process. If cost differences among solutions are small, a long-run simulation after the main ILS process is justified on these solutions to increase the accuracy of the results and to select the best-found solution for the stochastic COP being analysed.

Algorithm 2 General SimILS framework extending the original ILS framework

Procedure *SimILS*
  $s_0 = GenerateInitialSolution$
  $s^* = LocalSearch(s_0)$
  $(s^*, sf(s^*), statistics) = Simulation(s^*, long)$
  Repeat
    $s' = Perturbation(s^*, \text{history})$
    $s'^* = LocalSearch(s')$
    $(s'^*, sf(s'^*), statistics) = Simulation(s'^*, \text{short})$
    $s^* = AcceptanceCriterion(s^*, s'^*, \text{history})$
  Until termination condition met
  $(s^*, sf(s^*), statistics) = Simulation(s^*, \text{long})$
Return s^*, $sf(s^*)$
End

In practice, the role of simulation is mainly oriented either to: (i) estimate the expected cost value of a newly generated solution—whenever the objective function contains some stochastic components—, or (ii) check that a newly generated solution satisfies some probabilistic constraints. Therefore, two variants of the general SimILS framework presented in Algorithm 2 are proposed for COPs with a stochastic objective function (Algorithm 3), and for COPs with stochastic constraints (Algorithm 4). Examples of the first case include VRPs with stochastic travelling times, Scheduling problems with stochastic processing times, and Location problems with stochastic transportation and inventory costs. Examples of the second case include VRPs or Location problems with probabilistic constraints regarding the total demand to be served, and Scheduling problems with stochastic machine availability. Some specific implementations of the first case can be found in Juan *et al* (2011, 2013, 2014a, b), while an implementation of the second case is studied in Cabrera *et al* (2014). These implementations are discussed in more detail in the next section.

Algorithm 3 SimILS framework for COPs with stochastic objective function

Procedure *SimILS*
 $s_0 = GenerateInitialSolution$
 $s^* = LocalSearch\ (s_0)$
 $ss^* = s^*$
 $(ss^*, sf(ss^*), statistics) = Simulation\ (ss^*, \text{long})$
 $bsf^* = sf(ss^*)$
 Repeat
  $s' = Perturbation\ (s^*, \text{history})$
  $s'^* = LocalSearch\ (s')$
  $s^* = AcceptanceCriterion\ (s^*, s'^*, \text{history})$
  $(s^*, sf(s^*), statistics) = Simulation\ (s^*, \text{short})$
  If $sf(s^*) < bsf^*$
   $bsf^* = sf(s^*)$;
   $ss^* = s^*$
 Until termination condition met
 $(ss^*, sf(ss^*), statistics) = Simulation\ (ss^*, \text{long})$
 $(s^*, sf(s^*), statistics) = Simulation\ (s^*, \text{long})$
 Return $(ss^*; sf(ss^*))$ and $(s^*; sf(s^*))$
End

In the case of a COP with a stochastic objective function (Algorithm 3), the SimILS maintains two best solutions: the best local optimal solution, s^*, for the deterministic objective function (ie, with average data), and the best local optimal solution, ss^*, for the stochastic objective function obtained after the simulation. The algorithm also keeps track of the corresponding best objective function values, $f(s^*)$ and $bsf^* = sf(ss^*)$, respectively. At each iteration, the best solutions along with the corresponding values are updated, and they are reported at the end of the SimILS algorithm. Figure 4 illustrates the steps carried out by the SimILS in the presence of a stochastic objective function for a minimization problem. Starting with a local minimum, s^*, the Perturbation phase is applied to obtain a solution s'. A Local Search then finds a new local minimum s'^*. Both s^* and ss^* are updated (blue and red lines, respectively). The process is repeated to obtain a new s' and, afterwards, a new s'^*. This time, only s^* is updated since $sf(s^*) > bsf^*$.

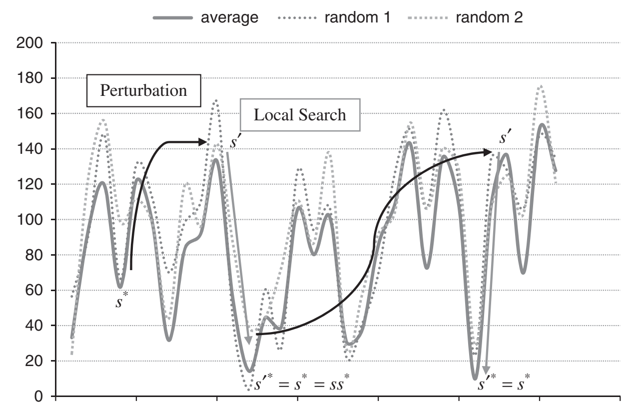

Figure 4 Exemplification of the SimILS for a COP with stochastic objective function.

In the case of a COP with stochastic constraints (Algorithm 4), the purpose of the simulation is to verify whether the newly generated solution satisfies these constraints with a certain probability. One can define this probability as a service level. The user needs to define a service level threshold, that is, a probability of satisfying a given constraint. For example, in COPs with stochastic demands, the user could require a solution to meet demand with 90% probability. In the initial phase of the SimILS, the algorithm looks for an initial solution that satisfies this service level threshold, and, in the second phase, it optimizes the objective function.

Algorithm 4 SimILS framework for COPs with stochastic constraints

Procedure *SimILS*
 $s_0 = GenerateInitialSolution$ (*input data: average values*)
 $s^* = LocalSearch(s_0)$
 $(s^*, sf(s^*), \text{service level}) = Simulation\ (s^*, \text{long})$
 Repeat
  $s' = Perturbation\ (s^*, \text{history})$
  $s^* = LocalSearch\ (s')$
  $(s^*, sf(s^*), \text{service level}) = Simulation\ (s^*, \text{short})$
 Until verifying service level threshold
 Repeat
  $s' = Perturbation\ (s^*, \text{history})$
  $s'^* = LocalSearch\ (s')$
  $(s'^*, sf(s'^*), \text{service level}) = Simulation\ (s'^*, \text{short})$
  $s^* = AcceptanceCriterion\ (s^*, s'^*, \text{service level, history})$
 Until termination condition met
 $(s^*, sf(s^*), \text{service level}) = Simulation\ (s^*, \text{long})$
Return $(s^*; sf(s^*))$
End

Finally, when a COP has both stochastic objective function and stochastic constraints, Algorithms 3 and 4 can be combined into Algorithm 5 to tackle the problem. There exist many problems of this type in real life (eg, a VRP with both stochastic demands and travelling times).

Algorithm 5 SimILS framework for COPs with stochastic objective function and constraints

Procedure *SimILS*
  $s_0 = GenerateInitialSolution$
  $s^* = LocalSearch\ (s_0)$
  Repeat
    $s' = Perturbation\ (s^*$, history)
    $s^* = LocalSearch\ (s')$
    (s^*, $sf(s^*)$, service level) = *Simulation* (s^*, short)
  Until verifying service level threshold
  $bdf^* = f(s^*)$;
  $ss^* = s^*$
  (ss^*, $sf(ss^*)$, *statistics*) = *Simulation* (ss^*, long)
  $bsf^* = sf(ss^*)$
  Repeat
    $s' = Perturbation\ (s^*$, history)
    $s'^* = LocalSearch\ (s')$
    (s'^*, $sf(s'^*)$, service level) = *Simulation* (s'^*, short)
    $s^* = AcceptanceCriterion\ (s^*, s'^*$, service level, history)
    If $sf(s^*) < bsf^*$
      $bsf^* = sf(s^*)$;
      $ss^* = s^*$
  Until termination condition met
  (ss^*, $sf(ss^*)$, *statistics*) = *Simulation* (ss^*, long)
  (s^*, $sf(s^*)$, *statistics*) = *Simulation* (s^*, long)
  Return (ss^*; $sf(ss^*)$) and (s^*;$sf(s^*)$)
End

4. Some application examples

In this section we describe several examples in which multi-start or ILS-like algorithms have been successfully extended to solve stochastic COPs in different fields. Notice that multi-start approaches can be seen as an extreme case of ILS-based approaches in which the perturbation process resets the base solution to the initial one. Following this logic, any multi-start process can be easily evolved to an ILS-based approach by using less aggressive perturbation processes—which, as discussed before, tend to accelerate the algorithm's convergence by keeping part of the base solution instead of restarting it to the initial solution.

The first applications correspond to COPs with a stochastic objective function (Algorithm 3). In Juan *et al* (2011, 2013), the authors combine simulation with a multi-start framework in order to solve the VRP with Stochastic Demands. In the former, simulation is used just after the multi-start local search has finished, whereas in the latter simulation is integrated inside the multi-start procedure as described in the previous section. The authors show that, apart from estimating the total costs associated with a given routing plan, simulation also allows the decision maker to obtain an estimate of the reliability level of each routing plan. In other words, by using simulation more insight into the properties of the provided solutions is obtained.

In Juan *et al* (2014a) the authors integrate simulation into an ILS framework to solve the Permutation Flow-Shop Problem with Stochastic Processing Times. First, they 'transform' the original stochastic problem into a deterministic problem using the expected processing times. Then, the realization of stochastic processing times converts the objective function into stochastic, as it has been generally described in Algorithm 3. To obtain an initial solution for the deterministic version, the NEH heuristic by Nawaz *et al* (1983) is applied. Next, to evaluate the performance of this initial solution when stochastic processing times are considered, a short Monte Carlo simulation is run and basic statistics on the stochastic makespan are gathered. The next steps are the ILS procedures: perturbation and local search. These two steps are applied as in the deterministic case. However, after obtaining a new base solution, a short simulation is run to generate its associated value for the stochastic problem. During the iteration of these steps, two solutions are always maintained: the best deterministic solution (ie, using expected processing times); and the best stochastic solution (ie, that with the best observed stochastic makespan in the simulation). Of course, the best solution for the deterministic case may not necessarily be the best solution for the stochastic version. The last step consists in performing a longer simulation with these two solutions in order to obtain more accurate statistics and makespan values. Worth mentioning is that the simulation phase is quite simple to implement, and it does not significantly increase the running time of the SimILS. This simulation phase consists in the following: given a solution (that is, a permutation of the jobs) and a probability function associated to the processing times, a random variable is generated for each job-machine processing time, and then an observed makespan is calculated (the critical path length of the associated disjunctive graph). The process is then repeated according to the number of iterations, using long or short simulations as desired. Also, it is important to say that the simulation component not only is used to obtain accurate estimates of the expected makespan associated with each proposed solution (permutation of jobs), but it is also employed to generate insight on the probabilistic characteristics of each proposed solution. In particular, without increasing the computational effort, simulation allows to estimate the probabilities of each solution being completed before a specific deadline.

In Juan *et al* (2014b), the authors also use a SimILS approach for solving the Inventory Routing Problem with Stochastic Demands and Stock-outs. The problem consists in defining a routing distribution plan that includes the product quantities to deliver to a set of retailers. The decision must consider both the stochastic retailers' demand and the available stock at each retailer. Solutions to this problem account not only for routing costs, but also for inventory-holding and inventory-stockout costs due to demand stochasticity. This kind of integrated inventory routing problems, in which uncertainty is also present, are quite common in supply chain management, and the combination of simulation with a multi-start or ILS algorithm turns to be an efficient way to cope with them despite their intrinsic difficulty. The presence of stochastic demands implies a stochastic objective function (ie, with inventory holding or stockout costs) plus stochastic constraints. Now the vehicle capacity constraints are affected by the quantity delivered, which depends on the stochastic demands. The authors use a SimILS framework with stochastic objective function and constraints (Algorithm 5). The problem solution (ie, routing and quantities delivered) is calculated using a particular service level for each retailer based on an estimated demand. The service level is given by a refill policy that goes from no refill to 100% refill of the estimated demand. Then, the simulation step calculates the solution's total costs (routing plus stocks) while identifying the service level, that is, the maximum possible quantity delivered without violating vehicle capacity constraints. As in the previous work, the algorithm uses short and long Monte Carlo simulations at different stages, leading to short running times. At the end, the algorithm yields two solutions, the best deterministic and stochastic solutions, and runs a long simulation to evaluate the service level and other relevant statistics. The authors also perform an extensive computational experiment to conclude that total costs and refill policies of their algorithm outperform other proposed approaches in the literature. In these previous approaches, a general refill policy is usually applied to all retailers, whereas in the proposed SimILS personalized policies for each retailer can be obtained. A final advantage of their SimILS is that any probability function can be used since the Monte Carlo simulation phase does not require any special assumption on the demand's stochastic behaviour. The method can be easily generalized to different demand probability functions and applied to different industries and business environments.

One last example of application can be found in Cabrera *et al* (2014), which is a COP with stochastic constraints (Algorithm 4). Here, the authors face the problem of guaranteeing, at the minimum possible cost, the availability of Internet services deployed over a large-scale set of distributed and non-dedicated resources. For that, they propose a local search algorithm that integrates discrete-event simulation to check the feasibility of each newly generated solution. In other words, they use simulation to estimate the availability level of a proposed 'promising' solution and make sure this level is over a user-defined threshold before accepting it as a feasible one.

5. Conclusions

ILS represents one of the most efficient yet easy-to-implement frameworks for solving medium-size and large-size instances of COPs in a wide variety of fields. However, as it happens with other metaheuristics, it was not designed to include stochastic elements in the objective function or in the constraints of the mathematical model that represents the real system. Since most real-life problems are filled with uncertainty, ILS needs to be extended to cope with this intrinsic stochasticity. By integrating simulation inside the local search process, the resulting SimILS framework extends the virtues of ILS to stochastic COPs as well. When conveniently implemented, the SimILS approach represents a natural, low parametrized, and efficient way to deal with complex decision-making problems under uncertain scenarios. Algorithms based on the SimILS architecture can also be useful when providing additional insight into the probabilistic characteristics of the generated solutions, something which cannot be easily obtained by using components different from simulation. Moreover, SimILS approaches do not need to assume any particular behaviour regarding the probability distributions that model the system's randomness. Several applications to vehicle routing, scheduling, inventory routing, and Internet computing contribute to illustrate the potential of the proposed framework.

Acknowledgements—This work has been partially supported by the Spanish Ministry of Economy and Competitiveness (TRA2013-48180-C3-P) and by the Ibero-American Programme for Science, Technology and Development (CYTED2010-511RT0419).

References

Altiparmak F, Dengiz B and Bulgak AA (2002). Optimization of buffer sizes in assembly systems using intelligent techniques. In: Yucesan E, Chen C-H, Snowdon JL and Chames JM (eds). *Proceedings of the 2002 Winter Simulation Conference*. IEEE Press: San Diego, CA, 8–11 December, pp 1157–1162.

April J, Glover F, Kelly JP and Laguna M (2003). Simulation-based optimization: Practical introduction to simulation optimization. In: Chick S, Sánchez PJ, Ferrin D and Morrice DJ (eds). *Proceedings of the 2003 Winter Simulation Conference*. IEEE Press: New Orleans, LA, 7–10 December, pp 71–78.

Arreola-Risa A, Giménez-García VM and Martínez-Parra JL (2011). Optimizing stochastic production-inventory systems: A heuristic based on simulation and regression analysis. *European Journal of Operational Research* **213**(1): 107–118.

Baker KR and Altheimer D (2012). Heuristic solution methods for the stochastic flow shop problem. *European Journal of Operational Research* **216**(1): 172–177.

Balasubramanian H, Banerjee R, Gregg M and Denton BT (2007). Improving primary care access using simulation optimization. In: Henderson SG, Biller B, Hsieh M-H, Shortle J, Tew JD and Barton RR (eds). *Proceedings of the 2007 Winter Simulation Conference*. IEEE Press: Washington DC, 9–12 December, pp 1494–1500.

Banerjee BP (1965). Single facility sequencing with random execution times. *Operations Research* **13**(3): 358–364.

Beraldi P and Ruszczyński A (2002). The probabilistic set-covering problem. *Operations Research* **50**(6): 956–967.

Bertsimas DJ (1992). A vehicle routing problem with stochastic demand. *Operations Research* **40**(3): 574–585.

Bianchi L *et al* (2006). Hybrid metaheuristics for the vehicle routing problem with stochastic demands. *Journal of Mathematical Modelling and Algorithms* **5**(1): 91–110.

Bianchi L, Dorigo M, Gambardella LM and Gutjahr WJ (2009). A survey on metaheuristics for stochastic combinatorial optimization. *Natural Computing* **8**(2): 239–287.

Burke E *et al* (2010). Iterated local search vs. hyper-heuristics: Towards general-purpose search algorithms. In: *IEEE Congress on Evolutionary Computation*. IEEE Press: Barcelona, Spain, pp 3073–3080.

Cabrera G, Juan AA, Lázaro D, Marquès JM and Proskurnia I (2014). A simulation-optimization approach to deploy internet services in large-scale systems with user-provided resources. *Simulation: Transactions of the Society for Modeling and Simulation International* **90**(6): 644–659.

Chiang W-C, Russell R, Xu X and Zepeda D (2009). A simulation/metaheuristic approach to newspaper production and distribution supply chain problems. *International Journal of Production Economics* **121**(2): 752–767.

Christiansen CH and Lysgaard J (2007). A branch-and-price algorithm for the capacitated vehicle routing problem with stochastic demands. *Operations Research Letters* **35**(6): 773–781.

Christiansen CH, Lysgaard J and Wøhlk S (2009). A branch-and-price algorithm for the capacitated arc routing problem with stochastic demands. *Operations Research Letters* **37**(6): 392–398.

Cordeau JF, Gendreau M, Laporte G, Potvin J-Y and Semet F (2002). A guide to vehicle routing heuristics. *Journal of the Operational Research Society* **53**(5): 512–522.

Delévacq A, Delisle P and Krajecki M (2012). Parallel GPU implementation of iterated local search for the travelling salesman problem. In: Hamadi Y and Schoenauer M (eds). *Learning and Intelligent Optimization*. Lecture Notes in Computer Science. Springer-Verlag: Berlin/Heidelberg, pp 372–377.

Dengiz B and Alabas C (2000). Simulation optimization using tabu search. In: Joines JA, Barton RR, Kang K and Fishwick PA (eds). *Proceedings of the 2000 Winter Simulation Conference*. IEEE Press: Orlando, FL, 10–13 December, pp 805–810.

Dong X, Chen P, Huang H and Nowak M (2013). A multi-restart iterated local search algorithm for the permutation flow shop problem minimizing total flow time. *Computers and Operations Research* **40**(2): 627–632.

Dorigo M (1992). *Optimization, learning and natural algorithms (in Italian)*. PhD Thesis, Politecnico di Milano, Italy.

Federgruen A and Zipkin P (1984). A combined vehicle routing and inventory allocation problem. *Operations Research* **32**(5): 1019–1037.

Fleury G, Lacomme P, Prins C and Ramdane-Chérif W (2002). Robustness evaluation of solutions for the capacitated arc routing problem. In: *Conference, AI Simulation and Planning in High Autonomy Systems*, Lisboa, Portugal, pp 290–295.

Fleury G, Lacomme P, Prins C and Ramdane-Chérif W (2005). Improving robustness of solutions to arc routing problems. *Journal of the Operational Research Society* **56**(5): 526–538.

Fu MC *et al* (2000). Integrating optimization and simulation: Research and practice. In: Joines JA, Barton RR, Kang K and Fishwick PA (eds). *Proceedings of the 2000 Winter Simulation Conference*. IEEE Press: Orlando, FL, 10–13 December, pp 610–616.

Gendreau M, Laporte G and Séguin R (1995). An exact algorithm for the vehicle routing problem with stochastic demands and customers. *Transportation Science* **29**(2): 143–155.

Gendreau M, Laporte G and Séguin R (1996a). A tabu search heuristic for the vehicle routing problem with stochastic demands and customers. *Operations Research* **44**(3): 469–477.

Gendreau M, Laporte G and Séguin R (1996b). Stochastic vehicle routing. *European Journal of Operational Research* **88**(1): 3–12.

Glover F (1977). Heuristics for integer programming using surrogate constraints. *Decision Sciences* **8**(1): 156–166.

Glover F (1986). Future paths for integer programming and links to artificial intelligence. *Computers and Operations Research* **13**(5): 533–549.

Glover F, Kelly JP and Laguna M (1996). New advances and applications of combining simulation and optimization. In: Charnes JM, Morrice DJ, Brunner DT and Swain JJ (eds). *Proceedings of the 1996 Winter Simulation Conference*. IEEE Press: Coronado, CA, 8–11 December, pp 144–152.

González S, Riera D, Juan AA, Elizondo MG and Fonseca P (2012). Sim-randsharp: A hybrid algorithm for solving the arc routing problem with stochastic demands. In: Laroque C, Himmelspach J, Pasupathy R, Rose O and Uhrmacher AM (eds). *Proceedings of the 2012 Winter Simulation Conference*. IEEE Press: Berlin, Germany, 9–12 December, pp 1–11.

Gutjahr W (2004). S-ACO: An ant-based approach to combinatorial optimization under uncertainty. In: Dorigo M, Birattari M, Blum C, Gambardella L, Mondada F and Stützle T (eds). *Ant Colony Optimization and Swarm Intelligence*. Lecture Notes in Computer Science. Springer-Verlag: Berlin/Heidelberg: pp 238–249.

Gutjahr WJ, Strauss C and Wagner E (2000). A stochastic branch-and-bound approach to activity crashing in project management. *INFORMS Journal on Computing* **12**(2): 125–135.

Holland JH (1973). Genetic algorithms and the optimal allocation of trials. *SIAM Journal on Computing* **2**(2): 88–105.

Ismail Z (2008). Solving the vehicle routing problem with stochastic demands via hybrid genetic algorithm-tabu search. *Journal of Mathematics and Statistics* **4**(3): 161–167.

Jaillet P (1985). *Probabilistic traveling salesman problems*. PhD Thesis, Operations Research Center, MIT, Cambridge, MA.

Juan AA, Barrios BB, Vallada E, Riera D and Jorba J (2014a). SIM-ESP: A simheuristic algorithm for solving the permutation flow-shop problem with stochastic processing times. *Simulation Modelling Practice and Theory* **46**: 101–117.

Juan AA, Faulin J, Grasman SE, Riera D, Marull J and Mendez C (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 AA, Faulin J, Jorba J, Cáceres-Cruz J and Marques J (2013). Using parallel and distributed computing for solving real-time vehicle routing problems with stochastic demands. *Annals of Operations Research* **207**(1): 43–65.

Juan AA, Grasman SE, Caceres-Cruz J and Bektaş T (2014b). A simheuristic algorithm for the single-period stochastic inventory-routing problem with stock-outs. *Simulation Modelling Practice and Theory* **46**: 40–52.

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

Kall P and Wallace SW (1994). *Stochastic Programming*. Wiley-Blackwell: Chichester, UK.

Kennedy J and Eberhart R (1995). Particle swarm optimization. In: *Proceedings IEEE International Conference on Neural Networks*. Perth, Australia, pp 1942–1948.

Kirkpatrick S, Gelatt CD and Vecchi MP (1983). Optimization by simulated annealing. *Science* **220**(4598): 671–680.

Kise H, Shiomi A, Uno M and Chao D (1982). An efficient algorithm for a chance-constrained scheduling problem. *Journal of the Operations Research Society of Japan* **25**(2): 193–203.

Kleinberg J, Rabani Y and Tardos É (2000). Allocating bandwidth for bursty connections. *SIAM Journal on Computing* **30**(1): 191–217.

Konak A and Kulturel-Konak S (2005). Simulation optimization using tabu search: An emperical study. In: Kuhl ME, Steiger NM, Armstrong FB and Joines JA (eds). *Proceedings of the 2005 Winter Simulation Conference*. IEEE Press: Orlando, FL, 4–7 December, pp 2686–2692.

Laporte G, Louveaux FV and Mercure H (1994). A priori optimization of the probabilistic traveling salesman problem. *Operations Research* **42**(3): 543–549.

Laroque C, Klaas A, Fischer J-H and Kuntze M (2012). Fast converging, automated experiment runs for material flow simulations using distributed computing and combined metaheuristics. In: Laroque C, Himmelspach J, Pasupathy R, Rose O and Uhrmacher AM (eds). *Proceedings of the 2012 Winter Simulation Conference*. IEEE Press: Berlin, Germany, 9–12 December, pp 102–111.

Legato P, Mazza RM and Trunfio R (2010). Simulation-based optimization for discharge/loading operations at a maritime container terminal. *OR Spectrum* **32**(3): 543–567.

Lei H, Laporte G and Guo B (2011). The capacitated vehicle routing problem with stochastic demands and time windows. *Computers and Operations Research* **38**(12): 1775–1783.

Lourenço HR, Martin OC and Stützle T (2003). Iterated local search. In: Glover F and Kochenberger G (eds). *Handbook of Metaheuristics*. Kluwer Academic Publishers: Norwell, MA, pp 321–353.

Lourenço HR, Martin OC and Stützle T (2010). Iterated local search: Framework and applications. In: Gendreau M and Potvin JY (eds). *Handbook of Metaheuristics*. Springer: New York, pp 363–397.

Makino T (1965). On a scheduling problem. *Journal of the Operations Research Society Japan* **8**: 32–44.

Marinakis Y, Iordanidou G-R and Marinaki M (2013). Particle swarm optimization for the vehicle routing problem with stochastic demands. *Applied Soft Computing* **13**(4): 1693–1704.

Moghaddam BF, Ruiz R and Sadjadi SJ (2012). Vehicle routing problem with uncertain demands: An advanced particle swarm algorithm. *Computers and Industrial Engineering* **62**(1): 306–317.

Nawaz M, Enscore JEE and Ham I (1983). A heuristic algorithm for m-machine, n-job flow shop sequencing problem. *International Journal of Management Science* **11**(1): 91–95.

Nguyen V-P, Prins C and Prodhon C (2012). A multi-start iterated local search with tabu list and path relinking for the two-echelon location-routing problem. *Engineering Applications of Artificial Intelligence* **25**(1): 56–71.

Penna PHV, Subramanian A and Ochi LS (2011). An iterated local search heuristic for the heterogeneous fleet vehicle routing problem. *Journal of Heuristics* **19**(2): 201–232.

Pinedo M (1984). Optimal policies in stochastic shop scheduling. *Annals of Operations Research* **1**(3): 305–329.

Ross KW and Tsang DHK (1989). The stochastic knapsack problem. *IEEE Transactions on Communications* **37**(7): 740–747.

Rothkopf MH (1966). Scheduling with random service times. *Management Science* **12**(9): 707–713.

Shanmugam G, Ganesan P and Vanathi PT (2011). Meta heuristic algorithms for vehicle routing problem with stochastic demands. *Journal of Computer Science* **7**(4): 533–542.

Shaw P (1998). Using constraint programming and local search methods to solve vehicle routing problems. In: Maher M and Puget JF (eds). *Principles and Practice of Constraint Programming—CP98*. Springer-Verlag: Berlin/Heidelberg, pp 417–431.

Shen Z, Ordóñez F and Dessouky MM (2009). The stochastic vehicle routing problem for minimum unmet demand. In: Chaovalitwongse WA, Furman KC and Pardalos PM (eds). *Optimization and Logistics Challenges in the Enterprise*. Springer Optimization and Its Applications. Springer: New York, pp 349–371.

Steinberg E and Parks MS (1979). A preference order dynamic program for a knapsack problem with stochastic rewards. *Journal of the Operational Research Society* **30**(2): 141–147.

Subramanian A and Battarra M (2012). An iterated local search algorithm for the travelling salesman problem with pickups and deliveries. *Journal of the Operational Research Society* **64**(3): 402–409.

Subramaniam G and Gosavi A (2007). Simulation-based optimisation for material dispatching in vendor-managed inventory systems. *International Journal of Simulation and Process Modeling* **3**(4): 238–245.

Subramanian A, Battarra M and Potts CN (2014). An iterated local search heuristic for the single machine total weighted tardiness scheduling problem with sequence-dependent setup times. *International Journal of Production Research* **52**(9): 2729–2742.

Tan KC, Cheong CY and Goh CK (2007). Solving multiobjective vehicle routing problem with stochastic demand via evolutionary computation. *European Journal of Operational Research* **177**(2): 813–839.

Teodorovic D and Pavkovic G (1992). A simulated annealing technique approach to the vehicle routing problem in the case of stochastic demand. *Transportation Planning and Technology* **16**(4): 261–273.

Tripathi M, Kuriger G and Wan H (2009). An ant based simulation optimization for vehicle routing problem with stochastic demands. In: Rossetti MD, Hill RR, Johansson B, Dunkin A and Ingalls RG (eds). *Proceedings of the 2009 Winter Simulation Conference*. IEEE Press: Austin, Texas, pp 2476–2487.

Vansteenwegen P and Mateo M (2014). An iterated local search algorithm for the single-vehicle cyclic inventory routing problem. *European Journal of Operational Research* **237**(3): 802–813.

Zhang T (2007). RLC_ACS: An improved ant colony algorithm for VRPSDP. In: *Proceedings of 2007 International Conference on Machine Learning And Cybernetics*. IEEE: Hong Kong, pp 978–983.

Zhang T, Chaovalitwongse WA and Zhang Y (2012). Scatter search for the stochastic travel-time vehicle routing problem with simultaneous pick-ups and deliveries. *Computers and Operations Research* **39**(10): 2277–2290.

Received 26 March 2014;
accepted 7 August 2014 after one revision