

## Citation for published version

Gonzalez Martin, S., Elizondo, M., Riera-Terrén, D., Juan, A.A. & Fonseca, P. (2012). Sim-RandSHARP: A Hybrid Algorithm for solving the Arc Routing Problem with Stochastic Demands. Winter Simulation Conference (WSC). Proceedings, 2012(), 1-11.

## DOI

<https://doi.org/10.1109/WSC.2012.6465034>

## Document Version

This is the Submitted Manuscript version.  
The version in the Universitat Oberta de Catalunya institutional repository, O2 may differ from the final published version.

## Copyright and Reuse

This manuscript version is made available under the terms of the Creative Commons Attribution Non Commercial Share Alike licence (CC-BY-NC-SA) <http://creativecommons.org/licenses/by-nc-sa/3.0/>, which permits others to reuse it non-commercially, as long as they credit you and license their new creations under the identical terms.

## Enquiries

If you believe this document infringes copyright, please contact the Research Team at: [repositori@uoc.edu](mailto:repositori@uoc.edu)



## **SIM-RANDSHARP: A HYBRID ALGORITHM FOR SOLVING THE ARC ROUTING PROBLEM WITH STOCHASTIC DEMANDS**

Sergio González  
Daniel Riera  
Angel A. Juan

Mónica G. Elizondo

IN3-Department of Computing, Multimedia and  
Telecommunication  
Open University of Catalonia  
Barcelona, 08018, Spain

Centro de Investigación y Desarrollo Tecnológico  
Universidad Autónoma de Nuevo León  
San Nicolás de los Garza, NL 66450, México

### **ABSTRACT**

This paper proposes a new hybrid algorithm for solving the Arc Routing Problem with Stochastic Demands (ARPSD). Our approach combines Monte Carlo simulation (MCS) with the RandSHARP algorithm, which is designed for solving the Capacitated Arc Routing Problem (CARP) with deterministic demands. The RandSHARP algorithm makes use of a CARP-adapted version of the Clarke and Wright Savings heuristic, which was originally designed for the Vehicle Routing Problem. The RandSHARP algorithm also integrates a biased-randomized process, which allows it to obtain competitive results for the CARP in low computational times. The RandSHARP algorithm is then combined with MCS to solve the ARPSD. In order to do that, the vehicle maximum capacity is restricted somewhat during the routing-design stage. This allows keeping a safety stock which can then be used during the actual delivery stage to cover possible unexpected demands. A reliability index is used to evaluate the robustness of the solution. Some numerical experiments contribute to illustrate the potential benefits of our approach.

### **1 INTRODUCTION**

The Capacitated Arc Routing Problem (CARP) is a well known NP-hard optimization problem which was first introduced by **Golden and Wong (1981)**. It can be informally described as follows. Consider: (a) an incomplete graph or network; (b) a set of customers located on some of the edges of the network with a demand to be satisfied; (c) a cost matrix representing the costs (or distances) of traversing each edge; and (d) an homogeneous fleet of vehicles with limited capacity. Then, the goal is to find a set of routes which minimizes total routing costs while satisfying all customers' demands. Notice that the Arc Routing Problem with Stochastic Demands (ARPSD) is a generalization of the CARP where customers' demands are not deterministic in nature but stochastic, i.e. each demand can be modeled as a random variable. Considering stochastic demands over deterministic ones is a more realistic but also difficult scenario. Unfortunately, most of the existing literature focuses on the deterministic case. Therefore, the main goal of our approach is to contribute to fill the lack of scientific works on the stochastic case by proposing a hybrid algorithm which takes advantage of both heuristics and simulation techniques.

Thus, the main difference between the CARP and ARPSD is that in the former, the customer's demands are known beforehand while in the latter, they have a stochastic nature, which means that only its statistical distribution and expected value are known beforehand, but its exact value is not revealed until the vehicle reaches the customer's edge. This random behavior of the customer's demands could

cause an expected feasible solution to become an unfeasible one if the final demand of any route exceeds the actual vehicle capacity. This situation is referred to as “route failure”, and when it occurs corrective actions must be introduced to obtain new feasible solutions.

The CARP has application in real life problems such as refuse collection, snow removal, inspection of distributed systems or routing of street sweepers. These applications have a stochastic behavior which makes the ARPSD more suitable to model them.

The remaining of this paper is structured as follows. First of all, in **Section 2** a literature review on the ARPSD is presented, including some of the most remarkable results on the area. Next, in **Section 3**, the ARPSD is briefly described. **Section 4** describes the approach we propose for solving the ARPSD. **Section 5** presents some numerical results which are discussed in **Section 6**. Finally, **Section 7**, summarizes the main conclusions of this work.

## 2 LITERATURE REVIEW

The research body concerning the ARPSD is very sparse. As far as we know, the ARPSD was first considered in **Fleury et al. (2002)** and further extended in **Fleury et al. (2005)**. In these works, the ARPSD was not approached directly. Instead, the scope was to evaluate the robustness of solutions for the CARP if the demands were in fact stochastic, and how this robustness could be improved. In particular, the work of **Fleury et al. (2005)** contains an application of the Hybrid Genetic heuristic, originally proposed by **Lacomme et al. (2001)** for the CARP. Different solutions were obtained by varying the vehicle capacity in each run of the heuristic. The solutions obtained were then evaluated by means of simulation studies. The ARPSD with Normal distributed demands was first approached directly by **Fleury et al. (2004)**. The authors propose a Memetic Algorithm, which is an extension of the algorithm suggested by **Lacomme et al. (2004)**. For each edge, the Normal distribution describing the demand was truncated to avoid negative demands and demands that exceeded the vehicle capacity. The problem was further restricted, since a route could only fail once, and if a failure occurred on a route it would always be immediately returned to the depot before serving the last edge.

The first exact algorithm for the ARPSD, and the only one we are aware of, was proposed by **Christiansen et al. (2009)**. Their work is further motivated by a previous work for solving the Vehicle Routing Problem with Stochastic Demands (VRPSD) presented in **Christiansen et al. (2007)**. In particular, they formulate the ARPSD as a Set Partitioning Problem and develop a Branch-and-Price algorithm in which the pricing incorporates demands’ stochastic nature. **Laporte et al. (2010)** have recently developed a local search approach for the stochastic version of the undirected CARP in the context of garbage collection. In their paper, the problem was cast within the framework of a stochastic program with recourse. A first-stage solution is constructed by means of a developed Adaptive Large Neighborhood Search Heuristic (ALNS), which considers the expected cost of recourse. Closed form expressions were derived for the expected cost of recourse by extending the concept of route failure commonly used in the VRPSD. Their computational results show that ALNS solutions were better than those obtained by first optimally solving a deterministic CARP and then computing the expected cost of recourse actions using random variables for the demands.

Finally, other works related to the ARPSD which are worth to be mentioned are found in **Chen et al. (2009)** and **Ismail et al. (2011)**. In the former, the authors address a realistic ARP based on a small-package delivery real-world application. In this problem, uncertainty as whether a street segment requires service on a particular day is considered and incorporated to a proposed model called Probabilistic Arc Routing Problem (PARP). To solve the PARP, they use a solution procedure which incorporates the street segments presence probabilities into an adapted local search. The local search they use was primarily designed by **Bertsimas et al. (1993)** for solving the Probabilistic Traveling Salesman Problem (PTSP) (**Jaillet 1988**). Regarding the work of **Ismail et al. (2011)**, these authors consider a real-life condition of the CARP in waste collection operations. In particular, they study how rain drops affect the total collected waste weight. Two CARP models were suited for the case study, each one considering stochastic demands and time windows. The authors developed a constructive heuristic, the so-called Nearest

Procedure Based on Highest Demand/Cost (NPHDC), which works in conjunction with switching rules to search for the best feasible solution.

### 3 THE ARC ROUTING PROBLEM WITH STOCHASTIC DEMANDS

The Stochastic (or Probabilistic) Arc Routing Problem (SCARP or PCARP) is a family of arc routing problems characterized by the random nature of at least one of their parameters or structural variables. This uncertainty is usually modeled by means of suitable random variables which, in most cases, are assumed to be independent. The ARPSD is a case of this family where the random parameter is the demand of the customers which is modeled by a random variable. Thus, the ARPSD is a combinatorial optimization problem that can be defined as follows. Let  $G = (V, E, C, Q)$  be a connected undirected graph which is incomplete, where:

1.  $V$  is a set of nodes, including the one representing the depot or distribution center.
2.  $E$  is a set of edges or arcs connecting some of the nodes of  $V$ .
3.  $C$  is a cost matrix representing the positive costs related to the movement from one node to another. These costs are usually based on the distances or lengths of the edges which are traversed on the movement from one node to another.
4.  $Q$  is a vector of demands representing the non-negative random demand associated with each edge. Every one of these demands  $Q_i$  is considered to be a random variable with known mean  $E[Q_i]=q_i$  and statistical distribution.

In addition, there is a fleet of identical vehicles, each of them with a maximum capacity  $W \gg \{ E[Q_i] / Q_i \text{ in } Q \}$ . Under these circumstances, the usual goal is to find a set of feasible vehicle routes that minimizes the total delivering costs while satisfying the following constraints:

1. Each route starts and ends at the depot node, so every route is a roundtrip.
2. All edges demands are satisfied.
3. Each edge with positive demand is served by exactly one vehicle. Notice, however, that every edge can be traversed as many times as required by the same or different vehicle.
4. The total demand to be served by any route cannot exceed the vehicle capacity  $W$ .

### 4 OVERVIEW OF OUR APPROACH

Our methodology is based on two main ideas: (a) the ARPSD can be seen as a generalization of the CARP, i.e. the CARP can be considered a special case of the ARPSD where the random demands have zero variance; and (b) while the ARPSD is yet an emerging research area, efficient metaheuristics do already exist for solving the CARP –in fact, state-of-the-art metaheuristics based on the use of Genetic Algorithms, Tabu Search, Simulated Annealing, Ant Colony Optimization or hybrid GRASP are able to provide near-optimal solutions for most known CARP benchmarks.

Accordingly, the key idea behind our approach is to transform the ARPSD instance into a new problem which consists of solving several “conservative” CARP instances, each characterized by a specific risk (probability) of suffering route failures. The term conservative refers to the fact that only a certain percentage of the vehicle total capacity is considered during the routing design phase. The remainder capacity will be free in this phase in order to have it available in case actual demands included in the route served by the vehicle are greater than expected. With that we are protecting somehow from route failures. This empty vehicle capacity can be considered as a safety stock since it reflects the level of extra stock that is maintained to buffer against possible route failures.

This idea is adapted from **Juan et al. (2011)**, where the VRPSD is presented as a generalization of the CVRP. In that paper, a Biased Randomized version of the Clarke and Wright Savings (CWS) heuristic as the algorithm to solve the conservative CVRP instances. In the current paper, we will use the Randomized Savings Heuristic for the Arc Routing Problem (RandSHARP), proposed in **Gonzalez et al. (2012)**. The RandSHARP algorithm is a Biased Randomization of the SHARP heuristic, being the

SHARP an adaption of the CWS heuristic to solve the CARP. The methodology defined to solve the ARPSD in the described way, consists on the following steps (see Fig. 1):

1. Consider an ARPSD instance defined by a set of  $n$  customers. Each customer has associated an stochastic demand characterized by its mean and statistical distribution. For simplicity, assume that all the customers have the same statistical distribution despite everyone have its own mean.
2. Considering the maximum vehicle capacity,  $W$ , set a value for  $k$ , the percentage of the vehicle capacity that can be used in the route design phase. Compute  $W^* = k W$ .
3. Consider a CARP problem named  $CARP(k)$ , consisting on the CARP version of the ARPSD where the customer's demands are represented by their means and the vehicle have the capacity restricted to  $W^*$ .

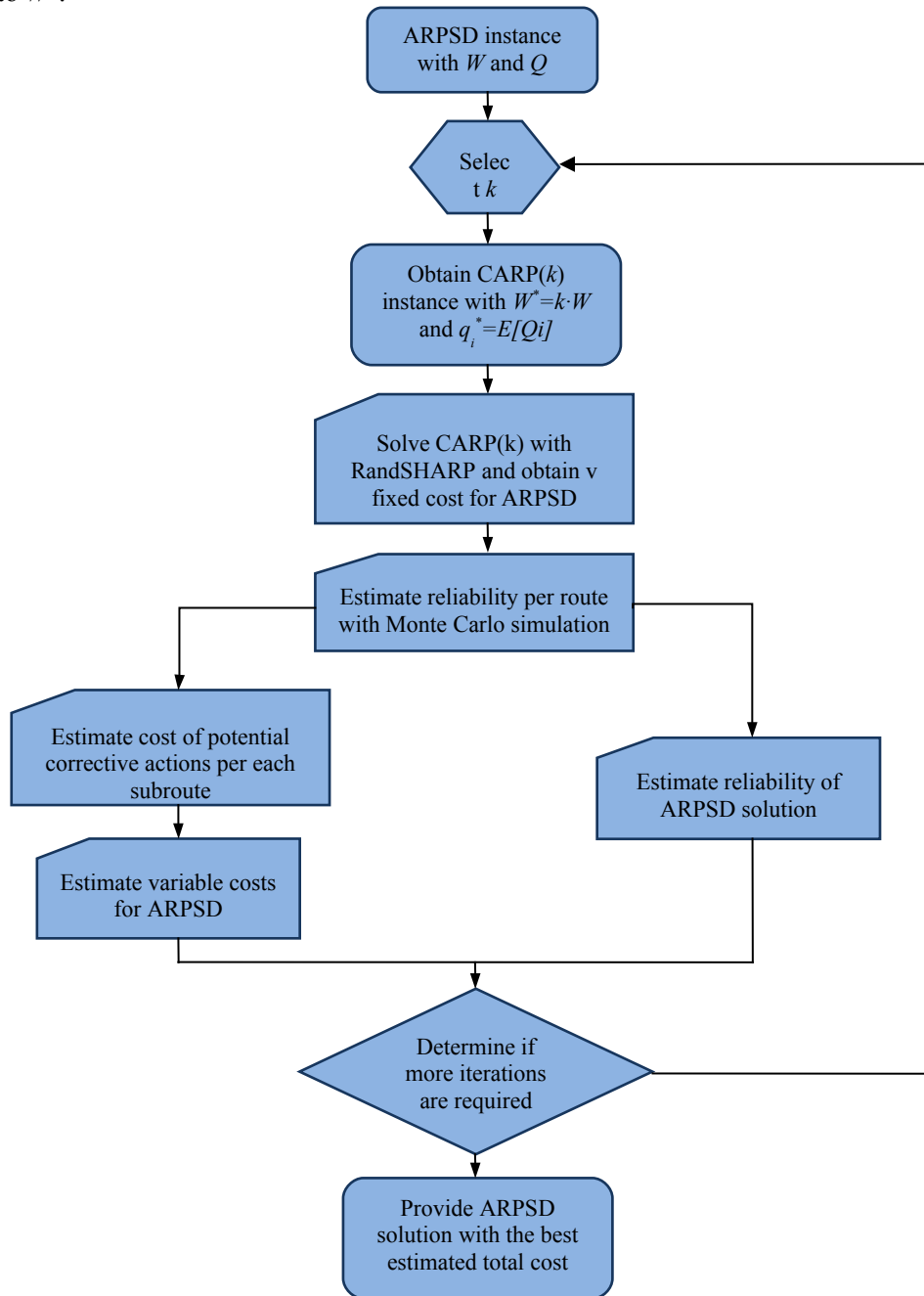


Figure 1 – Flow diagram of the methodology

4. Solve the  $CARP(k)$  problem with the RandSHARP algorithm. This solution will be an aprioristic solution for the original ARPSD. Furthermore, it will be a feasible solution as long as there are not any route failures. This means that the additional demand that might be originated during the execution phase in each route does not exceed the vehicle reserve capacity.
5. Using the previous solution with  $m$  routes, estimate the expected cost due to possible failures on any route. This is done using Monte Carlo simulation. To this end, random demands are generated and whenever a route failure occurs a corrective policy is applied registering the associated cost of this action. Every time a route fails, the cost of a round-trip from the depot to the failing customer's edge is computed as cost of corrective action. After iterating this process some thousands of times, a random sample of observations regarding these variable costs are obtained and an estimate for its expected value can be calculated. Then, the expected total costs due to possible route failures can be computed by the addition of these variable costs and the costs of the deterministic solution obtained in the design phase.
6. Using the aprioristic solution with  $m$  routes, obtain an estimate for the reliability of each route. In such context, the reliability index is defined as the probability that a route will not suffer a route failure. This reliability index is computed by direct Monte Carlo simulation using the statistical distribution that model customer demands in each route. Remark that in each route, over-estimated demands could sometimes be compensated by under-estimated demands.
7. Obtain an estimate for the reliability index associated with the aprioristic solutions. Under the assumption that customer demands are independent, which is a reasonable hypothesis, this can be attained by simply multiplying the reliability of each route. A solution reliability level can be considered as a measure for the feasibility of the solution in the ARPSD context.
8. Depending on the total costs and the reliability indices associated with the solutions already obtained, repeat the process from **Step 1** with a new value of the parameter  $k$  to explore different scenarios.
9. Finally, provide a sorted list with the best ARPSD solutions found so far as well as their corresponding properties such as cost or reliability index.

## 5 COMPUTATIONAL EXPERIMENTS

The methodology described in the previous section has been implemented as a Java application to evaluate its performance. Java SE6 over Netbeans IDE was used here instead of a more efficient language such as C or C++ for several reasons: (a) being an object-oriented programming language with advanced memory management features like the garbage collection and with a readily-available data structures, it allows a somewhat faster development of algorithmic software; (b) it offers immediate portability to different operating systems; and (c) it offers better replicability and duplicability than other languages. However, it has the counterpart that having a poorer performance than C or C++, mainly due to the fact that Java is executed over a virtual machine, adding this some overload to execution. To perform our experiments a standard personal computer was used to perform all tests, an Intel® Core™2 Quad CPU Q9300 at 2.50 GHz and 8 GB RAM running in Windows® 7 Pro operating system.

In order to generate the random demands, and for its use in Biased Randomized process of the RandSHARP algorithm implementation, a state-of-the-art pseudo random number generator has been employed. Specifically, we used the LFSR113 from the SSJ library (L'Ecuyer 2002). To test our methodology, the *gdb* dataset originally proposed in Golden et al. (1983) has been used. This dataset consists of 23 small-medium CARP problem instances, so we will consider the instances assuming that the customer demand defined on the instance is the mean of the stochastic variable which defines actual customer demand on ARPSD. Furthermore, we will consider that this demand is distributed following a Log-Normal statistical distribution. In order to assess the methodology on different circumstances, we

have performed the tests in three different scenarios, each one with a different variance for the Log-normal random variable: (1)  $0.05 \cdot E[Q_i]$ , (2)  $0.25 \cdot E[Q_i]$  and (3)  $0.75 \cdot E[Q_i]$ .

The experiments were run varying the  $k$  parameter from 0.75 to 1.00, with a step of 0.01, to evaluate both conservative and non-conservative scenarios. For each  $k$  parameter, a complete execution was performed with a duration of 180 seconds. Then, the best  $k$  value was selected and compared with the result obtained with the solution without safety stock. Results are displayed on **Tables 1-3**. In these tables, results of computational experiments are displayed. The tables are structured in two halves. The first, marked as *Best Known Solution*, contains the results for the deterministic solutions without safety stock. These solutions correspond to solve the ARPSD as a CARP considering the demands as deterministic with value  $q_i$ . In the other half, marked as *Our Best Solution*, the result for the best  $k$  found is displayed. These solutions are obtained selecting CARP( $k$ ) solution which obtains less total expected costs computed with Monte Carlo simulation.

Table 1: *gdb* dataset results –  $0.05 \cdot E[Q_i]$  variance

Instance Name	Best Known Solution					Our Best Solution				
	Routes	Costs (1)	Expected Costs (2)	Gap (1) – (2)	Reliability	Routes	$k$	Expected Costs (3)	Gap (1) – (3)	Reliability
gdb1	5	316	366.3	15.92%	0.73	5	1	366.3	15.92%	0.73
gdb2	6	345	403.1	16.84%	0.74	6	1	403.1	16.84%	0.74
gdb3	5	275	321.7	16.98%	0.73	5	1	321.7	16.98%	0.73
gdb4	4	287	341.9	19.13%	0.69	4	1	341.9	19.13%	0.69
gdb5	6	383	450.8	17.70%	0.74	6	1	450.8	17.70%	0.74
gdb6	5	298	346.3	16.21%	0.73	5	1	346.3	16.21%	0.73
gdb7	5	325	380.6	17.11%	0.73	5	1	380.6	17.11%	0.73
gdb8	11	360	371.5	3.19%	0.96	11	0.98	371.4	3.17%	0.96
gdb9	11	318	338.2	6.35%	0.91	11	0.98	337.3	6.07%	0.94
gdb10	5	285	307.2	7.79%	0.84	5	0.98	305.8	7.30%	0.83
gdb11	6	413	415.2	0.53%	0.98	6	0.93	413.4	0.10%	0.96
gdb12	7	478	480.9	0.61%	0.99	7	1	480.9	0.61%	0.99
gdb13	7	544	544.3	0.06%	1.00	7	0.93	544.2	0.04%	1.00
gdb14	5	104	106.8	2.69%	0.95	5	0.99	106.7	2.60%	0.95
gdb15	4	58	58.1	0.17%	0.99	4	1	58.1	0.17%	0.99
gdb16	6	129	130.7	1.32%	0.95	6	0.99	130.4	1.09%	0.97
gdb17	5	91	92.4	1.54%	0.95	6	0.99	91.0	0.00%	1.00
gdb18	5	168	174.8	4.05%	0.87	5	0.99	173.2	3.10%	0.87
gdb19	3	55	55.9	1.64%	0.96	3	1	55.9	1.64%	0.96
gdb20	5	123	123.5	0.41%	0.98	5	1	123.5	0.41%	0.98
gdb21	7	159	161.0	1.26%	0.96	7	0.96	160.3	0.82%	0.96
gdb22	9	202	204.5	1.24%	0.95	9	0.99	204.2	1.09%	0.97
gdb23	12	237	238.6	0.68%	0.98	12	1	238.6	0.68%	0.98
<i>Averages</i>				<i>6.67%</i>	<i>0.88</i>		<i>0.987</i>		<i>6.47%</i>	<i>0.89</i>

## 6 DISCUSSION OF PROPOSED METHODOLOGY AND RESULTS

The idea of considering a vehicle safety stock when designing CARP solutions which will be used for ARPSD is not new. **Fleury et al. (2005)** proposes four different scenarios with different values of safety stocks and compared their performance. It differs from our approach since our methodology evaluates a wider range of safety stock values and uses different criteria in the selection of the best solution. Furthermore, our methodology uses an additional Monte Carlo simulation procedure to assure the correct behavior of the solution for the CARP( $k$ ) problem for the ARPSD prior to select it as the solution. When analyzing the proposed methodology, we can remark some benefits from it:

- In some sense, it can be said that our methodology is reducing the complex ARPSD to a limited set of conservative CARP, where high quality algorithms are already available.
- The methodology is not restricted to single CARP algorithm, different algorithms could be used with the methodology in order to obtain robust solutions for the ARPSD.
- The methodology is valid for any statistical distribution, so it can handle different real life scenarios.
- As the decision-maker can consider different solutions with different values of  $k$ , the methodology makes the selection more flexible in order to consider different utility functions.
- The methodology is easily parallelizable as the solution for any value of  $k$  can be computed independently.

Table 2: *gdb* dataset results –  $0.25 \cdot E[Q_i]$  variance

Instance Name	Best Known Solution					Our Best Solution				
	Routes	Costs (1)	Expected Costs (2)	Gap (1) – (2)	Reliability	Routes	K	Expected Costs (3)	Gap (1) – (3)	Reliability
<i>gdb1</i>	5	316	365.3	15.60%	0.74	5	1	365.3	15.60%	0.74
<i>gdb2</i>	6	355	405.5	14.23%	0.75	6	1	405.5	14.23%	0.75
<i>gdb3</i>	5	275	317.5	15.45%	0.75	5	1	317.5	15.45%	0.75
<i>gdb4</i>	4	287	337.2	17.49%	0.71	4	1	337.2	17.49%	0.71
<i>gdb5</i>	6	377	442.5	17.37%	0.74	6	1	442.5	17.37%	0.74
<i>gdb6</i>	5	298	345.2	15.84%	0.75	5	1	345.2	15.84%	0.75
<i>gdb7</i>	5	325	377.6	16.18%	0.75	5	1	377.6	16.18%	0.75
<i>gdb8</i>	11	360	398.8	10.78%	0.87	11	1	398.8	10.78%	0.87
<i>gdb9</i>	11	322	363.9	13.01%	0.82	11	0,98	362.7	12.64%	0.85
<i>gdb10</i>	4	275	309.8	12.65%	0.68	4	1	309.8	12.65%	0.68
<i>gdb11</i>	6	411	428.2	4.18%	0.90	6	0,95	425.5	3.53%	0.92
<i>gdb12</i>	7	490	506.6	3.39%	0.97	8	0,95	503.4	2.73%	0.97
<i>gdb13</i>	7	544	551.9	1.45%	0.95	7	0,94	551.8	1.43%	0.95
<i>gdb14</i>	6	106	111.0	4.72%	0.90	6	1	111.0	4.72%	0.90
<i>gdb15</i>	4	58	59.4	2.41%	0.91	4	1	59.4	2.41%	0.91
<i>gdb16</i>	7	131	136.0	3.82%	0.91	7	0,93	134.7	2.82%	0.93
<i>gdb17</i>	6	91	92.7	1.87%	0.93	6	0,85	92.2	1.32%	0.95
<i>gdb18</i>	5	166	179.5	8.13%	0.79	5	0,98	177.2	6.75%	0.84
<i>gdb19</i>	3	55	58.4	6.18%	0.85	3	1	58.4	6.18%	0.85
<i>gdb20</i>	5	123	127.1	3.33%	0.91	5	1	127.1	3.33%	0.91
<i>gdb21</i>	7	158	165.7	4.87%	0.87	7	0,98	165.4	4.68%	0.86
<i>gdb22</i>	10	204	210.5	3.19%	0.90	10	0,95	210.0	2.94%	0.90
<i>gdb23</i>	12	237	247.4	4.39%	0.88	12	0,99	246.2	3.88%	0.89
<i>Averages</i>				8.72%	0.83		0.978		8.48%	0.84

Considering the results, it is clear that the more variance the customer’s demand has, the greater improvement is obtained with the use of safety stocks. Nevertheless, notice also that depending on the concrete problem instance, we obtain no gain with the use of safety stock. For example, for instances *gdb1* to *gdb8* we obtained that always the best solution were not those using a safety stock. Reviewing each one of these instances, it can be noticed that the total demand to be served is not an exact multiple of the vehicle capacity. So these instances, somehow, already have an implicit safety stock. Despite not considering this during the design phase, it is improving the robustness of the solutions as it allows overcoming some unexpected over-demands. For the instances where an improvement is obtained, it is clear that it is great as the customer’s demands are more disperse. Notice also that in this case we have not considered additional penalization when a route failure occurs, than a roundtrip of the vehicle to the depot to be loaded. In some cases there can be additional penalization costs (e.g due to the time lost in the



vehicle reload procedure) which will increase the over-cost of these recourse action and then increment the gap between our best solution and the solution without safety stock.

Table 3: *gdb* dataset results –  $0.75 \cdot E[Q_i]$  variance

Instance Name	Best Known Solution					Our Best Solution				
	Routes	Costs (1)	Expected Costs (2)	Gap (1) – (2)	Reliability	Routes	k	Expected Costs (3)	Gap (1) – (3)	Reliability
gdb1	5	316	362.9	14.84%	0.77	5	1	362.9	14.84%	0.77
gdb2	6	345	395.1	14.52%	0.78	6	1	395.1	14.52%	0.78
gdb3	5	275	315.0	14.55%	0.77	5	1	315	14.55%	0.77
gdb4	4	287	330.7	15.23%	0.75	4	1	330.7	15.23%	0.75
gdb5	6	377	435.8	15.60%	0.76	6	1	435.8	15.60%	0.76
gdb6	5	298	340.0	14.09%	0.77	5	1	340	14.09%	0.77
gdb7	5	325	373.6	14.95%	0.77	5	1	373.6	14.95%	0.77
gdb8	10	366	416.5	13.80%	0.79	10	1	416.5	13.80%	0.79
gdb9	11	318	382.9	20.41%	0.72	11	0.98	381.2	19.87%	0.77
gdb10	4	275	309.0	12.36%	0.70	4	1	309	12.36%	0.70
gdb11	6	411	440.9	7.27%	0.83	6	0.92	440	7.06%	0.83
gdb12	7	474	527.9	11.37%	0.87	7	1	527.9	11.37%	0.87
gdb13	7	544	566.6	4.15%	0.88	7	0.97	565.8	4.01%	0.88
gdb14	6	106	114.0	7.55%	0.86	6	0.95	113.8	7.36%	0.86
gdb15	4	58	61.3	5.69%	0.81	4	0.84	60.9	5.00%	0.83
gdb16	6	129	138.8	7.60%	0.81	6	0.92	138.5	7.36%	0.81
gdb17	6	91	94.8	4.18%	0.87	6	0.86	94.2	3.52%	0.88
gdb18	5	170	187.5	10.29%	0.71	5	0.98	182.8	7.53%	0.73
gdb19	3	55	60.9	10.73%	0.80	3	1	60.9	10.73%	0.80
gdb20	5	123	130.4	6.02%	0.85	5	0.93	130.3	5.93%	0.84
gdb21	8	162	172.4	6.42%	0.86	7	0.95	170.6	5.31%	0.80
gdb22	10	204	216.6	6.18%	0.83	10	0.93	214.6	5.20%	0.84
gdb23	12	237	253.2	6.84%	0.81	12	0.94	252.1	6.37%	0.81
<i>Averages</i>				10.64%	0.79		0.963		10.29%	0.80

Finally, regarding the reliability index, remark that the greater the variance is, lesser values of the reliability index are obtained. This is due to the fact that in case of high variances, over-demands can occur with greater values, so route failures are more likely to occur. Notice also that the gap obtained is greater as greater is the variance for the same reason, as more over-demand is get, greater penalty due to recourse actions is incurred.

## 7 CONCLUSIONS

This paper has presented a hybrid methodology for solving the ARPSD. The methodology, which combines Monte Carlo simulation with heuristics, is based on the idea of using safety stocks during the design stage so that unexpected demands can be satisfied during the delivery stage. The results for small- and medium-size instances have shown that an improvement on solutions robustness can be achieved by employing the appropriate safety stocks. However, sometimes the over-cost of this improvement on the robustness does not compensate the cost of corrective actions required with less robust solutions obtained without safety stocks. For future work it will be required to study high-size instances to check the performance of the methodology in those larger problems. In addition, a parallelization of the algorithm will be required as greater instances will require greater execution times to obtain high-quality solutions. Also, the addition of local search processes to the RandSHARP algorithm could be helpful to improve the quality of solutions in medium- and high-size instances.

## ACKNOWLEDGMENTS

This work has been partially supported by the Spanish Ministry of Science and Innovation (TRA2010-21644-C03) and by the Ibero-American Programme for Science, Technology and Development (CYTED2010-511RT0419) in the context of the IN3-HAROSA Network (<http://dpcs.uoc.edu>).

## REFERENCES

- Bertsimas, D.; and Howell, L. (1993). Further results on the probabilistic traveling salesman problem. *European Journal of Operational Research*, 65(1): 68-95.
- Chen, S.; Golden, B.; Wong, R.; and Zhong, H. (2009). Arc-routing models for small-package local routing. *Transportation Science*, 43(1): 43-55.
- Christiansen, C.; 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, C.; 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.
- 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*: 290-295.
- Fleury, G.; Lacomme, P.; and Prins, C. (2004). Evolutionary algorithms for stochastic arc routing problems. In G. R. Raidl, F. Rothlauf, G. D. Smith, G. Squillero, S. Cagnoni, J. Branke, D. W. Corne, R. Drechsler, Y. Jin, and C. G. Johnson, eds. *Applications of Evolutionary Computing*, volume 3005 of *Lecture Notes in Computer Science*. Springer-Verlag, Berlin and Heidelberg: 501-512.
- 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.
- Golden, B.; and Wong, R. (1981). Capacitated arc routing problems. *Networks*, 11(3): 305-315.
- Golden, B.; DeArmon, J.; and Baker, E. (1983). Computational experiments with algorithms for a class of routing problems. *Computers & Operations Research*, 10(1): 47-59.
- González, S.; Juan, A.; Riera, D.; Castellá, Q.; Muñoz, R.; and Pérez, A. (accepted). Development and Assessment of the SHARP and RandSHARP algorithms for the Arc Routing Problem. *AI Communications*.
- Ismail, Z.; and Ramli, M. (2011). Implementation weather-type models of capacitated arc routing problem via heuristics. *American Journal of Applied Sciences*, 8(4): 382-392.
- Jaillet, P. (1988). A priori solution of a traveling salesman problem in which a random subset of the customers are visited. *Operations Research*, 36(9):929-936.
- Juan, A.; Faulin, J.; Grasman, S.; 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*, 19: 751-765.
- L'Ecuyer, P. 2002. SSJ: A Framework for Stochastic Simulation in Java. In Proceedings of the 2002 Winter Simulation Conference 234-242.
- Lacomme, P.; Prins, C.; and Ramdane-Chérif, W. (2001). Competitive genetic algorithms for the capacitated arc routing problem and its extensions. *Lecture Notes in Computer Science*, 2037: 473-483.
- Lacomme, P.; Prins, C.; and Ramdane-Chérif, W. (2004). Competitive memetic algorithms for arc routing problems. *Annals of Operations Research*, 131(1-4):159-185.
- Laporte, G.; Musmanno, R.; and Vocaturo, F. (2010). An adaptive large neighbourhood search heuristic for the capacitated arc-routing problem with stochastic demands. *Transportation Science*, 44(1): 125-135.

## **AUTHOR BIOGRAPHIES**

**SERGIO GONZÁLEZ** is a Ph.D. candidate of Applied Optimization and Simulation in the Computer Science Department at the IN3-Open University of Catalonia (Barcelona, Spain). Mr. González holds a M.S. in Telecommunication Engineering and a M.S. in Software Libre. His research interests include Applied Optimization and Randomized Algorithms with focus on its Telecommunication application. His e-mail is [sgonzalezmarti@uoc.edu](mailto:sgonzalezmarti@uoc.edu).

**DANIEL RIERA** is Lecturer in the Department of Computer Science, Multimedia and Telecommunication (EIMT). He is academic director of Master in Bioinformatics (UOC), and also of Computer Science Engineering (EIMT). Dr. Daniel holds a Ph.D. in Computer Science (UAB), a M.S. in Advanced Techniques of Processes Automation (UAB), and a M.S. in Computer Science (UAB). His main research scopes include the model of discreet systems using Petri nets, and the optimization using Constraint Programming techniques. He has been researcher of the LOGISIM, Centre of simulation and optimization of logistic systems, from the Network of Centers of Support to Technological Innovation (XIT) of the CIDEM. Currently, he belongs to the research group in Software Engineering of GRES-UOC where investigates the verification of UML+OCL models by means of Constraint Programming. His email address is [driera@uoc.edu](mailto:driera@uoc.edu) and his web page is <http://www.uoc.edu/webs/drierat>.

**ANGEL A. JUAN** is an Associate Professor of Applied Optimization and Simulation in the Computer Science Department at the IN3-Open University of Catalonia (Barcelona, Spain). Dr. Juan holds a Ph.D. in Applied Computational Mathematics, an M.S. in Information Systems & Technology, and an M.S. in Applied Mathematics. His research interests include Applied Optimization and Randomized Algorithms. He has published over 100 peer-reviewed papers in international journals, books and proceedings. Currently, he is the coordinator of the CYTED-IN3-HAROSA@IB Network and editorial board member of both the Int. J. of Data Analysis Techniques and Strategies and the Int. J. of Information Systems & Social Change. His e-mail is [ajuanp@gmail.com](mailto:ajuanp@gmail.com) and his web page is <http://ajuanp.wordpress.com>.

**MÓNICA G. ELIZONDO** is a Ph.D. candidate of Systems Engineering in the Graduate Program in Systems Engineering at the Universidad Autónoma de Nuevo León (Nuevo León, Mexico). Miss. Elizondo holds a M.S. in Systems Engineering. Her research interests include Combinatorial Optimization, mathematical modeling of optimization problems and the development of efficient algorithms in the operation research field to solve such problems. Her e-mail is [moni@yalma.fime.uanl.mx](mailto:moni@yalma.fime.uanl.mx).