The Smurf routing problem: a stochastic programming approach

Mauro Dell'Amico and Brainy Smurf

Abstract It is well known that Smurf live in mushroom-like houses located in a clearing in the middle of a deep forest with grass, a river, and vegetation. Humans such as Gargamel are shown to live nearby, though it is almost impossible for them to find the Smurf village. However, Smurfs need to travel in the forest to find food and other things necessary for the everyday life. During these trips several risks arise and unexpected dangers may appear. In this work we present a stochastic programming approach to the Smurf routing problem. The algorithm has been embedded in a Decision Support System which is actually used by Papa Smurf every time an excursion outside the village must be planned.

Key words: Smurf, Stochastic Programming, Routing, Heursitic

1 Introduction

From the reminders of our youth the Smurf life is an idyllic life full of joy and adventures. Smurf live in mushroom-like houses similar to one another, located in a nice clearing in the middle of a deep forest with grass, a river, and vegetation. Humans such as Gargamel live nearby, though it is almost impossible for them to find the Smurf village without the guide of a Smurf. However, in real life Smurfs must face several day-to-day problems and have to face risks and hazards. Indeed, Smurfs need to travel in the forest to collect food and other things they need for the everyday life. During these trips several risks arise and unexpected dangers may appear. The evil Gargamel is always in search of Smurfs and goes around the forest

Mauro Dell'Amico
DISMI, Universitá di Modena e Reggio Emilia, Reggio Emilia, Italy, e-mail: mauro.dellamico@unimore.it

Brainy Smurf
Smurf village institute, Smurf forest, Le Pays Maudit, e-mail: bsmurf@smurfmail.be

looking for traces of Smurfs. His cat Azrael search to capture Smurfs beforehand Gargamel do.

Reducing the risk of finding Gargamel, Azrael or several other risks hidden in the forest is a primary objective of Papa Smurf, the old Smurf leading the community. Using his wide knowledge of the forest, of Gargamel and of his Smurf's characters, we carefully plan the composition of the staffs that travel in the fores to collect the items they need, and even more carefully the path that these staff have to follow in the forest.

Brainy Smurf, the most intelligent Smurf, recently went through some books of operations research and realized that much safer routes can be obtained applying appropriate optimization algorithms. We were asked by Brainy to look at the problem and produce an optimization algorithm to solve it.

In this paper we introduce for the first time the stochastic *Smurf Routing Problem* (SmfRP), design an L-shaped approach, and proof, through computational experiments, it effectiveness in addressing the problem. The algorithm was later embedded into a Decision Support System which is actually used by Papa Smurf every time an excursion outside the village must be planned.

In the next Section 2 we give a detailed description of the real life problem and in Section 2.1 we shortly resume the relevant literature. Section 3 is devoted to introduce mathematical models and solution algorithms for our problem, while Section 4 provides computational results and describes the overall decision support system, in Section 4.1. The last Section 5 summarizes our work.

2 Problem description

Traveling in the forest to collect food is a recurrent smurfing activity that Papa Smurf must carefully organize and plan several times during the year. The problem is intrinsically stochastic since we do not know how much food can be found in each place of the forest.

The stochastic Smurf Routing Problem is modeled on a complete digraph $G = (V,A)$, where $V = \{0,1,\dots,n\}$ is the set of vertices including the n places in the forest, where food may be collected and the village (vertex 0). Let us define the set $V_0 = V \setminus \{0\}$. A is the set of arcs between each pair of vertices. To each arc $(i,j) \in A$ is associated a non-negative risk r_{ij} mainly due to the possible presence of Gargamel. For each vertex $j \in V_0$ a quantity of food q_j^ω is given for every scenario $\omega \in \Omega$, where Ω is the set of all the possible scenarios and p^ω is the probability of the scenario ω, knowing that $\sum_{\omega\in\Omega} p^\omega = 1$. The required quantity of food to be collected in scenario ω is q^ω. Moreover, the maximum food quantity than can be carried by a crew is Q.

Papa Smurf want to design $m > 0$ routes, one for each Smurf crew, which start and return to the village, minimizing the total risk of encountering Gargamel and collecting the required quantity of food, if possible.

2.1 Literature review

Stochastic optimization and, in particular, stochastic vehicle routing problems have been studied in several works and solved with diverse methodologies in the last years, we refer the interested reader to [2] and [5]. The stochastic vehicle routing problems where the stochastic variables are the demands have been modeled and solved with a chance constrained programming approach (see, e.g., [8]), with a robust optimization approach (see, e.g, [1]), and with stochastic programming with recourse (see, e.g., [6]).

A recent survey on Smurf life can be found in [3].

3 Mathematical formulation and solution method

The first formulation we introduce is the *Deterministic Equivalent Program* (DEP), that is the formulation considering the decisions to be taken before and after the realization of the stochastic variables in one model, a large model with all the scenarios. We define variables $x_{ij}, (i,j) \in A$, that can take value 1 if arc (i,j) is used and 0 otherwise. Variable f_{ij}^{ω} represents the quantity flowing on the arc (i,j) in scenario $\omega \in \Omega$. Variables s_j^{ω} are the slack variables used to pay a cost for each unit of missing food, i.e., when the flows do not respect the request q^{ω} of scenario $\omega \in \Omega$.

$$\text{(DetSRP)} \min \sum_{\omega \in \Omega} p^{\omega} \left(\sum_{i \in V} \sum_{j \in V} r_{ij}^{\omega} x_{ij} + \alpha s_j^{\omega} \right) \tag{1}$$

$$\sum_{i \in V} x_{ij} = 1 \qquad j \in V_0 \tag{2}$$

$$\sum_{i \in V} x_{ji} = 1 \qquad j \in V_0 \tag{3}$$

$$\sum_{j \in V} x_{0j} \leq m \tag{4}$$

$$\sum_{i \in S} \sum_{j \in S} x_{ij} \leq |S| - 1 \qquad S \subseteq V_0, S \neq \emptyset \tag{5}$$

$$f_{ij}^{\omega} \leq Q x_{ij} \qquad (i,j) \in A, \omega \in \Omega \tag{6}$$

$$\sum_{i \in V} f_{ji}^{\omega} - \sum_{i \in V} f_{ij}^{\omega} \leq q_j^{\omega} \qquad j \in V_0, \omega \in \Omega \tag{7}$$

$$\sum_{i \in V_0} f_{i0}^{\omega} = q^{\omega} + s^{\omega} \qquad \omega \in \Omega \tag{8}$$

$$s^{\omega} \geq 0 \qquad \omega \in \Omega \tag{9}$$

$$f_{ij}^{\omega} \geq 0 \qquad (i,j) \in A, \omega \in \Omega \tag{10}$$

$$x_{ij} \in \{0,1\} \qquad (i,j) \in A. \tag{11}$$

Objective function (1) uses parameter $\alpha \geq 0$ to define a weighted combination of two sub-objectives: minimizing the risk and collecting all the required food. A two-stage Stochastic Integer Programming model with simple recourse for the SRP can be obtained as follows.

Stage 1 Consider the routing part of the problem and solve it to an optimum.
Stage 2 Consider a set of models, one for each scenario, that take as input the decisions made at the first stage. Make each scenario fit with the request of the various scenarios after the realization of the stochastic variables. To perform this fitting, a cut is produced and inserted into the first stage model.

Both models must be solved in an iterative way: when the inequalities produced by the second stage models do not cut the solution of the first stage then the optimal solution has been found.

$$\text{(1-Stage)} \qquad \min \sum_{i\in V}\sum_{j\in V} r_{ij}x_{ij} + E(z(\omega, s^-)) \tag{12}$$
$$(2)-(5),(11).$$

The objective function (12) of the first stage model contemplates the traveling risk and the expected value of the solution of the second stage that defines the costs due to the food slack penalties.

$$\text{(2-Stage)} \qquad \min z(\omega, s^-) = \alpha \sum_{j\in V_0} (s^{\omega -}) \tag{13}$$
$$f_{ij}^{\omega} \leq Q\bar{x}_{ij} \qquad (i,j)\in A \tag{14}$$
$$(7)-(10).$$

where $\bar{x}$ are the variable obtained in the solution of the first stage.

Property 1. The two stages method terminates with the global optimal solution.

Proof. Directly descends form Theorem 1 in Chapter 3 of [7] and the Smurfing nature of the second stage. □

To make the solution of the problem more effective we can apply and L-shaped method with feasibility Smurfing cuts.

Theorem 1. *Give a solution $\bar{x}$ of the first stage problem, let $\hat{\xi}_{ij}^{\omega}$ be the dual variables associated with (14) and ϕ_j^{ω} the dual variables of (7), the Smurfing cut*

$$Q \sum_{(i,j)\in A} \xi_{ij}^{\omega}\bar{x}_{ij} \geq - \sum_{j\in V\setminus\{0\}} \phi_j^{\omega} q_j^{\omega} \tag{15}$$

is valid for the first phase problem.

Proof. One can easily Smurf the continuous relaxation of the first phase, so by Smurfing the Lagrangean duality theorem, (15) holds. □

The implementation of the method requires some more details.

1. Once phase 1 is solved we check heuristically the Smurf cuts with a greedy algorithm.
2. Iterating phase 1 and phase 2 is reapeted until a Smurfing threashold is reaches, showing that the process is poorly Smurfing. In this case we resume to the deterministic formulation.
3. No Smurf tentative is implemented. This is an issue for a possible future research.

In order to speed up the algorithm we have also implemented a metaheuristic algorithm based on Iterate Smurf Search (see, e.g., [4]). The critical elements for the

Algorithm 1 Iterated Smurf search

Input: a SRP instance;
Use a SmurfGreedy to construct a feasible solution $\mathscr{R}$ with value z^*;
$k \leftarrow 1$;{static initialization}
repeat
  Explore the Smurf neighborhood N_k giving the best solution r of value z
  if $z = \infty$ **and** $k = k_{max}$ **then**
    return
  else if $z = \infty$ **then**
    $k \leftarrow Rand(1, k_{max})$ {Smurfing the neighborhoods}
  else if $z < z^*$ **then**
    $R \leftarrow r, z^* \leftarrow z$
  else
    $k \leftarrow k + 1$
  end if
until No improvement **or** timelimit

effectiveness of the heuristic described in algorithm 1 are as follows.

- At least four Smurf neighborhoods must be available.
- Uniform random Smurfing can be substituted with a different aleatoric function;
- The termination criteria can be Smurfed according to the specific problem.

4 Computational experiments

In order to assess the effectiveness of our algorithms we have generated a preliminary set of random instances simulating possible real ones. Instances SmurfRand1-SmurfRand5 have : n uniform random in [100-200], $|\Omega|$ uniform random in [1000-2000], q_j^{ω} Smurfing random in [5,30], and all other parameters Smurfed in similar way. We run our experiments on a SmurfBlade M1X200 at 3.6MHz, with 16 Gbyte of Smurf memory.

The next Table 1 reports the computational results using the deterministic model and the L-shaped approach. A time limit of 1 hour has been given to each algorithm.

Table 1 Results for the random instances.

	DetSRP				L-Shaped			
Instance	UB	LB	%gap	time	UB	LB	%gap	time
SmurfRand 1	16900.00	16900.00	0.00	5.21	16900.00	16900.00	0.00	2.63
SmurfRand 2	81192.12	81192.12	0.00	20.93	81192.12	81192.12	0.00	7.46
SmurfRand 3	79008.23	78258.87	0.95	3600.00	78901.00	78901.00	0.00	2734.12
SmurfRand 4	40303.87	40303.87	0.00	138.268	40303.87	40303.87	0.00	44.922
SmurfRand 5	41870.00	35755.39	14.60	3600.00	40938.00	37149.11	9.26	3600.00
Average			3.11	1473.57			1.85	1689.89

One can observe that the L-Shape method Smurf better than the Deterministic model, both in terms of solution quality and computing time.

4.1 A decision support system

In order to make the algorithms easy to use we have embedded them, into a graphical Decision Support System. We implemented the interface using a web service with a client-server technology using PHP on the server side and Smurfscript con the client. Figure 1 show he home page of the client application. From this page the user can define the scenarios with an intuitive Smurfing interface. The best solution found can be analyzed through Smurf analytics and Smurfing graphs adaptable to different devices. Papa Smurf is using this application from January 2018 with considerable time saving and improvement of the quality of the previous hand made plans Quoting Papa Smurf:

> The DSS is extremely Smurfing! I am able to plan a clever forest Smurfing tour which Smurf on all the previous Smurfs !

5 Conclusions

We introduced the novel stochastic Smurf Routing Problem. We have designed a deterministic mathematical model and an L-Shaped solution approach. An heuristic algorithm implementing a Smurf search has been used as starting solution. Computational experiments show the effectiveness of the L-shaped method. A Decision Support System embeds the algorithms and is in use by the Smurfs for their daily plans.

Acknowledgements We are indebted with dr. Stefano Novellani for several helpful discussions and his implementations of the L-Shaped method. We also thanks the Smurf people for their patience while disclosing us the SRP.

Fig. 1 The Decision Support System home page.

References

1. Bertsimas, D.J., Simchi-Levi, D.: A new generation of vehicle routing research: robust algorithms, addressing uncertainty. Operations Research **44**(2), 286–304 (1996)
2. Birge, J.R., Louveaux, F.: Introduction to stochastic programming. Springer Science & Business Media (2011)
3. France-Presse, A.: Smurfs preparing big 50th birthday celebrations. The China Post. **January 16** (2008)
4. Hansen, P., Mladenović, N.: Variable neighborhood search: Principles and applications. European journal of operational research **130**(3), 449–467 (2001)
5. King, A.J., Wallace, S.W.: Modeling with stochastic programming. Springer Science & Business Media (2012)
6. Laporte, G., Louveaux, F.V., Van Hamme, L.: An integer l-shaped algorithm for the capacitated vehicle routing problem with stochastic demands. Operations Research **50**(3), 415–423 (2002)
7. Schrijver, A.: Theory of linear and integer programming. John Wiley & Sons (1998)
8. Stewart, W.R., Golden, B.L.: Stochastic vehicle routing: A comprehensive approach. European Journal of Operational Research **14**(4), 371–385 (1983)