J Optim Theory Appl (2009) 143: 309–328
DOI 10.1007/s10957-009-9561-0

Stochastic Second-Order Cone Programming in Mobile Ad Hoc Networks

F. Maggioni · F.A. Potra · M.I. Bertocchi · E. Allevi

Published online: 14 May 2009

Abstract We propose a two-stage stochastic second-order cone programming formulation of the semidefinite stochastic location-aided routing (SLAR) model, described in Ariyawansa and Zhu (Q. J. Oper. Res. 4(3), 239–253, 2006). The aim is to provide a sender node S with an algorithm for optimally determining a region that is expected to contain a destination node D (the expected zone). The movements of the destination node are represented by ellipsoid scenarios, randomly generated by uniform and normal distributions in a neighborhood of the starting position of the destination node. By using a second-order cone model, we are able to solve problems with a much larger number of scenarios (20250) than it is possible with the semidefinite model (500). The use of a larger number of scenarios allows for the computation of a new expected zone, that may be very effective in practical applications, and for obtaining stability results for the optimal first-stage solutions and the optimal cost function values.

Keywords Mobile ad-hoc networks · Second-order cone programming · Stochastic programming · Scenarios generation

Communicated by A. Miele.

The work of F.A. Potra was supported in part by the National Science Foundation under Grant No. 0139701.

F. Maggioni · M.I. Bertocchi
Department of Mathematics, Statistics, Computer Science and Applications, University of Bergamo, Via dei Caniana 2, Bergamo 24127, Italy

F.A. Potra (✉)
Department of Mathematics & Statistics, University of Maryland, Baltimore County, USA
e-mail: fpotra@gmail.com

E. Allevi
Department of Quantitative Methods, University of Brescia, Contrada S. Chiara, 50, Brescia 25122, Italy

1 Introduction

Wireless mobile hosts, able of communicating with each other in the absence of a fixed infrastructure, are becoming increasingly important in modern communications. The Mobile Ad hoc NETworks (MANET), consisting of wireless mobile nodes, and the related routing protocols have been extensively studied in the last 15 years (see [1, 2] and the references therein). The routing protocols usually differ in the assumptions made when searching for a new route. In order to decrease the overhead of route discovery, Ko and Vaidya [1] suggest a special approach based on the use of local information. Their algorithm, known as the Location-Aided Routing (LAR) protocol, tries to reduce the number of nodes to whom the requested route is propagated by use of local information given, for example, by the Global Positioning System (GPS). The main concepts behind the algorithm are the expected zone, and the requested zone. The former is the region where the sender node S expects to find the destination node D in an elapsed time t_1, while the latter is the region defined by the sender node for spreading the route request to the destination node. The route being requested spreads over only if a neighbor node belongs to the requested zone. Since the requested zone should include the expected zone, it is very important to properly identify the expected zone. Ariyawansa and Zhu [3] proposed to structure the problem of the identification of the expected zone as a two-stage stochastic semidefinite programming problem, called stochastic location-aided routing (SLAR). In [4], a similar formulation was used in conjunction with a greedy process of directional search, and numerical results were given for a limited number of scenarios. The results from [4] reveal that, in general, SLAR can significantly reduce the overall overhead than existing deterministic algorithms, simply because the location uncertainty in the routing problem is better captured by this model.

In the present paper we formulate SLAR as a stochastic two-stage second-order cone programming. Since the computational complexity of solving the second-order cone programming problem is significantly lower than the computational complexity of solving the corresponding semidefinite programming problem, we were able to solve problems with a much larger number of scenarios (20250) than it was possible with the semidefinite model (500). The solution of the stochastic two-stage second-order cone programming with many scenarios allows us to compute a new expected zone that it is very appropriate for practical applications, and to validate our model.

Section 2 of our paper introduces some basic notions from stochastic semidefinite programming and second-order cone programming. Section 3 describes the formulation of SLAR as a stochastic two-stage semidefinite programming problem. Section 4 contains the formulation of SLAR as a stochastic two-stage second order cone programming problem. Finally, in Sect. 5, we describe the strategies used for scenario generation, we present the numerical results, and discuss their validation.

2 Basic Facts and Notation

Throughout this paper, $\Re^n$ denotes the space of all n-dimensional real vectors, $\Re^{n\times n}$ denotes the space of real $n \times n$ matrices, and $\Re_s^{n\times n}$ denotes the space of real $n \times n$

symmetric matrices. For $A \in \Re_s^{n\times n}$, $A \succeq 0$ ($A \succ 0$) means that A is positive semidefinite (positive definite). If $A, B \in \Re_s^{n\times n}$, then $A \succeq B$ ($A \succ B$) means that $A - B \succeq 0$ ($A - B \succ 0$). A subscripted vector, x_i, represents the ith block of a vector $x \in \Re^n$. The jth component of the vectors x and x_i are indicated by x_j and x_{ij}. We use $0 \in \Re^n$ and $1 \in \Re^n$, for the zero vector and the vector of all ones, respectively. We also use 0 and I, for the zero and identity matrices.

With the above notation, a semidefinite programming problem (SDP), in dual form, can be written as

$$\begin{aligned} \min \quad & c^T x, \\ \text{s.t.} \quad & \sum_{i=1}^{n} x_i A_i \preceq B. \end{aligned} \tag{1}$$

Here, $c \in \Re^n$, $A_1, \ldots, A_n, B \in \Re_s^{m\times m}$ are given data, and $x = (x_1, \ldots, x_n)^T \in \Re^n$ is a vector of decision variables. If the matrices $A_1, \ldots, A_n, B$ are diagonal, then the above problem reduces to the classical linear programming problem (LP), which asks for minimizing a linear function of the decision variables subject to m linear inequality constraints. The main difference between SDP and LP is that in the former case, inequalities are defined by the cone of positive semidefinite matrices. While linear programming has been studied for more than 60 years, semidefinite programming is a relatively new field in optimization that has widespread applications [5–8].

Since the data $\mathcal{D} = (c, A_1, \ldots, A_n, B)$, defining (1), are supposed to be completely known, we say that (1) is a deterministic semidefinite programming problem (DSDP). Stochastic programming was introduced in the 1950s, as a paradigm for dealing with uncertainty in data related to linear programming [9, 10]. The concept of two-stage stochastic linear programming, was subsequently discussed in numerous other publications (see, for example, the monographs [11, 12]). Recently, Ariyawansa and Zhu [13] have proposed an extension of this concept to the case of semidefinite programming. The two-stage stochastic semidefinite programming (SSDP) corresponding to (1) is formulated as

$$\begin{aligned} \min \quad & c^T x + E[Q(x,\omega)], \\ \text{s.t.} \quad & \sum_{i=1}^{n} x_i A_i \preceq B, \end{aligned} \tag{2}$$

where $Q(x,\omega)$ is the minimum of the problem

$$\begin{aligned} \min \quad & q(\omega)^T y, \\ \text{s.t.} \quad & \sum_{i=1}^{n} x_i S_i(\omega) + \sum_{i=1}^{p} y_i T_i(\omega) \preceq C(\omega), \end{aligned} \tag{3}$$

and

$$E[Q(x,\omega)] = \int_{\Omega} Q(X,\omega) P(\mathrm{d}\omega) \tag{4}$$

is the expected value of $Q(x,\omega)$. The first-stage data $\mathcal{D}=(c,A_1,\dots,A_n,B)\in\Re^n\times(\Re_s^{m\times m})^{n+1}$, appearing in (2), are deterministic and have the same meaning as the data of SDP (1), while the second-stage data

$$\mathcal{R}(\omega)=\left(q(\omega),S_1(\omega),\dots,S_n(\omega),T_1(\omega),\dots,T_p(\omega),C(\omega)\right)\in\Re^p\times\left(\Re_s^{l\times l}\right)^{n+p+1}, \tag{5}$$

are stochastic and depend on a random event $\omega\in\Omega$, where (Ω,P) is an underlying probability space.

The first stage (2) of SSDP is obtained by adding to the objective function of SDP (1) the expected value of some costs that are incurred for each realization of the random event ω. The idea is that the first stage decision variables $x\in\Re^n$ are to be determined "here and now", while the second stage decision variables $y\in\Re^p$ are to be determined after a realization of the random event ω has occurred, and therefore the objective function and the constraints of the second stage optimization problem (3) are known. The second stage optimization problem (3) can be interpreted as a recourse to the first stage optimization problem (2). Of course, the latter problem cannot be solved without knowing the expected value $E[Q(\mathbf{x},\omega)]$. Therefore, SSDP (2)–(4) is usually solved by generating K scenarios, that is, K instances

$$\mathcal{R}_k=\left(q_k,S_1^k,\dots,S_n^k,T_1^k,\dots,T_p^k,C^k\right),\quad k=1,\dots,K, \tag{6}$$

of the data (5), and by solving a large-scale deterministic SDP of the form

$$\begin{aligned}\min\quad & c^Tx+\sum_{k=1}^K p_kq_k^Ty_k,\\ \text{s.t.}\quad & \sum_{i=1}^n x_iA_i\preceq B,\\ & \sum_{i=1}^n x_iS_i^k+\sum_{i=1}^p y_{ki}T_i^k\preceq C^k,\quad k=1,\dots,K.\end{aligned} \tag{7}$$

In the objective function of (7), p_k represents the probability of the scenario described by the data $\mathcal{R}_k$ (the scenario $\mathcal{R}_k$, or simply the k-th scenario). If we view the generated scenarios (together with the corresponding probabilities) as an approximation of the "true" probability distribution of the random data vector $\mathcal{R}(\omega)$, the natural question is whether by solving SDP (7) we can solve the "true" two-stage problem (2)–(4) with a reasonable accuracy (see [14] for some very recent results relative to this question). Another point of view is that the "true" probability distribution of the random data vector $\mathcal{R}(\omega)$ is not known, and that by generating scenarios $\mathcal{R}_1,\dots,\mathcal{R}_K$ and determining probabilities $p_1,\dots,p_K$, we "create" a finite event space that hopefully accurately models the randomness occurring in a given application. We note that (7) was first suggested in a slightly different notation in [13, (17)].

The deterministic second-order cone programming (DSOCP) problem consists in minimizing a linear function over the intersection of an affine set and a Cartesian product of second-order (Lorentz) cones,

$$\mathcal{K}_n := \left\{x = (x_0; \bar{x}) \in \Re^n : x_0 \in \Re,\ x_0 \geq \|\bar{x}\|\right\}, \tag{8}$$

where $\|\cdot\|$ refers to the standard Euclidean norm, and n is the dimension of $\mathcal{K}_n$ (see [15]).

Since

$$x_0 \geq \|\bar{x}\| \Leftrightarrow\ Arw(x) := \begin{pmatrix} x_0 & -\bar{x}^T \\ -\bar{x} & x_0 I \end{pmatrix} \succeq 0, \tag{9}$$

it follows that DSOCP is a particular case of DSDP. However, the computational effort per iteration required by an interior point methods to solve the DSOCP problem is substantially less than that required to solve the corresponding DSDP problem.

In the stochastic programming problems to be investigated in this paper, we will generate scenarios by randomly generating ellipsoids of the form

$$E = \{u \in \Re^n : u^T H u + 2g^T u + \nu \leq 0\}, \tag{10}$$

where $H \in \Re_s^{n\times n}$ is a given positive-definite matrix, $g \in \Re^n$ is a given vector, and $\nu \in \Re$ is a given real number such that

$$\nu < g^T H^{-1} g. \tag{11}$$

The significance of condition (11) becomes obvious by denoting

$$u^0 = -H^{-1}g, \qquad \rho = \sqrt{g^T H^{-1} g - \nu},$$

and noticing that

$$u^T H u + 2g^T u + \nu = (u - u^0)^T H (u - u^0) - \rho^2 = \left\| H^{1/2}(u - u^0)\right\|^2 - \rho^2.$$

It follows that for $H = I$, the ellipsoid (10) reduces to the disk with center $u^0 \in \Re^n$ and radius ρ. It is obvious that for any positive real number t the triplet $(tH, tg, t\nu)$ defines the same ellipsoid as the triplet (H, g, ν). Therefore, we can assume that $\rho = 1$. In this case, the ellipsoid (10) can be written as

$$E = \left\{u \in \Re^n : \left\|H^{1/2}(u - u^0)\right\| \leq 1\right\}. \tag{12}$$

If $H = Q\Lambda Q^T$ is the spectral decomposition of H, where $Q = (q^1 q^2 \ldots q^n)$ is the matrix whose columns are the eigenvectors of H and $\Lambda = \mathrm{Diag}(\lambda_1; \ldots; \lambda_n)$ is the diagonal matrix of the corresponding eigenvalues, then a vector $u \in \Re^n$ belongs to the ellipsoid (12) if and only if

$$\sum_{i=1}^{n} \lambda_i ((u - u^0)^T q^i)^2 \leq 1.$$

If $n=2$, then Q is given by a rotation matrix of the form

$$Q=\begin{pmatrix}\cos\varphi & -\sin\varphi\\ \sin\varphi & \cos\varphi\end{pmatrix}. \tag{13}$$

The column vectors of Q give the directions of the two axes of the ellipse, φ being the angle between the first axis of the ellipse and the u_1-axis of the coordinate system. The quantities $\sigma_1=\lambda_1^{-1/2}$ and $\sigma_2=\lambda_2^{-1/2}$ represent the lengths of the corresponding semi-axes. Therefore a two dimensional ellipsoid (i.e., and ellipse) is completely determined by the angle φ, the center $u^0=(u_1^0,u_2^0)$, and the lengths of the two semiaxes σ_1,σ_2. Since these parameters have a clear geometrical meaning, they will be used in our numerical experiments to generate different scenarios. For given φ, $u^0=(u_1^0,u_2^0)$, σ_1,σ_2, we can represent the ellipsoid in the form (10) by setting

$$H=Q\begin{pmatrix}\sigma_1^{-2} & 0\\ 0 & \sigma_2^{-2}\end{pmatrix}Q^T,\qquad g=-Hu^0,\qquad \nu=u^{0T}Hu^0-1. \tag{14}$$

In the present paper, we have chosen to work with the representation (10) of an ellipsoid, because in this case the condition that the ellipsoid given by the triplet (H,g,ν) is included in the ellipsoid given by the triplet $(\hat{H},\hat{g},\hat{\nu})$ can be conveniently written as

$$\exists\tau\in\Re^+\cup\{0\}\quad \text{s.t.}\quad \begin{pmatrix}H & -g\\ -g^T & \nu\end{pmatrix}\preceq\tau\begin{pmatrix}\hat{H} & -\hat{g}\\ -\hat{g}^T & \hat{\nu}\end{pmatrix}. \tag{15}$$

It is important to note that if (15) holds, and $\hat{\nu}<\hat{g}^T\hat{H}^{-1}\hat{g}$, then condition (11) is automatically satisfied.

3 Stochastic Semidefinite Program for Modeling Networks with Moving Nodes

In this section we describe the semidefinite stochastic location-aided routing (SLAR) model considered in Ariyawansa and Zhu [3], and Zhu et al. [4]. In this model, a sender node S needs to find a route to a destination node D, through broadcasting a route request to its neighbors. Once D receives the route request (and this should happen within a time-out interval t_1, otherwise the route request has to be reinitiated), D responds by reversing the path followed by the route request received by D. In this model, while node D is supposed to move at a random speed, S is supposed to be static. The communication is considered successful when the reply message is sent back to the source node. We assume that:

- The nodes in the network are uniformly distributed.
- The source node S knows the location $l\in\Re^n$ of the destination node D at time t_0, and the node D knows the location, that for simplicity we suppose to be the origin $0\in\Re^n$, of node S at time t_0.
- The node D moves at a random speed $v(\omega_1)\in\Re^+$ along a (normalized) random direction $d(\omega_1)\in\Re^n$, which depends on an underlying outcome ω_1 in an event space Ω_1.

In the SLAR model, the source node S first determines a disk

$$C=\{u\in\Re^n : u^T u-2\tilde{u}^T u+\gamma\le 0\},\tag{16}$$

with $\gamma\in\Re$, center $\tilde{u}\in\Re^n$ and radius $\rho=\sqrt{\tilde{u}^T\tilde{u}-\gamma}$, where the destination node D is expected to be at time t_1. Therefore, the disk C, called the expected zone, is required to contain the disk C_0 centered in $l\in\Re^n$ with radius $v(t_1-t_0)$, where $v\in\Re^+$ is the minimum speed at which the node D is supposed to move. According to (15), this requirement is equivalent to

$$0\le\tau,\quad \begin{pmatrix} I & -\tilde{u}\\ -\tilde{u}^T & \gamma\end{pmatrix}\preceq\tau\begin{pmatrix} I & -l\\ -l^T & \|l\|^2-(t_1-t_0)^2v^2\end{pmatrix}.\tag{17}$$

After determining the expected region, the source node sends a route request to the center of C. Then message flooding is used inside of C to search for the destination node D. The cost of broadcasting is proportional to the distance, while the cost of flooding is proportional to the area of the flooded region, since we assume that the nodes in the network are uniformly distributed. Therefore, the cost of choosing the expected region (16) is given by

$$\alpha\|\tilde{u}\|+\beta\left(\tilde{u}^T\tilde{u}-\gamma\right),\tag{18}$$

where $\alpha\in\Re$ is the cost of broadcasting per unit of distance, and $\beta\in\Re$ is the cost of flooding per unit of area multiplied by π. We note that minimizing the nonlinear cost function (18) under the constraints (17), is equivalent to minimizing the linear cost function $\alpha d_1+\beta d_2$, under the constraints (17) and $d_1\ge\|\tilde{u}\|,\ d_2\ge\tilde{u}^T\tilde{u}-\gamma$. Finally, by noticing that the latter constraints can be written as

$$\begin{pmatrix} d_1 I & \tilde{u}\\ \tilde{u}^T & d_1\end{pmatrix}\succeq 0,\qquad \begin{pmatrix} I & \tilde{u}\\ \tilde{u}^T & d_2+\gamma\end{pmatrix}\succeq 0,\tag{19}$$

we deduce that minimizing (18) under the constraints (17) can be written as an SDP of the form (1) with

$$c=[\alpha,\beta,0,0,0]^T,\qquad x=[d_1,d_2,\tilde{u},\gamma,\tau]^T\in\Re^{n+4}.\tag{20}$$

By solving this deterministic SDP we obtain a minimum cost expected region C that contains the disk C_0, but given the random nature of the movement of node D, we cannot guarantee that at time t_1 this expected region will contain the destination node. The randomness of the movement of node D is modeled by assuming that D belongs to a random ellipsoid $E(\omega)$. For example, this allows the node to travel on a highway (given by the main axis of the ellipse), with the possibility of exiting the highway for relatively short distances.

Using this model of randomness, we can formulate the problem of determining the expected region as a two stage programming problem.

In Stage 1, we determine a disk C containing the disk C_0, as described above.

In Stage 2, after the realization of the random ellipsoid

$$E(\omega)=\{u\in\Re^n : u^T H(\omega)u+2g(\omega)^T u+v(\omega)\le 0\},\tag{21}$$

we enlarge the radius of the disk C, by a quantity $\zeta \in \Re^+ \cup \{0\}$, to obtain a disk

$$C^* = \{u \in \Re^n : u^T u - 2\tilde{u}^T u + \gamma - \zeta \leq 0\}, \tag{22}$$

with center $\tilde{u}$ and radius $\sqrt{\tilde{u}^T \tilde{u} - \gamma + \zeta}$, which contains $E(\omega)$. By introducing a new variable $\delta \in \Re^+ \cup \{0\}$, the Stage 2 constraints can be written as

$$\zeta \geq 0, \qquad \delta \geq 0, \qquad \begin{pmatrix} I & -\tilde{u} \\ -\tilde{u}^T & \gamma - \zeta \end{pmatrix} \preceq \delta \begin{pmatrix} H(\omega) & g(\omega) \\ g(\omega)^T & v(\omega) \end{pmatrix}. \tag{23}$$

We assume that at the beginning of Stage 2 the expected region C is already flooded, and that the cost of flooding per unit of area in Stage 2 is the same as in Stage 1. Therefore, the cost of flooding the expected zone C^* determined in Stage 2 is $\beta\zeta$. Therefore, the cost vector and the decision vector associated with Stage 2 are

$$q = [\beta, 0]^T, \qquad y = [\zeta, \delta]^T. \tag{24}$$

Using the above notation, our two stage stochastic programming problem can be written as

$$\begin{aligned} \min \quad & c^T x + E[Q(x, \omega)], \\ \text{s.t.} \quad & (17) \text{ and } (19), \end{aligned} \tag{25}$$

where $Q(x, \omega)$ is the minimum of the problem

$$\begin{aligned} \min \quad & q^T y, \\ \text{s.t.} \quad & (23), \end{aligned} \tag{26}$$

and

$$E[Q(x, \omega)] = \int_\Omega Q(X, \omega) P(\mathrm{d}\omega), \tag{27}$$

is the expected value of $Q(x, \omega)$. By using proper notation, (25)–(27) can be written under the form (2)–(4) (see [3, 4]). We note that the SLAR model above is almost identical with the SLAR model [3, (20)–(22)], with the exception of the fact that there the probability space Ω is considered to be finite, $\Omega = \{1, \ldots, K\}$. As we mentioned in the previous section, (2)–(4) is usually solved by generating K scenarios $\mathcal{R}_1, \ldots, \mathcal{R}_K$, determining their probabilities $p_1, \ldots, p_K$, and computing the solution of the large-scale SDP (7), so that we again get a finite probability space. In our case, this means generating K ellipsoids

$$E_k = \{u \in \Re^n : u^T H_k u + 2{g_k}^T u + \nu_k \leq 0\}, \quad k = 1, \ldots, K, \tag{28}$$

determining their probabilities $p_1, \ldots, p_K$, and solving the large-scale SDP

$$\min \quad \alpha d_1 + \beta d_2 + \sum_{k=1}^{K} p_k \beta \zeta_k,$$

$$\text{s.t.}\quad \begin{pmatrix} I & -\tilde{u} \\ -\tilde{u}^T & \gamma \end{pmatrix} \preceq \tau \begin{pmatrix} I & -l \\ -l^T & \|l\|^2 - (t_1 - t_0)^2 v^2 \end{pmatrix}, \quad \tau \geq 0,$$
$$\begin{pmatrix} d_1 I & \tilde{u} \\ \tilde{u}^T & d_1 \end{pmatrix} \succeq 0, \qquad \begin{pmatrix} I & \tilde{u} \\ \tilde{u}^T & d_2 + \gamma \end{pmatrix} \succeq 0,$$
$$\zeta_k \geq 0, \qquad \delta_k \geq 0, \qquad \begin{pmatrix} I & -\tilde{u} \\ -\tilde{u}^T & \gamma - \zeta_k \end{pmatrix} \preceq \delta_k \begin{pmatrix} H_k & g_k \\ g_k{}^T & \nu_k \end{pmatrix}, \quad k = 1, \ldots, K. \tag{29}$$

By solving the above SDP, we can define the "theoretical" expected zone as being the disk

$$C = \{u \in \Re^n : \|u - \tilde{u}\| \leq \rho\}, \quad \rho = \sqrt{\tilde{u}^T \tilde{u} - \gamma}, \tag{30}$$

which is completely described in terms of the first-stage variables $\tilde{u}$ and γ. The cost of flooding this disk is

$$\alpha \|\tilde{u}\| + \beta \rho^{\,2} = \alpha d_1 + \beta d_2. \tag{31}$$

If the stage 2 recourse action described in the first half of this section would be implementable in practice, then choosing C as the requested zone would lead to an expected total cost $\alpha \|\tilde{u}\| + \beta \tilde{\rho}^{\,2} = \alpha d_1 + \beta d_2 + \beta \tilde{\zeta}$, where

$$\tilde{\rho} = \sqrt{\tilde{u}^T \tilde{u} - \tilde{\gamma}}, \quad \tilde{\gamma} = \sum_{k=1}^{K} p_k \tilde{\gamma}_k = \gamma - \tilde{\zeta}, \; \tilde{\gamma}_k = \gamma - \zeta_k, \; \tilde{\zeta} = \sum_{k=1}^{K} p_k \zeta_k. \tag{32}$$

But this is exactly the quantity minimized by the SDP (29). Therefore, if the theoretical stage 2 recourse action considered in this section would be implementable in practice, the choice of C would be optimal. However, after flooding C, the only information obviously available to S is whether or not the node D had been reached by time t_1. In the affirmative case, a route between S and D is established. Otherwise, a recourse action is needed in order to establish communication. Theoretically D belongs to one of the ellipsoids E_k, but in general the information characterizing E_k is not available to S, so that S cannot take the theoretical recourse action of flooding the smallest disk centered at $\tilde{u}$ that contains E_k. The usual recourse action that is implemented in practice is to flood a larger requested zone that contains C. In the extreme, this larger requested zone may be the entire local network.

In view of the above comments it may be advantageous to also consider expected zones that are larger than the disk C given by (30). We propose, as a natural candidate for such an expected zone, the disk

$$\widetilde{C} = \{u \in \Re^n : \|u - \tilde{u}\| \leq \tilde{\rho}\}, \tag{33}$$

where the radius $\tilde{\rho}$ is defined by (32). The cost of flooding $\widetilde{C}$ is

$$\widetilde{\eta} = \alpha \|\tilde{u}\| + \beta \tilde{\rho}^{\,2} = \alpha d_1 + \beta d_2 + \beta \tilde{\zeta}, \tag{34}$$

which coincides with the minimum value of the objective function in (29). The expected region proposed in (33) is intimately related to the large-scale SDP (29), since

$\widetilde{C}$ is defined in terms of the optimal values of both first stage and second stage variables arising in (29), and since the cost of flooding $\widetilde{C}$ is equal the optimal objective value of SDP (29).

We can define a larger expected zone of the form

$$\widehat{C} = \{u \in \Re^n : \|u - \bar{u}\| \leq \widehat{\rho}\}, \quad \widehat{\rho} = \sqrt{\bar{u}^T \bar{u} - \widehat{\gamma}}, \;\; \widehat{\gamma} = \bar{\gamma} - \zeta, \tag{35}$$

where $\bar{u}$, $\bar{\gamma}$, ζ, are obtained by solving the large-scale SDP

$$\begin{aligned}
\min \quad & \alpha \bar{d}_1 + \beta \bar{d}_2 + \beta \zeta, \\
\text{s.t.} \quad & \begin{pmatrix} I & -\bar{u} \\ -\bar{u}^T & \bar{\gamma} \end{pmatrix} \preceq \bar{\tau} \begin{pmatrix} I & -l \\ -l^T & \|l\|^2 - (t_1 - t_0)^2 v^2 \end{pmatrix}, \quad \bar{\tau} \geq 0, \\
& \begin{pmatrix} \bar{d}_1 I & \bar{u} \\ \bar{u}^T & \bar{d}_1 \end{pmatrix} \succeq 0, \qquad \begin{pmatrix} I & \bar{u} \\ \bar{u}^T & \bar{d}_2 + \bar{\gamma} \end{pmatrix} \succeq 0, \qquad \zeta \geq 0, \\
& \bar{\delta}_k \geq 0, \qquad \begin{pmatrix} I & -\bar{u} \\ -\bar{u}^T & \bar{\gamma} - \zeta \end{pmatrix} \preceq \bar{\delta}_k \begin{pmatrix} H_k & g_k \\ g_k{}^T & \nu_k \end{pmatrix}, \quad k = 1, \ldots, K.
\end{aligned} \tag{36}$$

Geometrically, $\widehat{C}$ represents the smallest disk containing the disk C_0 and all the ellipsoids $E_1, \ldots, E_K$. It is easily seen that the minimum cost function of (29) is always less than or equal to the minimum cost function of (36),

$$\widehat{\eta} = \alpha \|\tilde{u}\| + \beta \widehat{\rho}^{\,2} = \alpha \bar{d}_1 + \beta \bar{d}_2 + \beta \widehat{\zeta}, \tag{37}$$

which represents the cost of flooding $\widehat{C}$. Moreover, (36) cannot be obtained from a two stage stochastic programming problem in the way (7) was obtained from (2)-(4). Nevertheless, $\widetilde{C}$ may be useful in determining the requested zone, which, we believe, should contain C, and be contained in $\widehat{C}$. The disk $\widetilde{C}$ satisfies these requirements, and, given its special relationship with the SDP model (29), would be an excellent candidate for the requested zone. The practical efficiency of choosing this requested zone could be tested by computer simulation, but this goes beyond the scope of the present paper which, as mentioned in the introduction, concerns itself only with the problem of determining the expected zone.

Although, as mentioned above, the SDP (36) has not been obtained from a two stage stochastic programming problem, we can still consider the "first stage" expected zone given by the disk

$$\bar{C} = \{u \in \Re^n : \|u - \bar{u}\| \leq \bar{\rho}\}, \quad \bar{\rho} = \sqrt{\bar{u}^T \bar{u} - \bar{\gamma}}, \tag{38}$$

where $\bar{u}$ and $\bar{\gamma}$ are obtained from by solving (36). The cost of flooding the disc $\bar{C}$ is given by

$$\bar{\eta} = \alpha \|\bar{u}\| + \beta \bar{\rho}^{\,2} = \alpha \bar{d}_1 + \beta \bar{d}_2. \tag{39}$$

We note that the model considered in [4] is slightly different. There, the disk C computed in Stage 1 is no longer required to contain the disk C_0. Also, in Stage 2, it is assumed that the destination node D belongs to one of the ellipsoids E_k with probability p_k, and that the ellipsoids are ordered such that $p_1 \geq p_2 \geq \cdots \geq p_K$. First, the route request is sent out to look for D by flooding the expected zone C. If the destination node D is in C, the route request reaches the destination and the reply message is sent back to the source, so that a route is established between the source and the destination nodes. Otherwise, a search request is sent to E_1, since it occurs with the highest probability, by flooding the smallest disk C_1^*, centered at $\tilde{u}$ that contains E_1. If D is in E_1, then a route is established between the source and the destination nodes. Otherwise, a search request is sent to E_2, by flooding the smallest disk C_2^*, centered at $\tilde{u}$ that contains E_2, etc. The implementation of this approach, which is greedy because of the assumption that $p_1 \geq p_2 \geq \cdots \geq p_K$, needs the solution of a large scale SDP similar to (29). Although in the present paper we will only study SLAR (29), our results can also be applied to the SLAR model considered in [4].

4 Stochastic Second-Order Cone Model for SLAR

The computational effort required by an interior-point method to solve a SOCP problem is much less than that required to solve the corresponding SDP problem. The aim of this section is to formulate the semidefinite stochastic location-aided routing (SLAR) problem presented in the previous section as a stochastic second-order cone SSOCP problem. We rewrite each semidefinite constraint as a second order cone constraint. Let us look first at a semidefinite constraint of the form

$$\begin{pmatrix} Z & w \\ w^T & \varsigma \end{pmatrix} \succeq 0, \tag{40}$$

where $Z = \mathrm{Diag}(z)$, $z, w \in \Re^n$, $\varsigma \in \Re$. We consider the index set $\mathcal{I}(z) = \{i \;:\; z_i > 0\}$. By taking the Schur complement of the submatrix of Z corresponding to this index set, it follows that (40) is equivalent to

$$z \geq 0, \quad \forall i \notin \mathcal{I}(z)\; w_i = 0, \quad \varsigma \geq \sum_{i \in \mathcal{I}(z)} \frac{w_i^2}{z_i}. \tag{41}$$

Following Alizadeh and Goldfarb [15, p. 14], we deduce that (41) holds if and only if there is a vector $f \in \Re^n$ such that

$$z \geq 0, \qquad f \geq 0, \qquad \varsigma \geq 1^T f, \qquad w_i^2 \leq \varsigma f_i z_i, \quad i = 1, \ldots, n. \tag{42}$$

Indeed, if (41) is satisfied, then (42) holds by taking $f_i = w_i^2/z_i$ for $i \in \mathcal{I}(z)$, and $f_i = 0$ for $i \notin \mathcal{I}(z)$. Conversely, it is obvious that if (42) is satisfied then (41) is also satisfied. We note that Alizadeh and Goldfarb did not explicitly require that $f \geq 0$, although it appears that this condition must be imposed. If $i \in \mathcal{I}(z)$, then $w_i^2 \leq \varsigma f_i z_i$ implies that $f_i \geq 0$, but if $i \notin \mathcal{I}(z)$ then f_i could be negative and in this case (42), without the condition $f \geq 0$, may not imply (41). However, if an interior point method is used for solving the resulting optimization problem, then at each iteration of the algorithm the matrix in (40) is positive definite so that $\mathcal{I}(z) = \{1, \ldots, n\}$, and in this case the constraint $f \geq 0$ may be omitted, which reduces the computational complexity.

Using (42), it follows that the second constraint from (17), which represents the condition of inclusion of the disk C_0 in the first stage disk C, holds if and only if there is a vector $r \in \Re^n$ such that

$$\tau \geq 1, \qquad r \geq 0, \qquad \gamma \leq \tau \|l\|^2 - \tau (t_1 - t_0)^2 v^2 - 1^T r, \tag{43}$$

$$\left(\tau l_j - \tilde{u}_j\right)^2 \leq r_j (\tau - 1), \quad j = 1, \ldots, n. \tag{44}$$

Notice that the restricted hyperbolic constraints (44) are equivalent to the following 3-dimensional second-order cone inequalities

$$\left\| \begin{pmatrix} 2(\tau l_j - \tilde{u}_j) \\ r_j - \tau + 1 \end{pmatrix} \right\| \leq r_j + \tau - 1 \Leftrightarrow \begin{pmatrix} r_j + \tau - 1 \\ 2(\tau l_j - \tilde{u}_j) \\ r_j - \tau + 1 \end{pmatrix} \in \mathcal{K}_3, \quad j = 1, \ldots, n, \tag{45}$$

and the linear constraints (43) are 1-dimensional second-order cone constraints.

On the other hand,

$$0 \preceq \begin{pmatrix} d_1 I & -\tilde{u} \\ -\tilde{u}^T & d_1 \end{pmatrix} \Leftrightarrow d_1 \geq \sqrt{\tilde{u}^T \tilde{u}} \Leftrightarrow \begin{pmatrix} d_1 \\ \tilde{u} \end{pmatrix} \in \mathcal{K}_{n+1}, \tag{46}$$

$$0 \preceq \begin{pmatrix} I & -\tilde{u} \\ -\tilde{u}^T & d_2 + \gamma \end{pmatrix} \Leftrightarrow d_2 + \gamma \geq \tilde{u}^T \tilde{u} \Leftrightarrow \begin{pmatrix} \sqrt{d_2 + \gamma} \\ \tilde{u} \end{pmatrix} \in \mathcal{K}_{n+1}. \tag{47}$$

We observe that the last of the three equivalent conditions above can be treated as a rotated quadratic cone (or hyperbolic constraint)

$$\widehat{\mathcal{K}}_{n+2} = \left\{ u \in \Re^{n+2} : 2u_1 u_2 \geq \sum_{j=3}^{n+2} u_j^2, u_1, u_2 \geq 0 \right\}, \tag{48}$$

with $u_2 = d_2 + \gamma$, $u_j = x_{j-2}$, $j = 3, \ldots, n+2$, intersected with the hyperplane $u_1 = 1/2$.

The second stage constraint

$$\begin{pmatrix} I & -\tilde{u} \\ -\tilde{u}^T & \gamma - \zeta_k \end{pmatrix} \preceq \delta_k \begin{pmatrix} H_k & g_k \\ {g_k}^T & \nu_k \end{pmatrix}, \tag{49}$$

which represents the condition of inclusion of the ellipsoid E_k into the disk C_k^*, is equivalent to

$$M_k := \begin{pmatrix} \delta_k H_k - I & \delta_k g_k + \tilde{u} \\ \delta_k g_k^T + \tilde{u}^T & \delta_k \nu_k - \gamma + \zeta_k \end{pmatrix} \succeq 0. \tag{50}$$

Following again Alizadeh and Goldfarb [15, p. 14], we use the spectral decomposition of H_k, $H_k = Q_k \Lambda_k Q_k^T$, and by denoting $h_k = Q_k^T (\delta_k g_k + \tilde{u})$, we obtain

$$\bar{M}_k := \begin{pmatrix} Q_k^T & 0 \\ 0 & 1 \end{pmatrix} M_k \begin{pmatrix} Q_k & 0 \\ 0 & 1 \end{pmatrix} = \begin{pmatrix} \delta_k \Lambda_k - I & h_k \\ h_k^T & \delta_k \nu_k - \gamma + \zeta_k \end{pmatrix}. \tag{51}$$

Clearly $M_k \succeq 0$ if and only if $\bar{M}_k \succeq 0$. Denoting $\Lambda_k = \mathrm{Diag}(\lambda_{k1}; \ldots; \lambda_{kn})$, and using (42), we deduce that (49) holds if and only if there is a vector $s_k = (s_{k1}; \ldots; s_{kn})$ such that

$$s_k \geq 0, \qquad \delta_k \geq \frac{1}{\lambda_{min}(\Lambda_k)}, \qquad \zeta_k \geq \gamma + 1^T s_k - \delta_k \nu_k, \tag{52}$$

$$h_{kj}^2 \leq s_{kj}\left(\delta_k \lambda_{kj} - 1\right), \quad j = 1, \ldots, n. \tag{53}$$

We mention that due to the positive definiteness of matrix H_k, (52) implies that the constraint $\delta_k \geq 0$ mentioned in the SLAR model (29) is automatically satisfied. The restricted hyperbolic constraint (53) is equivalent to the following 3-dimensional second-order cone inequalities

$$\begin{pmatrix} s_{kj} + \delta_k \lambda_{kj} - 1 \\ 2h_{kj} \\ s_{kj} - \delta_k \lambda_{kj} + 1 \end{pmatrix} \in \mathcal{K}_3, \quad j = 1, \ldots, n, \tag{54}$$

while the linear constraints (52) are 1-dimensional second-order cone constraints. In conclusion, the SLAR model (29) can be formulated as a stochastic second-order cone SSOCP problem with two $(n+1)$-dimensional second-order cone constraints (see (46), (47)), $n(K+1)$ 3-dimensional second-order cone constraints (see (45) and (54)) and with all the other constraints linear, in the following way:

$$\begin{aligned}
\min \quad & \alpha d_1 + \beta d_2 + \sum_{k=1}^{K} p_k \beta \zeta_k, \\
\text{s.t.} \quad & r \geq 0, \qquad \tau \geq 1, \qquad \gamma \leq \tau \,\|l\|^2 - \tau\,(t_1 - t_0)^2\, v^2 - 1^T r, \\
& \begin{pmatrix} r_j + \tau - 1 \\ 2\left(\tau l_j - \tilde{u}_j\right) \\ r_j - \tau + 1 \end{pmatrix} \in \mathcal{K}_3, \quad j = 1, \ldots, n, \\
& \begin{pmatrix} d_1 \\ \tilde{u} \end{pmatrix} \in \mathcal{K}_{n+1}, \qquad \begin{pmatrix} \sqrt{d_2 + \gamma} \\ \tilde{u} \end{pmatrix} \in \mathcal{K}_{n+1}, \\
& h_k = Q_k^T\left(\delta_k g_k + \tilde{u}\right), \qquad \zeta_k \geq 0, \quad k = 1, \ldots, K, \\
& s_k \geq 0, \qquad \delta_k \geq \frac{1}{\lambda_{min}(\Lambda_k)}, \qquad \zeta_k \geq \gamma + 1^T s_k - \delta_k \nu_k, \quad k = 1, \ldots, K, \\
& \begin{pmatrix} s_{kj} + \delta_k \lambda_{kj} - 1 \\ 2h_{kj} \\ s_{kj} - \delta_k \lambda_{kj} + 1 \end{pmatrix} \in \mathcal{K}_3, \quad j = 1, \ldots, n, \ k = 1, \ldots, K.
\end{aligned} \tag{55}$$

The decision variables in SSOCP (55) are

$$d_1, d_2, \tilde{u}, \gamma, \tau, r, \zeta_1, \ldots, \zeta_K, \delta_1, \ldots, \delta_K, s_1, \ldots, s_K, h_1, \ldots, h_K, \tag{56}$$

all the other quantities from SSOCP (55) being considered constant. As mentioned before, if an interior point method is used for solving (55), then the nonnegativity constraints on the variables $r, s_1, \ldots, s_K$, can be omitted.

5 Scenario Generation and Numerical Results

In our numerical experiments we have generated the ellipsoids (28) by using the observation made in Sect. 2, that in the two-dimensional case ($n=2$) an ellipsoid (now called an ellipse) is completely defined by its center $u^0=(u_1^0,u_2^0)$, the angle φ between the first axis of the ellipse and the u_1-axis of the coordinate system, and the lengths σ_1,σ_2 of the two semi-axes. In our numerical experiments we have randomly generated these quantities, and then we have obtained ellipses of the form (10) by using (14). We have obtained the quantities

$$u_1^{0,k}, u_2^{0,k}, \varphi_k, \sigma_1^k, \sigma_2^k, \quad k=1,2,\ldots,K, \tag{57}$$

corresponding to the ellipses (28), by using

- the uniform distribution in the interval $(\sqrt{8}-1,\sqrt{8}+1)$ for generating $u_1^{0,k}$;
- the normal distribution $\mathcal{N}(0,0.5)$ for generating $u_2^{0,k}$;
- the uniform distribution in the interval $[0,\frac{\pi}{2}]$ for generating φ^k;
- the normal distribution $\mathcal{N}(2,1)$ for generating $s_{u_1}^k$;
- the normal distribution $\mathcal{N}(1,0.5)$ for generating $s_{u_2}^k$.

In order to have realistic scenarios, we have imposed the following upper and lower bounds:

$$\begin{aligned}
&u_2^{0,\min} \le u_2^{0,k} \le u_2^{0,\max}, && \forall k=1,\ldots,K,\\
&\sigma_1^{\min} < \sigma_1^k \le \sigma_1^{\max}, && \forall k=1,\ldots,K,\\
&\sigma_2^{\min} < \sigma_2^k \le \sigma_2^{\max}, && \forall k=1,\ldots,K.
\end{aligned}$$

In the remainder of this section, we present some numerical results obtained for the semidefinite stochastic location-aided routing (SLAR) problem presented in Sect. 3, and for its SSOCP model presented in Sect. 4, in order to compare the performance of these models. We have randomly generated scenarios (in MATLAB 7.4.0), according to the method described above, where we have set

$$u_2^{0,\min}=-1, \qquad u_2^{0,\max}=1, \qquad \sigma_1^{\min}=\sigma_2^{\min}=0.1, \qquad \sigma_1^{\max}=\sigma_2^{\max}=3.$$

SSOCP (55) was implemented in GAMS 22.5, by using the second order cone programming solver from the software package Mosek (http://www.mosek.com/), while SLAR (29) was implemented in MATLAB 7.4.0 by using the dual semidefinite programming solver DSDP [16].

In our computational experiments we have supposed that the scenarios are equiprobable; furthermore we have fixed the location $l=(2,0)$ of the node D at initial time $t_0=0$, its lowest speed $v=1$, and the final time $t_1=1$, so that the disk C_0 has center $l=(2,0)$ and unitary radius,

$$C_0=\{u=(u_1,u_2)\in\Re^2 : u_1^2+u_2^2-4u_1+3\le 0\}.$$

The first and second stage costs were obtained by taking $\alpha=0.1$, $\beta=0.5$. This assumption implies that the cost of choosing the radius of the first stage circle C is

Table 1 Center, angle, and semiaxes of five randomly generated ellipsoids

k	u_1^0	u_2^0	φ	σ_1	σ_2
1	2.1332	−0.7902	1.2972	1.9214	0.6592
2	2.9051	−0.5123	1.5647	0.7656	1.1444
3	1.9848	−0.2146	0.6954	2.0558	0.8161
4	2.0417	−0.2325	1.5109	2.3710	1.3641
5	3.4630	0.5189	1.3645	1.6104	0.3094

Table 2 Expected regions and costs for model (36) in the case of 5 and 500 scenarios

K	$\tilde{u}_1$	$\tilde{u}_2$	$\bar{d}_1$	$\bar{d}_2$	$\bar{\gamma}$	$\bar{\tau}$	$\bar{\rho}$	$\bar{\eta}$	$\widehat{\rho}$	$\widehat{\eta}$
5	2.61	−0.23	2.62	5.87	1.01	3.01	2.42	3.20	2.61	3.67
500	2.74	0.17	2.75	12.40	−4.86	4.73	3.52	6.47	3.93	8

five times more expensive than the cost of choosing its center $\tilde{u}$. Because of the low value of α, we expect that the center $\tilde{u}$ of the circle will not be chosen close to the origin 0. Moreover, we suppose that at each scenario, the cost of changing the radius of the second stage circle, is the same as the cost of choosing the radius of the first stage circle. This means that the cost of flooding per unit of area is the same in the first and the second stage. For a detailed sensitivity analysis approach according to different values of the costs see Zhu et al. [4].

The purpose of the first test is to compare, on the same set of ellipsoid scenarios, the expected regions C, $\widetilde{C}$, $\bar{C}$, $\widehat{C}$, given by (30), (33), (38), (35), and the costs of flooding these regions. Note that cost of flooding $\widetilde{C}$, given by (34), coincides with the objective function of SDP (29), while the cost of flooding $\widehat{C}$, given by (37), coincides with the objective function of SDP (36). In order to elucidate the relationship between these expected regions, we treat in detail the case involving five different scenarios.

Table 1 describes five ellipsoids E_k, $k = 1, \dots, 5$, randomly generated as described above.

The first row of Table 2, and Fig. 1, refer to the solutions obtained using the model (36), given the five scenarios presented in Table 1. Note that the disk $\bar{C}$ coincides with the disk $\widehat{C}$, and that $\widehat{\eta}$ coincides with the optimal value of the objective function in (36).

We have then solved model (29) on the same set of five scenarios. The coordinates of the center $\tilde{u}_1$, $\tilde{u}_2$ and the radii ρ and $\tilde{\rho}$ of the resulting disks C and $\widetilde{C}$, as well as the costs η and $\tilde{\eta}$ of flooding those disks are reported in the first line of Table 3. We mention that the second-stage disks $C_1, \dots, C_5$, have the same center and their radii are given by

$$\rho_1 = 2.48, \qquad \rho_2 = 1.80, \qquad \rho_3 = 2.27, \qquad \rho_4 = 2.39, \qquad \rho_5 = 2.79.$$

As expected, each second-stage disk C_k, contains the disks C_0, C and the ellipse E_k for $k = 1, \dots, 5$. The reported results show that using model (29) allows for a saving of about 17% of the total cost, when compared with model (36). The saving

Fig. 1 Expected region given by model (36) in case of five scenarios

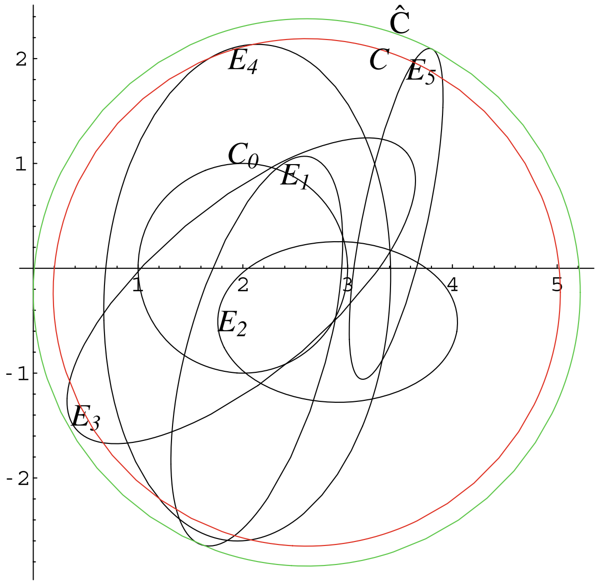

Table 3 Expected regions and costs for model (29) for K scenarios

K	$\tilde{u}_1$	$\tilde{u}_2$	d_1	d_2	γ	τ	ρ	η	$\tilde{\rho}$	$\tilde{\eta}$
5	2.28	−0.25	2.29	2.75	2.49	1.78	1.66	1.60	2.35	3.04
10	2.73	0.16	2.74	3.06	4.43	1.75	1.75	1.80	2.20	2.77
50	2.82	0	2.82	3.30	4.63	1.82	1.82	1.93	2.48	3.48
100	2.71	0.06	2.71	2.93	4.41	1.71	1.71	1.73	2.57	3.74
200	2.58	0.03	2.58	2.51	4.16	1.58	1.58	1.51	2.43	3.40
300	2.74	0.03	2.74	3.03	4.48	1.74	1.74	1.79	2.47	3.46
400	2.70	−0.03	2.70	2.91	4.41	1.71	1.71	1.72	2.47	3.49
500	2.67	0.01	2.67	2.78	4.34	1.67	1.67	1.66	2.46	3.47
1000	2.69	0.04	2.69	2.86	4.38	1.69	1.69	1.70	2.43	3.38
1480	2.68	0.02	2.68	2.82	4.36	1.68	1.68	1.68	2.44	3.41
2015	2.69	0.02	2.69	2.85	4.38	1.69	1.69	1.70	2.45	3.42
3010	2.67	0.03	2.67	2.78	4.33	1.67	1.67	1.66	2.44	3.40
4160	2.68	0.02	2.68	2.82	4.36	1.68	1.68	1.68	2.45	3.44
5240	2.70	0.03	2.70	2.88	4.39	1.70	1.70	1.71	2.45	3.43
6015	2.68	0.02	2.68	2.82	4.36	1.68	1.68	1.68	2.45	3.45
7440	2.68	0.04	2.68	2.83	4.36	1.68	1.68	1.68	2.45	3.44
8015	2.68	0.02	2.68	2.82	4.36	1.68	1.68	1.68	2.45	3.43
9051	2.67	0.03	2.67	2.80	4.35	1.68	1.67	1.67	2.45	3.43
10040	2.68	0.03	2.68	2.81	4.35	1.68	1.68	1.67	2.46	3.45
11268	2.67	0.02	2.67	2.80	4.34	1.67	1.67	1.67	2.45	3.43
20250	2.68	0.02	2.68	2.83	4.36	1.68	1.68	1.68	2.45	3.42

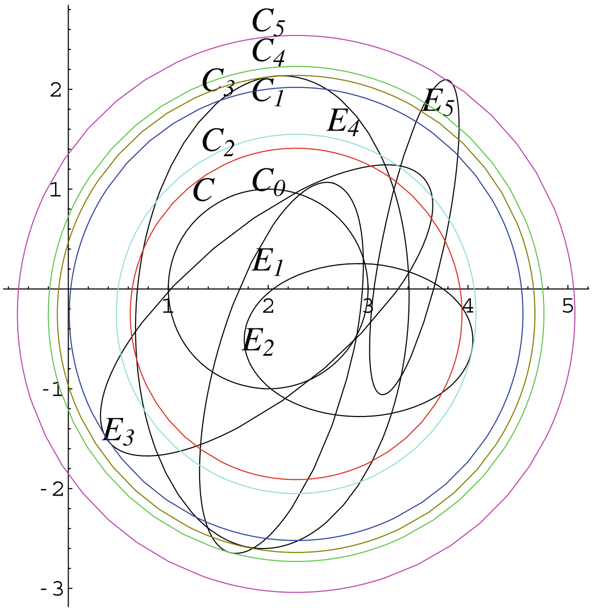

Fig. 2 Solution of our model in the case of five scenarios $k = 1, \ldots, 5$

drastically increases by considering a higher number of scenarios, as evidenced by comparing the costs listed in Tables 2 and 3, for $K = 500$.

Another purpose of our computational tests was to compare the solutions given by the SSOCP and SSDP models. The results obtained for the optimal values of the objective function and the decision variables are the same for both models. Since the SSDP model requires a lot of memory and computational time, we were able to solve this model with at most 500 scenarios. On the other hand, by using the SSOCP model we managed to consider as many as 20250 scenarios.

In order to validate the SSOCP model, we have first analyzed the sensitivity of the solutions to the number of scenarios and we have reported the relative results in Table 3. From these results, we have deduced that the model possesses an *in-sample stability*, i.e. the optimal objective values, given by $\tilde{\eta}$, are approximately the same when different numbers of scenarios are considered (for a definition of in-sample stability see Kaut and Wallace, [17]). In particular, Fig. 3 shows the convergence of the objective function as the number of scenarios increases.

The execution time for the largest problem solved, where 20250 scenarios were considered, was 17 seconds. The problem consists of 25 blocks of equations, 303769 single equations, 19 blocks of variables, and 263269 single variables. The interior point method from Mosek needed 43 iterations to solve this problem.

However, the values presented in Table 3 represent the *in-sample* values, so the costs are not directly comparable. To be able to estimate the effect of using a better scenario tree, we have to compare the *out-of-sample* costs (see again Kaut and Wallace [17]). For this purpose, we declare the tree composed by 11268 scenarios to be the true representation of the real world, and use it as a *benchmark* to evaluate the cost of optimal solutions obtained using other trees with a smaller number of

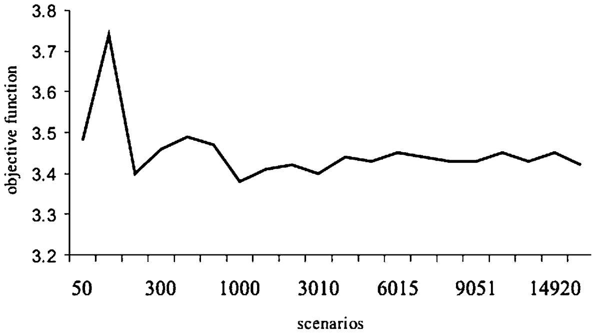

Fig. 3 Convergence of the optimal function value for an increasing number of ellipsoid scenarios

Table 4 Out-of-sample objective value obtained by substituting the first-stage solutions in the case of $K = 50, 100, 200, 300, 400$ scenarios, into the benchmark tree made by 11268 branches

K in 11268	obj. value
50 in 11268	3.44
100 in 11268	3.43
200 in 11268	3.44
300 in 11268	3.43
400 in 11268	3.44

Table 5 First stage decision solutions and optimal profit value for the *expected value problem* with one scenario given by (58)

K	$\tilde{u}_1$	$\tilde{u}_2$	d_1	d_2	γ	τ	ρ	η	$\tilde{\rho}$	$\tilde{\eta}$
1	2.77	0.04	2.77	3.15	4.54	1.77	1.77	1.85	1.77	1.85

branches or scenarios. We report in Table 4 some of the results of the out-of-sample analysis using as benchmark the case of 11268 scenarios.

To check the importance of modeling the randomness of the parameters, we compare the optimal solutions and objective value of the stochastic model with those obtained from the corresponding *deterministic* model, where we consider an unique scenario represented by an ellipse with the center $(u_1^{\text{mean}}, u_2^{\text{mean}}) = (2.8248, -0.0073)$, the angle $\varphi^{\text{mean}} = 0.7846$, the semiaxes $\sigma_1^{\text{mean}} = 1.7728$, $\sigma_2^{\text{mean}} = 1.0453$, given respectively by the mean of the centers, of the angles and of the semi-axes of the ellipses E_k, $k = 1, \dots, 11268$. Its parametric equation is given by

$$\begin{aligned} u_1 &= 1.7728 \cos(0.7846) \cos\vartheta - 1.0453 \sin(0.7846) \sin\vartheta + 2.8248, \quad \vartheta \in [0, 2\pi], \\ u_2 &= 1.7728 \sin(0.7846) \cos\vartheta + 1.0453 \cos(0.7846) \sin\vartheta - 0.0073, \quad \vartheta \in [0, 2\pi]. \end{aligned} \tag{58}$$

In the literature, this kind of problem is called *expected value problem* or *mean value problem*, (see Birge and Louveaux [11], and Kall and Wallace [18]). The solutions of the deterministic model are reported in Table 5 and shown in Fig. 4.

Because in a deterministic problem the future is completely known, the disks C, C_1, and $\widetilde{C}$ coincide, and consequently the total cost is much smaller than in the stochastic case. The resulting expected region appears to be too small to be useful in

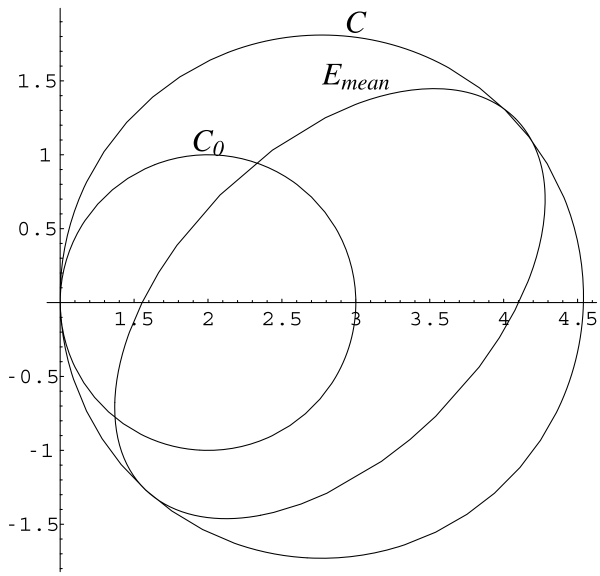

Fig. 4 Solution of the *expected value problem* with one scenario described by the ellipse given by (58)

practice. However, we have to remember that this is an *in-sample* objective value (using the terminology from [17]) and the true cost of the solution—or the *out-of-sample* objective value—is likely to be higher. To further investigate the relation between the deterministic problem and the stochastic problem we have solved the stochastic model with 11268 scenarios and the first-stage variables fixed to the deterministic solution. The result is a total cost of 3.62, much higher than the predicted (in-sample) cost of 1.85. Furthermore, we see that the resulting total cost is higher than the optimal solution for the benchmark tree with 11268 branches. The difference is known as the *value of stochastic solution* (VSS), (see e.g. Birge and Louveaux [11]). In our case, this is:

$$\begin{aligned} VSS &= \text{obj. val. (det. sol. on benchmark tree)} - \text{obj. val. (opt. sol. of benchmark tree)} \\ &= 3.62 - 3.43 = 0.19. \end{aligned}$$

This shows that one can save about 5% of the cost by using the stochastic model.

Another measure of the role of the randomness of the parameters in the model is given by the *Expected value of perfect information* (EVPI) (see again e.g. Birge and Louveaux [11]). EVPI is defined as the difference between the optimal objective value of the stochastic model with 11268 scenarios, also called *here-and-now* solution, and the expected value of the *wait-and-see* solution (WS), calculated by finding the optimal solution for each possible realization of the random variables, as follows:

$$\begin{aligned} \text{EVPI} &= \text{obj. val. (opt. sol. of benchmark tree)} - \text{obj. val. (WS)} \\ &= 3.43 - 2.43 = 1. \end{aligned}$$

The large value obtained for EVPI means that the randomness plays an important role in the problem.

6 Conclusions

In this paper we have presented a new algorithm for computing the expected zone in mobile ad hoc networks. The algorithm is obtained by reformulating the semidefinite programming model proposed by Ariyawansa and Zhu [3] as a second-order cone programming problem. Since the second-order cone programming problem can be solved much more efficiently than the corresponding semidefinite programming problem, we have been able to consider a much larger number of scenarios. This has allowed us to obtain sufficient numerical evidence for validating our model. We believe that the expected region $\widetilde{C}$ defined in this paper has the potential to be very useful in practice.

References

1. Ko, Y.-B., Vaidya, N.H.: Location-aided routing (LAR) in mobile ad hoc networks. Wireless Netw. **6**(4), 307–321 (2000)
2. Vyas, N.: Mobility pattern based routing algorithm for mobile ad hoc wireless networks. Ms Thesis, CSE Department, Florida Atlantic University (August 2000)
3. Ariyawansa, K.A., Zhu, Y.: A preliminary set of applications leading to stochastic semidefinite programs and chance-constrained semidefinite programs. Math. Comput. Model. (under review)
4. Zhu, Y., Zhang, J., Patel, K.: Location-aided routing with uncertainty in mobile ad-hoc networks: A stochastic semidefinite programming approach. ASU working paper. Submitted to InfoCom 2009, Rio de Janeiro, Brazil, April 19–25, 2009 (2008)
5. De Klerk, E.: Aspects of Semidefinite Programming. Series Applied Optimization, vol. 65. Kluwer Academic Publishers, Dordrecht (2002)
6. Pataki, G. (ed.): Computational Semidefinite and Second Order Cone Programming: The State of the Art. Springer, Heidelberg (2003)
7. Todd, M.J.: Semidefinite optimization. Acta Numer. **10**, 515–560 (2001)
8. Wolkowicz, H., Saigal, R., Vandenberghe, L.: Handbook of Semidefinite Programming International Series in Operations Research & Management Science. Kluwer Academic Publishers, Boston (2000)
9. Beale, E.M.L.: On minimizing a convex function subject to linear inequalities. J. R. Stat. Soc., Ser. B, Methodol. **17**, 173–184 (1955) Discussion, 194–203
10. Dantzig, G.B.: Linear programming under uncertainty. Manag. Sci. **1**, 197–206 (1995)
11. Birge, J.R., Louveaux, F.: Introduction to Stochastic Programming. Springer, New York (1997)
12. Ruszczyński, A., Shapiro, A. (eds.): Stochastic Programming. Series Handbooks in Operations Research and Management Science, vol. 3. Elsevier, Amsterdam (2003)
13. Ariyawansa, K.A., Zhu, Y.: Stochastic semidefinite programming: A new paradigm for stochastic optimization. 4OR, Q. J. Oper. Res. **4**(3), 239–253 (2006)
14. Shapiro, A.: Stochastic programming approach to optimization under uncertainty. Math. Program., Ser. B **112**(1), 183–220 (2008)
15. Alizadeh, F., Goldfarb, D.: Second-order cone programming. Math. Program., Ser. B **95**(1), 3–51 (2003)
16. Benson, S.J., Ye, Y., Zhang, X.: Solving large-scale sparse semidefinite programs for combinatorial optimization. SIAM J. Optim. **10**, 443–461 (1997)
17. Kaut, M., Wallace, S.W.: Evaluation of scenario generation methods for stochastic programming. Pac. J. Optim. **3**(2), 257–271 (2007)
18. Kall, P., Wallace, S.W.: Stochastic Programming. Wiley, Chichester (1994)