Optimal path planning using Cross-Entropy method

F. Celeste , F.Dambreville
CEP/Dept. of Geomatics Imagery Perception
94114 Arcueil France
{francis.celeste, frederic.dambreville}@etca.fr

J.-P. Le Cadre
IRISA/CNRS
Campus de Beaulieu
35042 Rennes France
lecadre@irisa.fr

Abstract - ***This paper addresses the problem of optimizing the navigation of an intelligent mobile in a real world environment, described by a map. The map is composed of features representing natural landmarks in the environment. The vehicle is equipped with a sensor which allows it to obtain range and bearing measurements from observed landmarks during the execution. These measurements are correlated with the map to estimate its position. The optimal trajectory must be designed in order to control a measure of the performance for the filtering algorithm used for the localization task. As the mobile state and the measurements are random, a well-suited measure can be a functional of the approximate Posterior Cramer-Rao Bound. A natural way for optimal path planning is to use this measure of performance within a (constrained) Markovian Decision Process framework. However, due to the functional characteristics, Dynamic Programming method is generally irrelevant. To face that, we investigate a learning approach based on the Cross-Entropy method.***

Keywords: Markov Decision Process, planning, estimation, Posterior Cramer Rao Bound, Cross Entropy method.

1 Introduction

In this paper we are concerned with the task of finding a plan for a vehicle moving around in its environment. That is to say, reaching a given goal position from an initial position. In many application, it is crucial to be able to estimate accurately the state of the mobile during the execution of the plan. So it seems necessary to couple the planning and the execution stages. One way to achieve that aim is to propose trajectories where the performance of the localization algorithm can be guaranteed. This problem have been well studied and in most approaches the environment is discretized and described as a graph whose nodes correspond to particular area and edges are actions to move from one place to another. Some of these previous contributions address the problem within a Markov Decision Process (MDP). Such approaches also use a graph representation of the mobile's state space and provide theoretical framework to solve it in some cases.

In the present paper, we will also use the constrained MDP framework, as in[1]. Our optimality criterion is based on the Posterior Cramer-Rao bound. However, the nature of the objective function for path planning makes it impossible to perform complete optimization of MDP with classical means. Indeed, we will show that the reward in one stage of our MDP depends on all the history of the trajectory. To solve the problem, the Cross-Entropy originally used for *rare-events* simulation seemed a valuable tool.

The paper is organized in five parts. In the second section we introduce the problem in details. Section three deals with the Posterior Cramér-Rao bound and its properties. We also derive its formulation for our problem and the criterion for the optimization. In section four, we make a short introduction of the Cross-entropy and show how to apply it to our needs. Finally, in section five the results of a first example are discussed.

2 Problem statement

Let X_k, A_k and Z_k respectively denote the state of the vehicle, the action and the vector of observations at time k. We consider that the state vector X_k is the position of the vehicle in the x-axis, y-axis, so that $X_k \triangleq (x_k, y_k)$. The action vector $A_k \triangleq (a_x, a_y)$ is restricted to the set $\{-d, 0, +d\} \times \{-d, 0, +d\}$ with $d \in \mathbb{R}^+$ given constant.
So, the state $\{X_k\}$ motion is governed by a dynamic model and an initial uncertainty, given by :

$$X_0 \sim \pi_0 \triangleq \mathcal{N}(\overline{X_0}, P0)$$
$$X_{k+1} = f(X_k, A_k) + w_k \quad w_k \sim \mathcal{N}(0, Q_k)$$

where $\{w_k\}$ is a white noise sequence and the symbol "$\sim$" means distributed according to [1].

The known map $\mathcal{M}$ is composed of N_f features $(m_i)_{1 \leq i \leq N_f}$ with respective position $(x_i, y_i) \in \mathbb{R}^2$. At time k, due to the sensor capabilities, only a few landmarks can be observed. So, the observation process depends on the number of visible features. Moreover, we do not consider data association and non detection problem for the moment. That is to say, each observation is made from one landmark represented in the map. If we denote $I_v(k)$ the indexes

[1] The symbol $\mathcal{N}$ means normal

of visible landmarks at time k, the measurement vector received at time k is :

$$Z_k = \{z^k(j)\}_{j \in I_v(k)}$$

with $\forall j \in I_v(k)$:

$$z^k(j) \triangleq \begin{cases} z_r^k(j) = \sqrt{(x_k - x_j)^2 + (y_k - y_j)^2} + \gamma_r^k(j) \\ z_\beta^k(j) = \arctan(\frac{y_j - y_k}{x_j - x_k}) - \theta_k + \gamma_\beta^k(j) \end{cases}$$

where θ_k is the global mobile orientation, $\{\gamma_r^k(j)\}$ and $\{\gamma_\beta^k(j)\}$ are considered as white Gaussian noise and mutually independent. Nevertheless, a more complex modeling must be considered if we want to take into account correlated errors in the map [2] or observations[3].

2.1 A Markov Decision Process model with constraints

As in [1], we first formulate the problem within the Markov decision Process framework. Generally, a Markov Decision Process is defined by its state and action sets, the state transition probabilities and reward functions. In our case, the state and action spaces are discrete and finite. Indeed the map is discretized in $N_s = N_x \times N_y$ locations and one action is defined as a move between two neighbor points (figure 1), where N_x and N_y are the number of grid points in both axis. At each point, the mobile can choose one action among N_a. So we can model the process with:

- $S = \{1, ..., N_s\}$ the state space.
- $A = \{1, ..., N_a\}$ the action space.
- $T_{ss'}(a) \triangleq Pr(s_{k+1} = s' | s_k = s, a_k = a)$ the transition function.
- $R_{ss'}(a)$ the reward or cost function, associated with transition from state s to state s', for an action a.

Finding the optimal path between two states s_i and s_f is equivalent to determine the *best* sequence of actions or decisions $V_a^* = (a_0^*, ..., a_K^*)$ allowing to simultaneously connecting them and optimizing a (global) criterion. In our application, we consider only 8 possible actions for one state (figure 1). Each of them can be selected except for the points located on the border of the map and in places where obstacles can be found. Moreover, we take into account operational constraints on the mobile dynamic between two following epochs as in [1]. It is possible to do that by defining one *authorized transition matrix* $\delta(a_k, a_{k-1})$ which indicates actions that can be chosen at time k according to the choice at time $k-1$. For example, if only $[-\frac{\pi}{2}, \frac{\pi}{2}]$ headings controls are allowed, such a matrix can be expressed as below :

[2] e.g. Markov Random Fields
[3] e.g. colored noise or biases

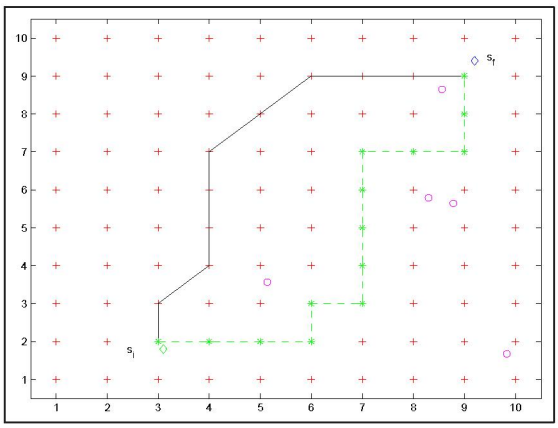

Figure 1: MDP grid with states (red crosses). Trajectories with (solid line) and without(dashed line) $[-\frac{\pi}{4}, \frac{\pi}{4}]$ constrained headings change.

$\delta =$

a_{k+1}	1	2	3	4	5	6	7	8
a_k								
1	1	1	1	0	0	0	0	1
2	1	1	1	1	0	0	0	0
3	1	1	1	1	1	0	0	0
4	0	1	1	1	1	1	0	0
5	0	0	1	1	1	1	1	0
6	0	0	0	1	1	1	1	1
7	1	0	0	0	1	1	1	1
8	1	1	0	0	0	1	1	1

The actions are clockwise numbered from 1 ("go up and right") to 8 ("go up"). The above δ matrix indicates that if action 1 is chosen at time k, only actions $1, 2, 3, 7$ and 8 can be selected at time $k+1$. Moreover, the mobile orientation θ_k is restricted to 8 values directly linked with the orientation of the elementary displacements.

In the Markov Decision Process context when the reward is completely known, optimal policy can be computed using Dynamic Programming technique or similar algorithms such as Value Iteration and Policy Iteration [8]. They are based on the principle of optimality due to Bellman. However, applicability of this principle implies some basic assumptions of the cost functional. It is not the case in our context where the criterion of optimality depends on the Posterior Cramér Rao Bound (PCRB) matrix which is introduced in next section. Indeed, as shown in [2], any cost functional must satisfy the "Matrix Dynamic Programming Property" to guarantee the principle of optimality. For the "determinant" functional used in our work, this property is not verified [2].

3 Posterior Cramér-Rao Bound

3.1 Definition

In this section, we briefly remind the properties of the Cramér-Rao bound for estimation problem. Let $\hat{X}(Z)$

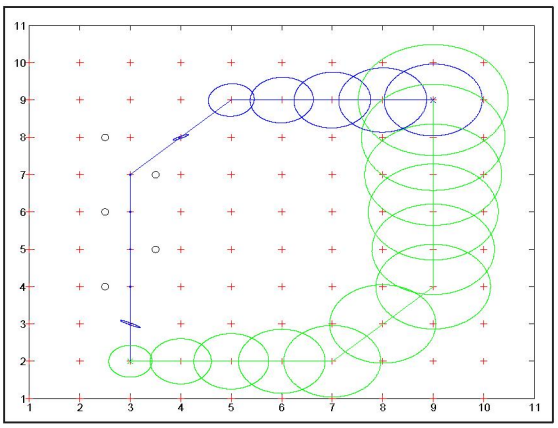

Figure 2: example of 90% error ellipses for 2 trajectories of the same length (features are in black (o))

4 Cross Entropy algorithm

In this section we briefly introduce the Cross Entropy algorithm. The reader interested in the C.E. method and its convergence properties should refer to [5, 6].

4.1 A short introduction

The Cross Entropy method was first used to estimate the probability of *rare events*. A rare event is an event with very small probabilities. It was then adapted for optimization assuming that sampling around the optimum of a function is a rare event.

4.1.1 Simulation of rare event

Let X a random variable on a space $\mathcal{X}$, p_x its probability density function (pdf) and ϕ a function on $\mathcal{X}$. Suppose, we are concerned with estimating $l(\gamma)$ the probability of the event $F_\gamma = \{x \in \mathcal{X} | \phi(x) \geq \gamma\}$ with $\gamma \in \mathbb{R}^+$. F_γ is a rare event if $l(\gamma)$ is very small. An unbiased estimate can be obtained via Crude Monte Carlo simulation. Given a random sample $(x_1, \cdots, x_N)$ drawn from p_x, this estimate is :

$$\hat{l}(\gamma) = \frac{1}{N} \sum_{i=1}^{N} I\left[\phi(x_i) \geq \gamma\right]$$

For *rare event*, the variance of $\hat{l}(\gamma)$ is very high and it is necessary to increase N to improve the estimation. The estimate properties can be improved with *variance minimization* technique such as *importance sampling* where the random sample is drawn from a more appropriate probability density function q. It can be easily shown that the optimal q^* pdf is given by $I\left[f(x) > \gamma\right] p_x(x)/l(\gamma)$. Nevertheless, q^* depends on $l(\gamma)$ which needs to be estimated. To get round this difficulty, it can be convenient to choose q in a family of pdfs $\{\pi(., \lambda) | \lambda \in \Lambda\}$. The idea is to find the optimal parameter λ^* such as $\mathcal{D}(q^*, \pi(., \lambda))$ is minimized where $\mathcal{D}$ is the *Kullback-Leibler* "pseudo-distance" :

$$\mathcal{D}(p, q) = \mathbb{E}_p \ln \frac{p}{q} = \int_X p(x) \ln p(x) dx - \int_X q(x) \ln p(x) dx$$

Minimizing $\mathcal{D}(q^*, \pi(., \lambda))$ is equivalent to maximize $\int_X q^*(x) \ln \pi(., \lambda) dx$ which implies :

$$\lambda^* \in \arg\max_{\lambda \in \Lambda} \mathbf{E}_p \left(I\left[\phi(x) \geq \gamma\right] \ln \pi(x, \lambda) \right) \tag{11}$$

The computation of the expectation in 11 must also be done using importance sampling. So we need a change of measure q, drawing one sample $(x_1, \cdots, x_N)$ from q and estimate λ^* as follows :

$$\hat{\lambda^*} \in \arg\max_{\lambda \in \Lambda} \sum_{i=1}^{N} I\left[\phi(x_i) \geq \gamma\right] \frac{p_x(x_i)}{q(x_i)} \ln \pi(x_i, \lambda) \tag{12}$$

The family $\{\pi(., \lambda) | \lambda \in \Lambda\}$ must be chosen to easily solve equation 12. Natural Exponential Families[5] (N.E.F)[5] are especially adapted. however, q is not known in equation 12. The C.E algorithm tries to overcome this difficulty by constructing adaptively a sequence of parameters $(\gamma_t | t \geq 1)$ and $(\lambda_t | t \geq 1)$ such as :

- F_{γ_1} is not a rare event.
- $F_{\gamma_{t+1}}$ is not a rare event for $\pi(., \lambda_t)$.
- $\lim_{t \to \infty} = \gamma$.

More precisely, given $\rho \in]0, 1[$:

- choose λ_0 such as $\pi(., \lambda_0) = p_x$.
- draw $(x_1, \cdots, x_N)$ from $\pi(., \lambda_{t-1})$.
- sort in increasing order $(\phi(x_1), \cdots, \phi(x_N))$ and evaluate the $(1 - \rho))$ quantile γ_t
- compute λ_t as

$$\lambda_t \in \arg\max_{\lambda \in \Lambda} \sum_{i=1}^{N} I\left[\phi(x_i) \geq \gamma_t\right] \frac{p_x(x_i)}{\pi(x_i, \lambda_{t-1})} \ln \pi(x_i, \lambda)$$

- if $\gamma_t < \gamma$, set $t = t + 1$ and go to step 2. Else estimate the probability of F_γ with :

$$\hat{l}(\gamma) = \frac{1}{N} \sum_{i=1}^{N} I\left[\phi(x_i) \geq \gamma\right] \frac{p_x(x_i)}{\pi(x_i, \lambda_t)}$$

This is the main C.E algorithm but other versions can be found in [5].

4.1.2 application to optimization

The C.E. was adapted to solve optimization problem. Consider the optimization problem :

$$\phi(x^*) = \gamma^* = \max_{x \in \mathcal{X}} \phi(x) \tag{13}$$

The principle of C.E for optimization is to translate the problem 13 into an *associated stochastic problem* and then solved it adaptively as the simulation of a rare event. If $\gamma*$ is the optimum of ϕ, F_{γ^*} is generally a rare event. The main idea is to define a family $\pi(., \lambda) | \lambda \in \Lambda$

[5] pdfs $f_\lambda(x) = c(\lambda) e^{\lambda \cdot \mathbf{t}(x)} \cdot h(x)$

where I is the indicator function and $\{x(j) \in \chi_{sa}\}$ means that the trajectory contains a visit to state s in which action a is taken. Since for each s, the row $P_s(.)$ is a discrete probability, 14 must be solved under the condition that the rows of $\mathbf{P_{sa}}$ sum up to 1. This yields to solve the following problem using Lagrange multipliers $(\mu_s)_{\{1 \leq s \geq N_s\}}$:

$$\max_{\mathbf{P_{sa}}} \min_{\mu_1, \cdots, \mu_{N_s}} \frac{1}{N} \sum_{j=1}^{N} I\left[\phi_k(x(j)) \geq \gamma_t\right] \ln \pi(x(j), \mathbf{P_{sa}}) \quad (17)$$
$$+ \sum_{s=1}^{N_s} \mu_s \sum_{a=1}^{a=8} (p_{sa} - 1)$$

with $\ln \pi(x(j), \mathbf{P_{sa}})$ given in 16. We can derive after differentiation with respect each parameter p_{sa} ($s \in \{1, \cdots, N_s\}$ $a \in \{1, \cdots, 8\}$) the following equation.

$$\frac{1}{N} \sum_{j=1}^{N} I\left[\phi_k(x(j)) \geq \gamma_t\right] I\left[\{x(j) \in \chi_{sa}\}\right] = -\mu_s \, p_{sa}$$

After summing over $a = 1, \cdot, 8$ and applying the condition $\sum_{a=1}^{8} p_{sa} = 1$, we can obtain μ_s and the final updating formula.

$$\mu_s = -\frac{1}{N} \sum_{j=1}^{N} I\left[\phi_k(x(j)) \geq \gamma_t\right] I\left[\{x(j) \in \chi_s\}\right] \quad (18)$$

$$p_{sa} = \frac{\sum_{j=1}^{N} I\left[\{\phi_k(x(j)) \leq \gamma_t\}\right] \cdot I\left[\{x(j) \in \chi_{sa}\}\right]}{\sum_{j=1}^{N} I\left[\{\phi_k(x(j)) \leq \gamma_t\}\right] \cdot I\left[\{x(j) \in \chi_s\}\right]} \quad (19)$$

where $\{x(j) \in \chi_s\}$ means that the trajectory contains a visit to state s. For the first iteration, $\forall s$, $P_s(.)$ is a uniform probability density function.

5 Simulation results

The algorithm has not been widely tested, and only one simple scenario is introduced in this paper. In this example, the map is defined on $[-2, 12] \times [-2, 12]$ with 6 m_j point features (figure 3). The state space is discretized in $N_s = 15 \times 15$ states. The initial and terminal states are respectively in positions $(0, 0)$ and $(10, 10)$.

	m_0	m_1	m_2	m_3	m_4	m_5
x	1.6	2.3	3.7	4.2	5.2	7.6
y	0.8	5.1	6.4	7.8	8.7	8.2

Table 2: the features of the map.

For the mobile dynamic model, the elementary displacement d is equal to 1 and the noise process covariance is defined by:

$$P_0 = \sigma^2 \cdot \mathcal{I}_{22}, \qquad Q_k = \sigma^2 \cdot \mathcal{I}_{22}, \forall k \geq 1$$

where $\mathcal{I}_{22}$ is the identity matrix of size 2×2 and $\sigma^2 = 0.05$.

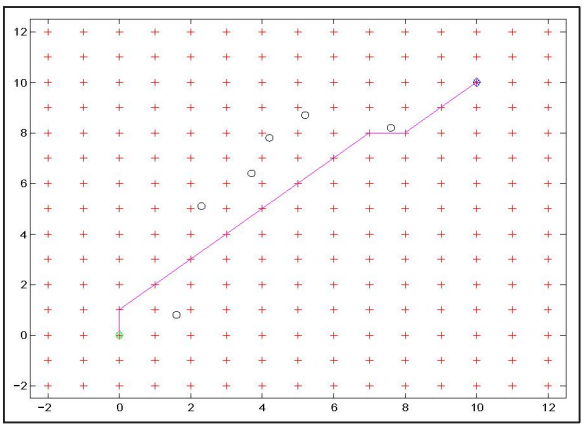

Figure 3: The maximum likelihood optimal solution after the first iteration of the CE. Starting and terminal states ($\diamond$) and map features ($\circ$).

The mobile can only apply $[-\frac{\pi}{2}; \frac{\pi}{2}]$ headings controls at each time. Thus the δ matrix is exactly the same defined as in section 2. Only trajectories with length $T \leq 30$ are admissible.
For the observation model, a landmark m_j is visible at time k provided that:

$$\begin{array}{l} r_{min} \leq z_r^k(j) \leq r_{max} \\ |z_\beta^k(j)| \leq \theta_{max} \end{array}$$

with $r_{min} = 0.001$, $r_{max} = 2$ and $\theta_{max} = 40$ deg.
The noise variance σ_a^2 on the range and σ_β^2 on the bearing are the same for all features and are time independent :

$$\sigma_a = 1.5.10^{-3}, \qquad \sigma_\beta = 0.5 \text{ deg.}$$

The computation of the PCRB matrices was performed with $N_c = 1000$ to estimate the observation term $J_k(Z)$. For the optimization step, the Cross Entropy algorithm was implemented with 1000 iterations, $N = 5000$ admissible trajectories and a selective rate $\rho = 0.1$. That is to say, the 50 best samples are used for updating the psa probabilities. Figure 4 show the optimal trajectory found by the algorithm at iteration 1000 for the performance function ϕ_1.

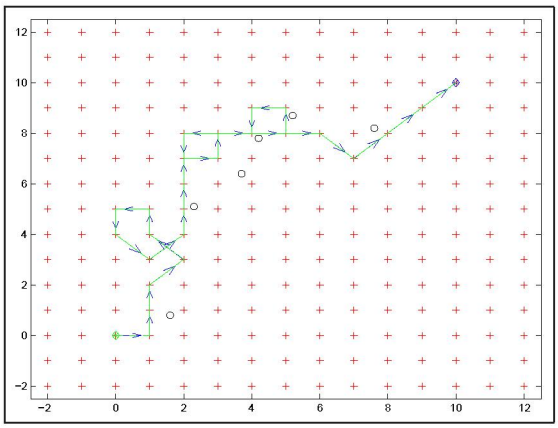

Figure 4: The most likely optimal trajectory found by the algorithm for ϕ_1.

5.1 Analysis

As expected (figure 4) the mobile is guided toward the area with landmarks in order to improve its performance of localization. Moreover, it operates to keep the landmarks visible while the maneuvers (δ) and the time constraints (T_{max}) allow it. We can also notice that the algorithm converge rapidly to a solution. To illustrate that we present in the next figure, the evolution of parameters γ and the minimum value of ϕ_1 (or the maximum of $-\phi_1$) at each iteration of the CE algorithm (figure 5).

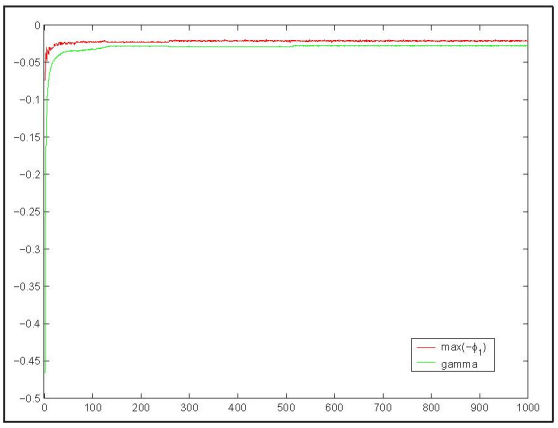

Figure 5: Evolution of γ (solid line) and the minimum value of the functional (dashed line).

When we look at precisely after convergence the densities ($P_s(.)$) for all s in the optimal trajectory we can notice that some of them are not a dirac probability law. In some state, The "most likely trajectory" optimal trajectory is composed of 30 states, only 23 have their associated P_s equivalent to a dirac probability law . Table 3 shows the probability density functions for the others (see figure 6) . In these states the algorithm converge toward a multi-modal density. Two actions can be chosen but with different probability. We can notice that this behavior are concentrated on state where maneuvers can be done to increase the time to observe the landmarks.

	3	9	16	18	20
1	0.4179	0	0	0	0.8134
2	0	0.4329	0.5903	0.7988	0
3	0	0.5671	0	0	0
4	0	0	0	0	0.1866
5	0	0	0	0	0
6	0	0	0.4097	0	0
7	0	0	0	0	0
8	0.5821	0	0	0.2012	0

Table 3: probability densities functions.

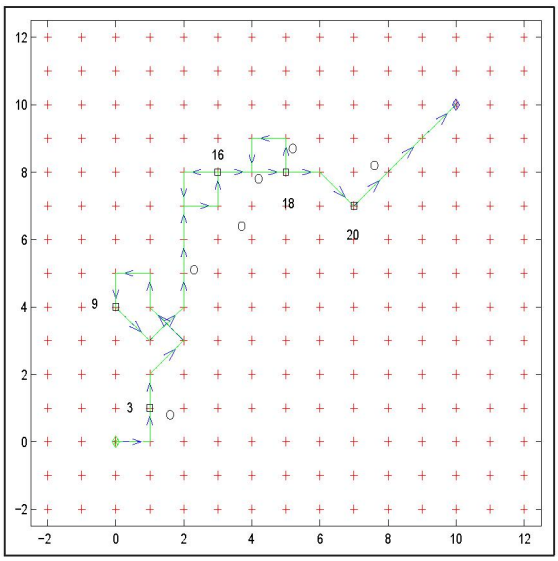

Figure 6: state (squares) with pdf different from a dirac after convergence.

6 Conclusions and perspectives

In this paper, we presented a framework to solve a path planning task for a mobile with the . The problem was discretized and a Markov Decision Process with constraints on the mobile maneuver was used. Our main goal was to find the optimal trajectory according to a measure of capability of estimating accurately the state of the mobile during the execution. Functionals of the Posterior Cramr-Rao Bound was used as the criterion of performance. The main contribution of the paper is the use of the Cross Entropy algorithm to solve the optimization step as Dynamic Programming could not be applied. This approach was tested on a simple first example and seems to be relevant.
Future work will first concentrate on the complete implementation of the algorithm and applications to more examples. More analysis on the probability has to be made. We will also investigate a continuous approach and try to approximate the computation of the observation contribution to the PCRB, which is time consuming. The tuning of the Cross-Entropy to our specific task was not studied, some experiments have to be carried out based on device given in [5]. Finally, we want to consider more complex maps such those used in Geographical Information Systems and take into account measurement models with data association and non detection problem.

References

[1] S. Paris, J-P. Le Cadre Planning for Terrain-Aided Navigation, Fusion 2002, Annapolis (USA), pp 1007–1014, 7-11 Jul. 2002.

[2] J.-P. Le Cadre and O. Tremois , *The Matrix Dynamic Programming Property and its Implications.*. SIAM Journal on Matrix Analysis, 18 (2): pp 818-826, April 1997.