International Journal of Informatics and Communication Technology (IJ-ICT)
Vol. 6, No. 2, August 2017, pp. 105~116
ISSN: 2252-8776, DOI: 10.11591/ijict.v6i2.pp105-116

A Markov Decision Model for Area Coverage in Autonomous Demining Robot

Abdelhadi Larach*, Cherki Daoui, Mohamed Baslam
Laboratory of Information Processing and Decision Support, Sultan Moulay Slimane University Faculty of sciences and Techniques, Beni-Mellal, Morocco

Article Info

Article history:

Received Feb 16, 2017
Revised Jun 29, 2017
Accepted Jul 19, 2017

Keywords:

Coverage Path Planning
Markov Decision Process
Robotic Path planning
Shortest Path Planning

ABSTRACT

A review of literature shows that there is a variety of works studying coverage path planning in several autonomous robotic applications. In this work, we propose a new approach using Markov Decision Process to plan an optimum path to reach the general goal of exploring an unknown environment containing buried mines. This approach, called Goals to Goals Area Coverage on-line Algorithm, is based on a decomposition of the state space into smaller regions whose states are considered as goals with the same reward value, the reward value is decremented from one region to another according to the desired search mode. The numerical simulations show that our approach is promising for minimizing the necessary cost-energy to cover the entire area.

Corresponding Author:

Abdelhadi Larach,
Laboratory of Information Processing and Decision Support,
Sultan Moulay Slimane University,
Faculty of sciences and Techniques, Beni-Mellal, Morocco.
Email: larachabdelhadi@gmail.com

1. INTRODUCTION

Shortest Path Planning (SPP) or Coverage Path Planning (CPP) is a task used in a large number of robotic applications, such as demining robots [1], painter robots [2], cleaning robots [3], etc. Several researches concerning SPP and CPP are presented in [4-6], SPP or CPP algorithms are classified in two categories: off-line algorithms, generally used in acknowledged environment and on-line algorithms, used in unrecognized environment. In on-line algorithms the policy is generally updated according to new environment observation. The CPP problem remains subject of research optimization, especially in an unknown environment.

In Robotic landmines detection, the agent must detect and find location of all possible mines, avoid all obstacles and follow the shortest path in an unknown environment without overlapping paths. Satisfying such requirements is not always easy and possible. In fact, the whole structure of the environment is not known a priori and the on-line algorithm must be able to seek the optimal strategy according to the knowledge acquired after each observation. E.Galseran [6] presented a survey on CPP in Robotic, several methods were proposed.

In this paper, we present a Discounted Markov Decision Model for robotics navigation in grid environment with a theoretical study which permit to propose a new approach for area coverage called Goals to Goals Area Coverage on-line Algorithm based on a decomposition of the state space into smaller regions whose all states are considered as goals and assigned with the same reward value. The reward value can be decreased from one region to another according to the desired search mode such as a line-sweep [13] or spatial cell diffusion [14] approaches.

2. MARKOV DECISION PROCESS

Markov Decision Processes are defined as controlled stochastic processes satisfying the Markov property and assigning reward values to state transitions [7, 8]. Formally, they are defined by the five-tuple *(S, A, T, P, R),* where *S* is the state space in which the process's evolution takes place; *A* is the set of all possible actions which control the state dynamics; *T* is the set of time steps where decisions need to be made; *P* denotes the state transition probability function where $P(S_{t+1}=j|S_t=i, A_t=a)=P_{iaj}$ is the probability of transitioning to a state *j* when an action a is executed in a state *i*, S_t (A_t) is a variable indicating the state (action) at time *t; R* provides the reward function defined on state transitions where R_{ia} denotes the reward obtained if the action a is applied in state i.

2.1. Discounted Reward Markov Decision Process

Let $P_\pi(S_t = j, A_t = a|S_0 = i)$ be the conditional probability that at time **t** the system is in state **j** and the action taken is *a*, given that the initial state is *i* and the decision maker is a strategy π; if R_t denotes the reward at time *t*, then for any strategy π and initial state *i*, the expectation of R_t is given by:

$$\mathbb{E}_\pi(R_t, i) = \sum_{\substack{j\in S \\ a\in A(j)}} P_\pi(S_t=j, A_t=a|S_0=i)R_{ja} \quad (1)$$

In discounted reward MDP, the value function, which is the expected reward when the process starts with state *i* and using the policy π is defined by:

$$V_\pi^\alpha(i) = \mathbb{E}[\sum_{t=0}^{\infty} \alpha^t \mathbb{E}_\pi(R_t, i)],\ i \in S \quad (2)$$

where $\alpha \in]0, 1[$ is the discount factor.

The objective is to determine V^*, the maximum expected total discounted reward vector over an infinite horizon. It is well known [7, 8] that V^* satisfies the Bellman equation:

$$V(i) = \max_{a\in A(i)} \{R_{ia} + \alpha \sum_{j\in S} P_{iaj} V(j)\}, i \in S \quad (3)$$

Moreover, the actions attaining the maximum in (3) give rise to an optimal pure policy π^* given by:

$$\pi^*(i) \in \operatorname*{argmax}_{a\in A(i)} \{R_{ia} + \alpha \sum_{j\in S} P_{iaj} V^*(j)\}, i \in S \quad (4)$$

2.2. Gauss-Seidel Value Iteration Algorithm

Gauss-Seidel Value Iteration (GSVI) Algorithm is one of the most iterative algorithms used for finding optimal or approximately optimal policies under Discounted MDP [8-10]. In this paragraph, we presente the following optimized pseudo-code ofGSVI Algorithm (Algorithm 1).

Algorithm 1. GSVI Algorithm.

GS**VI** (**In:** S, *P*, *A*, *R*, $\Gamma_a^+, \alpha, \varepsilon$; **Out**: V^*, π^*)
1. **For** all $i \in S$ **Do** $V^*(i) \leftarrow 0$; *//Initialization*
2. Bellman_err ← 2ε; *//For stopping criterion*
3. **While (**Bellman_err ≥ ε**) Do**
   For all i ∈ S **Do** *//Value improvement*

$$V_{tmp} \leftarrow \max_{a\in A(i)} \left\{ R_{ia} + \alpha \sum_{j\in\Gamma_a^+(i)} P_{iaj} V^*(j) \right\}$$

   Bellman_err ← **Max**($|V^*(i) - V_{tmp}|$, Bellman_err)
   $V^*(i) \leftarrow V_{tmp}$
4. **For all** $i \in S$ **Do** *//Policy calculation*

$$\pi^*(i) \leftarrow \operatorname*{argmax}_{a\in A(i)} \left\{ R_{ia} + \alpha \sum_{j\in\Gamma_a^+(i)} P_{iaj} V^*(j) \right\}$$

5. **Return** V^*, π^*

We denote by $\Gamma_a^+(i) = \{ j \in S : P_{iaj} > 0\}$ the successors list of pair $(i, a), i \in S, a \in A(i)$.

The algorithm 1 is based on a successors list of pair state-action, which permits to accelerate iteration compared to the classical GSVI Algorithm especially when the number of actions and successors per state is very less than the number of states. Indeed, the complexity of the proposed version is reduced to $\mathcal{O}(|\Gamma_a^+||S|)$ per iteration where $|\Gamma_a^+|$ is the average number of state-action successors.

3. MARKOV DECISION MODEL FOR ROBOTIC NAVIGATION

To model the Robotic Navigation in general or Demining Robot problem in particular case using a MDP, the five-tuple *(S, A, T, P, R)* must be defined and the environment representation must be choosen.

a. ***Grid's Environment:*** The Grid Based method -ideal for landmines detection- is used to model the environment which is entirely discretized according to a regular grid. The grid size can be chosen according to the robot structure and the field covered by the robot sensor.

b. ***States Space:*** Using the Grid method, the state space is therefore a set of grids, each grid cell has an associated value stating, obstacle, mine, free or goal state.

c. ***Actions Space:*** The robot can be controlled through nine actions: the eight compass directions and the action designed by θ that keeps the process in the state where it is; actions that move the robot to an obstacle or a mine state are eliminated; in a goal state the possible action is θ. Figure 1 shows the possible actions in an environment example, the black grid is an obstacle state and the green grid is a goal state.

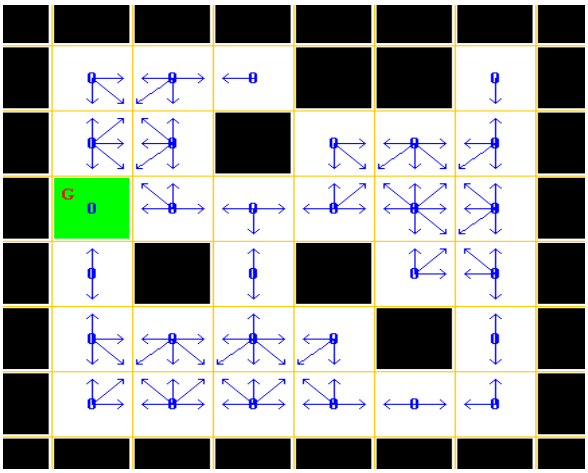

Figure 1. Environment example and possible robot actions in each state

Reward Function: A transition to a free or goal state is characterized by a cost of energy proportional to the distance travelled, it is equal to a predefined constant *x* if the action is diagonal, and $\frac{x}{\sqrt{2}}$ if the action is horizontal or vertical, the reward value is therefore equal to -*x* or $\frac{-x}{\sqrt{2}}$; the reward assigned to the action θ in a free state is equal to zero and for a goal state it is equal to a predefined constant R_b.

Transition Function: The transition function defines the uncertainty due to the effects of actions; it is a data of problem and can be determined by reinforcement learning.

Robot Sensor: Several landmines detection methods can be used such as Metal Detector Technologies, Electromagnetic Methods [11], etc. In this work, we suppose that the robot can detect the buried mines existing in near states (Figure 2) by using his owner sensing system, an arm movement or multi-sensors technologie would cover these states.

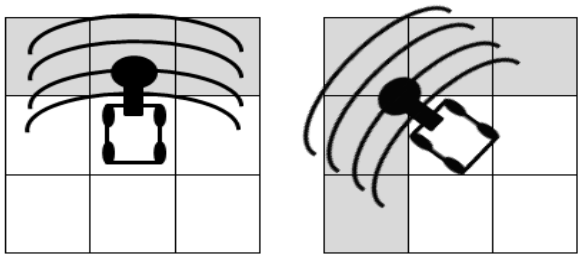

Figure 2. States covered by the robot sensor

Robot Structure: several robot mechanical structures can be used in demining robot such as Omnidirectional or Multi Directional Conductive control [12], etc.

Remark 1. Using the proposed model, the time complexity of algorithm 1 is $\mathcal{O}(|\Gamma_a^+||S'|)$ per iteration, where $|S'|$ is the number of free states; for a goal state the expected reward is a constant value equal to $\frac{R_b}{1-\alpha}$ (given by (3)) since the only possible action that can be taken in a goal state is θ; $|\Gamma_a^+|$ is very less than $|S'|$ and can be considered us constant; so the Algorithm 1 is linear per iteration.

4. THEORETICAL MODEL STUDY

In this section, we present a theoretical study for the considered model, which is the basis of our approach.

Proposition 1. If the state space do not contains any goal state then for any free state s_0, $\pi^*(s_0) = \theta$, π^* is an optimal strategy.

Proof. Suppose ∃ an optimal pure strategy π^* and ∃ $s_0 \in S$ such that: $\pi^*(s_0) \neq \theta$; the expected reward when the system start at state s_0 is:

$$V_{\pi^*}^{\alpha}(s_0) = V^*(s_0) = \mathbb{E}_{\pi^*}(R_0, s_0) + \mathbb{E}[\sum_{t=1}^{\infty} \alpha^t \mathbb{E}_{\pi^*}(R_t, s_0)] \quad (5)$$

The expectation of R_t time t=0 and using strategy π^* is given by:

$$\mathbb{E}_{\pi^*}(R_0, s_0) = \sum_{\substack{j \in S \\ a \in A(j)}} P_{\pi^*}(S_0 = s_0, A_0 = a | S_0 = s_0) R_{s_0 a} = R_{s_0 a} \quad (6)$$

We have: $R_{s_0 a} \leq \frac{-x}{\sqrt{2}}$ then $V_{\pi^*}^{\alpha}(s_0) \leq \frac{-x}{\sqrt{2}} < 0$ and the fact that $V_{\pi^*}^{\alpha}(s_0) < 0$ implies that $\pi^*(s_0)$ cannot be an optimal action since there exists a strategy π': $\pi'(s_0) = \theta$ and $V_{\pi'}^{\alpha}(s_0) = 0$, which contradicts the supposition.

Figure 3 (left) shows a simulation result in an environment example where no goal state is defined; as we can see, $\forall s_0 \in S, \pi^*(s_0) = \theta$.

Proposition 2. Let s_0 be a free initial state, if there is no path leading from s_0 to the goal state G then $\pi^*(s_0) = \theta$.

Proof. The proof is similar to the proof of Proposition 1, indeed, for any strategy π such that $\pi(s_0) \neq \theta$, $V_{\pi}^{\alpha}(s_0) < 0$.

Figure 3(right) shows an example of simulation result, as we can see for all $s_0 \in S$ where there is no path to the goal state G, $\pi^*(s_0) = \theta$.

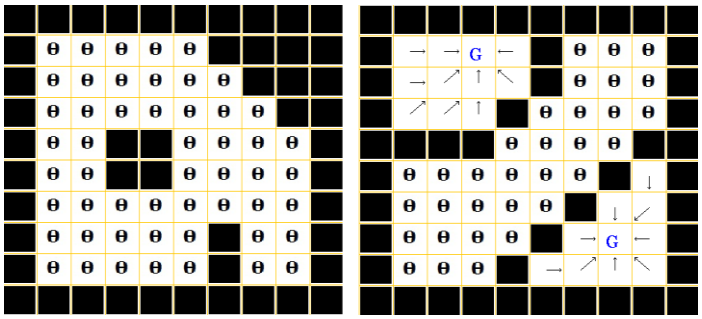

Figure 3. Optimal strategies examples (no goal state (left) and no path to the goal state for some states (right))

Proposition 3. Let *G* be a goal state with reward value $R_b=0$, then for any initial free state s_0, $\pi^*(s_0) = \theta$.
Proof. It is clear since for any strategy π such that $\pi(s_0) \neq \theta$, $V_\pi^\alpha(s_0) < 0$.

Figure 4 shows a simulation example in an environment where there exist four goals states with reward value $R_b=0$, as it can be seen$\pi^*(s_0) = \theta$ for all free initial state.

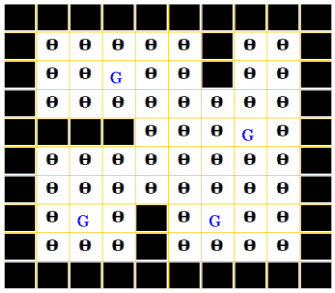

Figure 4. Optimal strategy in an environment example with four goals states within reward value R_b=0

Proposition 4. Let s_0 be a free initial state, suppose that there exist a path of length ***l***, from s_0 to the gaol state *G* and let R_b be the reward value obtained when the action θ is applied in a goal state *G*.

$$If\ R_b > \left(\frac{x}{\alpha^l} - x\right)\ then\ \pi^*(s_0)\ \neq\ \theta \tag{7}$$

Proof. Let $V^*(s_0)$ be the expected reward when the process start at state s_0.

$$V^*(s_0)\ =\ \sum_{t=0}^{\infty} \alpha^t R_t = \sum_{t=0}^{l-1} \alpha^t R_f^t + \sum_{t=l}^{\infty} \alpha^t R_b \tag{8}$$

where: $R_f^t = E_{\pi^*}(R_t, s_0) = \sum_{\substack{j\in S \\ a\in A(j)}} P_{\pi^*}(S_t = j, A_t = a | S_0 = s_0) R_{ja}$ is the expectation of the reward value obtained when some compass action is taken in the state S_t at time A_t.
The equation (8) can be modified as follow:

$$V^*(s_0) = \sum_{t=0}^{l-1} \alpha^t R_f^t + \alpha^l \sum_{t=0}^{\infty} \alpha^t R_b \tag{9}$$

Let $V^*(s_g)$ be the expected reward when the process starts at goal state *G*, by using (3), we have:

$$V^*(s_g) = \sum_{t=0}^{\infty} \alpha^t R_b = \frac{R_b}{1-\alpha} \tag{10}$$

Using (10), the equality (9) can be reformed as follow:

$$V^*(s_0)\ = \sum_{t=0}^{l-1} \alpha^t R_f^t + \frac{R_b \times \alpha^l}{1-\alpha} \tag{11}$$

The fact that: $R_{ja} \geq -x$ implies that $R_f^t \geq -x$, then:

$$\sum_{t=0}^{l-1} \alpha^t R_f^t \geq -x \sum_{t=0}^{l-1} \alpha^t \quad (12)$$

By applying (12) in (11), we have the resulting equation:

$$V^*(s_0) \geq \frac{R_b \times \alpha^l}{1-\alpha} - x\left(\sum_{t=0}^{l-1} \alpha^t\right) \quad (13)$$

The fact that $\sum_{t=0}^{l-1} \alpha^t = \frac{1-\alpha^l}{1-\alpha}$, equation (13) becomes:

$$V^*(s_0) \geq \frac{R_b \times \alpha^l - x + x \times \alpha^l}{1-\alpha} \quad (14)$$

The fact that $R_b > \left(\frac{x}{\alpha^l} - x\right)$ implies that $R_b \times \alpha^l - x + x \times \alpha^l > 0$.
Equations (14) imply that $V^*(s_0)$ is great than zero.

$$V^*(s_0) \geq \frac{R_b \times \alpha^l - x + x \times \alpha^l}{1-\alpha} > 0 \quad (15)$$

The fact that $V^*(s_0) > 0$ implies that $\pi^*(s_0) \neq \theta$.

Figure 5 shows a strategy example generated using algorithm 1 with parameters: R_b=200, α=0.2 and $x = 1$; for a path length l=3, $\boldsymbol{l}_3 = \frac{1}{\alpha^3} - 1 = 125 < R_b = 200$; and for l=4, $\boldsymbol{l}_4 = \frac{1}{\alpha^4} - 1 = 624 > R_b = 200$, we can conclude from the values of $\boldsymbol{l}_3$ and $\boldsymbol{l}_4$ that: If $\boldsymbol{l} < 4$ then $\pi^*(s_0) \neq \theta$.

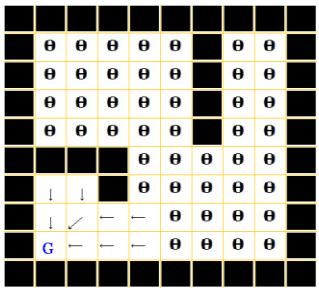

Figure 5. An optimal strategy example using parameters: α=0.2, x=1 and Rb=200

Proposition 5. Let G be a goal state with reward value $R_b = \frac{x}{\alpha^{|S|}}$. For any initial free state s_0, if there exists a path to the goal state G then $\pi^*(s_0) \neq \theta$.

Proof. Let $\boldsymbol{l}$ be the path length to the goal state G; using proposition 4, if $R_b > \left(\frac{x}{\alpha^l} - x\right)$ then $\pi^*(s_0) \neq \theta$.
Let show that $\frac{x}{\alpha^{|S|}} > \left(\frac{x}{\alpha^l} - x\right)$, we have $|S| > l$ then $\alpha^{|S|} < \alpha^l$ and $\frac{x}{\alpha^{|S|}} > \frac{x}{\alpha^l} > \frac{x}{\alpha^l} - x$, since $x > 0$.

Figure 6 shows a path strategy example with parameters: α=0.2, x=$\alpha^{|S|}$ and R_b=1 (note that if (x=$\alpha^{|S|}$) then R_b=1); as it can be seen, $\pi^*(s_0) \neq \theta$ for all free initial state s_0.

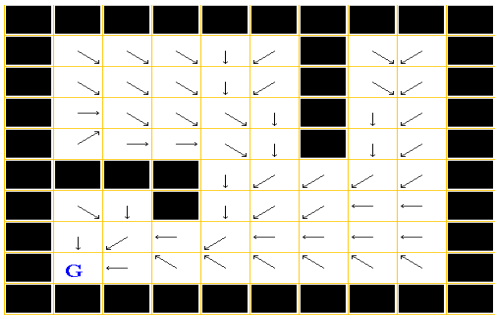

Figure 6. An optimal strategy example using parameters: α=0.2, x=$\alpha^{|S|}$ and R_b=1

Proposition 6. Let G_1 (G_2) be a goal state such that $R_b^1 = 1$ ($R_b^2 = 1 + \frac{1}{\alpha^{|S|}}$). Suppose ∃ a path of length l_1 (l_2) from initial state s_0 to the goal state G_1 (G_2), then the optimal strategy navigates robot to the goal state G_2 for all $l_1 \geq 1$.

Proof. Suppose ∃ an optimal strategy π_1^* that moves the robot to the goal state G_1. Equation (11) implies that:

$$V_{\pi_1^*}^{\alpha}(s_0) < \frac{\alpha R_b^1}{1-\alpha} = \frac{\alpha}{1-\alpha} < \frac{1}{1-\alpha} \tag{16}$$

Let π_2 be a strategy that moves robot to the goal state G_2, equation (15) and the fact that $l_2 < |S|$ imply that:

$$V_{\pi_2}^{\alpha}(s_0) \geq \frac{R_b^2 \times \alpha^l - x + x\alpha^l}{1-\alpha} > \frac{R_b^2 \times \alpha^{|S|} - x + x\alpha^{|S|}}{1-\alpha} \tag{17}$$

By using the fact that $R_b^2 = 1 + \frac{1}{\alpha^{|S|}}$ and $x = \alpha^{|S|}$, inequation (17) can be rewritten as:

$$V_{\pi_2}^{\alpha}(s_0) > \frac{\left(1+\frac{1}{\alpha^{|S|}}\right)\alpha^{|S|} - x + x\alpha^{|S|}}{1-\alpha} = \frac{1+\alpha^{2|S|}}{1-\alpha} > \frac{1}{1-\alpha} \tag{18}$$

Inequalities (16) and (18) imply that:

$$V_{\pi_2}^{\alpha}(s_0) > \frac{1+\alpha^{2|S|}}{1-\alpha} > \frac{1}{1-\alpha} > V_{\pi_1^*}^{\alpha}(s_0) \tag{19}$$

The fact that $V_{\pi_2}^{\alpha}(s_0) > V_{\pi_1^*}^{\alpha}(s_0)$ implies that π_1^* cannot be an optimal strategy.

5. GOALS TO GOALS AREA COVERAGE ALGORITHMS

In this section, we present the proposed on-line algorithm for area coverage in autonomous demining robot based on the model defined in section 3, and the theoretical studies presented in the previous section.

We begin by the first version (Algorithm 2) called Goals to Goals Randomize Area Coverage, in which we suppose that all states are goals with the same reward value, except then initial state. Beginning by the initial position, the robot detect the near states, update detected states, calculate path strategy using algorithm 1 and move according to an optimal action; these steps are repeated until the optimal action is equal to θ.

Algorithm 2. Goals to Goals Randomize Area Coverage.

Data: S, P, A, R, Γ_a^+, α, ε; $x = \alpha^{|S|}$, s_0: initial state

1. Set all states as goals (except initial state) with the same reward: $R_b = 1$.
2. **Repeat**
   - 2.1. Observe near states with sensor
   - 2.2. Update observed states
   - 2.3. Calculate strategy using algorithm 1
   - 2.4. Move robot using an optimal action

Until (the optimal action is equal to θ**)**

Theorem. The algorithm 2 works correctly and terminates after covering the entire area except the non-reachable regions from the start state s_0.

Proof. The proof follows from the propositions 1, 2, 4 and 5. In fact, propositions 1 and 2 imply that if there is at least one state not explored and reachable from the current robot position, the optimal action is different to θ. Moreover, after each observation, the status of near states (supposed goals) is updated; the navigation to the nearest goal is assured by propositions 3 and 4 and the robot stops after exploring the entire environment except the unreachable states from the start state s_0 (proposition 2).

Figure 7 shows a path generated using algorithm 2 in an obstacle free environment (ten randomized mines positions), as we can see, the entire area is covered, but the search path is pseudo-random and the robot overlaps some regions previously detected.

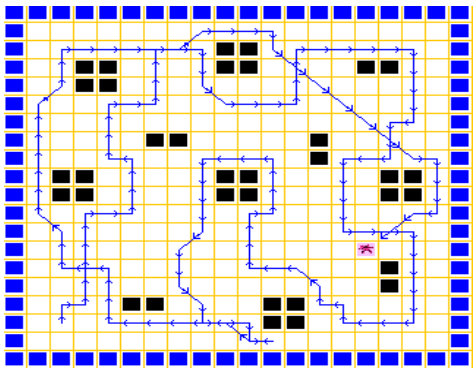

Figure 7. Environment example and path generated using algorithm 2

To minimize the overlapping paths, the authors propose a second version (algorithm 3) based on decreasing the rewards values according some Search Mode such as a line-sweep (Figure 8) based approach described in [13] or spatial cell diffusion approach presented in [14].

In each smaller region (for example in figure 8, a smaller region contain nine states) all states are considered as goals with the same fixed reward value and the search mode is assured by the proposition 6.

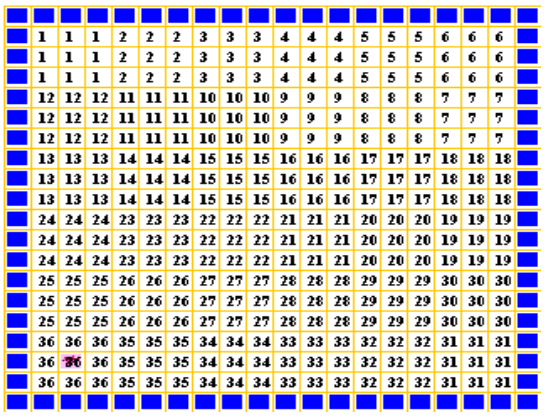

Figure 8. Example of rewards values decremented according the line-sweep search mode

Algorithm 3. Goals to Goals Search mode Area Coverage

Data: S, P, A, R, Γ_a^+, α, ε; $x = \alpha^{|S|}$, s_0: initial state
1. Decompose the state space into the p smaller regions.
2. Define the decreasing reward values according to the desired search mode.
3. **For each** smaller region $i = p,\ldots,1$ **Do** Set all states in region **i** as goals with $\boldsymbol{R_i = 1 + \frac{i}{\alpha^{|S|}}}$.
4. **Repeat**
   4.1. Observe near states
   4.2. Update observed states
   4.3. Calculate strategy using algorithm 1
   4.4. Move robot using an optimal action
   Until (the optimal action is equal to θ**)**

6. SIMULATION RESULTS

The proposed algorithms are simulated using JAVA language implementation; figures 9, 10 and 11 show the simulation results for several search modes, in an obstacle free environment example, with ten randomized buried mines. It can be seen from figures 9, 10 and 11 that the Algorithm 3 works correctly with low repetition rate compared to Algorithm 2.

In some case, it is optimal and beneficial that the robot finishes the area coverage near its initial position; the variant of Line Sweep search mode (Figure 11) can be used.

To evaluate these search modes, we can use the following valuation function:

$$E = \sum_{i=0}^{T} \frac{R_{s_t}}{x} = \sum_{i=0}^{T} C_{s_t} \quad (20)$$

where $C_{s_t} = \frac{1}{\sqrt{2}}$ or 1 and T is the number of decisions to complete the area coverage; this valuation function is proportional to the time required for the entire exploration; the number of rotation can be added to this valuation function.

Table 1. Valuation Functions: Comparison of Several Search Modes

Search mode	$\bar{E}$
Randomize	100
Spatial Cell Diffusion	83
Line Sweep	80
Variant of Line Sweep	82

Table 1 presents the average value of E for each search mode and for several irregular buried mines locations generated randomly in the same environment example, us it can be seen, these search modes in free obstacle environment achieves the area coverage with low cost of energy, especially the Line Sweep search mode which is beneficial in the decomposition method for real environment with obstacles.

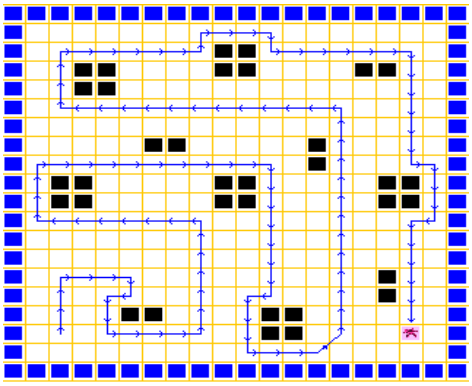

Figure 9. Online strategy generated using algorithm 3 for spatial cell diffusion search mode

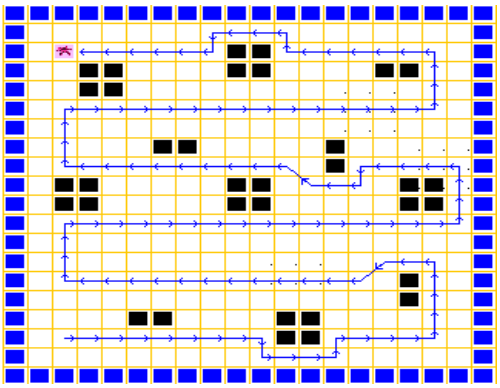

Figure 10. Online strategy generated using algorithm 3 for Line-Sweep search mode

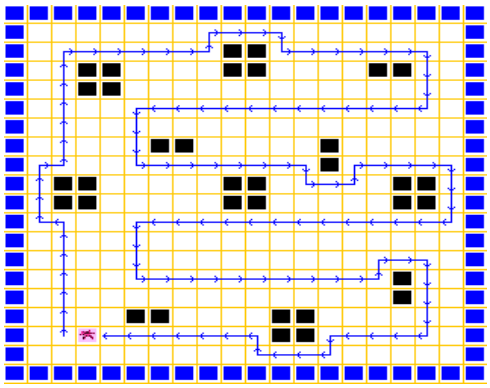

Figure 11. Online strategy generated using algorithm 3 for a variant of Line-Sweep search mode

Remark 2. Area coverage without overlapping path is not always easy and possible especialy in an unknown environment. Figure 12 shows a strategy example with some overlapping paths.

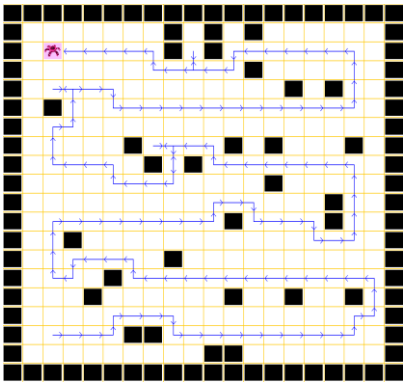

Figure 12. Online strategy example with some overlapping paths

Remark 3. For a large state space and real environment with obstacles, line sweep decomposition into monotone subregions can be used [13]; in each subregion, an adequate line-sweep search mode is used to ensure optimal area coverage. The robot explores subregions one by one; each subregion covered can be changed as goals states with reward $R_b = 0$, so that there will be no return to the explored region (Proposition 3) and so, the decision time of Algorithm 1 can be reduced to real time since it is proportional to the number of free states (Remark 1).

Figure 13 shows an environment example with obstacles divided into two subregions, after exploring the left first region, the states are transformed to goals with reward value equal to zero; the path generated is shown in Figure 14.

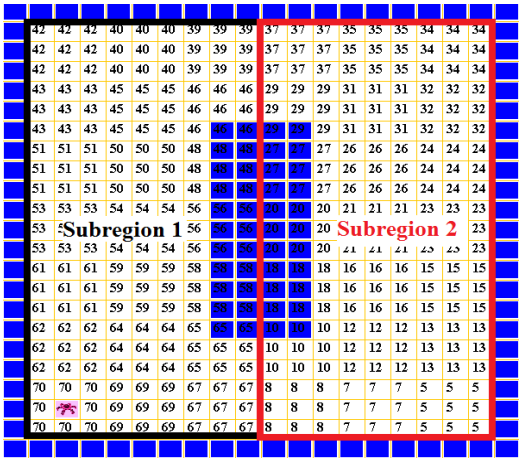

Figure 13. Environment example with obstacle (blue region) divided into two subregions

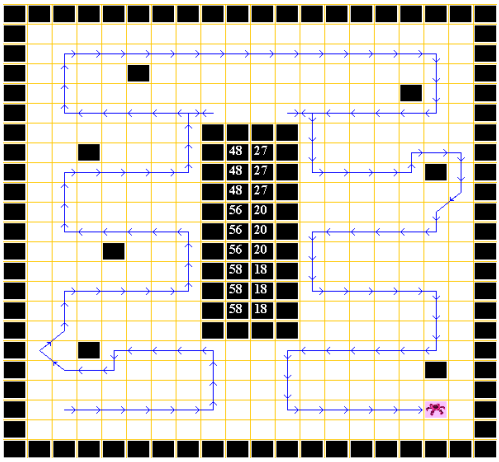

Figure 14. Online strategy generated for the environment example in figure 13

Remark 4. For minimizing the memory consumption in large state space, each subregion can be aggregated to one state after and before exploration, only the active subregion is disaggregated.

7. CONCLUSIONS

In this work, we have presented a Discounted Markov Decision Model for robotic navigation in grid's environment by adding a fictitious action in each state witch is a key of the simplest proposed algorithms, the use of this model is certainly not limited to autonomous robot landmines detection but it can be used in several robotics applications. We have also presented a theoretical study of Discounted MDP in the proposed model to ensure an optimal path strategy. This theoretical study permit us to prove the correctness of the proposed new approach to find a best on-line path strategy for detecting buried mines in a free obstacle environment where hidden mines are randomly distributed in different positions.

The simulation results show that our approach is encouraging and promising for an extension, in future work, to different types of environment and to the Partially Observable MDP where the robot position is stochastic.

REFERENCES

[1] H. Najjaran and N. Kircanski, "Path Planning for a Terrain Scanner Robot", *in Proc. 31st Int. Symp. Robotics, Montereal, QC, Canada 2000*, pp. 132-137.

[2] P. Atkar et al, "Uniform coverage of automotive surface patches", *The International Journal of Robotics Research*, 24(11), pp.883-898, 2005.

[3] J. S. Oh et al, "Complete Coverage Navigation of Cleaning Robots using Triangular-Cell-Based Map", *IEEE Transactions on Industrial Electronics*, vol. 51, no. 3, pp. 718-726, 2004.

[4] P. Tokekar; N. Karnad; V. Isler, "Energy-Optimal Trajectory Planning for Car-Like Robots," *Autonomous Robots*, vol. 37, no.3, pp.279-300, 2014.

[5] J. W. Kang; S. J. Kim; M. J. Chung, "Path Planning for Complete and Efficient Coverage Operation of Mobile Robots", *IEEE International Conference on Mechatronics and Automation*, pp. 2126-2131, 2007.

[6] E. Galceran; M. Carreras, "A Survey on Coverage Path Planning for Robotics", *Robotics and Autonomous Systems*, 61, pp. 1258-1276, 2013.

[7] R. E. Bellman. "Dynamic Programming. Princeton University Press", Princeton, NJ, 1957.

[8] M. Puterman, "Markov Decision Processes: Discrete Stochastic Dynamic Programming", John Wiley & Sons, Inc., New York, USA, 1994.

[9] D. J. White, "Markov Decision Processes" John Wiley & Sons, Inc, New York, 1994.

[10] D. P. Bertsekas, "Dynamic Programming and Optimal Control," Belmont: Athena Scientific, 2001.

[11] C. P. Goonerante; S. C. Mukhopahyay; G. Sen Gupta, "A Review of Sensing Technologies for Landmine Detection: Unmanned Vehicle Based Approach", *in: Proc. 2nd International Conference on Autonomous Robots and Agents December 13-15 2004 Palmerston North, New Zealand*.

[12] K. Suresh; K. Vidyasagar; A. F. Basha**,** "Multi Directional Conductive Metal Detection Robot Control", *International Journal of Computer Applications*, volume 109 no.4, 2015.

[13] W. H. Huang, "Optimal Line-Sweep-Based Decompositions for Coverage Algorithms", *in Robotics and Automation, 2001. Proceedings 2001 ICRA. IEEE International Conference on*, vol. 1. IEEE, 2001, pp. 27–32.

[14] S. W. Ryu et al, "A Search and Coverage Algorithm for Mobile Robot", *in Ubiquitous Robots and Ambient Intelligence (URAI), 2011 8th International Conference on. IEEE*, pp. 815–821, 2011.