Patrolling in A Stochastic Environment

Student Paper Submission
(Suggested Track: Modeling and Simulation)

Sui Ruan [1] (Student)
E-mail: sruan@engr.uconn.edu

Candra Meirina [1] (Student)
E-mail: meirina@engr.uconn.edu

Feili Yu [1] (Student)
E-mail: yu02001@engr.uconn.edu

Krishna R. Pattipati [13]
University of Connecticut, Dept. of Electrical and Computer Engineering
371 Fairfield Road, Unit 1157
Storrs, CT 06269-1157
Fax: 860-486-5585
Phone: 860-486-2890
E-mail: krishna@engr.uconn.edu

Robert L. Popp [2]
Information Exploitation Office, DARPA
3701 N. Fairfax Drive, Arlington, VA 22203-1714
Phone: (703)248-1520
E-mail: rpopp@darpa.mil

* This work is supported by the Office of Naval Research under contract #N00014-00-1-0101
[1] Electrical and Computer Engineering Department, University of Connecticut, Storrs, CT 06269-1157, USA.
[2] Information Exploitation Office, DARPA, 3701 N. Fairfax Drive, Arlington, VA 22203, USA
[3] Correspondence: krishna@engr.uconn.edu

Patrolling in a Stochastic Environment

Sui Ruan[a], Candra Meirina[a], Feili Yu[a], Krishna Pattipati[a] and Robert L. Popp [b]

Abstract

The patrolling problem considered in this paper has the following characteristics: Patrol units conduct preventive patrolling and respond to call-for-service. The patrol locations (nodes) have different priorities, and varying incident rates. We design a patrolling scheme such that the locations are visited based on their importance and incident rates. The solution is accomplished in two steps. First, we partition the set of nodes of interest into subsets of nodes, called sectors. Each sector is assigned to one patrol unit. Second, for each sector, we exploit a response strategy of preemptive call-for-service response, and design multiple sub-optimal off-line patrol routes. The net effect of randomized patrol routes with immediate call-for-service response would allow the limited patrol resources to provide prompt response to random requests, while effectively covering the nodes of different priorities having varying incidence rates. To obtain multiple routes, we design a novel learning algorithm (Similar State Estimate Update) under a Markov Decision Process (MDP) framework, and apply softmax action selection method. The resulting patrol routes and patrol unit visibility would appear unpredictable to the insurgents and criminals, thus creating the impression of virtual police presence and potentially mitigating large scale incidents.

I. Introduction

In a highly dynamic and volatile environment, such as a post-conflict stability operation or a troubled neighborhood, military and/or police units conduct surveillance via preventive patrolling, together with other peace keeping or crime prevention activities. Preventive patrol constitutes touring an area, with the patrol units scanning for threats, attempting to prevent incidents, and intercepting any threats in progress. Effective patrolling can prevent small scale events from cascading into large scale incidents, and can enhance civilian security. Consequently, it is a major component of stability operations and crime prevention. In crime control, for example, for the greatest number of civilians, deterrence through ever-present police patrol, coupled with the prospect of speedy police action once a report is received, appears crucial in that the presence or potential presence of police officers on patrol severely inhibits criminal activity[1]. Due to limited patrolling resources(e.g., manpower, vehicles, sensing and shaping resources), optimal resource allocation and planning of patrol effort are critical to effective stability operations and crime prevention[2].

The paper is organized as follows: In section II, the stochastic patrolling problem is modeled. In section III, we propose a solution approach based on a MDP framework. Simulation results are presented in section IV. In section V, the paper concludes with a summary and future research directions.

II. Stochastic Patrolling Model

The patrolling problem is modeled as follows:

- A finite set of nodes of interest: $\aleph = \{i;\ \ i = 1, .., I\}$. Each node $i \in \aleph$ has the following attributes:

[a]Electrical and Computer Engineering Department, University of Connecticut, Storrs, CT 06269-1157, USA. E-mail: [sruan, meirina, yu02001, krishna]@engr.uconn.edu, [b]Information Exploitation Office, DARPA, 3701 N. Fairfax Drive, Arlington, VA22203, USA. Email: rpopp@darpa.mil. This work is supported by the Office of Naval Research under contract No. 00014-00-1-0101.

  - fixed location (x_i, y_i);
  - incident rate $\lambda_i(1/hour)$: we assume that the number of incident occurring at node i in a time interval (t_1, t_2), denoted by $\mathbf{n_i}(t_2, t_1)$, is a Poisson random variable with parameter $\lambda_i(t_2 - t_1)$:
$$P(\mathbf{n_i}(t_2, t_1) = k) = \frac{e^{-\lambda_i(t_2-t_1)}(\lambda_i(t_2 - t_1))^k}{k!}$$
  - importance index δ_i: a value indicating the relative importance of node i in the patrolling area.
- The connectivity of the nodes: for any node j directly connected to node i, we denote it as $j \in adj(i)$, and the length of the edge connecting them as $e(i, j)$;
- A finite set of identical patrol units, each with average speed v, i.e., the estimated time for a unit, $\overline{t}$, to cover a distance d, is $\overline{t} = \frac{d}{v}$. Each unit would respond to a call-for-service immediately when a request is received; otherwise, the patrol unit traverses along prescribed routes.

  In this paper, we focus our attention on the problem of routing for effective patrolling, and assume that whenever a patrol unit visits a node, the unit can clear all incidents on that node immediately. Some real world constraints, such as the resources required and incident clearing times are not considered; future work would address these extensions.

III. Proposed Solution

Our solution to the patrolling problem consists of two steps. First, we partition the set of nodes of interest (corresponding to a city for example) into subsets of nodes called sectors. Each sector is assigned to one patrol unit. Second, for each sector, we exploit a response strategy of preemptive call-for-service response, and design multiple off-line patrol routes. The patrol unit randomly selects predefined routes to conduct preventive patrolling; whenever a call-for-service request is received, the patrol unit would stop the current patrol and respond to the request immediately; after completing the call-for-service, the patrol unit would resume the suspended patrol route. The net effect of randomized patrol routes with immediate call-for-service response would allow limited patrol resources to provide prompt response to random requests, while effectively covering the nodes of different priorities having varying incidence rates.

The sector partitioning sub-problem is formulated as a combinatorial optimization problem, and solved via political districting algorithms presented in [5]. The off-line route planning subproblem for each sector is formulated as an infinite-horizon Markov Decision Process (MDP)[4], based on which a novel learning method, viz., Similar State Estimate Update, is applied. Furthermore, we apply Softmax action selection method[8] to prescribe multiple patrol routes to create the impression of virtual patrol presence and unpredictability.

A. Area partitioning for patrol unit assignment

The problem of partitioning a patrol area can be formulated as follows:

- A region is composed of a finite set of nodes of interest: $\aleph = \{i;\ \ i = 1, .., I\}$. Each node $i \in \aleph$ is centered at position (x_i, y_i), and value $\varphi_i = \lambda_i \delta_i$;

- There are r areas to cover the region, such that all nodes are covered, with minimum over lap, and the sum of values for each area is similar, and areas are compact.

This is a typical political districting problem. Dividing a region, such as a state, into small areas, termed districts. to elect political representatives is called political districting[6]. A region consists of I population units such as counties (or census tracks), and the population units must be grouped together to form r districts. Due to court rulings and regulations, the deviation of the population per district cannot exceed a certain proportion of the average population. In addition, each district must be contiguous and compact. A district is contiguous, if it is possible to reach any two places of the district without crossing another district. Compactness essentially means that the district is somewhat circular or a square in shape rather than a long and thin strip. Such shapes reduce the distance of the population units to the center of the district or between two population centers of a district. This problem was extensively studied in [5], [6].

B. Optimal Routing in a Sector

1) MDP modeling: In a sector, there are n nodes of interest, $N = \{1, ..., n\} \subseteq \aleph$. A Markov Decision Process (MDP) representation of the patrolling problem is as follows:

- ◇ Decision epochs are discretized such that each decision epoch begins at the time instant when the patrol unit finishes checking on a node, and needs to move to a next node; the epoch ends at the time instant when the patrol unit reaches the next node, and clears all incidents at that node.
- ◇ States $\{s\}$: a state, defined at the beginning of decision epoch t, is denoted as $s = \{i, \underline{\mathbf{w}}\}$, where $i \in N$ is the node the patrol unit is currently located at, and $\underline{\mathbf{w}} = \{w_j\}_{j=1}^{n}$ denotes the times elapsed since the nodes are last visited;
- ◇ Actions $\{a\}$: an action, also defined at the beginning of decision epoch t, is denoted as $a = (i, j)$, where i is the patrol unit's current location, and $j \in adj(i)$, an adjacent node of i, denotes the next node to be visited;
- ◇ State transition probabilities $P(s'|s, a)$: given state s, and action a, the probability of s' being the next state;
- ◇ Reward $g(s, a, s')$: the reward for taking action $a = (i, j)$ at state $s = (i, \underline{\mathbf{w}})$ to reach next state $s' = (j, \underline{\mathbf{w}}')$. At time t', the patrol unit reaches node j and clears $\mathbf{n}_j(t')$ incidents, and earns the reward at time t' of $g(s, a, s') = \delta_j \mathbf{n}_j(t')$.
- ◇ Discount mechanism: the reward g potentially earned at future time t' is valued as $ge^{-\beta(t'-t)}$ at current time t, where β is the discount rate;
- ◇ Objective is to determine an optimal policy, i.e., a mapping from states to actions, such that the overall expected reward is maximized.

The value function (expected reward) of a state, s at time t_0, for policy $\mathbf{\Pi}$ (a mapping from state to action) is defined as:

$$V^{\Pi}(s) = E[\sum_{k=0}^{\infty} g_{k+1} e^{-\beta(t_{k+1}-t_0)}], \tag{1}$$

where g_{k+1} is the reward earned at time t_{k+1}. Note that $V^{\Pi}(s)$ is independent of time, t, i.e., a constant state-dependent stationary value corresponding to a stationary policy.

Dynamic Programming [4][7] and Reinforcement Learning [8] can be employed to solve the MDP problem. In this work, we first prove that under any deterministic policy $\mathbf{\Pi}$, the structure of value function (V^{Π}) of a state, $s = (i, \underline{\mathbf{w}})$, is a linear function: $V^{\Pi}(s = (i, \underline{\mathbf{w}})) = (\underline{c}_i^{\Pi}(s))^T \underline{\mathbf{w}} + d_i^{\Pi}(s)$. Therefore, the optimal policy satisfies $V^*(s = (i, \underline{\mathbf{w}})) = (\underline{c}_i^*(s))^T \underline{\mathbf{w}} + d_i^*(s)$. Here, we denote $\mathbf{c}^{\Pi}(s)$, $\underline{d}^{\Pi}(s)$ as the parameters for policy $\mathbf{\Pi}$, while $\mathbf{c}^*(s)$, $\underline{d}^*(s)$ are the concomitant parameters for the optimal policy $\mathbf{\Pi}^*$. Based on this structure, we construct the linear function as an approximation of optimal value function, denoted as: $\widetilde{V}^*(s = (i, \underline{\mathbf{w}})) = (\underline{c}_i^*)^T \underline{\mathbf{w}} + d_i^*$, where $\underline{c}_i^*$ and d_i^* are constants independent of $\{\underline{\mathbf{w}}\}$. This special structure of the value function enables us to design a novel learning algorithm, the so-called **S**imilar **S**tate **E**stimate **U**pdate($SSEU$) to obtain a deterministic near-optimal policy, from which a near-optimal patrolling route can be obtained. The $SSEU$ algorithm employs the ideas from Monte-Carlo and Temporal Difference (specifically, $TD(0)$) methods[8]), while overcoming the inefficiencies of these methods on the patrolling problem.

At state $s = \{i, \{\underline{\mathbf{w}}\}\}$, when action $a = (i, j)$ is undertaken, the state transverses to $s' = \{j, \{\underline{\mathbf{w}}'\}\}$. Note that under our modeling assumption, the state transition by action a is deterministic, while the reward accrued by action a at state s is stochastic in the sense that the number of incidents at node j is random. Therefore, the Bellman's equation for the patrolling problem can be simplified as:

$$\begin{aligned} V^*(s) &= \max_a E[e^{-\beta \frac{e(i,j)}{v}} g(s,a,s') + e^{-\beta \frac{e(i,j)}{v}} V^*(s') | s, a] \\ &= \max_a \alpha(s,s')\{E[g(s,a,s')] + V^*(s')\} \end{aligned} \tag{2}$$

Here $g(s, a, s')$ is the reward for taking action $a = (i, j)$ at state $s = (i, \underline{\mathbf{w}})$ to reach state $s' = (j, \underline{\mathbf{w}}')$. The expected reward is $E[g(s, a, s')] = \delta_j \lambda_j [w_j + \frac{e_{ij}}{v}]$, and $\alpha(s, s') = e^{-\beta \frac{e(i,j)}{v}}$ accounts for discount factor for state transition from s to s'.

The greatest challenge in using MDPs as the basis for decision making lies in discovering computationally feasible methods for the construction of optimal, approximately optimal or satisfactory policies[7]. Arbitrary MDP problems are intractable; producing even satisfactory or approximately optimal policies is generally infeasible. However, many realistic application domains exhibit considerable structure and this structure can be exploited to obtain efficient solutions. Our patrolling problem falls into this category.

Theorem 1: For any deterministic policy in the patrolling problem, i.e., $\mathbf{\Pi} : s \longrightarrow a$, $\forall s \in \mathbf{S}, \forall a \in \mathbf{A}(s)$, the state value function has the following property:

$$V^{\Pi}(s = (i, \underline{\mathbf{w}})) = (\underline{c}_i^{\Pi}(s))^T \underline{\mathbf{w}} + d_i^{\Pi}(s) \quad \forall i \in N \tag{3}$$

Proof: Under any deterministic policy, $\mathbf{\Pi}$, for an arbitrary state $s = (i, \underline{\mathbf{w}})$ at t, the follow-on state trajectory is deterministic as the state transition is deterministic in the patrolling problem. We denote the state trajectory in a format "$node(time,\ reward)$" as:

$$i_0(=i)(t,0) \longrightarrow i_1(t+T_1,r_1)... \longrightarrow i_N(t+T_N,r_N) \longrightarrow ... \tag{4}$$

Thus, the value function of state s under policy $\mathbf{\Pi}$ is

$$V^{\Pi}(s=(i,(\underline{\mathbf{w}})) \;\; = \;\; E[\sum_{k=0} r_k e^{-\beta T_k}] = \sum_j f_{ij} \tag{5}$$

where r_k is the reward earned at decision epoch t_k and f_{ij} signifies its expected sum of rewards earned at node j. Since the sequence of visits to node j is:

$$j(t+T_{j,1},r_{j,1}) \longrightarrow ... \longrightarrow j(t+T_{j,2},r_{j,2}) \longrightarrow ... \longrightarrow j(t+T_{j,N},r_{j,N}),... \tag{6}$$

and expected reward of first visit to node j following state s is: $E(r_{j,1}) = \delta_j \lambda_j (w_j + T_{j,1}) e^{-\beta T_{j,1}}$, and k^{th} $(k>1)$ visit to node j is $E(r_{j,k}) = \delta_j \lambda_j (T_{j,k} - T_{j,k-1}) e^{-\beta T_{j,k-1}}$. Therefore, we have

$$\begin{aligned} f_{ij} \;\; &= \;\; \delta_j \lambda_j [w_j + T_{j,1}] e^{-\beta T_{j,1}} + \delta_j \lambda_j [T_{j,2} - T_{j,1}] e^{-\beta T_{j,2}} + ... \delta_j \lambda_j [T_{j,N} - T_{j,N-1}] e^{-\beta T_{j,N}} ... \\ &= \;\; c_{ij} w_j + d_{ij}. \end{aligned} \tag{7}$$

Here, $c_{ij} = \delta_j \lambda_j e^{-\beta T_{j,1}}$, and $d_{ij} = \sum_{k=1}^{\infty} \delta_j \lambda_j [T_{j,k} - T_{j,k-1}] e^{-\beta T_{j,k}}$. Since $T_{j,k} - T_{j,k-1}$, $(k = 1, ..., \infty)$ are dependent on policy Π and state s, we have $V^{\Pi}(s=(i,\underline{\mathbf{w}})) = (\underline{c}_i^{\Pi}(s))^T \underline{\mathbf{w}} + d_i(s)$. ■

Based on this observation, we employ linear function approximation for $V*(s)$ as follows:

$$V^*(s=(i,\underline{\mathbf{w}})) = \widetilde{V}^*(s=(i,\underline{\mathbf{w}})) \approx (\underline{c}_i^*)^T \underline{\mathbf{w}} + d_i^*; \;\; \forall i \in N \tag{8}$$

where $\underline{c}_i^* = \{c_{ij}\}_{j=1}^n$, c_{ij}^* is the expected value of $\delta_j \lambda_j e^{-\beta T_{j,1}}$, $j = 1, ..., n$ under optimal policy Π^*; d_i^* is the expected value of $\sum_{j=1}^n \sum_{k=1}^{\infty} \delta_j \lambda_j [T_{j,k} - T_{j,k-1}] e^{-\beta T_{j,k}}$ under optimal policy Π^*.

Starting from an arbitrary policy, we could employ the following value and policy iteration method[8] to evaluate and improve the policies iteratively to gradually approach an optimal policy,

$$V^{t+1} = \max_{\forall a=(i,j),\;\; j \in adj(i)} \alpha(s,s') \{E[g(s,a=(i,j),s')] + V^t(s')\}. \tag{9}$$

$$a^{t+1} = \arg \max_{\forall a=(i,j),\;\; j \in adj(i)} \alpha(s,s') \{E[g(s,a=(i,j),s')] + V^t(s')\}. \tag{10}$$

2) Similar State Estimate Update Method (Learning Algorithm): We seek to obtain estimates r^* of optimal policy, where $r^* = (\mathbf{c}, \underline{d})^*$, by minimizing the Mean-Squared-Error as:

$$\min_r MSE(r) = \min_r \sum_{s \in \mathbf{S}} (V^*(s) - \widetilde{V}(s,r))^2, \tag{11}$$

where $V^*(s)$ is the true value at state s under optimal policy, $\widetilde{V}(s,r)$ is the linear approximation as defined in Eq(3).

At iteration step t, we observe a new example $s_t \mapsto V^t(s_t)$. Stochastic gradient-descent methods adjust the parameter vector by a small amount in the direction that would most reduce the error on that example:

$$r^{t+1} = r^t + \gamma_t [V^t(s_t) - \widetilde{V}(s_t, r^t)] \nabla \widetilde{V}(s_t, r^t) \tag{12}$$

Here ∇ is the gradient operator with respect to r^t, and γ_t is a positive step-size parameter. Stochastic approximation theory [3] requires that $\sum_{k=1}^{\infty}\gamma_k = \infty$ and $\sum_{k=1}^{\infty}\gamma_k^2 < \infty$.

There are two classes of simulation-based learning methods to obtain r^*, viz., Monte-Carlo and Temporal-Difference learning methods[8]. These methods require only experience - samples of sequences of states, actions, and rewards from on-line or simulated interaction with environment. Learning from simulated experience is powerful in that it requires no a priori knowledge of the environment's dynamics, and yet can still attain optimal behavior. Monte-Carlo methods are ways of solving the reinforcement learning problem based on averaging the sample returns. In Monte Carlo methods, experiences are divided into episodes, and it is only upon the completion of an episode that value estimates and policies are changed. Monte-Carlo methods are thus incremental in an episode-by-episode sense. In contrast, Temporal Difference methods update estimates based in part on other learned estimates, without waiting for a final outcome[8].

Monte-Carlo method, as applied to the patrolling problem, works as follows: based on current estimated r^t, run one pseudo-episode (sufficiently long state trajectory); gather the observations of rewards of all states along the trajectory; apply the stochastic gradient descent method as in Eq(12) to obtain r^{t+1}. Then, repeat the process until converged estimates (r^*) are obtained. A disadvantage of Monte-Carlo method here is that, for infinite MDP, to make the return, $V^t(s_t)$, accurate for each state, the episode has to be sufficiently long; this would result in large memory requirement and a long learning cycle.

Temporal Difference, $TD(0)$ method, as applied to the patrolling problem works as follows: simulate one state transition with r^t; then immediately update estimates to be r^{t+1}. Define d_t as the return difference due to transition from state s to s':

$$d_t = \alpha(s,s')[g(s,a,s') + \widetilde{V}(s',r^t)] - \widetilde{V}(s,r^t) \tag{13}$$

where $\alpha(s,s')$ is the discount factor for state transition from s to s'. The $TD(0)$ learning method updates estimates r^{t+1} according to the formula

$$r^{t+1} = r^t + \gamma_t d_t \nabla\widetilde{V}(s,r^t) \tag{14}$$

A disadvantage of $TD(0)$ as applied to the patrolling problem is the following. Since adjacent states are always from different nodes, r_j^t ($r_j = (\mathbf{c}_j, \underline{d}_j)$) is used to update r_i^{t+1} ($i \neq j$); this could result in slow convergence or even divergence.

To overcome the disadvantages of Monte-Carlo and $TD(0)$ methods, while exploiting their strengths in value learning, we design a new learning method, termed the **S**imilar **S**tate **E**stimate **U**pdate ($SSEU$). We define states where the patrol unit is located at the same node as being similar, e.g., $s_1 = (i, \underline{\mathbf{w}}_1)$ and $s_2 = (i, \underline{\mathbf{w}}_2)$ are similar states. Suppose that the generated trajectory under current estimation ($\mathbf{c}^t$ and d^t) for two adjacent similar states of node i, i.e., state $s = (i, \underline{\mathbf{w}}^t)$ and $s' = (i, \underline{\mathbf{w}}^{t_N})$ is: $i_0(=i)(t,\ 0)$, $i_1(t_1,\ g_1)$, $i_2(t_2\ g_2)$, ..., $i_N(=i)(t_N,\ g_N)$. Based on this sub-trajectory, we obtain the new observations of C_{ij}^{new}, for nodes $j = i_1, i_2, ..., i_N$ as follows:

$$c_{ij}^{new} = \delta_j \lambda_j exp^{-\beta(t_j^1 - t)}, \tag{15}$$

and the new observations of d_i^{new}:

$$d_i^{new} = \sum_{k=1}^{N} g_k e^{-\beta(t_k - t)} + V^t(s')e^{-\beta(t_N - t)} - \sum_{j=i_1}^{i_N} c_{ij}^{new} w_j \tag{16}$$

Consequently, the parameters c_{ij} and d_i are updated by:

$$c_{ij}^{t+1} = c_{ij}^t + \frac{c_{ij}^{new} - c_{ij}^t}{N_{ij}^c} \quad d_i^{t+1} = d_i^t + \frac{d_i^{new} - d_i^t}{N_i^d} \tag{17}$$

where N_{ij}^c is the number of update of c_{ij}, and N_i^d is the number of update of d_i.

To make our learning algorithm effective, there are two other issues to consider. First, to avoid the possibility that some nodes are much less frequently visited than others, we apply $exploring - starts$ rule, where we intentionally begin episodes from those nodes that are less frequently visited based on the simulation histories. Second, to escape from local minima, we employ the ϵ-greedy method. The simplest action selection rule is to select the action with highest estimated action value as in Eq(10). This method always exploits current knowledge to maximize immediate reward, and it spends no time at all sampling apparently inferior actions to verify whether they might be profitable in the long term. In contrast, ϵ-greedy behaves greedily most of the time, but every once in a while, with a small probability ϵ, selects an action at random, uniformly, and independently of the action-value estimates. In ϵ-greedy, as in Eq(18), all non-greedy actions are given the minimal probability of selection, $\frac{\epsilon}{|A(s)|}$, and the remaining bulk of the probability, $1 - \epsilon + \frac{\epsilon}{|A(s)|}$, is given to the greedy action [8], where $|A(s)|$ is the cardinality of action set, $A(s)$ in state s. This enables the learning method to get out of local minima, and thus provides the balance between exploitation and exploration.

The details of Similar State Update learning algorithm can be found in Fig.1. The $\mathbf{c}^*$ and $\underline{d}^*$ obtained by this method can provide a near-optimal patrol route by concatenating greedy actions for each state, as described in Eq(10).

C. Strategy for Generating Multiple Patrolling Routes

In this section, we design a method for generating multiple satisfactory routes by Softmax action selection strategy. In order to impart virtual presence and unpredictability to patrolling, the unit needs multiple and randomized patrol routes. We employ Softmax action selection method[8], where the greedy action is still given the highest selection probability, but all the others are ranked and weighed according to their value estimates. The most common softmax method uses a Gibbs distribution. It chooses action a at state s with probability:

$$\frac{e^{[Q^*(s,a) - Q^*]/\tau}}{\sum_{a' \in A(s)} e^{[Q^*(s,a') - Q^*]/\tau}}, \quad \text{where } \; Q^* = \max_a Q^*(s,a); \tag{19}$$

where $A(s)$ denotes the set of feasible actions at state s, and $Q^*(s,a)$ is action-value function for optimal policy Π^*,

$$Q^*(s,a) = \alpha(s,s')\{E[g(s,a,s')] + V^*(s')\}. \tag{20}$$

Learning Algorithm: Similar State Estimate Update
(With $Exploring-Starts$ and ϵ-greedy rules)

Initialize:
$\mathbf{c}=\mathbf{0}$, $\underline{d}=\mathbf{0}$, $Frequencies=\underline{\mathbf{0}}$
Repeat

- **Step 0 (Episode Initialization):** beginning with an empty episode ρ, pick up a node $i_0=\arg\min Frequencies$ and initialize $\underline{\mathbf{w}}=\underline{0}$, append the state $s=(i_0,\underline{\mathbf{w_0}})$ to ρ.
  Set $t=0$,
  $Frequencies(i_0)++$;
- **Step 1 (Parameters Update):** Get the last node of episode, i.e., $s'=(i,\underline{\mathbf{w}}')$, find the latest similar state of s' in ρ, i.e., $s=(i,\underline{\mathbf{w}})$, if no such node, go to step 2;
  else obtain the sub-trajectory beginning at s and ending at s', update $\underline{c}_i^{t+1}$ and d_i^{t+1} as in Eq.(17), then go to step 2;
- **Step 2 (Policy Improvement):** Decide the action for state s:

$$j=\begin{cases}\max_{k\in adj(i)}\alpha(s,s')\{E[g(s,a=(i,k),s')]+V^t(s')\} & \text{w.p. } 1-\epsilon,\\ rand(adj(i)) & \text{w.p. } \epsilon,\end{cases} \tag{18}$$

  set $\Delta_t=\frac{e(i,j)}{v}$;
  calculate $\underline{\mathbf{w}}'=\underline{\mathbf{w}}+\Delta_t,\quad \underline{\mathbf{w}}'_j=0$;
  update $t=t+\Delta_t$;
  append state $s'=(j,\underline{\mathbf{w}}')$ to episode ρ;
  $Frequencies(j)++$;
  if ρ is sufficiently long, go to step 0; else go to step 1.

until $\mathbf{c}$ and $\underline{d}$ converge.

Fig. 1. Similar State Estimate Update (Learning Algorithm)

Here, τ is a positive parameter called temperature. High temperatures cause the actions to be nearly equiprobable. Low temperatures cause a greater difference in selection probability for actions that differ in their value estimates. In the limit as $\tau\longrightarrow 0$, softmax action selection reverts to a greedy action selection.

IV. Simulation and Results

We illustrate our approach to patrol routing using a simple example that represents a small county, as in Fig. 2. The nodes, incident rates (λ_i) and importance indices (δ_i) are given Table I.

The results for patrolling strategies from the similar state estimate update ($SSEU$) method and the one-step greedy strategy are compared in Table II. In the one-step greedy strategy, at each state, the neighboring node which results in the best instant reward is chosen as the next node, i.e., $j=\arg\max_{\forall k\in adj(i)}\alpha(s,s')\{E[g(s,a=(i,k),s')]\}$. If this patrol area is covered by one patrol unit, the expected overall reward of the unit following the route obtained by the $SSEU$ method is $2,330$ and the reward per unit distance is 17.4; while following the route from one-step greedy strategy, the expected overall reward is $1,474$, and the expected reward per unit distance is 6.00. If this patrol area is divided into two sectors, i.e., sector a and sector b, as in Fig. 2, the $SSEU$ method results in the following rewards: for sector a, the overall expected reward is $1,710$ and the expected reward per unit distance is

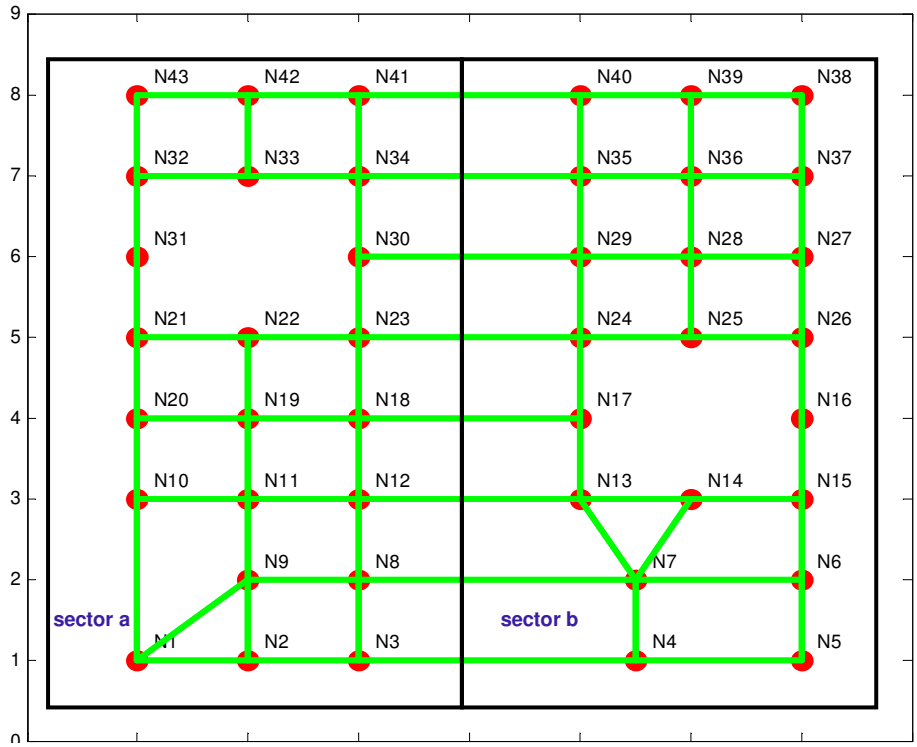

Fig. 2. Illustrative Example of Patrolling

TABLE I
EXAMPLE DESCRIPTION

node	λ_i	δ_i	node	λ_i	δ_i	node	λ_i	δ_i	node	λ_i	δ_i
N1	2	2	N12	2	2	N23	2	2	N34	8	4
N2	2	2	N13	2	2	N24	2	2	N35	2	2
N3	2	2	N14	2	2	N25	2	2	N36	2	1
N4	2	2	N15	2	2	N26	2	4	N37	6	4
N5	2	2	N16	2	2	N27	1	2	N38	2	1
N6	2	2	N17	3	4	N28	2	2	N39	2	2
N7	2	2	N18	2	2	N29	2	2	N40	4	6
N8	4	2	N19	1	2	N30	2	2	N41	2	2
N9	2	2	N20	4	10	N31	2	2	N42	2	2
N10	1	2	N21	2	1	N32	4	2	N43	2	2
N11	1	2	N22	1	2	N33	2	2			
Velocity of patrol (v): 1 unit distance/ unit time						discount rate (β): 0.1/unit time					

19.43; for sector b, the overall expected reward is $1,471$ and the expected reward per unit distance is 13.8. The one-step greedy strategy results in the following rewards: for sector a, the expected overall reward is $1,107$, and the expected reward per unit distance is 10.9; for sector b, the expected overall reward is $1,238$, and the expected reward per unit distance is 8.94. Thus, patrol routes obtained by the $SSEU$ method are highly efficient compared to the short-sighted one-step greedy strategy in this example. In this scenario, the nodes with high incident rates and importance indices are spread out and sparse. Typically, the $SSEU$ method is effective for general configurations of patrol area. Another observation from the simulation is that the net reward from sector a and sector b, i.e., 3,181, with two patrolling units, is 36% better than the net reward $(2,330)$ when there is only one patrol unit.

Furthermore when a unit patrols on a smaller area, higher overall reward per area and higher reward per unit distance are expected. After applying softmax action selection method on the near-optimal strategy from $SSEU$ method on sector a, we obtained multiple sub-optimal routes for this sector; four of them are listed in Table III.

TABLE II
PATROLLING ROUTES UNDER DIFFERENT STRATEGIES

Strategy	Patrol Route	Expected Reward	Reward /distance
SSEU (whole county)	1, 10, 20, 21, 31, 32, 33, 34, 41, 40, 39, 36, 37, 27, 26, 25, 28, 29, 30, 34, 33, 32, 31, 21, 20, 19, 18, 17, 24, 29, 35, 40, 39, 38, 37, 36, 35, 34, 33, 32, 31, 21, 20, 10, 1, 2, 3, 8, 12, 13, 17, 24, 23, 30, 34, 41, 40, 39, 36, 37, 27, 26, 16, 15, 6, 5, 4, 7, 8, 9, 11, 10, 20, 21, 31, 32, 43, 42, 41, 34, 35, 40, 39, 38, 37, 36, 28, 25, 24, 17, 18, 19, 20, 21, 22, 23, 30, 34, 33, 32, 43, 42, 41, 40, 39, 36, 37, 27, 26, 16, 15, 14, 13, 17, 18, 19, 20, 10, (1)	2,330	17.4
One-step Greedy (whole county)	1, 9, 8, 7, 8, 3, 2, 1, 9, 8, 12, 18, 17, 13, 14, 15, 6, 5, 4, 7, 8, 3, 2, 1, 9, 11, 12, 18, 17, 24, 25, 26, 16, 15, 14, 13, 17, 18, 23, 30, 34, 33, 32, 31, 21, 20, 10, 1, 2, 3, 8, 7, 6, 5, 4, 3, 8, 9, 11, 12, 13, 17, 24, 29, 28, 25, 26, 16, 15, 14, 7, 8, 3, 2, 1, 9, 8, 12, 18, 17, 13, 14, 15, 6, 5, 4, 7, 8, 3, 2, 1, 10,20, 19, 22, 23, 30, 34, 41, 40, 35, 36, 37, 27, 26, 25, 24, 17, 18, 12, 8, 9, 11, 10, 20, 21, 31, 32, 43, 42, 33, 34, 30, 29, 28, 25, 26, 16, 15, 14, 13, 17, 24, 23, 18, 12, 8, 3, 2, 9, 11, 19, 20, 10, 20, 21, 31, 32, 43, 42, 41, 40, 39, 38, 37, 36, 35, 34, 33, 32, 31, 21, 20, 19, 22, 23, 30, 34, 41, 40, 39, 40, 35, 34, 33, 32, 43, 42, 41, 40, 39, 38, 37, 27, 26, 25, 28, 29, 24, 17, 13, 7, 4, 5, 6, 15, 16, 26, 25, 28, 36, 37, 38, 39, 40, 35, 34, 30, 23, 18, 12, 8, 3, 2, (1)	1,474	6.00
$SSEU$ (sector a)	1, 10, 20, 21, 31, 32, 33, 34, 30, 23, 18, 19, 20, 21, 31, 32, 33, 34, 41, 42, 43, 32, 31, 21, 20, 10, 11, 9, 8, 12, 18, 23, 30, 34, 33, 32, 31, 21, 20, 19, 18, 23, 30, 34, 41, 42, 43, 32, 31, 21, 20, 10, 11, 12, 8, 3, 2, 9, 8, 12, 18, 23, 30, 34, 33, 32, 31, 21, 20, 19, 22, 23, 30, 34, 41, 42, 43, 32, 31, 21, 20, 10, 11, 12, 8, 9, (1)	1,710	19.43
one-step greedy (sector a)	1, 9, 8, 3, 2, 9, 8, 12, 18, 23, 30, 34, 33, 32, 31, 21, 20, 10, 20, 19, 20, 21, 20, 10, 20, 19, 20, 21, 20, 10, 20, 19, 18, 12, 8, 3, 2, 9, 11, 12, 8, 3, 2, 9, 8, 12, 18, 23, 30, 34, 41, 42, 43, 32, 33, 34, 30, 34, 41, 34, 33, 32, 31, 21, 20, 10, 20, 19, 22, 23, 18, 12, 8, 3, 2, 9, 11, 10, 20, 21, 20, 19, 20, 10, 20, 21, 31, 32, 43, 42, 41, 34, 30, 23, 18, 12, 8, 3, 2, 9, 11, 19, 20, 10, (1)	1,107	11.0
$SSEU$ (sector b)	4, 5, 6, 7, 13, 17, 24, 29, 35, 40, 39, 36, 37, 27, 26, 25, 28, 29, 35, 40, 39, 38, 37, 27, 26, 16, 15, 14, 13, 17, 24, 29, 35, 40, 39, 36, 37, 27, 26, 25, 28, 29, 35, 40, 39, 38, 37, 27, 26, 16, 15, 6, 5, 4, 7, 13, 17, 24, 29, 35, 40, 39, 36, 37, 27, 26, 25, 28, 29, 35, 40, 39, 38, 37, 27, 26, 16, 15, 14, 13, 17, 24, 29, 35, 40, 39, 36, 37, 27, 26, 25, 28, 29, 35, 40, 39, 38, 37, 27, 26, 16, 15, 6, 7, (4)	1,471	13.8
one-step greedy (sector b)	4, 5, 4, 7, 6, 15, 14, 13, 17, 24, 25, 26, 16, 15, 6, 5, 4, 7, 14, 13, 17, 24, 29, 28, 25, 26, 16, 15, 6, 5, 4, 7, 14, 13, 17, 24, 29, 35, 40, 39, 36, 37, 27, 26, 25, 28, 29, 24, 17, 13, 7, 6, 15, 16, 26, 25, 28, 29, 35, 40, 39, 38, 37, 36, 37, 27, 26, 16, 15, 14, 13, 17, 24, 25, 28, 29, 35, 40, 39, 40, 35, 40, 39, 40, 35, 40, 39, 38, 37, 36, 28, 25, 26, 16, 15, 6, 5, (4)	1,238	8.94

V. SUMMARY AND FUTURE WORK

In this paper, we considered the problem of effective patrolling in a dynamic and stochastic environment. The patrol locations are modeled with different priorities and varying incident rates. We identified a solution approach, which has two steps. First, we partition the set of nodes of interest into sectors. Each sector is assigned to one patrol unit. Second, for each sector, we exploited a response strategy of preemptive call-for-service response, and designed multiple off-line patrol routes. We applied the MDP methodology and designed a novel learning algorithm to obtain a deterministic optimal patrol route. Furthermore, we applied Softmax action selection method to device multiple patrol routes for the patrol unit to randomly choose from. Future work includes the following: a) consiging

TABLE III
MULTIPLE PATROLLING ROUTES

Route	Patrol route	Expected Reward	Reward /distance
Route I:	1, 10, 20, 21, 22, 23, 30, 34, 33, 32, 43, 42, 33, 34, 41, 42, 33, 34, 30, 23, 22, 19, 20, 10, 11, 12, 8, 9, 2, 3, 8, 12, 11, 19, 20, 21, 31, 32, 43, 42, 41, 34, 33, 32, 31, 21, 22, 23, 18, 19, 20, 10, 11, 12, 8, 9, 2, 3, 8, 12, 18, 23, 30, 34, 41, 42, 33, 34, 30, 23, 18, 19, 22, 21, 20, 19, 18, 23, 30, 34, 33, 32, 31, 21, 20, 10, 11, 9,	1,525	17.05
Route II:	1, 10, 20, 21, 22, 19, 20, 21, 31, 32, 43, 42, 41, 34, 33, 32, 31, 21, 20, 10, 11, 9, 8, 3, 2, 9, 8, 12, 18, 23, 22, 19, 20, 10, 11, 9, 8, 12, 18, 23, 30, 34, 41, 42, 33, 32, 31, 21, 20, 19, 22, 23, 30, 34, 41, 42, 33, 32, 43, 42, 33, 34, 30, 23, 18, 19, 22, 23, 18, 19, 20, 10, 11, 12, 8, 9, 11, 12, 18, 23, 30, 34, 33, 32, 31, 21, 20, 19, 11, 12, 8, 3, 2, 9, 8, 3, 2, (1)	1.831	18.68
Route III:	1, 2, 9, 11, 10, 20, 19, 11, 9, 8, 3, 2, 9, 8, 3, 2, 9, 11, 10, 20, 21, 31, 32, 43, 42, 41, 34, 33, 32, 43, 42, 33, 34, 30, 23, 18, 12, 8, 9, 11, 19, 20, 21, 31, 32, 33, 34, 30, 23, 22, 19, 20, 10, 11, 12, 8, 9, 2, 3, 8, 9, 2, 3, 8, 12, 18, 23, 30, 34, 41, 42, 43, 32, 33, 34, 30, 23, 18, 12, 11, 19, 20, 10, 11, 9, (1)	1,300	15.22
Route VI:	1, 2, 9, 8, 12, 18, 23, 22, 21, 20, 19, 18, 12, 8, 3, 2, 9, 8, 12, 18, 23, 30, 34, 41, 42, 33, 32, 31, 21, 20, 10, 11, 9, 8, 12, 18, 23, 22, 19, 20, 10, 11, 12, 18, 23, 30, 34, 33, 42, 43, 32, 31, 21, 20, 10, 11, 12, 8, 3, 2, 9, 8, 12, 18, 19, 20, 10, 11, 9, 8, 12, 18, 19, 22, 23, 30, 34, 41, 42, 33, 32, 31, 21, 20, 10, 11, 9, 8, 3, 2, 9,	1,389	15.20

the incident processing time and resource requirement at each node; b) including patrol unit's resource capabilities in the patrolling formulation; c) and applying adaptive parameter updates for incident rates and importance rates at each node.

REFERENCES

[1] D. J. Kenney, *Police and Policing: comteporary issues*, Praeger, 1989.

[2] R. C. Larson, *Urban Police Patrol Analysis,* The MIT Press, 1972.

[3] J. Tsitsiklis, B. V. RoyAn, *Analysis of Temporal-Difference Learning with Function Approximation*, IEEE Transactions on Automatic Control, Vol. 42, No. 5, May 1997, pp. 674-690.

[4] M. L. Puterman, *Markov Decision Processes: Discrete Stochastic Dynamic Programming,* Wiley-Interscience Publication, 1994.

[5] R.S., Garfinkel, and G.L. Nemhauster, *Optimal Political Districting by Implicitly Enumeration Techniques* vol. 16, pp. 495-508, 1970.

[6] D. Du and P. M. Pardalos, *Handbook of Combinatorial Optimization*, Kluwer Academic Publishers, 1998.

[7] D. P. Bertsekas, J. Tsitsiklis, *Neuro-Dynamic Programming,* Athena Scientific 1996.

[8] R. S. Sutton and A. G. Barto, *Reinforcement Learning: An Introduction,* The MIT Press.

[9] D. P. Bertsekas, J. N. Tsitsiklis, and C. Wu, *Rollout Algorithms for Combinatorial Optimization,* Journal of Heuristics, vol. 3(3), pp 245-262, 1997.