Robot Monitoring for the Detection and Confirmation of Stochastic Events

Ahmad Bilal Asghar Stephen L. Smith

***Abstract*— In this paper we consider a robot patrolling problem in which events arrive randomly over time at the vertices of a graph. When an event arrives it remains active for a random amount of time. If that time active exceeds a certain threshold, then we say that the event is a true event; otherwise it is a false event. The robot(s) can traverse the graph to detect newly arrived events, and can revisit these events in order to classify them as true or false. The goal is to plan robot paths that maximize the number of events that are correctly classified, with the constraint that there are no false positives. We show that the offline version of this problem is NP-hard. We then consider a simple patrolling policy based on the traveling salesman tour, and characterize the probability of correctly classifying an event. We investigate the problem when multiple robots follow the same path, and we derive the optimal (and not necessarily uniform) spacing between robots on the path.**

I. INTRODUCTION

Consider the following motivating example. A robot traverses a parking lot, issuing tickets to vehicles that have overstayed the allowed amount of parking time T. If a vehicle has been present for more than T time after it was first spotted by the robot, then it gets a ticket. However, there is a possibility that some vehicles overstay their allowed parking time but leave before the robot tickets them. Our goal is to define a monitoring policy for the robot which minimizes the number of un-ticketed, overstaying vehicles.

The *event detection and confirmation problem* considered in this paper is as follows. A robot or a group of robots patrol a weighted graph by traversing its edges. Events arrive at the vertices and remain active for a randomly distributed amount of time. If an event remains active for more than a given time $T > 0$, then we say it is a *true event*, otherwise it is a *false event*. For a robot to verify that an event is true, it must first detect the event by visiting the vertex, and then must revisit the vertex at least T time units later to confirm the event. Thus, the goal is for the robots to maximize the expected number of true events that are successfully confirmed. This is a classification problem in which false positives are not permitted: Each event is initially classified as false, and it can be classified as true only if it is confirmed.

Related work: While the proposed event detection and confirmation problem has not, to our knowledge, been directly studied there are several closely related problems. In the patrolling problem [1], [2], [3], the goal is to monitor an environment or boundary using one or more robots/sensors. The performance criteria is to minimize the maximum time between visits to any region in the environment. In [1], the problem is considered for multiple robots, and it is shown that good patrolling performance can be achieved by computing a single traveling salesman tour (TSP) [4], and then equally distributing the robots along this tour.

In [5] the patrolling is extended to environments in which each region has a different importance level, and the goal is to minimize the time between visits to a region, weighted by that regions importance. The work in this paper can be thought of as a natural extension of patrolling in which an action must be taken if an event is detected during the patrol (that action being confirmation).

Our problem is also related to the TSP with time windows [6], [7], where the input is a graph along with a time window assigned to each vertex. The goal is to find the shortest tour that visits each vertex exactly once, and within its time window. We show that the event confirmation aspect of our problem is closely related to TSP with time windows, since each event must be confirmed at least T time units after detection, but before the event expires.

Another closely related problem is the pickup and delivery problem [8], where one seeks to pickup a set of customers at their desired origin locations and drop them off at their desired destination locations, all within their specified time windows. Our problem can be thought of as a variation in which the pickup time (i.e., the event arrival time) is unknown to the robot, the pickup and destination locations coincide, and the dropoff time window depends on the time that the pickup occurred (i.e., the event was detected).

The stochastic aspect of the problem bears a close resemblance to dynamic vehicle routing (DVR) [9], where spatially distributed customers arrive stochastically over time, and the goal is to minimize the expected time between a customers arrival and the time it is visited by a vehicle. The most closely related work in this area is [10], in which the customers exit the system if they are not visited within a time window. However, DVR differs from the proposed work in three regards: i) the environment is a Euclidean space rather than a graph, ii) the customer is known to the vehicles upon arrival, and iii) a second confirmation visit is not required.

Contributions: In this paper we introduce the event detection and confirmation problem. There are three main contributions. First, we characterize the complexity of the problem by relating its offline counterpart to the TSP with time windows. Second, we propose a simple periodic visit strategy based on the TSP and analyze the probability of confirming a true event. Third, we give some insight into the multi-robot problem, and show that unlike traditional patrolling [1] when robots are all placed on the same path, it is not always optimal for them to be equally spaced.

This research is partially supported by the Natural Sciences and Engineering Research Council of Canada (NSERC).

The authors are with the Department of Electrical and Computer Engineering, University of Waterloo, Waterloo ON, N2L 3G1 Canada (abasghar@uwaterloo.ca; stephen.smith@uwaterloo.ca)

A. Preliminaries

We require a few basic properties of the Poisson and exponential distributions [11]. The *exponential distribution* with parameter μ is a continuous distribution with a probability density function $f(x) = \mu e^{-\mu x}$ if $x \geq 0$ and $f(x) = 0$ otherwise.

A *Poisson process* with parameter λ is a stochastic counting process such that the time between successive events is exponentially distributed. The expected number of event arrivals in a time interval $[t_1, t_2]$ is $\lambda(t_2 - t_1)$. The Poisson process also satisfies the property of *stationary increments* where the number of arrivals in an interval of time is independent of the number of arrivals prior to that interval.

The following result will be useful in our analysis.

Lemma I.1 (Poisson Arrival Time Distribution, [12])**.** *Given that k events arrived in the time interval $(a, b]$, the times $t_1, t_2, \ldots, t_k$ of these arrivals, considered as unordered random variables, are independent and uniformly distributed on $(a, b]$.*

A consequence of this result is that if we know an event arrived in an interval of time, then its arrival time is uniformly distributed over that time interval.

II. Problem Statement and Hardness

In this section we introduce the event detection and confirmation problem and we characterize its hardness.

A. Problem Statement

The Event Detection and Confirmation problem is defined on an undirected weighted graph $G = (V, E, w)$, where V is the vertex set, E is the set of edges and $w : E \to \mathbb{R}$ represents edge weights. The vertices depict the locations to be monitored by $m \geq 1$ robots. We take the metric closure [5] of G in order to obtain a complete graph, in which the length of each edge is equal to the shortest path distance in the original graph. For simplicity we will refer to this complete graph as $G = (V, E, w)$.

Events arrive at each vertex $v \in V$ according to a Poisson random process [11] with a parameter λ_v. Similarly, we assume that the activity period of an event at vertex is exponentially distributed with parameter μ_v.[1] The events are distinct and they can be identified by the robots. Moreover, only one event can be active at a vertex at a time. The arrivals and active times of events at different vertices are independent.

There is also a critical time T as input to the problem. We call an event a *true event* if it remains active for at least time T. The robots, while on their patrolling path, perform two tasks: *detection* and *confirmation*. The detection of an event is discovering it for the first time at a vertex, and the confirmation is observing an event at a vertex after it has been active for at least time T. The robots can classify an event as true if and only if they confirm that event. Notice that if a true event becomes inactive before being confirmed, it cannot be classified by the robot as a true event.

[1] In queueing theory, the Poisson distribution and exponential distribution are often used to model customer arrival rates and customer service times, respectively [13]. Most of the analysis in this paper holds for more general distributions: the ability to obtain closed-form expressions, however, leverages these specific distributions.

Fig. 1. The events remaining active until $t' + T$ are true, but they cannot be confirmed unless they stay until $(n + 2)\tau$.

When a robot reaches a vertex in its tour, it faces one of the following scenarios: i) The vertex is empty: then the robot can delete the event from its database which was recorded to be at that vertex (if any); ii) There is a new event at the vertex: then the robot stores it against that vertex with the current time stamp; iii) There is an event at the vertex which was detected some previous check at that vertex: In this case the robot looks up the time stamp of that event and compares it with current time to see whether it is a true event or not.

Event detection and confirmation problem: Find patrolling paths for the robots to minimize the probability of incorrectly classified events. The problem does not allow the robots to classify a false event as true, so the optimization task can be stated as maximizing the probability of correctly classified true events.

Proposition II.1 (Hardness of Offline Problem)**.** *The problem of finding a feasible tour for the off-line version (with the arrival times and activity periods of events available beforehand) of Event Detection and Confirmation Problem is NP-Complete.*

Due to space considerations, the proof [**?**] is omitted from this paper. Since it is computationally intractable to even determine a feasible patrolling path given all the problem data beforehand, we do not expect there to exist a tractable algorithm for optimally solving the on-line problem. In the following section we look at the probability of confirmation at a single vertex in the graph. Using this we can analyze the performance of a policy based on a TSP tour.

III. Analysis for a Single Vertex

Let us consider a deterministic patrolling policy which periodically visits each vertex, and let the visit period for a vertex v be τ. Inter-arrival and staying times of events at vertex v are distributed exponentially with the parameters λ and μ respectively (we drop the subscript v in this section for simplicity of notation). Since the arrival and departure process at a vertex are independent of the states of other vertices, we can focus the analysis on a single vertex.

A. Confirmation Avoiding Interval for True Events

Our end goal is to maximize the probability of correctly classified true events. There is a chance that a true event becomes inactive without being confirmed, because the robot does not know the exact arrival time of the event and it can only visit the vertex of the event periodically. In the following we characterize the time interval on which a true event can become inactive and avoid confirmation.

Proposition III.1 (False Negatives)**.** *Suppose an event arrives at a vertex between two consecutive visits made by the robot at 0 and τ, and the arrival time of the event is given* *by $t' \in (0, \tau]$. Then, if that event becomes inactive in the time interval*

$$\big(t' + T, (n+2)\tau\big), \text{ where} \tag{1}$$

$$n = \begin{cases} \frac{T}{\tau} - 1, & \text{if } T \text{ is a multiple of } \tau \\ \left\lfloor \frac{T}{\tau} \right\rfloor, & \text{otherwise}, \end{cases} \tag{2}$$

it will be a true event which can not be confirmed.

Proof. The starting point of the interval $(t' + T, (n+2)\tau)$ is trivial since the event will become true after $t = t' + T$. For the end point, notice that the robot detected the event at $t = \tau$, and it will only be able to confirm the event on times that are integer multiples of τ. By the definition of n, $n\tau < T \leq (n+1)\tau$. So when the robot observes the same event at $(n+2)\tau$, it confirms that is has been there for more than T, since $(n+2)\tau - \tau = (n+1)\tau \geq T$. Moreover, there can be cases for some T and t' when $t' + T < (n+1)\tau$, as shown in Figure **??**, but since the robot detected the event at τ and $(n+1)\tau - \tau = n\tau < T$, the robot does not know that the event has been active for more than T and cannot confirm it. □

The events which become inactive in the interval (1) will be true events, but cannot be correctly classified by the robot as true. We will use this fact along with the exponential active times of the events in the following section to calculate the chances of correctly classifying a true event.

B. Probability of Correctly Classifying True Events

The events which were detected at $t = \tau$ will be classified as true if they remain active until $t = (n+2)\tau$, as shown in Proposition III.1. If the robot detects an event, then it knows that the event arrived in the interval between the last two visits to its vertex. By the property of stationary increments, the time scale can be shifted to say that the arrival time of the event is given by $t' \in (0, \tau]$. Using the consequence of Lemma I.1, the arrival time t' is uniformly distributed over $(0, \tau]$. We write this distribution of t' in $(0, \tau]$ as

$$f(t') = \frac{1}{\tau} \quad \text{for } t' \in (0, \tau]. \tag{3}$$

We will use this uniform density along with the interval(1) to find the probability of confirming true events.

Proposition III.2 (Probability of Successful Classification)**.** *The probability of confirming a true event at vertex v with arrival rate λ, departure rate μ, and a robot with visit period τ is*

$$\mathbb{P}[\texttt{confirm|v}] = \frac{e^{-\mu[(n+2)\tau - T]}(e^{\mu\tau} - 1)}{\mu\tau}, \tag{4}$$

where n is defined in (2).

Proof. According to the confirmation avoiding interval given in (1), the events arriving at time $t' \in (0, \tau]$ and departing after $(n+2)\tau$ will be confirmed by the robot. Using the exponential staying time distribution ,we can find the probability of confirming a true event given that it arrived at time $t' \in (0, \tau]$.

$$\mathbb{P}[\texttt{confirm|v and t'}] = \frac{\displaystyle\int_{(n+2)\tau}^{\infty} \mu e^{-\mu(t - t')} dt}{\displaystyle\int_{T}^{\infty} \mu e^{-\mu t} dt}, \tag{5}$$
$$= e^{-\mu[(n+2)\tau - T]} e^{\mu t'}$$

The numerator in (5) represents the events that will stay long enough to be confirmed, and the denominator represents all the events that are true. Using the arrival time density in the interval $(0, \tau]$ from (3), we can un-condition the arrival time.

$$\mathbb{P}[\texttt{confirm|v}] = \int_0^{\tau} \mathbb{P}[\texttt{confirm|v and t'}] \, f(t') dt',$$

which gives us the desired expression. □

The Event Detection and Confirmation problem seeks to maximize the number of true events that are confirmed. So, one would want to maximize the probability given in (4). We will extend this expression to the complete path and then try to maximize the probability of confirming true events over the whole graph.

IV. A Single Robot Policy Based on the TSP

In this section, we will derive the expression for the probability of confirming true events over the graph, and then use it in a special case to recommend a policy based on the TSP tour of the graph.

A. Probability of Correct Classification for the Tour

We start with the analysis of any patrolling policy with possibly different periodic visit times to vertices, and then specialize the equation for the case when the periodic visit times to the vertices are equal and the events' activity period is governed by the same process for all the vertices.

Using equation (4), for a vertex v with arrival and departure rates given by λ_v and μ_v respectively, the robot visiting that vertex with a period τ_v will confirm true events on that vertex with a probability given by

$$\mathbb{P}[\texttt{confirm|v}] = \frac{e^{-\mu[(n_v+2)\tau_v - T]}(e^{\mu_v\tau_v} - 1)}{\mu_v\tau_v},$$
$$\text{where } n_v = \begin{cases} \frac{T}{\tau_v} - 1, & \text{if } T \text{ is a multiple of } \tau_v \\ \left\lfloor \frac{T}{\tau_v} \right\rfloor, & \text{otherwise}, \end{cases} \tag{6}$$

Proposition IV.1 (Probability Expressions)**.** *The probability of correctly classifying true events for the periodic tour is given by*

$$\mathbb{P}[\mathit{confirm}] = \frac{\sum_v \mathbb{P}[\texttt{confirm|v}]\,\lambda_v}{\sum_v \lambda_v}, \tag{7}$$

where $\mathbb{P}[\texttt{confirm|v}]$ is given in equation (6). Moreover, in the special case where $\tau_v = \tau$, and $\mu_v = \mu$, for all $v \in V$,

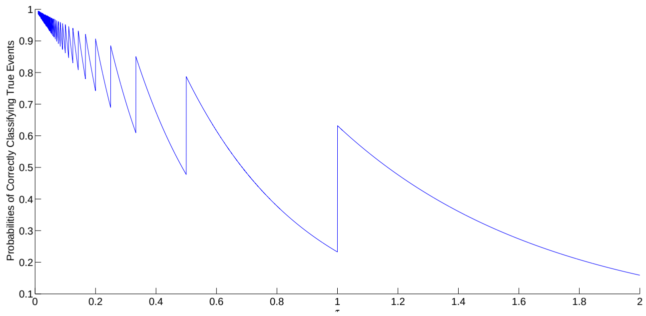

Fig. 2. The probability of correctly classifying true events versus τ, with $T = 1$, $\mu = 1$.

then

$$\mathbb{P}\left[\mathit{confirm}\right] = \frac{e^{-\mu[(n+2)\tau - T]}(e^{\mu\tau} - 1)}{\mu\tau}, \quad \text{where } n = \begin{cases} \frac{T}{\tau} - 1, & \text{if } T \text{ is a multiple of } \tau \\ \left\lfloor \frac{T}{\tau} \right\rfloor, & \text{otherwise}. \end{cases} \tag{8}$$

Proof. We want to remove the condition of arrival being on a certain vertex v from equation (6). We know that

$$\mathbb{P}\left[\texttt{confirm}\right] = \sum_v \mathbb{P}\left[\texttt{confirm}\,|\,\texttt{v}\right] \mathbb{P}\left[\texttt{arrival at v}\right].$$

Since the arrivals of events at different vertices are independent processes, the probability of arrival of an event being on vertex v is

$$\mathbb{P}\left[\texttt{arrival at v}\right] = \frac{\lambda_v}{\sum_v \lambda_v}.$$

Therefore,

$$\mathbb{P}\left[\texttt{confirm}\right] = \frac{\sum_v \mathbb{P}\left[\texttt{confirm}\,|\,\texttt{v}\right] \lambda_v}{\sum_v \lambda_v}.$$

If we consider the special case when $\mu_v = \mu, \tau_v = \tau, \forall v \in V$, then $\mathbb{P}\left[\texttt{confirm}\,|\,\texttt{v}\right]$ is same for all the vertices, and can be factored out of the summation, giving us equation (8). □

Remark IV.2 (Dependence on λ). Since the number of total events as well as confirmed events depend on λ , the probability expression (8) is independant of λ. •

B. *Policy based on a TSP tour*

Expression (8) holds for a periodic tour of the graph and a TSP tour minimizes the time τ for a given speed of the robot. However, there are cases when decreasing the robot speed and thus increasing τ results in a higher probability. This is due to the discontinuity of n in equation (2) and can be seen in Figure 2. Intuitively it means that timing the visits such that T is a multiple of τ decreases the chances of missing the confirmation of true events. Based on this observation, we arrive at following single robot policy.

Policy for a Single Robot:

(i) Calculate the TSP tour of the graph, and find the minimum time $\tau_{\min}$ to complete that tour by the robot at its maximum speed.
(ii) Decrease $\tau_{\min}$ to the nearest divisor of T, and call it τ_{peak}.
(iii) Calculate the probability from equation (8) for $\tau = \tau_{\min}$ and $\tau = \tau_{\text{peak}}$ and choose the period with higher probability.

Remark IV.3 (Omitting Vertices from Tour). Equation (7) suggests that missing some vertices on the tour can give a better probability. An instance of the problem can be easily constructed where missing a far away vertex from the tour will result in a much lower τ for the other vertices and hence increase the probability of correctly classifying true events over the whole graph. However, such policies raise the possibility that "intelligent" events would begin to choose this unvisited vertex more frequently, altering the arrival rates. This becomes a problem in game theory, and thus we leave it for future work. •

V. Multiple Robots

In this section we consider the case of multiple robots. We assume that the communication graph between the robots is strongly connected, so that any two robots can communicate without significant delay. Thus, we can assume that the database containing all active events is shared among the robots. This means that it is possible to have robot i detect an event and robot j confirm it.

A. *Specializing Robot Capabilities*

One possible solution in the multi-robot case is to utilize *specialization* in which a robot performs exclusively detection, or exclusively confirmation.

Definition V.1 (Specialized robot capability)**.** We say that a robot is a detection (confirmation) robot if it is capable of performing only event detections (confirmations).

First, it is easy to see that specialization cannot be optimal. In specializing we eliminate the possibility of a confirmation robot detecting an event, even if it is the first robot to visit the vertex after the events' arrival. Similarly, we eliminate the possibility of a detection robot confirming an event.

However, there are cases where specialization may be required, for example, if the sensors needed for detection and confirmation differ. The following simple lemma shows that when specializing, detection is the bottleneck.

Lemma V.2 (Specialization among robots)**.** *Given n_d detection robots, confirmation can be performed optimally using only n_d confirmation robots.*

Proof. Let the paths followed by the n_d detection robots be $P_1, \ldots, P_{n_d}$. We then create n_d confirmation paths by placing a confirmation robot on each detection path, but with a time lag of exactly T seconds.

An event that is detected on a given path P_i will be confirmed optimally exactly T time units later by the corresponding confirmation robot. □

$$\mathbb{P}\left[\texttt{confirm}\right] = \begin{cases} \frac{1}{\mu\tau}(e^{-\mu((n+1)\tau-T)}(e^{\mu t_{\text{lag}}}(1-e^{-\mu\tau})), & t_{\text{lag}} \leq T - n\tau, \\ \frac{1}{\mu\tau}(e^{-\mu((n+1)\tau-T)}(e^{\mu t_{\text{lag}}} + e^{\mu(\tau - t_{\text{lag}})} - 2), & T - n\tau < t_{\text{lag}} \leq (n+1)\tau - T, \\ \frac{1}{\mu\tau}(e^{-\mu((n+1)\tau + t_{\text{lag}} - T)}(e^{\mu\tau} - 1), & t_{\text{lag}} > (n+1)\tau - T. \end{cases} \tag{10}$$

$$\mathbb{P}\left[\texttt{confirm}\right] = \begin{cases} \frac{1}{\mu\tau}(e^{-\mu((n+1)\tau-T)}(e^{\mu t_{\text{lag}}}(1-e^{-\mu\tau})), & t_{\text{lag}} \leq (n+1)\tau - T, \\ \frac{1}{\mu\tau}(e^{-\mu((n+1)\tau-T)}(2 - e^{-\mu t_{\text{lag}}} - e^{-\mu(\tau - t_{\text{lag}})}), & (n+1)\tau - T < t_{\text{lag}} \leq T - n\tau, \\ \frac{1}{\mu\tau}(e^{-\mu((n+1)\tau + t_{\text{lag}} - T)}(e^{\mu\tau} - 1), & t_{\text{lag}} > T - n\tau. \end{cases} \tag{11}$$

The consequence of this result is that detection is the bottleneck when looking at specialized robots. Thus, in this case, one can use existing techniques to design patrolling paths for detection, and then use Lemma V.2 for the confirmation paths. In the next section we focus on the more complex case in which each robot can both detect and confirm.

B. Optimal Spacing Between Robots on a Common Path

In this section we look at the case where each robot can both detect and confirm events. We focus on the special case in which there are two robots moving along the same tour, with a period of $\tau > 0$. We seek the optimal spacing of these two robots along the tour. We discuss the extension to m robots at the end of this section.

To this end, define the variable to optimize as t_{lag} which is the time lag between the first and second robot on the common tour. Since the robots travel the tour with period τ, we have $t_{\text{lag}} \in (0, \tau)$. Consider an event that arrives at a vertex at time $t' \geq 0$. We can shift the time scale such that $t' \in [0, \tau)$.

Then, let us consider the earliest possible time that this event can be detected and confirmed: we call these times t_{det} and t_{conf}, respectively, where $t_{\text{conf}} \geq t_{\text{det}} + T$ and $t_{\text{det}} \geq t'$.

If these times are known, then the probability of confirming a true event, given that it arrives at time t' is

$$\begin{aligned} \mathbb{P}\left[\texttt{confirm}|t'\right] &= \frac{\mathbb{P}\left[\texttt{active} > t_{\text{conf}} - t'\right]}{\mathbb{P}\left[\texttt{active} > T\right]} \\ &= \frac{e^{-\mu(t_{\text{conf}} - t')}}{e^{-\mu T}} \\ &= e^{\mu t'} e^{-\mu(t_{\text{conf}} - T)}, \end{aligned} \tag{9}$$

where we have used the fact that an event's active time is exponentially distributed with parameter μ. Now, we can calculate t_{det} and t_{conf} as a function of t_{lag} using the following two cases, each containing two sub-cases.

Case 1: If $t' \in (0, t_{\text{lag}}]$ then $t_{\text{det}} = t_{\text{lag}}$.

(i) If $t_{\text{lag}} + T \leq (n+1)\tau$ then the earliest time that the event can be confirmed is $t_{\text{conf}} = (n+1)\tau$.

(ii) If $t_{\text{lag}} + T > (n+1)\tau$, then the earliest time that the event can be confirmed is $t_{\text{conf}} = (n+1)\tau + t_{\text{lag}}$.

Case 2: If $t' \in (t_{\text{lag}}, \tau]$ then $t_{\text{det}} = \tau$.

(i) If $\tau + T \leq (n+1)\tau + t_{\text{lag}}$ i.e., $t_{\text{lag}} \geq T - n\tau$, then the earliest time that the event can be confirmed is $t_{\text{conf}} = (n+1)\tau + t_{\text{lag}}$.

(ii) If $t_{\text{lag}} < T - n\tau$, then the earliest time that the event can be confirmed is $t_{\text{conf}} = (n+2)\tau$.

Based on the four cases and equation (9), we can compute the probability of detection as a function of t_{lag} as

$$\mathbb{P}\left[\texttt{confirm}\right] = \int_0^\tau \mathbb{P}\left[\texttt{confirm}|t'\right] f(t') dt',$$

where $f(t')$ is the uniform distribution from equation (3).

When evaluating this integral, there are two more cases:

- If $T - n\tau \leq (n+1)\tau - T$ we get equation (10).
- If $T - n\tau > (n+1)\tau - T$, we get equation (11).

Now, from these expressions we can optimize t_{lag}. Notice that in the case when $t_{\text{lag}} = \tau/2$, the expressions in (10) and (11) both simplify to equation (8) with τ replaced by $\tau/2$.

Proposition V.3 (Optimal value of t_{lag})**.** *The equations* (10) *and (11) achieve their global maxima at one (or more) of the following points: i)* $t_{\text{lag}} = \frac{\tau}{2}$*; ii)* $t_{\text{lag}} = T - n\tau$*; or iii)* $t_{\text{lag}} = (n+2)\tau - T$.

Proof. The equations (10) and (11) are piecewise continuous and have discontinuities at points $t_{\text{lag}} = T - n\tau$ and $t_{\text{lag}} = (n+2)\tau - T$. The continuous pieces defined on the first and third interval of the equations are strictly monotone and achieve their maximum values at the discontinuities. The third continuous part has an extremum at $t_{\text{lag}} = \frac{\tau}{2}$. Thus its maximum lies either at $\frac{\tau}{2}$ or at one of the discontinuities. □

These values of t_{lag} will optimize the probability for a given value of τ. However, as we observed in the single robot case, decreasing the speed of the robot to increase τ to a divisor of $2T$ can result in a higher probability. Based on this, we arrive at the following policy for two robots:

Policy for Optimizing the Spacing of Two Robots:

(i) Evaluate the expression (10) or (11) depending on whether $T - n\tau \leq (n+1)\tau - T$ or not, at the points $t_{\text{lag}} = \tau/2$, $t_{\text{lag}} = T - n\tau$ and $t_{\text{lag}} = (n+2)\tau - T$.

(ii) Decrease τ to the nearest divisor of $2T$, call it τ_n and evaluate equation (11) at $t_{\text{lag}} = \tau_n/2$.

(iii) Choose the lag which gives maximum probability among the above four candidates.

The following remark discusses the extension to m robots.

Remark V.4 (Generalizing to m robots)*.* In the case that there are m robots, there are $m-1$ variables t_{lag} to optimize. The number of cases to consider becomes too large to complete the same analysis. However, based on the observations made for two robots, the following can be said for the n-robot case:

(i) If τ/m is a multiple of T, then equally space the robots on the path.
(ii) If $\tau < mT$, decrease τ to the nearest divisor of mT and using this new period, equally space the robots.
(iii) If $\tau > mT$, choose the spacing such that the robots follow each other by a time lag of T.

This policy follows from the observation that in the two robot case this procedure often yields the optimal t_{lag}. However, it is not, in general guaranteed to find the optimal spacing.•

VI. Application Example

Let us apply the patrolling policies derived above to an actual parking lot situated at a market place near University of Waterloo. The vehicles parked at the parking spots will serve as events at the vertices of a graph and overstaying vehicles will be considered true events. For the purposes of simulation, we assume the expected staying time of vehicles to be around 75 minutes and the allowed parking time to be two hours. This gives $\mu = \frac{1}{75}$ and $T = 120$. Moreover, the length of the patrolling path based on a TSP tour is calculated to be approximately 870 meters.

We assume the robot has a maximum speed of 1 m/s, which gives a minimum tour period of 14.5 minutes. The probability of ticketing for a tour with this period comes out to be 0.7905 from equation (8). However, if we increase the period to 15 minutes by decreasing the speed of robot to 0.967 m/s, using equation (8) the probability increases to 0.9063 — an increase of 14.6%.

In case of two robots, the probability of ticketing an overstaying vehicle using a period of 14.5 minutes and the robots equally spaced will be 0.9128 from equation (11) . But, if one robot follows the other with a time lag of $T - n\tau = 4$ minutes or $(n+1)\tau - T = 10.5$ minutes, then the probability increases to 0.9221. However, there is still room for improvement. If both the robots decrease their speed to make their period a multiple of T and then follow each other with a lag of $\tau/2 = 7.5$ minutes, the probability will be 0.9515 which can be calculated using equation (11).

This example and Figure 3 show that a lag of $\tau/2$ often results in a better probability, and even if it does not, optimal lag provides very little advantage. However, when $\tau > 2T$ (large environments), optimal lag provides much better results.

VII. Conclusions and Future Work

In this paper we considered a robot patrolling problem called the event detection and monitoring problem. We characterized the probability of confirming a true event for a TSP tour. We also gave some initial insights into the multiple robot problem.

For future work we would like to compare the performance of a TSP tour with that of a min-max latency tour [5], where visit frequency can be proportional to the event arrival rate at a given vertex. We would also like to study the problem from

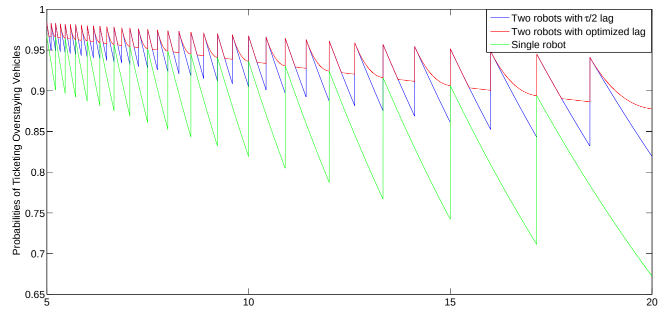

Fig. 3. The probability of ticketing overstaying vehicles as a function of τ for 1) a single robot, 2) two robots with lag of $\tau/2$, and 3) two robots with optimized lag. Here $T = 120$, and $\mu = 1/75$.

a game theoretic perspective, where the events distribution may change as a function of the patrolling policy. In this case we need to look at randomizing the path, in order to decrease its predictability. Randomized policies have been considered in perimeter patrolling problems [2], and thus will form a solid basis on which to build.

References

[1] Y. Chevaleyre, "Theoretical analysis of the multi-agent patrolling problem," in *Intelligent Agent Technology, IEEE/WIC/ACM International Conf. on*, 2004, pp. 302–308.
[2] N. Agmon, S. Kraus, and G. Kaminka, "Multi-robot perimeter patrol in adversarial settings," in *Robotics and Automation. IEEE Int. Conf. on*, May 2008, pp. 2339–2345.
[3] M. Baseggio, A. Cenedese, P. Merlo, M. Pozzi, and L. Schenato, "Distributed perimeter patrolling and tracking for camera networks," in *Decision and Control, IEEE Conf. on*. IEEE, 2010, pp. 2093–2098.
[4] B. Korte and J. Vygen, *Combinatorial Optimization: Theory and Algorithms*, 4th ed., ser. Algorithmics and Combinatorics. Springer, 2007, vol. 21.
[5] S. Alamdari, E. Fata, and S. L. Smith, "Persistent monitoring in discrete environments: Minimizing the maximum weighted latency between observations," *International Journal of Robotics Research*, vol. 33, pp. 138–154, 2014.
[6] M. W. Savelsbergh, "Local search in routing problems with time windows," *Annals of Operations research*, vol. 4, no. 1, pp. 285–305, 1985.
[7] G. Laporte, "Fifty years of vehicle routing," *Transportation Science*, vol. 43, no. 4, pp. 408–416, 2009.
[8] S. Ropke and D. Pisinger, "An adaptive large neighborhood search heuristic for the pickup and delivery problem with time windows," *Transportation science*, vol. 40, no. 4, pp. 455–472, 2006.
[9] F. Bullo, E. Frazzoli, M. Pavone, K. Savla, and S. Smith, "Dynamic vehicle routing for robotic systems," *Proceedings of the IEEE*, vol. 99, no. 9, pp. 1482–1504, Sept 2011.
[10] M. Pavone, N. Bisnik, E. Frazzoli, and V. Isler, "A stochastic and dynamic vehicle routing problem with time windows and customer impatience," *ACM/Springer Journal of Mobile Networks and Applications*, vol. 14, no. 3, pp. 350–364, 2009.
[11] G. Grimmett and D. Stirzaker, *Probability and Random Processes*. Oxford University Press, 2001.
[12] S. M. Ross, *Stochastic processes*. John Wiley & Sons New York, 1996, vol. 2.
[13] L. Kleinrock, *Queueing Systems. Volume I: Theory*. John Wiley, 1975.