# Stochastic event capture using mobile sensors subject to a quality metric

**Conference Paper** · January 2006

DOI: 10.1145/1161089.1161102 · Source: DBLP



**3 authors:**

Nabhendra Bisnik
Intel


Alhussein A. Abouzeid
Rensselaer Polytechnic Institute


Volkan Isler
University of Minnesota Twin Cities

# Stochastic Event Capture Using Mobile Sensors Subject to a Quality Metric *

Nabhendra Bisnik
Dept. of Electrical, Computer and Systems Engineering, RPI
Troy, NY, USA
bisnin@rpi.edu

Alhussein Abouzeid
Dept. of Electrical, Computer and Systems Engineering, RPI
Troy, NY, USA
abouzeid@ecse.rpi.edu

Volkan Isler
Dept. of Computer Science, RPI
Troy, NY, USA
isler@cs.rpi.edu

## ABSTRACT

Mobile sensors cover more area over a period of time than the same number of stationary sensors. However, the quality of coverage achieved by mobile sensors depends on the velocity, mobility pattern, number of mobile sensors deployed and the dynamics of the phenomenon being sensed. The gains attained by mobile sensors over static sensors and the optimal motion strategies for mobile sensors are not well understood. In this paper we consider the problem of event capture using mobile sensors. The events of interest arrive at certain points in the sensor field and fade away according to arrival and departure time distributions. An event is said to be captured if it is sensed by one of the mobile sensors before it fades away. For this scenario we analyze how the quality of coverage scales with the velocity, path and number of mobile sensors. We characterize the cases where the deployment of mobile sensors has no advantage over static sensors and find the optimal velocity pattern that a mobile sensor should adopt.

We also present algorithms for two motion planning problems: (i) for a single sensor, what is the *minimum speed* and sensor trajectory required to satisfy a bound on event loss probability and (ii) for sensors with fixed speed, what is the *minimum number of sensors* required to satisfy a bound on event loss probability. When events occur only along a line or a closed curve our algorithms return optimal velocity for the minimum velocity problem. For the minimum sensor problem, the number of sensors used is within a factor two of the optimal solution. For the case where the events occur at arbitrary points on a plane we present heuristic algorithms for the above motion planning problems and bound their performance with respect to the optimal. The results of this paper have wide range of applications in areas like surveillance, wildlife monitoring, hybrid sensor networks and under-water sensor networks.

## Categories and Subject Descriptors

C.2 [**Computer Systems Organization**]: Computer-Communication Networks; F.2 [**Theory of Computation**]: Analysis of Algorithms and Problem Complexity; G.3 [**Mathematics of Computing**]: Probability and Statistics

## General Terms

Algorithms, Theory, Performance

## Keywords

Mobile robot motion planning, Robot sensing systems, Reliable coverage, Event capture

## 1. INTRODUCTION

A wide range of applications have been proposed for wireless sensor networks, which include surveillance, environmental monitoring, eco-system monitoring, forest fire response, health care etc [16, 28], [21, 31], [1, 23], [13, 27], [18, 37]. In traditional wireless sensor networks, static sensor nodes are randomly scattered over the sensor field with high density so that most of the sensor field is covered and the sensor network remains connected. However this approach has several disadvantages. The positions of the sensors are fixed after deployment therefore points that are not covered by the initial deployment are never covered. In a surveillance network, if an adversary gains knowledge about the positions of the sensors it can take advantage of it and thus render the sensor network useless. Failure of a few sensors may lead to disconnected components of nodes although sensor density may be high in some areas. Static sensor networks are also not able to cope with dynamic environments where new obstructions may appear after initial deployment, thus hindering proper sensing and communication operations. In short, static sensor networks require a large number of redundant nodes in order to maintain coverage and connectivity for a long period of time. Deploying a dense network may often be infeasible, due to financial constraints, or undesirable, due to the negative effects a dense network may have on the sensor field.

Recent advances in robotics and low power embedded systems have made mobile sensors [6, 9, 24, 30] a viable choice for the sensing applications mentioned above. Mobile sensors are able to mitigate most of the problems faced by static sensors and have been successfully deployed for sensing large and formidable sensor fields [1]. Since mobile sensors can move, a small number of mobile sensors may be deployed to ensure that all points would eventually be covered. A randomized motion strategy would make it difficult for the adversary to come up with ways to remain undetected by the sensors. Being mobile the sensors can exchange information with each other and the sink whenever they come within each others transmission ranges, thus keeping the network connected for a long

*The work of the first and second authors was funded in part by the National Science Foundation under grants CNS-322956 and CNS-546402.

time. However mobile sensors have certain drawbacks. Although a mobile sensor is able to cover more area than a stationary sensor over a period of time, the instantaneous area covered by both are the same. So without proper motion planning, a substantial portion of the sensor field may not be covered by the mobile sensors for a long time period. The sensors may therefore miss a lot of events that may occur at these uncovered locations, which may lead to an unacceptable quality of coverage. This problem may be severe if the phenomenon being covered is highly dynamic (either spatially or temporally) in nature.

In this paper we investigate how the quality of coverage in mobile sensor networks depends on parameters such as sensor speed, event dynamics and number of sensors deployed. We also present optimal and heuristic path planning algorithms for satisfying the required coverage quality of the sensor network.

We consider a scenario where events appear and disappear at certain points within a sensor field and the events have to be *captured* using mobile sensors. An event is said to be *captured* if a mobile sensor senses it before it disappears. If the event fades away without being captured by any of the mobile sensors then the event is said to be *lost*. The points where the event may occur are known a priori and are referred to as *Points of Interest (PoIs)*. The distributions of arrival and departure times of the events at a PoI are also completely known. The goal is to plan the motion of the mobile sensors such that the required quality of coverage (QoC) metric is satisfied for event capture. The two QoC metrics considered in this paper are (i) fraction of events captured and (ii) probability that an event is lost.

The contributions of this paper may be classified into *analysis* and *algorithms*. First we perform a detailed analysis of how the expected fraction of events captured by mobile sensors vary with the number and velocity of the mobile sensors and the event dynamics. We characterize the cases where the QoC obtained by static sensors would be better than that achieved by the same number of mobile sensors. We also investigate what is the minimum number of mobile sensors (this number obviously depends on event dynamics and velocity) after which no substantial gain in coverage is achieved by deploying more mobile sensors. Furthermore, for a mobile sensor that is free to vary its velocity along the course of its path we investigate what velocities should it adopt in order to maximize the fraction of events captured. Second, we present algorithms for *bounded event loss probability (BELP)* problem. As the name suggests, the goal of BELP problem is to plan sensor motion such that the probability that an event is lost is bounded from above. We consider two versions of BELP: (i) What is the minimum velocity with which a single mobile sensor can solve BELP (MV-BELP) and (ii) Given a fixed velocity, what is the minimum number of mobile sensors required to solve BELP (MS-BELP). For the case where the PoIs are scattered over a plane and the sensors are free to move in an unconstrained manner, we show that both versions of BELP cannot be solved optimally in polynomial time. However we present optimal and approximate algorithms for the cases where the sensors are only constrained to move along certain paths. This may occur in scenarios where mobile sensors should move along only trusted paths in order to avoid being damaged or getting stuck. We also present heuristic algorithms for the general planar case and bound the performance of the heuristic algorithm with respect to the optimal.

The results presented in this paper may be applied to wide range of areas that involve arrival of events at spatially distributed points which have to be sensed/served within a critical time, otherwise the events disappear or the service is worthless, e.g. surveillance, monitoring, underwater sensor networks and supply chain management. Here we briefly discuss the example of underwater sensor network. In underwater sensor networks, acoustic communication is the only viable means of communication for large distances. Acoustic communication consumes large amount of power and allows only small data rates due to the large multipath fading effect. Traditionally the underwater sensors are deployed on the ocean or river bed and the data is collected by retrieving them from the bed once their lifetime is over. Recently there has been considerable success in designing and deploying underwater autonomous vehicles (UAVs) [2,33]. Such UAVs may be used to harvest data from the underwater sensors. Here the underwater sensors correspond to the PoIs and UAVs correspond to mobile sensors. The generation of new information at the sensors corresponds to arrival of event. The rate depends on the nature of phenomenon being sensed. The sensed information must be retrieved within a finite time in order to avoid overflow of sensor memory and to keep the delay in relaying information to the source low. The time interval within which the information must be retrieved corresponds to the event departure.

## 1.1 Contributions

1. We provide analytical results on how the quality of coverage in mobile sensor networks scale with the number of mobile sensors, their velocity, velocity pattern and event dynamics. In this paper we analyze the case where the PoIs are located on a simple closed curve and the sensors move along the curve. However, the methodology used for the analysis may be easily extended to any general case, where the locations of PoIs and the trajectory of mobile sensors are known. These results may serve as guidelines for deploying a mobile sensor network.

2. We formulate the bounded event loss probability (BELP) problem to satisfy the quality of coverage at PoIs. We consider two versions of BELP: (i) minimum velocity BELP (MV-BELP) and (ii) minimum sensors BELP (MS-BELP).

3. For special cases where the sensors move only along the line or simple closed curve on which the PoIs are located, we present optimal algorithms for solving the MV-BELP problem.

4. For MS-BELP problem, we present approximate algorithms for the above-mentioned special cases. The number of sensors used by approximate algorithms is shown to be within factor 2 of the optimal solution.

5. Although the general BELP problem (where PoIs are arbitrarily placed on a plane and mobile sensors may move in unrestricted fashion) is still under study, we present heuristic algorithms for MS-BELP and MV-BELP problems and investigate their deviation from the optimal solutions.

## 1.2 Paper Outline

In the next section we present a brief overview of the related work. The analysis of how the fraction of events captured vary with the parameters of the network and event dynamics is presented in Section 3. The BELP problem is presented and discussed in detail in Section 4. The algorithms for the BELP problem, in three scenarios with increasing order of difficulty, are presented in Sections 5-7. We summarize our results and discuss the future work in Section 8.

# 2. RELATED WORK

Considerable research effort has been invested recently in studying coverage properties of static sensor networks [15, 26, 35, 36] and path planning for mobile robots [19, 20]. However the effects of mobility on coverage and the trade-offs involved have not been sufficiently studied. In [26], the authors study the coverage problem by using computational geometry and graph theoretic techniques, and propose optimal polynomial time worst case and average case algorithms for calculating coverage. In [29], the authors study unreliable sensor grids and derive necessary and sufficient condition on the probability of sensor failure and the sensing area that ensures coverage along with maintaining connectivity. Energy efficient coverage in wireless sensor networks is studied in [11, 17, 32, 38] and references therein. The principles of coverage are applied to develop mechanisms for exposing the path of a moving target in [8, 25].

In recent years there has been interest in understanding how the coverage properties of a sensor network may be improved by introducing mobility to the sensor devices. In [14] and [39], the authors propose virtual force based algorithms in order to guide sensor movements for improving the coverage properties after random deployment. In [34], the authors propose algorithms to detect the vacancies in a sensor field and use them to guide sensor motion in order to increase coverage. The average area covered by mobile sensors over a period of time has been characterized in [22]. It is shown that for a mobile sensor network with density $\lambda$, with each sensor moving according to a mobility model similar to random walk with expected velocity $E[V_s]$, the expected area covered in time interval $(0, t)$ is given by $1 - \exp\left(-\lambda(\pi r^2 + 2rE[V_s]t)\right)$.

Online algorithms for allocating tasks to mobile sensors in a hybrid sensor network, consisting of mobile and static sensors, is studied in [3–5]. The tasks to be served by mobile sensors appear at random location within the sensor field. The static sensors become aware of the arrival of the task and they guide the mobile sensors to the position where the task occurs. However, in our system model, the mobile sensors only have information about the stochastic nature of the arrival of events. Also we do not have any "guides" that know the global system state. In this paper we consider path planning without help of any guides and based only on the stochastic nature of the events being sensed.

# 3. FRACTIONS OF EVENTS CAPTURED

In this section we present an analysis of coverage quality of mobile sensor networks, in terms of the *fraction of events captured* by the mobile sensors, for a simplistic scenario. This analysis may serve as guideline to differentiate the cases where mobility is helpful from the cases where it is not. We consider $a$ PoIs, numbered 1 through $a$, scattered along a simple closed curve $\mathcal{C}$ of length $D$. The mobile sensors are allowed to move along the curve $\mathcal{C}$ only, e.g. the closed curve may be a circular corridor and the PoI may be doors that open into the corridor. The analysis presented in this section may be easily extended to arbitrary location of PoIs, provided the locations and the path traversed by the sensors is known. We present results for the closed curve case because they are easier to understand and interpret. The sensor motion strategy considered in this section is continuous traversal of $\mathcal{C}$ in counter-clockwise direction. The mobile sensors can sense the event at a PoI (the PoI is *visible* to the sensor) if the distance between the sensor and the PoI along $\mathcal{C}$ is less than $r$. This scenario is illustrated in Figure 1.

The state of each PoI alternates between 0 and 1. State 1 corresponds to a event being present at a PoI while state 0 corresponds to no event. The times spent by a PoI $i$ in state 0 and 1 are exponentially distributed with means $\frac{1}{\lambda_i}$ and $\frac{1}{\mu_i}$ respectively. Thus $(\lambda_i, \mu_i)$ characterize the *event dynamics* at PoI $i$. For the analysis in this section we assume that $\lambda_i = \lambda$ and $\mu_i = \mu \; \forall \, i$.

We consider three different types of sensor deployments for this

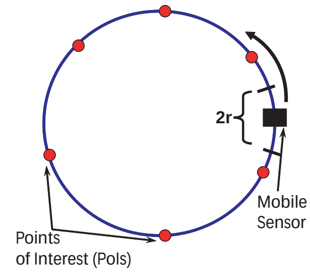



**Figure 1: Analytical model**

scenario:

1. *Single sensor moving with constant velocity*: For this case we determine the expected fraction of events captured as a function of event dynamics ($\lambda$, $\mu$) and sensor parameters ($r$, $v$). We characterize the cases where mobility *decreases* the quality of coverage.

2. *Multiple sensors moving with fixed velocity*: For this case we characterize the gain of deploying multiple mobile sensors. The sensors move along $\mathcal{C}$ with velocity $v$, while remaining equidistant from each other.

3. *Single sensor moving with variable speed*: Here we consider sensors that are able to move with any velocity between 0 and $v_{\max}$. For this case we characterize the optimal velocity pattern that the mobile sensor must use in order to maximize the fraction of events covered.

## 3.1 Single Sensor Fixed Velocity Case

A *state cycle* is a $0 \to 1 \to 0$ or $1 \to 0 \to 1$ cycle of the state of a PoI. During each round trip around $\mathcal{C}$, the sensor visits each PoI exactly once. The time for which a PoI is visible to the mobile sensor during a round trip around $\mathcal{C}$ equals $\frac{2r}{v}$ seconds. So if the PoI became visible to the sensor at time $t$, the sensor would capture any event at the PoI in the time interval $[t, t + \frac{2r}{v}]$. The number of events captured by the sensor during a visit depends upon the state of the PoI at the beginning of the visit and the number of state cycles during the duration of the visit.

LEMMA 1. *Let* $C(\tau)$ *denote the number of* state cycles *observed at a PoI during time* $(t, t+\tau)$. *Then*

$$E[C(\tau)] = \frac{\lambda\mu}{\lambda+\mu}\left(\tau - \frac{1}{\lambda+\mu}\left(1 - \exp(-(\lambda+\mu)\tau)\right)\right) \quad (1)$$

PROOF. The state cycle is a renewal process whose inter arrival time is the sum of two exponential random variables. According to [12], the Laplace transform of the expected number of renewals in time $\tau$, $L_{\mathcal{N}}(r)$, is given by

$$L_{\mathcal{N}}(r) = \frac{L_{\mathcal{F}}}{r(1 - L_{\mathcal{F}}(r))} \quad (2)$$

where $L_{\mathcal{F}}(r)$ is the Laplace transform of the pdf of inter arrival time of the renewal process.

Let $T$ denote the inter arrival time of the state cycles, then $T = T_1 + T_2$, where $T_1$ and $T_2$ are exponential distributions with mean $\frac{1}{\mu}$ and $\frac{1}{\lambda}$ respectively. Then $P[T \le t]$ is given by

$$P[T \le t] = F_T(t) = 1 - \frac{1}{\lambda-\mu}\left(\lambda\exp(-\mu t) - \mu\exp(-\lambda t)\right) \quad (3)$$

Thus the pdf of $T$ is given by

$$f_T(t) = \frac{\mathrm{d}F_T(t)}{\mathrm{d}t} = \frac{\lambda\mu}{\lambda-\mu}\left(\exp(-\mu t) - \exp(-\lambda t)\right) \quad (4)$$

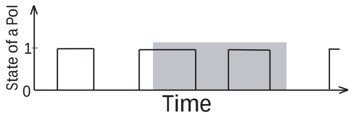



**Figure 2: The gray area indicates the duration of a visit to a PoI.**

$L_T(r)$, the Laplace transform of $f_T(t)$, is given by

$$L_T(r) = \int_0^\infty f_T(t)\exp(-rt)\mathrm{d}t = \frac{\lambda\mu}{(r+\mu)(r+\lambda)} \quad (5)$$

Let $L_C(r)$ denote the Laplace transform of $E[C(\tau)]$. According to (2), $L_C(r)$ is given by

$$L_C(r) = \frac{L_T(r)}{r(1-L_T(r))}$$

Therefore,

$$L_C(r) = \frac{\lambda}{\lambda+\mu}\frac{1}{r^2} - \frac{\lambda\mu}{(\lambda+\mu)^2}\frac{1}{r} + \frac{\lambda\mu}{(\lambda+\mu)^2}\frac{1}{r+\mu+\lambda} \quad (6)$$

So $E[C(\tau)] = L^{-1}(L_C(r))$, where $L^{-1}$ is the inverse Laplace transform, which directly leads to (1). $\square$

Let $S_i(t)$ denote the state of PoI $i$ at time $t$. From the analysis of a two state Markov chain it follows that

$$P[S_i(t)=0] = \frac{\mu}{\mu+\lambda} \text{ and } P[S_i(t)=1] = \frac{\lambda}{\mu+\lambda} \quad (7)$$

LEMMA 2. *Let $N_i(t, t+\frac{2r}{v})$ denote the number of events captured by the mobile sensor during a visit to a PoI $i$ that started at time $t$. Then*

$$\begin{aligned} E[N_i(t,t+\frac{2r}{v})] &= \frac{\lambda}{\lambda+\mu}\Bigg(1-e^{(-\mu\frac{D-2r}{v})} + \frac{\lambda\mu}{\lambda+\mu}\times\Bigg(\frac{2r}{v}-\frac{1}{\lambda+\mu} \\ &\quad (1-e^{(-(\lambda+\mu)\frac{2r}{v})})\Bigg)\Bigg) + \frac{\mu}{\lambda+\mu}\Bigg(\Big(1-e^{(-\lambda\frac{2r}{v})}\Big)\times \\ &\Big[1+\frac{\lambda\mu}{\lambda+\mu}\frac{2r}{v}-\frac{\lambda\mu}{(\lambda+\mu)^2}\Big] - \frac{\lambda\mu}{\lambda+\mu}\Big[\frac{1}{\lambda}-\Big(\frac{2r}{v}+\frac{1}{\lambda}\Big)e^{(-\lambda\frac{2r}{v})}\Big] \\ &\quad + \frac{\lambda^2}{(\lambda+\mu)^2}\Big[e^{(-\lambda\frac{2r}{v})}-e^{(-(\lambda+\mu)\frac{2r}{v})}\Big]\Bigg) \end{aligned} \quad (8)$$

PROOF. We proceed by evaluating $E[N_i(t,t+\frac{2r}{v})|S_i(t)=1]$ and $E[N_i(t,t+\frac{2r}{v})|S_i(t)=0]$ and combining them using (7) to find $E[N_i(t,t+\frac{2r}{v})]$.

Figure 2 illustrates the states of a PoI observed by a sensor during a visit to the PoI. It should be noted that when the state of a PoI at the beginning of a visit is 1, the number of events captured during the visit equals $1 + C(2r/v)$ i.e. number of $1 \to 0 \to 1$ cycles during the visit duration plus 1. However, the event captured at the beginning of the event may be the same as the event captured by the robot while leaving the PoI during the last visit. The probability that this is true, denoted by $P_r$, is given by

$$P_r = P\left[S_i(t') = 1 \ \forall \ t - \frac{D-2r}{v} \le t' \le t\right] = e^{(-\mu\frac{D-2r}{v})} \quad (9)$$

The expected number of distinct events captured by a sensor during a visit, given that the state of the PoI at the beginning of the visit is 1, is given by

$$\begin{aligned} E\left[N_i(t,t+\frac{2r}{v})|S_i(t)=1\right] &= 1 - P_r + E\left[C\left[\frac{2r}{v}\right]\right] \\ = 1 - e^{(-\mu\frac{D-2r}{v})} + \frac{\lambda\mu}{\lambda+\mu}&\left(\frac{2r}{v} - \frac{1}{\lambda+\mu}(1-e^{(-(\lambda+\mu)\frac{2r}{v})})\right) \end{aligned} \quad (10)$$

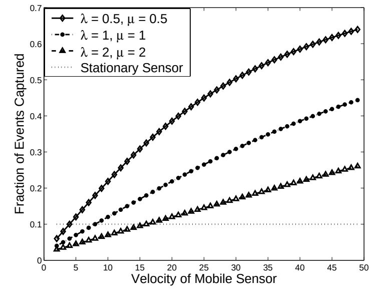



**Figure 3: Fraction of events captured against velocity of the robot for different sets of $\lambda$ and $\mu$.**

Now consider the case when the state of PoI is 0 at the beginning of a visit. If the state of the PoI flips from 0 to 1 at some time $t'$ during the visit, the number of events captured during the event equals $1 + C\left(t+\frac{2r}{v}-t'\right)$. If the state does not flip from 0 to 1 during the visit then the number of events captured during the visit equals 0. Thus $E[N_i(t,t+\frac{2r}{v})|S_i(t)=0]$ can be expressed as

$$\begin{aligned} &E[N_i(t,t+\frac{2r}{v})|S_i(t)=0] = \\ &\int_t^{t+\frac{2r}{v}} P[S_i(t')=0, S_i(t'+\mathrm{d}t')=1]\Big(1+E\Big[C(t+\frac{2r}{v}-t')\Big]\Big) \end{aligned}$$

Since the time spent by the PoI in state 0 is exponentially distributed with mean $\frac{1}{\lambda}$, $P[S_i(t')=0, S_i(t'+\mathrm{d}t')=1] = \lambda\exp(-\lambda(t'-t))\mathrm{d}t'$. Thus

$$\begin{aligned} &E[N_i(t,t+\frac{2r}{v})|S_i(t)=0] = \\ &\qquad \int_t^{t+\frac{2r}{v}} \lambda e^{(-\lambda(t'-t))}\cdot\Big(1+E\Big[C\Big(t+\frac{2r}{v}-t'\Big)\Big]\Big)\mathrm{d}t' \end{aligned} \quad (11)$$

Substituting $t_1 = t' - t$, we get

$$E[N_i(t,t+\frac{2r}{v})|S_i(t)=0] = \int_0^{\frac{2r}{v}} \lambda e^{(-\lambda t_1)}\Big(1+E\Big[C\Big(\frac{2r}{v}-t_1\Big)\Big]\Big)\mathrm{d}t_1 \quad (12)$$

Thus

$$\begin{aligned} E[N_i(t,t+\frac{2r}{v})|S_i(t)=0] &= \Big(1-e^{(-\lambda\frac{2r}{v})}\Big)\Big[1+\frac{\lambda\mu}{\lambda+\mu}\frac{2r}{v}-\frac{\lambda\mu}{(\lambda+\mu)^2}\Big] - \\ \frac{\lambda\mu}{\lambda+\mu}\Big[\frac{1}{\lambda}-\Big(\frac{2r}{v}+\frac{1}{\lambda}\Big)e^{(-\lambda\frac{2r}{v})}\Big] &+ \frac{\lambda^2}{(\lambda+\mu)^2}\Big[e^{(-\lambda\frac{2r}{v})}-e^{(-(\lambda+\mu)\frac{2r}{v})}\Big] \end{aligned} \quad (13)$$

Combining (10) and (13) we have

$$\begin{aligned} E[N_i(t,t+\frac{2r}{v})] &= P[S_i(t)=1]\times E[N_i(t,t+\frac{2r}{v})|S_i(t)=1] \\ &\quad + P[S_i(t)=0]\times E[N_i(t,t+\frac{2r}{v})|S_i(t)=0] \end{aligned}$$

which yields (8). $\square$

LEMMA 3. *The expected number of events captured by the sensor in an entire round trip around $\mathcal{C}$, denoted by $N_{\mathrm{trip}}$, is given by*

$$N_{\mathrm{trip}} = \sum_{i=1}^{a} E[N_i(t,t+\frac{2r}{v})] = aE[N_1(t,t+\frac{2r}{v})] \quad (14)$$

Lemma 3 directly follows from the fact that all PoIs have i.i.d. event dynamics and are visited exactly once during a round trip.

THEOREM 1. *The expected fraction of events captured by a single sensor moving around $\mathcal{C}$ with velocity $v$, denoted by $F_1(v)$, is given by*

$$F_1(v) = \frac{v(\lambda+\mu)}{\lambda\mu D} E[N_1(t, t+\frac{2r}{v})] \tag{15}$$

PROOF. Let $N_{T_\infty}$ denote the expected number of events captured by the sensor in time $T_\infty$, where $T_\infty \to \infty$. $N_{T_\infty}$ is equal to the product of number of round trips completed by the sensor in time $T_\infty$ and the expected number of events captured during a round trip. Therefore

$$N_{T_\infty} = \frac{vT_\infty}{D} N_{\text{trip}} = a\frac{vT_\infty}{D} E[N_1(t, t+\frac{2r}{v})] \tag{16}$$

The actual number of events that occur at the PoIs, denoted by $N'_{T_\infty}$ is equal to $aC(T_\infty)$. For $T_\infty \to \infty$, it is equal to

$$N'_{T_\infty} = a\frac{\lambda\mu}{\lambda+\mu} T_\infty \tag{17}$$

Equation (15) is obtained using $F_1(v) = N_{T_\infty}/N'_{T_\infty}$ □

Using (15) we may answer questions like what is the effect of mobility on quality of coverage? Or, what are the gains achieved by a mobile sensor over a stationary one? To answer these questions, consider a situation where the PoIs are located is such a manner so that only one of them could be covered by a stationary sensor at any given time i.e. the distance between any two PoIs is more than $2r$. So if a stationary sensor is deployed, then the fraction of events captured by the stationary sensor is simply $1/a$. Therefore the mobility is useful only if $F_1(v) > 1/a$. Figure 3 shows the plot of $F_1(v)$ against $v$ for events with varying dynamics. Here $D = 50$, $r = 1$ and $a = 10$. Also plotted alongside is the fraction of events covered if a stationary sensor was deployed. Through this plot it is easy to see that if the mobile sensor moves slowly then its quality of coverage is in fact worse than that of a stationary sensor. The critical velocity required achieve a better QoC than a stationary sensor increases as the rate of arrival and departure of events increases.

## 3.2 Multiple Sensors, Fixed Velocity

In this subsection we evaluate the quality of coverage for multiple mobile sensors. We consider $m$ mobile sensors each moving around $\mathcal{C}$ with velocity $v$. The distance between two adjacent sensors is assumed to be the same and equal to $D/m$. The case where distance between adjacent mobile sensors is less than $2r$, i.e. $D/m \le 2r$, is trivial since in this case each PoI would always be seen by one of the mobile sensors and hence all events would be captured. We focus our attention on the case where $D/m > 2r$ i.e. $m < D/2r$.

THEOREM 2. *Let $F_m(v)$ denote the fraction of events captured by $m$ sensors moving around $\mathcal{C}$ with velocity $v$. Then*

$$F_m(v) = \frac{mv(\lambda+\mu)}{\lambda\mu D} E[N_{11}(t, t+\frac{2r}{v})] \tag{18}$$

*where*

$$E[N_{11}(t, t+\frac{2r}{v}] = \frac{\lambda}{\lambda+\mu}\left(1-e^{(-\mu\frac{D/m-2r}{v})}+\frac{\lambda\mu}{\lambda+\mu}\left(\frac{2r}{v}-\frac{1}{\lambda+\mu}\times (1-e^{-(\lambda+\mu)\frac{2r}{v}})\right)\right)+\frac{\mu}{\lambda+\mu}\left((1-e^{(-\lambda\frac{2r}{v})})\left[1+\frac{\lambda\mu}{\lambda+\mu}\frac{2r}{v}-\frac{\lambda\mu}{(\lambda+\mu)^2}\right]- \frac{\lambda\mu}{\lambda+\mu}\left[\frac{1}{\lambda}-\left(\frac{2r}{v}+\frac{1}{\lambda}\right)e^{(-\lambda\frac{2r}{v})}\right]+\frac{\lambda^2}{(\lambda+\mu)^2}\left[e^{-\lambda\frac{2r}{v}}-e^{-(\lambda+\mu)\frac{2r}{v}}\right]\right) \tag{19}$$

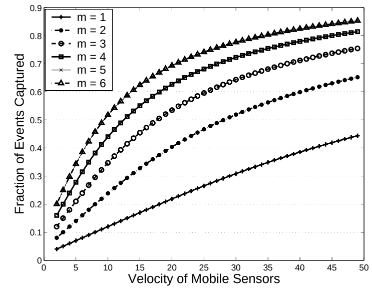



**Figure 4: Fraction of events captured against velocity of the sensing robots for various numbers of deployed sensors.**

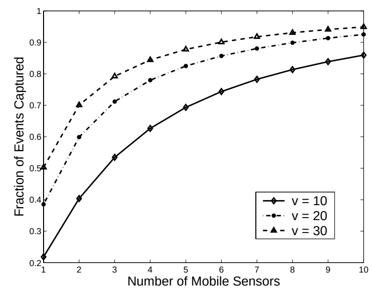



**Figure 5: Fraction of events captured against number of mobile sensors deployed for various velocities.**

The proof of Theorem 2 is similar to that of Theorem 1 and is presented in a related technical report [7].

Figure 4 shows the plots of fraction of events captured against velocity for various values of $m$. For these plots, $D = 100$, $r = 1$, $a = 10$ and $\lambda = \mu = 1.0$. It is observed that the gain in performance does not commensurate with the increase in number of sensors after a certain threshold. In fact the fraction of events covered for $m = 5$ and $m = 6$ do not differ significantly. The dotted lines in Figure 4 show the fraction of events captured if the sensors were stationary. It is observed that the gains of mobility are higher when number of sensors deployed are less. Figure 5 shows the plots of fraction of events captured against number of sensors deployed. For this plot $\lambda = \mu = 0.5$. It is observed that the gains of increasing the number of sensors diminish with higher speed.

## 3.3 Single Sensor, Variable Velocity

As observed in the previous subsections, the quality of coverage of mobile sensors with low velocity is worse than that of stationary sensors. This is because the slow sensors spend most of the times traveling around regions of $\mathcal{C}$ where no PoIs can be seen. We refer to the union of such regions as *futile regions*. High velocity enables the sensors to cover the futile regions in shorter time. However at high velocity the duration of a visit to a PoI is also decreased. This reduces the number of events that the sensor can capture during a visit to the PoI. Intuitively it is appealing that mobile sensors slow down while any PoI is visible, to increase the number of events captured during that visit, and move at the maximum speed in the futile regions. In this subsection investigate the gains, if any, of varying sensor speed in this fashion.

We consider one mobile sensor capable of moving at any speed up to $v_{max}$. Since there is no incentive to slow down in the futile regions, the sensor moves with velocity $v_{max}$ in the futile regions. However, while a PoI is visible to the sensor it moves with speed $v_c \in (0, v_{max}]$. We refer to $v_c$ as *capture speed*.

For a mobility pattern, the time taken by the sensor to move around $\mathcal{C}$ depends on length of the futile region which in turn depends on the location of PoIs on $\mathcal{C}$. Although we can analyze this case for a particular PoI placement, in order to be able to obtain general results we consider random placement of PoIs along $\mathcal{C}$. Assume that the distance of a PoI (along $\mathcal{C}$) from a reference PoI is uniformly distributed between 0 and $D$. For such a random placement we find expressions for expected round trip time and expected inter-visit time and use that to evaluate the fraction of events captured for a given motion strategy.

THEOREM 3. *The fraction of events captured by the mobile sensor moving with variable speed, denoted by $F_v(v_c)$, is given by*

$$F_v(v_c) = \frac{\lambda+\mu}{\lambda\mu T_{\text{trip}}} E\big[N_{1v}\big(t, t+\frac{2r}{v_c}\big)\big] \tag{20}$$

*Where,*

$$E[N_{1v}(t, t+\frac{2r}{v_c})] = \frac{\lambda}{\lambda+\mu}\Big(1 - e^{-\mu T_{\text{visit}}} + \frac{\lambda\mu}{\lambda+\mu}\Big(\frac{2r}{v_c} - \frac{1}{\lambda+\mu}\times (1 - e^{-(\lambda+\mu)\frac{2r}{v_c}})\Big)\Big) + \frac{\mu}{\lambda+\mu}\Big((1-e^{-\lambda\frac{2r}{v_c}})\Big[1+\frac{\lambda\mu}{\lambda+\mu}\Big] - \frac{\lambda\mu}{\lambda+\mu}\Big[\frac{1}{\lambda} - \Big(\frac{2r}{v_c}+\frac{1}{\lambda}\Big)e^{-\lambda\frac{2r}{v_c}}\Big]\Big) \tag{21}$$

*where,*

$$T_{\text{trip}} = \frac{E[W]}{v_{max}} + \frac{D - E[W]}{v_c},\ T_{\text{visit}} = \frac{E[W]}{v_{max}} + \frac{D - E[W] - 2r}{v_c}$$

$$\text{and } E[W] = \Big(1 - \frac{2r}{D}\Big)^{a-1}(D-2r)$$

A detailed proof of Theorem 3 is presented in [7].

There exists an optimal capture velocity if there exists a capture velocity $v_c = v_c^\star$ such that $\mathrm{d}F_v(v_c)/\mathrm{d}v_c = 0$ at $v_c^\star$. Unfortunately $\mathrm{d}F_v(v_c)/\mathrm{d}v_c = 0$ at $v_c^\star$ is too complex to be solved explicitly. So we present numerical computation results to show how $F_v(v_c)$ varies with $v_c$. Figure 6 shows plots of $F_v(v_c)$ against capture velocity for various number of PoIs. For this plot $v_{max} = 40$ m/s, $D = 100$, $r = 1$ and $\mu = \lambda = 1.0$. It is observed that only for $a = 2$, it is advantageous to have $v_c < v_{max}$. If number of PoIs equals $a$, then in general the sensor is missing out events on about $a - 1$ PoIs while it is visiting a PoI. So if $a$ is large enough then the sensor might miss a large number of events if it spends a lot of time during a visit to a PoI. For $a = 2$, when the sensor slows down while sensing a PoI it misses out events at only one PoI. However it makes up for the lost events at the other PoI by sensing more events at the PoI being currently visited. Thus $a = 2$ is the only case where slowing down while visiting a PoI may not be disadvantageous. In general, the best policy is to keep moving with the maximum possible speed.

# 4. BOUNDED EVENT LOSS PROBABILITY (BELP) PROBLEM

In the previous section we found expressions for the quality of coverage of mobile sensors in terms of expected fraction of events captured. In this section we consider a stricter quality of coverage metric: *event loss probability*. We consider a set of PoIs $\mathcal{S}$, such that $|\mathcal{S}| = a$. Each PoI has event dynamics $\lambda_i$ and $\mu_i$ and is located at $X_i$. Let $\mathcal{E}_i$ denote the event that an event occurs at PoI $i$ and is not captured by any of the mobile sensors. The goal is to generate a motion plan for the mobile sensors such that

$$P[\mathcal{E}_i] < \epsilon \ \ \forall\ 1 \le i \le a \tag{22}$$

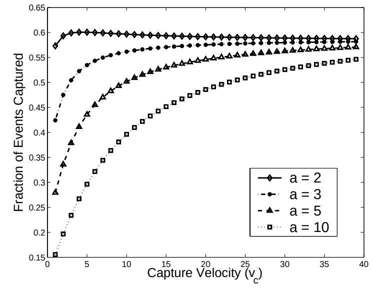



**Figure 6: Fraction of events captured against capture velocity for various values of $a$.**

i.e. probability that any event is not captured is bounded by $\epsilon$. This problem is referred to as the *Bounded Event Loss Probability (BELP)* problem.

Note that the constraint on probability of event being missed is a stronger condition than the fraction of events captured. For example, if events rarely occur at a PoI that is far from the rest of the PoIs then a large fraction of events can be covered by ignoring that PoI completely. However if a bound on probability of event loss has to be maintained for all PoIs then no PoI may be completely ignored. Also, if (22) is satisfied then the fraction of events captured would be at least $1 - \epsilon$.

We now discuss the characteristics of BELP solution and how hard it is to find one. The event loss probability at PoI $i$, $P[\mathcal{E}_i]$, depends on the time between two consecutive visits to the PoI. The following lemma gives the relationship between the event loss probability and inter visit time.

LEMMA 4. *Let $T$ denote the time between two consecutive visits to a PoI with event dynamics $\lambda$ and $\mu$. Then the probability that an event is lost between the visits, $\mathcal{P}(T, \lambda, \mu)$, is given by*

$$\mathcal{P}(T,\lambda,\mu) = 1 - \frac{\mu}{\mu^2-\lambda^2}(\mu e^{-\lambda T} - \lambda e^{-\mu T}) - \frac{\lambda e^{-\mu T}}{\lambda+\mu} - \frac{\mu\lambda}{\mu^2-\lambda^2}\Big(\frac{\mu}{\mu-\lambda}e^{-\lambda T} - \frac{\lambda}{\mu-\lambda}e^{-\mu T} - \lambda T e^{-\mu T}\Big) \tag{23}$$

PROOF. If the PoI was in state 0 at the end of first visit then no event is lost if $T$ is less than the time it takes to complete a $0 \to 1 \to 0$ cycle. According to (3), this happens with probability $(\lambda e^{-\mu T} - \mu e^{-\lambda T})/(\lambda - \mu)$. Therefore, $P[\text{loss}|0]$, is given by

$$P[\text{loss}|0] = P[\text{cycle in } T] = 1 - \frac{1}{\lambda-\mu}(\lambda e^{-\mu T} - \mu e^{-\lambda T})$$

On the other hand if the state of PoI at the end of first visit was 1 then the loss probability, $P[\text{loss}|1]$, is given by

$$P[\text{loss}|1] = \int_0^T P[1 \to 0 \text{ transition at } \tau] P[\text{cycle in } T - \tau]$$
$$= 1 - e^{-\mu T} - \frac{\mu}{\mu-\lambda}\Big(\frac{\mu}{\mu-\lambda}e^{-\lambda T} - \frac{\lambda}{\mu-\lambda}e^{-\mu T} - \lambda T e^{-\mu T}\Big) \tag{24}$$

$\mathcal{P}(T, \lambda, \mu)$ can now be determined using

$$\mathcal{P}(T,\lambda,\mu) = \frac{\mu}{\lambda+\mu}P[\text{loss}|0] + \frac{\lambda}{\lambda+\mu}P[\text{loss}|1]$$

which yields (23). □

Let $T_{\text{crit}_i}(\epsilon)$ denote the inter-visit time for PoI $i$ such that $\mathcal{P}(T_{\text{crit}_i}(\epsilon), \lambda_i, \mu_i) = \epsilon$. Since $\mathcal{P}(T, \lambda, \mu)$ is a strictly increasing function of $T$, $T_{\text{crit}_i}(\epsilon)$ is unique and $\mathcal{P}(T, \lambda_i, \mu_i) < \epsilon\ \forall\ T < T_{\text{crit}_i}(\epsilon)$. $T_{\text{crit}_i}(\epsilon)$ is referred to as the *critical time* of PoI $i$. Thus

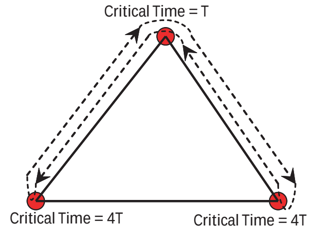



**Figure 7: The TSP path requires sensor to move with velocity $3/T$, while the optimal path shown with dashed line requires the sensor to move with $2/T$.**

the BELP problem boils down to finding mobility schedules for the mobile sensors such that the time between any two consecutive visits to each PoI is less than their respective critical times. For the rest of the paper we ignore the argument $\epsilon$ of the critical time and simply refer to it as $T_{\mathrm{crit}_i}$

We define $\mathcal{A}(i, j, v)$ as the *feasibility function*. $\mathcal{A}(i, j, v)$ is equal to 1 if it is feasible to solve BELP for PoIs $i$ and $j$ with a single mobile sensor having velocity $v$ and is 0 otherwise. In other words $\mathcal{A}(i, j, v)$ equals 1 if a mobile sensor moving back and forth between PoIs $i$ and $j$ visits each PoI ($i$ and $j$) at least once within their critical times. That is

$$\mathcal{A}(i, j, v) = \begin{cases} 1, & \text{if } \frac{2(|X_i - X_j| - 2r)}{v} < \min(T_{\mathrm{crit}_i}, T_{\mathrm{crit}_j}) \\ 0, & \text{otherwise} \end{cases}$$

where $|X_i - X_j|$ is the distance between PoIs $i$ and $j$.

A set $N \subseteq \mathcal{S}$ is said to be a *feasible set* for velocity $v$ if it is possible to solve BELP for $N$ with a single sensor having velocity $v$. The necessary condition for feasibility of a set is $\mathcal{A}(i, j, v) = 1 \; \forall \; i, j \in N$.

We consider two versions of BELP problem:

- *Minimum Velocity BELP (MV-BELP)* Given a set of PoIs, their locations and event dynamics, what is the *minimum velocity* with which a mobile sensor must move to satisfy (22).

- *Minimum Sensors BELP (MS-BELP)* Given a set of PoI, their locations and event dynamics, what is the *minimum number of mobile sensors*, each moving with velocity $v$, that need to be deployed so that (22) is satisfied.

It is easy to see that both versions of BELP are NP-hard problems. MV-BELP requires finding the optimal path that the mobile sensor must take to visit PoIs such that time elapsed between two consecutive visits is less than the critical time. The shortest path required to visit a set of points is a well studied problem, better known as Traveling Salesman Problem (TSP). TSP is known to be an NP-complete problem. When critical times of all PoIs is the same, the MV-BELP problem reduces to finding TSP path and setting the velocity of the sensor equal to length of TSP path divided by the critical time. Thus TSP is a special case of MV-BELP problem and thus at least as hard as TSP. It is not always true that the TSP path is the optimal path for MV-BELP problem. This is made clear in Figure 7. The solid path in the figure is the TSP path while the dashed path is the optimal path.

For $v = 0$ MS-BELP is reduced to the minimum set cover problem, which is an NP-hard problem. Since set cover is a special case of MS-BELP problem, MS-BELP is at least as hard as set cover. When $v > 0$ let $C(v)$ denote the collection of all feasible subsets of $\mathcal{S}$ for a mobile sensor traveling with velocity $v$. The MS-BELP problem is to find the collection of feasible subsets $C' \subset C(v)$, of minimum cardinality, such that set $\mathcal{S}$ is covered by $C'$. Determining whether a subset is feasible is also a non-trivial problem since it requires finding optimal path for visiting the PoIs belonging to that set, which is also at least as hard as TSP.

According to the above discussion, it is not possible to develop polynomial time optimal algorithms for any of the above problems for general two dimensional placement of PoIs and sensor motion. However certain restrictions on placement of PoI and sensor motion allow us to efficiently solve both cases of BELP. We refer to such cases as *restricted* BELP.

In the next three sections we consider three different scenarios, in increasing order of difficulty. We present algorithms for MV-BELP and MS-BELP problems for each scenario. The three scenarios are

1. *Linear case*: All PoIs are located along a straight line and the mobile sensors can move only along the straight line.

2. *Closed curve case*: All the PoIs are located on a simple closed curve and the mobile sensors can move only along the closed curve.

3. *General 2-D case*: The PoIs are located on a 2-D plane and the mobile sensors are free to move on the plane in any arbitrary fashion.

The first two cases, namely linear and closed curve cases, put restriction on the paths used by the mobile sensor. For the MV-BELP problem this avoids the necessity of finding the optimal path, thus significantly reducing the complexity of the problem. For the MS-BELP problem, although the restriction on paths make it easier to verify whether a set of PoIs is feasible or not but still the problem of choosing the minimum number of feasible sets to cover all PoIs is still an instance of set cover problem. Thus restricted MS-BELP problem still remains a NP-hard problem. In real life the restrictions on paths traversed by sensors may often since it is not always to feasible/desirable to take any arbitrary path to visit the PoIs. It would be preferable that the sensors traverse along well trusted paths to avoid getting stuck or lost. For example, if there are bunch of PoIs at the outskirts of a dense forest area then it would be desirable that the sensor moves along the outskirts of the forest area rather than venture into the dense forest, where it runs higher risk of getting stuck, lost or damaged. We first present algorithms for linear case followed by those for the closed curve and general 2-D case.

*A note on the coverage model:*

In the basic BELP formulation, the only constraint is that a PoI must be covered by some sensor every $T_{\mathrm{crit}_i}$ time units. Therefore, in the multiple sensor case, one might imagine solutions where there exists some PoIs, each of which is covered by multiple sensors. For example, a PoI may be covered by $k$ sensors in a round robin manner, such that each sensor visits the PoI within $kT_{crit_i}$ time units (The case $k = 2$ can be easily constructed for PoIs on the line). However, such a solution may not be desirable as it requires the sensors to move synchronously. If the sensors lose synchronicity, possibly when a sensor slows down or gets stuck for a short time, then a situation may arise where no sensor visits the PoI within some $T_{\mathrm{crit}_i}$ time units. Without a central controller, this loss of coverage can continue indefinitely. In order to avoid the synchronous motion requirement, in the rest of the paper we only consider solutions where each point is covered by a unique sensor.

## 5. BELP: THE LINEAR CASE

In this case the PoIs are located along a line. Let $X_i$ ($1 \leq i \leq a$) denote the position of PoI $i$. Without loss of generality, assume that PoIs are ordered in increasing order of their positions, i.e. $X_i > X_j$ if $i > j$, and that $X_1 = 0$.

## 5.1 The linear case - minimum velocity problem

**Algorithm 1** MVBELP_LINE($X$, $T$, $a$)

return $\max_{1\le i\le a} \max\left(\frac{2(X[i]-X[1]-2r)}{T[i]}, \frac{2(X[a]-X[i]-2r)}{T[i]}, 0\right)$

Algorithm 1 presents pseudo code for the optimal algorithm for the line case of the MV-BELP problem. MVBELP_LINE takes arrays of locations and critical times of PoIs ($X[i] = X_i$, $T[i] = T_{\text{crit}_i}$) along with the number of PoIs as input. It returns the minimum velocity with which the mobile sensor satisfies (22) while moving back and forth between points $x = r$ to points $x = X_a - r$. In doing so, the mobile sensor observes a PoI $i$ while moving from left to right (from $x = r$ to $x = X_a - r$) and again while moving from right to left (from $x = X_a - r$ to $x = r$). So the maximum time elapsed between two consecutive visits to PoI $i$ equals $\max\left(\frac{X_i - X_1 - 2r}{v}, \frac{X_a - X_i - 2r}{v}, 0\right)$. In order to satisfy BELP at PoI $i$ the velocity of mobile sensor must be greater than $v_{min_i} = \max\left(\frac{2(X[i]-X[0]-2r)}{T[i]}, \frac{2(X[a]-X[i]-2r)}{T[i]}, 0\right)$. MVBELP_LINE sets the velocity to be $\max_i v_{min_i}$, thus satisfying QoC at all PoIs.

THEOREM 4. *MVBELP_LINE returns the minimum velocity required to cover a set of PoIs along a line while satisfying (22).*

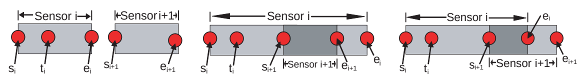
PROOF. Let $v_{\min}$ denote the velocity returned by MVBELP_LINE. ¿From the above discussion it is clear that the BELP is satisfied at all PoIs by a mobile sensor moving back and forth between $x = r$ and $x = X_a - r$ with velocity $v_{\min}$. We now show that $v_{\min}$ is optimal. According to MVBELP_LINE $v_{\min}$ is either 0 or equal to $\max\left(\frac{2(X_k - X_1 - 2r)}{T_{\text{crit}_k}}, \frac{2(X_a - X_k - 2r)}{T_{\text{crit}_k}}\right)$ for some $1 \le k \le a$. If $v_{\min} = 0$, then it is trivially optimal. Now assume that $v_{\min} \neq 0$ and there exists $v^\star < v_{\min}$ such that it possible for a mobile sensor to maintain the required quality of coverage while moving with velocity $v^\star$. However if this is the case then there exists a $1 \le k \le a$ such that $v^\star < \max\left(\frac{2(X_k - X_1 - 2r)}{T_{\text{crit}_k}}, \frac{2(X_a - X_k - 2r)}{T_{\text{crit}_k}}\right)$. It implies that $k$ cannot be covered along with either PoI 1 or $a$. This contradicts with the assumption that $v^\star$ satisfies the quality of coverage constraints. Thus $v_{\min}$ is the minimum velocity with which the quality of coverage can be maintained. ☐

## 5.2 The linear case - minimum sensor problem

Algorithm 2 presents a greedy algorithm for the MS-BELP problem for the line case. MSBELP_LINE takes the position, critical times and number of PoIs, along with the velocity of mobile sensors, as input. The array $\Gamma_i$ is the array of PoIs that are assigned to be covered by mobile sensor $i$. The algorithm starts allocation to $\Gamma_i$ by adding the leftmost PoI that has not been included in any other sensor to $\Gamma_i$. Then the algorithm sequentially looks at all the PoIs located to the right of $\Gamma_i[0]$. If it finds a PoI to the right of $\Gamma_i[0]$, that has not been allocated to any other sensor, say PoI $j$, then it inspects if QoC at $j$ may be maintained by sensor $i$ while maintaining QoC at all the PoIs already included in $\Gamma_i$. If $\Gamma_i + \{j\}$ is a feasible set then $j$ is added to $\Gamma_i$, otherwise the algorithm moves on to inspect the PoI to the right of $j$. When the algorithm has inspected all PoIs from $\Gamma_i[0]$ to $a$. If all PoIs have not been assigned to a mobile sensor, then the algorithm starts allocating the unassigned PoIs to $\Gamma_{i+1}$ in a similar fashion. When all the PoIs have been assigned to a mobile sensor, the algorithm returns the number of mobile sensors required to cover the given PoIs. The running time of the algorithm is $O(a^2)$.

**Algorithm 2** MSBELP_LINE($X$, $T$, $a$, $v$)

```
Set i = k = 0
While all PoIs not assigned
  k = k + 1; i = 0
  Γ_k[i] = leftmost PoI not yet assigned
  for j = Γ_k[0] + 1 to a
    if j not assigned and A(l, j, v) = 1 ∀ l ∈ Γ_k
      i = i + 1
      Γ_k[i] = j
    end if
  end for
end while
return k
```




(a) Disjoint (b) Complete overlap (c) Partial overlap

**Figure 8: All possible relationships between trajectories of two mobile sensors doing back and forth motion.**

In the rest of this paper the first (leftmost) PoI belonging to set $\Gamma_i$ referred to as $s_i$, while the last (rightmost) PoI is referred as $e_i$. That is, sensor $i$ sweeps the portion of line between PoI $s_i$ and $e_i$. If $|\Gamma_i| = 1$, $s_i = e_i$. This is illustrated in figure 5.2. We now prove some properties of the MSBELP_LINE algorithm that we will use in order to bound the performance of the algorithm with respect to the optimal.

**Property 1**: *For all $i$ ($1 \le i \le k$), the QoC of all PoIs belonging to $\Gamma_i$ is satisfied if a single sensor is deployed to cover the PoIs in $\Gamma_i$ ($k$ is the number of sensors used).*

PROOF. The MSBELP_LINE algorithm accepts a new PoI $l$ into the $\Gamma_i$ only if it is feasible to maintain QoC at all points already belonging to $\Gamma_i$ and at $l$ after including $l$ in $\Gamma_i$. Thus for all $l \in \Gamma_i$, $\mathcal{A}(s_i, l, v) = 1$ and $\mathcal{A}(l, e_i, v) = 1$. This implies that if a mobile sensor moves back and forth between $s_i$ and $e_i$, then each PoI belonging to $\Gamma_i$ is visited at least once during any time duration equal to its critical time. So a single sensor may satisfy QoC at all points belonging to $\Gamma_i$. ☐

This property implies that MSBELP_LINE divides PoIs into sets such that the QoC at all PoIs belonging to a set may be satisfied using a single mobile sensor.

**Property 2**: $s_i < s_j \ \forall\ 1 \le i < j \le k$.

This property easy to see since the first PoI added to a set $\Gamma_i$ is the leftmost PoI not yet assigned to any other set.

**Property 3**: *For all $i$ ($1 \le i \le k-1$), $\exists$ a PoI $t_i \in \Gamma_i$, such that $s_i \le t_i < s_{i+1}$ and $\mathcal{A}(t_i, s_{i+1}, v) = 0$. This implies that one or both of the following is true: (i) $\mathcal{A}(t_i, l, v) = 0 \ \forall\ l \ge s_{i+1}$, (ii) $\mathcal{A}(l, s_{i+1}, v) = 0 \ \forall\ l \le t_i$.*

PROOF. In order to add PoIs to $\Gamma_i$, MSBELP_LINE looks at the leftmost PoI that has not been assigned to any set and checks if it is feasible to cover that PoI along with all the PoIs already belonging to $\Gamma_i$. According to Property 2, $s_{i+1} > s_i$. Also we know that $s_{i+1}$ is not added to $\Gamma_i$ which implies that while constructing $\Gamma_i$, the algorithm did not find it feasible to add $s_{i+1}$ to $\Gamma_i$. In other words $\exists\ t_i \in \Gamma_i$ such that $t_i < s_{i+1}$ and $\mathcal{A}(t_i, s_{i+1}, v) = 0$. If such a $t_i$ did not exist then $s_{i+1}$ would have been added to $\Gamma_i$

Recall that $\mathcal{A}(t_i, s_{i+1}, v) = 0$ implies that one or both of the following is true: (i) $\frac{2(|X_{s_{i+1}} - X_{t_i}| - 2r)}{v} > T_{\text{crit}_{t_i}}$ or (ii)

$\frac{2(|X_{s_{i+1}} - X_{t_i}| - 2r)}{v} > T_{\mathrm{crit}_{s_{i+1}}}$. Note that condition (ii) is true for cases illustrated in Figures 8(b) and 8(c) while both conditions (i) and (ii) may be true for 8(a). If (i) is true then it implies that a mobile sensor cannot sense PoI $t_i$, then move to sense $s_{i+1}$ or any point located to the right of it and return to $t_i$ within the critical time of $t_i$. Thus it is infeasible to cover $t_i$ while covering $s_{i+1}$ or any other point located to the right of $s_{i+1}$. Similarly if (ii) is true, then it is infeasible to cover $s_{i+1}$ while covering $t_i$ or any point that lies to the left of $t_i$. This proves the above property. □

THEOREM 5. *Let $k_{\mathrm{OPT}}$ denote the minimum number of mobile sensors required to cover a set of PoIs and let $k$ denote the number of sensors used by MSBELP_LINE. Then*

$$k \leq 2k_{\mathrm{OPT}} + 1 \tag{25}$$

PROOF. ¿From property 3, $\forall\, i\ (1 \leq i \leq k-1)$, $\exists\, t_i \in \Gamma_i$ such that $s_i \leq t_i < s_{i+1}$ and

$$\mathcal{A}(t_i, l, v) = 0\ \forall\ l \geq s_{i+1} \tag{26}$$

Or,

$$\mathcal{A}(l, s_{i+1}, v) = 0\ \forall\ l \leq t_i \tag{27}$$

where $s_{i+1}$ is the leftmost PoI in $\Gamma_{i+1}$. Now we construct sets $H_1$ and $H_2$ in the following manner: For each $1 \leq i \leq k-1$ add $t_i$ to $H_1$, find $t_i$ s.t. $\mathcal{A}(t_i, s_{i+1}, v) = 0$. Property 3 implies that such a $t_i$ exists. For the $t_i$ and $s_{i+1}$ pair, add $t_i$ to $H_1$ if (26) holds and add $s_{i+1}$ to $H_2$ if (27) holds. Thus for each ($i$, $i+1$) pair, such that $1 \leq i \leq k-1$ at least one PoI is added to either $H_1$ or $H_2$. Therefore

$$|H_1| + |H_2| \geq k - 1 \tag{28}$$

We will now show that for all $l, m \in H_2$ $(l \neq m)$, $\mathcal{A}(l, m, v) = 0$. By definition of $H_2$, we know that there exist $i$ and $j$ ($i \neq j$, $2 \leq i, j \leq k$) such that $l = s_i$ and $m = s_j$. Let $i < j$, then $l = s_i \leq t_{j-1} < s_j = m$. From structure of $H_2$, we know that it is infeasible to cover $m$ while covering $t_{j-1}$ or any point to its left. Thus $\mathcal{A}(l, m, v) = \forall\ l, m \in H_2$. Similarly we can show that $\mathcal{A}(l, m, v) = 0\ \forall\ l, m \in H_1$. In other words it is infeasible to cover two points belonging to either $H_1$ or $H_2$ using a single mobile sensor. This property implies that

$$k_{\mathrm{OPT}} \geq \max(|H_1|, |H_2|) \tag{29}$$

If (29) is not true, then this implies that the optimal strategy would have to use one mobile to sense at least two PoIs belonging to $H_1$ or $H_2$. But this would lead to violation of QoC requirement at those points.

Also from (28),

$$\max(|H_1|, |H_2|) \geq (k-1)/2 \tag{30}$$

¿From (29) and (30), it follows that

$$k_{\mathrm{OPT}} \geq (k-1)/2 \tag{31}$$

Rearranging (31), we get (25). □

# 6. BELP: THE SIMPLE CLOSED CURVE CASE

In this subsection we present solutions to the MV and MS-BELP problems for the case where the mobile sensors are constrained to only move along a simple closed curve joining the PoIs. Consider a set of PoIs each with event dynamics $\lambda_i$, $\mu_i$ and critical time $T_{\mathrm{crit}_i}$ located on a simple closed curve $\mathcal{C}$. Without loss of generality, assume that the PoIs are numbered, 0 to $a-1$, with the PoI id increasing along the counter-clockwise direction. Let $l_{ij}$ denote the shortest distance traveled along $\mathcal{C}$ to reach PoI $j$ from PoI $i$ while traveling in counter-clockwise direction, and $D$ denote the total length of the curve. The algorithms for this case are based on the algorithms for the linear case.

**Algorithm 3** MVBELP_CURVE($L$, $T$, $a$)

```
Set minT = min_{0≤i≤a−1} T[i]
Set minV = D/minT
for i = 0 to a − 1
  for j = 0 to a − 1
    X[j + 1] = L[i][ mod_a(i + j)]
    T′[j + 1] = T[ mod_a(i + j)]
  end for
  V = MVBELP_LINE(X, T′, a)
  minV = min(minV, V)
end for
return minV
```

## 6.1 The closed curve case - minimum velocity problem

There are two possible paths that a mobile sensor may take in order to cover all the PoIs: (i) keep on traveling in a loop along $\mathcal{C}$, or (ii) move back and forth between PoIs $i$ and $\bmod_a(i-1)$[1] (i.e. $i$ to $\bmod_a(i-1)$ in counter-clockwise direction and $\bmod_a(i-1)$ to $i$ in clockwise direction) for some $0 \leq i \leq a$. The number of type (ii) paths equals $a$ while there is only one type (i) path. Thus the total number of possible paths that the mobile sensor may take equals $a+1$. The minimum speed of the mobile sensor if it takes type (i) path equals $\frac{D}{\min_i T_{\mathrm{crit}_i}}$. The minimum speed required for each of the type (ii) paths may be determined using MVBELP_LINE by opening the curve into a straight line, such that the PoI $i$ is at the left end of the line and PoI $\bmod_a(i-1)$ is on the right end of the line. The path, among the $a+1$ possible paths, that requires the least velocity is the optimal path that the mobile sensor must take and the velocity required to cover PoIs along that path is the optimal velocity.

This is the basis of the MVBELP_CURVE algorithm (Algorithm 3) that returns the optimal sensor velocity required to cover the PoIs while moving along a simple closed curve $\mathcal{C}$. The algorithm takes the relative distance $l_{ij}$ measured along $\mathcal{C}$ in counter-clockwise direction, the critical times of the PoIs and the number of PoIs as input. The algorithm then evaluates the minimum velocity required to cover the given PoIs along each of the possible paths. In order to evaluate the minimum velocity for the type (ii) paths, the algorithm creates a line topology, with $i$ at one end and $\bmod_a(i-1)$ at the other end for each $0 \leq i \leq a-1$, which corresponds to the back and forth motion of sensor between PoI $i$ and $\bmod_a(i-1)$. This line topology is passed to MVBELP_LINE that returns the minimum velocity required to cover the PoIs using that path. The algorithm obviously returns the optimal velocity since it compares the optimal velocity for all possible paths and returns the minimum velocity possible. The running time of MVBELP_CURVE is $O(a^2)$.

## 6.2 The closed curve case - minimum sensor problem

The PoIs may be divided into the following two categories, $\mathcal{S}_1$ and $\mathcal{S}_2$, depending on their critical time: (i) $i \in \mathcal{S}_1$ if $T_{\mathrm{crit}_i} < D/v$,

[1]The function $\bmod_a(x)$ returns the value of $x$ modulo $a$, i.e. remainder of $\frac{x}{a}$.

or (ii) $i \in \mathcal{S}_2$ if $T_{\text{crit}_i} \geq D/v$, i.e. $S_2$ is the set of PoIs whose QoC may be maintained by a mobile sensor traveling in a loop around $\mathcal{C}$.

We first consider solution for case where $\mathcal{S}_2 = \phi$. For this case we will show that there exists an optimal solution that is also the optimal solution for some linear topology obtained by opening the curve into a line. Thus greedy algorithm that we developed for the linear case in the last section may be used to allocate the PoIs to mobile sensors.

If $\mathcal{S}_2 = \phi$ then there exists no PoIs whose QoC can be satisfied by a mobile sensor moving around the curve $\mathcal{C}$ and therefore all sensors in this case would perform back and forth motion between a pair of PoIs. Similar to the definition in the previous section, let $s_k$ and $e_k$ denote the PoIs located at the extreme ends of curve swept by sensor $k$ such that the sensor moves between $s_k$ and $e_k$ in counter-clockwise manner. For two mobile sensors $k'$ and $k$, one of the following three cases is true: (i) $s_{k'} \leq e_{k'} < s_k \leq e_k$ - i.e. the curve swept by the sensors is disjoint (Figure 8(a)), (ii) $s_{k'} < s_k \leq e_k < e_{k'}$ - i.e. the curve swept by sensor $k$ is completely contained within the curve swept by sensor $k'$ (Figure 8(b)), (iii) $s_{k'} < s_k < e_{k'} < e_k$ - i.e. the curve swept by sensors $k'$ and $k$ partially overlaps (Figure 8(b)).

We will show that if the curves swept by two sensors overlap (case (iii)), then the PoIs may be reassigned to the sensors such that the curves swept by the sensors are disjoint, without introducing any extra mobile sensors. *This allows us to prove that there is an optimal allocation of the PoIs to mobile sensors such that the PoI $s_k$ is not visited by any other mobile sensor, for some mobile sensor $k$.* In other words, the curve swept by sensor $k$ is neither contained within the curve swept by any other sensor, nor does the curve swept by sensor $k$ partially overlaps with the curve swept by some other sensor. Thus the curve may be opened to form a line by fixing that PoI as the first PoI on the line and arranging all other PoIs along the line such that the relative counter-clockwise distance between the PoIs is preserved.

CLAIM 1. *If $\mathcal{S}_2 = \phi$, there exists an optimal solution such there exists a mobile sensor $k$ such no other mobile sensor passes through point $s_k$.*

PROOF. Suppose no such sensor exists in a optimal solution. Consider a sensor $k$ whose coverage curve partially overlaps with that of another sensor $k'$ such that $k'$ passes through $s_k$ (as shown in Figure 8(c)). We will now show that the coverage curves of the two sensors may be made disjoint without increasing the number of sensors deployed.

Starting from $s_k$ check to see if the PoIs covered by sensor $k$ and lying between $s_k$ and $e_{k'}$ may be covered by sensor $k'$. If all PoIs between $s_k$ and $e_{k'}$ can be covered by the sensor $k'$, then we have a new starting point, $s'_k$, for sensor $k$ such that no other sensor passes through it and we are done.

Now suppose all PoIs between $s_k$ and $e_{k'}$ cannot be covered by the sensor $k'$. This would imply that there exists a PoI, say $t_k$, between $s_k$ and $e_{k'}$ such that $t_k$ cannot be covered by $k'$. This would imply that the time taken for a round trip between $s_{k'}$ and $t_k$ is less than the critical time of $t_k$, that is

$$\frac{2(l_{s_{k'}t_k} - 2r)}{v} > T_{\text{crit}_{t_k}}$$

Now since $t_k$ is already covered by sensor $k$, it implies that the round trip time between $t_k$ and $e_k$ is less than the critical time of $t_k$.

$$\frac{2(l_{s_{k'}t_k} - 2r)}{v} > T_{\text{crit}_{t_k}} \geq \frac{2(l_{t_k e_k} - 2r)}{v}$$

Now consider a PoI $d_{k'}$ between $t_k$ and $e_{k'}$ (including $e_{k'}$) which is already covered by sensor $k'$. Since $d_{k'}$ is covered by $k'$, its critical time satisfies the following inequality

$$\frac{2(l_{d_{k'}e_k} - 2r)}{v} < T_{\text{crit}_{d'}} \leq \frac{2(l_{s_{k'}d_{k'}} - 2r)}{v}$$
$$\frac{2(l_{s_k d_{k'}} - 2r)}{v} < T_{\text{crit}_{d'}} < \frac{2(l_{s_{k'}d_{k'}} - 2r)}{v}$$

These inequalities imply that the QoC at all PoIs between $t_k$ and $e_{k'}$, that are originally covered by sensor $k'$, may be satisfied by sensor $k$. Therefore we can assign all the PoIs between $t_k$ and $e_{k'}$ to sensor $k$. This modification in the allocation satisfies QoC at all the PoIs. At the same time, $t_k$, the starting PoI of the new curve swept by sensor $k$ is visited by sensor $k$ only. Thus partially overlapping curves may be converting into non-overlapping curves without increasing the number of sensors deployed. □

Claim 1 implies that there exists an optimal solution such that no mobile sensor passes between some PoI $i$ and $\mod_a(i+1)$ $(0 \leq i \leq a-1)$. So if we know the PoI $i$ for which this is true, then we can find an assignment of the PoIs to the sensors in the following manner. We open the curve $\mathcal{C}$ to form a line topology with $\mod_a(i+1)$ at the left end and $i$ at the other, while preserving the distances between the PoIs. The optimal assignment for the case of PoIs on $\mathcal{C}$ would be same as that for this line topology, since we know that no sensor traverses the section between PoIs $i$ and $\mod_a(i+1)$. We then use MSBELP_LINE in order to find the optimal assignment for the new topology. From Theorem 5 it immediately follows that the number of sensors used by this algorithm would be at most twice the optimal plus one.

However we do not know which portion of $\mathcal{C}$ is not traversed by any mobile sensors in an optimal solution. So for each PoI $i$, we open $\mathcal{C}$ to form a line topology with $i$ at left end and $\mod_a(i+1)$ at the right end and run MSBELP_LINE for each such topology. The line topology that requires minimum number of sensors may be used for assignment of sensors to the PoIs. MSBELP_LINE is executed $a$ times in this solution.

Now consider the case where $\mathcal{S}_2 \neq \phi$. For this case there exist certain PoIs whose QoC may be satisfied by a sensor moving around $\mathcal{C}$. Note that only one sensor, moving around $\mathcal{C}$, is required to satisfy QoC at all PoIs belonging to $\mathcal{S}_2$. None of the PoIs belonging to $\mathcal{S}_1$ will have their QoC satisfied by a sensor moving around $\mathcal{C}$. The optimal strategy for this case would either have one or no sensor that goes around $\mathcal{C}$. We approach the assignment problem for this case in the following manner. First we find the number of sensors required if one sensor goes around $\mathcal{C}$ by running MSBELP_LINE $|\mathcal{S}_1|$ times over the PoIs in set $\mathcal{S}_1$ as described above. Then find number of sensors required if no mobile sensors goes around $\mathcal{C}$. This can be done by running MSBELP_LINE $a$ times as described above. Comparison of the number of sensors required for both cases would yield the number of sensors required using the greedy strategy and corresponding assignment of PoIs to sensors. Again the number of sensors used would be within two times the optimal plus one.

Algorithm 4, MSBELP_CURVE, finds the optimal number of sensors required to cover PoIs on a simple closed curve. The algorithm is based on the above discussions. The algorithm first finds the optimal assignment if none of mobile sensors circles around $\mathcal{C}$. This is done by creating a line topology for each PoI, with the PoI at the extreme left end and running MSBELP_LINE over it. Then the algorithm compares with assignment for the case where one mobile sensor is allowed to circle around $\mathcal{C}$. A line topology is created for each PoI that is not covered by the sensor circling $\mathcal{C}$. The line topology contains only those PoIs that are not covered by the

**Algorithm 4** MSBELP_CURVE($L, T, a, v$)

Set $minK = \infty$
for $i = 0$ to $a-1$
  for $j = 1$ to $a-1$
    $X[j] = L[i][\mod_a(i+j-1)]$
    $T'[j] = T[\mod_a(i+j-1)]$
  end for
  $K =$ MSBELP_LINE($X, T', a, v$)
  $minK = \min(minK, K)$
end for
for $i = 0$ to $a-1$
  if $T[i] < D/v$
    for $j = 1$ to $a-1$
      if $X[j] < D/v$
        $X[j] = L[i][\mod_a(i+j-1)]$
        $T'[j] = T[\mod_a(i+j-1)]$
      end if
    end for
    $K =$ MSBELP_LINE($X, T', a, v$)
    $minK = \min(minK, K+1)$
  end if
end for

circling sensor. MSBELP_LINE is run over each line topology to find the optimal assignment. If the critical times of all the nodes is less than $D/v$, then the second part of the algorithm is not needed. Since MSBELP_LINE runs in $O(a^2)$ time, MSBELP_CURVE runs in $O(a^3)$ time.

# 7. BELP: GENERAL 2-D CASE

As already mentioned, if the PoIs are scattered over a plane and the mobile sensors are allowed to move in an unconstrained fashion, the MV-BELP and MS-BELP problem are NP-complete and NP-hard respectively. Therefore in this section we suggest some heuristic algorithms for MV-BELP and MS-BELP problem.

## 7.1 General 2-D case - minimum velocity problem

The main hurdle in solving MV-BELP is finding the optimal path that covers all the PoIs. Therefore we focus on finding "good enough" path that visits all the PoIs and then set the velocity of the sensor equal to length of the path divided by the $\min_i T_{\text{crit}_i}$. One possible approach is to use solution of Traveling Salesman Problem with Neighborhoods (TSPN) [10] in order to find good paths to visit the PoIs.

The TSPN consists of a set of points and a neighborhood around the points. A point is said to be visited if we visit any point in its neighborhood. The TSPN problem is to find the shortest path a traveling salesman should take in order to visit all the points. For BELP problem, the neighborhoods of the PoIs are simply the disc of radius $r$ (sensing radius) around the PoIs. The TSPN itself is a NP-complete problem but fortunately there are many good approximate algorithms [10] that may be used to find the path.

The heuristic algorithm, MVBELP_2D, is thus summarized in Algorithm 5.

**Algorithm 5** MVBELP_2D

Calculate TSPN($\mathcal{S}$)
Return $\frac{|\text{TSPN}(\mathcal{S})|}{\min_i T_{\text{crit}_i}}$

THEOREM 6. *Let $v_{\text{TSPN}}$ denote the velocity returned by MV-BELP_2D and let $v^\star$ denote the optimal velocity, then*

$$\frac{v_{\text{TSPN}}}{v^\star} \leq \frac{T_{\max}}{T_{\min}} \tag{32}$$

*where $T_{\max} = \max_i T_{\text{crit}_i}$ and $T_{\min} = \min_i T_{\text{crit}_i}$.*

PROOF. The optimal algorithm would visit the neighborhoods of all PoIs at least once in time interval $(t, t+T_{\max})$, for all $t$. Thus the sensor would cover at least distance $|TSPN(\mathcal{S})|$ in the time period $(t, t+T_{\max})$. Therefore

$$v^\star \geq |TSPN(\mathcal{S})|/T_{\max}$$

The velocity returned by the heuristic algorithm equals $|TSPN(\mathcal{S})|/T_{\min}$. Therefore ratio $v_{\text{TSPN}}/v^\star$ is equal to

$$\frac{v_{\text{TSPN}}}{v^\star} \leq \frac{|TSPN(\mathcal{S})|/T_{\min}}{|TSPN(\mathcal{S})|/T_{\max}} = \frac{T_{\max}}{T_{\min}}$$

□

If $f(a)$ is the approximation ratio of the TSPN algorithm used in Algorithm 5, then $v_{\text{TSPN}}/v^\star \leq f(a)T_{\max}/T_{\min}$. If the TSPN algorithm proposed in [10] is used then $f(a) = k$, where $k$ is a constant independent of $a$. Thus for the case where critical times of all the PoIs is the same, we have a constant factor approximation algorithm for MVBELP problem.

## 7.2 General 2-D problem - minimum sensor problem

For the MS-BELP problem in 2-D we present a heuristic algorithm, MSBELP_2D (Algorithm 6), which is also based on the solution of TSPN problem. The algorithm calculates the TSPN path for visiting all the PoIs and uses MSBELP_CURVE to find the assignment of PoIs to mobile sensors if the sensors move only along the TSPN path.

**Algorithm 6** MSBELP_2D

Calculate TSPN($\mathcal{S}$)
Apply MSBELP_CURVE() over TSPN path

The following lemma justifies using TSPN as a subproblem for solving MS-BELP problem.

LEMMA 5. *Let $k_{\text{OPT}}$ denote the number of sensors used by an optimal solution and $r_{\max}$ denote the maximum distance between a pair of PoIs, then*

$$|\text{TSPN}(\mathcal{S})| \leq k_{\text{OPT}}(vT_{\max} + r_{\max}) \tag{33}$$

Using Lemma 5 we can bound the MSBELP_2D's deviation from the optimal value.

THEOREM 7. *Let $k$ denote the number of sensors used by MS-BELP_2D, then*

$$\frac{k}{k_{\text{OPT}}} \leq \frac{2T_{\max}}{T_{\min}} + \frac{2r_{\max}}{vT_{\min}} \tag{34}$$

The proofs of Lemma 5 and Theorem 7 are presented in [7].

As mentioned in the previous subsection, there are TSPN algorithms with constant approximation ratio. If such an algorithm is used for solving the TSPN in MSBELP_2D, then the bound in (34) would be simply scaled by the same constant.

The approximation ratio of MSBELP_2D depends not only on the ratio of critical times, but also on the relative location of PoIs and velocity of sensors. Favorable situations for applying the algorithm are: equal critical times, closely placed PoIs, large sensor velocity and large minimum critical time.

# 8. CONCLUSIONS AND FUTURE WORK

In this paper we studied the problem of providing quality of coverage using mobile sensors. We present analytical results that quantify the effect of controlled mobility on the fraction of events captured and how it is effected by the dynamics of phenomenon being covered. The analytical results provide guidelines for choosing the velocity and the number of sensors to be deployed for satisfying constraints on fraction of events captured.

We also studied the following motion planning problems in order to bound the probability of event loss: (i) Finding the minimum velocity for covering a set of PoIs with a single sensor (MV-BELP), and (ii) Finding the minimum number of sensors to be deployed if the velocity of each sensor is fixed (MS-BELP). The MV-BELP and MS-BELP problems are shown to be NP-hard. We provide optimal algorithms for the the special case of MV-BELP where the sensors are only allowed to move on the line or curve along which the PoIs are located. For the similar restricted case of MS-BELP problem we present an algorithm that uses at most $2 \cdot OPT + 1$ mobile sensors, where $OPT$ is the minimum number of mobile sensors used by an optimal algorithm. For the general version of MV-BELP and MS-BELP, where the PoIs are scattered over a plane, we present heuristic algorithms based on TSP problem and bound their performance with respect to the optimal solution.

The performance of the heuristic algorithm for MV-BELP presented in this paper depends on the ratio of critical times of the PoIs. This may be undesirable if the sensors are covering variety of events that have a large range of critical times. Even for a small number of PoIs the performance of the heuristic algorithm may be arbitrarily bad. The next step would be to develop good approximation algorithms for MV-BELP and MS-BELP problems whose approximation ratio would depend on the number of PoIs, rather than the ratio of critical times.

We did not consider the communication requirements of the mobile sensors in this paper. In many scenarios the sensors may require to communicate with each other or to relay the gathered information to a base station. Adding communication requirements is likely to constrain the solution further, thus our results in this paper may serve as a reference for the more constraint cases. Incorporating the communication requirements and collaboration of mobile sensors in the event capture problem is the focus of our future research.


## Acknowledgement

A.A. Abouzeid would like to thank Dr. Wesley H. Huang for initial discussions on an older version of the mobile coverage (which we termed "site visitation") problem.


# 9. REFERENCES


[1] James San Jacinto Mountain Reserve. http://www.jamesreserve.edu.

[2] I. F. Akyildiz, D. Pompili, and T. Melodia. Underwater acoustic sensor networks: research challenges. *Ad Hoc Networks*, 3(3):257–279, 2005.

[3] M. Batalin, W. Kaiser, R. Pon, G. S. Sukhatme, G. Pottie, Y. Yu, J. Gordon, M. H. Rahimi, and D. Estrin. Task allocation for event-aware spatiotemporal sampling of environmental variables. In *IROS*, pages 1846–1853, 2005.

[4] M. Batalin and G. S. Sukhatme. Sensor network-mediated multi-robot task allocation. In *The Third International Naval Research Laboratory Multi-Robot Systems Workshop*, pages 27–38, Mar 2005.

[5] M. A. Batalin, M. Rahimi, Y. Yu, D. Liu, A. Kansal, G. S. Sukhatme, W. J. Kaiser, M. Hansen, G. J. Pottie, M. Srivastava, and D. Estrin. Call and response: experiments in sampling the environment. In *SenSys '04*, pages 25–38, New York, NY, USA, 2004. ACM Press.

[6] S. Bergbreiter and K. Pister. Cotsbots: An off-the-shelf platform for distributed robotics. In *IROS*, pages 1632–1637, 2003.

[7] N. Bisnik, A. Abouzeid, and V. Isler. Event capture problem in mobile sensor networks. Technical Report, ECSE Dept., RPI. Available at http://www.ecse.rpi.edu/abouzeid/mobileCoverage.pdf.

[8] T. Clouqueur, V. Phipatanasuphorn, P. Ramanathan, and K. K. Saluja. Sensor deployment strategy for target detection. In *WSNA '02*, pages 42–48, 2002.

[9] K. Dantu, M. H. Rahimi, H. Shah, S. Babel, A. Dhariwal, and G. S. Sukhatme. Robomote: enabling mobility in sensor networks. In *IPSN*, pages 404–409, 2005.

[10] A. Dumitrescu and J. S. B. Mitchell. Approximation algorithms for tsp with neighborhoods in the plane. In *SODA '01*, pages 38–46, 2001.

[11] C. fan Hsin and M. Liu. Network coverage using low duty-cycled sensors: random & coordinated sleep algorithms. In *IPSN'04*.

[12] R. G. Gallager. *Discrete Stochastic Processes*, chapter 3, pages 63–64. Kluwer Academic Publishers, first edition, 1995.

[13] C. Gui and P. Mohapatra. Virtual patrol: a new power conservation design for surveillance using sensor networks. In *IPSN*, pages 246–253, 2005.

[14] A. Howard, M. Mataric, and G. Sukhatme. Mobile sensor network deployment using potential fields: A distributed. In *DARS*, June 2002.

[15] C.-F. Huang and Y.-C. Tseng. The coverage problem in a wireless sensor network. In *WSNA '03*, pages 115–121, 2003.

[16] J. M. Kahn, R. H. Katz, and K. S. J. Pister. Next century challenges: Mobile networking for "smart dust". In *MOBICOM*, pages 271–278, 1999.

[17] S. Kumar, T. H. Lai, and J. Balogh. On k-coverage in a mostly sleeping sensor network. In *MobiCom '04*, pages 144–158, 2004.

[18] H. T. Kung and D. Vlah. Efficient location tracking using sensor networks. In *WCNC*, 2003.

[19] J.-C. Latombe. *Robot Motion Planning*. Kluwer Academic Publishers, Norwell, MA, USA, 1991.

[20] S. M. LaValle. *Planning Algorithms*. Cambridge University Press (also available at http://msl.cs.uiuc.edu/planning/).

[21] M. Leoncini, G. Resta, and P. Santi. Analysis of a wireless sensor dropping problem in wide-area environmental monitoring. In *IPSN*, pages 239–245, 2005.

[22] B. Liu, P. Brass, O. Dousse, P. Nain, and D. Towsley. Mobility improves coverage of sensor networks. In *MobiHoc '05*, pages 300–308, 2005.

[23] A. Mainwaring, D. Culler, J. Polastre, R. Szewczyk, and J. Anderson. Wireless sensor networks for habitat monitoring. In *WSNA '02*, pages 88–97, 2002.

[24] M. B. McMickell, B. Goodwine, and L. A. Montestruque. Micabot: a robotic platform for large-scale distributed robotics. In *ICRA*, pages 1600–1605, 2003.

[25] S. Megerian, F. Koushanfar, G. Qu, G. Veltri, and M. Potkonjak. Exposure in wireless sensor networks: theory and practical solutions. *Wireless Networks*, 8(5):443–454, 2002.

[26] S. Meguerdichian, F. Koushanfar, M. Potkonjak, and M. B. Srivastava. Coverage problems in wireless ad-hoc sensor networks. In *INFOCOM*, pages 1380–1387, 2001.

[27] S. Meguerdichian, F. Koushanfar, G. Qu, and M. Potkonjak. Exposure in wireless ad-hoc sensor networks. In *MobiCom '01*, pages 139–150, 2001.

[28] G. J. Pottie and W. J. Kaiser. Wireless integrated network sensors. *Communications of the ACM*, 43(5):51–58, 2000.

[29] S. Shakkottai, R. Srikant, and N. B. Shroff. Unreliable sensor grids: Coverage, connectivity and diameter. In *INFOCOM*, 2003.

[30] G. T. Sibley, M. H. Rahimi, and G. S. Sukhatme. Robomote: A tiny mobile robot platform for large-scale ad-hoc sensor networks. In *ICRA*, pages 1143–1148, 2002.

[31] D. C. Steere, A. Baptista, D. McNamee, C. Pu, and J. Walpole. Research challenges in environmental observation and forecasting systems. In *MobiCom '00*, pages 292–299, New York, NY, USA, 2000. ACM Press.

[32] D. Tian and N. D. Georganas. A coverage-preserving node scheduling scheme for large wireless sensor networks. In *WSNA '02*, pages 32–41, 2002.

[33] I. Vasilescu, K. Kotay, D. Rus, M. Dunbabin, and P. Corke. Data collection, storage, and retrieval with an underwater sensor network. In *SenSys '05*.

[34] G. Wang, G. Cao, and T. L. Porta. Movement-assisted sensor deployment. In *INFOCOM*, 2004.

[35] X. Wang, G. Xing, Y. Zhang, C. Lu, R. Pless, and C. Gill. Integrated coverage and connectivity configuration in wireless sensor networks. In *SenSys '03*, pages 28–39, New York, NY, USA, 2003. ACM Press.

[36] G. Xing, C. Lu, R. Pless, and J. A. O'Sullivan. Co-grid: an efficient coverage maintenance protocol for distributed sensor networks. In *IPSN'04*, pages 414–423, New York, NY, USA, 2004. ACM Press.

[37] H. Yang and B. Sikdar. A protocol for tracking mobile targets using sensor network. In *WSNA*, 2003.

[38] F. Ye, G. Zhong, J. Cheng, S. Lu, and L. Zhang. PEAS: A robust energy conserving protocol for long-lived sensor networks. In *ICDCS '03*, page 28.

[39] Y. Zou and K. Chakrabarty. Sensor deployment and target localization based on virtual forces. In *INFOCOM*, 2003.