Automatica 49 (2013) 2154–2159

Contents lists available at SciVerse ScienceDirect

Automatica

journal homepage: www.elsevier.com/locate/automatica

Brief paper

Decentralized sweep coverage algorithm for multi-agent systems with workload uncertainties☆

Chao Zhai, Yiguang Hong [1]

Key Laboratory of Systems and Control, Institute of Systems Science, Chinese Academy of Sciences, Beijing 100190, China

ARTICLE INFO

Article history:
Received 16 April 2012
Received in revised form
26 February 2013
Accepted 3 March 2013
Available online 3 May 2013

Keywords:
Multi-agent systems
Sweep coverage
Decentralized control
Workload uncertainties

ABSTRACT

This paper proposes a sweep coverage formulation for a multi-agent system to cover a region with uncertain workload density, and provides a decentralized coverage algorithm based on the formulation. To achieve the coverage, the covered region is divided into a finite number of stripes, and an algorithm is proposed by incorporating two operations on stripes: workload partition and sweeping. Theoretical analysis is given to estimate the error between the actual coverage time and the optimal time, and numerical analysis is provided to illustrate the proposed algorithm.

1. Introduction

In recent years, much attention has been paid to various coordination problems in Chen and Zhang (2011), Hong, Hu, and Gao (2006), Nancy (1997) and Ren and Cao (2011). The cooperative coverage for a group of agents to effectively cover a region of interest is one of the important coordination problems for its wide application in the field of sensor networks, robotic systems, and even social systems. Practical coverage tasks such as search and rescue, exploration, monitoring, and environmental surveillance have been widely investigated with different formulations in Cassandras and Li (2005), Cheng and Andrey (2009a,b) and Hussein, Stipanović, and Wang (2007). For example, the sensor deployment algorithm was designed to monitor random events with a frequency density function and maximize the joint detection probabilities of random events in Cassandras and Li (2005), while an excellent dynamic coverage strategy was developed such that each point in the given domain is effectively covered (with collision avoidance) in Hussein and Stipanović (2007). Additionally, an awareness-based model was presented to investigate the coverage control problem over large scale domains in Wang and Hussein (2010).

The sweep coverage as one of the important coverage problems is a dynamic coverage problem in Gage (1992), and a good survey was given in Choset (2001). To solve the problem, a group of agents with the sensing capability move across the given region to detect targets of interest or complete the workload in the region. It is a difficult problem since all the agents have to sweep cooperatively in order to complete the task and even minimize the coverage time. Cheng and Andrey (2009b) proposed a formation-based sweep coverage strategy, while Choset (2000) proposed a decomposition method to cover a given area. Moreover, Butler, Rizzi, and Hollis (2000) presented a sensor-based coverage algorithm for a team of square robots to cover a finite rectangular environment. Additionally, Min and Yin (1998) reported a heuristic algorithm for multiple robots to cooperatively sweep an area with obstacles based on a market-based negotiation mechanism.

The objective of this paper is to discuss the sweep coverage with uncertain workload density. Most of the existing approaches were obtained for the uniform/known workload distribution, which is not applicable to the case of uncertain/nonuniform workload distribution. Due to the uncertainties of the nonuniform workload, it is impossible to give a fixed coverage strategy for each agent to achieve the minimum coverage time in advance. Moreover, the scheme of dynamic workload assignment is adopted to deal with the uncertainties online. The main contributions of the paper include: (i) we propose the decentralized sweep coverage algorithm with combined two operations: workload partition (to handle the uncertainties based on the agents' sensors) and sweep (to complete the coverage by agents' actuators); we estimate an upper bound of

☆ This work was supported by the NNSF of China under Grant 61174071. The material in this paper was partially presented at the 2012 American Control Conference (ACC2012), June 27–29, 2012, Montréal, Canada. This paper was recommended for publication in revised form by Associate Editor Wei Ren under the direction of Editor Frank Allgöwer.

E-mail addresses: zhaichao@amss.ac.cn (C. Zhai), yghong@iss.ss.ac.cn (Y. Hong).

[1] Tel.: +86 10 62651449; fax: +86 10 62587343.

0005-1098/$ – see front matter
http://dx.doi.org/10.1016/j.automatica.2013.03.017

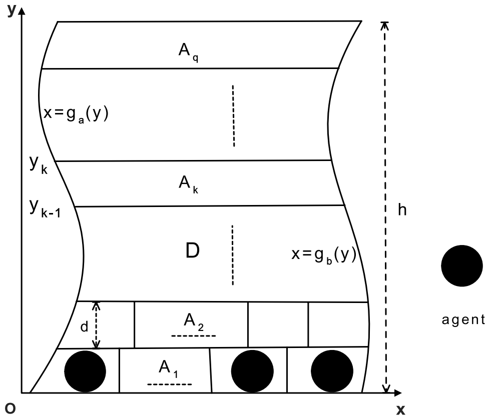

Fig. 1. Sweep coverage of mobile agents in the region D.

the extra time (as the difference between the actual coverage time and the optimal coverage time) resulting from the uncertainties.

2. Formulation and algorithm

In this section, we give a formulation for n mobile agents to sweep a given region with unknown workload distribution and then propose an algorithm.

To increase the coverage effectiveness, the whole region can simply be divided into n subregions, and each agent is responsible for the coverage job on its own subregion to complete the sweep coverage in a cooperative way. If the workload density in a covered region is known for all the agents in advance, the optimal strategy to complete the coverage may be carried out by partitioning the whole region into subregions with equal workload for all the agents so as to complete the sweeping task at the same time. Nevertheless, the limited sensing range of each agent and the unknown workload distribution make it impossible to assign average workload to each agent offline before any action. Therefore, we have to provide online coverage algorithms.

We first consider a bounded and closed set D enclosed by two parallel lines (corresponding to $y = 0$ and $y = h$, respectively) and two continuous curves described by $x = g_a(y)$ and $x = g_b(y)$ with $g_b(y) > g_a(y)$ as shown in Fig. 1. Later, we will discuss to cover a generalized set S in Section 4.

To focus on the workload partition and sweeping operation we assume all the agents line up at the bottom of region D (regarded as the start line of the agents) as their initial positions for simplicity, and they will move to the top (as the terminal line) of D by sweeping all the bounded region. Actually, for the agents with random initial positions, we can employ the decentralized algorithm to drive all agents to line up first as done in the sweep coverage control shown in Cheng and Andrey (2009b).

Suppose each agent has its actuation range with the diameter d, and therefore, when it sweeps, it will clean up a stripe with the width d. Usually $d \ll h$, and approximately the whole region D is partitioned into stripes with the same width d. For convenience, we assume $h = qd$ for some integer $q > 0$. Then the agents sweep these stripes one by one for the coverage of D.

The workload distribution is denoted by $\rho(x, y) > 0$, which is continuous and uncertain on D. We give the following assumption for $\rho(x, y)$.

Assumption 2.1. There exist positive constants $\bar{\rho}$ and $\underline{\rho}$ such that

$$\underline{\rho} \le \rho(x, y) \le \bar{\rho}, \quad \forall (x, y) \in D.$$

Table 1
Decentralized sweep coverage algorithm.

For $i = 1, \ldots, n$, i-th agent performs as follows.
set $k = 1$
while A_k is not the last stripe **do**
    while m^k is not completed **do**
        if m_i^k is not completed **then**
            update the i-th partition mark on A_{k+1} and sweep i-th sub-stripe on A_k
        else
            update the i-th partition mark on A_{k+1}
        end if
    end while
    stop partition operation and sweep the i-th sub-stripe on A_{k+1}
    set $k = k + 1$
end while
sweep i-th sub-stripe on A_q

Denote the vertical range of the k-th stripe, A_k, as $[y_{k-1}, y_k)$ with $y_0 = 0, y_q = h$ and $y_k = y_{k-1} + d$ for $k = 1, \ldots, q$. Then the workload distribution on the stripe A_k becomes

$$\omega_k(\tau) = \int_{y_{k-1}}^{y_k} \rho(\tau, y)dy, \quad 1 \le k \le q,$$

where $\rho(x, y) = 0,\ \forall (x, y) \notin D$. Set $x_0^k = \min_{(x,y)\in A_k} x$, and $x_n^k = \max_{(x,y)\in A_k} x$, which are constant on A_k. On each stripe, there are n sub-stripes for n agents. Correspondingly, there are $n - 1$ partition marks to separate the n sub-stripes. In other words, partition mark i is the common boundary of sub-stripes i and $i+1$ (swept by agents i and $i + 1$, respectively). The workload on the i-th sub-stripe of A_k is given by

$$m_i^k = \int_{x_{i-1}^k}^{x_i^k} \omega_k(\tau)d\tau \tag{1}$$

where x_i^k denotes the horizontal position of partition mark i for $i = 1, \ldots, n - 1$ on A_k. Furthermore, denote $m^k = \int_{x_0^k}^{x_n^k} \omega_k(\tau)d\tau$ as the workload on A_k.

For simplicity, we assume each agent can only communicate with its nearest neighbors and it has the same sweeping rate ν (i.e., the amount of workload completed by each agent in unit time). Obviously, the completion time of sweeping each sub-stripe depends on ν and workload on the sub-stripe.

If $\rho(x, y)$ is known, we partition the workload in advance to make each agent have the same workload so that the sweeping of D can be completed in the shortest time, so the shortest/optimal coverage time is

$$T_{\text{opt}} = \frac{1}{n\nu} \iint_D \rho(x, y)dxdy. \tag{2}$$

However, due to the uncertain workload density, we cannot make the workload partition beforehand, and we have to carry out the partition operation online.

Here we propose a simple coverage algorithm, Decentralized Sweep Coverage Algorithm (DSCA), in Table 1, which contains two basic operations: partitioning the workload on each stripe and sweeping stripe by stripe. In fact, the partitioning operation on a stripe (say A_k) for agent i can be implemented as follows:

- Sense the workload on its own sub-stripe (i.e., m_i^k) and the $i+1$-th sub-stripe (i.e., m_{i+1}^k);
- Update x_i^k, which is the partition mark between i-th and $(i+1)$-th sub-stripes, by

$$\dot{x}_i^k = \kappa(m_{i+1}^k - m_i^k), \quad i = 1, \ldots, n - 1 \tag{3}$$

for a given constant $\kappa > 0$, where the initial positions of partition marks on stripe A_k are the final positions of partition marks on A_{k-1} for $k \ge 2$.

In the algorithm, each agent is allowed to measure the workload around its position on the next stripe. Then each agent sweeps its own sub-stripe and simultaneously partitions the next stripe according to the proposed workload partition algorithm. For the simplicity of the decentralized algorithm, the agents do not help each other (otherwise, we have to demand more sensing information to avoid overlapped sweeping by different agents if one agent can work on different sub-stripes); moreover, the agents move simultaneously to the next stripe after sweeping one stripe. Therefore, if an agent completes the workload on its own sub-stripe, it stops the sweeping operation, but continues the partition operation. Once the workload on the current stripe is completed, all agents start to sweep their own just-partitioned sub-stripes on the next stripe, and meanwhile, partition the follow-up stripe. The procedure keeps repeated until D is swept.

Remark 2.1. Distributed protocols can be designed to guarantee that all the agents move from one stripe to the next (say, A_k to A_{k+1}) simultaneously. For example, each agent exchanges the workload information of sub-stripes with its adjacent neighbors to get the most workload on the sub-stripes $m_*^k = \max_{1\le i\le n} m_i^k$. There were distributed algorithms to find the optimal values of agents such as the LCR (Le Lann–Chang–Roberts) algorithm in Nancy (1997). With a modified LCR algorithm, each agent can get m_*^k in the distributed way. Omitting the detailed sweep procedure, we can roughly regard the completion time of sweeping stripe A_k as

$$T_k = \frac{m_*^k}{\nu} = \frac{1}{\nu}\max_{1\le i\le n} m_i^k \tag{4}$$

which can be taken as the common switching time for all the agents to move from A_k to A_{k+1}. In fact, T_k can be computed by each agent based on LCR since ν is known.

To make our algorithm feasible, we assume that the partition marks on each stripe will not intersect with lateral boundaries of the region D during coverage in what follows. Clearly, this assumption is true if D is rectangular, and can be guaranteed if the workload near the lateral boundaries of each stripe is quite small.

Thus, the total coverage time of the region D is $T^* = \sum_{k=1}^{q} T_k$ based on (4). The objective of our research is to estimate the coverage time T^* or equivalently to estimate the extra time between the coverage time and the optimal one, that is, $\Delta T = T^* - T_{\text{opt}}$.

3. Key lemmas

In this section, we give three lemmas for the coverage analysis in the next section. Because the analysis is mainly given on stripe A_k, we use x_i instead of x_i^k in (3) for simplicity in the sequel, when there is no confusion.

Lemma 3.1. $\lim_{t\to+\infty} |m_i^k - m_j^k| = 0,\ \forall i, j = 1, \ldots, n$, *with* m_i^k *defined in* (1).

Proof. Take a Lyapunov function on stripe A_k

$$H_k = \sum_{i=1}^{n}\left(m_i^k - \frac{m^k}{n}\right)^2 = \sum_{i=1}^{n}(m_i^k)^2 - \frac{1}{n}(m^k)^2 \tag{5}$$

to describe how uniform the workload partition on A_k is. Clearly, m^k is constant on A_k. $\dot m_i^k = \omega_k(x_i)\dot x_i - \omega_k(x_{i-1})\dot x_{i-1},\ i = 1, \ldots, n$, and particularly, $\dot m_1^k = \omega_k(x_1)\dot x_1 - \omega_k(x_0)\dot x_0 = \omega_k(x_1)\dot x_1$ and $\dot m_n^k = \omega_k(x_n)\dot x_n - \omega_k(x_{n-1})\dot x_{n-1} = -\omega_k(x_{n-1})\dot x_{n-1}$.

Then the derivative of H_k with respect to (3) is

$$\begin{aligned}\dot H_k &= 2\sum_{i=1}^{n} m_i^k \cdot \dot m_i^k - \frac{2m^k}{n}\dot m^k \\ &= 2\sum_{i=1}^{n}(\omega_k(x_i)\dot x_i - \omega_k(x_{i-1})\dot x_{i-1})m_i^k \\ &= -2\kappa\sum_{i=1}^{n-1}(m_{i+1}^k - m_i^k)^2\omega_k(x_i).\end{aligned}$$

Since $\kappa > 0$ and $\omega_k(x_i) \ge d\underline{\rho} > 0$, $\dot H_k < 0$ on the stripe A_k except for the set $E_* = \{m_1^k = m_2^k = \cdots = m_n^k\}$, which implies the convergence of E_*. □

Lemma 3.1 shows that, the partitioning algorithm given in (3) can yield the optimal partition if it runs for infinite time. However, it only runs for a finite time. How long of the partitioning algorithm runs on A_{k+1} is determined by the time spent by the agent with the most workload on its sub-stripe of A_k. Next, we have

Lemma 3.2. *With* H_k *defined in* (5) *on stripe* A_k,

$$\frac{\lambda_{\min}}{n}H_k \le \sum_{i=1}^{n-1}(m_{i+1}^k - m_i^k)^2 \le \lambda_{\max}H_k$$

where $\lambda_{\min}$ *and* $\lambda_{\max}$ *are the minimum and maximum eigenvalues of the positive definite matrix* Γ *as follows:*

$$\Gamma = \begin{pmatrix} 2 & 0 & 1 & \cdot & \cdot & 1 & 1 & 2 \\ 0 & 3 & 0 & \cdot & \cdot & 1 & 1 & 2 \\ 1 & 0 & 3 & \cdot & \cdot & 1 & 1 & 2 \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ 1 & 1 & 1 & \cdot & \cdot & 3 & 0 & 2 \\ 1 & 1 & 1 & \cdot & \cdot & 0 & 3 & 1 \\ 2 & 2 & 2 & \cdot & \cdot & 2 & 1 & 5 \end{pmatrix} \in R^{(n-1)\times(n-1)},$$

for $n > 2$ *or* $\Gamma = 4$ *for* $n = 2$.

Proof. Taking $\delta_i = m_i^k - m^k/n,\ i = 1, \ldots, n$ yields $\sum_{i=1}^{n}\delta_i = 0$ or $\delta_n = -\sum_{i=1}^{n-1}\delta_i$. Then $H_k = \|\delta\|^2 + (\sum_{i=1}^{n-1}\delta_i)^2$, where $\delta = (\delta_1, \ldots, \delta_{n-1})^T$ and $\|\cdot\|$ is the Euclidean norm. Moreover,

$$\begin{aligned}\sum_{i=1}^{n-1}(m_{i+1}^k - m_i^k)^2 &= \sum_{i=1}^{n-1}(\delta_{i+1} - \delta_i)^2 \\ &= \sum_{i=1}^{n-2}(\delta_{i+1} - \delta_i)^2 + \left(\sum_{i=1}^{n-2}\delta_i + 2\delta_{n-1}\right)^2 \\ &= \delta^T\Gamma\delta\end{aligned}$$

which implies the positive definiteness of Γ.

Since $\lambda_{\min}\|\delta\|^2 \le \delta^T\Gamma\delta \le \lambda_{\max}\|\delta\|^2$ and $\|\delta\|^2 \le H_k \le n\|\delta\|^2$, we obtain

$$\frac{\lambda_{\min}}{n}H_k \le \delta^T\Gamma\delta \le \lambda_{\max}H_k.$$

The proof is thus completed. □

Remark 3.1. $\lambda_{\min}$ and $\lambda_{\max}$ can be computed with the aid of numerical methods provided in Demmel (1997) in practice.

Denote $\hat m_i^k$ and $\tilde m_i^k$ as the workload on the i-th sub-stripe of A_k with the initial and final positions of partition marks, respectively. Take

$$H_k^a = \sum_{i=1}^{n}\left(\hat m_i^k - \frac{m^k}{n}\right)^2 \quad\text{and}\quad H_k^b = \sum_{i=1}^{n}\left(\tilde m_i^k - \frac{m^k}{n}\right)^2,$$

to describe the uniformity of the workload partition at the initial and final partition positions on A_k, respectively. Note that the initial

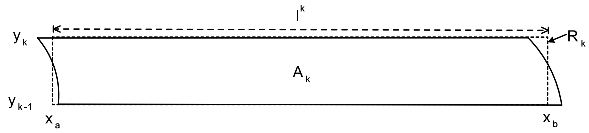

Fig. 2. The transformation from A_k to R_k.

positions of partition marks on A_{k+1} are the final partition positions on A_k. Therefore, we obtain the next lemma.

Lemma 3.3. *H_{k+1}^a and H_k^b satisfy the following inequality*

$$H_{k+1}^a \leq \alpha_d^2 H_k^b + \beta_d \frac{(dl^k)^2}{n}, \quad k = 1, \ldots, q-1 \tag{6}$$

with $l^k = \frac{1}{d}\int_{y_{k-1}}^{y_k}(g_b(y) - g_a(y))dy$, $\alpha_d = \tau_d \frac{\bar{\rho}}{\underline{\rho}}$, $\beta_d = \tau_d^2 \frac{\bar{\rho}^4}{\underline{\rho}^2} - \sigma_d^2 \underline{\rho}^2$ and $\sigma_d = \min_{1\leq k\leq q} \frac{l^{k+1}}{l^k}$, where τ_d is a constant related to d, $g_a(y)$ and $g_b(y)$.

Proof. To simplify the analysis, we construct a rectangular region R_k without changing the area and the workload on the corresponding sub-stripes on the stripe A_k (and therefore, H_k is unchanged). In this way, R_k is enclosed by four lines, corresponding to $y = y_{k-1}$, $y = y_k$, $x = x_a$, and $x = x_b$ (see Fig. 2), with

$$x_a = \frac{1}{d}\int_{y_{k-1}}^{y_k} g_a(y)dy, \qquad x_b = \frac{1}{d}\int_{y_{k-1}}^{y_k} g_b(y)dy.$$

Let $l^k = x_b - x_a$ be the length of the rectangular stripe R_k with workload m^k, and $\tilde{l}_i^k$ $(1 \leq i \leq n)$ be the length of the i-th sub-stripe on R_k, whose boundaries are the final positions of the $i-1$-th and i-th partition marks. Then $l^{k+1} \geq \sigma_d l^k$, $\forall 1 \leq k \leq q$ with $\sigma_d = \min_{1\leq k\leq q} \frac{l^{k+1}}{l^k}$. For the workload on each sub-stripe of R_k, we have

$$(\underline{\rho}\tilde{l}_i^k)^2 \leq \frac{1}{d^2}(\tilde{m}_i^k)^2 = \frac{1}{d^2}\left(\int_{x_{i-1}}^{x_i}\omega_k(\tau)d\tau\right)^2 \leq (\bar{\rho}\tilde{l}_i^k)^2.$$

Still taking the positions of partition marks of A_k on R_k, the initial partition positions on R_{k+1} are also the final boundaries between sub-stripes on R_k. Denote $\hat{l}_i^{k+1}$ as the length of the i-th sub-stripe formed by the initial $i-1$-th and i-th partition positions on R_{k+1}. Then $\hat{l}_i^{k+1} = \tilde{l}_i^k$ $(2 \leq i \leq (n-1))$ and $\hat{l}_i^{k+1} \leq \tau_d \tilde{l}_i^k$ with $\tau_d = \max_{1\leq k\leq q, i=1,\ldots,n} \frac{\hat{l}_i^{k+1}}{\tilde{l}_i^k}$. As a result,

$$(\hat{m}_i^{k+1})^2 \leq (d\bar{\rho}\hat{l}_i^{k+1})^2 \leq \tau_d^2 \frac{\bar{\rho}^2}{\underline{\rho}^2}(\tilde{m}_i^k)^2$$

which implies

$$\sum_{i=1}^n (\hat{m}_i^{k+1})^2 \leq \alpha_d^2 \sum_{i=1}^n (\tilde{m}_i^k)^2. \tag{7}$$

Similarly, we obtain

$$-\frac{(m^{k+1})^2}{n} \leq -\frac{\sigma_d^2\underline{\rho}^2}{n\bar{\rho}^2}(m^k)^2. \tag{8}$$

Combining (7) with (8) gives

$$H_{k+1}^a \leq \alpha_d^2 H_k^b + \frac{\tau_d^2\frac{\bar{\rho}^2}{\underline{\rho}^2} - \sigma_d^2\frac{\underline{\rho}^2}{\bar{\rho}^2}}{n}(m^k)^2 \leq \alpha_d^2 H_k^b + \beta_d \frac{(dl^k)^2}{n}$$

which completes the proof. □

Remark 3.2. Generally, $\tau_d \geq 1$ because $\hat{l}_i^{k+1} = \tilde{l}_i^k$, $2 \leq i \leq (n-1)$, and τ_d tends to 1 as the width of stripes d goes to 0. Moreover, we can prove $\beta_d \geq 0$. Clearly, $\tau_d = \sigma_d = 1$ if the region D is a rectangle.

4. Main results

In this section, we give main results for the coverage.

The cooperative coverage is to complete the coverage task as soon as possible. If the workload is known, we can give the optimal partition off line in advance to achieve the optimal coverage time T_{opt} defined in (2). However, due to the workload uncertainties, we have to conduct the partition online during the sweep operation, and the unequal workload partitioned in finite time yields more time to complete the coverage. The following is the main result to estimate the extra time to cover D.

Theorem 4.1. *With the DSCA given in Table 1, the extra time to sweep the region D spent in addition to the optimal coverage time is bounded by*

$$\begin{aligned}\Delta T \leq\; & \frac{\sqrt{H_1^b}}{v}\left(1 + \sum_{k=1}^{q-1}\alpha_d^k e^{-\frac{c}{n}\sum_{i=1}^k t_i}\right) \\ & + \frac{d}{v}\sqrt{\frac{\beta_d}{n}}\sum_{k=1}^{q-1} l^k \left(\sum_{i=0}^{q-1-k}\alpha_d^i e^{-\frac{c}{n}\sum_{j=k}^{k+i} t_j}\right)\end{aligned} \tag{9}$$

with $c = \lambda_{\min}\underline{\rho}\kappa d$ and $t_k = \frac{\underline{\rho}}{nv}\int_{y_{k-1}}^{y_k}(g_b(y) - g_a(y))dy$.

Proof. Since the agents sweep their own sub-stripes simultaneously, the total time spent on sweeping the whole stripe is determined by the most sub-stripe workload m_*^k. Obviously, the optimal partition of the stripe yields sub-stripes with equal workload. We first estimate the difference between the most workload and the average workload on the subregion of each stripe. Then the extra time to sweep each stripe compared to the optimal partition is calculated. Finally, we obtain an upper bound of the extra time of sweeping D.

The time derivative of H_k with respect to (3) is given by

$$\dot{H}_k = -2\kappa\sum_{i=1}^{n-1}(m_{i+1}^k - m_i^k)^2\omega_k(x_i).$$

From Lemma 3.2, we have

$$\sum_{i=1}^{n-1}(m_{i+1}^k - m_i^k)^2\omega_k(x_i) \geq \frac{\lambda_{\min}}{n}\omega_k(x_i)H_k \geq \frac{\lambda_{\min}d\underline{\rho}}{n}H_k.$$

Thus,

$$\dot{H}_k \leq -2\frac{\lambda_{\min}\kappa d\underline{\rho}}{n}H_k \tag{10}$$

which implies $H_k(t) \leq H_k^a e^{-2\frac{\lambda_{\min}\kappa d\underline{\rho}}{n}t}$. The time spent on partitioning the stripe A_{k+1} is exactly the time to sweep the stripe A_k, which has the lower bound $t_k = \frac{d\underline{\rho}l^k}{nv} = \frac{\underline{\rho}}{nv}\int_{y_{k-1}}^{y_k}(g_b(y) - g_a(y))dy$. Therefore,

$$H_{k+1}^b \leq H_{k+1}(t_k) \leq H_{k+1}^a e^{-2\frac{\lambda_{\min}\kappa d\underline{\rho}}{n}t_k}.$$

Recalling (6) gives

$$H_{k+1}^b \leq \left(\alpha_d^2 H_k^b + \beta_d\frac{(dl^k)^2}{n}\right)e^{-2\frac{\lambda_{\min}\kappa d\underline{\rho}}{n}t_k}. \tag{11}$$

As a result,

$$\sqrt{H_{k+1}^b} \leq \left(\alpha_d\sqrt{H_k^b} + dl^k\sqrt{\frac{\beta_d}{n}}\right)e^{-\frac{\lambda_{\min}\kappa d\underline{\rho}}{n}t_k}.$$

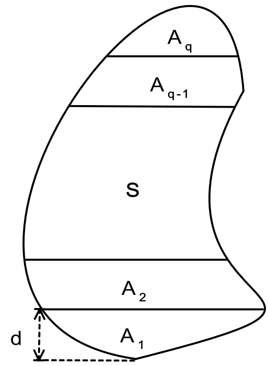

Fig. 3. A generalized bounded region S to be covered.

Hence, the extra time to sweep the whole region D is bounded by

$$\Delta T \le \frac{1}{\nu}\sum_{k=1}^{q}\sqrt{H_k^b} \le \frac{\sqrt{H_1^b}}{\nu}\left(1+\sum_{k=1}^{q-1}\alpha_d^k e^{-\frac{c}{n}\sum_{i=1}^{k}t_i}\right) + \frac{d}{\nu}\sqrt{\frac{\beta_d}{n}}\sum_{k=1}^{q-1}l^k\left(\sum_{i=0}^{q-1-k}\alpha_d^i e^{-\frac{c}{n}\sum_{j=k}^{k+i}t_j}\right)$$

which completes the proof. □

Remark 4.1. The first term of (9) is an error caused by the initial partition error H_1^b, while the second term mainly results from the non-uniform workload distribution and the irregular boundaries. If we initially partition n sub-stripes with equal workload (that is, $H_1^b = 0$), then $\Delta T \to 0$ as $n \to \infty$.

Remark 4.2. The computation, in fact, requires κ to be bounded for the estimation accuracy of (9). If we adopt the Euler method to implement the coverage algorithm with the sampling period or stepsize T_s, then κ should be chosen such that

$$\frac{\kappa d^2\bar{\rho}^2 l_{\max}}{\nu}\left(\frac{1-\gamma^q}{1-\gamma}-q\right)T_s$$

is sufficiently small, where $\gamma = e^{\frac{4\kappa d^2\bar{\rho}^2 l_{\max}}{\nu}}$ and $l_{\max} = \max_{1\le k\le q} l^k$. If we employ the Runge–Kutta method, we can get a less conservative range of κ.

Our algorithm can also be applied to a generalized region S shown in Fig. 3. Let us divide S into q stripes of width d, and S consists of three parts: A_1, $\hat{D} = \bigcup_{k=2}^{q-1} A_k$, and A_q, where $\hat{D}$ is a region with two parallel edges similar to D. Let $\Delta\hat{T}$ denote the upper bound of the extra time to sweep $\hat{D}$, which can be estimated using (9). Moreover, the extra time $\tilde{T}_l$ to sweep the stripe A_l satisfies

$$\tilde{T}_l = \frac{n\max\limits_{1\le i\le n} m_i^l - m^l}{n\nu} \le \frac{(n-1)m^l}{n\nu} \le \frac{\bar{\rho}(n-1)s_l}{n\nu}$$

where s_l denotes the area of A_l for $l = 1, q$. Thus, we have

Corollary 4.1. *With the DSCA, the extra time to sweep S is bounded by*

$$\Delta T \le \Delta\hat{T} + \frac{\bar{\rho}(n-1)(s_1+s_q)}{n\nu}. \tag{12}$$

Then we provide numerical examples to verify our coverage algorithm. Consider 5 agents in a rectangular region D with $l = 6$, $h = 8$, $d = 1$ (i.e., $q = 8$), $\kappa = 4$, $\nu = 5$, $\underline{\rho} = 1$, $\bar{\rho} = 2$ and $\rho(x, y) = 1.5 + 0.5\sin(x + y)$. The sweep process is shown in Fig. 4(a), where the sub-stripes of the same color are swept by the same agent. The first stripe has been divided initially into 5 sub-stripes with equal workload. The extra time to sweep D for the DSCA is 0.01, less than the estimation 0.03 according to (9). Then we consider a bounded region S enclosed by four curves described by $y = 5 - \sqrt{25-(x-3.2)^2}$, $y = 3 + \sqrt{25-(x-3.2)^2}$, $x = 0.2\sin\frac{\pi(y-4)}{3} + 0.2$, and $x = 0.2\sin\frac{\pi(y-4)}{3} + 6.2$, as shown in Fig. 4(b). All the parameters are the same as above. The extra time to sweep S is 0.1 in the simulation, less than the estimation 1.3 given by (12). Clearly, the estimation for S is more conservative than that for the rectangular region D.

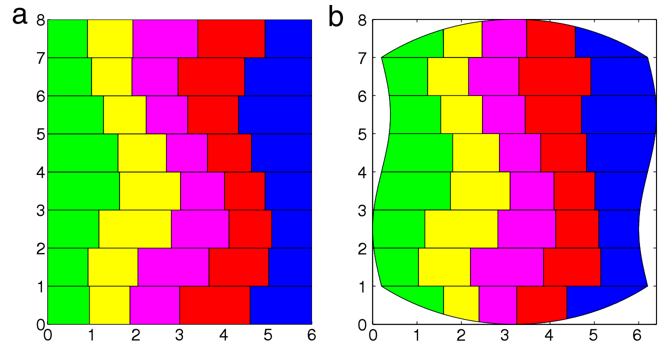

Fig. 4. (Color online) Sweeping of 5 agents on sets D and S.

5. Conclusions

In this paper, a sweep coverage algorithm was proposed to handle the dynamic sweep problem for multi-agent systems in a region with uncertain workload. The decentralized coverage algorithm was designed to achieve the coverage of the given region, and the theoretical analysis was also conducted to estimate an upper bound of the coverage time spent more than the optimal time.

References

Butler, Z., Rizzi, A., & Hollis, R. (2000). Complete distributed coverage of rectilinear environments. In *Proc. of the workshop on the algorithmic foundations of robotics* (pp. 51–62). Dartmouth, New Hampshire, USA.
Cassandras, C. G., & Li, W. (2005). Sensor networks and cooperative control. *European Journal of Control*, *11*(4–5), 436–463.
Chen, Z., & Zhang, H. (2011). No-beacon collective circular motion of jointly connected multi-agents. *Automatica*, *47*(9), 1929–1937.
Cheng, T., & Andrey, V. Savkin (2009a). A distributed self-deployment algorithm for the coverage of mobile wireless sensor networks. *IEEE Communications Letters*, *13*(11), 877–879.
Cheng, T., & Andrey, V. Savkin (2009b). Decentralized coordinated control of a vehicle network for deployment in sweep coverage. In *Proc. of the IEEE conference on control and automation* (pp. 275–279). Christchurch, New Zealand.
Choset, H. (2000). Coverage of known spaces: the boustrophedon cellular decomposition. *Autonomous Robots*, *9*(3), 247–253.
Choset, H. (2001). Coverage for robotics—a survey of recent results. *Annals of Mathematics and Artificial Intelligence*, *31*(1–4), 113–126.
Demmel, J. (1997). *Applied numerical linear algebra*. Philadelphia, PA: SIAM.
Gage, D. (1992). Command control for many-robot systems. In *Proc. of annual AUVS technical symposium* (pp. 22–24). Huntsville, Alabama, USA.
Hong, Y., Hu, J., & Gao, L. (2006). Tracking control for multi-agent consensus with an active leader and variable topology. *Automatica*, *42*(9), 1177–1182.
Hussein, I. I., & Stipanović, D. (2007). Effective coverage control for mobile sensor networks with guaranteed collision avoidance. *IEEE Transactions on Control Systems Technology*, *15*(4), 642–657.
Hussein, I.I., Stipanović, D., & Wang, Y. (2007). Reliable coverage control using heterogeneous vehicles. In *Proc. of the IEEE conference on decision and control* (pp. 6142–6147). New Orleans, LA.
Min, T., & Yin, H. (1998). A decentralized approach for cooperative sweeping by multiple mobile robots. In *Proc. of the IEEE/RSJ international conference on intelligent robots and systems* (pp. 380–385). Victoria, BC, Canada.
Nancy, A. Lynch (1997). *Distributed algorithms*. Morgan Kaufmann.
Ren, W., & Cao, Y. (2011). *Distributed coordination of multi-agent networks*. London: Springer-Verlag.
Wang, Y., & Hussein, I. I. (2010). Awareness coverage control over large-scale domains with intermittent communications. *IEEE Transactions on Automatic Control*, *55*(8), 1850–1859.

Chao Zhai received the B.Eng. degree in Automation from Henan University, China in 2007 and the M.Eng. degree in Control Theory and Control Engineering from Huazhong University of Science and Technology, China in 2009. He is currently a Ph.D. candidate of Academy of Mathematics and Systems Science, Chinese Academy of Sciences, Beijing, China. His research interests include coverage control of multi-agent systems and network and evolutionary game.

Yiguang Hong received his B.Sc. and M.Sc. degrees from Dept. of Mechanics, Peking University, China, and his Ph.D. degree from Chinese Academy of Sciences (CAS). He is currently a professor in the Academy of Mathematics and Systems Science, CAS. He is the director of Key Lab of Systems and Control, CAS, and the director of Information Technology Division of National Center for Mathematics and Interdisciplinary Sciences, CAS.

His research interests include nonlinear control, multi-agent systems, and robotics. He has published one book and about 80 journal papers. Also, he is a recipient of Guangzhaozhi Award of Chinese Control Conference (in 1997), Young Author Prize of IFAC World Congress (in 1999), Young Scientist Award of CAS (in 2001), Youth Award for Science and Technology of China (in 2006), and National Natural Science Prize of China (in 2008).