Proceedings of the 2004 IEEE
International Conference on Robotics & Automation
New Orleans, LA • April 2004

Energy-Time-Efficient Adaptive Dispatching Algorithms for Ant-Like Robot Systems

H. Jacky Chang, C. S. George Lee, Yung-Hsiang Lu, and Y. Charlie Hu
School of Electrical and Computer Engineering
Purdue University
West Lafayette, IN 47907-2035
{chang26, csglee, yunglu, ychu}@purdue.edu
https://engineering.purdue.edu/ResearchGroups/DEAR

Abstract— **In this paper, we investigate energy-time-efficient dispatching methods for ant-like robots to cover an unmapped region effectively. These ant-like robots have very limited energy and sensor ability, making them practical and inexpensive to build. Our dispatching model was based on bio-inspired algorithms from an ant colony system. We assumed that all the ant-like robots will start from their home starting point, the nest, and the region is composed of floor tiles that can be modelled as vertices in a graph. In this dispatching system, the ants-like robots leave pheromone on the tiles and use this information to cover the region. We developed and analyzed two different adaptive dispatching algorithms with different communication methods to the nest. We further compared these two adaptive dispatching algorithms with two non-adaptive methods. Extensive computer simulations validated the proposed adaptive algorithms can dispatch ant-like mobile robots in covering an unmapped region in energy-time efficiency.**

I. INTRODUCTION

Exploring an unmapped terrain by mobile robots is a fundamental and complex problem [1]. The research results have found many potential real-world applications in cleaning, mowing, search-and-rescue operation, and planetary exploration [2-6]. If the terrain is known and represented in a metric graph, the single-mobile-robot-exploring problem reduces to finding a path covering all the vertices with minimal time. This is a special case of the travelling-salesman problem (TSP) – a known NP-complete problem [7]. If the terrain is unknown, the unmapped exploration by a single mobile robot has also been treated as a graph construction [8]. To effectively speed up the unmapped covering, a popular on-line search algorithm, Learning Real-Time A* (LRTA*) [9], has been developed to cover an unknown region in polynomial time [10]. If multiple cooperative robots were used, they provided a different perspective and approach to reducing the exploration time [11].

Bio-inspired algorithms extracted from an ant colony provide excellent strategies to solve the unknown terrain covering problem. The ants have simple sensor ability, limited computational power and a decentralized control system. These features make it practical to build a large number of ant-like robots to explore an unknown terrain. Several covering algorithms were developed for ant-like robot systems [10], [13], [14]. In [15], the authors found two factors affecting the covering time significantly. The first one is the number of robots employed (k) and the other is the cut-resistance, $\rho(G)$, calculated from the environment as a graph G. The covering-time bound is $O(\frac{n}{k}+\rho(G))$; where n is the number of vertices. This time bound shows that increasing the number of cooperative ants can significantly increase the covering speed. When the number of ants is large, the environment becomes the dominant factor.

Most of the past research focused on finding the time-efficient covering methods for a fixed number of robots. In [16], the author used a cost function and optimized the robot group size in the planar-plane-search tasks. Once the tasks began, the number of robots was fixed. Hence, the adaptiveness of the number of the robots and the environment information were not fully utilized. A similar problem can be found in Mobile Agent Planning (MAP) [17], which can adaptively send out agents depending on the fluctuation of network traffic. Here MAP only deals with minimal searching time and the map (or the graph) is well-known.

In this paper, we present two environment-based adaptive dispatching algorithms to minimize the energy consumption and the covering time simultaneously. We first formulate our dispatching problem by introducing a performance index called energy-time-product, ETP. We then discuss the characteristics of ant-like robot and the environment that will affect the ETP index. The vertex-walking model and LRTA* as the covering method are adopted to test our adaptive dispatching algorithms. Computer simulation results showed that the proposed adaptive dispatching algorithms can produce energy-time-efficient covering in an unmapped region.

II. CHARACTERISTICS OF ANT-LIKE ROBOT SYSTEMS AND PROBLEM DESCRIPTION

Since our ant-like mobile robots behave slightly different from the physical ants, we need to clearly define their characteristics in our problem. To get the biggest benefit from the bio-inspired ant system, we maintain the most important

---

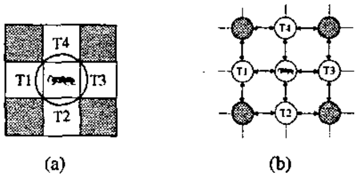
†This work was supported in part by the National Science Foundation under Grant IIS-0329061.

characteristics of the ants: the pheromone and adaptive task allocation in our systems. Next, we define and introduce a performance index called Energy-Time-Product (ETP) to help us evaluate our adaptive dispatching strategies.

A. Behavior of an Ant System

We use an ant-like mobile robot as a simplified robot model from a real ant*. In this model, ant-like robots and the environment have several important attributes:

- An ant-like robot lives in the discrete-time domain.
- The map and obstacles are composed of tiles that can be seen as vertices in an undirected graph G in Fig. 1.
- The environment is finite and has boundary.
- In each time cycle, ant-like robots do not perform actions at identical time instants. They move in a random fashion within the time period to avoid the conflict of occupying the same tile within the same time period.
- An ant can only travel to four neighboring tiles or vertices. In case of a tie, the ant will make a spiral-movement decision.
- Each movement costs one unit of time and one unit of energy.
- The communication among the ants is based on the pheromone (marks) left on the tiles.
- An ant has limited memory.



(a) (b)

Fig. 1. The ant can move to tiles or vertices, T1, T2, T3, and T4. (a) The circle is the effective sensor range of the ant. (b) The equivalent form in an undirected graph. [10]

B. Definition of Energy-Time-Product Index

Our problem is to dispatch ant-like robots to cover an unknown region with energy-time efficiency. However, there is a trade-off between energy consumption and covering time. For example, one can send out all the ant-like robots at the beginning to search the region. Obviously, this dispatching method can minimize the covering time but cannot minimize the covering energy cost. Thus, we define a new performance index, Energy-Time-Product (ETP), to measure the quality of a proposed dispatching strategy. This is the same concept of Power-Delay-Product (PDP) [18] in the hardware design that uses PDP as an index to judge a device or circuit. The ETP is defined as

$$ETP = E_T \times T_T = [\sum_{t=1}^{T_T} \sum_{i=1}^{N_{Act}(t)} P_i \times 1] \times T_T \qquad (1)$$

where E_T and T_T are the total energy and time consumed in the covering process, respectively, $N_{Act}(t)$ is the activated number of ants at time t, and P_i is the power of ant_i. When an ant is activated, P_i is set to one.

From (1), we want to determine the parameter, $N_{Act}(t)$, to minimize ETP. We shall present two environment-based adaptive dispatching algorithms to send out ants dynamically depending on the environment and the covering situation. In these strategies, $N_{Act}(t)$ and T_T are taken into consideration at the same time, and thus ETP can be minimized in different unknown environments.

III. ENVIRONMENT-BASED ADAPTIVE DISPATCHING ALGORITHMS

Before analyzing and developing dispatching algorithms, we shall first study the impact of the environment. This information will lead us to develop appropriate algorithms. We shall then develop and analyze two dispatching algorithms with different communication restrictions. The first one assumes that the ants can remotely communicate with the nest and ask for assistance without going back to the nest (i.e., to control $N_{Act}(t)$). The second one is more realistic. Like physical ants, they can only communicate face to face with other ants. Thus, in this dispatching algorithm, the ant needs to go back to the nest and request assistance from the other ants.

A. Impact of the Environment

According to the research in [15], the covering time highly depends on the number of ants and the environment. We consider two maps to demonstrate how the environment influences the covering time and energy consumption. In Fig. 2, (a) and (b) are two different maps needed to be covered by ants. In Fig. 2(a), ants start from the center of the planar plane and disperse quickly in the direction shown by the arrows. Every ant has contributed to reducing the covering time. By increasing the number of ants from N to $N+1$, the covering time can be improved about $\frac{T_T}{N+1}$ until the number of ants starts to saturate the map. This phenomenon has been confirmed in a 30×30 planar plane simulation result shown in Fig. 3. Figure 2(b) shows a single-file path, where all the ants can move only in one direction and one tile at a time. This means that no matter how many ants are sent out, the covering time is a constant, and the other ants are only wasting the energy. Hence, a good energy-time-efficient covering method must take the environment into consideration.

B. Environment-Based Adaptive Dispatching Algorithm I (ENAD-I)

In the covering problem, one needs to determine $N_{Act}(t)$ by discovered environment information. Because the environment is unmapped, a basic approach is that if an ant has observed a large enough uncovered vertices then it will request the nest to send out an additional ant. Thus, the dispatching algorithm is adaptive to the explored environment. Since this additional ant consumes energy and time travelling to the needed location, part of the found uncovered vertices might be covered by other

---

*If there is no confusion, we shall use "ants" and "ant-like robots" synonymously in the remaining of this paper.

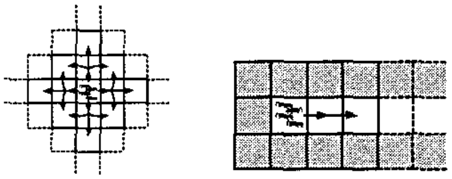

(a) (b)

Fig. 2. (a) The arrows show the ants disperse quickly in a planar plane. (b) The ants can only move to one tile and one direction on a path. The shaded squares represent obstacles or the wall.

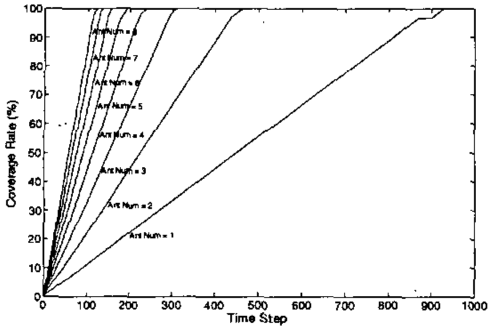


Fig. 3. The coverage rate with different number of ants in a 30 × 30 planar plane.

ants. According to the ants' behavior, most of these vertices are covered by the ant that requests for assistance. Therefore, we can use the accumulated number of found vertices and the travelling time of this additional ant to estimate the *ETP* cost and determine whether to request for assistance or not.

The travelling time of the additional ant can be seen as a single-pair shortest path problem. Using the modified Dijkstra's algorithm, the shortest distance to the nest can be found. The modification means when the ant covers a new vertex or goes through an old vertex, it updates the shortest path mark according to its neighboring vertex. Thus, the discovered shortest distance might not be the real shortest distance. In (2), T_{ν_i} is the time to cover the found uncovered vertices with the help of the additional ant at vertex ν denoted as ant number i or Ant_i (i.e., it takes Δ_i time steps to cover the region by one ant). T_{ν_i} equals to the travelling time of the additional ant (the known shortest distance to the nest) plus the time which is required to explore uncovered vertices by two ants, where Δ_i is the accumulated uncovered vertices by Ant_i, and π_ν is the found shortest distance from vertex ν to the nest.

$$T_{\nu_i} = \pi_\nu + \frac{\Delta_i - \pi_\nu}{2} = \frac{\Delta_i + \pi_\nu}{2} \tag{2}$$

$$\Delta_i^2 \times P_i \geq T_{\nu_i}^2 \times P_i \tag{3}$$

In (3), the left inequality term is the *ETP* cost of not calling for help and the right inequality term is the *ETP* cost of calling for help. When the *ETP* cost of calling for help is smaller than not calling for help, the ant sends out a message to the nest, and an additional ant is requested to start exploration. By solving (3), we obtain the condition $\Delta_i \geq \pi_\nu$. In our proposed environment-based adaptive dispatching algorithm I (ENAD-I), the ant transmits the request remotely to the nest. With this communication capability, the environment information can be reported to the nest quickly. The ENAD-I algorithm is described below:

Environment-Based Adaptive Dispatching Algorithm I (ENAD-I): Given a number of ants N, an unknown region as a graph G, and one vertex as the nest ν_{nest}. This ENAD-I algorithm determines an adaptive dispatching of N ants to cover the region G while minimizing the energy and coverage time according to the *ETP* index. The algorithm also determines when the ant should call for help and an additional ant will be sent out from the nest. The algorithm terminates when the coverage is done or the running time equals to the maximum execution time T_{max}.

E1. [*Ant system initialization.*] For $i = 1$ to N, initialize the current position of Ant_i, $CP_i \leftarrow \nu_{nest}$. Set Ant_i to the sleeping mode, $Mode_{Ant_i} \leftarrow$ *Sleeping Mode*. Set the count of the accumulated number of found uncovered vertices to 0, $\Delta_{i_{acc}} \leftarrow 0$. Activate one ant to the covering mode $Mode_{Ant_1} \leftarrow$ *Covering Mode*, and the number of activated ants is set to 1, *Act_Num* $\leftarrow 1$.

E2. [*Environment initialization.*] V is the set of vertices of G. For all $\nu \in V$, the pheromone on the ground is set to empty, $Ph_\nu \leftarrow 0$. Set the shortest distance from each vertex to the nest to ∞, excluding ν_{nest}, for all $\nu \in V, \pi_\nu \leftarrow \infty, \pi_{\nu_{nest}} \leftarrow 0$.

E3. [*Main loop.*] For $i = 1$ to *Act_Num*, execute Steps **E4-E7**.

E4. [*LRTA* covering algorithm.*] For Ant_i, run the algorithm LRTA* for one time step [10]. It decides the next movement of Ant_i and updates the pheromone on CP_i. $(Ph_{CP_i}, CP_i) \leftarrow LRTA^*(Ph_{CP_i}, CP_i)$

E5. [*Modified Dijkstra's algorithm.*] Update the shortest distance to the nest,

$$\pi_{CP_i} \leftarrow \min_{\omega \in \text{neighbor of } CP_i} (\pi_\omega) + 1$$

E6. [*Estimation of uncovered vertices.*] Accumulate the number of found uncovered vertices, where W_{CP_i} is the number of uncovered neighboring vertices of CP_i, $\Delta_{i_{acc}} \leftarrow \Delta_{i_{acc}} - 1 + W_{CP_i}$.

Estimate the actual number of the found uncovered vertices, $\Delta_i \leftarrow 4\sqrt{\Delta_{i_{acc}} + 1}$.

E7. [*Determination of calling for help by ETP cost.*] Check the calling for help condition.

IF *Act_Num* $< N$ and $\Delta_i \geq \pi_{CP_i}$, **THEN** {transmit the request for help to the nest, *Act_Num* $\leftarrow$ *Act_Num* $+1$ and $Mode_{Ant_{Act\_Num+1}} \leftarrow$ *Covering Mode* and reset the accumulated number of found uncovered vertices, $\Delta_{i_{acc}} \leftarrow 0$.}

E8. [*Termination of Main loop.*]

IF $T \geq T_{max}$ or the coverage is done, **THEN** stop.

ELSE $T \leftarrow T + 1$ and go to Step **E3**.

END ENAD-I Algorithm.

C. Environment-Based Adaptive Dispatching Algorithm II (ENAD-II)

In nature, the ants communicate to each other by antenna contact (i.e., face to face communication). We assume the ant-like robots will have the same behavior. Hence, if an exploring ant needs further assistance for exploration, it must travel back to the nest and request for additional help from other ants. Figure 4 illustrates how this process is being carried out.

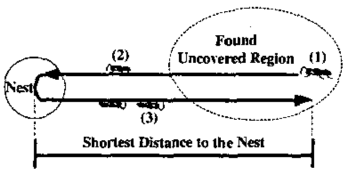


Fig. 4. Three different modes of operation for an ant in covering an unknown region. (1) First, the ant is in the covering mode. (2) When the ant discovers that the uncovered region is large enough, it switches to the going-home mode and leaves the pheromone on the ground. (3) After the ant reaches the nest, it and an additional ant change to the tracking mode to follow the trace to the found uncovered location.

Although the exploring ant in Algorithm ENAD-I can request help remotely from the nest, the additional ant cannot recognize the shortest path to the uncovered region immediately. In Algorithm ENAD-II, the ant that is going home can leave traces on the ground, and the activated ant can follow the traces to the uncovered vertices. We develop Algorithm ENAD-II by extending the same concept from Algorithm ENAD-I. In (4), T_{ν_i} equals to two ants' travelling time plus the time of covering the found uncovered region. In (5), the energy cost is the summation of one ant going home, two ants coming back and two ants covering the uncovered vertices. Similar to (3), we re-formulate to estimate the condition for requesting help from the nest from (6), and the condition is $\Delta_i \geq 4.95\pi_\nu$.

$$T_{\nu_i} = 2\pi_\nu + \frac{\Delta_i}{2} \tag{4}$$

$$E_{\nu_i} = P_i \times (\pi_\nu + 2\pi_\nu + \frac{\Delta_i}{2}) \tag{5}$$

$$\Delta_i^2 \times P_i \geq ETP_{\nu_i} = E_{\nu_i} \times T_{\nu_i} \tag{6}$$

The Environment-Based Adaptive Dispatching Algorithm II (ENAD-II) is described below.

Environment-Based Adaptive Dispatching Algorithm II (ENAD-II): Given a number of ants N, an unknown region as a graph G, and one vertex as the nest ν_{nest}. This ENAD-II algorithm determines an adaptive dispatching of N ants to cover the region G while minimizing the energy and coverage time according to the ETP index. The algorithm also determines when the ant should go back to the nest and request assistance for an additional ant to help the exploration. The algorithm terminates when the coverage is done or the running time equals to the maximum execution time T_{max}.

N1. [*Ant system initialization.*] Initialize the ant system as in Steps **E1-E2** in Algorithm ENAD-I. For $i = 1$ to N, {the Ant_i knows that the number of ants in the nest is not empty, $Ant_{i\text{emp}} \leftarrow 0$. For all $\nu \in V$, the tracking pheromone on the ground is set to to empty, $Pt_{\nu,i} \leftarrow 0$.}

N2. [*Main loop.*] For $i = 1$ to Act_Num, execute Step **N3**.

N3. [*Checking Ant's Mode.*] **IF** $Mode_{Ant_i}$ is in *Covering Mode*, **THEN** execute Steps **N4-N5**;
IF $Mode_{Ant_i}$ is in *Going-Home Mode*, **THEN** execute Step **N6**;
IF $Mode_{Ant_i}$ is in *Tracking Mode*, **THEN** execute Step **N7**.

N4. [*Covering Mode.*] Execute Steps **E4-E6** as in Algorithm ENAD-I.

N5. [*Determination of calling for help by ETP cost.*] Check the calling for help condition.
IF $Ant_{i\text{emp}}$ equals 0 and $\Delta_i \geq 4.95\pi_{CP_i}$, **THEN** {$Mode_{Ant_i} \leftarrow$ *Going-Home Mode*, and reset the accumulated number of found uncovered vertices, $\Delta_{i_{acc}} \leftarrow 0$.}

N6. [*Going-Home Mode.*] **IF** the Ant_i reaches the nest (CP_i equals to ν_{nest}), **THEN** {**IF** $Act\_Num < N$, **THEN** {increase the activated number of ants, $Act\_Num \leftarrow Act\_Num + 1$, and set these two ants' mode to the tracking mode, $Mode_{Ant_{Act\_Num+1}} \leftarrow$ *Tracking Mode*, $Mode_{Ant_i} \leftarrow$ *Tracking Mode*. Set two ants to track the pheromone of Ant_i. $Ant\_Track_i \leftarrow i$, $Ant\_Track_{Act\_Num+1} \leftarrow i$. Two ants leave the nest to the uncovered vertices.} **ELSE** {The nest is empty, $Ant_{i\text{emp}} \leftarrow 1$. Set the Ant_i to the tracking mode and track the pheromone of Ant_i, $Mode_{Ant_i} \leftarrow$ *Tracking Mode*, $Ant\_Track_i \leftarrow i$.}}
ELSE {find the next closest vertex to the nest, $\omega : \min_{\omega \in \text{neighbor of } CP_i} (\pi_\omega)$, Ant_i moves to vertex ω, $CP_i \leftarrow \omega$. Leave the tracking pheromone on CP_i, $Pt_{CP_i,i} \leftarrow 1$.}

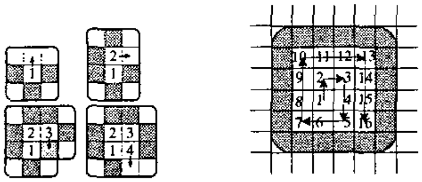
N7. [*Tracking Mode.*]
IF Ant_i arrives at the location of the found uncovered vertices where there is no neighboring vertex ω such that Pt_{ω,Ant_Track_i} equals 1,
THEN {set the ant's mode to the covering mode $Mode_{Ant_i} \leftarrow$ *Covering Mode*.}
ELSE {find the next closest vertex to the uncovered region, $\omega : \max_{\omega \in \text{neighbor of} CP_i} (Pt_{\omega,Ant\_Track_i})$. Evaporate the tracking pheromone, $Pt_{\omega,Ant\_Track_i} \leftarrow 0$. Move to vertex ω, $CP_i \leftarrow \omega$.}

N8. [*Termination of Main loop.*]
IF $T \geq T_{max}$ or the coverage is done, **THEN** stop.
ELSE $T \leftarrow T + 1$ and go to Step **N2**.

END ENAD-II Algorithm.

D. Estimation of Uncovered Vertices

In this subsection, we discuss our reason in estimating Δ_i from Δ_{acc_i} in Algorithms ENAD-I (Step **E6**) and ENAD-II (Step **N3**). Because each ant has limited memory, it accumulates the number of uncovered vertices, Δ_{acc} and



(a) (b)

Fig. 5. The estimation of found uncovered region. The arrow indicates the ant's walking behavior in a planar plane. (a) The shaded tiles are the observed uncovered region. At $T = 3$, one tile with bold boundary is double counted by the ant. The estimated uncovered region is the rounded rectangles minus the covered region with tiles marked with numbers. (b) The shaded tiles are uncovered tiles at $T = 16$.

cannot distinguish which vertices have been encountered. The accumulated number of found uncovered vertices is not the actual number of uncovered vertices. Therefore, we need to estimate Δ_i from Δ_{acc_i} to establish the condition for calling for help. The estimation is based on the behavior of an ant. It explores the nearby vertices first. For example, as shown in Fig. 5, the Ant_i starts from tile 1 and goes to tile 2 and so on. The $\Delta_{acc_i} = \{3, 5, 7, 8...\}$ correspond to $T = \{1, 2, 3, 4...\}$ and the number of actual uncovered tiles can be estimated by $\Delta_i = 4\sqrt{\Delta_{acc_i} + 1}$, where T is the elapsed time step. At $T = 16$, $\Delta_{acc_i} = 24$, and $\Delta_i = 20$, the actual uncovered tiles are the shaded tiles in Fig. 5 (b).

IV. SIMULATION RESULTS.

We compared two non-adaptive dispatching methods with ENAD-I,II for different number of ants in our computer simulation. The first non-adaptive dispatching method is called "Fixed Dispatching" in which the system dispatches all the ants at the starting time. The second one is "Linear Dispatching" in which the system dispatches the ants at a fixed time interval. (e.g., we have 10 ants in the nest at the period, t_p=3. We dispatch the ants one by one according to the time sequence, $T = 1, 4, 7, 10...$). We choose these two non-adaptive methods for several reasons. First, the fixed dispatching is optimal for a known planar plane (see Section III-A). Second, the linear dispatching has a strong sense in "If the covering mission is not completed yet, keep dispatching ants to cover the unknown region." If the dispatching period and the number of ants are correctly selected, they can achieve minimal ETP. The major problem of these two non-adaptive methods is that these two parameters cannot be pre-determined before the ants explore the region. Hence, in many cases, these two methods fail to achieve minimal ETP.

Four different maps (a planar plane, a single-file path, a planar plane with two U-shaped obstacles, and an office layout) are generated to test these four dispatching methods. The period of linear dispatching is set to 2 time units. In Fig. 6, as expected, the best dispatching method is the fixed-dispatching method for a planar-plane map. In Fig. 7, the single-file-path map is the worst case for the two non-adaptive methods. Algorithms ENAD-I,II only dispatch one ant and perform quite well. In Fig. 8, as the number of ants increases at the nest, ENAD-II is the best one among them. This case also provides an interesting result; that is, "If we only have a small number of ants and we know the map is large, then the best strategy is to dispatch all of them at the starting time." We also simulated our adaptive dispatching algorithms in a real 40 × 40 office map and the results are shown in Fig. 9. In the office map simulation, when there are more than 16 ants in the nest, ENAD-II only uses 16 ants to cover the region.

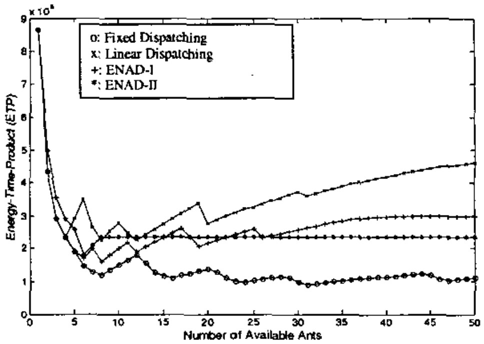


Fig. 6. The ETP cost of four different dispatching methods in a 30 × 30 planar-plane map (The lower the better).

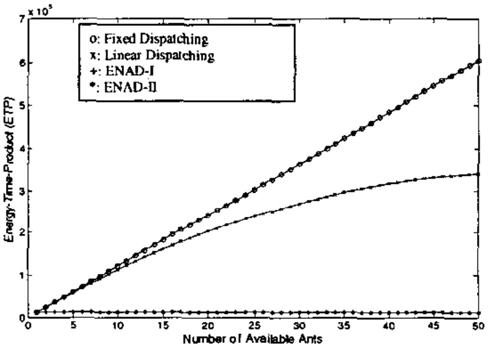


Fig. 7. The ETP cost of four different dispatching methods in a single-file-path map.

Tables I and II show the comparison of the average of activated number of ants and the average ETP cost in fifty simulations. The first simulation has only one ant in the nest at the beginning and the fiftieth simulation has fifty. From these tables, the averages of ETP of ENAD-I,II show that they produce minimal ETP in a single-file-path map and a reasonable ETP cost in a planar-plane or office map. In the office map, the average number of activated ants in ENAD-II is 14.3, which is much smaller than 25.5 in the non-adaptive methods. Hence, the ants are dispatched in an efficient way. These results show that the performance of ENAD-II is adaptive to the environment and produces a good ETP performance in general cases. Furthermore, if the number of ants is very large in the nest, ENAD-II is a practical dispatching algorithm to cover the unknown region.

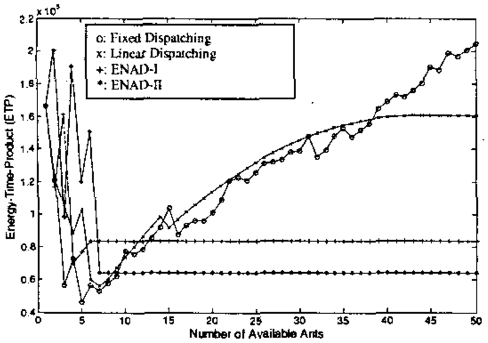


Fig. 8. The *ETP* cost of four different dispatching methods in a 10 × 40 planar plane with two U-shaped obstacles.

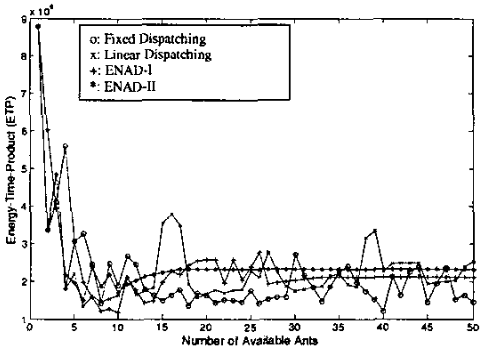


Fig. 9. The *ETP* cost of four different dispatching methods in an office layout.

TABLE I

COMPARISON OF DIFFERENT DISPATCHING METHODS IN PLANAR-PLANE AND SINGLE-FILE-PATH MAPS

Methods	Planar Plane		Single-File Path	
	Avg. ETP	Avg. Ants	Avg. ETP	Avg. Ants
Fixed	1.46E+05	25.5	3.09E+05	25.5
Linear	3.58E+05	25.5	2.20E+05	25.5
ENAD-I	2.67E+05	24.6	1.21E+04	1.00
ENAD-II	2.54E+05	7.40	1.21E+04	1.00

TABLE II

COMPARISON OF DIFFERENT DISPATCHING METHODS IN TWO U-SHAPED OBSTACLES AND AN OFFICE MAP

Methods	2 U-Shape Obstacles		Office Map	
	Avg. ETP	Avg. Ants	Avg. ETP	Avg. Ants
Fixed	1.27E+05	25.5	2.16E+06	25.5
Linear	1.27E+05	24.9	2.34E+06	25.5
ENAD-I	8.71E+04	5.70	2.29E+06	23.1
ENAD-II	7.51E+04	6.60	2.49E+06	14.3

V. CONCLUSIONS

This paper described two adaptive dispatching algorithms, ENAD-I,II, for an ant-like robot system to cover an unknown terrain. We introduced a performance index called energy-time-product, ETP, to evaluate the performance of the dispatching algorithms both in energy and time. The strengths of our algorithms are that they are adaptive to the environment and they utilize fewer ants to achieve energy-time-efficient coverage. Computer simulations have validated the performance of proposed adaptive dispatching algorithms in different unknown maps. Experimental verification on a team of mobile robots will be scheduled and performed in the near future.

REFERENCES

[1] H. Durrant-Whyte. Uncertain geometry in robotics. *IEEE Journal of Robotics and Automation*, 4(1):23–31, Feb. 1988.

[2] C. Hofner and G. Schmidt. Path planning and guidance techniques for an autonomous mobile cleaning robot. In *International Conference on Proceedings of the IEEE/RSJ/GI*, volume 1, pages 610–617. Intelligent Robots and Systems '94. 'Advanced Robotic Systems and the Real World', Sept. 1994.

[3] C. Luo and S. Yang. A real-time cooperative sweeping strategy for multiple cleaning robots. In *Proceedings of the 2002 IEEE Internatinal Symposium on Intelligent Control*, pages 660–665, Oct. 2002.

[4] K. Suzuki, S. Makami, J. Akita, and E. Osawa. Development of cooperative small mowing robots. In *IEEE International Symposium on Computational Intelligence in Robotics and Automation*, volume 3, pages 1498–1502, July 2003.

[5] M. Herbert, C. Caillas, E. Krotkov, I. Kweon, and T. Kanade. Terrain mapping for a roving planetary explorer. In *IEEE International Conference on Robotics and Automation*, volume 2, pages 997–1002, May 1989.

[6] J. Jennings, G. Whelan, and W. Evans. Cooperative search and rescue with a team of mobile robots. In *8th International Conference on Advanced Robotics ICAR '97*, pages 193–200, July 1997.

[7] E. M. Arkin and R. Hassin. Approximation algorithms for the geometric covering salesman problem. *DAMATH: Discrete Applied Mathematics and Combinatorial Operations Research and Computer Science*, 55, 1994.

[8] G. Dudek, M. Jenkin, E. Milios, and D. Wilkes. Robotic exploration as graph construction. *IEEE Transactions on Robotics and Automation*, 7:859–865, Dec. 1991.

[9] R. E. Korf. Real-time heuristic search. *Artificial Intelligence*, 42(2-3):189–211, March 1990.

[10] S. Koenig, B. Szymanski, and Y. Liu. Efficient and inefficient ant coverage methods. *Annals of Mathematics and Artificial Intelligence*, 31:41–76, 2001.

[11] D. Kurabayashi, J. Ota, T. Arai, and E. Yoshida. Cooperative sweeping by multiple mobile robots. In *IEEE International Conference on Robotics and Automation*, volume 2, pages 1744–1749, April 1996.

[12] H. Choset. Coverage for robotics - a survey of recent results. *Annals of Mathematics and Artificial Intelligence*, 31:113–126, 2001.

[13] Y. Gabriely and E. Rimon. Spanning-tree based coverage of continuous areas by a mobile robot. *Annals of Mathematics and Artificial Intelligence*, 31:77–98, 2001.

[14] I. A. Wagner, M. Lindenbaum, and A. M. Bruckstein. Efficiently searching a graph by a smell-oriented vertex process. *Annals of Mathematics and Artificial Intelligence*, 24(1-4):211–223, 1998.

[15] I. A. Wagner, M. Lindenbaum, and A. M. Bruckstein. Distributed covering by ant-robots using evaporating traces. *IEEE Transactions on Robotics and Automation*, 15(5):918–933, Oct. 1999.

[16] A. T. Hayes. Group size and efficiency in collective search tasks. In *In Proc. of the 6th Int. Symp. on Distributed Autonomous Robotic Systems DARS-02*, pages 289–298, June 2002.

[17] J.-W. Baek, J.-H. Yeo, and H.-Y. Yeom. Agent chaining: an approach to dynamic mobile agent planning. pages 579–586. Proceedings. 22nd International Conference on Distributed Computing Systems, 2-5 July 2002.

[18] J. M. Rabaey. *Digital Integrated Circuits: A Design Perspective*. Prentice Hall, 2 edition, 2003.