ADAPTIVE STRATEGY SELECTION FOR MULTI-ROBOT SEARCH BASED ON LOCAL COMMUNICATION AND SENSING

Damien Bright

KnowledgeLab, University of Southern Denmark
Campusvej 55, DK-5230 Odense M, Denmark

Keywords: simulation, collective robotics, stigmergy, reinforced random walk, optimization.

Abstract: This paper presents a simulation model for experimenting with locally adaptive movement strategies for robots involved in collective robotic search tasks in rapidly changing and uncertain environments. The model assumes that the nature of the environment restricts inter-robot communication and uses a form of stigmergy based local communication which has been widely applied in collective robots. The model is based on a biased random walk where the degree of bias is linked to a local control variable which can change depending on the evaluation of local adaption strategies. The local adaption strategies use an approach based on activation functions to control the choice of which candidate paths should be inhibited or have increased preference over random motion. Experiments aim to test the effectiveness of this approach for optimal collective search in various test domains. A series of initial experiments is presented demontrating aspects of the model.

1 INTRODUCTION

Studies of foraging behaviour in insects and animals have been used by many researchers in collective robotics as a model for experimenting with search behaviour for robots (eg (Balch and Arkin, 1994)). The modeling of foraging can be divided at a general level into macroscopic or microscopic approaches where the former is used to model the actions of large numbers of entities and the latter the actions of individuals. It has been found that there are advantages in the use of microscopic models with a high degree of localization and the use of external communication mechanisms in improving the ability of collaborating robots ((Holland and Melhuish, 1999), (Wagner et al., 1999),(Montgomery and Randall, 2002)) to perform foraging or search type tasks. This is often because robots generally have independent control software and it is advantageous to remove the need to maintain direct communication and shared knowledge of individuals exact locations between multiple robots which can be difficult to support in uncertain and highly dynamic (i.e. rapidly changing) environments.

External communication mechanisms have attracted considerable interest from collective robotics and AI life researchers. In particular, the concept of stigmergy (Holland and Melhuish, 1999) which describes a form of indirect communication utilizing the environment as the communication medium has been widely studied. Stigmergeric cues or markers represent a change made to the environment that communicates information and can take many forms E.g. this could involve the use of beacons to guide robots for tasks such as robot search and rescue. A much studied form of stigmergeric marker is the use of pheromone for trail marking where a chemical marker (pheromone) is deposited. Entities in the system have a pre-disposition to follow the strongest pheromone trails they encounter and pheromone trails evaporate (decay) over time which is useful for optimization. Usually pheromone is considered as an attractant but (Montgomery and Randall, 2002) has introduced the concept of anti-pheromone as a useful tool for exploring a search space. This is related to the fact that a useful pure reactive strategy for search is to avoid previously covered terrain which can be marked with negative bias in the form of anti-pheromone.

Recent research in areas such as adaptive control and optimization methods has examined the idea of locally adaptive strategy within some search space. For example this approach has been used in Genetic algorithm (GA) work e.g. see (Igel et al., 2005) and ”self-tuning” methods for robotic control (Patterson

Bright D. (2005).
ADAPTIVE STRATEGY SELECTION FOR MULTI-ROBOT SEARCH BASED ON LOCAL COMMUNICATION AND SENSING.
In *Proceedings of the Second International Conference on Informatics in Control, Automation and Robotics - Robotics and Automation*, pages 335-340
DOI: 10.5220/0001186503350340

and Kautz, 2001). A large body of past research in general optimization techniques such as gradient descent and simulated annealing (Glover and Kochenberger, 2003) has also shown that the inclusion of random processes can be important to achieving global maxima/minima over local maxima/minima. Robotic control which is purely reactive (like gradient following) can lead to the same type of local maxmima/minima issues. The addition of randomness or noise into robotic motion has been shown to help to avoid this (Balch and Arkin, 1994) but the strategy used to determine the amount of random motion is generally fixed (ie not locally adaptive) and therefore not well suited to time-dependent problems requiring adaption to changing or uncertain environmental conditions.

The main contribution of this paper is the development of a simulation model based on a novel approach for representing local self-tuning or adaption strategies within the context of optimization of multi-robot search in complex and dynamic domains. A key aim of the simulation model is to be able to study in detail the use and effect of local strategies which vary the degree to which reactive behaviour to marker trails and reinforcement of marker trails should dominate over random motions. The random walk model used in this paper can be compared to a markov decision process and has influences from reinforcement learning theory (Kaelbling et al., 1994). This paper is structured as follows: First a brief summary is given of related work, then the numerical model is defined and a set of initial results is presented. A summary and future work section then describes some of the aims and proposed uses of the simulation model.

2 RELATED WORK

A large body of research in AI life applied to optimization and guidance problems has made use of indirect communication techniques based on social insects such as trail laying. This has led to the term synthetic pheromone describing data structures inspired by chemical markers called pheromones from biological systems. Research in collective robotics (Holland and Melhuish, 1999) has made substantial use of stigmergy and other concepts from AI life research. Approaches based on such techniques can a provide robust and adaptive indirect coordination mechanism for collaborating entities such as robots (Wagner et al., 1999). Multiple robots using such techniques are particularly efficient for tasks such as mapping unknown terrain which are well suited to being performed collectively. Each robot needs only relatively simple functionality to achieve complex group behaviour. This reduces the complexity and cost of each robot. Different approaches to navigation strategy for indoor searching have been examined by (Gonzales-Banos and Latombe, 2002). One approach to aid search behaviour has been the use of coverage maps (Stachniss and Burgard, 2003) which in a similar way to maker trails can be viewed as a form of indirect communication stored in the environment. Lately there has been increased use of multiple robots to perform specialized tasks (eg search and rescue (Baltes and Anderson, 2003)). Some common problems with multiple robot guidance are:

(1) Pure reactive navigation often suffers from local minima issues due to limitations such as sensor range and/or accuracy. It has been found that a combination of goal directed behaviour and reactive behaviour can be an effective (Balch and Arkin, 1994) strategy but this often requires more complex robot behaviour such as the use of path planning algorithms.

(2) centralized versus de-centralized control and whether to use localized and indirect communication. Decentralized control and indirect communication (see (Holland and Melhuish, 1999), (Wagner et al., 1999)) can be very useful in complex and dynamic environments where the environment is spatially/geographically complex or where the positions of objects that exist in the environment change or the environment itself is subject to uncertainty or change. Also robots with limited ability to transmit and communicate over longer distances can benefit from an approach based on local communication.

Random walk models have been widely used to model movement patterns such as dispersion. It is possible under certain conditions to look at local control decisions in a biased random walk as a form of Markov decision process. (Azar et al., 1992) examine optimal strategy applicable to time independent long term behaviour of a random walk on a finite graph where local movement decisions can be viewed in terms of a controller selecting from a set of available actions to bias the behaviour of a markov chain. This type of approach has relevance for this paper but is not able to address time dependent local strategy formation. A reinforced random walk model was first proposed by Coppersmith and Diasconis (Coppersmith and Diaconis, 1987) as way of modeling a person exploring a new city (See also (Davis, 1990)). Random walk models can be used as an important part of more specific models for spatial exploration and cooperative interaction. For example a biased random walk model which uses feedback with the environment to influence a walkers movement is the active walker model originally formulated by (Lam and Pochy, 1993).

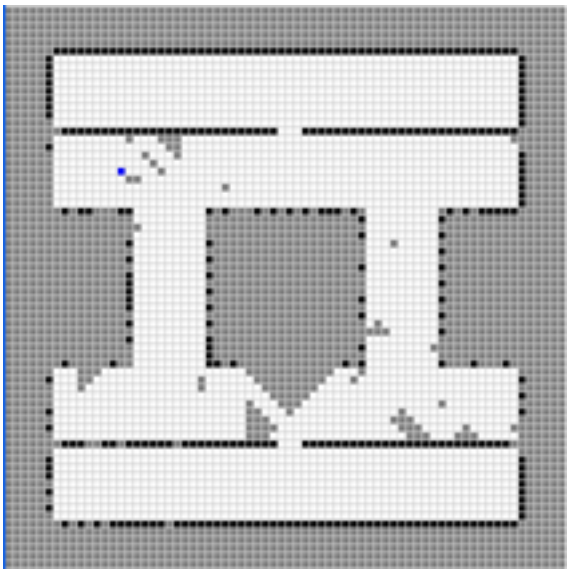

Figure 1: Single robot search on box canyon maze domain with $n_x = 70, n_y = 70$, $r_{seed} = 2392093$, $R_d = 0.0$, and $\nu = 0.95$, $nsteps = 5000$, Starting position is $(55, 35)$. No strategy change is applied for this run.

3 NUMERICAL MODEL

The discrete time simulation model that is proposed here is based on the mapping of values from a global time dependent field $m(x, y, t)$, which represents the dynamic (changing) part of the environment. m is used to store data for stigmergy style collective communication and gives the value of a marker (ie a type of synthetic pheromone) m at position (x, y) in a 2D domain at time t. m is mapped to local vectors of the form $v_{ij}, i = 0..N$ which represent the probability of making one of N possible local choices (eg a local control decision of picking an adjacent square to move to) for robot j in the system on the next timestep. Marker values m can be positive or negative representing an attractant marker or repulsant marker respectively. The field $m(x, y, t)$ influences the movement of robots in the system but also changes due to feedback affects from individual robots when an robot changes its local environment by laying a marker trail. Therefore the model represents a stigmergy based form of indirect communication between robots using the environment as the communication medium.

Locally visible values of $m(x, y, t)$ can be viewed as positive or negative weights applied to local robot decisions. The default trail laying behaviour of a robot is to deposit an initial positive concentration (represented by a value) of a marker at the grid square (with position (x, y)) it occupies at each timestep. This initial value is represented in the model as a value m with initial magnitude 1.0. The model also supports a robot laying a negative pheromone trail in which case the initial value of m would be -1.0. This means that unreinforced marker will be represented as a m value in the range $[-1.0, 1.0]$. The concentration of a marker can be reinforced by a grid square being visited multiple times with the result that the magnitude of reinforced m can increase without bound over time. The magnitude of the marker can also be made to decay over time by specifying a decay rate R_d which is applied to all squares containing a value of m at each timestep. Robots can also have variable sensor range limitations. In this model the sensor range along a straight line path is represented as a scalar value R_{sense} which is measured in terms of a number of grid cells.

3.1 Domain specification

Domain boundaries and objects in the domain can be represented in two ways in the model. There is a static 2D function $f(x, y)$ which can be used to define locations (x, y) which block movement. These can be used to represent objects in the domain where it is assumed that the grid cells have a binary structure (occupied or free). Global values from this function are mapped to a local filter vector $f_i, i = 0..N$ which has values of 0 or 1 and can be used to filter the available local choices. An robot cannot choose squares that have been filtered out and this can be used to set up rigid type boundary conditions.

Alternatively a type of reflective boundary condition can be set up by specifying sufficiently large negative and non-decaying values of $m(x, y, t)$. In this case the boundary value at a square acts as a repulsive force on robots in the system in such a way that an robot will tend to move away in the opposite direction. Combining these methods provide a flexible way to specify complex domains and domain objects.

3.2 Local mappings

There are two important types of mapping that are used in the model. First local vectors v_i of the general form:

$$v_i, i = 0..N \tag{1}$$

are used. Each v_i represents a value related to the probability of making choice i. The range of i represents the available number of local choices. In a square type domain grid this can be used as part of making a movement decision to one of 8 possible adjacent squares (also known as a Moore neighborhood) so in this case $N = 7$ in 1. All the results presented in this paper use $N = 7$.

A superposition of vectors of type v_i can be used to combine a number of different effects that may in-

fluence a movement decision (see main equations below) of an robot. This is a flexible way to incorporate many factors that may influence decisions in the model. Because many of the factors that can influence movement decisions are related to the state of the environment it is necessary to extract global values from $m(x, y, t)$ or $f(x, y)$ for the $(N + 1)$ local squares visible by an individual robot in the system at each timestep. This is the first type of mapping used (global to local).

The other type of mapping relates to negative values of $m(x, y, t)$. Because the value of v_i represents a probability value it should be greater than zero. Therefore there needs to be a way to map negative values of $m(x, y, t)$ (which represent negative weights) to a positive probability value. Since values of $m(x, y, t)$ are used to influence robot movement in our model it has been chosen to map a negative value of $m(x, y, t)$ to a positive value of the same magnitude but in the opposite direction. This means a repulsive effect on an robot movement decision is made equivalent to an attractive effect in the opposite direction.

3.3 Main update equations

First we need to define a set of local vectors. A decision vector d_i represents the final probabilities of an robot moving in one of i possible directions; a weight vector w_i represents weight values assigned to each of the i possible path choices which are locally visible to an robot and which represents an estimate of the maximum gain associated with choosing to move along a particular path; a random decision vector r_i represents pure random choice (ie equal values for all i directions); a reinforcement decision vector l_i represents locally visible values of the component of $m(x, y, t)$ due to reinforcement; and a filter vector f_i represents a mask on which of the i choices are allowed or not allowed due to rigid boundary conditions.

The basis of the model is a biased random walk equation applied to the movement of each robot in the system at each time step Δt. For each robot $A_k, k = 1..M$, where M is the total number of robots, this equation takes the form:

$$d_i = [(1 - \nu_i)r_i + \nu_i m_i]f_i \qquad (2)$$

where $i = 0..N$ and the parameter ν_i controls the degree to which pure random choice (represented by r_i) dominates over weighted or biased choice represented by m_i. There is also a step size s associated with the random walk. Equation 2 and the stepsize together define the biased random walk. The vector m_i represents weights based on locally visible values of $m(x, y, t)$ in the neighborhood of the walker. Given only a single walker and m_i purely based on trail laying (of marker) by the single walker, then a high value of ν_i leads to a random walk that approximates a self avoiding walk.

Using 2 it is possible to experiment with local adaptive control strategies for selecting values of ν_i and s for each walker at each time step. This paper focuses on the choice of ν_i and experiments with adaptive values of s are left to future work. The approach taken in this paper is to link increases in the value of ν_i to direction choices that are weighted by reinforced values of the $m(x, y, t)$ field. This creates a subset of (greedy) candidate directions from the complete set of possible direction choices. This subset may be empty if no reinforced field values are locally visible. These direction choices are candidates for increased bias in their probability of selection against random choice. The aim of a local adaptive strategy for ν_i is to further reduce this subset of possible candidates (if this can be done) by inhibiting the choice of some candidates. In this paper the strategy is evaluated by calculating a set of measures along a path up to distance R_{sense} (maximum sensor range) in each candidate direction. Initially 3 measures have been chosen: (1) the magnitude of reinforcement; (2) the distance $dist$ that can be traversed before any boundaries or field objects (such as another robot walker) are encountered; (3) a path gradient estimate calculated along the path up to $dist$. Using a simple perceptron type activation function a strategy is activated depending on the values of the measures and a set of weights. Strategies that are evaluated but not activated inhibit the choice of a candidate direction.

More formally, we need to first calculate a reinforcement vector l_i:

$$l_i = \begin{cases} m_i - 1.0 & \text{if } m_i > 1.0 \\ 0 & \text{if } m_i < 1.0 \end{cases}$$

Then for each l_i a vector $r_{ij}(j = 0..2)$ is calculated which contains the required measures. Given the three chosen measures, we have $r_{i0} = l_i$, r_{i1} as the distance that can be traversed along the path before a boundary or object is encountered and r_{i2} is the path gradient estimate.

We define a strategy π for selecting a local control variable z as π_z. The local strategy for selecting $\nu(x, y, t)$ is then defined as:

$$\pi_\nu : \nu = \nu_0 + p(\mathbf{r}_i, \theta) * f(l_i, \nu_0, \nu_{max}) \qquad (4)$$

where f is an monotonically increasing scaling function of l_i with lower limit ν_0 and upper limit ν_{max}, ν_0 equals a constant positive parameter σ in the range $[0, 1]$, ν_{max} is 1.0, θ is a threshold value, and p is an activation function defined as:

$$p(\mathbf{r}_i, \theta) = \begin{cases} 1.0 & \text{if } \sum_{i=0}^{2} w_i r_i > \theta, \\ -1.0 & \text{if } \sum_{i=0}^{2} w_i r_i < \theta. \end{cases}$$

where the w_i are weight parameters. Equation 4 is equivalent to conditionally increasing ν, after certain conditions are satisfied which result in the activation function firing, by a factor between ν_0 and ν_{max} which depends on the magnitude of l_i. This is based on a simple perceptron like behaviour to choose an approximate new scaled value of ν in the range $[\nu_0, \nu_{max}]$.

The choice of a gradient measure in caculating $\mathbf{r}_i$ is used as a rough estimate of whether movement along a particular path will lead towards less frequently traversed parts of the domain (based on the $m(x, y, t)$ field). As part of calculating the gradient estimate, a scalar discount factor ζ is introduced to discount values of m which are more than one cell away from the walker and which may have uncertainty associated with them due to changes in a cell value that will take place by the time the walker gets to that cell.

At each timestep Δt the simulation moves each robot according to (Eq. 2) and updates m which is discretized on a i by j grid as follows:

$$m_{i,j}(t + \Delta t) = m_{i,j}(t) + \xi_{i,j}(t), \quad \forall 0 < i < nx, 0 < j < ny \quad (6)$$

where $\xi_{i,j}(t)$ is based on the pheromone deposited by robots during their movement (positive or negative) and the pheromone decay rate R_d for the current timestep. On each timestep the following heuristic is applied:

1. Calculate m_i from $m(x, y, t)$
2. Perform mappings (global to local, negative to positive)
3. $\forall m_i > 1.0$, set $l_i = m_i - 1.0$ (else $l_i = 0$) and normalize m_i, l_i
4. Calculate a set of r_i (greedy candidate measures)
5. Apply local strategies π_ν
6. Calculate and normalize d_i

where normalization for a vector v_i is calculated as $v_i^{norm} = \frac{v_i}{\sum_i v_i}$. Due to normalization the strengthening of the probability of making one choice leads automatically to the weakening of the probability of choosing the other available choices.

3.3.1 Model parameters

The key model parameters are M the total number of robots, n_x, n_y representing the discrete grid dimensions, σ representing a constant ratio of random effects versus bias effects on robot movement, σ which represents a threshold used to control the effect of reinforcement in the model, R_d which represents the decay rate applied to the magnitude of field values $m(x, y, t)$, $nsteps$ represents the number of steps taken in the simulation, and r_{seed} representing the seed value used for the random number generator. $w_i (i = 0..1)$ are weight parameters which are in the series of initial experiments described below are assigned values using the following rules: $w_0 = 1.0/\mathbf{r}_0$, $w_1 = \mathbf{r}_1/R_{\text{sense}}$; if $((|\mathbf{r}_2| > 1) AND (\mathbf{r}_2 > 0))$ then $w_2 = -1.0$ else $w_2 = 1.0$.

4 RESULTS

A series of initial tests have been performed using a Java implementation of the model where Java.util.Random was used as the random number generator for randomly chosen motion. These tests have all been performed with the following fixed model parameters: $nx = 70, ny = 70$, $r_{seed} = 2392093$, $R_d = 0.0$, and with variable values for the other parameters. The metric chosen here to evaluate the model for domain coverage has been percentage coverage (of the bounded domain) versus time (number of iterations). A standard sigmoid function has been used for the scaling function f with $\exp(-(\mathbf{r}_0 - \sigma))/2$.

The first set of tests used just one robot to search a maze type domain. Initially tests were undertaken with no use of strategies and different values of σ (i.e. this is the fixed value of $\nu = \sigma$ case). It was found that as σ was increased in this case (with the random walk becoming more like a self-avoiding walk) that the domain coverage also increased. At domain boundaries a reflective boundary condition needs to be strongly enforced and this is achieved using high negative marker values and reinforcement (in direction choice) to define the boundary. In Fig. 1 the use of local strategies to adapt the value of ν was compared with the fixed value (No strategy) case for $\sigma = 0.75$. The results appear to indicate that the use of adaptive local strategy can increase performance (domain coverage versus time) but more experiments need to be performed to examine this in detail. The strategy parameter sets used were Strategy1 = ($\theta = 2.8, \sigma = 0.75$) and Strategy2 = ($\theta = 2.5, \sigma = 0.75$) with other parameters set as described above.

The next set of simulation tests demonstrated that the model scales well as the number of robots is increased. Figure (3) shows that as the number of robots is increased from 1, to 8 the use of local strategy adaption still provides benefits over the no strategy case but it is not as pronounced as in the single robot test. The strategy parameter set ($\theta = 2.5, \sigma = 0.95$) is used for these simulation runs.

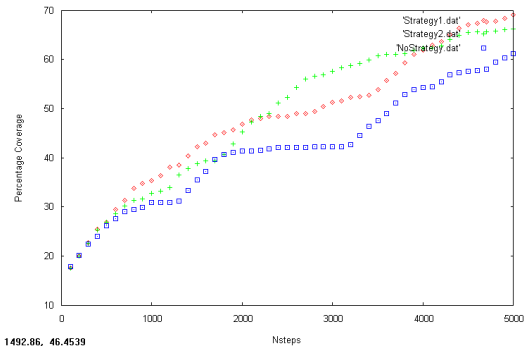

Figure 2: Single robot search on maze domain with $n_x = 70, n_y = 70$, $r_{seed} = 2392093$, $R_d = 0.0$, and $\sigma = 0.75$. "NoStrategy.dat" does not apply local strategy selection. "Strategy1.dat" and "Strategy2.dat" both apply strategy selection. Starting position is $(55, 35)$.

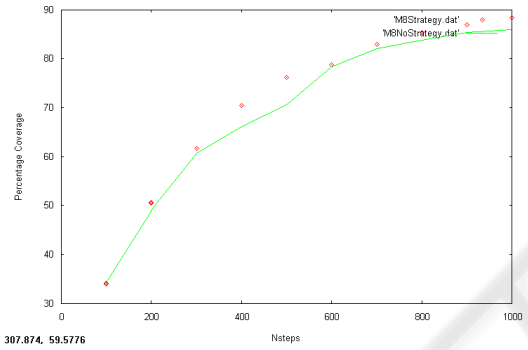

Figure 3: Multi robot search using 8 robots on box canyon domain with $n_x = 70, n_y = 70$, $r_{seed} = 2392093$, $R_d = 0.0$, and $\sigma = 0.95$. Starting positions are $((10, 60), (10, 10), (60, 60), (60, 10), (15, 60), (15, 10), (50, 60), (50, 10))$.

5 CONCLUSION

A simulation model is presented which can simulate multiple robots using stigmergy as a decentralized coordination mechanism for solving foraging and domain coverage type tasks. The model is derived from a biased random walk using localized decisions as the basis of walker movement. It is scalable to any number of robots and able to represent complex domains/environments. This model can be used to study the effect of random versus biased decision making (based on weighted estimates of path suitability) through a set of local control parameters which allow experimentation with locally adaptive strategy selection. A set of simple tests is used to demonstrate some of the model features. The model introduced here will allow the study of optimal local strategy for movement to be studied in detail in a series of experiments. These more detailed experiments will be the subject of future work.

REFERENCES

Anderson, R. (1988). *Random-walk learning: A neurobiological correlate to trial-and-error, In: Neural Networks and Pattern Recognition*. Academic Press, Boston.

Azar, Y., Broder, A., Karlin, A., Linial, N., and Phillips, S. (1992). Biased random walks. In *24th Annual ACM Symposium on Theory of Computing*, pages 1–9.

Balch, T. and Arkin, R. (1994). communication in reactive multiagent robotic systems. *Autonmous Robots*, 1(1):27–52.

Baltes, J. and Anderson, J. (2003). Flexible binary space partitioning for robotic rescue. In *Proc. Int. Conf. IEEE IROS 2003 - Intelligent Robots and Systems*.

Coppersmith, D. and Diaconis, P. (1987). Random walks with reinforcements. *Stanford Univ. Preprint*.

Davis, B. (1990). Reinforced random walk. *Prob. Th. Rel. Fields*, 84:203–229.

Glover, F. and Kochenberger, G. A. (2003). *Handbook of Metaheuristics*. Kluwer publishing.

Gonzales-Banos, H. and Latombe, J. (2002). Navigation strategies for exploring indoor environments. *Int. J. Robot. Res.*, 21(10-11):829–848.

Holland, O. and Melhuish, C. (1999). Stigmergy, self-organization, and sorting in collective robotics. *Artificial Life*, 5:173–202.

Igel, C., Friedrichs, F., and Wiegand, S. (2005). Evolutionary optimization of neural systems: The use of strategy adaptation. In *Trends and Applications in Constructive Approximation, Int. Series of Numerical Mathematics*. Birkhuser Verlag.

Kaelbling, L., Cassandra, A., and Littman, M. (1994). Acting optimally in partially observable stochastic domains. In *Twelfth National Conference on Artificial Intelligence*.

Lam, L. and Pochy, R. (1993). Active-walker models: growth and form in non-equilibrium systems. *Computation simulation*, 7:534.

Montgomery, J. and Randall, M. (2002). Anti-pheromone as a tool for better exploration of search space. In *Third International Workshop on Ant Algorithms, ANTS 2002*.

Patterson, D. J. and Kautz, H. (2001). Autowalksat:a self-tuning implementation of walksat. *Electronic Notes in Discrete Mathematics (ENDM)*, 9.

Rekleitis, I., Dudek, G., and Milios, E. (2001). Multi-robot collaboration for robust exploration. *Annals of Mathematics and Artificial Intelligence*, 31(1-4):7–40.

Stachniss, C. and Burgard, W. (2003). Mapping and exploration with mobile robots using coverage maps. In *Proc. of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*.

Wagner, I. A., Lindenbaum, M., and Bruckstein, A. M. (1999). Distributed covering by ant-robots using evaporating traces. *IEEE transactions on robotics and automation*, 15(5).