10.1109/SIS.2005.1501606

Detection of Multiple Source Locations using a Glowworm Metaphor with Applications to Collective Robotics

K.N. Krishnanand[1] and D. Ghose[2]

***Abstract*— This paper presents a glowworm swarm based algorithm that finds solutions to optimization of multiple optima continuous functions. The algorithm is a variant of a well known ant-colony optimization (ACO) technique, but with several significant modifications. Similar to how each moving region in the ACO technique is associated with a pheromone value, the agents in our algorithm carry a luminescence quantity along with them. Agents are thought of as glowworms that emit a light whose intensity is proportional to the associated luminescence and have a circular sensor range. The glowworms depend on a local-decision domain to compute their movements. Simulations demonstrate the efficacy of the proposed glowworm based algorithm in capturing multiple optima of a multimodal function. The above optimization scenario solves problems where a collection of autonomous robots is used to form a mobile sensor network. In particular, we address the problem of detecting multiple sources of a general nutrient profile that is distributed spatially on a two dimensional workspace using multiple robots.**

I. Introduction

Ant colony optimization (ACO) techniques [1,2] have interesting applications in collective robotics research [4,5,6,7] because they provide a distributed approach to solve difficult optimization and coordination problems that are typical to the field. With this motivation, we present a glowworm swarm based algorithm that finds solutions to optimization problems with multiple optima continuous functions.

A. Emergent behavior

These ant algorithms are primarily inspired by the concept of emergent group behavior or swarming behavior [1], which is reminiscent of those found in biological examples such as flock of birds, school of fish, social ant colonies, and swarms of social bacteria. According to this concept, simple primitives and local interactions between individual members of the colony lead to intelligent group behavior. Usually, the emergent collective behavior is utilized in an appropriate manner in order to perform a desired complex task.

B. Applications to Collective Robotics

The optimization scenario that we consider in this paper solves problems where a collection of autonomous robots is used to form a mobile sensor network. In particular, we address the problem of detecting multiple sources of a general nutrient profile, that is distributed spatially on a two dimensional workspace, using multiple robots. This problem is representative of a wide variety of applications. For instance, during a nuclear spill, or a hazardous chemical spill [7] in an industrial plant, it is imperative to detect the multiple sources of the spills and contain all the spills in a quick and efficient manner before they can cause a great loss to the environment and people in the vicinity. Other applications include search and rescue in a building on fire or putting off forest fires (where robots have to depend on temperature gradients [4] to reach all the different fire locations). Another application is localization and decommissioning of hostile sensors or transmitters, scattered over a landscape, by sensing signals radiating from them.

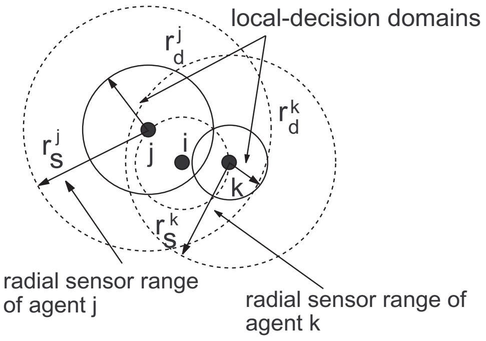
local-decision domains
r_d^j
r_d^k
r_s^j
r_s^k
j
i
k
radial sensor range of agent j
radial sensor range of agent k

Fig. 1. ${}^{k}_{d}$ ( ) = ( ) ${}^{j}_{d}$ ${}^{k}_{s}$ ${}^{j}_{s}$. *Agent i is in the sensor range of (and is equidistant to) both j and k. But, they have different decision-domains. Hence, only j uses the information of i*

The paper is organized as follows. Prior work is discussed in Section II. A complete description of the glowworm algorithm—the main contribution of the paper—is given in Section III. Section IV provides two theoretical proofs related to the luminescence update rule of the glowworm algorithm. Section V presents the simulation results followed by a discussion of a specific simulation case and the effect of the discrete nature of agent movements on convergence in Section VI. A brief mention of the collective robotics platform that we built to test the algorithms developed in this paper is given in Section VII. The paper concludes with a few remarks in Section VIII.

II. Related Work

Detecting, seeking, and tracking gradient-inducing sources has received attention recently [4,5,6] in the

This work is partially supported by a project grant from the Ministry of Human Resources Development, India, and by the DRDO-IISc Mathematical Engineering Program.

[1] Graduate Student, Department of Aerospace Engineering, Indian Institute of Science, Bangalore 560 012, India. Email: krishna@aero.iisc.ernet.in

[2] Professor, Department of Aerospace Engineering, Indian Institute of Science, Bangalore 560 012, India. Email: dghose@aero.iisc.ernet.in

0-7803-8916-6/05/$20.00 ©2005 IEEE

collective robotics community. Dhariwal et al. [4] present an approach where mobile robots emulate bacterial-taxis behavior [8] in order to move towards and locate gradient-inducing sources such as a fire (induces a temperature gradient in its vicinity) and an oil spill (induces a concentration gradient of oil). Hayes et al. [5] describe a spiral-surge algorithm in which a collection of autonomous mobile robots use a combination of spiral plume finding, surge, and spiral casting behaviors to find the source of an odor plume. Muller et al. [8] present an optimization algorithm based on a model of bacterial chemotaxis to find solutions to multimodal functions. *Taxis* refers to the locomotory response of a cell to its environment. In a taxis, a cell responds such that it changes both its direction and the duration of the next movement step.

Our algorithm is loosely based on Bilchev and Parmee's [1, 2] ant-colony approach to continuous optimization problems, but with several significant modifications. In their approach, a finite set of regions are first randomly placed in the search space. Each path between the nest location and a region i is associated with a virtual pheromone $\tau_i(t)$ at each iteration t. Initially, $\tau_i(t=0) = \tau_0$ for all agents. The probability that an agent selects region i is given by

$$p_i(t) \quad = \quad \frac{\tau_i^\alpha(t)\eta_i^\beta(t)}{\sum_{j=1}^N \tau_j^\alpha(t)\eta_j^\beta(t)} \qquad (1)$$

where $\eta_i(t)$ reflects the local-desirability of a portion of the solution, α and β represent relative weights, and N is the number of regions. The agent then moves to the selected region's center, measures the value of the objective function at that point, moves a short distance in a random direction, shifts the region's center to the new point if it finds an improvement in the solution, and then comes back to the nest. The pheromone update associated with the region is given by

$$\tau_i(t+1) = \begin{cases} (1-\rho)\tau_i(t) + \gamma\Delta J, & \text{if } \Delta J > 0 \\ (1-\rho)\tau_i(t), & \text{Otherwise} \end{cases} \qquad (2)$$

where ΔJ is the improvement made in the solution and γ is a proportionality constant.

This process is repeated with a new probability distribution according to (1). With the increase in number of iterations, the pheromone concentration associated with *inferior* regions decay (and may disappear eventually) and *good* regions get reinforced with time, finally converging to the solution.

III. The Glowworm Metaphor

In our algorithm, instead of finite regions randomly placed in the search space, we talk of physical entities (agents) that are randomly distributed in the workspace. Similar to how each moving region is associated with a pheromone value, the agents in our algorithm carry a luminescence quantity along with them. Agents are thought of as glowworms that emit a light whose intensity is proportional to the associated luminescence and have a circular sensor range whose radius is r_s.

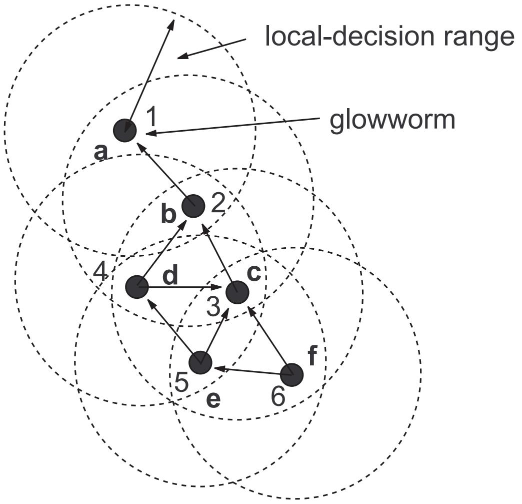

Fig. 2. *Emergence of a directed graph based on the relative luminescence level of each agent and availability of only local information. Agents are ranked according to the increasing order of their luminescence values. For instance, the agent* **a** *whose luminescence value is highest is ranked '1' in the figure.*

The glowworm belongs to a family of beetles known as the *Lampyridae* or fireflies and produces natural light (bioluminescence) that is used as a signal to attract a mate. The light is also used to attract prey. If a glowworm is hungry its light glows brighter in order to make the attraction of insects more effective [3].

Using the glowworm algorithm, we notice that the number of peaks captured is a strong function of the radial sensor range. For instance, if the sensor range of each agent covers the entire workspace, all the agents move to the global optimum point and the local optima remain undetected. In order to detect multiple peaks, the sensor range cannot be kept constant and must be made a varying parameter. For this purpose, we associate each agent i with a local-decision domain whose radial range r_d^i is dynamic in nature $(0 < r_d^i \leq r_s^i)$. In Bilchev and Parmee's algorithm, the probability distribution that determines the region to be selected requires pheromone-information of all the regions. However, note that agents in our algorithm have to depend only on information available in the local-decision range to make their decisions (Figure 1). The resulting algorithm is highly decentralized and caters well to the requirements of collective robotics systems.

A. Algorithm description

Our algorithm starts by placing the glowworms randomly in the workspace so that they are well dispersed. Initially, they contain equal quantity of luminescence. Each iteration consists of a luminescence update phase followed by a movement phase based on a transition rule. The luminescence update depends on the function value at the glowworm position and so, even though a glowworm starts with the same luminescence value, it changes due to the function value. During the movement phase, each glowworm decides, using a probabilistic mechanism, to move towards a neighbor that has a luminescence value more than its own, that is, they are attracted to neighbors that glow brighter. Figure 2 shows the emergence of a directed graph among a set of six glowworms based on their relative luminescence levels and availability of only local information. For instance, there are four glowworms (a, b, c, and d) that have relatively more luminescence than the glowworm e. Since e is located in the sensor-overlap region of c and d, it has only two possible directions of movement. For each glowworm i, the probability of moving towards a neighbor j is given by:

$$p_j(t) \quad = \quad \frac{\tau_j(t)}{\sum_{j \epsilon N_i(t)} \tau_j(t)} \tag{3}$$

where $j \ \epsilon \ N_i(t), N_i(t) = \{j : d(i,j) < r_d^i$ and $\tau_i(t) < \tau_j(t)\}$, t is the time index, $d(i,j)$ represents the euclidian distance between glowworms i and j, $\tau_j(t)$ represents the luminescence level associated with glowworm j at time t, and r_s represents the radial range of the luminescence sensor. A suitable function is chosen to adaptively update the local-decision domain range of each glowworm. This is given by:

$$r_d^i(t+1) \quad = \quad \frac{r_s}{1 + \beta D_i(t)} \tag{4}$$

where

$$D_i(t) \quad = \quad \frac{N_i(t)}{\pi r_s^2} \tag{5}$$

is the neighbor-density of agent i at iteration t and β is a constant parameter. Figure 3 shows the plot of the local-decision radial range as a function of number of neighbors, assuming different values for β.

During the luminescence update phase, each glowworm adds to its previous luminescence level, a luminescence quantity proportional to the value of the objective function at that point. Also, a fraction of the previous luminescence value is subtracted from the previous luminescence value to simulate the decay in luminescence with time. The luminescence update rule is given by:

$$\tau_j(t+1) \quad = \quad \max\{0, (1-\rho)\tau_j(t) + \gamma J_j(t+1)\} \tag{6}$$

where ρ is the luminescence decay constant ($0 < \rho < 1$) and γ is a proportionality constant for enhancing the luminescence as a function of $J_j(t)$ which represents the value of the objective function at agent j's location. The pseudo-code of the glowworm-metaphor algorithm is given below.

B. Glowworm swarm optimization algorithm

```
$deploy\_agents\_randomly$;
$\forall\ i$, set $\tau_i(0) = \tau_0$
$\forall\ i$, set $r_d^i(0) = r_0$
set maximum iteration number $= iter\_max$;

while $(t \leq iter\_max)$ do:
{
    for each glowworm $i$ do:
    {
        $\tau_i(t) = update\_luminescence(i)$;
                  % (using (6))
    }
    for each glowworm $i$ do:
    {
        $N_i(t) = find\_neighbors(i)$;
        for each glowworm $j \ \epsilon \ N_i(t)$ do:
        {
            $p_j(t) = find\_probability(j)$; % using(3)
        }
        $j = select\_glowworm\_j$; % (using $p_j(t)$)
        $X_i(t) \ \leftarrow \ X_i(t) + s d_{ij}(t)$;

            where $d_{ij}(t) = \frac{(X_j(t) - X_i(t))}{\|X_j(t) - X_i(t)\|}$,

        $s$ is the step size, and $X_i(t)$ is the location of
            glowworm $i$ at iteration $t$.

        $r_d^i(t) = update\_decision\_range(i)$;
                          %(using (4))
    }
    $t \ \leftarrow \ t+1$;
}
```

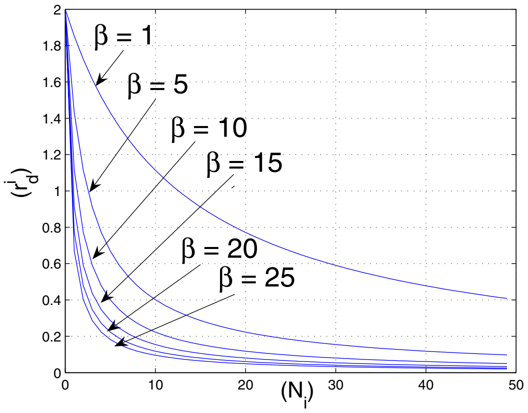

Fig. 3. *Plot of* $^{i}_{d}$ *as a function of* $_{i}$

C. Conventional ant-algorithms vs glowworm metaphor approach

Usually ant-algorithms differ in the manner in which the pheromone update is made. Travelling salesman problem (TSP) is a typical example of an ACO application where pheromone update on city routes is an exact bio-mimicry of the trail-laying phenomenon found in foraging of ants. In general, most algorithms involve adding a pheromone value at specific locations where the agents visit. The pheromone level at these spots decrease with time (and may disappear eventually) in order to emulate the evaporation behavior of actual pheromones. However, the luminescence update mechanism of our algorithm is inspired by the insight provided by the principle of stigmergy rather than a bio-mimicry of such behavior found in ants. The difference mainly arises from the fact that the luminescence locations do not stay at places visited by agents but move along with the agents. Therefore, the luminescence level indicates the net improvement made by the agent by traversing an emergent path from its initial location to the present location. An agent that makes relatively more net improvement in the objective function during its movements has a higher luminescence level and attracts more agents towards it than others, thus enabling agents to move towards favorable places of the environment. In this luminescence update scheme, the number of luminescence sources is equal to the number of agents. Note that, when the solution is reached, the number of luminescence locations will be approximately equal to the number of maxima of the objective function as most of the agents settle at one of the multiple peaks.

IV. Luminescence Convergence Proof

We provide two theoretical proofs for the luminescence update rule proposed in the previous section. First, we prove that, due to luminescence decay, the maximum luminescence level τ_{max} is bounded asymptotically. This proof is similar to Stützle and Dorigos' [9] proposition proving the bounded nature of pheromone levels in their ant-colony algorithm. Second, we show that the luminescence τ_j of all glowworms co-located at a peak X_i converge to the same value τ_i^*.

Theorem 1: Assuming that the luminescence update rule in (6) is used, the luminescence level $\tau_j(t)$ for any glowworm j is bounded above asymptotically as follows:

$$\lim_{t \to \infty} \tau_j(t) \leq \lim_{t \to \infty} \tau_j^{max}(t) = \left(\frac{\gamma}{\rho}\right) J_{max} \tag{7}$$

where J_{max} is the global maximum value of the objective function.

Proof: Given that the maximum value of the objective function is J_{max} and the luminescence update rule in (6) is used, the maximum possible quantity of luminescence added to the previous luminescence level at any iteration t is γJ_{max}. Therefore, at iteration 1, the maximum luminescence of any glowworm j is $(1-\rho)\tau_0 + \gamma J_{max}$. At iteration 2, it is $(1-\rho)^2\tau_0 + [1 + (1-\rho)]\gamma J_{max}$, and so on. Generalizing the process, at any iteration t, the maximum luminescence $\tau_j^{max}(t)$ of any glowworm j is then given by:

$$\tau_j^{max}(t) = (1-\rho)^t \tau_0 + \sum_{i=0}^{t-1} (1-\rho)^i \gamma J_{max} \tag{8}$$

Clearly,

$$\tau_j(t) \leq \tau_j^{max}(t) \tag{9}$$

Since $0 < \rho < 1$, from (8) we have that

$$\text{as } t \to \infty, \tau_j^{max}(t) \to \left(\frac{\gamma}{\rho}\right) J_{max} \tag{10}$$

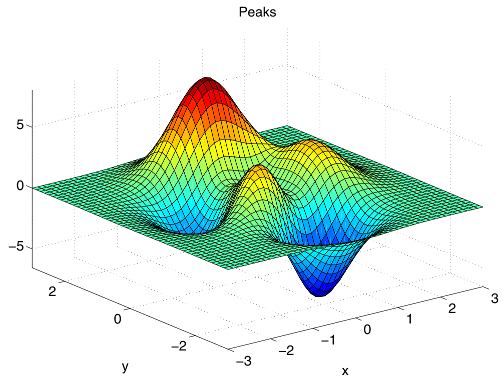

Fig. 4. *3-D plot of the nutrient profile*

Using (9) and (10), we have the result in (7). □

Theorem 2: For all glowworms j co-located at peak-locations X_i^* associated with objective function values $J_i^* \leq J_{max}$ (where $i = 1, 2, \ldots, m$ and m is the number of peaks), assuming that the luminescence update rule in (6) is used, we have that, $\tau_j(t)$ increases or decreases monotonically and asymptotically converges to $\tau_i^* = \left(\frac{\gamma}{\rho}\right) J_i^*$.

Proof: According to (6), $\tau_j(t) \geq 0$ always. The stationary luminescence τ_i^* associated with peak i satisfies the following condition:

$$\begin{aligned} \tau_i^* &= (1-\rho)\tau_i^* + \gamma J_i^* && (11)\\ \Rightarrow \tau_i^* &= \left(\frac{\gamma}{\rho}\right) J_i^* && (12) \end{aligned}$$

If $\tau_j(t) < \tau_i^*$ for glowworm j co-located at peak-location X_i^*, then using (12) we have

$$J_i^*(t) > \left(\frac{\rho}{\gamma}\right) \tau_j(t) \tag{13}$$

$$\begin{aligned} \text{Now, } \tau_j(t+1) &= (1-\rho)\tau_j(t) + \gamma J_i^* && (14)\\ &> (1-\rho)\tau_j(t) + \gamma \left(\frac{\rho}{\gamma}\tau_j(t)\right) && (15)\\ \Rightarrow \tau_j(t+1) &> \tau_j(t) && (16) \end{aligned}$$

i.e., $\tau_j(t)$ increases monotonically.

Similarly, if $\tau_j(t) > \tau_i^*$ for glowworm j co-located at peak-location X_i^*, then using (12) and (14), it is easy to show that

$$\tau_j(t+1) < \tau_j(t) \tag{17}$$

i.e., $\tau_j(t)$ decreases monotonically.

From (16) and (17), it is clear that the luminescence $\tau_j(t)$ of glowworm j co-located at a peak-location X_i^* asymptotically converges to τ_i^*. □

			${}^{i}_{d}$	
50	0.4	0.6	1.5	0.01

TABLE I

LIST OF PARAMETERS USED IN SIMULATIONS.

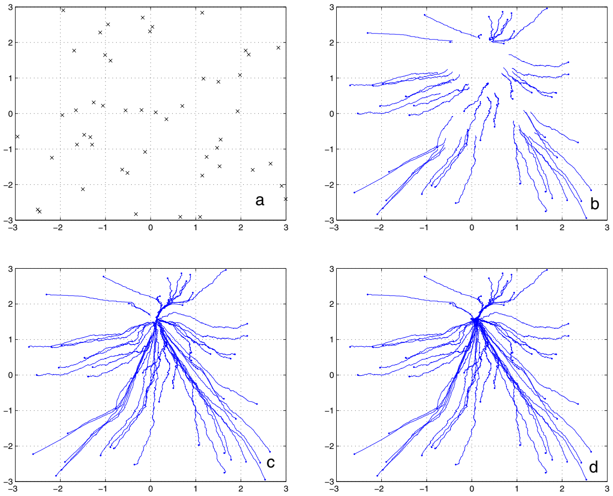

Fig. 5. *(a) Agents are randomly distributed on a two-dimensional workspace of size 6X6 units. (b), (c), and (d) represent the emergence of the solution after 250, 500, and 750 iterations, respectively; local-decision domain range* ${}^{i}_{d}$ *= 3 (only one peak is detected).*

V. SIMULATION RESULTS

The algorithm is validated through MATLAB simulations. The distribution of the nutrient profile in the environment is modelled by a continuous function $J(x, y)$ given by:

$$\begin{aligned} J(x, y) &= 3(1-x)^2 \exp(-(x^2) - (y+1)^2) \\ &- 10(x/5 - x^3 - y^5) \exp(-x^2 - y^2) \\ &- (1/3) \exp(-(x+1)^2 - y^2) \end{aligned} \quad (18)$$

Here, $J(x, y)$ has multiple peaks and valleys (Figure 4). Local maxima are located at (0.06, 1.625), (-0.45, -0.55), and (1.35, 0.05). The nominal values of various parameters used in the simulations are shown in Table I where n is the number of glowworms. A set of 50 glowworms are randomly deployed in a two-dimensional workspace of size $6X6$ square units. Figure 5(a) shows the initial placement.

A. Constant local-decision range

As a first step, the radial range r_d^i of each glowworm is kept constant, in order to characterize the sensitivity of the number of peaks detected to the size of the local-decision domain. As noted earlier, the local-decision range greatly influences the determination of various peaks. When the decision-range is more than 2, all the glowworms moved to the global maximum. Figure 5 shows the emergence of the solution as the number of iterations increases when $r_d^i = 3$. When the range is taken as 1.85, only two peaks are identified (Figure 6). A radial range of

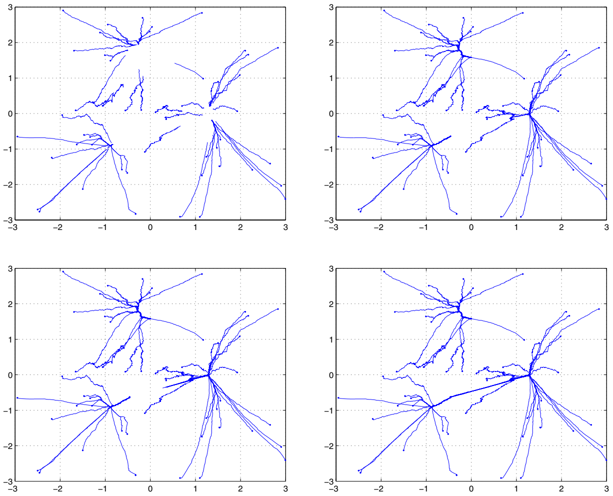

Fig. 6. *Emergence of the solution after 250, 500, 750, and 1000 iterations, respectively; local-decision domain range* ${}^{i}_{d}$ *=1 85 (two peaks are detected).*

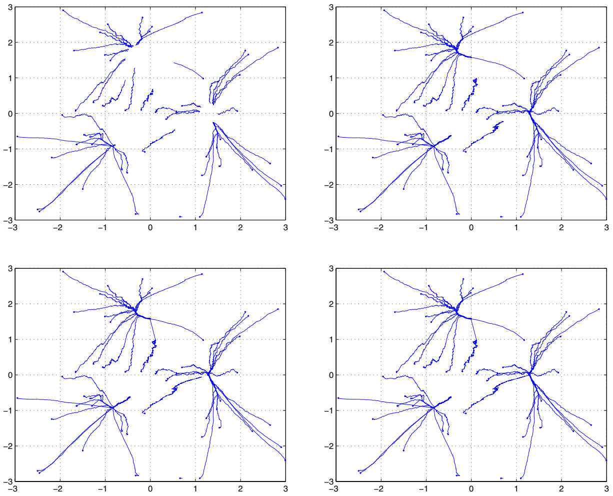

Fig. 7. *Emergence of the solution after 250, 500, 750, and 1000 iterations, respectively; local-decision domain range* ${}^{i}_{d}$ *=1 8 (three peaks are detected).*

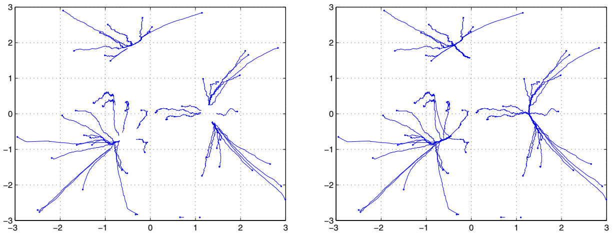

Fig. 8. *Emergence of the solution after 250 and 500 iterations, respectively; local-decision domain range* ${}^{i}_{d}$ *= 1 5 (three peaks are detected).*

1.8 or 1.5 resulted in detection of all the three peaks (Figures 7 and 8).

B. Variable local-decision range

We observe that the number of peaks captured is a strong function of the radial sensor range. Since we assume that *a priori* information about the objective function (e.g., number of maxima and minima) is not available, the decision-range cannot be kept constant and must be made a varying parameter. A value of $\beta = 25$ was chosen for the experiments after some trial and error as it resulted in satisfactory performance. Figure 9(a) shows the emergence of the solution when the local-decision domain range is made to vary according to (4) at each iteration t. During this simulation, a value of $r_d^i(0) = 3$ is chosen for each glowworm i. Note that all the peaks are detected within 2000 iterations. In particular, 24, 10, and 14 glowworms get co-located at (0.06, 1.625), (-0.45, -0.55), and (1.35, 0.05), respectively. The remaining two glowworms are always isolated from the rest of the group. Hence, they are not attracted towards any source and thereby, get co-located at a position that is in close proximity to their initial locations. Figure 9(c) shows the co-location of all the glowworms on the equi-contour plot of the objective function. Figure 9(b) shows the luminescence history of each glowworm. Note that after the steady state is reached, all the glowworms co-located at a particular location possess the same luminescence quantity. According to Theorem 2, the value of τ_j of all glowworms j co-located at a peak-location X_i^* is given by $(\frac{\gamma}{\rho})J_i^*$. It was observed that the value of τ at the three peak-locations (0.06, 1.625), (-0.45, -0.55), and (1.35, 0.05) are 12.5 (=$(\frac{0.6}{0.4})$8.1), 5.67 (=$(\frac{0.6}{0.4})$3.78), and 5.4 (=$(\frac{0.6}{0.4})$3.6), respectively. Note that the τ values obtained in simulations are in exact agreement with their analytical values given by Theorem 2 (Figure 9(b)). We observe that, for a majority of the glowworms, $r_d^i(t)$ is oscillatory in nature initially, but exhibits a monotonically decreasing tendency with the increase in iteration number. Figure 9(d) shows the plot of $r_d^1(t)$ of glowworm 1 that is initially deployed at $(0.03, 2.44)$ and finally gets co-located at the peak-location $(0.06, 1.625)$.

VI. Discussion

A. Response to presence of a forbidden region

In our simulations, we find that a simple local-gradient based algorithm captures the multiple peaks of the objective function by using relatively less number of evaluations when compared to the ant-based algorithm. In a work on "social foraging swarms," Liu and Passino [6] assume that agents can compute local gradients of the nutrient profile. However, our algorithm has advantage in situations where agents have no way of collecting gradient information. Figure 10 depicts a situation where the agent A is unaware of the source location because of the occlusion created by the obstacle O. However, the inter-agent communication between neighbors makes global

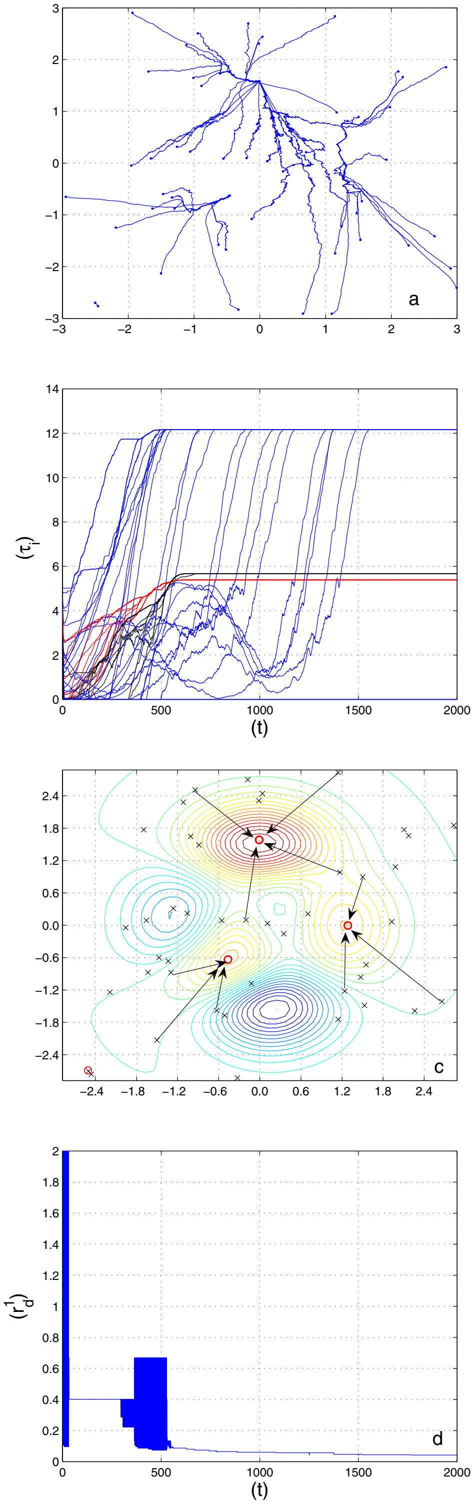

Fig. 9. *(a) Emergence of the solution after 200 iterations, when ${}^{i}_{d}$ is varied according to equ.(4) (all the three peaks are detected). (b) Luminescence history of all the 50 glowworms. (c)After steady state is reached, a collection of glowworms get co-located at each peak, excepting two glowworms that get co-located close to their initial locations as they are always isolated from the rest of the group members. (d) Plot of ${}^{1}_{d}(\ )$ as a function of iteration number .*

information available locally to the agent, providing the agent with feasible directions to move towards the source.

The above situation is simulated by biassing the random initial placement of the glowworms in a manner that none of the glowworms is deployed in a circular (forbidden) region of radius 1 unit centered at (0.6, -0.6). The size of the forbidden region and r_d^i $(= 2)$ are chosen such that a glowworm located on the boundary of the forbidden region cannot sense another glowworm that is located on the other side of the region. Also, the forbidden region is placed such that the peak located at (1.35, 0.05) lies within the forbidden region. The simulation result in Figure 11(a) shows that, following the biassed-random deployment as described above, there is no instant when a glowworm enters the forbidden region and since one of the peaks is obscured by the forbidden region, only two peaks are detected. Consider the glowworm ***a*** that is initially deployed at (1.47, -2.31). Figure 11(b) shows the glowworms that lie within the decision-domain of ***a***. Since the region of intersection **S** (refer to Figure 11(b)) of ***a***'s decision-domain and the forbidden region is void of any neighbors, ***a*** avoids entrance into the forbidden region. As the glowworms ***b***, ***c***, and ***d*** have higher luminescence values than its own (Table II), glowworm ***a*** obtains three feasible directions to move at iteration 1. Since ***a*** obtains feasible directions of movement at every time step, it continues to move by avoiding the forbidden region and finally gets co-located at the global-maximum.

B. *Effect of step-size on convergence*

According to the glowworm algorithm presented in this paper, a glowworm with the maximum luminescence at a particular iteration remains stationary during that iteration. Ideally, the above property leads to a dead-lock situation when all the glowworms are located such that the peak-location lies outside the convex-hull formed by the glowworm positions. Since the agent movements are restricted to the interior region of the convex-hull, all the glowworms converge to a glowworm that attains maximum luminescence value during its movements within the convex-hull. As a result, all the glowworms get co-located away from the peak-location. However, the discrete nature of the movement update rule automatically takes care of this problem which could be described in the following way. During the movement phase, each glowworm moves a distance of finite step-size s towards a neighbor. Hence, when a glowworm i approaches closer to a glowworm j that is located nearest to a peak such that the inter-agent distance becomes less than s, i crosses the position of j and becomes a leader to j. In the next iteration, i remains stationary and j crosses the position of i thus regaining its leadership. This process of interchanging of roles between i and j repeats until they reach the peak. In Figure 11(a), the peak-location (-0.45, -0.55) lies outside the convex-hull C. Initially, a majority of the glowworms in the region C reach close to the location (-0.84, -1.06) which lies within the convex-hull. However, due to the constant step-size, all the glowworms climb the gradient-hill as described above and get co-located at the peak in the vicinity.

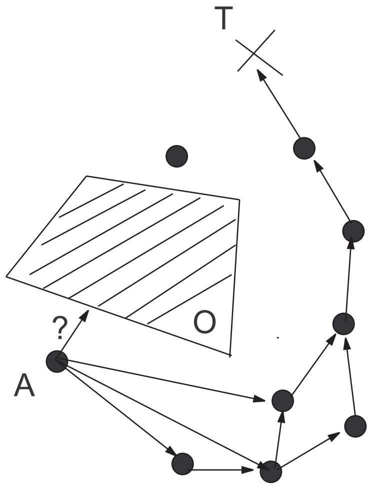

Fig. 10. *Situation where inter-agent communication helps a robot to select a feasible direction towards the source*

Glowworm	$_i(1)$	$_i(1)$
a	(1.47, -2.31)	2.79
b	(1.38, -2.91)	2.96
c	(1.87, -2.28)	2.95
d	(2.80, -1.76)	3.00
e	(0.30, -2.40)	1.75
f	(0.10, -1.78)	0.00

TABLE II

VII. KINBOTS

We built a set of four wheeled-mobile robots called *Kinbots*, to conduct experiments related to collective robotics [10]. These Kinbots can be used to implement the algorithms developed in this paper. Kinbots are equipped with infrared sensor-based interaction modules (Figure 13(a)) that provide a hardware capability to implement luminescence emission/detection and a *leader-following* behavior (Visit our website at http://guidance.aero.iisc.ernet.in/robotics for more information). Each Kinbot broadcasts the luminescence value in terms of a 8-bit data (serially) that could be detected by other Kinbots present in a circular neighborhood. Therefore, when a Kinbot receives luminescence values from multiple directions it considers neighbors whose luminescence values are more than its own and then decides (in a probabilistic manner) to move towards one of them, before it starts processing the next iteration. Experiments using Kinbots are in progress and their results will be reported later.

VIII. CONCLUDING REMARKS

We propose to use a glowworm metaphor based approach in order to find solutions to optimization problems that deal with multiple optima functions. The algorithm developed in this paper could be applied to a class

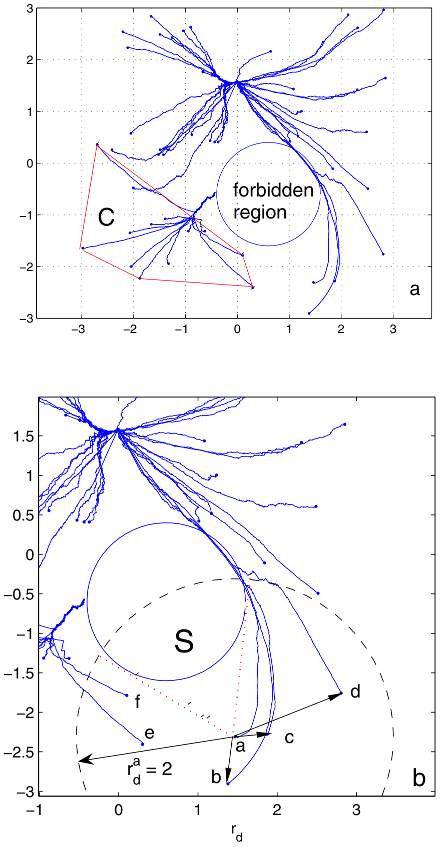

Fig. 11. *(a) Response of the glowworm algorithm to the presence of a circular forbidden region of radius* 1 *unit centered at (0.6,-0.6). Emergence of the solution after 1000 iterations;* $^{i}_{d} = 2$ *(b) Glowworm* ***a*** *has three feasible directions to move at iteration* 1*.*

of problems related to collective robotics. In particular, the glowworm metaphor based optimization algorithm can be used to detect multiple source locations of a general nutrient profile, that is distributed spatially on a two dimensional workspace, using a collection of mobile robots. Based purely on local information agents can exhibit taxis behavior towards nutrient sources. However, we observe that the number of source locations that are detected is a strong function of the local-decision domain of each agent. Therefore, in an extension of the simulation experiments, we have made the decision domain range to vary as a function of the *neighbor-density*, which improved the performance of the algorithm in terms of maximizing the number of peaks detected. For further work, we want to consider more complicated multi-nodal continuous functions to test the applicability of our algorithm. Initial experiments with Kinbots indicate that they demonstrate good candidacy to test the efficacy of the algorithms developed in this paper.

REFERENCES

[1] E. Bonabeau, M. Dorigo, G. Theraulaz. *Swarm Intelligence: From Natural to Artificial Systems*, Oxford University Press, 1999, pp. 183-203.

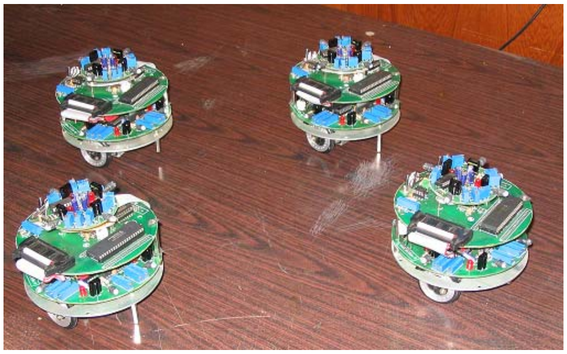

Fig. 12. *A set of four Kinbots*

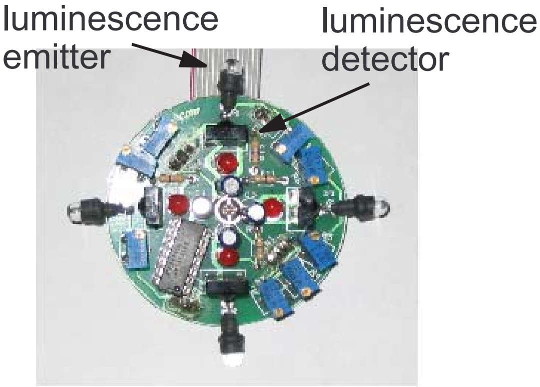

Fig. 13. *The interaction module used to implement luminescence emission/detection and leader following behavior*

[2] G. Bilchev and I.C. Parmee, "The ant colony metaphor for searching continuous design spaces," *Proc. of AISB Workshop on Evolutionary Computing Lecture Notes in Computer Science*, 1995, pp. 25-39.

[3] J.Tyler. "Glow-worms,"
http://website.lineone.net/∼galaxypix/Tylerbookpt1.html

[4] A. Dhariwal, G.S. Sukhatme, and Aristides A.G. Requicha "Bacterium-inspired robots for environmental monitoring," *Proceedings of IEEE International Conference on Robotics and Automation*, New Orleans, LA, April 2004, pp. 1436-1443.

[5] A.T. Hayes, A. Martinoli, and R.M. Goodman, "Distributed odor source localization," *IEEE Sensors Journal*, Vol. 2, No. 3, June 2002, pp. 260-271.

[6] Y. Liu and K.M. Passino. "Stable social foraging swarms in a noisy environment," *IEEE Transactions on Automtic Control*, Vol. 49, No. 1, January 2004, pp. 30-43.

[7] R. Russell, D.Theil, R.Deveza, and A.Mackay-Sim. "A robotic system to locate hazardous chemical leaks," *Proceedings of IEEE International Conference on Robotics and Automation*, Nagoya, Japan, 1995, pp. 556-561.

[8] S.D. Muller, J.Marchetto, S.Airaghi, and P.Koumoutsakos. "Optimization based on bacterial chemotaxis," *IEEE Transactions on Evolutionary Computation*, Vol. 6, No. 6, February 2002, pp. 16-29.

[9] T. Stützle and M.Dorigo. "A short convergence proof for a class of ant colony optimization algorithms," *IEEE Transactions on Evolutionary Computation*, Vol. 6, No. 4, August 2002, pp. 358-365.

[10] K.N. Krishnanand and D. Ghose. "Kinbots: A mobile robot platform for collective robotics applications," *Technical Report* GCDSL/2004/07, Department of Aerospace Engg., IISc, Bangalore, August 2004.