Theoretical foundations for rendezvous of glowworm-inspired agent swarms at multiple locations

K.N. Krishnanand*, D. Ghose

Guidance, Control, and Decision Systems Laboratory, Department of Aerospace Engineering, Indian Institute of Science, Bangalore, India

Received 4 April 2007; received in revised form 14 November 2007; accepted 19 November 2007

Available online 24 November 2007

Abstract

We present theoretical foundations for a variation of the multi-agent rendezvous problem involving design of local control strategies that enable agent swarms, with hard-limited sensing ranges, to split into disjoint subgroups, exhibit simultaneous taxis behavior toward, and eventually rendezvous at, multiple unknown locations of interest. The algorithm used to solve the above problem is based on a glowworm swarm optimization (GSO) technique, developed earlier, that finds multiple optima of multi-modal objective functions. We characterize the various phases of the algorithm that help us to develop a theoretical framework required for analysis. In particular, we show through simulations that the implementation of the GSO algorithm at the individual agent level gives rise to two major phases at the group level – splitting of the agent-swarm into subgroups and local convergence of agents in each subgroup to the peak locations. We provide local convergence results under certain restricted set of assumptions, leading to a simplified model of the algorithm, making it amenable to analysis, while still reflecting most of the features of the original algorithm. In particular, we find an upper bound on the time taken by the agents to converge to an isolated leader and on the time taken by the agents to converge to one of the leaders with non-isolated and non-overlapping neighborhoods. Finally, we show that agents under the influence of multiple leaders with overlapping neighborhoods asymptotically converge to one of the leaders. We present some illustrative simulations to support the theoretical findings of the paper.

Keywords: Glowworm swarm optimization; Multiple source localization; Multi-robot rendezvous

1. Introduction

Consensus problems in multi-agent networks, in particular, robotic networks, appear in different forms [1–3], where a collection of agents transits from an initially random state to a final steady state such that all the members of the group eventually agree upon their individual state values. The state could represent physical quantities such as heading angle, frequency of oscillation, position, and so on. Vicsek et al. [1] analyze the alignment of heading angles of multiple particles using the approach of statistical mechanics. In synchronization of coupled oscillators, a consensus is reached regarding the frequency of oscillation of all agents [2]. The multi-agent rendezvous problem, posed by Ando et al. [3], involves devising local control laws that enable all the members to steer toward and eventually rendezvous at a single unspecified location.

Localization of signal sources using mobile robot swarms has received some attention recently in the collective robotics community. Examples of such signal sources include sound [4], light [5], leaks in pressurized systems [6], hazardous plumes/aerosols resulting from nuclear/chemical spills [7, 8], fire-origins in forest fires [9], deep-sea hydrothermal vent plumes [10], hazardous chemical discharge in water bodies [11], oil spills [12], etc. The problem is compounded when there are multiple sources. For instance, several forest fires at different locations give rise to a temperature profile that peaks at the locations of the fire. Similar phenomenon can be observed in nuclear radiations and electromagnetic radiations from signal sources. In all the above situations, there is an imperative need to simultaneously identify and neutralize all the multiple sources using a swarm of robots before they cause a great loss to the environment and people in the vicinity.
Most prior work related to source localization was devoted to seeking, detecting, and tracking of a single emission source location. Dhariwal et al. [5] present an approach where mobile robots emulate bacterial-taxis behavior [13] in order to move toward, and locate, gradient-inducing sources. Hayes et al. [14] describe a spiral surge algorithm in which a collection of autonomous mobile robots use spiral plume finding, surge, and spiral casting behaviors to find the source of an odor plume.

However, very little attention has been given to simultaneous localization of multiple sources [15]. This problem has particular relevance to situations where it is imperative to locate and contain all the radiating sources in a quick and efficient manner before they can cause a great loss to the environment and people in the vicinity. Cui et al. [16] present a multiple source localization algorithm in which mobile agents use a grid map to represent the unknown environment, collect concentration values from all other agents through ad hoc communication, and calculate a positive gradient direction using a biasing expansion swarm approach (BESA). According to this approach, each agent moves to one of the neighboring eight cells on which the net influence (called the biasing parameter) of other agents is maximum. Each agent’s influence is proportional to the concentration of the diffused source at its location and inversely proportional to the square of its distance to the cell. Note that each agent uses global information to decide its movements.

Based on the nature of the emission source and the ambient medium, the source localization problem can be broadly classified into two categories:

1. Signals such as sound, light, and other electromagnetic radiations propagate in the form of a wave [16]. Therefore, the nominal source profile that spreads in the environment can be represented as a multi-modal function. For instance, a set of light sources randomly placed in an indoor environment gives rise to a light intensity profile on the floor that is characterized by bright spots (maxima) surrounding the light source locations and shadows (minima) underneath objects such as chairs and tables. Hence, the problem of localizing the signal origins can be modelled as optimization of multi-modal functions.

2. Chemical signals emitted by sources such as hazardous aerosols, gas leaks, etc., disperse through the environment by molecular diffusion and bulk flow [16,17]. The chemical source gradually dissolves into the ambient fluid medium resulting in odor plume. A plume can be defined as those regions of space that contain the set of all molecules released from a single source [18]. Turbulence in the flow of ambient fluid renders the plume discontinuous and patchy where the patches of ambient fluid are interposed between patches of odor [19]. Owing to these reasons, the source profile in these cases cannot be modelled using a static multi-modal function and should be represented using dynamic gas models that account for factors like diffusion and turbulence.

One of the main goals of the work on the GSO algorithm is to evaluate its effectiveness in handling multiplicity of sources. Therefore, we restrict our attention to the application of GSO algorithm to the first class of source localization problems where the assumption of using a multi-modal function to represent the nominal source profile is justified.

In this paper, we provide theoretical foundations for a variation of the multi-agent rendezvous problem involving design of local control strategies that enable groups of mobile agents with hard-limited sensing ranges to split into subgroups, exhibit simultaneous taxis behavior toward, and eventually rendezvous at, multiple unknown locations of interest. Solutions to the above problem can be used to perform the class of multiple source localization tasks discussed above. Our approach is based on a glowworm swarm optimization (GSO) algorithm [20–22] that finds multiple optima of multi-modal objective functions.

The GSO method is inspired from swarm intelligence – a result of actions performed by a large number of relatively simple individuals that are solely based on neighbor interactions and local information from the environment – exhibited by biological swarms like ants, termites, bees, wasps, and bacteria [29]. Usually, the emergent group behavior serves to accomplish certain complex colonial goals. The significant difference between the GSO-based approach and most earlier approaches to agreement problems is the use of a virtual local decision domain by the agents in order to compute their movements.

The GSO algorithm was first introduced in [20] with simulations experiments to demonstrate the algorithm’s capability to achieve simultaneous capture of multiple optima of multi-modal functions. Numerical simulation results to show the algorithm’s efficacy in capturing multiple peaks of a wide range of multi-modal functions were presented in [21–24]. The various parameters of the GSO algorithm were classified into algorithmic constants that are kept fixed and work well for different problem scenarios and variable parameters that influence the performance of the algorithm, and have to be selected.

The GSO algorithm, originally designed for solving optimization problems, is suited for application to collective robotics tasks [25], unlike other bio-inspired optimization algorithms like ant-colony optimization (ACO) and particle swarm optimization (PSO) whose application to similar robotic tasks would require significant algorithmic modifications. Application of the GSO algorithm to multiple source localization tasks is demonstrated through real-robot experiments in [25], where four wheeled mobile robots implement the GSO algorithm to collaborate and achieve a sound source localization task. Application of the GSO algorithm to hazard sensing in ubiquitous environments is described in [26]. However, a theoretical treatment of the algorithm addressing issues of convergence has not been considered in these papers. In the present paper, we take up this aspect and present the theoretical foundations for the GSO algorithm. Preliminary theoretical results that prove convergence under a restricted set of assumptions are presented in [27]. Before we present our theoretical framework, we characterize the various phases of the algorithm that helps us to develop a theoretical framework required for analysis. We show
through simulations that the implementation of the GSO algorithm at the individual agent level gives rise to two major phases at the group level: (i) Splitting of the agent-swarm into subgroups and (ii) Local convergence of agents in each subgroup to the peak locations. We provide local convergence results under certain restricted set of assumptions, leading to a simplified model of the algorithm, making it amenable to analysis, while still reflecting most of the features of the original algorithm. In particular, we find an upper bound on the time taken by the agents to converge to an isolated leader and on the time taken by the agents to converge to one of the leaders with non-isolated and non-overlapping neighborhoods. Finally, we show that agents, under the influence of multiple leaders with overlapping neighborhoods, asymptotically converge to one of the leaders. We present some illustrative simulations to support the above theoretical findings.

The paper is organized as follows. A brief description of the underlying principles and various phases of the GSO algorithm and a simulation experiment to show its efficacy in capturing multiple peaks of multi-modal functions are presented in Section 2. In Section 3, we present the group level phases of the GSO algorithm that help us to develop a theoretical framework required for analysis. In Section 4, we derive the simplified GSO model that we consider for analysis, by making modifications to the actual algorithm, and present the main theoretical results of the paper. In Section 5 we present simulation results using the simplified the GSO model. Finally, we provide some concluding remarks in Section 6.

2. The GSO algorithm

2.1. Overview

In the GSO algorithm, the agents are initially deployed randomly in the objective function space. The agents in the GSO algorithm carry a luminescence quantity called luciferin along with them. Agents are thought of as glowworms that emit a light whose intensity of luminescence is proportional to the associated luciferin. Each glowworm uses the luciferin to (indirectly) communicate the function-profile information at its current location to the neighbors. Rybski et al. [30] implement a foraging algorithm with MinDart, a team of simple robots, where each robot uses a lighted beacon (to attract/recruit other nearby robots) that is loosely related to the luminescence-based attractor mechanism used in the GSO algorithm. 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. However, a priori information about the objective function (e.g., number of maxima and minima and inter-peak distances) is not available. Therefore, it is difficult to find an optimal value of \( r \) that will work well for different source placements. Moreover, sensors used by real robots have a fixed hard-limited sensing range. Therefore, in order to detect multiple peaks, we introduce a virtual local decision domain whose range \( r_j \) (Fig. 1(a)) is adaptive and bounded above by a circular sensor range of the Agent \( a \), which is closest to the source is ranked ‘1’. A value of \( \rho = 1 \) is considered in this example. Therefore, from (1), the luciferin level reduces with increase in distance from the source. A directed graph emerges based on the relative luciferin level of each agent and availability of only local information. Each Glowworm \( i \) regards only those incoming luciferin data as useful that are broadcast by other glowworms located within the adaptive local decision domain in order to compute its movements. Each glowworm selects a neighbor that has a luciferin value more than its own, using a probabilistic mechanism, and moves toward it. The above idea of agents moving toward neighbors that are located in relatively favorable regions of the environment is similar to the trophallactic behavior-inspired control strategies presented by Schmickl and Crailsheim [31] for a multi-robotic cleaning application, where each robot queries the “nutritional” status of the robots in its local neighborhood and uses this information to navigate uphill in the gradient.

2.2. Algorithm description

The underlying principles of the GSO algorithm and numerical experiments to show the algorithm’s efficacy in capturing multiple peaks of a wide range of multi-modal functions are presented in [20–22]. Therefore, in this paper, we limit ourselves to a brief description in order to aid the analytical modelling of the GSO algorithm. As mentioned earlier, in signal source applications, the nominal source profile that spreads in the environment can be represented as a multi-modal function. For instance, a set of light sources randomly placed in an indoor environment gives rise to a light intensity profile on the floor that is characterized by bright spots (maxima) surrounding the light source locations, dark regions (minima) that are located far from source locations, and shadows (minima) underneath objects such as chairs and tables. While finding the light sources corresponds to finding the multiple maxima of the light intensity profile, moving toward hiding regions in the environment corresponds to seeking the multiple minima of the function profile. Generalizing the above idea, the problem of

![Fig. 1. (a) Agent \( i \) is in the sensor range of (and is equi-distant to) both \( j \) and \( k \). But, it is contained within the decision domain of \( j \) and outside the decision domain of \( k \). Hence, only \( j \) uses the information of \( i \). (b) Agents are ranked according to the increasing order of their distance to the source for instance, the Agent \( a \) which is closest to the source is ranked ‘1’. A value of \( \rho = 1 \) is considered in this example. Therefore, from (1), the luciferin level reduces with increase in distance from the source. A directed graph emerges based on the relative luciferin level of each agent and availability of only local information. Each Glowworm \( i \) regards only those incoming luciferin data as useful that are broadcast by other glowworms located within the adaptive local decision domain in order to compute its movements. Each glowworm selects a neighbor that has a luciferin value more than its own, using a probabilistic mechanism, and moves toward it. The above idea of agents moving toward neighbors that are located in relatively favorable regions of the environment is similar to the trophallactic behavior-inspired control strategies presented by Schmickl and Crailsheim [31] for a multi-robotic cleaning application, where each robot queries the “nutritional” status of the robots in its local neighborhood and uses this information to navigate uphill in the gradient.](image-url)
localizing the signal origins can be modelled as optimization of multi-modal functions. The exposition of the algorithm is presented for maximization problems. However, the algorithm can be easily modified and used to find multiple minima of multi-modal functions.

The algorithm starts by placing the glowworms randomly in the workspace so that they are well dispersed. Initially, they contain equal quantity of luciferin. Each iteration consists of a luciferin update phase followed by a movement phase based on a transition rule.

**Luciferin-update phase:** The luciferin update depends on the function value at the glowworm position and so, even though all glowworms start with the same luciferin value during the initial iteration, these values change according to the function values at their current positions. During the luciferin update phase, each glowworm adds, to its previous luciferin level, a luciferin quantity proportional to the measured value of the sensed profile (temperature, radiation level) at that point. In the case of a function optimization problem, this would be the value of the objective function at that point. Also, a fraction of the luciferin value is substracted to simulate the decay in luciferin with time. The luciferin update rule is given by:

\[ \ell_i(t + 1) = (1 - \rho)\ell_i(t) + \gamma J_i(t + 1) \]  

where, \( \rho \) is the luciferin decay constant (\( 0 < \rho < 1 \)) and \( \gamma \) is the luciferin enhancement constant and \( J_i(t) \) represents the objective function value at agent \( i \)’s location at time \( t \).

**Movement-phase:** During the movement-phase, each glowworm decides, using a probabilistic mechanism, to move toward a neighbor that has a luciferin value more than its own. That is, they are attracted to neighbors that glow brighter. The illustration in Fig. 1(b) shows a set of six glowworms placed in the vicinity of a signal source. We assume that the signal strength reduces monotonically with distance from the source and a value of \( \rho = 1 \) is considered for each agent. Therefore, using (1), the luciferin value of each agent is a function of only its relative position with respect to the source and reduces monotonically with increase in distance from it. In Fig. 1(b), the bisector \( b_1 \) of the line segment \( x_a x_b \) shows that the source is closer to \( a \) than \( b \). The bisector \( b_2 \) of the line segment \( x_c x_d \) shows that the source is closer to \( c \) than \( d \). Proceeding in a similar way, we can sort the agents according to increasing order of their distance from the source or alternatively, according to the decreasing order of their luciferin values as \( a, b, c, d, e, f \), and \( f \) such that \( \ell_a > \ell_b > \ell_c > \ell_d > \ell_e > \ell_f \). A directed graph emerges among the set of six glowworms based on their relative luciferin levels and availability of only local information. For instance, there are four glowworms (\( a, b, c, \) and \( d \)) that have relatively more luciferin 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 toward a neighbor \( j \) is given by:

\[ p_{j}(t) = \frac{\ell_j(t) - \ell_i(t)}{\sum_{k \in N_i(t)} \ell_k(t) - \ell_i(t)} \]  

where, \( j \in N_i(t) \neq \emptyset \). \( N_i(t) = \{ j : d_{ij}(t) < r_d(t) \text{ and } \ell_i(t) < \ell_j(t) \} \) is the neighborhood of Glowworm \( i \) at time \( t \), \( d_{ij}(t) \) represents the euclidean distance between glowworms \( i \) and \( j \) at time \( t \), \( r_d(t) \) represents the variable local decision range associated with Glowworm \( i \) at time \( t \), and \( r_s \) represents the hard-limited radial range of the luciferin sensor. Note that \( N_i(t) \) should be a non-empty set for \( p_{j}(t) \) to be defined. When \( N_i(t) \) is an empty set, Glowworm \( i \) is isolated, and hence remains stationary at time \( t \). Let the Glowworm \( i \) select a Glowworm \( j \in N_i(t) \) with \( p_{j}(t) \) given by (2). Then, the discrete-time model of the glowworm movements can be stated as:

\[ x_i(t + 1) = x_i(t) + s \left( \frac{x_j(t) - x_i(t)}{\|x_j(t) - x_i(t)\|} \right) \]  

where, \( x_i(t) \in \mathbb{R}^m \) is the location of glowworm \( i \), at time \( t \), in the \( m \)-dimensional real space \( \mathbb{R}^m \), \( \| \| \) represents the euclidean norm operator, and \( s (>0) \) is the step size.

The expression for \( p_{j}(t) \) in (2) needs some justification. The probability distribution is formulated in such a way that the chances of a glowworm selecting to move toward a neighbor is directly proportional to the neighbor’s relative luciferin level with respect to its own luciferin level. Usage of luciferin differences in the probability formula allows the luciferin levels to take negative values.

A suitable function is chosen to adaptively update the local decision domain range of each glowworm. This is given by:

\[ r_d(t + 1) = \min(r_s, \max(0, r_d(t) + \beta(n_t - |N_i(t)|))) \]  

where, \( \beta \) is a constant parameter and \( n_t \in N \) (set of natural numbers) is used as a threshold parameter to control the number of neighbors.

**Local search due to leapfrogging:** An Agent \( i \) is said to leapfrog over another Agent \( j \) when it overtakes and stops at a point that is ahead of Agent \( j \). Leapfrogging among agents in the GSO algorithm occurs in the following way. According to the GSO algorithm, a glowworm with the maximum luciferin at a particular iteration remains stationary during that iteration. This 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 agent movements are restricted to the interior region of the convex-hull, all the glowworms converge to a glowworm that attains maximum luciferin value during its movements within the convex-hull. As a result, all the glowworms get co-located (meet at a single location) away from the peak location. Two agents are said to be co-located with each other when their positions are identical. A precise mathematical definition of co-location is deferred to a later section (Definition 5 in Section 4). However, the discrete nature of the movement-update rule automatically takes care of this problem in the following way. During the movement phase, each glowworm moves a distance of finite step size \( s \) toward a neighbor. Hence, when a Glowworm \( i \) approaches closer to a neighbor \( j \) such that the inter-agent distance becomes less than \( s \), \( i \) leapfrogs over \( j \) and becomes a leader to \( j \). In the next iteration, \( i \) remains
stationary and \( j \) overtakes the position of \( i \) thus regaining its leadership. This process of interchanging of roles between \( i \) and \( j \) occurs repetitively giving rise to a local search behavior of the glowworm pair along a single ascent direction. A group of glowworms use the same principle to perform an improved local search and eventually converge to the peak location. This phenomenon was also observed in simulations in [20–22].

Assume two glowworms \( i \) and \( j \) located in a linear uphill direction of slope \( k \), at a distance \( d(t) \) from each other and with luciferin levels \( \ell_i(t) \) and \( \ell_j(t) \), respectively, at time \( t \) such that \( \ell_i(t) < \ell_j(t) \), \( d(t) < r_d(t) \), and \( d(t) < s \). Therefore, \( i \) leapfrogs over the position of \( j \). Let \( J \) be the function-profile value at \( i \)'s location at time \( t \). Since we assume a constant gradient field, \( J + kd(t) \) is the function-profile value at \( j \)'s location at time \( t \) as well as \( t + 1 \) (since \( j \) is stationary at time \( t \)) and \( J + ks \) is the function-profile value at \( i \)'s location at time \( t + 1 \). The leapfrogging continues at \( t + 1 \) if the following condition is satisfied:

\[
\ell_i(t + 1) > \ell_j(t + 1) \\
\Rightarrow (1 - \rho)\ell_i(t) + \gamma(J + ks) \\
> (1 - \rho)\ell_j(t) + \gamma(J + kd(t)) \\
\Rightarrow \ell_j(t) - \ell_i(t) < \frac{\gamma k(s - d(t))}{1 - \rho}.
\]

From (7), it is clear that \( \rho, \gamma, s, \) and the distance between two glowworms \( d(t) \) play a role in the leapfrogging process.

In a potential robotics application of the GSO algorithm, isolated robots in the initial deployment do not move until they acquire local leaders. This results in an inferior overall search performance. The number of isolated robots can be reduced by ensuring good connectivity (within \( r_s \) sensor range) between robots. However, a balance between connectivity among robots and coverage of the workspace is required for the algorithm to work well for different function profiles. Thus, initial agent density plays an important role in the performance of the algorithm. In a probabilistic sense, uniformly distributed deployment of \( n \) robots in a \( W \times W \) workspace ensures an appropriate average separation of \( \frac{W}{\sqrt{n}} \) between neighboring robots. However, increasing the agent density by increasing \( n \) can reduce instances of isolated robots, but may not improve performance beyond a certain value of \( n \). Order of \( \frac{W}{\sqrt{n}} \) can be used as a measure of agent density required for the algorithm to work.

2.3. Simulation experiment to illustrate GSO

Simulation results demonstrating the capability of the glowworm algorithm to capture multiple peaks of a number of complex multi-modal functions have been reported in [20–22] for constant and variable local decision range cases. We have shown that when constant decision range is used, the number of peaks captured decreases with increase in the value of decision range. Interestingly, when the decision range is made adaptive, even though \( r_d'(0) \) is chosen to be greater than the maximum distance between the peaks, all the peaks are captured [20]. In this paper, we demonstrate the basic working of the algorithm using the Peaks function (Fig. 2) from MathWorks [32]:

\[
J_1(x, y) = 3(1 - x)^2 \exp(-(x^2) - (y + 1)^2) \\
- 10(x/5 - x^3 - y^5) \exp(-x^2 - y^2) \\
- (1/3) \exp(-(x + 1)^2 - y^2).
\]

Peaks is a function of two variables where multiple peaks and valleys are obtained by translating and scaling multiple Gaussian distributions [33]. Local maxima (peaks) are located at \((0, 1.58), (0.46, -0.63), \) and \((1.28, 0)\). Local minima are located at \((-1.4, 0.25)\) and \((0.3, -1.55)\) with different function values. A set of 50 glowworms is randomly deployed in a two-dimensional workspace of size \(6 \times 6 \) square units where \((x, y) \in [-3, 3] \times [-3, 3]\).

Algorithmic constants and parameters: The values of the algorithmic constants – luciferin decay constant \( \rho \), luciferin enhancement constant \( \gamma \), decision range update constant \( \beta \), and desired number of neighbors \( n_1 \) – are kept fixed for all experiments. These values have been arrived at based on a large number of numerical experiments [23]. Note that these are not parameters to be selected by an user as they are fixed irrespective of the problem chosen. Further studies have been reported in [24] to show that the values chosen for the algorithmic constants work well for a wide range of simulation scenarios and that only \( n \) and \( r_s \) are the parameters that influence the algorithm’s performance. In the simulations, a step size of \( s = 0.03 \) is used. The values chosen for the algorithmic constants are \( \rho = 0.4, \gamma = 0.6, \beta = 0.08, \) and \( n_1 = 5 \).

Initially, the radial range \( r_d' \) 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. Fig. 3(a)–(c) show the emergence of the solution, after 200 iterations, when the local decision range \( r_d' \) of all glowworms is kept constant at 2 (only one peak is captured), 1.8 (two
peaks are captured), and 1.5 (all three peaks are captured), respectively.

Fig. 3(d) 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 200 iterations. In particular, 23, 19, and 8 glowworms get co-located at the maxima of $(0, 1.58), (-0.46, -0.63)$, and $(1.28, 0)$, respectively. After the steady state is reached, all the glowworms, co-located at a particular location, possess the same luciferin quantity.

2.4. Modifications of algorithmic behavior for robotic implementation

A robotic implementation of the GSO algorithm would require a collection of mobile robots where each robot has the following capabilities:

(1) Sensing and broadcasting profile value (luciferin level) at its location.

(2) Detection of number of neighbors and their localization with respect to its own position.

(3) Receiving profile values (luciferin levels) from all neighbors.

(4) Selecting a neighbor probabilistically and moving a step distance toward it.

(5) Variation of the local decision range as a function of number of neighbors.

(6) Collision avoidance with obstacles and other robots in the environment.

(7) Leapfrogging behavior for performing local search.

The leapfrogging effect of the GSO algorithm requires a glowworm to move over another glowworm whenever two glowworms are within a step distance $s$ of each other. This is illustrated in Fig. 4(a) where glowworm $A$ moves over glowworm $B$ to reach the position $A(t + 1)$. In Fig. 4(a), $d(<s)$ is the distance between the robots $A$ and $B$ at time $t$. Therefore, when $A$ leapfrogs over the position of $B$, it stops at a distance $s - d$ on the ray $\overrightarrow{AB}$ at time $t + 1$. However, this is not directly realizable with physical mobile robots. We describe three methods to achieve an explicit robotic implementation of the leapfrogging behavior. The first method (Fig. 4(b)) involves interchange of agent-roles, through communication, where robot $B$ moves to the desired position of robot $A$ and robot $A$ replaces robot $B$’s position. The illustration in Fig. 4(c) shows another method where the robot $A$ takes a detour about robot $B$’s position and reaches its desired...
position. As this method is difficult to implement, we propose a modification where the detour is achieved using a blend of two circular paths as shown in Fig. 4(d). We also observe that the obstacle avoidance behavior automatically gives rise to an implicit leapfrogging effect, which is described using the illustration in Fig. 4(e). Robot A alternatively performs collision avoidance with, and seeks to move toward, robot B until it crosses the equi-contour line on which robot B is situated. Thereafter, since robot A becomes a leader to robot B, robot B performs similar movements with respect to robot A, thus leading to an implicit leapfrogging behavior of the glowworm-pair. Results from embodied simulation experiments where the incorporation of collision avoidance behavior into the GSO algorithm enables the glowworms to exhibit implicit leapfrogging behavior, perform local search, and eventually capture multiple signal sources are reported in [24]. Results from real-robot experiments where two robots perform leapfrogging with respect to each other by toggling between basic-GSO and collision avoidance behaviors, using unreliable communication channels, to taxis toward a single light source are also reported in [24].

From the description of the GSO algorithm, which is originally designed for multi-modal optimization problems, the modifications required for real-robot implementation that are mentioned above, and the embodied simulation and real-robot implementation results reported earlier, it is clear that the GSO algorithm is well suited to implementation in a set of mobile robots for collective robotics applications.

2.5. Robustness issues

From a multi-agent systems (MAS) perspective, the performance of GSO algorithm has been shown to be robust to factors like individual agent failure, presence of noise, and handling of undifferentiable function profiles. These are discussed below:

Influence by noisy sensor data: The GSO algorithm was tested in the presence of sensor noise in [28]. A Gaussian distribution was used to model the sensor noise. The GSO algorithm was compared with a gradient-ascent algorithm. The GSO algorithm showed good performance even with fairly high noise levels. There was a degradation of performance only with significant increase in levels of measurement noise. Whereas, gradient-based algorithm degraded fast in the presence of noise.

Non-differentiable objective functions: In [23], we evaluated the behavior of the algorithm on function profiles with discontinuities where the objective function is not differentiable. We considered the Staircase function [34] and Plateau’s function [35] for this purpose. In the Staircase function, the peaks are flat regions that are surrounded by step discontinuities. The Staircase function presents a case where the function value increases in discrete steps as we move, from one stair to the next, toward the highest stair (global peak region). The Plateaus function contains multiple plateaus and is similar to Ackley’s Plateau function described in [35]. Even though, the Plateaus function is similar to the Staircase function in terms of the nature of peaks (both have flat regions as peaks), the plateaus have equal objective function values, thereby providing no uphill direction. Results from simulation experiments on these functions show that the GSO algorithm can handle non-differentiable objective functions [23].

2.6. Existing convergence results

In our earlier work [21], we provided some theoretical results for the luciferin update rule. First, we proved that, due to luciferin decay, the maximum luciferin level \( \ell_{\text{max}} \) is bounded above asymptotically. This proof is similar to Stützle and Dorigos’ [36] proposition proving the bounded nature of pheromone levels in their ant-colony algorithm. Second, we showed that the luciferin value \( \ell_j \) of all glowworms co-located at a peak \( X_i \) converge to the same value \( \ell_i^* \). The theorem statements are given below without proofs (refer to [21] for the proofs):

**Theorem 1.** Assume that the luciferin update rule in (1) is used, the luciferin level \( \ell_j(t) \) for any glowworm \( j \) is bounded above asymptotically as follows:

\[
\lim_{t \to \infty} \ell_j(t) \leq \lim_{t \to \infty} \ell_{\text{max}}(t) = \left( \frac{\gamma}{p} \right) J_{\text{max}}
\]

where, \( J_{\text{max}} \) is the global maximum value of the objective function.

**Theorem 2.** For all glowworms \( j \) co-located at peak locations \( X_i^* \) associated with objective function values \( J_i^* \leq J_{\text{max}} \) (where, \( i = 1, 2, \ldots, m \), with \( m \) as the number of peaks), if the luciferin update rule in (1) is used, then \( \ell_j(t) \) increases or decreases monotonically and asymptotically converges to \( \ell_i^* = \left( \frac{\gamma}{p} \right) J_i^* \).

3. Group level phases of GSO algorithm

Implementation of the GSO algorithm at the individual agent level gives rise to two important phases at the group level that are described below:
Fig. 5. (a) Emergence of agent-movements when $J(x, y)$ is used. (b) Emergence of agent-movements when $J(x, y)$ is used. The swarm splits according to the Voronoi-partition of the peak locations in both the cases.

<table>
<thead>
<tr>
<th>Table 1</th>
<th>Values of $a_i$, $x_i$, and $y_i$ used to generate the function profile $J(x, y)$</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>1</td>
</tr>
<tr>
<td>$a_i$</td>
<td>2.616</td>
</tr>
<tr>
<td>$x_i$</td>
<td>2.933</td>
</tr>
<tr>
<td>$y_i$</td>
<td>-0.559</td>
</tr>
</tbody>
</table>

3.1. Splitting of the agent-swarm into subgroups

The local decision domain update rule (4) enables each agent to select its neighbors in such a way that its movements get biased toward the nearest peak. The above individual agent behavior leads to a collective behavior of agents that constitutes the autonomous splitting of the whole group into subgroups whose number is equal to the number of peak locations and where each subgroup of agents gets allocated to a nearby peak (that is a peak to which the agent-distance, averaged over the subgroup, is minimum among those distances to all the peaks in the environment). The simulation result in Fig. 3(d) demonstrates this group-behavior where the swarm splits into three subgroups and movements of agents in each subgroup get biased toward one of the peaks.

We conduct another set of simulations in order to clearly characterize the splitting behavior of the agent-swarm. In particular, we show how the same initial placement of agents gives rise to different splitting behaviors as conditioned by factors like the placements of various peaks, peak values, and slope of the function profile in the vicinity of the peaks. We consider the $J_2(x, y)$ function (10) for this set of experiments.

$$J_2(x, y) = \sum_{i=1}^{Q} a_i \exp(-b_i((x-x_i)^2 + (y-y_i)^2))$$  \hspace{1cm} (10)$$

where, $Q$ represents the number of peaks and $(x_i, y_i)$ represents the location of each peak. The function $J_2(x, y)$ represents a linear sum of two dimensional exponential functions centered at the peak locations. The constant $b_i$ determines how the function profile changes slope in the vicinity of the peak $i$. Initially, we keep $b_i$ equal for all the individual exponentials by choosing $b_i = 3, i = 1, \ldots, Q$. A workspace of $(-5, 5) \times (-5, 5)$ and a set of ten peaks ($Q = 10$) are considered for the purpose. The values of $a_i$, $x_i$, and $y_i$ are generated according to $a_i = 1 + 2\theta$, $x_i = -5 + 10\theta$, and $y_i = -5 + 10\theta$, where, $\theta$ is uniformly distributed random variable within the interval $[0, 1]$.

Note that $a_i$, $x_i$, and $y_i$ are random variables and each set of instantiations of these random variables gives rise to a different multi-modal function profile. The function profiles generated in the above manner are representative of a varied set of problem scenarios in the sense that they have peaks at different random points giving rise to profiles with closely spaced peaks, distant peaks, and peaks on the edges of the workspace. Moreover, generating $a_i$ randomly for each peak gives rise to profiles with equal/unequal peaks. We generate two different function profiles, each represented by $J_2^1$ and $J_2^2$. The corresponding values of $(a_i, x_i, y_i, i = 1, \ldots, 10)$ for the two function profiles are shown in Tables 1 and 2. Fig. 5(a) and (b) show the emergence of solution for each function profile when $n = 300$ and $r_a = 5$. The random initial agent placement is kept same for both the simulations. We observe that when the slope profiles in the vicinity of the peaks are identical, the swarm splits into subgroups according to the Voronoi-partition of the peak locations (Fig. 5(a) and (b)), that is, agents deployed in the Voronoi-partition of a peak location remain in the same partition, during their movements, and eventually converge to the corresponding Voronoi-center that coincides with the peak location. However, a few agents located near the border region of a Voronoi-partition migrate into adjacent partitions and eventually get co-located at the respective peaks. Moreover, the same placement of agents, with a change in the location of the peaks, gives rise to a different partitioning of the agents. However, note that the splitting behavior of the swarm...
Table 2
Values of \(a_i, x_i, \) and \(y_i\) used to generate the function profile \(J_2(x, y)\)

<table>
<thead>
<tr>
<th></th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
</tr>
</thead>
<tbody>
<tr>
<td>(a_i)</td>
<td>1.822</td>
<td>1.966</td>
<td>2.957</td>
<td>1.210</td>
<td>2.626</td>
<td>1.977</td>
<td>2.761</td>
<td>2.612</td>
<td>2.748</td>
<td>1.503</td>
</tr>
<tr>
<td>(x_i)</td>
<td>-1.214</td>
<td>-3.807</td>
<td>-4.768</td>
<td>-3.613</td>
<td>4.976</td>
<td>-1.716</td>
<td>-1.751</td>
<td>3.798</td>
<td>2.993</td>
<td>0.135</td>
</tr>
<tr>
<td>(y_i)</td>
<td>3.677</td>
<td>-4.550</td>
<td>-2.056</td>
<td>0.772</td>
<td>4.277</td>
<td>0.856</td>
<td>-2.574</td>
<td>0.088</td>
<td>1.222</td>
<td>0.699</td>
</tr>
</tbody>
</table>

respects the Voronoi-partition of the new set of peak locations (Fig. 5(b)). The locations of agents at different time instants \(t = 0, 20, 40, 60\) are plotted in Fig. 6(a)–(d), which show the formation of subgroups as a function of time.

Next, we characterize the agent splitting in the case where different slope profiles \((b_i)\) are used for the individual exponentials in the \(J_2(x, y)\) function. For this purpose, we consider two peaks at \((-3, 0)\) and \((3, 0)\), with \(a_1 = 3, b_1 = 0.8\) and \(a_2 = 2, b_2 = 0.1\), respectively. Fig. 7 shows the emergence of agent movements when 100 agents are randomly deployed in the workspace. The line \(L\), that passes through each equi-valued contour at a point where the gradient shifts direction from one peak to the other, divides the workspace into two basins of attraction. Note that all agents on the concave side of line \(L\) converge to the left peak and a majority (86%) of the agents on the convex side of the line \(L\) converge to the right peak. When there are more than two peaks, we can obtain the basins of attraction for each peak in a similar manner as described above and show that the swarm splits into subgroups according to the basins of attraction and eventually converge to the respective peak locations. It is easy to see that the partitioning of the region obtained based on the basins of attraction of the peaks coincides with the Voronoi-partition of the peaks when their slopes become equal.

### 3.2. Local convergence of agents in each subgroup to the peak locations

The relative initial placement of the agents with respect to various peaks in the environment gives rise to different subgroup peak configurations. Accordingly, the respective local convergence behaviors are different from each other. Among these, we consider two major configurations that occur frequently:

**Configuration 1:** The peak is located within the convex-hull of the initial positions of agents in the subgroup. The convergence behavior of this configuration can be explained in the following way. For simplicity, we consider a radially symmetric function profile with a single peak at the center and an initial placement of three agents \(a, b,\) and \(c\) as shown in...
Each agent makes a deterministic movement toward \( a \) or \( b \) (since, \( \ell_a > \ell_b > \ell_c \)). Note from Fig. 8(b) that the agent movements at any time instant are within the convex-hull of all the current positions of agents. Agent \( a \) does not move until after two time steps (i.e., at \( t_p + 2 \)), when \( b \) would have crossed the equi-valued contour \( C_a(t_p + 2) \), leading to the condition \( \ell_b > \ell_a \). Now, \( b \) remains stationary and \( a \) starts moving toward \( b \). This cycle repeats, leading to the asymptotic convergence of agents to the peak.

**Configuration 2:** The peak is located outside the convex-hull of initial agent positions and all the agents are situated on one side of the peak. In order to describe the convergence behavior, we consider the same radially symmetric function profile with a single peak at the center. However, we consider an initial placement of agents \( a, b, \) and \( c \) such that \( b \) and \( c \) are located on one side of the tangent line \( T \) at agent \( a \)’s location as shown in Fig. 8(c).

At time instant \( t_p \), agent \( a \) remains stationary, agent \( b \) makes a deterministic movement toward \( a \), and agent \( c \) moves either toward \( a \) or \( b \) (since, \( \ell_a > \ell_b > \ell_c \)). Agent \( a \) does not move until after two time steps (i.e., at \( t_p + 2 \)), when \( b \) leapfrogs over \( a \), leading to the condition \( \ell_b > \ell_a \) (Fig. 8(d)). Unlike in the previous case, agents movements at any instant are not necessarily within the convex-hull of the current agent positions. For example, at \( t_p + 2 \), agent \( a \) moves outside the convex-hull when it leapfrogs over \( b \). However, the leapfrogging mechanism ensures that the agents eventually converge to the peak.

4. **Theoretical framework for a simplified GSO model**

We restrict our analysis of the GSO algorithm to the local convergence of agents to a leader (to be defined more precisely later). Leaders may represent agents that are closer to the peaks than the other agents. For instance, in Fig. 8, agent \( a \) is the leader. Note that multiple peaks may give rise to multiple leaders. From the above analysis of local convergence behavior of the agents in the vicinity of a peak, it is clear that the leader does not move until another agent crosses the equi-valued contour that passes through the leader’s position in the first case, and until another agent leapfrogs over the leader as in the second case. We obtain the simplified model by making the following modifications to the actual GSO algorithm:

**Modification 1:** Each agent \( i \) contains a constant luciferin value \( \ell_i \).

**Modification 2:** The local decision domain range is kept constant and is made equal to the maximum sensing range \( r_s \).

**Modification 3:** The step size \( s \) is modified such that an agent reaches the leader’s location in one step, whenever it is situated within a distance \( s \) to the leader.

The above assumptions are reasonable within the theoretical framework of local convergence to leaders that we consider here and can be justified as follows: We consider multi-modal function profiles with a property that the inter-peak distance is more than \( r_s \) and the function value in the local vicinity of each peak, monotonically decreases with distance from the peak in all directions. However, the function need not be radially symmetric at each peak. Note that during agent-movements in such a local region, even though the absolute values of agents’ luciferin levels change with time, the relative order of luciferin levels of the agents in the same subgroup do not change, which is also reflected by the constant luciferin case used for our analysis. Usually, when multiple leaders (corresponding to multiple peaks) are located with inter-leader distances of the order of \( r_s \), a typical situation occurs where an agent is located in the \( r_s \)-neighborhoods of, and hence under the influence of, all the leaders. The primary advantage of using an adaptive local decision domain is to enable an agent to select its neighbors such that its movements get biased toward a nearby leader leading to its co-location with it in finite time. Keeping the range of the decision domain constant modifies the above...
behavior in the sense that each leader has an influence on the agent which is a function of its associated luciferin level as well as its distance to the agent, leading to the asymptotic convergence of the agent to one of the leaders. However, a careful observation will reveal that this assumption still models the agent movements in the actual algorithm during the period when local decision domain encompasses multiple leaders and until a stage is reached when the decision domain shrinks and encompasses only the nearest leader. The modification made to the step size $s$ influences the first subgroup-peak configuration in the following way: Note from the description given earlier that the leader (agent which is closest to the peak) changes whenever another agent crosses the equi-valued contour passing through the current leader’s position. However, with the modification of $s$, there exists a time when all agents are within $s$-distance to the current leader and hence they converge in one step to that leader. The modification in $s$ affects the second subgroup-peak configuration discussed above by removing the leapfrogging behavior of the actual GSO algorithm, leading to the convergence of the agents to the one that is located closest to the peak. Note that the algorithmic constants – luciferin decay constant $\rho$, luciferin-enhancement constant $\gamma$, decision-range update constant $\beta$, and desired number of neighbors $n_i$ – need not be considered during the analysis of the simplified GSO model as we assume that agents maintain constant luciferin and local decision range values. However, in [24, we show in simulations that fixed values for these algorithmic constants work well for different scenarios.

Therefore, from the above discussion, it is clear that while the above modifications lead to a simplified GSO model, making it amenable to analysis, the model still reflects most of the important features of the actual algorithm and hence the analysis carried out here is also relevant to the original GSO algorithm.

4.1. Notations

$I = \{1, 2, \ldots, n\}$ is the index set used to identify the $n$ agents (glowworms).

$x = (x_1, \ldots, x_n)$ is the state vector containing the states (positions) of all group members.

$d_{ij} = \|x_i - x_j\|$ is the euclidean distance between locations of agents $i$ and $j$.

$r_s$ is the radial sensor range of each agent.

$\ell_i$ is the luciferin-value associated with agent $i$.

$N_i = \{j : j \neq i, j \in I, d_{ij} \leq r_s\}$ is the set of distance-based neighbors of agent $i$.

$H_i = \{j : j \in I \text{ and } \ell_j < \ell_i\}$ is the set of luciferin-based neighbors of agent $i$.

$L_i = N_i \cap H_i$ is the leader set of agent $i$.

$t_k$ is the $k$th discrete time instant.

$G(V, E)$ is the directed graph with the set of nodes (glowworms) $V = \{v_1, v_2, \ldots, v_n\}$ and the edge set $E \subseteq V \times V$ (an edge $(i, j)$ exists iff $j \in L_i$).

$L = \{i : i \in I \text{ and } L_i(t_k) = \emptyset, \text{ for all } t_k\}$ is the Leader set of $G$.

$F = G - L$ is the follower-set of $G$.

$T = \{0, 1, \ldots, k, k + 1, \ldots\}$ is the discrete time-index set.

In the following, the terms glowworm, node, and agent are used interchangeably.

4.2. Assumptions

**Assumption 1**: Each agent is represented by a point.

**Assumption 2**: Agents move instantaneously with a step size of $s$ at each iteration.

**Assumption 3**: Collisions between the agents are ignored.

**Assumption 4**: The value of $\ell_j$ remains constant, for all $j \in I$.

**Assumption 5**: The range of local decision domain of each agent is kept constant and made equal to the maximum sensor range, i.e., $r^t_d = r_s$.

Note: Assumption 4 is related to Modification 1 and Assumption 5 is related to Modification 2 discussed earlier.

4.3. GSO model for analysis

The discrete-time model of the agent dynamics given in (3) is restated below:

$$x_i(t_{k+1}) = x_i(t_k) + s \left( \frac{x_j(t_k) - x_i(t_k)}{\|x_j(t_k) - x_i(t_k)\|} \right)$$

where, $j \in L_i, i$ selects to move toward $j$ with a probability $p_j$ given by (2) and

$$s = \begin{cases} \delta, & \text{if } d_{ij}(t_k) \geq \delta \\ d_{ij}(t_k), & \text{otherwise} \end{cases}$$

where, $0 < \delta < r_s$.

The above definition of step size $s$ incorporates Modification 3 discussed earlier.

We provide some definitions and lemmas before presenting the main results of the paper.

**Definition 1**: An agent $i$ is said to be stationary at time $t_k$, if $x_i(t_{k+1}) = x_i(t_k)$.

**Definition 2**: An agent $i$ is said to be stationary for all time starting from $t_j$, if $x_i(t_{k+1}) = x_i(t_k)$, for all $k \geq j$.

**Definition 3**: $G(V, E)$ is said to be stationary at time $t_k$, if all the agents are stationary at time $t_k$, i.e., $x_i(t_{k+1}) = x_i(t_k)$, for all $i \in I$.

**Definition 4**: $G(V, E)$ is said to be stationary for all time after $t_j$, if all the agents are stationary for all time after $t_j$, i.e., $x_i(t_{k+1}) = x_i(t_k)$, for all $k \geq j$ and for all $i \in I$.

**Definition 5**: Two nodes $i$ and $j$ are co-located at time $t_k$ if $x_i(t_k) = x_j(t_k)$ or $d_{ij}(t_k) = 0$.

**Definition 6**: A leader $l_i \in L$ is said to be isolated, if $\|l_i - l_j\| > 3r_s$ for all $j \in L, j \neq i$. 
At any time $t$.

Lemma 4. Suppose that $0 < s_1 < \delta$ and $s_2 \leq \delta$. Using the triangle inequality, we have

$$d_{ij}(t_{k+1}) \leq (d_{ij}(t_k) - s_1) + s_2 = (d_{ij}(t_k)) - (s_1 - s_2) \leq d_{ij}(t_k) \text{(since } s_1 - s_2 \geq 0) \leq r_s.$$  \hfill (13)

Case 2: Suppose that $0 < s_1 < \delta$ and $s_2 = \delta$. This implies, $i$ reaches $j$’s position at $t_k$ in one step, i.e., $x_i(t_{k+1}) = x_j(t_k)$ while $j$ moves a step distance of $\delta$ units. Therefore, $d_{ij}(t_{k+1}) = \delta$. From Assumption 2 and (13), we get $L_i(t_{k+1}) \neq \emptyset$. Similarly, we can show that, if $L_i(t_q) \neq \emptyset$ for some $q > k$, then $L_i(t_{q+1}) \neq \emptyset$. Thus, we get the desired result by induction. \hfill \Box

Lemma 4 implies that once an agent acquires at least one leader, it continues to have one.

Lemma 5. Suppose that $G$ becomes stationary at some time $t_k$. If nodes $i$ and $j$ are not co-located and $\ell_i \neq \ell_j$, then $\|x_i(t_k) - x_j(t_k)\| > r_s$.

Proof. Since values of $\ell$ are distinct, assume that $\ell_i < \ell_j \Rightarrow j \in H_l(t_k)$. Now, $x_i(t_{k+1}) = x_j(t_k)$ (since $G$ becomes stationary at time $t_k$) and $d_{ij}(t_k) \neq 0 \Rightarrow j \notin L_i(t_k)$ (otherwise $i$ can move toward $j$). This is true only if $j \notin N_i(t_k)$ (since $L_i = N_i \cap H_l$ and $j \in H_l$). \hfill \Box

Lemma 6. If for a node $i$, $L_i(t_q) \neq \emptyset$ at any time $t_q$, then when node $i$ becomes stationary at $t_k$, there exists at least one $j$ co-located with $i$ such that $\ell_i > \ell_j$.

Proof. $L_i(t_q) \neq \emptyset \Rightarrow L_i(t_k) \neq \emptyset$ (from Lemma 4). Suppose that $d_{ij}(t_k) \neq 0$, for all $j \in L_i(t_k)$. From (11), $x_i(t_{k+1}) = x_i(t_k)$, which is a contradiction. Therefore, there exists at least one $j$ co-located with $i$ when it becomes stationary such that $\ell_i < \ell_j$. Hence the result. \hfill \Box

Lemma 7. At any time $t_k$, let $L(t_k) = \{i : L_i(t_k) = \emptyset\}$, then $L \subseteq \hat{L}(t_k)$.

Proof. $L$ is the leader set of $G$, i.e., if $i \in L$ then $L_i(t_k) = \emptyset$, for all $t_k \Rightarrow i \in \hat{L}(t_k)$, for all $i \in L \Rightarrow L \subseteq \hat{L}(t_k)$. \hfill \Box

Lemma 8. Let $\hat{L}$ be defined as in Lemma 7. Then, $\hat{L}(t_{k+1}) \subseteq L(t_k)$.

Proof. For $i \in I$ and $i \notin L(t_k)$, $L_i(t_k) \neq \emptyset \Rightarrow L_i(t_{k+1}) \neq \emptyset$ (from Lemma 4) $\Rightarrow i \notin \hat{L}(t_{k+1})$. This implies that $i \not\in \hat{L}(t_{k+1}) \Rightarrow i \in \hat{L}(t_k)$. Hence we get the result. \hfill \Box

Lemma 8 implies that a non-member of $\hat{L}(t_k)$ cannot become a member of $\hat{L}(t_{k+1})$. However, a member $i \in \hat{L}(t_k)$ may lose its membership in $\hat{L}(t_{k+1})$. For instance, this happens when an agent, say $j$, with $\ell_i < \ell_j$ enters within the $r_s$ range of $i$ at $t_{k+1}$.

Lemma 9. There exists a time $t_q$ so that $\hat{L}(t_k) = \hat{L}(t_{k+1})$, for all $k \geq q$. 

\begin{figure}[h]
\centering
\includegraphics[width=0.8\textwidth]{movement.png}
\caption{Movements of agents $i$ and $j$ at iteration $t_k$.}
\end{figure}

\textbf{Lemma 1.} At least one agent remains stationary always.

\textbf{Proof.} From Assumption 4, there exists $i^* \in I$ such that $\ell_{i^*}(t_k) \geq \ell_i(t_k)$, for all $i \in I$ and $i \neq i^*$ and for all $t_k$. Therefore, $H_{i^*}(t_k) = \emptyset$, for all $t_k$. This implies that $L_{i^*}(t_k) = N_{i^*}(t_k) \cap H_{i^*}(t_k) = \emptyset$, for all $t_k$. This gives $x_{i^*}(t_k) = 0$, for all $t_k$. From (11), $x_{i^*}(t_{k+1}) = x_{i^*}(t_k)$, for all $t_k$. Therefore, the agent $i^*$ remains stationary for all time $t_k$. \hfill \Box

Note that Assumption 5 is not required to prove the above result.

\textbf{Lemma 2.} An agent with an empty leader set at some time $t_k$ is stationary at $t_k$.

\textbf{Proof.} The proof is straightforward and hence is omitted. \hfill \Box

\textbf{Lemma 3.} If $G$ is stationary at time $t_q$, then it is stationary for all $t_k$, $k > q$.

\textbf{Proof.} We prove the result using mathematical induction. Let $G$ be stationary at time $t_q$. From Definition 1, $x_i(t_{q+1}) = x_i(t_q)$ for all $i \in I$. Therefore, the result is true for $k = q + 1$. Now, let $x_i(t_{q+1}) = x_i(t_q)$, for all $i \in I$ and for some $k > q$. From (11) and (12), this is true, if and only if for all $i$, either

(i) $L_i(t_k) = \emptyset$ or
(ii) $L_i(t_k) \neq \emptyset$ and $d_{ij}(t_k) = 0$, for all $j \in L_i(t_k)$.

Suppose that (i) is true. Since there are no agent movements at $t_k$, the leader sets continue to be empty at $t_{k+1}$, i.e., $L_i(t_{k+1}) = \emptyset$, for all $i \in I$. From (11), we get $x_i(t_{k+2}) = x_i(t_{k+1})$, for all $i \in I$.

Suppose that (ii) is true. Since there are no agent movements,

$L_i(t_{k+1}) = L_i(t_k) \neq \emptyset \Rightarrow d_{ij}(t_{k+1}) = d_{ij}(t_k) = 0$, for all $j \in L_i(t_{k+1})$. Using (11) and (12) again, we get $x_i(t_{k+2}) = x_i(t_{k+1})$. We proved that if $G$ is stationary at time $t_k$, $k > q$, then it is stationary at $t_{k+1}$. Thus, the result follows by induction. \hfill \Box

\textbf{Lemma 4.} For any agent $i$, if $L_i(t_k) \neq \emptyset$ at some time $t_k$, then $L_i(t_k) \neq \emptyset$, for all $q \geq k$.

\textbf{Proof.} Given $L_i(t_k) \neq \emptyset$. Let $j \in L_i(t_k)$ such that $i$ selects to move toward it. This implies that $d_{ij}(t_k) \leq r_s$. Agent $i$ moves a distance $s_1$ units on the line joining $x_i(t_k)$ and $x_j(t_k)$ during the same time, let $j$ move $s_2$ units in a direction that makes an angle $\theta$ with the vector $x_j(t_k) - x_i(t_k)$ (see Fig. 9).
Theorem 3. If a leader $l$ is isolated (Definition 6), then all the members of $N_l(t_0)$ are co-located with $l$, for all $t_k, k \geq K(q)$, where $q = |F|$, $K(i) = \sum_{j=1}^{i} \left[ \frac{d_{f_j}(t_{K(j-1)})}{\delta} \right]$, $K(0) = 0, f_j \in F, j = 1, \ldots, q$, and $\ell_{f_j} \geq \ell_{f_2} \geq \cdots \geq \ell_{f_q}$.

Proof. Consider an isolated leader $l$. Since $\ell_{f_1} > \ell_{f_j}$ for all $f_j \in N_l(t_0)$, all followers are influenced by the leader. Since the values of $\ell$ are constant, we can sort the followers as $f_1, f_2, \ldots, f_q$ according to ascending order of their associated $\ell$ values. Therefore, we have $\ell_{f_1} > \ell_{f_2} \geq \cdots \geq \ell_{f_q}$.

Consider the movements of $f_1$. Since $L_{f_1}(t_0) = \{l\}$, for all $t_k, f_1$ makes deterministic movements toward $l$ and reaches it in $t_K(1)$ iterations, where $K(1) = \left[ \frac{d_{f_1}(t_0)}{\delta} \right]$. Therefore, for all $t_k, k \geq K(1), f_1$ is co-located with $l$.

Consider the movements of $f_2$. Since either $L_{f_2}(t_0) = \{l\}$ or $L_{f_2}(t_0) = \{l, f_1\}$, for all $t_k$, at each iteration it moves either toward $l$ or $f_1$. Based on its distance to $l$ at $t_0$ and its moves, 

it may or may not reach $l$ within $t_K(1)$ iterations. However, at $t_K(1)$ (assume that $f_2$ is not co-located with $l$ at that time), since $l$ and $f_1$ are co-located, $f_2$ has only one direction to move. Let $d_{f_2}(t_{K(1)})$ be the distance between $f_2$ and $l$ at $t_{K(1)}$. Then $f_2$ converges to $l$ in $\left[ \frac{d_{f_2}(t_{K(1)})}{\delta} \right]$ steps after $t_{K(1)}$. Therefore, for all $t_k, k \geq K(2), f_2$ is co-located with $l$, where $K(2) = K(1) + \left[ \frac{d_{f_2}(t_{K(1)})}{\delta} \right]$.

Note that this is a conservative bound as $f_2$ could get co-located with $l$ much before $t_{K(2)}$.

Now, suppose that $f_3$ is co-located with $l$, for all $t_k, k \geq K(i)$. Let $d_{f_{i+1}}(t_{K(i)})$ be the distance between $f_{i+1}$ and $l$ at time $t_{K(i)}$. Now $f_{i+1}$ takes $\left[ \frac{d_{f_{i+1}}(t_{K(i)})}{\delta} \right]$ steps to reach $l$. This implies $K(i+1) = K(i) + \left[ \frac{d_{f_{i+1}}(t_{K(i)})}{\delta} \right]$.

Therefore, by induction, we can show that all the members of $N_l(t_0)$ will be co-located with $l$, for all $t_k, k \geq K(q)$.

Theorem 4. Let $K$ be defined as in Theorem 3 and $q = |F|$. Suppose that two leaders $l_1$ and $l_2$ are not isolated but their neighborhoods are non-overlapping, i.e., $2r_s < \|l_1 - l_2\| \leq 3r_s$, then all the followers get co-located with either one of the leaders, for all $t_k, k \geq K(q)$.

Proof. Consider the followers $f_1, f_2, \ldots, f_q$ such that $\ell_{f_1} > \ell_{f_2} \geq \cdots \geq \ell_{f_q}$.
Consider the movements of $f_1$. Obviously, $f_1 \in N_l \cup N_k$. Otherwise, there will be no attracting influence on $f_1$ and hence it remains stationary, which then contradicts the characteristics of a follower. Moreover, $f_1 \notin N_l$ because the relationship $\ell_{f_1} > \ell_{l_1}$ contradicts the fact that a leader must have maximum $\ell$-value in its neighborhood. Therefore, $f_1 \in N_l(t_k)$, for all $k$. This implies that $f_1$ gets co-located with $l_1$, for all $t_k, k \geq K(1)$.

Consider the movements of $f_2$. Clearly, $f_2 \notin N_l$. Also, $f_2 \notin N_l \cup N_k$ is possible for $t_k$ such that $k \leq K(1)$, however, during this time $L_{f_2} = \{f_1\}$. Otherwise, $f_2$ becomes a leader which is a contradiction. But once $f_1$ is co-located with $l_1$, $f_2$ should be within $r_i$-distance of $l_1$. Therefore, for all $t_k, k \geq K(2)$, $f_2$ is co-located with $l_1$. Similar analysis can be carried out till follower $f_g$ and we can show that $f_1, f_2, \ldots, f_g$ will be co-located with $l_1$ for all $t_k, k \geq K(g)$.

Consider the movements of $f_{g+1}$. If $f_{g+1} \in N_l(t_0)$, it remains in $N_l$ and gets co-located for all $t_k, k \geq K(g+1)$. If $f_{g+1} \in N_l(t_0)$, there is a possibility that it leaves $N_l$ and enters $N_l$ before $f_1, \ldots, f_g$ reach $l_1$. However, once it enters $N_l$, it cannot leave $N_l$. After $t_{K(g)}$, $f_{g+1} \in N_l(t_k \cup N_l)$. Therefore, if $f_{g+1} \in N_l(t_0)$ (or $f_{g+1} \in N_l$), it gets co-located with $l_1$ (or $l_2$), for all $t_k, k \geq K(g+1)$. Note that while evaluating the expression $[d_{ij}(t_{K(g)})]_\theta$, $l_1 = l_1$ (or $l_2$) if $f_j \in N_l(t_{K(g)})$ (or $f_j \in N_l(t_{K(g)}))$. In general, if $f_i \in \{f_{g+1}, \ldots, f_g\}$ reaches either one of the leaders in time $t_{K(i)}$, then $f_{g+1}$ reaches either one of them in time $t_{K(i+1)}$. Therefore, all the followers $f_1, \ldots, f_g$ are co-located at $l_1$, for all $t_k, k \geq K(g)$ and the rest of the followers $f_{g+1}, \ldots, f_q$ are co-located at either one of the leaders for all $t_k, k \geq K(q)$. □

Theorem 5. Let $K$ be defined as in Theorem 3 and $q = |F|$. If two leaders $l_1$ and $l_2$ have overlapping neighborhoods, i.e., $r_1 \leq ||l_1-l_2|| \leq 2r_1$, then all followers $f_j$ such that $j \in F$ and $l_1 < \ell_{f_j} < \ell_{l_1}$ converge to one of the leaders in finite time and remaining followers located in the overlap region of neighborhoods asymptotically converge to one of the leaders.

Proof. Let the followers $f_1, f_2, \ldots, f_q$ satisfy $\ell_{f_1} > \ell_{f_2} \geq \cdots \geq \ell_{f_q} > \ell_{l_1} > \ell_{l_2}$. Consider the movements of the members of $F_1 = \{f_1, \ldots, f_q\}$. The set $F_1$ has the property that $f_i \in N_l \setminus (N_l \cap N_k)$, for all $f_i \in F_1$. Using Theorem 3, we can show that all members of $F_1$ are co-located with $l_1$, for all $t_k, k \geq K(g)$.

Consider the movements of $f_{g+1}$ after $t_{K(i)}$. If $f_{g+1} \in N_l(t_{K(i)}) \cap (N_l \cap N_k)$, then $f_{g+1}$ reaches $l_1$ in time $t_{K(g+1)}$, with $l_1 = l_1$. If $f_{g+1} \notin N_l(t_{K(i)}) \cap (N_l \cap N_k)$, then $f_{g+1}$ reaches $l_2$ in time $t_{K(g+1)}$, with $l_1 = l_2$.

Suppose that $f_{g+1} \in N_l \cap N_k$.

Let $\Gamma = \{x : x \in (ax_a + (1-a)x_b), 0 \leq a \leq 1\}$, where $x_a$ and $x_b$ are points of intersection of the line segment $l_1l_2$ with circles centered at $x_1$ and $x_2$, respectively. Based on the position of $f_{g+1}$ with respect to the line $\Gamma$, we obtain two cases (Fig. 11).

Case 1: $f_{g+1} \in \Gamma$.

Let $p_1$ and $p_2$ ($p_1 + p_2 = 1$) be the probabilities that $f_{g+1}$ makes a step movement toward $l_1$ and $l_2$, respectively, at each time $t_k$. Let $d_{l_1l_2}$ be the distance between the two leaders’ positions. Since the agent moves in discrete steps, the convergence problem can be formulated as a finite state Markov chain with $n = \lceil 2r_1 - d_{l_1l_2} \rceil$, where the agent’s position at any iteration $t_k$ is represented by one of the states. For instance, in Fig. 12, $x_a$ coincides with State 1 and $x_b$ coincides with State $n$. Note that we need not consider the movements of the follower on the line segments $l_1x_a$ and $l_2x_b$ because, once the agent reaches $x_a$ ($x_b$), it will be influenced by $l_1$ ($l_2$) only, reaching it in finite number of steps. The one-step probabilities can be expressed in the form of a transition probability matrix given by:

$$
P = [p_{ij}]_{n \times n} = \begin{bmatrix}
p_{11} & 0 & 0 & 0 
p_1 & p_2 & 0 & 0 
0 & p_1 & p_2 & 0 
0 & 0 & 0 & p_1 
0 & 0 & 0 & p_2 
0 & 0 & 0 & 1
\end{bmatrix}.
$$

(15)

Let $P^n_{ij} = [p^n_{ij}]$ be the $n$-state transition matrix where $p^n_{ij}$ represents the probability that an agent, initially in state $i$, ends up in state $j$ at the $n$th step. The steady state probability state vector equation is then given by:

$$
P^\infty P = P^\infty
$$

(16)

where $P^\infty = \lim_{n \to \infty} [p^n_{ij}]$. Let

$$
P^\infty = \begin{bmatrix}
1 & 0 & 0 & 0 & 0 
p_{21} & p_{22} & 0 & 0 & 0 
p_{21} & p_{22} & 0 & 0 & 0 
p_{21} & p_{22} & 0 & 0 & 0 
p_{21} & p_{22} & 0 & 0 & 0
\end{bmatrix}.
$$

(17)

Consider $P^{\infty} P = P^{\infty}$, where $P^{\infty}$ is the $i$th row of $P^{\infty}$. Then,

$$p_{11}p_{12} + p_{11} = p_{11};
p_{11}p_{12} = p_{12};
p_{11}p_{14} + p_{22}p_{22} = p_{13}$$
\[ P_1 P_{i(n-1)} + P_2 P_{i(n-3)} = P_{i(n-2)} \]
\[ P_2 P_{i(n-2)} = P_{i(n-1)} \]
\[ P_{i(n)} + P_2 P_{i(n-1)} = P_{i(n)} \tag{18} \]

From (18), we get
\[ p_{i2} = 0; \quad p_{i3} = 0 \]
\[ p_{1j} + p_{2j} = p_{i(j+1)}, \quad j = 2, 3, \ldots, (n - 3) \]
\[ p_{1} + p_{2} = p_{i(n-1)}; \quad p_{i(n)} + p_{2} = p_{i(n-1)} \tag{19} \]

From the set of equations in (19), we get the following relations:
\[ p_{ij} = 0 \quad \text{for} \quad j = 2, 3, \ldots, (n - 1) \]
\[ \sum_{j=1}^{n} p_{ij} = 1, \quad \text{for all} \quad i = 1, 2, \ldots, n \]
\[ \Rightarrow p_{i1} + p_{i2} = 1, \quad \text{for all} \quad i = 1, 2, \ldots, n. \tag{20} \]

However, (20) does not give the complete solution to (18). The result in (20) shows that the agent starting in any one of the \( n \) states, converges to either \( x_a \) or \( x_b \) asymptotically with probability 1. Once \( f_{g+1} \) reaches \( x_a (x_b) \), it reaches \( l_1 (l_2) \) in \([d_{l_1 l_2} - \epsilon]\) time steps.

As an illustration, the explicit probabilities of reaching either \( l_1 \) or \( l_2 \) from any intermediate state when \( n = 5 \) is given below:
\[
P_S^\infty = \begin{bmatrix}
1 & 0 & 0 & 0 & 0 \\
\frac{p_1 (1 - p_2)}{1 - 2p_1 p_2} & 0 & 0 & 0 & \frac{p_2^3}{1 - 2p_1 p_2} \\
\frac{p_2}{1 - 2p_1 p_2} & 0 & 0 & 0 & \frac{p_2^3}{1 - 2p_1 p_2} \\
\frac{p_2}{1 - 2p_1 p_2} & 0 & 0 & 0 & \frac{p_2^3}{1 - 2p_1 p_2} \\
0 & 0 & 0 & 0 & 1
\end{bmatrix}. \tag{21}
\]

**Case 2:** \( f_{g+1} \in N_{l_1} \cap N_{l_2} \setminus \Gamma \).

At each iteration, \( f_{g+1} \) moves a distance \( \delta \) toward either \( l_1 \) or \( l_2 \). Let \( d_{l_1 l_2} \) be the distance between \( l_1 \) and \( l_2 \). Let \( h(t) \) and \( h(t + 1) \) be the perpendicular distances from \( x_{f_{g+1}}(t) \) and \( x_{f_{g+1}}(t + 1) \), respectively, to \( x_{l_1}, x_{l_2} \). Clearly, \( h(t + 1) < h(t) \). Therefore, \( h(t) \) is a monotonically decreasing function. Now, we show that there exists a time \( t_q \) such that
\[ d_{l_1 l_2} < d_{f_{g+1}l_1}(t_k) + d_{f_{g+1}l_2}(t_k) < d_{l_1 l_2} + \delta, \quad \text{for all} \quad t_k, k \geq q. \tag{22} \]

Referring to Fig. 13, the angles \( \theta_1 \) and \( \theta_2 \) are given by:
\[
\theta_1 = \arccos \left( \frac{d_{f_{g+1}l_1}^2 + d_{f_{g+1}l_2}^2 - d_{l_1 l_2}^2}{2d_{f_{g+1}l_1}d_{f_{g+1}l_2}} \right) \tag{23}
\]
\[
\theta_2 = \arccos \left( \frac{d_{f_{g+1}l_2}^2 + d_{f_{g+1}l_1}^2 - d_{l_1 l_2}^2}{2d_{f_{g+1}l_2}d_{f_{g+1}l_1}} \right). \tag{24}
\]

A lower bound on the perpendicular distance to the line joining \( l_1 \) and \( l_2 \) such that \( d_{f_{g+1}l_1}(t_k) + d_{f_{g+1}l_2}(t_k) = d_{l_1 l_2} + \delta \) can be calculated as:
\[
\epsilon = \frac{\delta (\delta + 2d_{l_1 l_2})}{2(d_{l_1 l_2} + \delta)}. \tag{25}
\]

Using (25), the minimum perpendicular distance \( d_p \) moved by \( f_{g+1} \) at each step is given by:
\[
d_p = \delta \cos \phi
\]
where,
\[
\phi = \frac{\pi}{2} - \arctan \left( \frac{\epsilon \tan \theta}{d \tan \theta - \epsilon} \right) \tag{26}
\]
\[
\theta = \min(\theta_1, \theta_2).
\]

From Fig. 13, the initial perpendicular distance \( h(t_0) \) is given by:
\[
h(t_0) = \frac{d_{l_1 l_2} \tan \theta_1 \tan \theta_2}{\tan \theta_1 + \tan \theta_2}. \tag{27}
\]

Therefore, after \( t_q, q = \left[ \frac{h(t_0) - \epsilon}{d_p} \right] \), the perpendicular distance to the line segment \( l_1 l_2 \) is within \( \epsilon \) and hence (22) is satisfied. The consequence of (22) is that
\[
\left[ d_{f_{g+1}l_1}(t_k) + d_{f_{g+1}l_2}(t_k) \right] < d_{l_1 l_2}, \quad \text{for all} \quad k \geq q.
\]

Hence, a similar analysis as in Case 1 can be used to prove asymptotic convergence to one of the leaders. Moreover, since the movement of \( f_{g+1} \) is always on the line joining its own position and that of \( l_1 \) or \( l_2 \), it cannot converge to a position between the leader positions and gets co-located with either \( l_1 \) or \( l_2 \). A similar analysis can be done to show asymptotic convergence of the remaining followers \( f_{g+2}, \ldots, f_q \) to either one of the leaders. \( \square \)

**Remarks.** We can infer from the above theoretical results that, even though agent movements are probabilistic in nature, agents get co-located with leaders in finite time if the leader neighborhoods are non-overlapping. Otherwise, the agents converge to leaders asymptotically.

### 4.5. Results for the variable luciferin case

In all the previous theorems, we assumed that the agents have constant luciferin. However, the luciferin values of agents in the actual GSO algorithm vary with time and their position in the workspace. Therefore, we relax the constant luciferin assumption to some extent by varying the luciferin values as a function of the agent’s position from a leader. This reflects the case where the first term in the luciferin update rule, given in (1), is zero (or \( \rho = 1 \)).
Consider a single leader \( l \) such that the luciferin value of a follower is equal to that of the leader when it is co-located with the leader and decreases monotonically with distance from the leader, then there exists a sequence \( \{f_1, f_2, \ldots, f_q\} \) in which the followers reach the leader and all the members of \( N_l(t_0) \) are co-located with \( l \), for all \( t_k, k \geq K(q) \), where, \( q = |F| \). \( K(i) = \sum_{j=1}^{i} \left[ \frac{d_{f_j}(t(K(j-1)))}{\delta} \right] \). \( K(0) = 0 \), and \( f_j \in F, j = 1, \ldots, q \).

**Proof.** Consider a single leader \( l \). Since \( \ell_l > \ell_{f_j} \) for all \( f_j \in N_l(t_0) \), all followers are influenced by the leader. Since the value of \( \ell_{f_j}(t) \) is a function of distance between the leader \( l \) and the follower \( f_j \) at time \( t \), it changes with time as \( f_j \) moves toward the leader. Consider three followers \( f_1, f_2, \) and \( f_k \) such that \( d_{f_1}(t) < d_{f_2}(t) < d_{f_k}(t) \) (Fig. 14).

\[
\Rightarrow \ell_1 > \ell_{f_2} > \ell_{f_k} > \ell_{f_k} (t) \tag{28}
\]

Since, \( L_{f_1}(t) = \{l\} \), \( f_1 \) moves toward \( l \) by a distance \( \delta \). Suppose that \( f_1 \) decides to move toward \( f_1 \) and \( f_k \) decides to move toward \( l \) at time \( t \), respectively. This gives

\[
d_{f_1}(t+1) < d_{f_1}(t+1) < d_{f_k}(t+1) \tag{29}
\]

\[
\Rightarrow \ell_{f_1}(t+1) > \ell_{f_k}(t+1) > \ell_{f_k}(t+1) \tag{30}
\]

From (28) and (30), note that \( \ell_{f_1}(t) > \ell_{f_2}(t) \), but \( \ell_{f_1}(t+1) < \ell_{f_k}(t+1) \). Therefore, we cannot sort the followers according to ascending order of their associated \( \ell \) values as it was possible in Theorem 3. However,

\[
d_{f_1}(t) = d_{f_1}(0) - \delta t \leq d_{f_1}(0) - \delta t \leq d_{f_1}(t) \tag{31}
\]

Therefore, \( f_1 \) always remains closer than \( f_j \) to the leader. Now at time \( t_0 \), let

\[
f_1 = \arg \left\{ \min_{f \in F} \{d_{f_1}(t_0)\} \right\} \tag{32}
\]

From (32), \( \ell_{f_1}(t_k) \geq \ell_{f_j}(t_k) \), for all \( f_j \in F, i \neq 1 \) and for all \( t_k \geq t_0 \). Therefore, \( f_1 \) makes deterministic movements toward \( l \) and reaches it in \( t_{K(1)} \) iterations, where \( K(1) = \left[ \frac{d_{f_1}(t_0)}{\delta} \right] \).

Therefore, for all \( t_k, k \geq K(1) \), \( f_1 \) is co-located with \( l \). Next, identify follower \( f_2 \) such that

\[
f_2 = \arg \left\{ \min_{f \in F, \neq i} \{d_{f_1}(t(K(i)))\} \right\} \tag{33}
\]

Using analysis that is similar to that of the movements of follower \( f_2 \) in Theorem 3, we can show that for all \( t_k, k \geq K(2) \), \( f_2 \) is co-located with \( l \), where

\[
K(2) = K(1) + \left[ \frac{d_{f_1}(t(K(1)))}{\delta} \right] \tag{34}
\]

Now, suppose that \( f_j \) is co-located with \( l \), for all \( t_k, k \geq K(j) \). Identify follower \( f_{j+1} \) such that

\[
f_{j+1} = \arg \left\{ \min_{f \in F, \neq k} \{d_{f_1}(t(K(k))\} \right\} \tag{35}
\]

Let \( d_{f_{j+1}}(t(K(k))) \) be the distance between \( f_{j+1} \) and \( l \) at time \( t_{K(k)} \). Now, \( f_{j+1} \) takes \( \left[ d_{f_{j+1}}(t(K(k))) \right] \) steps to reach \( l \). This implies,

\[
K(j + 1) = K(j) + \left[ \frac{d_{f_{j+1}}(t(K(k)))}{\delta} \right] = \sum_{i=1}^{j+1} \left[ \frac{d_{f_i}(t(K(k)))}{\delta} \right] \tag{36}
\]

Therefore, by induction, we can show that there exists a sequence \( \{f_1, f_2, \ldots, f_q\} \) in which the followers reach the leader and all the members of \( N_l(t_0) \) will be co-located with \( l \), for all \( t_k, k \geq K(q) \). □

Next, we consider two leaders \( l_1 \) and \( l_2 \) and a set of followers \( F \) where the luciferin value of each follower is assumed to be a function of its distance from the two leaders. We restrict the analysis to a one-dimensional case. In particular, we use the following function to model the luciferin values of the followers:

\[
\ell(x) = a_1 e^{-b_1|x-x_1|} + a_2 e^{-b_2|x-x_2|} \tag{37}
\]

where, \( a_1, b_1, a_2, \) and \( b_2 \) are positive constants.

**Theorem 7.** Consider two leaders \( l_1 \) and \( l_2 \) located at positions \( x_1 \) and \( x_2 \), respectively, on a real axis such that \( 0 \leq x_1 \leq x_2 - r_s \), and a set of followers \( F \) such that \( x_1 < x_f < x_2 \) for all \( f_i \in F \). If the luciferin value of each follower varies according to (36), then the line segment \( x_1 x_2 \) can be partitioned into three regions \( R_a = [x_1, x_i], R_b = [x_1, x_i + r_s], \) and \( R_c = (x_1 + r_s, x_2) \) where,

\[
x_i = \frac{b_1 x_1 + b_2 x_2 - \ln \left[ \frac{a_2 (e^{b_2 x_2} - 1)}{a_1 (1 - e^{b_1 r_s})} \right]}{b_1 + b_2} \tag{38}
\]

and all the followers in regions \( R_a \) and \( R_c \) converge to the leaders \( l_1 \) and \( l_2 \), respectively, in finite time and all the followers in region \( R_b \) asymptotically converge to one of the leaders.

**Proof.** Consider two leaders \( l_1 \) and \( l_2 \) located at \( x_1 \) and \( x_2 \), respectively, on the real axis such that \( 0 \leq x_1 \leq x_2 - r_s \). Assume that the luciferin value of each follower is given by (36) and given that each follower \( f_i \) lies on the line segment \( x_1 x_2 \), we can rewrite (36) that holds for the region \( [x_1, x_2] \) as below:

\[
\ell(x) = a_1 e^{-b_1(x-x_1)} + a_2 e^{-b_2(x-x_2)} \tag{39}
\]
where, \( x_1 \leq x \leq x_2 \). An example plot of the above function is shown in Fig. 15. The values chosen for the various constants are \( a_1 = 1, b_1 = 1, a_2 = 0.8, b_2 = 0.4, x_1 = 1, \) and \( x_2 = 9 \).

![Plot of \( \ell(x) \) for \( x \in [x_1, x_2] \) when \( a_1 = 1, b_1 = 1, a_2 = 0.8, b_2 = 0.4, x_1 = 1, \) and \( x_2 = 9 \).](image)

Applying natural logarithm on both sides of the above equation,

\[
\frac{d\ell(x)}{dx} \bigg|_{x=x^*} = -a_1 b_1 e^{-b_1(x^*-x_1)} + a_2 b_2 e^{-b_2(x_2-x^*)} = 0
\]

\[
e^{b_1(x^*-x_1)-b_2(x_2-x^*)} = \frac{a_1 b_1}{a_2 b_2}.
\]

Using the first-order sufficient conditions for a minimum,

\[
\frac{d\ell(x)}{dx} \bigg|_{x=x^*} = -a_1 b_1 e^{-b_1(x^*-x_1)} + a_2 b_2 e^{-b_2(x_2-x^*)} = 0
\]

\[
\Rightarrow x^* = \frac{b_1 x_1 + b_2 x_2 - \ln \left( \frac{a_1 b_1}{a_2 b_2} \right)}{b_1 + b_2}.
\]

Note that any line parallel to the \( x \)-axis intersects the luciferin profile at two points \((x_i, \ell(x_i))\) and \((x_j, \ell(x_j))\) such that \( \ell(x_i) = \ell(x_j) \).

Now, consider \( x_i \) and \( x_j + r_i \) such that

\[
\ell(x_i) = \ell(x_i + r_i).
\]

We solve for \( x_i \) in the following way. Using (36) and (40),

\[
a_1 e^{-b_1(x_i-x_1)} + a_2 e^{-b_2(x_2-x_i)} = a_1 e^{-b_1(x_i+r_i-x_1)} + a_2 e^{-b_2(x_2-x_i-r_i)}
\]

\[
= a_1 e^{-b_1(x_i-x_1)} e^{-b_1 r_i} + a_2 e^{-b_2(x_2-x_i)} e^{b_2 r_i}
\]

\[
\Rightarrow a_1 e^{-b_1(x_i-x_1)} (1 - e^{-b_1 r_i}) = a_2 e^{-b_2(x_2-x_i)} (e^{b_2 r_i} - 1).
\]

Taking exponential terms to one side and rewriting the above equation,

\[
e^{-b_1(x_i-x_1)+b_2(x_2-x_i)} = \frac{a_2 (e^{b_2 r_i} - 1)}{a_1 (1 - e^{-b_1 r_i})}.
\]

Applying natural logarithm on both sides of the above equation,

\[
-(b_1 + b_2) x_i + b_1 x_1 + b_2 x_2 = \ln \left( \frac{a_2 (e^{b_2 r_i} - 1)}{a_1 (1 - e^{-b_1 r_i})} \right)
\]

\[
\Rightarrow \ell(x_i) = \frac{b_1 x_1 + b_2 x_2 - \ln \left( \frac{a_1 b_1}{a_2 b_2} \right)}{b_1 + b_2}.
\]

From (43) and (44), \( f_b \) does not influence the movement of \( f_a \). Moreover, since \( d_{f_a f_a}(t) > r_s \), for all \( f_c \in R_c, f_a \) cannot sense the presence of, and its movements are not influenced by, \( f_c \). These two conditions, prevent \( f_a \) to move toward the leader \( l_2 \). However, it is provided with an uphill direction toward the leader \( l_1 \). Using the analysis from Theorem 6, we can prove that \( f_a \) reaches \( l_1 \) in finite number of time steps. We can use similar analysis for the follower \( f_c \) to prove that it converges to the leader \( l_2 \) in finite number of time steps.

However, for all \( f_b \in R_b, f_b \) is influenced by followers in \( R_a \) and \( R_c \) as well. Hence, \( f_c \) has an uphill direction toward both the leaders until a time \( t_k \) when it reaches either \( x_i \) or \( x_i + r_s \). Using Theorem 5, we can show that \( f_b \) reaches one of the leaders asymptotically. \( \square \)

Note that though Theorem 7 was proved for a composite exponential function, it can be extended to include most monotonically decaying functions which on combining produce a convex profile between the two leaders.

5. Simulations on simplified GSO model

We conduct four experiments in order to verify the theoretical results arrived at in the previous section. In all the experiments, each agent is associated with a constant luciferin level and a constant local decision domain \((r_d^l = 1.5, \) for all \( i = 1, \ldots, n)\). The first three experiments are designed to represent the assumptions of isolated, non-isolated and non-overlapping, and overlapping leader neighborhoods used in the first three theorems, respectively. In the last experiment, we consider a more general placement of agents and attempt to identify leaders and followers based on whether they move or remain stationary for all time, and characterize the various leader neighborhoods based on the inter-leader distances.

5.1. Simulation experiment 1: Isolated leader

A set of 9 agents is randomly deployed in a circle of radius 1.5 units and centered at the leader’s location \((0, 0)\). The luciferin value of the leader is chosen as \( \ell_l = 10 \). The emergence of the agent movements is shown in Fig. 16(a). Agents are ranked according to their luciferin levels. For instance, Agent 1 has the highest luciferin value among all
Table 3
Luciferin levels of the agents, distance of agent $j$ when agent $j - 1$ reaches the leader $d_{ji}(t_{K(j-1)})$ at time $t_{K(j-1)}$, and the number of iterations taken by agent $j$ to reach the leader ($K(j)$)

<table>
<thead>
<tr>
<th>Agent</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\ell_j$</td>
<td>8.12</td>
<td>6.98</td>
<td>6.63</td>
<td>6.59</td>
<td>6.46</td>
<td>6.10</td>
<td>5.77</td>
<td>4.93</td>
<td>4.29</td>
</tr>
<tr>
<td>$d_{ji}(t_{K(j-1)})$</td>
<td>0.539</td>
<td>0.224</td>
<td>0.023</td>
<td>0</td>
<td>0.024</td>
<td>0.116</td>
<td>0.045</td>
<td>0.199</td>
<td>0.097</td>
</tr>
<tr>
<td>$t_{K(j)}$</td>
<td>18</td>
<td>26</td>
<td>27</td>
<td>27</td>
<td>28</td>
<td>32</td>
<td>35</td>
<td>41</td>
<td>45</td>
</tr>
</tbody>
</table>

![Fig. 16](image1.png)

Fig. 16. (a) Agent movements in an isolated leader’s neighborhood. (b) Emergence of agent movements in the case of non-isolated and non-overlapping leader neighborhoods.

the followers. Table 3 shows the luciferin levels of the agents, distance of agent $j$ when agent $j - 1$ reaches the leader $d_{ji}(t_{K(j-1)})$ at time $t_{K(j-1)}$, and the number of iterations taken by agent $j$ to reach the leader ($K(j)$). From Theorem 3, $K(9) = \sum_{j=1}^{9} \left[ \frac{d_{ji}(t_{K(j-1)})}{s_j} \right]$. Using the values from Table 3, we get

$$K(9) = \begin{bmatrix} 0.5385 \ 0.03 \\ 0.2237 \ 0.03 \\ 0.0234 \ 0.03 \\ 0.0241 \ 0.03 \end{bmatrix} + \begin{bmatrix} 0.1158 \ 0.03 \\ 0.0449 \ 0.03 \\ 0.1985 \ 0.03 \\ 0.0986 \ 0.03 \end{bmatrix} = 45.$$ 

Note that the above result, computed using Theorem 3, coincides with the simulation result given in Table 3.

5.2. Simulation experiment 2: Non-isolated and non-overlapping leader neighborhoods

Two leaders are located at $(−2, 0)$ and $(2, 0)$, with $\ell_1 = 100$ and $\ell_2 = 50$, respectively. Note that the above placement of leaders leads to non-isolated and non-overlapping leader neighborhoods in the sense defined in the previous section. A set of 98 agents is randomly deployed such that all agents located inside the circle of radius 1.5 units and centered at $(2, 0)$ have a luciferin level less than 50 and the remaining agents have a luciferin level less than 100. This ensures that $L_2$ remains stationary for all time since any agent with a luciferin level more than 50 is outside its sensing range of 1.5 units. The emergence of agent-movements is shown in Fig. 16(b). All agents with $\ell > 50$ reach the leader $L_1$ and most of the agents that are located inside the neighborhood of $L_2$ converge to $L_2$. However, note that a few agents leave the $L_2$-neighborhood, enter the $L_1$-neighborhood, and get co-located at $L_1$. This is in perfect agreement with the observation made in the proof of Theorem 4 that an agent can leave the neighborhood of $L_2$ and once it enters the $L_1$-neighborhood, it remains in the same neighborhood, and eventually converges to $L_1$.

5.3. Simulation experiment 3: Overlapping leader neighborhoods

In order to illustrate the case of overlapping neighborhoods, we consider a single agent that is simultaneously influenced by the presence of two leaders. For this purpose, we consider two leaders at locations $(−1, 0)$ and $(1, 0)$, respectively. The agent is located at $(0, 0.5)$, which is in the intersection of the leader neighborhoods of both the leaders. One of the realizations of agent movements when $p_1 = 0.45$ (probability of moving toward $L_1$), and hence $p_2 = 0.55$, is shown in Fig. 17(a). A similar plot for the case $p_1 = 0.6$ is shown in Fig. 17(b). Fig. 18 shows the plot of number of iterations taken to reach one of the leaders as a function of $p_1$, averaged over 100 experimental trials. We observe that when $p_1 < 0.5$, the agent converges always to $L_2$ and when $p_1 > 0.5$, it converges always to $L_1$. However, when $p_1 = 0.5$, it converged forty six times to $L_1$ and sixty four times to $L_2$. The average time taken to reach one of the leaders is minimum at $p_1 = 0$ ($\sqrt{\frac{0.5^2+1}{0.03}} = 38$), increases with increase in $p_1$, reaches a maximum at $p_1 = 0.5$, and then decreases reaching a minimum again at $p_1 = 1$. The result in Fig. 18 signifies the fact that the randomness and the asymptotic behavior of the agent are more prominent in the
vicinity of $p_1 = 0.5$. This is evident from the large variability and relatively more convergence time in this region. The agent movements become more deterministic as $p_1$ tends away from 0.5.

5.4. Simulation experiment 4: Random placement of agents

A set of 100 agents is randomly deployed in a workspace of $(-5, 5) \times (-5, 5)$. Each agent is randomly associated with an integer value of luciferin level between 1 and 100. The histogram of the luciferin levels of the agents is shown in Fig. 19(a). Fig. 19(b) shows how leaders are formed based on factors such as the placement of agents and their associated luciferin levels. In particular, five leaders $L_1, \ldots, L_5$ are formed. Each one of them never encounters an agent, within its sensing range, which has a luciferin level more than that of its own and hence remains stationary for all time. In Fig. 19(b), $L_1$ and $L_3$ form non-isolated and non-overlapping neighborhoods. Note that one of the agents escapes the neighborhood of $L_3$ and enters the $L_1$-neighborhood, a behavior of agent movements that is in agreement with the conclusions reached in Theorem 4.

Note that $(L_1, L_4)$, $(L_1, L_5)$, $(L_2, L_3)$, and $(L_2, L_5)$ form overlapping neighborhoods. A general observation in the overlapping neighborhood case is that agents located in the intersection region of any two neighborhoods eventually get

Fig. 17. Overlapping leader neighborhoods. Agent movements when (a) $p_1 = 0.45$. (b) $p_1 = 0.6$.

Fig. 19. (a) Histogram of luciferin levels of the agents. (b) Formation of leaders and followers based on the placement of agents and their associated luciferin levels. The numbers represent the luciferin levels associated with the leaders.

Fig. 18. Average time taken to reach one of the leaders as a function of $p_1$ over a set of 100 experimental trials. The ’*’ above and below the average for each value of $p_1$ represents the maximum and minimum number of iterations, respectively, taken to reach one of the leaders.
co-located at the leader with relatively more luciferin level than the other.

6. Concluding remarks

We present the theoretical foundations for the glowworm swarm optimization (GSO) algorithm that is used as a multi-agent system protocol for rendezvous of mobile agent swarms at multiple locations. A careful investigation of the group level phases of the GSO algorithm reveals that the theoretical analysis of the algorithm can be divided into two subproblems: (i) Splitting of agent-swarm into subgroups and (ii) Local convergence of agents in each subgroup to the source locations. Initially, local convergence results are proved for the case in which the luciferin level and decision domain range of the agents are constant. In particular, we define a leader to be an agent that is closer to a source location and consider the movements of other agents in the presence of multiple leaders. We show that the distance between the leaders gives rise to three leader neighborhood cases and present local convergence results with respect to each case. In particular, we find an upper bound on the time taken by the agents to converge to an isolated leader and on the time taken by the agents to converge to one of the leaders with non-isolated and non-overlapping neighborhoods. We show that agents under the influence of multiple leaders with overlapping neighborhoods asymptotically converge to one of the leaders. Later, the constant luciferin assumption is relaxed to some extent and similar results are proved. We illustrate the theoretical results of the paper using simulations.

Acknowledgements

We are thankful to all the referees for their valuable comments that proved very useful in revising the paper.

References


K.N. Krishnanand received his B.E. (Hons.) degree in Electrical and Electronics from Birla Institute of Technology and Science, Pilani, India and M.Sc. (Engg.) degree in Aerospace Engineering from Indian Institute of Science, Bangalore, India, in 1998 and 2004, respectively. He worked as research scientist at Institute of Robotics and Intelligent Systems (IRIS), Bangalore, India, during 1998–2001. He is currently, pursuing his Ph.D study in swarm intelligence based optimization with applications to collective robotics at the Department of Aerospace Engineering, Indian Institute of Science, Bangalore, India. His research interests include swarm robotics, multi-robot systems, bio-inspired optimization algorithms, theoretical analysis of natural and robotic swarms, and agreement problems in multi-agent networks.

D. Ghose is a Professor in the Department of Aerospace Engineering at the Indian Institute of Science, Bangalore, India. He obtained a B.Sc. (Engg.) degree from the National Institute of Technology (formerly the Regional Engineering College), Rourkela, India, in 1982, and an ME and a Ph.D. degree, from the Indian Institute of Science, Bangalore, in 1984 and 1990, respectively. His research interests are in guidance and control of aerospace vehicles, collective robotics, multiple agent decision-making, distributed decision-making systems, and scheduling problems in distributed computing systems. He is an author of the book Scheduling Divisible Loads in Parallel and Distributed Systems published by the IEEE Computer Society Press (presently John Wiley). He is in the editorial board of the IEEE Transactions on Systems, Man, and Cybernetics, Part A: Systems and Humans, and the IEEE Transactions on Automation Science and Engineering. He has held visiting positions at the University of California at Los Angeles and several other universities. He is an elected fellow of the Indian National Academy of Engineering.