Data-Driven Predictive Modeling to Enhance Search Efficiency of Glowworm-Inspired Robotic Swarms in Multiple Emission Source Localization Tasks

Payal Nandi
Old Dominion University, pnand001@odu.edu

DATA-DRIVEN PREDICTIVE MODELING TO ENHANCE SEARCH EFFICIENCY OF GLOWWORM-INSPIRED ROBOTIC SWARMS IN MULTIPLE EMISSION SOURCE LOCALIZATION TASKS

by

Payal Nandi
B.Tech. July 2016, Chennai Hindustan University, India

A Thesis Submitted to the Faculty of
Old Dominion University in Partial Fulfillment of the
Requirements for the Degree of

MASTER OF SCIENCE

MECHANICAL ENGINEERING

OLD DOMINION UNIVERSITY
August 2023

Approved by:

Dr. Krishnanand N. Kaipa (Director)

Dr. Miltos Kotinis (Member)

Dr. Michel A Audette (Member)

ABSTRACT

DATA-DRIVEN PREDICTIVE MODELING TO ENHANCE SEARCH
EFFICIENCY OF GLOWORM-INSPIRED ROBOTIC SWARMS IN MULTIPLE
EMISSION SOURCE LOCALIZATION TASKS

Payal Nandi
Old Dominion University, 2023
Director: Dr. Krishnanand N. Kaipa

In time-sensitive search and rescue applications, a team of multiple mobile robots broadens the scope of operational capabilities. Scaling multi-robot systems (< 10 agents) to larger robot teams (10 – 100 agents) using centralized coordination schemes becomes computationally intractable during runtime. One solution to this problem is inspired by swarm intelligence principles found in nature, offering the benefits of decentralized control, fault tolerance to individual failures, and self-organizing adaptability. Glowworm swarm optimization (GSO) is unique among swarm-based algorithms as it simultaneously focuses on searching for multiple targets. This thesis presents *GPR-GSO*—a modification to the GSO algorithm that incorporates Gaussian Process Regression (GPR) based data-driven predictive modeling—to improve the search efficiency of robotic swarms in multiple emission source localization tasks. The problem formulation and methods are presented, followed by numerical simulations to illustrate the working of the algorithm. Results from a comparative analysis show that the GPR-GSO algorithm exceeds the performance of the benchmark GSO algorithm on evaluation metrics of swarm size, search completion time, and travel distance.

ACKNOWLEDGMENTS

Thank you to the Old Dominion University Mechanical and Aerospace Engineering Department and faculty for providing me with the knowledge and skills needed to further my education. A special thank you to my advisor, Dr. Kaipa, for supporting and guiding me through my master’s work and graduate education. Thank you to Dr. Audette and Dr. Kotinis for being a part of my thesis committee and devoting time and effort to this study. And thank you to family and friends who have supported and encouraged me to continue to pursue my goals.

TABLE OF CONTENTS

Page

Page

LIST OF TABLES

LIST OF FIGURES

Figure Page

Page

Page

CHAPTER I

INTRODUCTION

1.1 Background and Motivation

Autonomous mobile robots are increasingly being deployed in search and rescue missions, especially those involving significant risk to humans, in recent years [1]. Examples of such critical tasks include humanitarian demining [2], handling nuclear waste [1], and localizing sources of emissions such as chemical, gas, and nuclear leaks [2]. Such time-sensitive applications can benefit from a team of multiple mobile robots to broaden the scope of operational capabilities. The search task considered in this thesis consists of localizing a group of multiple emission sources randomly distributed in a physical two-dimensional space. Missions demanding simultaneous searches over landscapes covering vast areas require scaling the size of multi-robot systems (< 10 agents) to larger robot teams (10 – 100 agents). Centralized control and coordination schemes do not scale well to such large teams as they become computationally intractable during the runtime of the search mission. One solution to address this problem is inspired by swarm intelligence [14] principles found in nature, offering the benefits of decentralized control, fault tolerance to individual failures, and self-organizing adaptability [8-9]. Glowworm swarm optimization (GSO) is unique among swarm-based algorithms as it simultaneously focuses on searching for multiple targets [4-7]. However, GSO is a model-free algorithm in that the decision-making of the members in the swarm does not rely on the underlying model of the landscape of the emission sources. Instead, movement-decisions of each member are strictly based on the measurement of the emission at its location and measurements communicated by other members in a local neighborhood.

1.2 Thesis Contributions

One of the primary contributions of this thesis is based on the premise that the search efficiency of a robotic swarm can be enhanced by real-time modeling of the emission landscape based on the information gathered (measurements made) by the swarm members as they move along the search process. Consequently, this thesis presents GPR-GSO—a modification to the GSO algorithm that incorporates Gaussian Process Regression (GPR) based data-driven predictive modeling—to improve the search efficiency of robotic swarms in multiple emission source localization tasks. The problem formulation and methods are presented, followed by numerical simulations to illustrate the working of the algorithm. Results from a comparative analysis show that the GPR-GSO algorithm exceeds the performance of the benchmark GSO algorithm on evaluation metrics of swarm size, search completion time, and travel distance.

1.3 Related Work: Robotic Algorithms for Multiple Emission Source Localization

Related to the topic of robotic algorithms for multiple emission source localization, researchers have tackled a broader problem in the field of multimodal optimization, where the goal is to locate all local optima of a function without knowing its gradient [2]. This complexity becomes even more pronounced in the context of robotic multisource localization, where practical limitations such as communication constraints, sensing capabilities, data storage, and robot platform design significantly impact the computational complexity of potential strategies. Moreover, the nature of the emission sources plays a critical role in determining the search performance, as different emissions, such as chemicals, light, sound, RF waves, and radioactivity, exhibit distinct emission profiles affected by the environment [2]. Research in this area has led to the development of general solutions to the multisource problem, considering various approaches. One group of algorithms draws inspiration from classical hill climbing or gradient ascent techniques [2], while another is inspired by the behavior of biological systems like bacterial colonies or swarms [27]. A third family employs probabilistic methods, including Bayesian occupancy grids and Bayesian filtering techniques [28,29]. These general solutions aim to address the challenge of allocating network resources to simultaneously find an unknown number of emission sources in a given environment, distinguishing this task from traditional target observation scenarios where known targets are tracked.

It is worth noting that the strategies employed for coverage exploration and environmental monitoring with mobile robotics systems are related to multisource localization but do not offer a complete solution, as they often deal with sensing regions and limited spatial information. The survey presented by McGill and Taylor [2] discusses relevant works [30-40] that contribute to this evolving field, exploring methods that enable real-time modeling of the emission landscape based on information gathered by the swarm members during the search process. The survey's findings and insights from other related research collectively advance our understanding of decentralized control and coordination principles inspired by swarm intelligence, offering benefits such as fault tolerance, adaptability, and improved performance in challenging and time-sensitive applications like search and rescue missions involving multiple emission sources.

The techniques developed in this thesis were primarily motivated by the success of the Bayes Swarm algorithm in decentralized path planning for autonomous vehicles [8] [9]. In this research, authors investigated the potential of the Gaussian process model (GPM) to enhance the performance of the Glowworm Swarm Optimization (GSO) algorithm for path planning in underwater environments. The authors investigated Bayes Swarm [8] [9] algorithm to understand its strengths and showed that it outperforms traditional algorithms such as random walk. The Bayes Swarm is a decentralized algorithm that extends Gaussian process modeling and integrates physical robot constraints and other robots' decisions to perform informative path planning while mitigating knowledge uncertainty. Additionally, the authors simulated a parallelized implementation of Bayes Swarm to allow asynchronous search planning over complex multi-modal signal distributions. Their experimental results showed that Bayes Swarm outperforms GSO, prompting us to incorporate GPM for predictive modeling in GSO and analyze its performance.

In the realm of swarm robotic systems (SRSs) [1], search and tracking algorithms play a pivotal role in efficiently locating and monitoring targets. As discussed in the survey article [1], two main categories of algorithms have been explored: those inspired by swarm intelligence (SI) and those based on other methods. The SI algorithms exhibit a natural fit for SRSs due to their emphasis on decentralized local control, local communication, and the emergence of global behavior through self-organization. Consequently, many existing search and tracking algorithms for SRSs have been derived from prominent SI algorithms. One such algorithm that finds application in both SI and swarm robotic systems is the Particle Swarm Optimization (PSO) [15], [16], [17]. Initially developed to simulate flocking behavior in birds, PSO models potential solutions as particles that move through a search space towards positions where optimal results are achieved. In the context of swarm robotics [12], [13], each robot is mapped to a particle, and distributed methods are employed to approximate global solutions due to real-time constraints and NP-completeness of optimization problems. PSO has been successfully adapted for multi-modal problems by utilizing multiple swarms, such as the Species-based PSO (SPSO) [18] and Niching PSO (NichePSO) [19], which locate multiple optimal solutions for multi-modal optimization problems. One notable search algorithm for multi-target localization is the Multi-Robot Particle Swarm Optimization (MR PSO) proposed in [23]. The algorithm is designed to find a known number of stationary targets within an indoor environment. Each target is equipped with a mobile phone emitting RF signal, and robots are equipped with sensors to detect these signals. At the beginning of the search, each robot is assigned to a specific target, forming subswarms that stay in their local neighborhood throughout the search. These subswarms exchange the best detection positions with each other. The algorithm also uses an adaptive Received Signal Strength (RSS) weighting factor in the velocity calculations to slow down the robot as it approaches the target, reducing overshooting.

Another notable algorithm, the Bees Algorithm (BA) [24], models the foraging behavior of a bee colony to locate the richest and closest food sources. Gaussian Process Regression versions of the BA, such as the Distributed Bees Algorithm (DBA) [25, 26], have been designed for multi-target search and coverage in unknown areas. In the DBA, robots in the swarm distribute themselves based on target fitness, where higher quality targets attract more robots. The DBA overcomes the centralization and lack of collective component issues in the original BA by employing a distributed approach, allowing robots to communicate and calculate their utilities with respect to different targets. The survey article provides valuable insights into various search and tracking algorithms for swarm robotic systems, addressing different challenges such as multi-modal optimization and target search in unknown environments. By drawing connections between SI algorithms and swarm robotic systems, researchers have been able to adapt and apply these algorithms effectively to address complex search and localization tasks. The exploration of such algorithms offers promising avenues for enhancing the search efficiency of robotic swarms in multiple emission source localization tasks as discussed in [22] and other relevant works referenced in the survey.

Another algorithm, the Clustering PSO (CPSO), is presented in [20]. It is aimed at addressing dynamic optimization problems were locating and tracking multiple changing optima over time is essential. CPSO starts from an initial swarm, the "cradle swarm," and uses a hierarchical clustering method to create subswarms. Each subswarm is assigned to different promising subregions, and the number of subswarms is adaptively adjusted based on the number of targets and automatically calculated for each subswarm. Though CPSO has not been implemented on a robotic system, it shows promise to be adapted into an SR algorithm.

Bacterial Foraging Optimization (BFO) was proposed as a multimodal function optimization algorithm inspired by the chemotactic behavior of bacteria such as E. coli in environments with nutrients [21]. BFO has been applied to chemical concentration map building, where a multi-robot system (MRS) searches an unknown area to find the region with the highest chemical gas concentration and build the concentration map. The algorithm has proven effective for multiple target tracking and has been tested on moving peaks benchmark problem [1].

Overall, these algorithms show promise for swarm robotic systems' search and tracking tasks, with each having its strengths and weaknesses. Further research on this topic will continue to improve their effectiveness and efficiency in real-world applications. Informed by a thorough survey of the literature in swarm robotic algorithms for search tasks, we investigated the proposed modification to the Glowworm swarm optimization (GSO) algorithm, called GPR-GSO, that incorporates Gaussian Process Regression (GPR) based data-driven predictive modeling, and its potential to improve the search efficiency of robotic swarms in multiple emission source localization tasks.

CHAPTER II

PROBLEM FORMULATION

2.1 Cooperative Search Task: Multiple Emission Source Localization by a Robotic Swarm

The search task considered in this thesis consists of using a swarm of mobile robots to carry out a coordinated search and localize a group of multiple emission sources randomly distributed in a physical two-dimensional space. The different aspects—emission sources, robot movement models, and search/coordination strategies—of the problem can be formalized as follows.

2.1.1 *Emission Sources*

Examples of emission sources in potential applications include light, sound, chemical/nuclear leaks, origins of fires, etc. [2]. In particular, the emission sources considered in this thesis have the following properties:

1. Each source is assumed to emit (radiate, release, discharge, etc.) a signal (light, odor, chemical, etc.) and generate an emission landscape such that the intensity of the signal is maximum at the source and decays monotonically with distance from the source.
2. The number of emission sources spread in the environment are unknown.
3. The sources are randomly distributed in the search space. Therefore, the inter-source distances are assumed to be unknown.
4. It is assumed that there are no external disturbances (wind, obstacles, etc.), distorting the profile of the emission landscape. Therefore, the emission source landscapes considered in this thesis can be mathematically modeled by continuous

multimodal functions. One example is the Peaks function [63] MATLAB also provides a predefined function for the same:

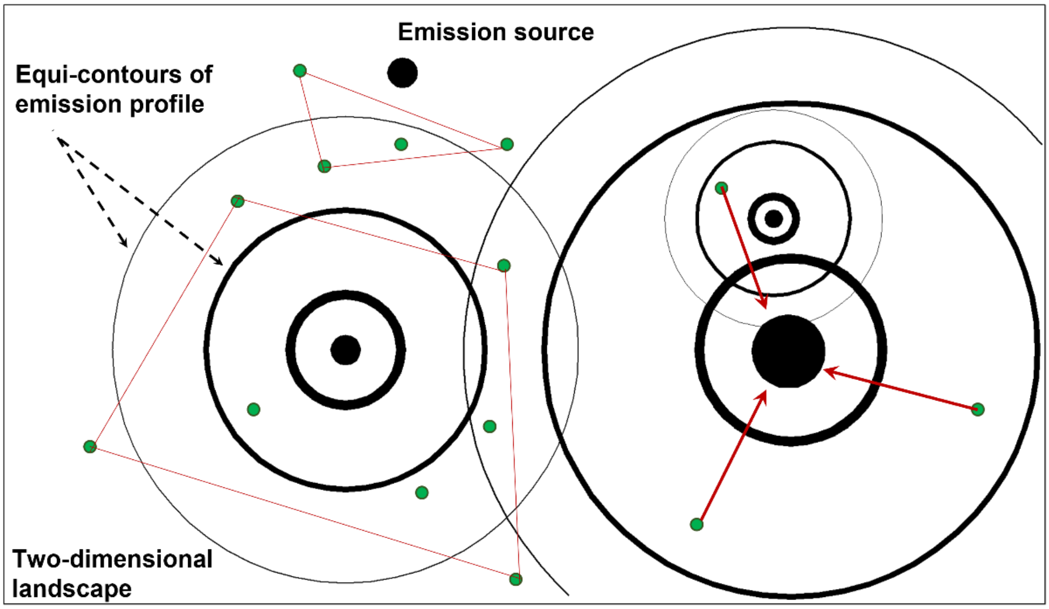


Figure 1. Properties of emission sources

2.1.2 *Robotic Swarm Model*

The robotic swarm is assumed to consist of a collection of N mobile robots that move in two-dimensional (2D) space. The kinematics and dynamics associated with real mobile robots deployed in a practical scenario are ignored, since the thesis is primarily focused on investigating the role of robots' decision-making mechanism in enhancing the search efficiency of the swarm. Accordingly, a simple point robot model is used to model the movements of robots in the swarm. In particular, the following assumptions are considered in the point-robot model:

- Each robot is free to move in any direction in a two-dimensional space R^2

- A robot's movements are discrete in nature. That is, each robot moves a fixed distance s at each step or iteration of the search process.

2.1.3 *Search and Coordination Strategies*

After a thorough literature survey of swarm robotic algorithms that have been developed for multiple emission source localization tasks, the *glowworm swarm optimization* (GSO) is chosen as the baseline algorithm to generate the search and coordination strategies of robots in the swarm. The choice of GSO is primarily motivated by its ability to address the problem of source multiplicity directly via adaptive splitting of swarm into subswarms.

2.1.4 *Data-driven Modeling to Enhance Search Efficiency of the Robotic Swarm*

Note that GSO is a model-free algorithm in that the decision-making of the members in the swarm does not rely on the underlying model of the landscape of the emission sources. Instead, movement-decisions of each member are strictly based on measurement of the emission at its location and measurements communicated by other members in a local neighborhood. Consequently, one of the primary contributions of this thesis is based on the premise that the search efficiency of the robotic swarm can be enhanced by real-time modeling of the emission landscape based on the information gathered (measurements made) by the swarm members as they move along the search process.

2.2 Specific Aims

The goal of this thesis is to use data-driven modeling to enhance the search efficiency of swarm robotic algorithm used for localization of multiple emission sources.

- **Specific Aim #1.** Devise a modification to the glowworm swarm optimization algorithm architecture to incorporate a real-time data-driven modeling of the emission landscape and leverage such estimated information to guide the movement-decisions of the swarm members during their search for multiple emission sources.

- **Specific Aim #2.** Investigate the efficacy of the Gaussian Process Regression glowworm swarm optimization (GPR-GSO) algorithm by performing a comparative analysis with GSO as the baseline algorithm using metrics of swarm size, search completion time, and travel distance of robots.

CHAPTER III

METHODS

As described in Chapter II, the primary thesis objective is to investigate the search efficiency of the GSO algorithm modified by incorporating data-driven modeling to guide the movement decisions of a swarm of robots deployed to localize a set of multiple emission sources. The proposed approach of modeling the emission source landscape in real time as the swarm members move along the search process is implemented using the *Gaussian Process Modeling* (GPM) technique. As the GSO algorithm serves as the baseline swarm search and coordination technique, a summary of the algorithm follows next.

3.1. GSO Algorithm

In GSO, a swarm of Glowworms are initially randomly distributed in the search space. Glowworms are modeled after glowworms. Accordingly, they carry a luminescent quantity called *luciferin* along with them. The glowworms emit a light whose intensity is proportional to the associated luciferin and interact with other Glowworms within a variable neighborhood. In particular, the neighborhood is defined as a local-decision domain that has a variable neighborhood range bounded by a radial sensor range $r_s(0 < r_d^i \leq r_s)$. A glowworm *i* considers another glowworm *j* as its neighbor if *j* is within the neighborhood range of *i* and the luciferin level of *j* is higher than that of *i*. The decision domain enables selective neighbor interactions and aids in formation of disjoint sub-swarms. Each glowworm is attracted by the brighter glow of other glowworms in the neighborhood. Glowworms in GSO depend only on information available in their neighborhood to make decisions.

3.2. GPR- GSO Algorithm

The core idea behind the GPR-GSO algorithm is that members of the swarm can leverage the information (e.g., emission intensity measurements) they gather at visited sites to guide their movements in a manner that helps them localize the emission sources faster than is possible with a model-free approach, where members use their individual measurement and purely local information obtained from neighbors to decide their movements. In particular, the information gathered by the swarm members can be data-mined to generate a surrogate of the underlying multimodal function profile of emission landscape, which holds knowledge of approximate number of emission sources and their locations in the two-dimensional environment. Such information can assist swarm members in refining their search by biasing their movements with estimated directions to nearby emission source locations. The emission source direction estimates can be noisy during the initial phase of the search process. However, the accuracy of the modeled emission landscape improves with increase in the data accumulated by the swarm members as they travel from one site to another, resulting in increasingly better guidance of the member movements toward the emission sources. The proposed approach of modeling the emission source landscape during runtime as the swarm members move along the search process is implemented using *Gaussian Process Models* (GPMs) that are trained on data comprising input/output pairs of swarm member locations and their emission source measurements at their locations. This training data serves as the set of discrete observations of the prior unknown continuous cost function, underlying the physical landscape of emission sources. The MATLAB Peaks function $z = 3 * (1-x).^2 .* \exp(-(x.^2) - (y+1).^2) - 10 * \left(\frac{x}{5} - x.^3 - y.^5\right) .* \exp(-x.^2 - y.^2) - \left(\frac{1}{3}\right) * \exp(-(x+1).^2 - y.^2)$ is considered as the cost function for the study in this thesis. The GPM provides a surrogate (estimate) of the cost function with a certainty bound of 95%. This estimated cost function can be used to generate a continuous mean function, which consists of all the estimated local and global peaks. The local and global peaks can be extracted using a *local max filter*. The extracted peaks information is exploited by each swarm member to locally compute a *source-direction vector*, which represents a unit direction vector toward its nearest estimated emission source. At every iteration of the GPR-GSO algorithm, each swarm member now moves in a direction that is obtained as a weighted sum of the unit direction vectors provided by GSO (surrogate local gradient vector) and GPM (source-direction vector) computations, respectively. Swarm members guided by such modified directions are expected to enhance their search process and converge to the nearby emission sources rapidly. A summary of the GPM technique is described next.

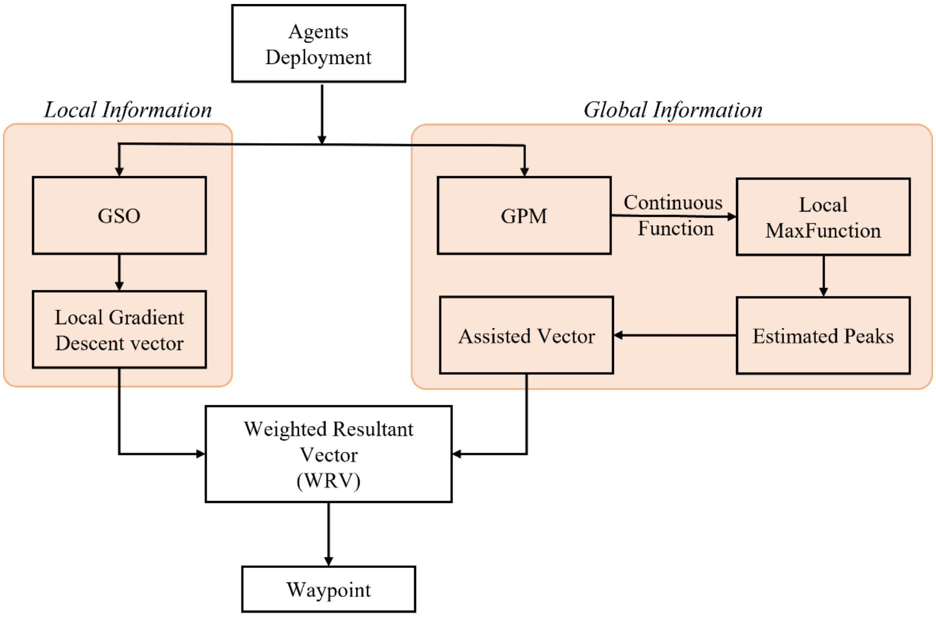


Figure 2. Schematic of GPR-GSO algorithm

A Gaussian process (GP) is a probabilistic model that defines a distribution over functions, where the goal is to make predictions or estimates based on observed data. The training Gaussian process involves two main steps: specifying a prior and updating the model with observed data to obtain a posterior distribution.

3.2.1 *Gaussian Process Models (GPMs)*

A Gaussian process (GP) is a probabilistic model that defines a distribution over functions, where the goal is to make predictions or estimates based on observed data. The training of a Gaussian process involves two steps:

1) ***Specifying a prior model distribution.*** A prior is defined by specifying a mean function and a covariance function (also called as kernel function). In our method we used a squared kernel function. The mean function represents the expected value of the function at any given input point. It is typically set to zero or a constant value.

The covariance function determines the similarity between function values at different input points. It quantifies the correlation between observations based on their input locations.

2) ***Updating the model with observed data to obtain a posterior distribution.*** Given a set of observed input/output pairs $\{[x_i(t), y_i(t), z_i(t)]: i = 1, 2, .., n\}$, where $z_i(t)$ is the emission intensity measurement made by robot i at its location $\left(x_i(t), y_i(t)\right)$ at iteration t . The goal is to update the prior distribution to obtain a posterior distribution that incorporates the observed data. The posterior distribution is computed using Bayes' rule, which involves multiplying the prior distribution by the likelihood of the data. The likelihood is determined by assuming that the observed data points are drawn independently from a Gaussian distribution with a mean predicted by the GP and a noise parameter. The posterior distribution of a GP is also a Gaussian distribution, and it can be fully characterized by its mean and covariance functions. Once the GP is trained, it can be used for prediction or estimation. Given a new input point, the GP provides a predictive distribution over the corresponding output value, which is typically, a Gaussian distribution centered around the predicted mean with a variance that represents the uncertainty of the prediction.

3) ***Local Max Operator***

The *local max* operator offers a numerical approximation method to identify a peak within a specified local neighborhood of an objective space. It is applied to identify all the peak estimates of the predicted cost function generated by the GPM. The method works by considering a discrete 2D grid of the objective space and setting a grid point as a *local peak* if the function value at that point is greater than equal to that of all its eight neighboring grid points (two grid point neighbors along x axis, two grid point neighbors along the y axis, and four grid points neighbors along the diagonals). Mathematically, let's consider a function $f(x)$. The local maximum occurs at a point $x = c$ if the following conditions are satisfied:

- $f(c) \geq f(x) \,\forall\, x \in (c - \delta, c + \delta)$, where δ is a small positive number.
- $f'\,(c) = 0$ or $f'\,(c)$ is undefined.
- $f'(x) < 0$ for $x < c$ and $f'(x) > 0$ for x

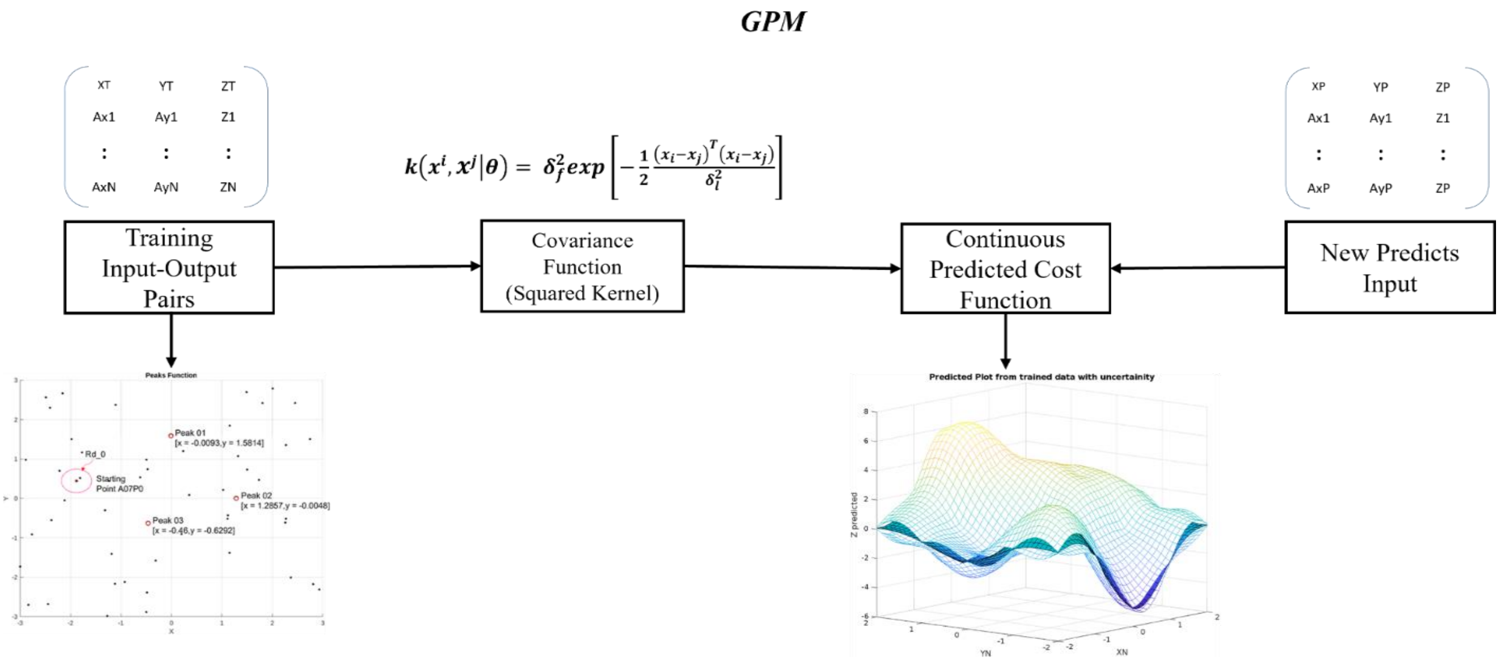

GPM
XT YT ZT
Ax1 Ay1 Z1
: : :
AxN AyN ZN
$k(x^i, x^j|\theta) = \delta_f^2 exp\left[-\frac{1}{2}\frac{(x_i-x_j)^T(x_i-x_j)}{\delta_l^2}\right]$
XP YP ZP
Ax1 Ay1 Z1
: : :
AxP AyP ZP
Training Input-Output Pairs
Covariance Function (Squared Kernel)
Continuous Predicted Cost Function
New Predicts Input
Peaks Function
Rd_0
Starting Point A07P0
Peak 01 [x = -0.0093,y = 1.5814]
Peak 02 [x = 1.2857,y = -0.0048]
Peak 03 [x = -0.46,y = -0.6292]
X
Y
Predicted Plot from trained data with uncertainity
Z predicted
YN
XN

Figure 3. Workflow of GPM

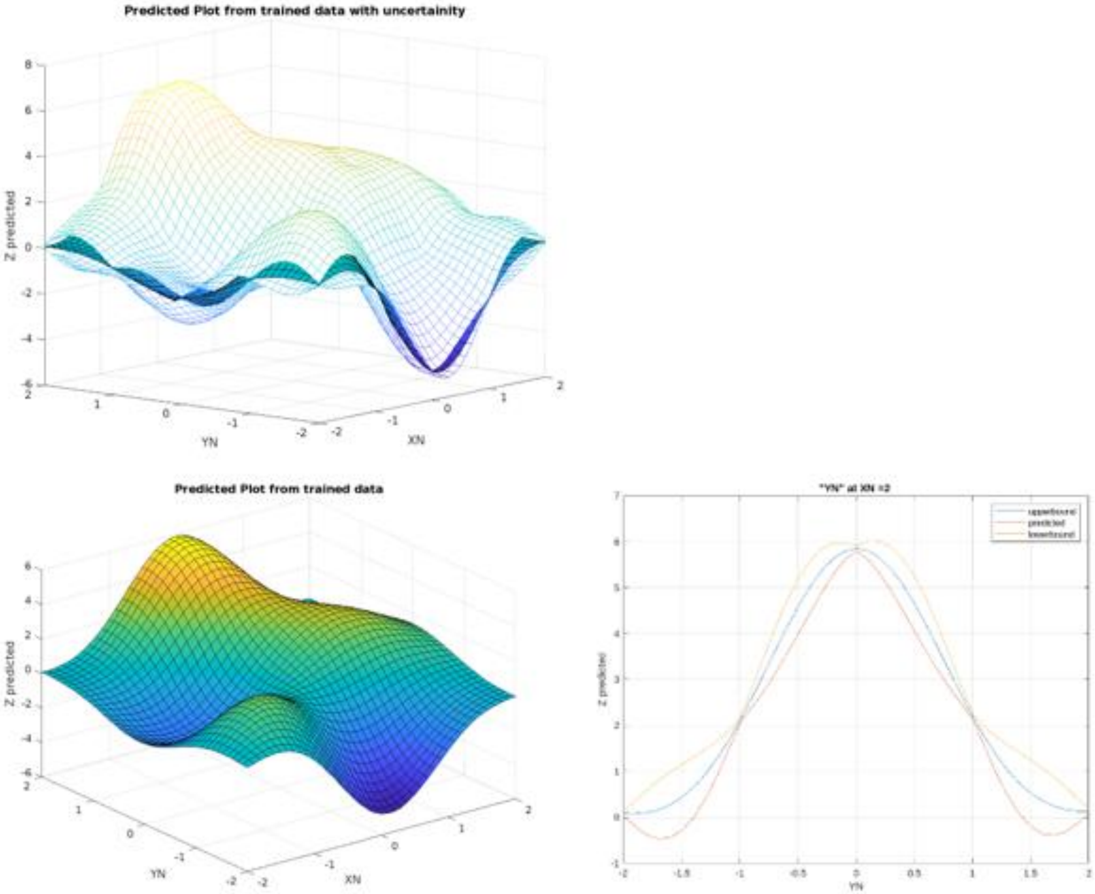


Figure 4. Predicted plots with uncertainty and trained data

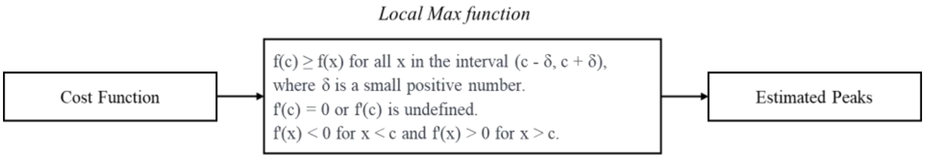

Local Max function
Cost Function
f(c) ≥ f(x) for all x in the interval (c - δ, c + δ),
where δ is a small positive number.
f'(c) = 0 or f'(c) is undefined.
f'(x) < 0 for x < c and f'(x) > 0 for x > c.
Estimated Peaks

Figure 5. Local max function

3.3. Evaluation Metrics

A comparative analysis was performed using the original GSO as the benchmark algorithm. The metrics used to evaluate the algorithmic performance are defined below:

- ***Iterations for Convergence*** is defined as the number of iterations taken by an Glowworm to travel from its initial location and converge at one of the emission source locations.
- ***Path Length-to-Euclidian Distance Ratio***. Given the initial and final locations of a glowworm, let the Euclidean distance $\boldsymbol{d}_{\in}$ represent the shortest distance between them. Let path length $\boldsymbol{d_p}$ of a glowworm be defined as the distance traveled along its path from its initial location to final location. Now, the path length-to-Euclidean distance ratio is given by $\boldsymbol{\rho} = \frac{\boldsymbol{d_p}}{\boldsymbol{d}_{\in}}$. Note that for comparison purpose. Glowworms in both algorithm implementations of GSO and GPR-GSO, the initial locations of all glowworms are kept same.
- ***Source Localization Error.*** To analyze the accuracy of the predicted location of each emission source, the estimated source locations were sorted based on the respective local peak intensity values and matched with the nearest true peaks (emission source locations). Peaks at the edges were eliminated as the emission sources are assumed to be located interior of the search space. Now, the source

localization error of each estimated source location is computed as its Euclidean distance from the location of the nearest true emission source.

- ***Swarm Size*** is defined as the minimum number of glowworms needed to find all the emission sources for a specified accuracy defined in terms of the source localization error.

CHAPTER IV

EXPERIMENTAL VALIDATION

In this chapter, I present results from a series of numerical simulations conducted to validate the techniques developed in this thesis. In particular, experiments are performed in support of accomplishing the specific aims formulated in Chapter II. First, experimental results are presented to illustrate the algorithmic steps of GSO that serves as a benchmark technique to carry out the task of multiple emission source localization. Second, experimental results are presented to illustrate the algorithmic steps of GPR-GSO to perform the same localization task. Third, results from a comparative analysis are presented in which the performance of GPR-GSO is compared with that of GSO on various metrics (e.g., swarm size, number of iterations for convergence, Euclidean distance-path length ratio, etc.) developed as a part of methods in Chapter III.

4.1 GSO for Multiple Emission Source Localization Tasks

The experiments in this section illustrate the working of GSO by showing its algorithmic steps in the context of a multiple emission source localization task. The MATLAB Peaks function is chosen to represent the landscape of the multimodal emission source profile. Note that the function profile consists of three emission sources (local peaks) at the locations -0.0093,1.5814, 1.2857,0.0048, and -0.46, -0.6292, respectively. Hereafter, Glowworms in GSO are referred to as glowworms. Figure 5 shows the paths traced by fifty glowworms as they navigate from their initially deployed locations, search through the landscape, and eventually converge to the emission sources. Next, the evolution of the path traced by one of the glowworms (A07), as shown in Fig. 6, is described to illustrate the algorithmic steps (e.g., formation of adaptive neighborhoods, movement directions of individual glowworms, etc.) of GSO. Figure 6 depicts the path taken by Glowworm A07 in the Glowworm Swarm Optimization (GSO) algorithm. from its initial position to the goal. The figure illustrates the movement and navigation capabilities of Glowworm A07, highlighting its trajectory towards the desired objective within the landscape.

Figure 7b showcases the deployment of Glowworm A07 in the GSO algorithm, with a highlighted red circle representing its radial sensor range. Figure 7c showcases the beginning of next iteration of Glowworm A07, with two red circles representing its increased radial sensor range. Figure 7d showcases the vectors joining Glowworm A07 to the neighbors.

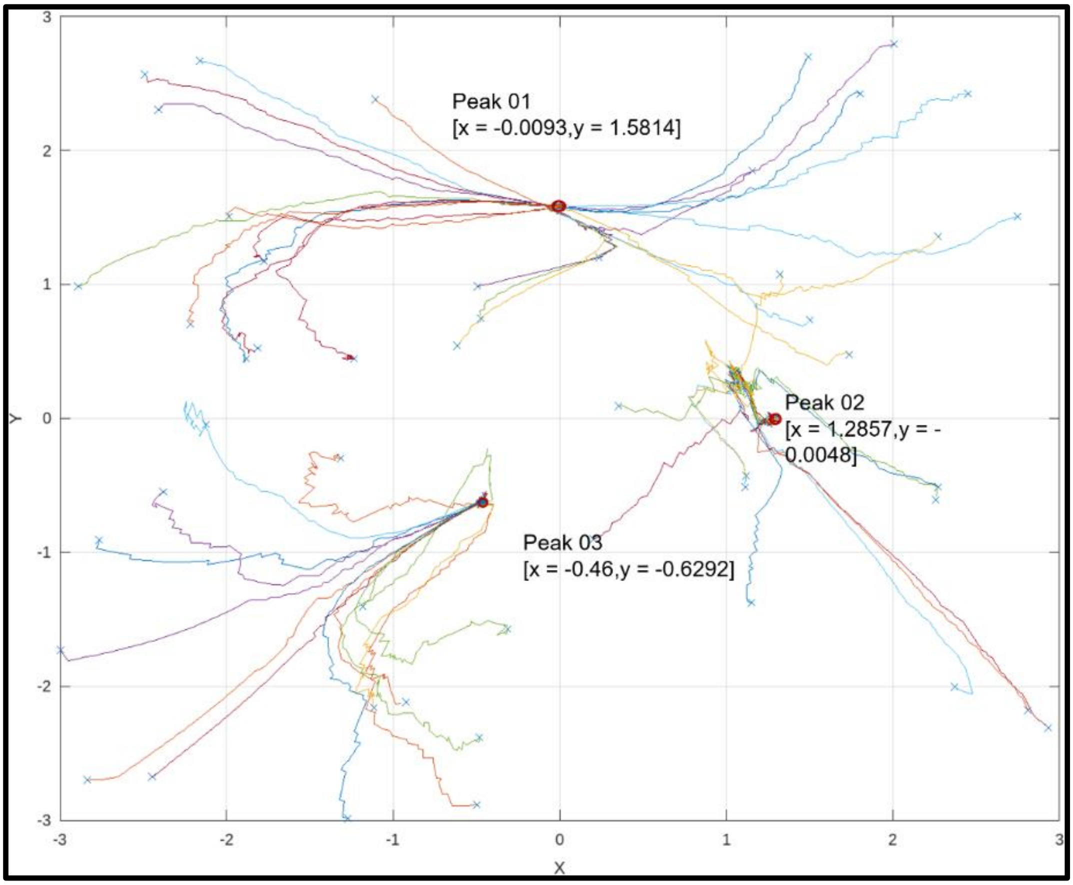


Figure 6. Glowworm Swarm Optimization (GSO)

We can observe that Glowworm A07 successfully reaches emission source 01 after approximately 140 steps. Each iteration of GSO consists of three steps. Figure 7b shows the first step of iteration-zero, comprising the initial deployment of Glowworm A07, its initial radial neighborhood, and the location of emission sources in the landscape. In the figure, the initial position of Glowworm A07 is shown as a red-colored star and the source locations are shown as red-colored circles, and the radial neighborhood is shown as a pink-colored circle. Note that there is a glowworm (black dot) present in the neighborhood of Glowworm A07.

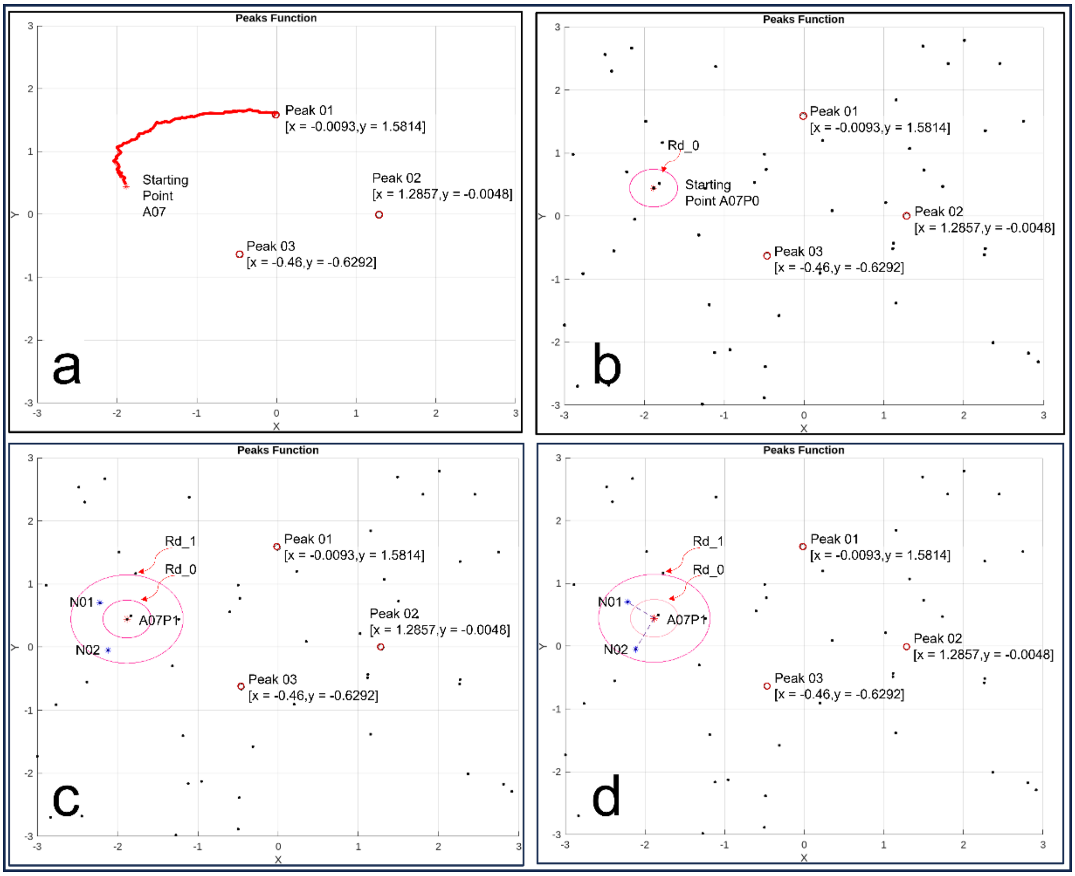


Figure 7. Path taken by Glowworm A07 in the GSO algorithm

However, it fails to meet the criteria of a neighbor based on luciferin value. Since, A07 is neighbor-less at initial deployment, the remaining two steps of the first GSO iteration are not executed. As a result, Glowworm A07 remains stationary. In a similar manner, all other glowworms without any neighbors remain stationary. Figure 6c illustrates the first iteration in which the neighborhood of Glowworm A07 expands and two new neighbors (blue-colored stars) are found as a result. Figure 6d shows Glowworm A07’s movement direction toward its leader, which corresponds to one of its neighbors N02. The leader glowworm's movement direction is indicated by the blue arrow, while Glowworm A07's movement toward N02 is represented by a red arrow. Figure 7 shows Glowworm A07 taking a step to its leader.

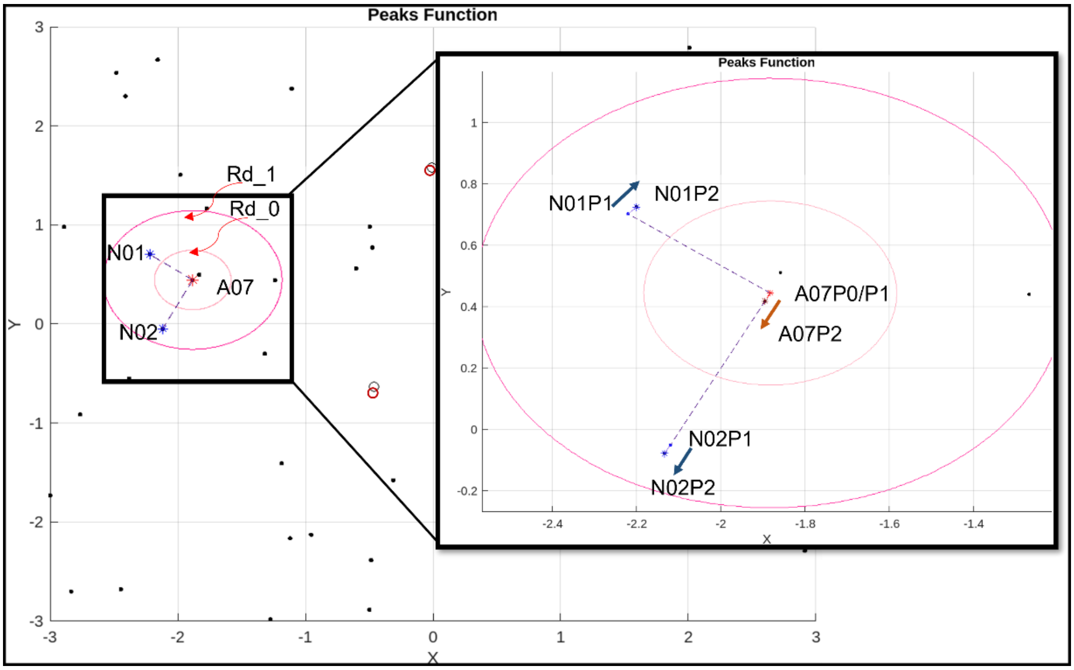


Figure 8. Movement of Glowworm A07 as it takes a step toward its leader

In Figure 8, Glowworm A07 in the GSO exhibits a new sensor radius (8a) showcasing its adaptive nature; the vectors joining Glowworm A07 to its neighbors are shown in 8b, and 8c shows Glowworm A07 moving into position P2, following the completion of Step 1 in this iteration.

As a result, the neighborhood range (Rd_3) of Glowworm A07 increases, expanding its perception of the surrounding landscape. This expanded neighborhood range enables Glowworm A07 to identify a new neighbor (N03), which is depicted as a blue star. Figure

9a marks the beginning of the second iteration. At this stage, Glowworm A07 increases its radial sensor radius (Rd_3), extending its reach to a wider area of the landscape. The figure highlights the adaptive nature of the GSO algorithm, as the sensor radius dynamically adjusts based on the number of neighbors detected in the previous iteration. In this iteration, Glowworm A07 detects a new neighbor (N03) within the expanded range, as shown by the blue star. In Figure 9b, the lines connecting Glowworm A07 to all its neighbors represent the evaluation of potential movement directions. Glowworm A07 evaluates the luciferin values of its neighbors to determine the most favorable direction for movement shown in Figure 4.4b. Among the three neighbors, Glowworm A07 selects the direction towards neighbor N02, which concludes the second iteration of Glowworm A07. Figure 9c presents a comprehensive overview of the details discussed above, showcasing the completion of the iteration and the movement of Glowworm A07 towards its intended source. This iteration demonstrates the progress made by Glowworm A07 in reaching its optimal peak, as guided by the GSO algorithm. By observing the movement of Glowworm A07 and its convergence towards the intended source, we gain valuable insights into the effectiveness of the algorithm in navigating the landscape and finding optimal solutions.

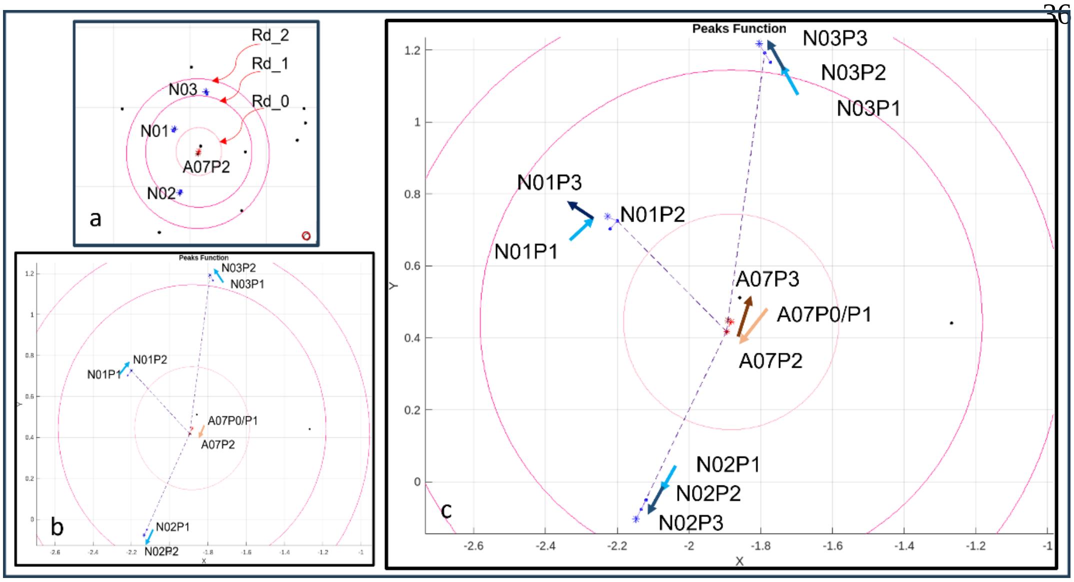


Figure 9. Glowworm A07's steps toward the lead neighbor

Figure 10 illustrates the beginning of the third iteration. At Step 1, Glowworm A07 remains in position A07P4, the location it reached in the previous iteration. Glowworm A07 now focuses on increasing its neighborhood radius, allowing for a broader perception of the landscape. The increment in the neighborhood radius is proportional to the number of neighbors detected in the previous iteration, as explained in Chapter 3. This adaptive adjustment ensures that Glowworm A07 can gather more comprehensive information about its surroundings, potentially uncovering new neighbors that were previously beyond its reach.

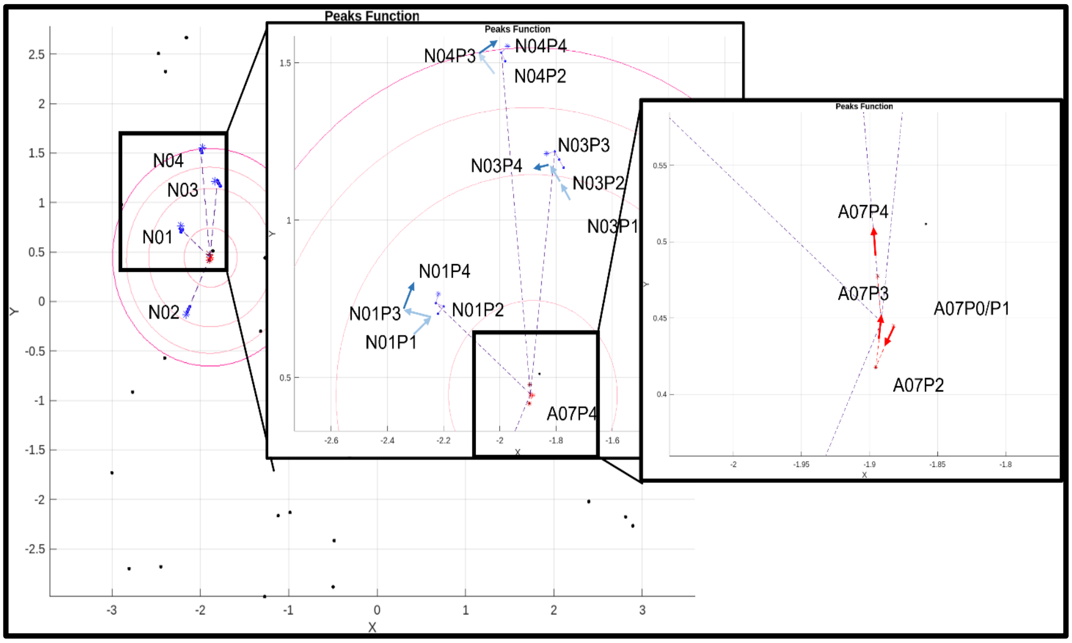


Figure 11. Glowworm A07 exhibits a new sensor radius, showcasing its adaptive nature at the beginning of Iteration 3

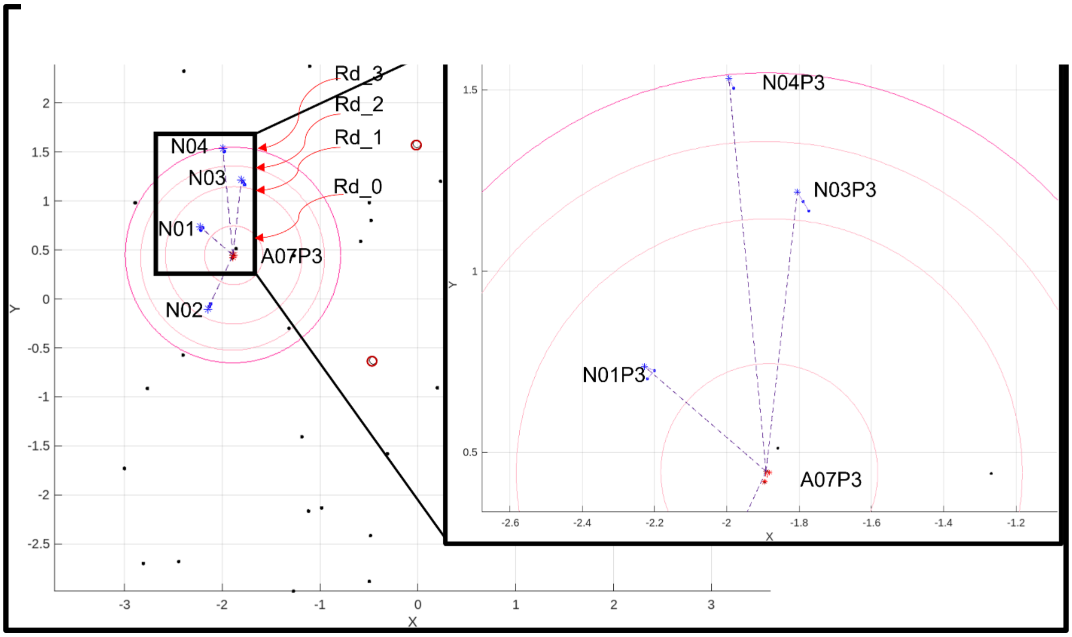


Figure 10. Glowworm A07 in the GSO showcases movement toward the lead neighbors. Positions of neighbors are depicted in the middle-zoomed section in blue. Positions of Glowworm A07 are highlighted in red in the last section.

In Figure 11 Glowworm A07 in the GSO showcases movement towards the lead neighbors. Positions of neighbor towards N04 are depicted in the middle-zoomed section in blue. Positions of Glowworm A07 is highlighted in red in the last section. The zoomed-in section of the figure provides a clear view of the position changes occurring at each iteration, allowing for a detailed analysis of Glowworm A07's progress. We also, observe all the possible new movement directions that Glowworm A07 can take. This figure showcases the exploration and evaluation of different movement options based on neighboring glowworms. The figure provides a comprehensive visual representation of the evaluation process, highlighting the diverse options available to Glowworm A07 and emphasizing the algorithm's ability to adapt and consider multiple paths towards optimal solutions. By zooming in on the movement of Glowworm A07 and its neighbors, the figure provides a detailed view of the incremental steps taken by Glowworm A07 and the corresponding adjustments made by other neighboring glowworms. This visualization allows us to track the convergence process and observe the interplay between Glowworm A07 and its surroundings, indicating the algorithm’s progress in converging towards optimal peaks.

Moving on to Figure 12a Glowworm A07 initiates another iteration. At Step 1, while remaining at position A07P4, Glowworm A07 once again increases its sensor radius to expand its perception of the landscape. However, in this particular iteration, we observe only a slight increase in the radius, which is proportional to the number of neighbors detected in the previous iteration. This adaptive adjustment in the sensor radius ensures that Glowworm A07 can strike a balance between the exploration of new areas and the exploitation of already identified peaks. The figure provides an overview of Glowworm A07's position and sensor range, setting the stage for further exploration and potential discoveries in subsequent steps. The figure also illustrates all the possible movement directions available to Glowworm A07 for Step 3, which involves selecting the movement direction based on the highest luciferin value among the available neighbors. By evaluating the luciferin values associated with each potential movement direction, Glowworm A07 can make an informed decision about the most promising direction to pursue. Figure 12b presents the fourth iteration, focusing on Step 3. Glowworm A07 continues moving towards N04, as observed in the previous iterations. The figure highlights the movement of Glowworm A07 and its neighbors, allowing us to observe their incremental steps towards convergence. For simplicity, only the motion of neighbors and the time when they were identified as neighbors are tracked, while the other glowworms are represented as black dots are shown at Position 5, which indicates the completion of Iteration 4. This visualization provides insights into the algorithm's progress in narrowing down the search space and converging towards optimal peaks, highlighting the coordinated movements of Glowworm A07 and its neighbors as they approach the desired solutions. By providing a detailed analysis of each figure, we gain a comprehensive understanding of the GSO algorithm's progression and convergence towards optimal solutions.

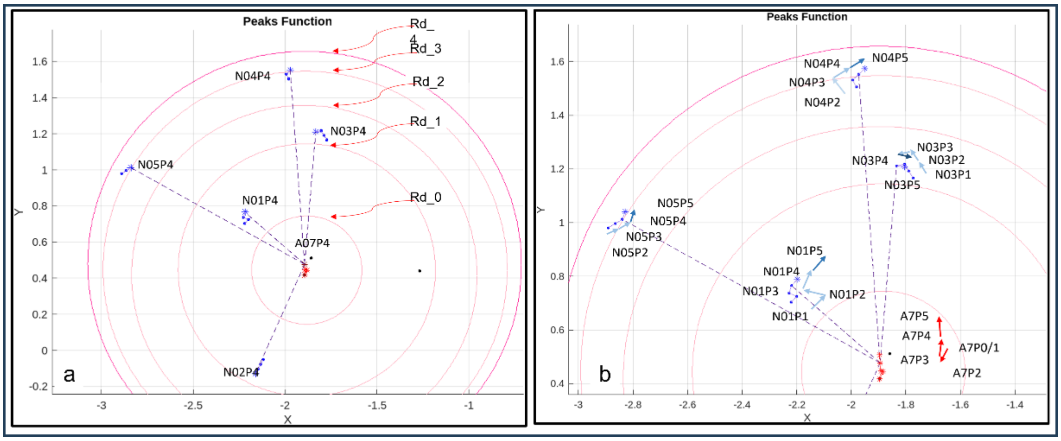


Figure 12. a) Glowworm A07 in the GSO exhibits a new sensor radius, showcasing its adaptive nature. A total of five circles increasing the radius, along with a line joining the neighbors, b) movements toward the lead neighbor

4.2 GPR-GSO

Figure 13 provides a comprehensive view of the movement of all 50 glowworms over 100 iterations using the GPR-GSO algorithm. By referring to Figure 13, we can observe a pattern shift compared to the GSO algorithm, as discussed in Section 3.2. This figure allows me to analyze the differences in the paths followed by glowworms in both algorithms.

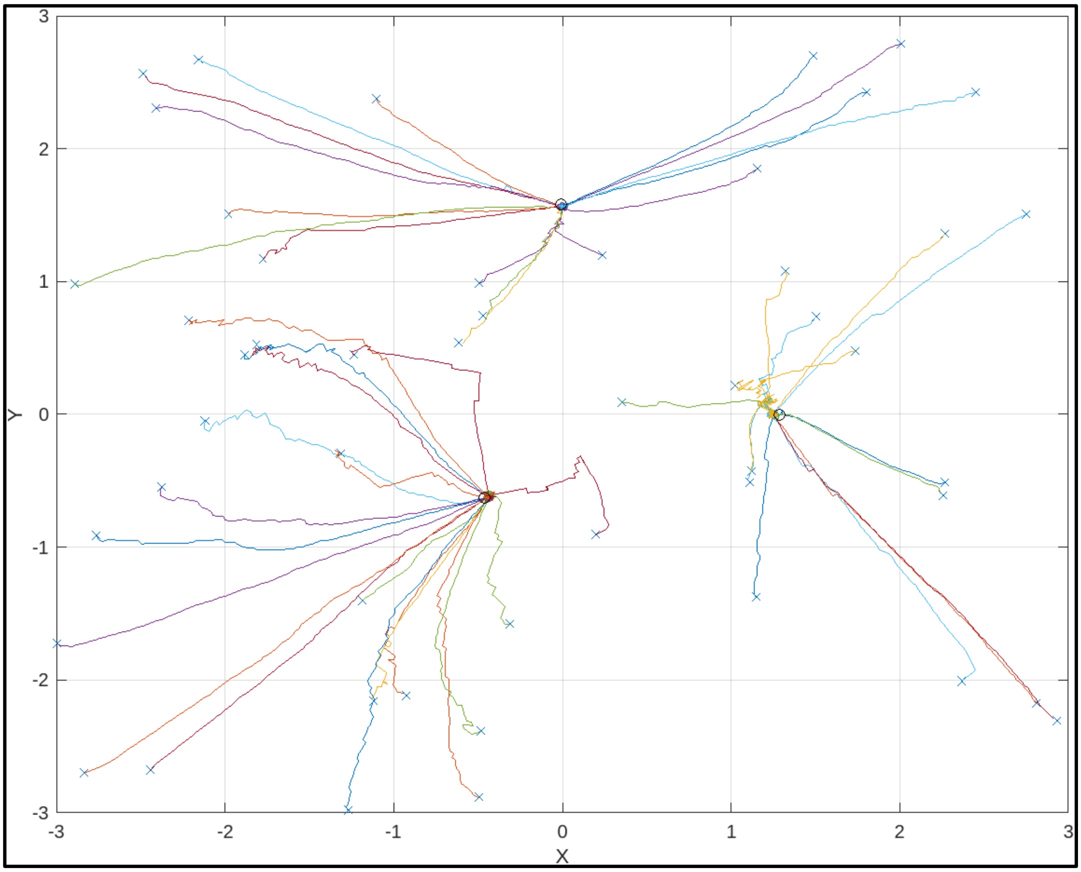


Figure 13. Pathway of all agents demonstrating the work of the Gaussian Process Registration Glowworm Swarm Optimization (GPR-GSO) algorithm through a graphical representation of its algorithmic steps. This is a visual overview of how glowworms move.

The starting deployed positions are the same as in GSO, generated using the *rng* function [43], enabling a detailed comparison. The algorithm ran for 140 iterations, providing insights into the convergence and exploration capabilities of the GPR-GSO algorithm, which will help to analyze the difference between the paths followed in both algorithms.

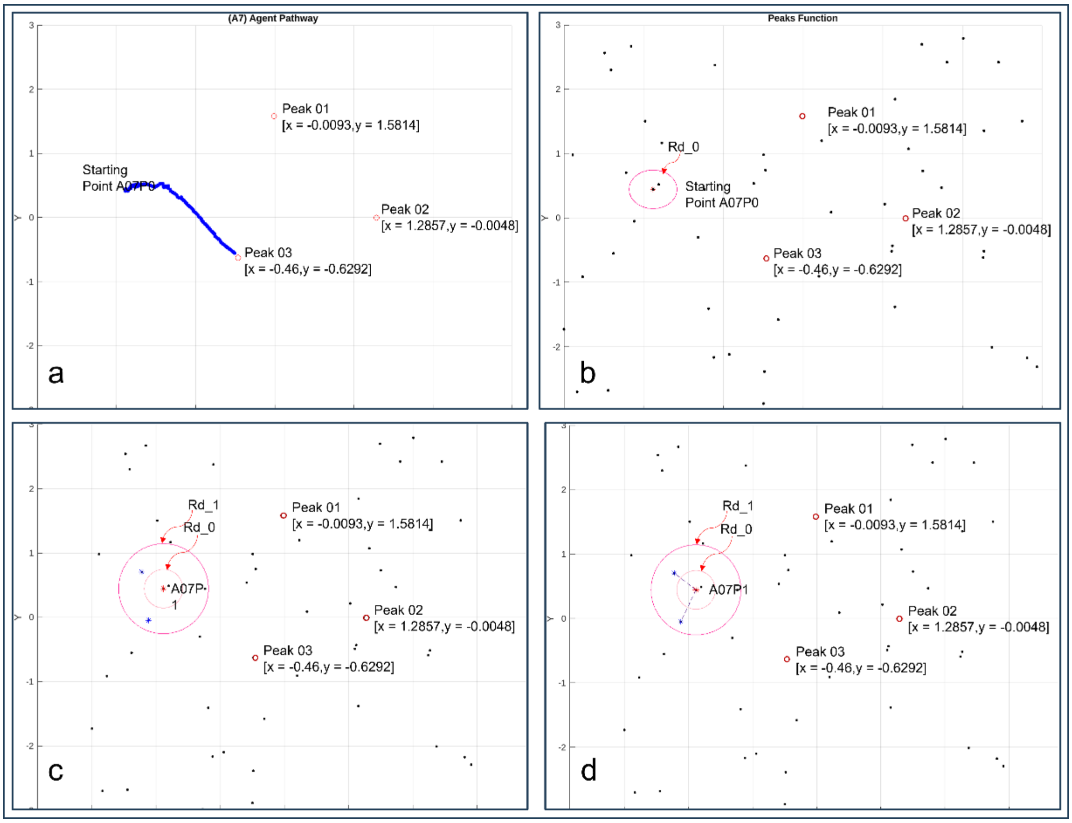


Figure 14. The path taken by Glowworm A07 in the GPR-GSO algorithm from its a) initial position to the goal illustrating the movement and navigation capabilities of Glowworm A07, highlighting its trajectory towards the desired objective within the landscape, b) the deployment of Glowworm A07 in the GPR-GSO algorithm, with a highlighted red circle representing its radial sensor range, c) the beginning of next iteration of Glowworm A07, with two red circles representing its increased radial sensor range, and d) the vectors joining Glowworm A07 to the neighbors.

Figure 14 displays the path traced by Glowworm A07 using the GPR-GSO. The algorithm ran for 140 iterations. But the glowworm converged earlier around 75 steps which was achieved by taking a more directed route. Here we can see that the Glowworm A07 in Figure 14a traversed the path reaching source peak 03, example such as this helps distinguish the subtle but important difference of behavior in the algorithm and how it helps to reach convergence with fewer steps.

Figure 14 represents the initial state of the algorithm, where Glowworm A07 gets deployed and scans the surrounding area with the default radial sensor perimeter. In this iteration, Glowworm A07 detects one glowworm neighbor, but it fails to meet the criteria for considering it as a neighbor. As a result, no movement takes place, and Glowworm A07 remains stationary. This figure illustrates the initial step of the algorithm, emphasizing the importance of neighbor detection and selection in subsequent iterations. Moving to Figure 14c, we observe Glowworm A07 remaining in its previous position as the first iteration's Step 1 begins. Glowworm A07 updates its sensor radius Rd_1 and successfully detects two neighbors. This figure showcases the initial steps of the algorithm, highlighting the neighbor detection process and the expansion of Glowworm A07's awareness of its surroundings. Figure 14d focuses on Step 2 of the first iteration. In this step, Glowworm A07 evaluates all the possible directions to move based on the information gathered in the previous step. This figure visually represents the exploration of movement options, providing a comprehensive view of the potential paths Glowworm A07 can take in its search for optimal peaks.

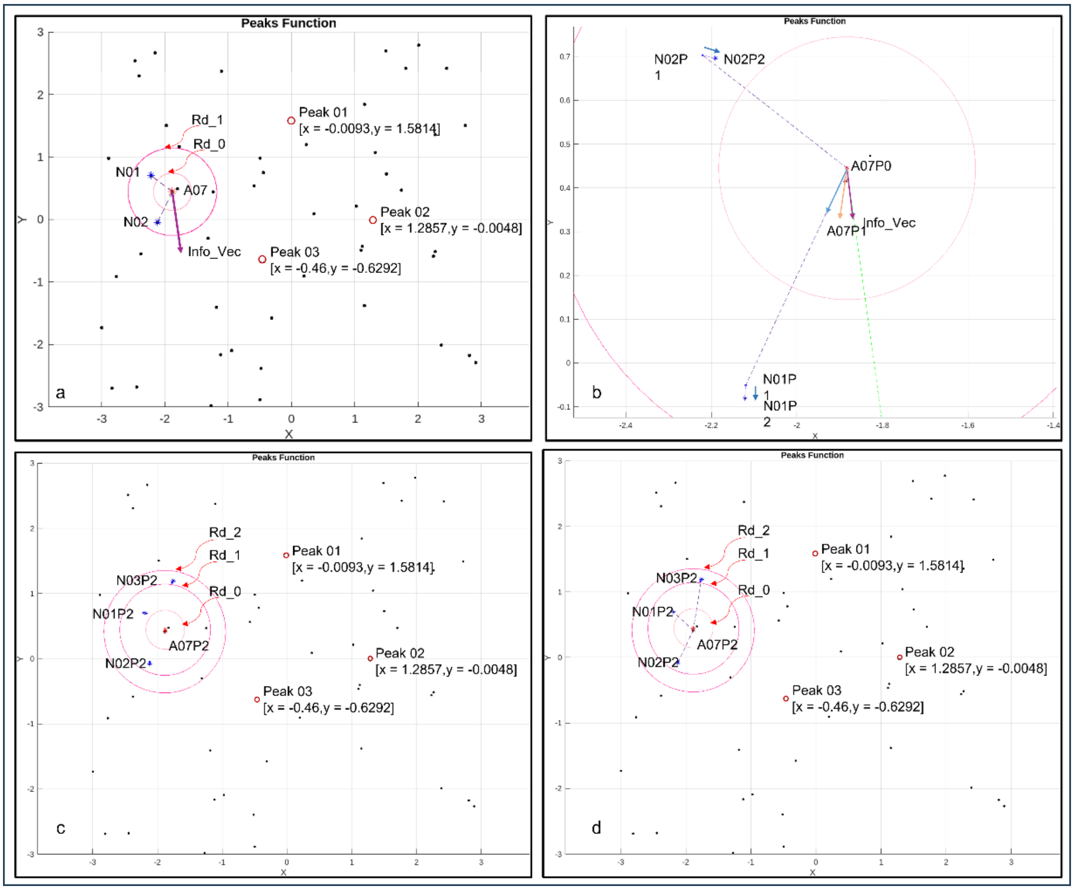


Figure 15. *Glowworm A07 in the Gaussian Process Regression Glowworm Swarm Optimization (GPR-GSO) algorithm, a) the informed vector guiding its movement towards the nearest peaks, as well as the connection lines representing its neighboring agents, b) Glowworm A07 moves along weighted resultant vector, c) in the beginning of next iteration a third sensor is visible, and d) the lines joining the neighbors.*

Moving forward to Figure 15a, we delve into Step 3 of the first iteration. This step involves considering the information derived from the GP model and a local maxima finding algorithm. The figure represents this information as an informed vector, which points towards the bottom of the landscape and away from the known peaks' locations. This behavior arises from the GP model's training, where the glowworms are initially deployed at a distance, and the covariance is proportional to the data's position. As the figure depicts, the prediction improves in subsequent iterations, leading to more accurate informed vectors. Figure 15b showcases the final step (Step 4) of the first iteration. Glowworm A07 can be seen moving a unit distance between N02 and the informed vector in the direction of weighted resultant vector highlighted in purple. The algorithm aims for Glowworm A07 to be equally influenced by the informed vector while also considering its leader. The weight of influence is determined based on the required situation and can be defined using convexity. This figure provides insights into the movement strategy of Glowworm A07 and its coordination with neighboring Glowworms. In Figure 15c, the second iteration begins with Glowworm A07 increasing its radial sensor perimeter Rd_02 from its new position, A07P2. The increment in the sensor radius is relatively smaller compared to the previous iteration. As Glowworm A07 already had two neighbors, the first step successfully identifies a new neighbor, N03. This figure emphasizes the adaptive nature of the algorithm, where the sensor radius dynamically adjusts based on the number of neighbors detected in the previous iteration. Figure 15d focuses on Step 2 of the second iteration. In this step, again the Glowworm A07 evaluates all the possible movement directions based on the available information. The figure represents the exploration of movement options, providing an overview of the potential paths Glowworm A07 can consider for its subsequent movement.

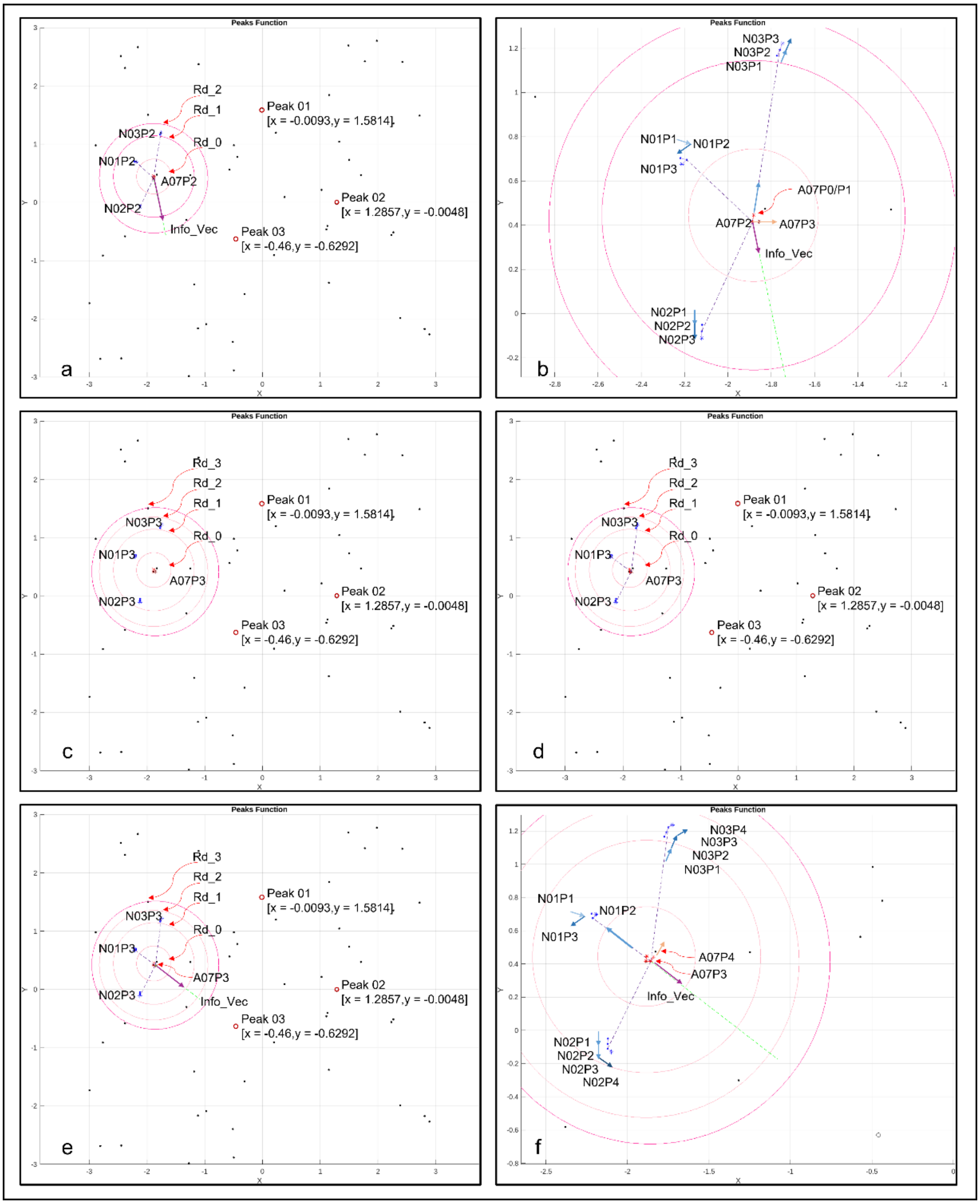


Figure 16. a) Glowworm A07 in the GPR-GSO algorithm, depicting the informed vector, as well as the connection lines representing its neighboring glowworms, b) movement towards the WRV, c) beginning of new iteration, d) lines joining the neighbors. e) the updated informed vector, and f) depicting the movement toward the WRV.

In Figure 16a, Step 3 of the second iteration takes place. Glowworm A07 considers the information provided by the informed vector, which points towards the peaks nearest to its current position. This time, the estimation has significantly improved compared to the previous iteration, indicating the effectiveness of the training process taking into global information. This figure highlights the integration of prediction information into the decision-making process of Glowworm A07.

The final step (Step 4) of the second iteration is showcased in Figure 16b. Glowworm A07 decides to move between the direction of N03 and the informed vector, following the new updated weighted resultant vector aiming to approach the nearest peak. This figure represents the coordinated movement of Glowworm A07 and emphasizes the iterative nature of the algorithm in reaching optimal solutions. Figure 16c initiates another iteration, where Glowworm A07 remains in position A07P4 and repeats Step 1. The sensor radius increment in this iteration is relatively small, and while the step is partially successful, no new neighbors are identified, and the previous neighbors remain the same. This figure demonstrates the dynamics of neighbor detection and the influence of the sensor radius increment on the algorithm’s behavior. Step 2 of the third iteration begins as shown in Figure 16d. Glowworm A07 decides on the paths to follow based on the available information, exploring potential movement directions. The figure provides an overview of the movement options considered by Glowworm A07. Figure 16e focuses on Step 3 of the third iteration. Glowworm A07 determines Neighbor N01 as the leader glowworm based on the highest probability of luciferin value at that situation. Glowworm A07 also considers the updated informed vector, which now points accurately towards the peaks after three rounds of training. This figure highlights the coordination between Glowworm A07 and its leader, as well as the integration of prediction information derived from the GP model. The final step

(Step 4) of the third iteration is depicted in Figure 16. Glowworm A07 moves a unit distance equally between N01 and the informed vector, which can also be referred to as the ascent area. This figure illustrates the movement of all the glowworms from their previous positions to their current positions.

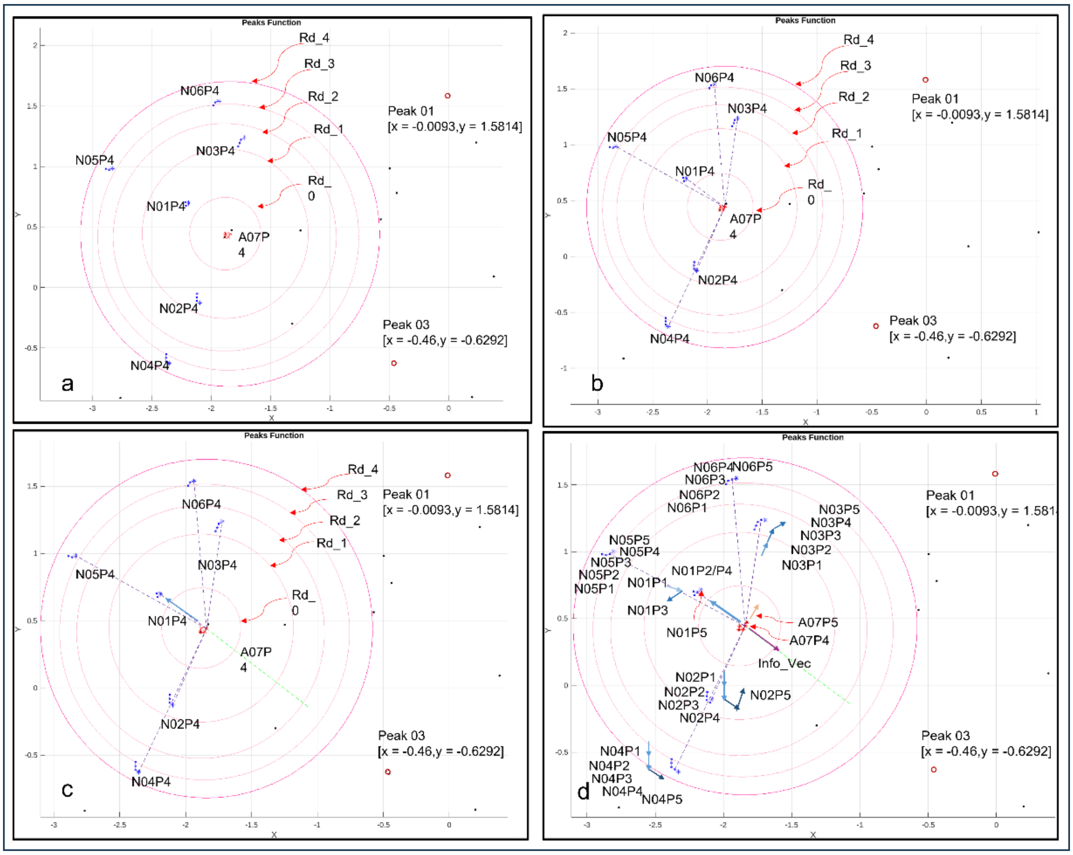


Figure 17. a) Glowworm A07 in the GPR-GSO algorithm, the beginning of new iteration, b) the connection lines representing its neighboring glowworms, c) the informed vector, d) the updated informed vector and the movement towards along the WRV.

Figure 17a presents an additional iteration to validate the accuracy of predictions using the GPM. As Glowworm A07 moves to position A07P4, Step 1 of the fourth iteration begins. Like previous iterations, the glowworm's movement starts with a small increment, and while it is successful in identifying two new neighbors, no new neighbors are detected. Figure 17b highlights the continuous neighbor detection process and the algorithm's adaptability based on the sensor radius increment. Next, during the second step of the fourth iteration, Glowworm A07 evaluates the luciferin values of the five neighbor glowworms and selects N01 as the leader. This figure emphasizes the leader selection process and its influence on Glowworm A07's movement decisions. The third step of the fourth iteration takes place in Figure 17c. Glowworm A07 considers the updated informed vector, which has been trained using the emission values sensed by all glowworms at their positions visited during the previous iterations. The figure illustrates an improved accuracy of the informed vector in pointing towards the nearest peaks from Glowworm A07’s current position. This highlights the improved prediction capabilities of the GP model as it receives more training data. Finally, in Figure 17d the movement of Glowworm A07 between the lead neighbor N01 and the informed vector is depicted. Glowworm A07 follows the guidance provided by the leader glowworm and integrates it with the prediction information to navigate towards the nearest peak. This figure illustrates the coordination between Glowworm A07, its leader, and the informed vector, emphasizing the iterative nature of the algorithm in achieving convergence. These figures collectively provide a visualization of the movement and decision-making process of Glowworm A07 and all glowworms in the GPR-GSO algorithm. They showcase the algorithm’s adaptive capabilities, exploration strategies, and convergence behavior over multiple iterations, highlighting the effectiveness of the algorithm in reaching optimal peaks.

4.3 Efficiency Analysis

4.3.1 Path Traces of All Glowworms

Figures 18 and 19 are crucial in understanding. and comparing the paths taken by glowworms in the two algorithms of GSO and GPR-GSO, respectively. These figures provide visual representations of the glowworm movements and distribution until the estimated peaks are reached. By examining the paths taken by glowworms in both algorithms, we can discern the differences in their approaches and assess the impact on overall algorithm efficiency. Furthermore, the analysis of glowworm distribution sheds light on the variations between GSO and GPR-GSO and how these differences contribute to enhanced performance, while gaining valuable insights into the distinct characteristics and benefits of utilizing the GPR-GSO algorithm for optimizing peak exploration and exploitation. Figure 18 illustrates the paths traveled by all glowworms using the GSO algorithm until the estimated peaks are reached.

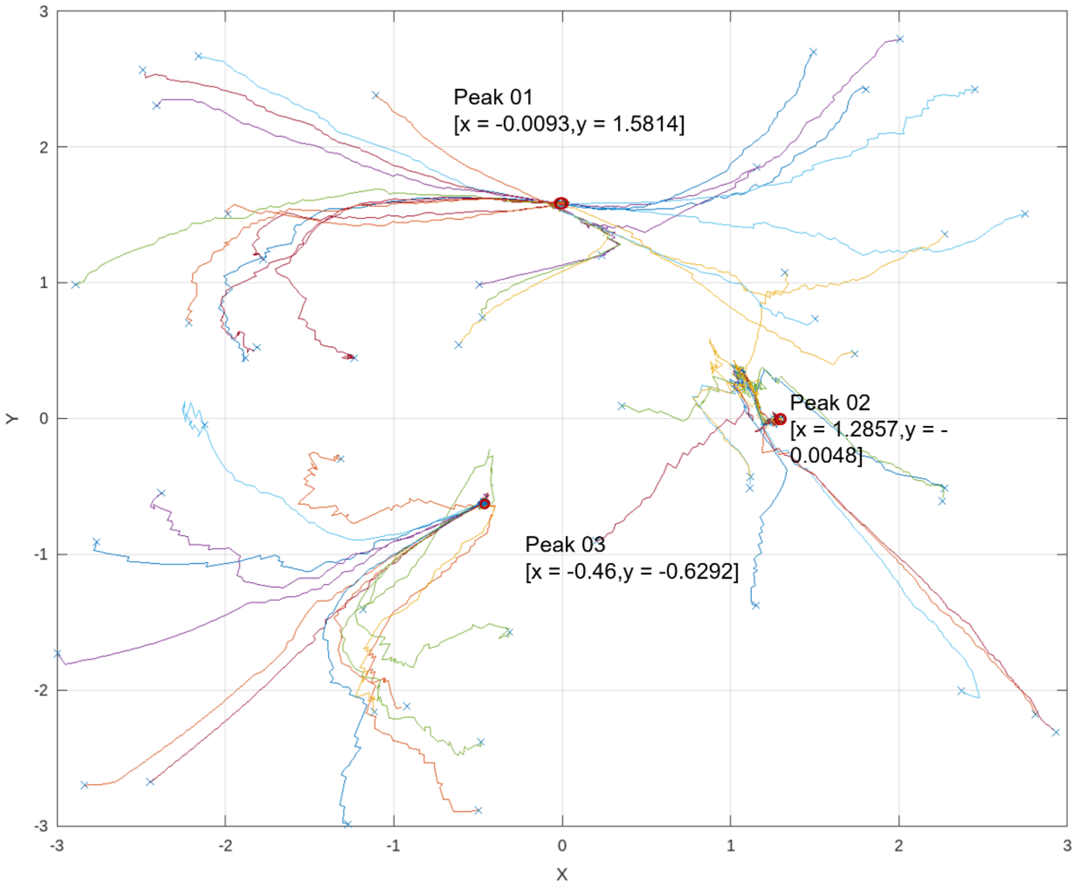


Figure 18. GSO Movements

Comparing this Figure 18 and 19 enables a visual analysis of the disparities between the GSO and GPR-GSO methodologies. The variances in glowworm distribution and movement patterns provide insights into the potential enhancements in GSO algorithm

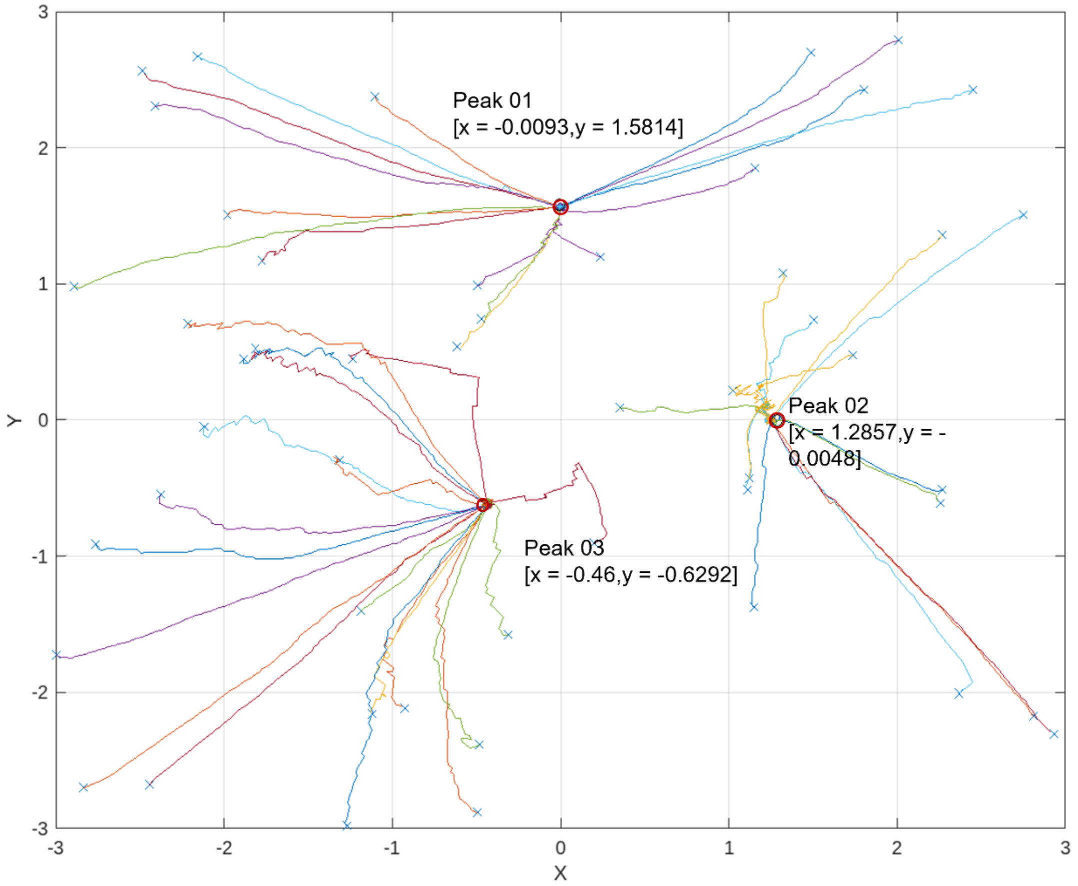


Figure 19. GPR-GSO movements

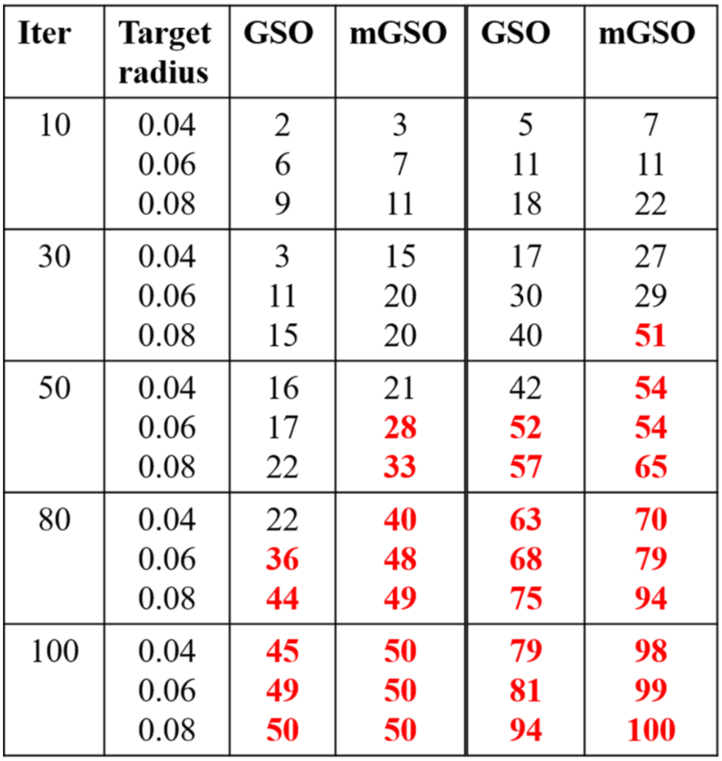
performance. Notably, the luciferin values derived from neighboring glowworms' estimated peaks influence the distribution of glowworms, resulting in a greater number of glowworms gravitating towards the peak with the highest value, thereby delaying convergence. These figures offer valuable insights for comprehending the advantages and distinctions associated with the implementation of the GPR-GSO algorithm in peak optimization tasks. It is worth noting that the GP model reaches a high level of accuracy below 0.005 around the 12th to 13th iteration, and the Weighted Random Vector (WRV) provides more precise directional information, hence a diversion can be observed around the region where probability to move to either of the peaks nearly equal in terms of distance. Additionally, the way glowworms distribute themselves is purely a consequence of the WRV based on their initial deployment distance from the estimated peaks, which plays a significant role in enhancing the algorithm's efficiency. Notably, the deployment distance from the estimated peaks influences the redistribution of glowworms, leading to more effective and efficient convergence. These figures provide valuable information for understanding the benefits and distinctions of employing the GPR-GSO algorithm in source localization tasks.

4.3.2 Number of Iterations to Convergence

The test was run with 50 glowworms both for GSO and GPR-GSO untill all the glowworms converge. The GSO algorithm took about 150 iterations for all glowworms to converge, and GPR-GSO took about 119 iterations for all 50 glowworms to converge. It also observed that as early as 38th iteration, convergence starts taking place in GPR-GSO. In contrast first convergence occurs at 58th iteration in GSO. This analysis shows us on an average how many iterations were required for 50 glowworms to reach within the 0.1 diameter of the respective peaks. Here too, GPR-GSO outperfoms GSO by an average of 23 iterations or it is 17.69 % faster than GSO. Mean and standard deviation values are listed in Table 1.

	GSO	GPR-GSO
Mean	92.6800	69.6200
Std	18.8749	26.2538

Table 1. Mean and Standard Deviation of the GSO and GPR-GSO



Iter	Target radius	GSO	mGSO	GSO	mGSO
10	0.04 0.06 0.08	2 6 9	3 7 11	5 11 18	7 11 22
30	0.04 0.06 0.08	3 11 15	15 20 20	17 30 40	27 29 **51**
50	0.04 0.06 0.08	16 17 22	21 **28** **33**	42 **52** **57**	**54** **54** **65**
80	0.04 0.06 0.08	22 **36** **44**	**40** **48** **49**	**63** **68** **75**	**70** **79** **94**
100	0.04 0.06 0.08	**45** **49** **50**	**50** **50** **50**	**79** **81** **94**	**98** **99** **100**

Table 2. Target Radius of GSO and mGSO

Similarly, the convergence at distance .04, .06, .08 was also evaluated for a group of fifty and hundred glowworms group size, respectively. The convergence of more than fifty percent in a group is highlighted in red. Even in case of hundred glowworms, almost fifty of them reached the intended location within 30 iterations whereas for GSO alone, it took 20 steps further to reach the intended location. This again highlights the importance of the global information in path planning of the glowworms that can significantly impact search performance. Here, both the local and global information are weighted equally. However, there is a possibility to leverage the global information to decide the next waypoint that can improve performance, while returning to equal weightage as the distance from intended source is shorter than 10%, in which case the local information will be more reliable.

4.3.3 Path Length

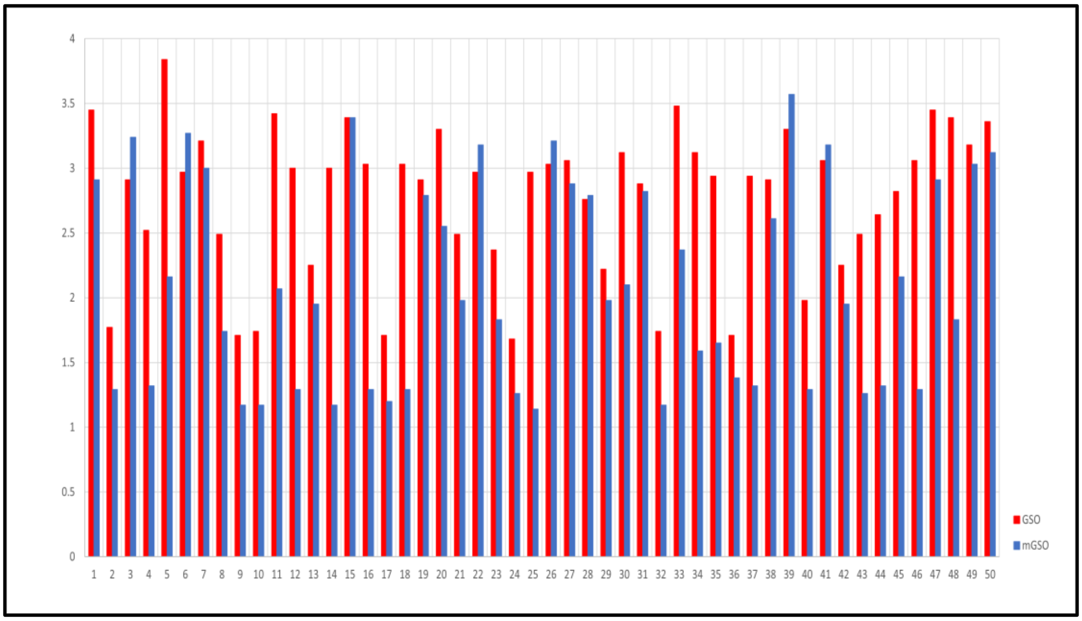


Figure 20. Path Lengths

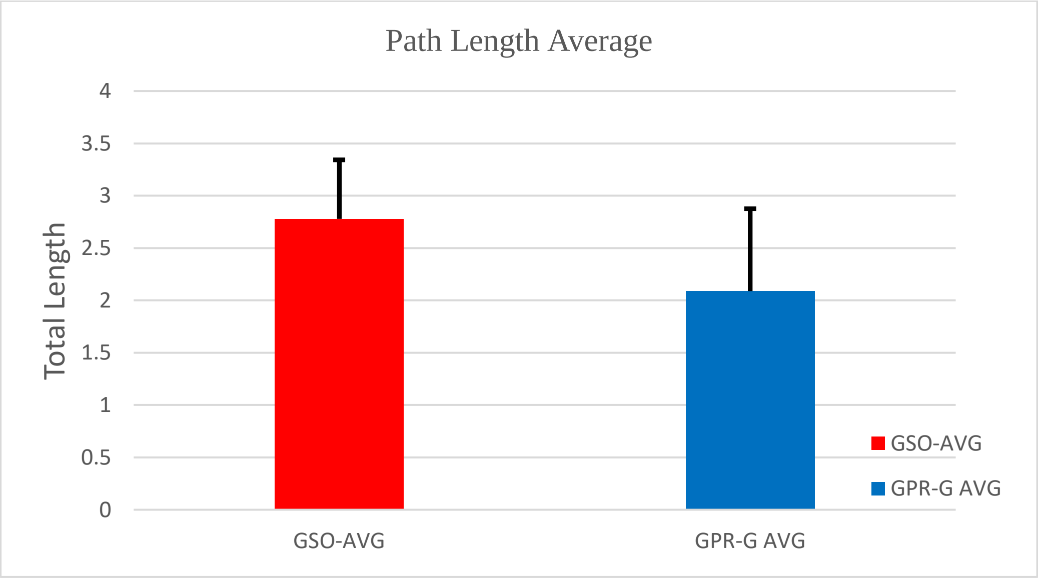


Figure 21 Path Length Average

The path traveled by glowworms was analyzed in both GSO and GPR-GSO. The distance traveled by each glowworm was calculated by multiplying the number of steps taken to converge with the step-length. The glowworms are assumed to converge when they are within 0.01-0.03 units of the emission source for the first time. As mentioned earlier, each step is of uniform length. A step size of 0.03 is defined, which means it travels 0.3-unit length before identifying a new leader to follow. This parameter can be increased or decreased depending how frequently sampling is required. Various step size has been experimented starting from 0.9 to 0.01 to understand the impacts of how frequently it needs to update and identify a new leader, large step size leads to failure of identifying all the peaks and too small step size leads to redundant sampling data, so 0.03 is an optimum step size to consider as in our case the search space is -3 x 3. The glowworms landed far away from the sources traversed almost equal distance in both cases, but glowworms landed in the surroundings and converged quickly in GPR-GSO. The average length of all the glowworms in GPR-GSO the traveled is shorter path (define in magnitude and percentage) and have a higher possibility to move towards the nears peaks from its deployed location. The number of glowworms converged at different iterations which also shows a trend which is significantly better than GSO.

4.3.4 Individual Glowworms traces (path length vs straight line distance)

The ratio of the distance of the deployed locations of the glowworms to their respective peaks over the path length traveled following GSO and GPR-GSO is calculated. The closeness to one defines the straightness of the path, and if the ratio is far lesser than one, then the lengthier the path was taken to reach the goal. This is based on convergence respective to the individual glowworms to their targets, the figure below demonstrates the same.

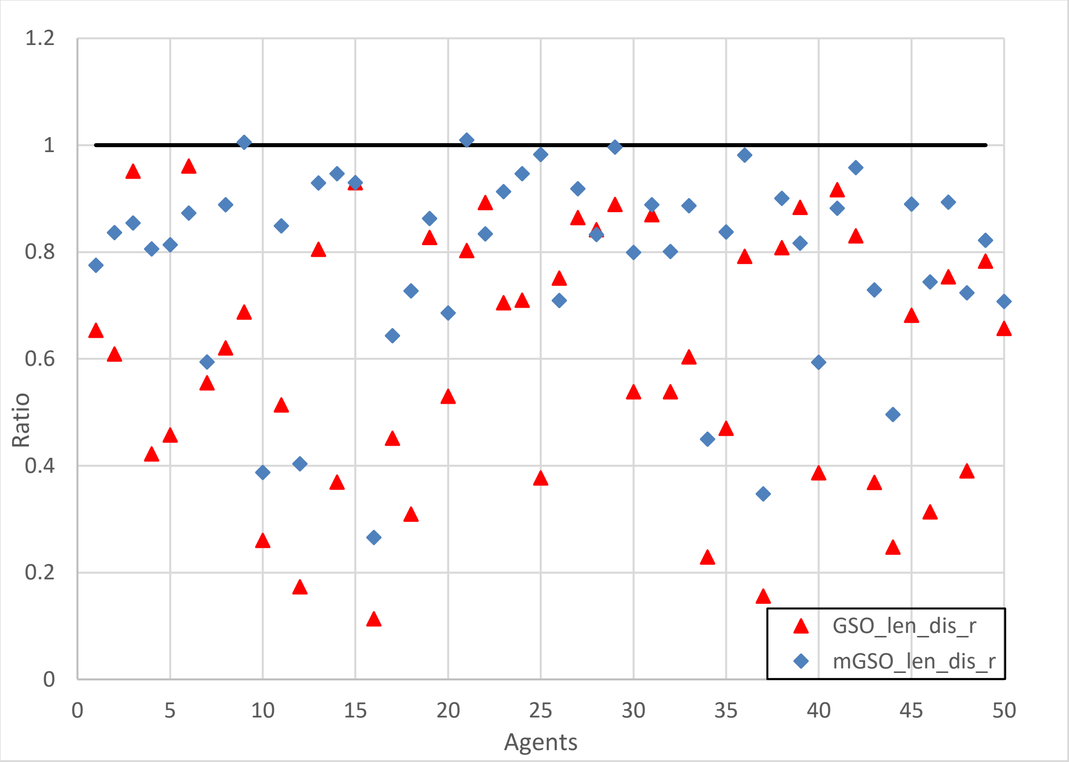


Figure 22. Euclidean Distance to Path-length Ratio

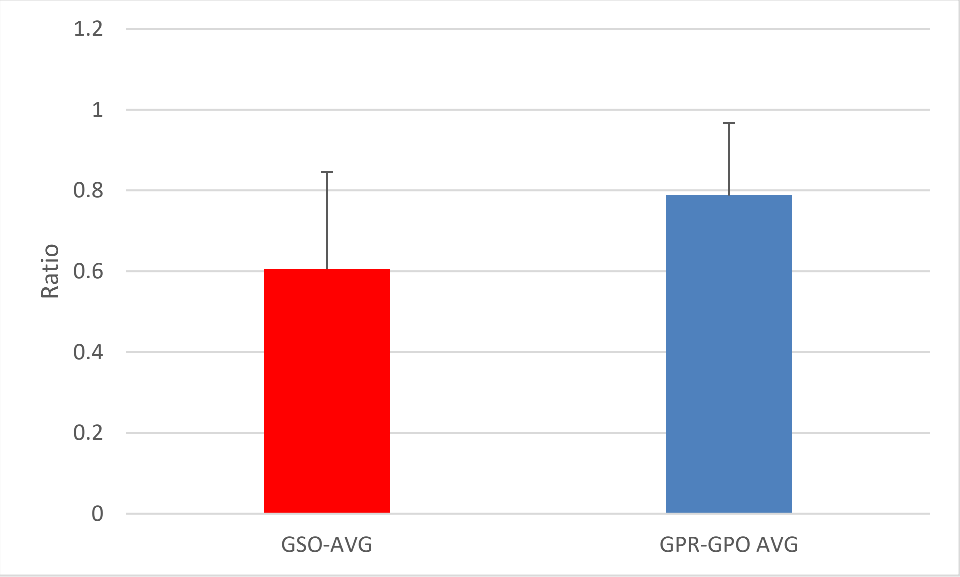


Figure 23 Average of Euclidean Distance to Path-length Ratio

It is clear that the paths traversed by glowworms in GPR-GSO are closer to a straight line, while the paths traversed by glowworms are longer in GSO. There are cases where the glowworms converge to different target sources in GSO and GPR-GSO. Some of such examples are displayed below.

a. Different target cases

The figure on the left demonstrates Glowworm A02's path in GPR-GSO. The figure on the right shows Glowworm A02's path followed in GSO; we can see the glowworm chose to move towards Peak 02 initially.

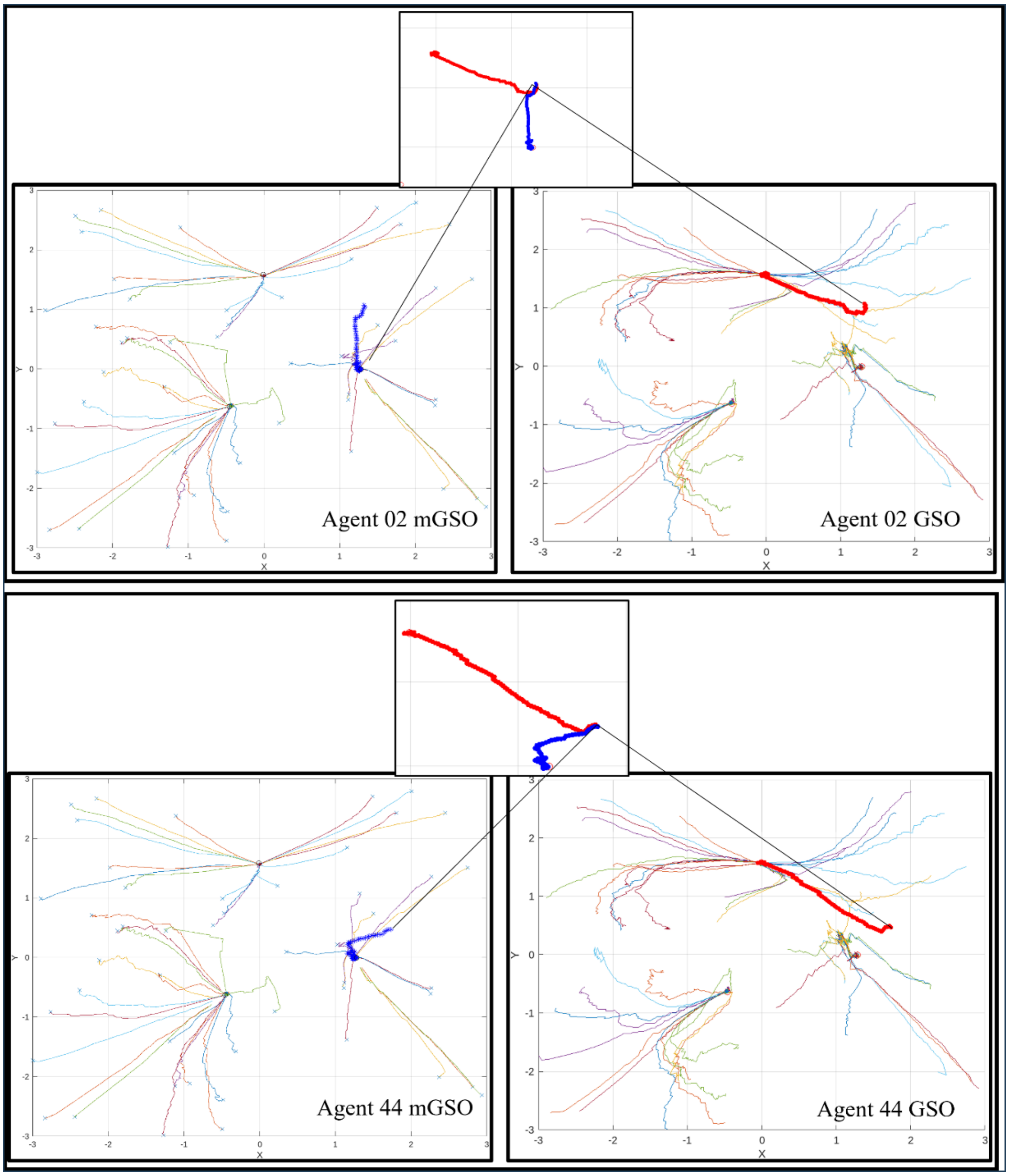


Figure 24. Different Targets

b. Same target cases

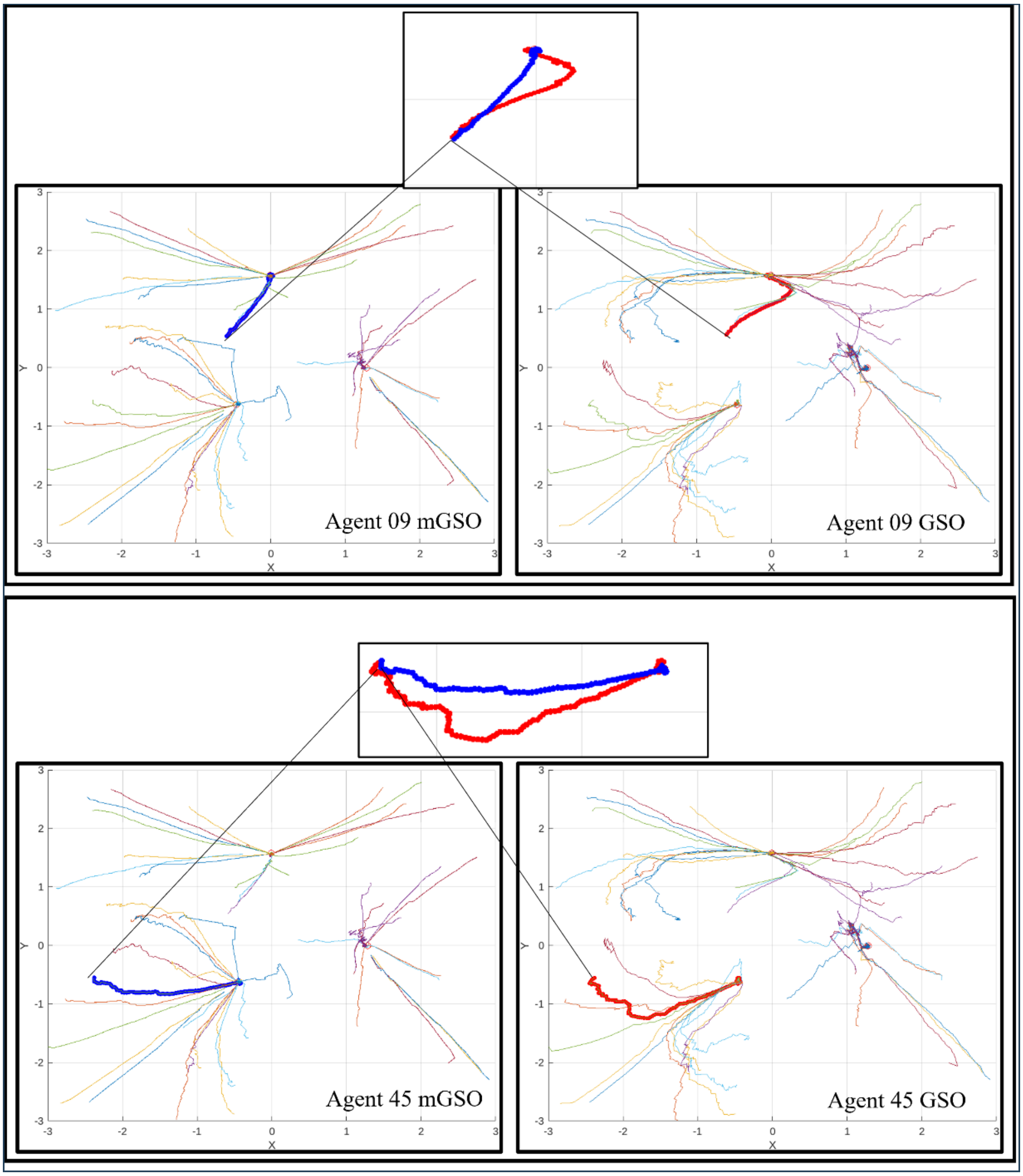


Figure 25. Same Targets

In the case where the glowworm moved towards the same target both in GSO and GPR-GSO, the path followed by GSO is more curvy as it was influenced by the local neighbors alone, in turn it took longer routes to converge, where it is clearly visible when the global information was incorporated, the path was more directed to the intended peak and hence required less exploration, and shorter routes and faster convergence was achieved.

4.3.5 Cumulative vs. Non-cumulative (Number of Iterations)

The GPM needs to be trained using input-output pairs to find the optimal number of data points needed to find the best predicted cost function, which is the multimodal function profile spread across the search space of the glowworms. As swarm size is directly related to number of samples at each iteration, different swarm size was tested to compute the predicted peaks error over different iterations. It is observed that with a swarm size of 100, the error is minimized to 0 after the first iteration, but it also increases the sample data size which makes it computationally heavy. Bringing the swarm size down to 80, the error went down to same percent as 100 s glowworms after 5 iterations. Although there was an increase of error, it established after 10th iterations and stays that way. As the size of the glowworms was reduced to 50, the error was initially high, but it sharply drops to the levels of 100 swarm size after 15 iterations. So, the optimum swarm size was 50 glowworms. This is the case where all the previous data was used to train the GPM. Training data set size is given by $N * i$, where N is the number of glowworms and i is the number of iterations.

The glowworms' location information and their luciferin values constitute the input/output pair $\{[x, y], [z]\}$. For N = 100, at each iteration, we have 100 new data points. Therefore, the number of data points increases as 100, 200, 300, and so on. The other method devised is to only collect the data at current locations, which never drives error to zero. This is because, as the glowworms start moving near to the peaks, the data are less scattered and more concentrated over a small area of the search space and it leads to high deviation in rest of the search space, and hence the predicted model fails to provide useful information to the glowworms. Figure 24 demonstrates the same in all cases of 100, 80, 50 swarm sizes.

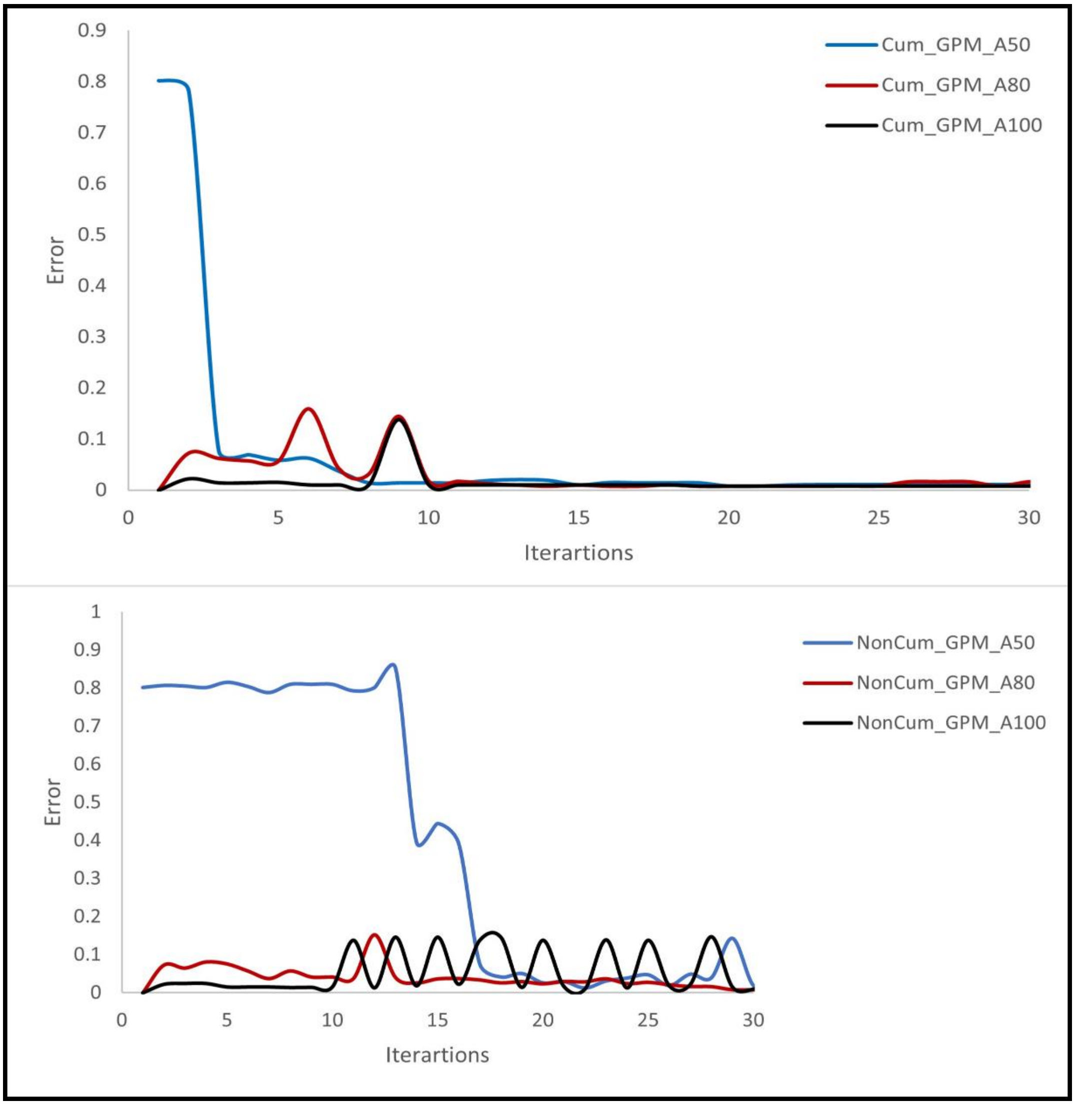


Figure 26. Cumulative and non-cumulative swarm sizes

Estimated Peaks Error

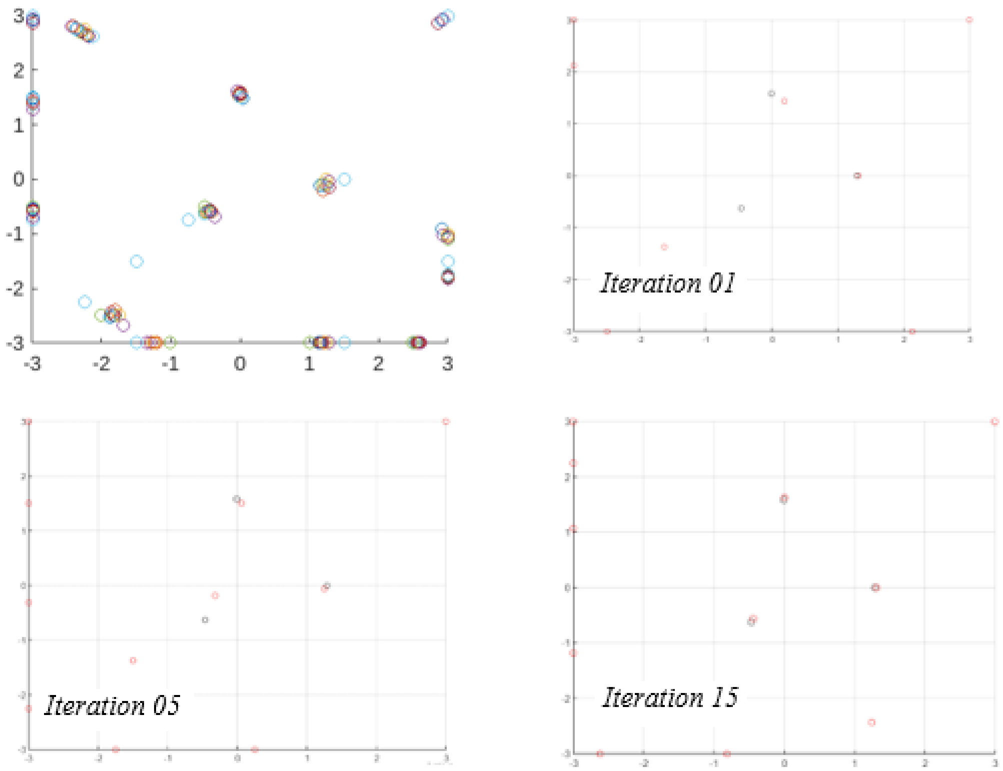


Figure 27. Estimated peaks at different resolutions and iterations before Local Max Filter is applied to the GPM Derived Cost Function

Figure 25a represents all the estimated peaks at different resolution and iterations before local max filter is applied to the GPM derived cost function. The figure on the right demonstrates the varying estimated peaks as the GPM gets trained with each iteration, we have observed that estimation errors is down to 0.001 by the 15 iterations, which also gives an idea for approximate sample size needed to train the GPM.

4.4 Reduction in Deviation using GPR-GSO

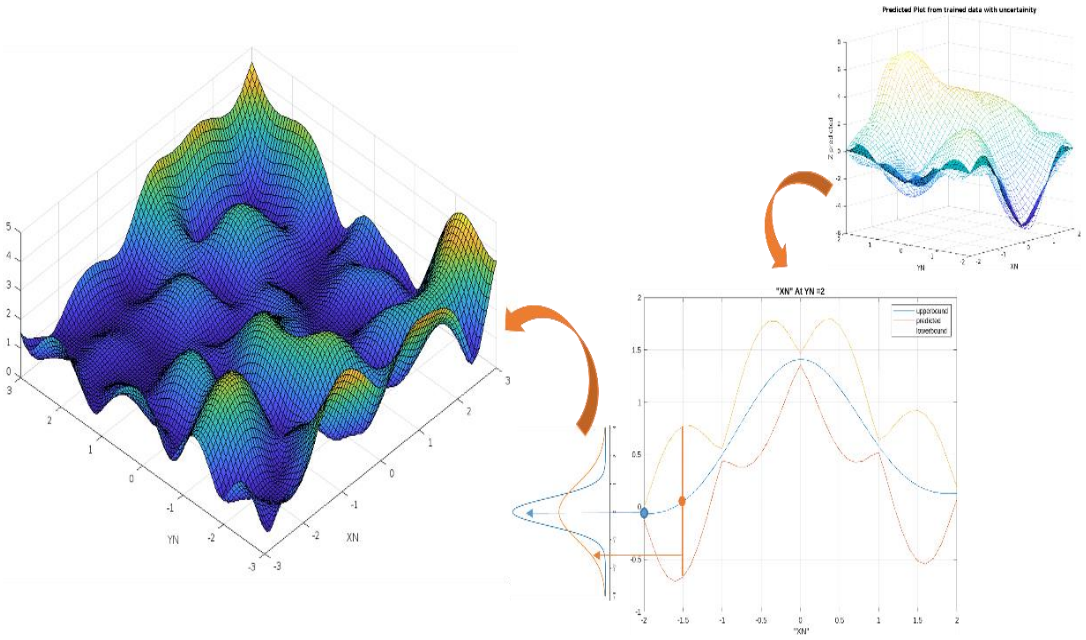


Figure 28. Reduction in Deviation using GPR-GSO

In the context of deriving the deviation as a cost function, the GPM is trained using a set of train data pairs. The GPM learns the underlying patterns and correlations in the data, allowing it to make predictions for unseen inputs. These predictions are not only point estimates but also come with associated uncertainties. The GPM provides not only the predicted mean values but also the variance or uncertainty associated with each prediction.

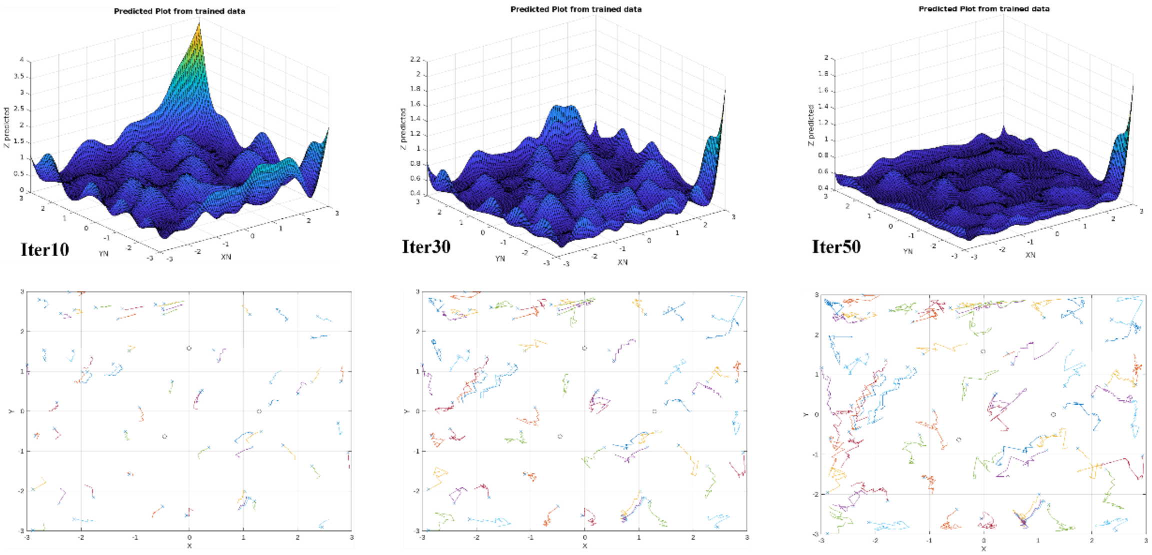


Figure 29. Iterations 10, 30, and 50

The deviation can be obtained by considering the range within which the predicted values are likely to fall with a 95% level of confidence. By calculating the width of these prediction intervals, the 95% deviation can be determined. This deviation serves as a measure of the spread or uncertainty in the predictions and can be used as a cost function in various optimization tasks. This bound is converted as a deviation matrix for the search space which is used as a cost function in the GPR-GSO. Initially the bounds are small around presence of glowworms. As the iterations increase, the glowworms try to move to peaks which represents the high deviations. As the iterations progress, the cost function is updated as the deviations are changed as the glowworms are displaced.

CHAPTER V

CONCLUSIONS AND FUTURE WORK

This thesis presented techniques for search and rescue at disaster sites using autonomous robotic swarms, with a specific focus on the task of multiple emission source localization. The GPR-GSO algorithm, which combines properties of the original GSO algorithm and Gaussian Process Regression (GPR) based data-driven predictive modeling, was developed to improve the search efficiency of robotic swarms. The problem formulation, methods, and illustrative numerical simulations were presented. Results from a comparative analysis were presented to show that the GPR-GSO algorithm exceeds the performance of the benchmark GSO algorithm on evaluation metrics of swarm size, search completion time, and travel distance. Future work includes further experimental evaluation of the GPR-GSO algorithm by varying parameters like swarm size, initial swarm deployment distribution, number of emission sources and their placements, and other algorithmic parameters of GSO and GPR methods. A comparative analysis with other benchmark swarm robotic search algorithms can also be conducted in the future. This work can also be extended by the physical implementation of the GPR-GSO algorithm in a swarm of mobile robots for applications in two- and three-dimensional environments.

REFERENCES

[1] Couceiro, Micael S., et al. “Benchmark of Swarm Robotics Distributed Techniques in a Search Task.” Robotics and Autonomous Systems, vol. 62, no. 2, Feb. 2014, pp. 200–13.

[2] McGill, K. and Taylor, S. 2011. Robot algorithms for localization of multiple emission sources. ACM Comput.Surv. 43, 3, Article 15 (April 2011), 25 pages.

[3] Senanayake, Madhubhashi, et al. “Search and Tracking Algorithms for Swarms of Robots: A Survey.” Robotics and Autonomous Systems, vol. 75, Jan. 2016, pp. 422–34.

[4] Krishnanand, K.N., Ghose, D. Glowworm swarm optimization for simultaneous capture of multiple local optima of multimodal functions. Swarm Intell 3, 87–124 (2009).

[5] Krishnanand, K. N., and D. Ghose. “Theoretical Foundations for Rendezvous of Glowworm-Inspired Glowworm Swarms at Multiple Locations.” Robotics and Autonomous Systems, vol. 56, no. 7, July 2008, pp. 549–69.,

[6] Krishnanand, K. N., and D. Ghose. “Detection of Multiple Source Locations Using a Glowworm Metaphor with Applications to Collective Robotics.” Proceedings 2005 IEEE Swarm Intelligence Symposium, 2005. SIS 2005., IEEE, 2005, pp. 84–91.

[7] Krishnanand, K. N., et al. “Glowworm-Inspired Robot Swarm for Simultaneous Taxis towards Multiple Radiation Sources.” Proceedings 2006 IEEE International Conference on Robotics and Automation, 2006. ICRA 2006., IEEE, 2006, pp. 958–63.

[8] Ghassemi, Payam, and Souma Chowdhury. “An Extended Bayesian Optimization Approach to Decentralized Swarm Robotic Search.” Journal of Computing and Information Science in Engineering, vol. 20, no. 5, Oct. 2020, p. 051003.

[9] Ghassemi, P, & Chowdhury, S. "Decentralized Informative Path Planning with Balanced Exploration-Exploitation for Swarm Robotic Search." Proceedings of the ASME 2019 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference. Volume 1: 39th Computers and Information in Engineering Conference. Anaheim, California, USA. August 18–21, 2019. V001T02A058. ASME.

[10] Ghosh, Sumit. “Understanding Complex, Real-World Systems through Asynchronous, Distributed Decision-Making Algorithms.” Journal of Systems and Software, vol. 58, no. 2, Sept. 2001, pp. 153–67.

[11] Srinivas, Niranjan, et al. “Information-Theoretic Regret Bounds for Gaussian Process Optimization in the Bandit Setting.” IEEE Transactions on Information Theory, vol. 58, no. 5, May 2012, pp. 3250–65.

[12] G. Beni, From swarm intelligence to swarm robotics, in: E. Şahin, W.M. Spears (Eds.), Swarm Robotics, in: Lecture Notes in Computer Science, vol. 3342, Springer, Berlin Heidelberg, 2005, pp. 1–9.

[13] A.J.C. Sharkey, Swarm robotics and minimalism, Connect. Sci. 19 (3) (2007)245–260.

[14] G. Beni, J. Wang, Swarm intelligence in cellular robotic systems, in: Proceed. NATO Advanced Workshop on Robots and Biological Systems, 1989.

[15] R. Eberhart, J. Kennedy, A new optimizer using particle swarm theory, in: Proceedings of the Sixth International Symposium on Micro Machine and Human Science, 1995. MHS'95, 1995, pp. 39–43.

[16] J. Kennedy, R. Eberhart, Particle swarm optimization, in: IEEE International Conference on Neural Networks, 1995. Proceedings, vol. 4, 1995, pp. 1942–1948.

[17] R. Eberhart, Y. Shi, Particle swarm optimization: developments, applications and resources, in: Proceedings of the 2001 Congress on Evolutionary Computation, 2001, vol. 1, 2001, pp. 81–86.

[18] X. Li, Adaptively choosing neighbourhood bests using species in a particle swarm optimizer for multimodal function optimization, in: K. Deb (Ed.), Genetic and Evolutionary Computation, GECCO 2004, in: Lecture Notes in Computer Science, vol. 3102, Springer, Berlin Heidelberg, 2004, pp. 105–116.

[19] R. Brits, A.P. Engelbrecht, F.V.D. Bergh, A niching particle swarm optimizer, in: Proceedings of the Conference on Simulated Evolution and Learning, 2002, pp. 692–696.

[20] S. Yang, C. Li, A clustering particle swarm optimizer for locating and tracking multiple optima in dynamic environments, IEEE Trans. Evol. Comput. 14 (6) (2010) 959–974.

[21] K. Passino, Biomimicry of bacterial foraging for distributed optimization and control, IEEE Control Syst. Mag. 22 (3) (2002) 52–67.

[22] Marques, L., Nunes, U., And De Almeida, A. T. 2006. Particle swarm-based olfactory guided search. Auton. Robot. 20, 277–287.

[23] K. Derr, M. Manic, Multi-robot, multi-target particle swarm optimization search in noisy wireless environments, in: 2nd Conference on Human System Interactions, 2009. HSI'09, 2009, pp. 81–86.

[24] D. Karaboga, An idea based on honey bee swarm for numerical optimization, Tech. Rep. TR06, Erciyes University, Engineering Faculty, Computer Engineering Department, 2005, Oct.

[25] A. Jevtić, P. Gazi, D. Andina, M. Jamshidi, Building a swarm of robotic bees, in: World Automation Congress (WAC), 2010, 2010, pp. 1–6.

[26] A. Jevtić, A. Gutierrez, D. Andina, M. Jamshidi, Distributed bees algorithm for task allocation in swarm of robots, IEEE Syst. J. 6 (2) (2012) 296–304.

[27] Pang, s. And farrell, J. A. 2006. Chemical plume source localization. IEEE Trans. Syst.Man Cybernet.—Part B: Cybernet. 36, 5, 1068–1080.

[28] OLFATI-SABER, R. 2006. Flocking for multi-agent dynamic systems: Algorithms and theory. IEEE Trans. Aut. Contr. 51, 3, 401–420.

[29] OLFATI-SABER, R. 2007. Distributed tracking for mobile sensor networks with information-driven mobility. In Proceedings of the American Control Conference. 4606–4612.

[30] Qun Meng,Songhao Wang, Szu Hui Ng(2021) Combined Global and Local Search for Optimization with Gaussian Process Models. Informs Journal on Computing 34(1):622-637.

[31] Gaussian Process Models Simple Machine Learning Models Capable of Modelling Complex Behaviours

[32] Berns, F., Hüwel, J. & Beecks, C. Automated Model Inference for Gaussian Processes: An Overview of State-of-the-Art Methods and Algorithms. SN COMPUT. SCI. 3, 300 (2022)

[33] Havenstrøm, Simen Theie, et al. “Deep Reinforcement Learning Controller for 3D Path Following and Collision Avoidance by Autonomous Underwater Vehicles.” Frontiers in Robotics and AI, vol. 7, Jan. 2021, p. 566037

[34] Odonkor, Philip, et al. “Distributed Operation of Collaborating Unmanned Aerial Vehicles for Time-Sensitive Oil Spill Mapping.” Swarm and Evolutionary Computation, vol. 46, May 2019, pp. 52–68.

[35] Kennedy, J., and R. Eberhart. “Particle Swarm Optimization.” Proceedings of ICNN’95 - International Conference on Neural Networks, vol. 4, IEEE, 1995, pp. 1942–48.

[36] Song, D., Kim,C.-Y.,andYi,J.,2012,“Simultaneous Localization of Multiple Unknown and Transient Radio Sources Using a Mobile Robot,” IEEE Trans. Rob.,28(3),pp.668–680.

[37] Mondada, F., Bonani, M., Raemy, X., Pugh,J.,Cianci,C.,Klaptocz,A., Magnenat,S.,Zufferey,J.-C., Floreano,D.,and Martinoli, A.,2009,“The E-Puck,a Robot Designed for Education in Engineering,” Proceedings of the 9th Conference on Autonomous Robot Systems and Competitions,Vol.1, Castelo Branco, Portugal, May 7, pp.59–65.

[38] Lei, Bowen, et al. “Bayesian Optimization with Adaptive Surrogate Models for Automated Experimental Design.” NPI Computational Materials, vol. 7, no. 1, Dec. 2021, p. 194. ef)

[39] Wang, J. 2020. “An Intuitive Tutorial to Gaussian Processes Regression.,” Tut ArXiv. /abs/2009.10862

[40] Automatic Model Construction with Gaussian Processes David Kristjanson Duvenaud University of Cambridge.

[41] C. E. Rasmussen & C. K. I. Williams, Gaussian Processes for Machine

Learning, the MIT Press, 2006, ISBN 026218253X. c 2006 Massachusetts Institute of Technology. Gaussian Processes for Machine Learning.

[42] H. Sit. "A Quick Guide to Understanding Gaussian Process Regression (GPR) and Using scikit-learn's GPR Package." Towards Data Science, Jun 19, 2019. Online. Accessed: July 23, 2023.

[43] Akifumi Wachi University of Tokyo wachi@space.t.u-tokyo.ac.jp Yanan Sui and Yisong Yue Calfornia Institute of Technology fysui, yyueg@caltech.edu Masahiro Ono Jet Propulsion Laboratory, California Institute of Technology Safe Exploration and Optimization of Constrained MDPs using Gaussian Processes

[44] C. Brecque. "The Intuitions behind Bayesian Optimization with Gaussian Processes." Towards Data Science, Sep 26, 2018. [Online]. https://towardsdatascience.com/the-intuitions-behind-bayesian-optimization-with-gaussian-processes. Accessed: July 23, 2022.

[45] Tiger, Mattias, and Fredrik Heintz. "Gaussian Process Based Motion Pattern Recognition with Sequential Local Models." 2018 IEEE Intelligent Vehicles Symposium (IV), IEEE, 2018, pp. 1143–49.

[46] Gaussian Process Regression Flow for Analysis of Motion Trajectories Kihwan Kim Dongryeol Lee Irfan Essa fkihwan23, dongryel, irfang@cc.gatech.edu Georgia Institute of Technology, Atlanta, GA, USA

[47] Arulampalam, M. S., Maskell, S., Gordon, N., And Clapp, T. 2002. A tutorial on particle filters for online nonlinear/non-Gaussian Bayesian tracking. IEEE Trans. Signal Proc. 50, 2, 174–188.

[48] Bachmayer, R. And Leonard, N. E. 2002. Vehicle networks for gradient descent in a sampled environment. In Proceedings of the 41st IEEE Conference on Decision and Control. 1–6.

[49] Barlow, G. J., Choong, K. O., and Smith, S. F. 2008. Evolving cooperative control on sparsely distributed tasks for UAV teams without global communication. In Proceedings of the Genetic and Evolutionary Computation Conference. 177–184.

[50] Baudoin, Y., Acheroy, M., Piette, M., and Salmon J. P. 1999. Humanitarian demining and robotics. J. Mine Act. Foc. Mach. Assist. Demin. 3, 2, 433–435.

[51] Bell, W. J. and Tobin, T. R. 1982. Chemo-orientation. Biolog. Rev. Cambridge Phil. Soc. 57, 219–260.

[52] Borah, D. and Balagopal, A. 2004. Localization and tracking of multiple near-field sources using randomly distributed sensors. In Proceedings of the 38th Asilomar Conference on Signals, Systems, and Computers. 1323–1327.

[53] Cortes, J. and Bullo, F. 2009. Non smooth coordination and geometric optimization via distributed dynamical systems. SIAM Rev. 51, 1, 163–189.

[54] Cortes, J.,Martinez, S.,Karatax, T., and Bullo, F. 2004. Coverage control for mobile sensing networks. IEEE Trans. Robot. Automat. 20, 2, 243–255.

[55] Cui, X., Hardin, C. T., Ragade, R. K., and Elmaghraby, A. S. 2004a. A swarm approach for emission sources localization. In Proceedings of the 16th International Conference on Tools with Artificial Intelligence. 424–430.

[56] S.J. Benkoski, M.G. Monticino, J.R. Weisinger, A survey of the search theory literature, Naval Res. Logist. 31 (4) (1991) 469–494.

[57] M. Dorigo, E. Sahin, Swarm robotics—special issue, Auton. Robots 17 (2004) 111–113.

[58] J. Pugh, A. Martinoli, Multi-robot learning with particle swarm optimization, in: Proceedings of the Fifth International Joint Conference on Autonomous Glowworms and MultiGlowworm Systems, 2006.

[59] O. Michel, Webots: professional mobile robot simulation, J. Adv. Robot. Syst. 1(2004) 39–42.

[60] W. Jatmiko, K. Sekiyama, T. Fukuda, Modified particle swarm robotic odor source localization in dynamic environments, Int. J. Intell. Control Syst. 11 (2) (2006) 176–184.

[61] Peaks Function - MATLAB Peaks. https://www.mathworks.com/help/matlab/ref/peaks.html. Accessed 23 July 2023.

[62] Control Random Number Generator - MATLAB Rng. https://www.mathworks.com/help/matlab/ref/rng.html. Accessed 23 July 2023.

VITA

Payal Nandi

Education

- **Old Dominion University** Norfolk, VA
  Master of Science in Mechanical Engineering; GPA: (3.7/4) *Jan. 2021 – Present*
- **Hindustan University** Chennai, India
  Bachelor of Engineering in Aerospace; CGPA: (3.7/4) *Aug. 2012 – July. 2016*

Industrial Experience

- **Motion Planning for Collaborative Robotic Arm** Hampton, VA
  Laser and Plasma Technologies: Intern *AUG 2022 - DEC 2022*
  - **Frame work development**: Developed an architecture to reconstruct the 3D surfaces using point cloud data obtained from LIDAR and motion planning for a Sawyer cobot arm (7DOF) using a waypoint approach.
  - **Implementation**: ROS served as the main framework, MQTT as the communication interface, MoveIt for motion planning, execution, visualization, Gazebo for simulating the robotic environment and Octomap to reconstruct a 3D digital model of the structure using synthetic Lidar data generated with Webots.
  - **Evaluation** : The iso-planar (Cartesian) tool path method was employed to generate waypoints from the octomap surface reconstruction data.

Academic Project

- **Sampling-based path planning approach applied to enhance search efficiency of GSO** Norfolk, VA
  Collaborative Robotics and Adaptive Machines Laboratory LAB:Research Assistance *JAN 2022 - Present*
  Thesis || Guide: Dr.Krishnanand N.Kaipa
  - **Frame work**: Developed an additional informed vector using a probabilistic approach to achieve a cumulative heading vector for improved search efficiency. The framework was demonstrated in a 2D objective space but can be scaled to N dimensions.
  - **Implementation**: Utilized Gaussian Process Models (GPM) to determine a probabilistic continuous function from iterative agent samples and utilized local maxima algorithm to determine the target for the informed vector.
  - **Comparison** : The use of the informed vector enabled agents to reach the target in 30% fewer iterations for a given number of agents, and the efficiency can be further increased with a higher number of agents.
- **Comparative analysis of state estimator of position using Extended Kalman filter and GPM** Norfolk, VA
  Collaborative Robotics and Adaptive Machines Laboratory LAB:Research Assistance *JAN 2022 - AUG 2022*
  Independent Study || Guide: Dr.Sharan Asundi
  - **Develop Algorithm** : Development of the non-linear state estimator and determine minimum GPS ping time interval for power optimization of -ODU's Sealion CubeSat mission- on-board computer.
  - **Implementation**: Extended Kalman Filter (EKF) was implemented in MATLAB, along simulated GPS orbit to dynamical determine the triangulation.
  - **Comparison**: Comparative analysis was tested for one orbit (100 min) and it was determine that GPM provides better state estimation with frequency of 13 min GPS ping, on cost of high computational power. Whereise EKF is more computation efficient with frequency of 16 min GPS ping . Here computational power is based on CubeSat OBC.
- **Path planning for Autonomous Underwater Vehicle** Norfolk,VA
  Underwater Robotics, Course Project *May-July 2021*
  - **Planned**: Conducted a comparative study of A* and RRT algorithms for AUV path planning in both static and dynamic environments using AUV models and implemented algorithms using Matlab functions integrated within a Simulink environment.
  - **Evaluate**: Performance of A* algorithm excels in path planning for static environments, providing an optimal path, conversely, RRT provides feasible path for dynamic environments generated rapidly.
- **Predicting Asteroid Diameter** Norfolk, VA
  Data Science Course Project *Jan-May 2021*
  - **Develop**: Developed an algorithm to predict the asteroid diameter using available orbital elements
  - **Predict**: Predicted the diameter using machine learning based on the available independent variables from the data-set.
  - **Evaluate**: Evaluated the best algorithm among linear regression, elastic regression, k-nearest neighbours, decision tree regression, random forest regression, neural network regression using (R^2) scores
- **Education program of Indian Space Research Organisation:** Bangalore, India
  Summer Internship *Jan-April 2016*
  - **Relative Attitude Estimation for Spacecraft**: Developed Relative Attitude Estimation algorithm for Spacecraft in MATLAB used for Rendezvous and Docking by LARA Method (Linear Algebra Resection Approach) Under education program of Indian Space Research Organisation, Bangalore