ISSN: 0971-1260
Vol-22-NO-30 Special Issue- December-2019

Implementation of Garbage Collection and Recycling Problem Using An Optimized Glowworm Swarm Optimization Algorithm

[1]Anita Sharma, [2]Dr. Kamlesh Lakhwani,
[1,2]Department of Computer Science and Engineering
Lovely Professional University, Phagwara, Punjab, India
[1]anita.19395@lpu.co.in
[2]kamlesh.20980@lpu.co.in

Abstract

This paper represents the Glowworm Swarm Optimization Algorithm which solves the problem of garbage collection by using its mechanism which is originated by the social behavior and the phenomenon of bioluminescent communication of glowworms. The paper presents a solution to the problem of garbage collection by groups of autonomous robots. The algorithm is applied on the robots to manage their movement around the area containing garbage. Using this algorithm number of robots are trained to interact with each other, also finding the multiple optima of multimodal functions attempting to simulate the way a group of animals behave like a single cognitive entity. The simulation is carried in such a manner that the items are collected and then deposited in the desired stations. Finally, experiments are conducted using the benchmark functions to get the best fitness values for the robots. In the end the comparison is carried for GSO with PSO that is designed for the parallel computation of multiple optima.

Keywords:
Glowworm swarm optimization ,Computational Intelligence, multiple optima.

1. Introduction

Computational Intelligence [1], a sub branch of artificial intelligence involves adaption, learning, association, evolution, and fuzzy linguistic to build some intelligent systems. It enables intelligent behaviors in complex and changing environments by studying the adaptive mechanisms. Computational Intelligence, a field of artificial intelligence, is a combination of learning, adaption, evolution, association and fuzzy linguistic to make some intelligent programs. Now as computers have become sophisticated, scientists have become more interested in the possibility of creating autonomous machines, so as to help to solve complex problems faster and effectively. Biology after this influence computing science the early days of computer science.

Copyright © 2019Authors

ISSN: 0971-1260
Vol-22-NO-30 Special Issue- December-2019

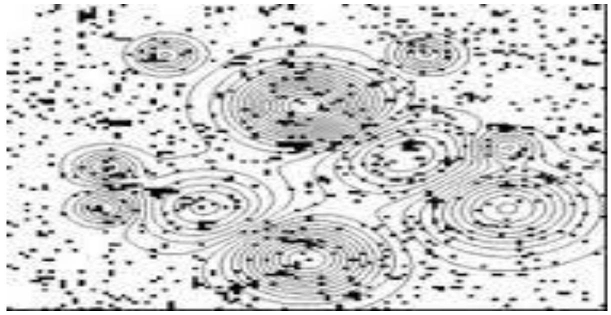


Fig 1: Group of swarms

Swarm Intelligence (SI) [2] , a branch of computational Intelligence[11] that is inspired by a cooperative nature of individuals as such, a group of ant with a queen ant migrate to form a new colony, group of glowworms searching for high light intensity glowworm. Swarm[12] is a group of individuals collaborating to do a common task in order to get the efficient results. Every particle in swarm is a simple agent moving through a search space which is multi- dimensional sampling an objective function at various positions. In a search space the best solution can be represented with a point. Some potential solutions are plotted on the graph and seeded with some initial values. The particle's performance is evaluated with a predefined fitness function having the characteristics of the optimized problem. With time the particles will accelerate towards those with best possible fit values. Within the swarms the structures are relatively quite simple but collectively their behaviors are complex. The behavior of swarm is said to be emergent as it does arise from a rationale choice or from any finalized analysis. The complex behaviors cannot be predicted or deduced from simple behaviors of the individuals. Every agent in the system is not aware of what is happening globally and also does not have any global perspective.

*1.1 **Swarm Robotics***

Swarm robotics[9] consist of a huge group of relatively simple robots that interact and work cooperatively among others in order to solve problems which are not in their own capability. This exhibits interesting features like high degrees of parallelism, duplicity and robustness. These robots are highly adaptive to the changes in environment, solutions are scalable to the problem and swarm size.

1.2 Garbage and recycle collection Problem

The problem of GRC (Garbage Recycling and Collection problem) [3] using an application of swarm intelligence with the help of robots is earlier implemented using many other techniques and also using different methods[9]. The basic idea in this is to train multiple robots to interact with each other so as to behave as a single entity. This work in this is done with the help of two approaches PSO (particle swarm optimization) and ACO (Ant colony optimization). While using PSO star and circle topologies are used along with two prototypes in the experiments[4]. The circle gave the best results between both of these. When both these topologies were mixed, more optimal solution was found. PSO has a specific type of memory that fulfills the requirements of the problem very effectively[10]. This

Copyright © 2019Authors

ISSN: 0971-1260
Vol-22-NO-30 Special Issue- December-2019

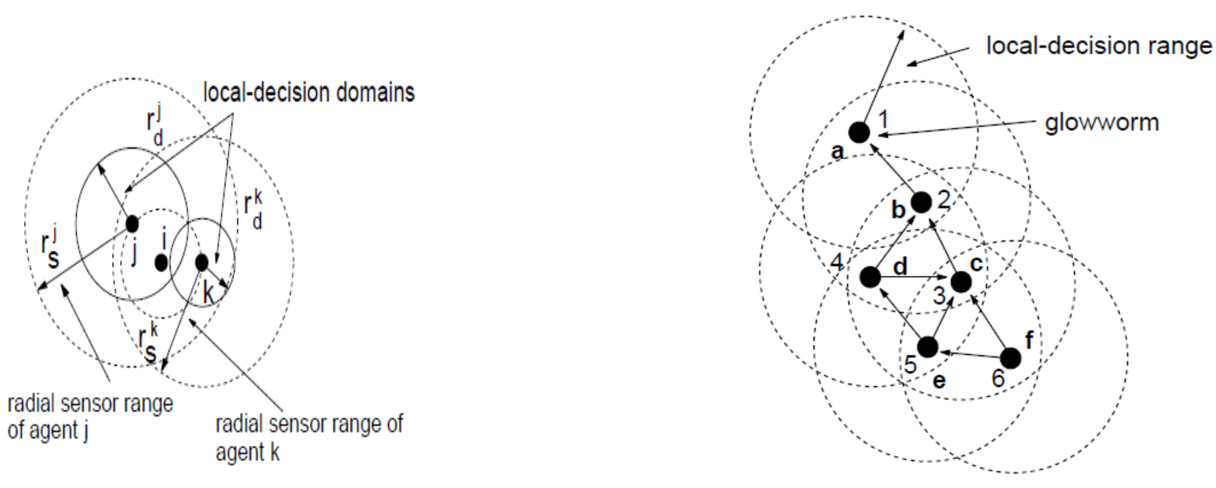
problem was also solved with ACO approach. The main purpose of ACO is to find the optimized path to collect the garbage from various points and then dump them into the garbage station. Both ACO and PSO are then compared to get the best algorithm which gives optimum solution for the problem proposed. Between both of these PSO and ACO, PSO came out to be the best with much higher accuracy. But PSO has many drawbacks which effects the accuracy of this application. As the Glowworm Swarm Optimization has glowworms as its agents, any physical entities are considered to be randomly distributed in a work area. The agents in GSO [4] carry luminescence quantity along with them known as luciferin. Glowworms are considered as agents which emits light, the intensity of which is proportional to the associated luciferin value and have a variable decision range, which is bounded by a circular range.

1.3 Glowworm Swarm Optimization Algorithm

In glowworm algorithm[5] the agents are distributed in an area at a random. These glowworms carry luciferin which is a luminescence quantity. In general glowworms are like fireflies which carries light, the intensity of these is directly proportional to the associated luciferin count.

Every agent i.e a glowworm searches for another glowworm in its local decision domain and then make decisions. After searching for the brightest neighbor the glowworm moves towards the neighbors. With these movements, the glowworms split into subgroups and move to a common area with their taxis behavior leading to the detection of multiple optima from a local optima of an objective function. The glowworm looks for the neighbor with the high intensity and the decision range gets updated dynamically with every iteration.

Also with the low frequency of neighbors the glowworm increases its searching area with increase in radius. At the end the glowworms gather at the multiple optima of the objective function.



(a) **(b)**

Figure 2. (a) r^{k}_{d}< d(i,k) = d(i,j) <rjd<rks<rja. Agent i is in the sensor range of agents j and k (equidistant to both), having different [2] decision-domains. Hence, only j uses the information of i. **(b)** luciferin level of agent in directed graph. Ranking of agents according to the luciferin value.

Copyright © 2019Authors

ISSN: 0971-1260
Vol-22-NO-30 Special Issue- December-2019

Higher the intensity of the light [11], better is the position of a glowworm in the domain. The GSO is capable of carrying concurrent searches by breaking the targets into disjoint sets.

1.3.1 Algorithm Description

Initially all the glowworms are placed at random positions in a work area to make them dispersed properly. Each state "i" of the glowworm at time "t" is described using a set of variables which includes a position in search area, luciferin level and a range in the neighborhood. Initially the agents contain equal amount of light intensity i.e luciferin amount. Every glowworm goes through numerous iterations with multiple stages in it .

Luciferin-update phase

This phase updates the amount[7] of luciferin which depends on the functional value of the glowworm. In this phase some amount of luciferin is added to its previous value depending upon the luciferin value. The value can be any signal like wi-fi strength, temperature etc.

The luciferin count is computed by :

$$li(t) = (1 - \rho)li(t - 1) + \gamma J(xi(t)) \quad (1.1)$$

where, γ refers to the luciferin enhancement constant , ρ refers to the luciferin decay constant ($0 < \rho < 1$)and Jj(t) refers to the objective function at time t for agent's location.

Movement –phase

In this phase each glowworm[8] decides to navigate towards the neighbor with the higher intensity i.e that has the higher luciferin count than its own after using a probabilistic mechanism. The glowworms move towards the brighter glowworm.[13] The glowworm's neighbor needs to meet the following requirements:

- The glowworm should be within the decision domain of glowworm j,
- The luciferin value larger than the glowworm.

Glowworm i navigates toward a neighbor j which comes from Ni((t) with some probability is $p_{ij}(t)$. Probability is calculated by:

$$p_{ij}(t) = \frac{l_j(t)-li(t)}{\sum_{k \in Ni(T)k} l(t)-_i l(t)} \quad (1.2)$$

After moving, glowworm i's location is updated, the update formula is

$$x_i(t+1) = \frac{xi(t)+st*\{x_j(t)-x_i(t)\}}{x_j(t)-x_i(t)} \quad (1.3)$$

where st is the step size.

Copyright © 2019Authors

ISSN: 0971-1260
Vol-22-NO-30 Special Issue- December-2019

Local decision range update rule

In this phase, he position of glowworm is updated and the neighbor range is also updated. The radius of the domain is increased if the number of agents are less in the search space. For defining the area to be searched the radial sensor range is used which describes the position of the glowworm. If the peaks are high the glowworm is in the high intensity area as the peaks depend upon the sensor range.Using all the local optima the global optima is created.

The formula for neighborhood range update is

$$r_i^d(t+1) = \min\{r_s, \max\{0, r_i^d(t)+b\,(n_t-|N_i(t)|)\}\} \qquad (1.4)$$

where and n_t is a parameter used to control the number of neighbors b is a constant parameter. The parameters taken for the are ρ, γ, s_t, β, n_t for the algorithm which we can get them using the experimental method and evolutionary learning.

Figure 4 shows methodology of the proposed model where all the phases of the glowworm Swarm optimization Algorithm are incorporated. The robots carry the garbage and deposit in the respective stations using the principle of GSO.

1.4 Experimental Results

Various graphs are generated to give the fitness of the robots and efficiency of the application with GSO. With the results it is clear that GSO has higher fitness value as compared to previously used techniques. The results are shown for 1000 units of garbage which are collected by the robots [1]. In glowworm swarm optimization the glowworms work with the help of some steps. Some standard benchmark functions are implemented to find the best results and for the comparison with other algorithms. These benchmark functions are minimized as much possible Some of the benchmark functions are given:

Funciton	Function Formula	Domain	Type
Sphere	$f_1(x) = \sum_{i=1}^{n} x_i^2$	[-100,100]	UM
SumSquares	$f_2(x) = \sum_{i=1}^{n} ix_i^2$	[-10,10]	UM
Quartic	$f_3(x) = \sum_{i=1}^{n} ix_i^4 + random[0,1)$	[-1.28,1.28]	UM
Schwefel 2.22	$f_4(x) = \sum_{i=1}^{n} \lvert x_i \rvert + \prod_{i=1}^{n} \lvert x_i \rvert$	[-10,10]	UM
Schwefel 1.2	$f_5(x) = \sum_{i=1}^{n} (\sum_{j=1}^{i} x_j)^2$	[-100,100]	UM
Rosenbrock	$f_6(x) = \sum_{i=1}^{n-1} [100(x_{i+1} - x_i^2)^2 + (x_i - 1)^2]$	[-30,30]	UM
Dixon-Price	$f_7(x) = (x_1 - 1)^2 + \sum_{i=2}^{n} i(2x_i^2 - x_{i-1})^2$	[-10,10]	UM
Griewank	$f_8(x) = \frac{1}{4000} \sum_{i=1}^{n} x_i^2 - \prod_{i=1}^{n} \cos(\frac{x_i}{\sqrt{i}}) + 1$	[-600,600]	MM
Ackley	$f_9(x) = -20 / \exp(\frac{1}{5} \sqrt{\frac{1}{n} \sum_{i=1}^{n} x_i^2}) - \exp(\frac{1}{n} \sum_{i=1}^{n} \cos(2\pi x_i)) + 20 + e$	[-32,32]	MM
Rastrigin	$f_{10}(x) = \sum_{i=1}^{n} [x_i^2 - 10\cos(2\pi x_i) + 10]$	[-5.12,5.12]	MM

*UM: Uni-modal, MM:Multi-modal

Figure 1 Benchmark Functions

Copyright © 2019Authors

ISSN: 0971-1260
Vol-22-NO-30 Special Issue- December-2019

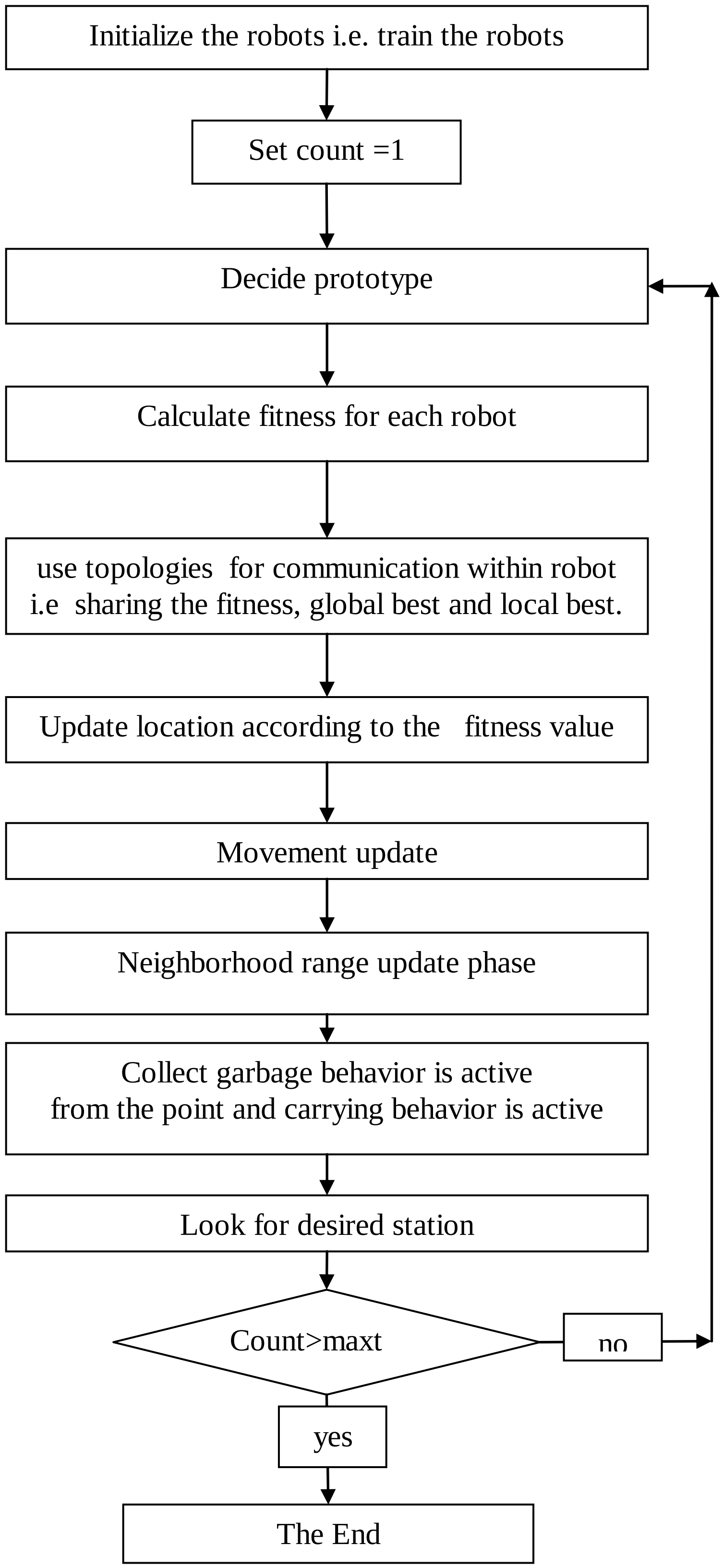


Figure 4 Flowchart of methodology

Copyright © 2019Authors

ISSN: 0971-1260
Vol-22-NO-30 Special Issue- December-2019

The luciferin decay constant p=0.4, the luciferin enhancement constant gamma=0.6, the neighborhood change rate beta=0.08, the neighborhood threshold n_t =5. The step-size s= 0.8. These values are constant and standard as these are calculated from many standard functions. The whole process in GSO algorithm is based upon the intensity of the glowworm. More the intensity of glowworm more glowworms it will attract. The GSO has some predefined constants which are used as it is in the algorithm. The simulations are carried on MATLAB R2014a and the simulation platform used is Windows 8.1, i3 , 4GB memory. The maximum iterations are 1000. This work has been carried out in three prototypes and the results are then compared. 3 cases are made with respect of 3 prototypes having different number of robots in each. Robots behave differently in different environment. Their fitness also changes with respect to the target. The performance of GSO can be analyzed by various graphs.

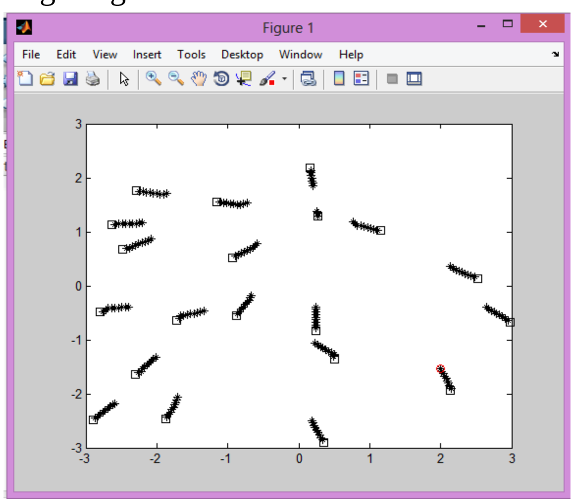
The value for lbest and gbest can be obtained from the GSO and performance can be evaluated.

Case 1: In this case garbage is not added into the arena. Only the agents i.e the glowworms have their definite positions so all will have the same luciferin count means same fitness value.

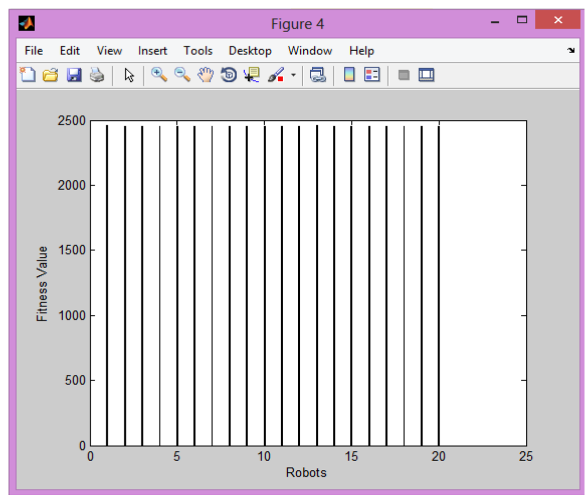
Number of robots =20

Number of iterations = 10

Units of garbage =0



(a)



(b)

Fig: 5 (a) initializing the population with no garbage (b) Fitness graph

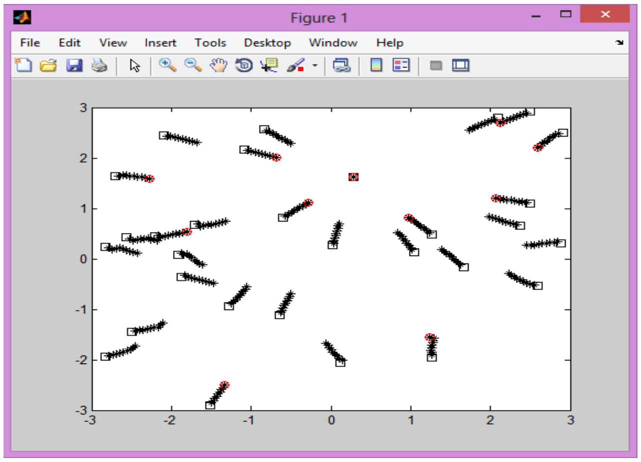
Case 2: In this case the items of garbage are added and the intensity of each glowworm will vary with respect to its distance from the target.

Number of robots = 20

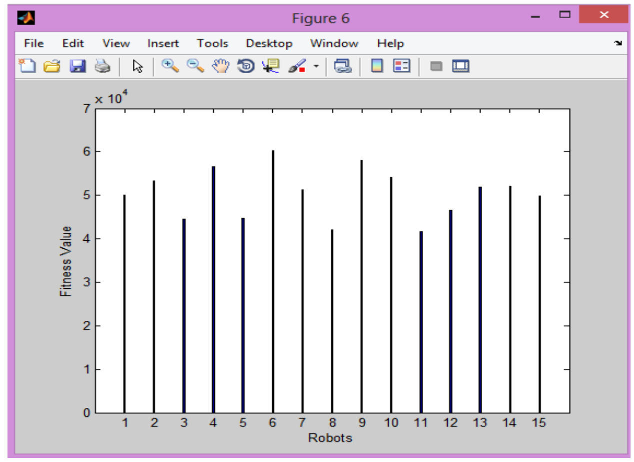
Number of iterations =100

Copyright © 2019Authors

ISSN: 0971-1260
Vol-22-NO-30 Special Issue- December-2019



(a)



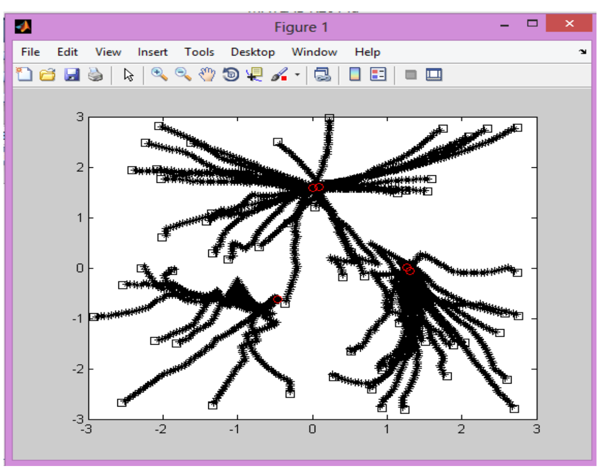
(b)

Fig 6 (a) Robots with garbage (b) **Fitness graph**

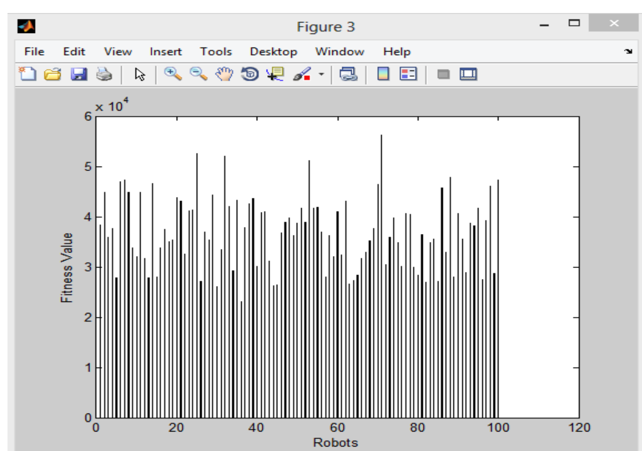
Case 3: Brightest glowworm attract other glowworms i.e robots will move towards the robot with highest fitness value and will reach the target with an optimized path. The glowworms are attracted by the brightest glowworm. These glowworms are agents i.e the robots collecting garbage from the arena. The glowworm nearest to the target will have high luciferin count i.e the robot nearest to the target will have maximum fitness value to which others will be attracted and by this their cooperative behavior will be shown. The red circles are the items of garbage and the near by robot has higher fitness value and is the brightest so all the other agents will move towards that following the optimized path. In this case the collision rate is also taken into account i.e robots should avoid walls and other robots coming on its way.

Number of robots =40

Number of iterations =200



(a)



(b)

Copyright © 2019Authors

ISSN: 0971-1260
Vol-22-NO-30 Special Issue- December-2019

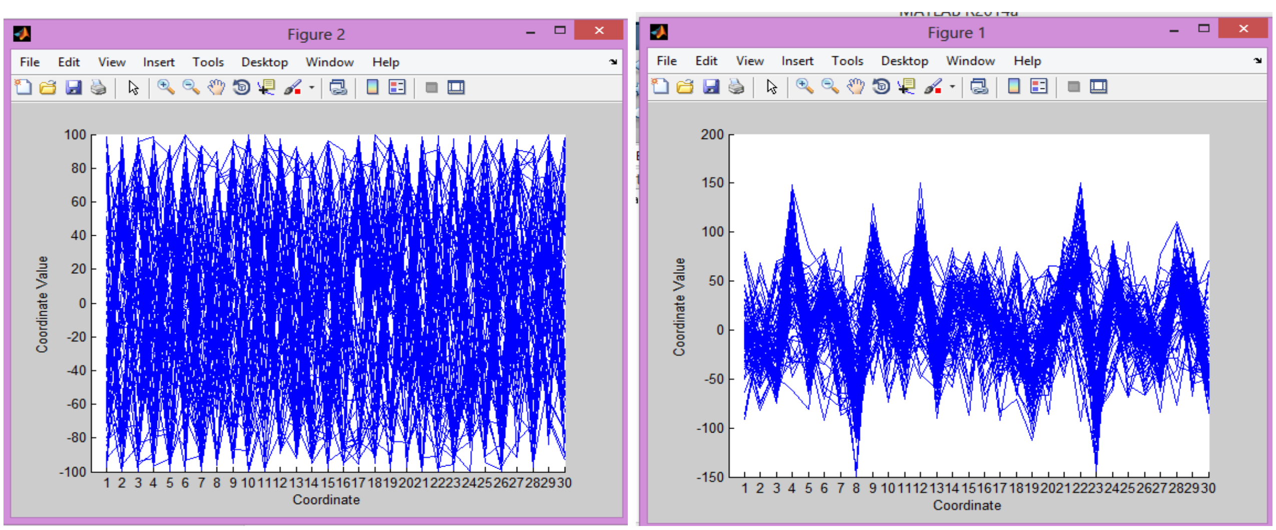
Fig:7 (a) Glowworm attracted to the brightest (b) Fitness of each robot individually

Table 1 Results of cases

Case	Robots	Iterations	Gbest value	Average gbest
1	3	1	7.3861e+04	6.3838e +04
	5	5	7.0032e+04	
	20	10	4.7623e+04	
2	20	100	2.4538e+03	1.4897e+04
	30	150	1.0155e+04	
	35	180	1.0e+04	
3	40	200	0.385e+01	0.3283e +04
	80	400	0.58e+01	
	100	500	0.02e+01	

The gbest values of the three cases are calculated in Table 1 and it can be seen that the value increases when the robot approaches the garbage and the number of iterations are increased. Gbest while approaching to zero is the best for all the functions. Here the experiments are done with one of the benchmark functions.

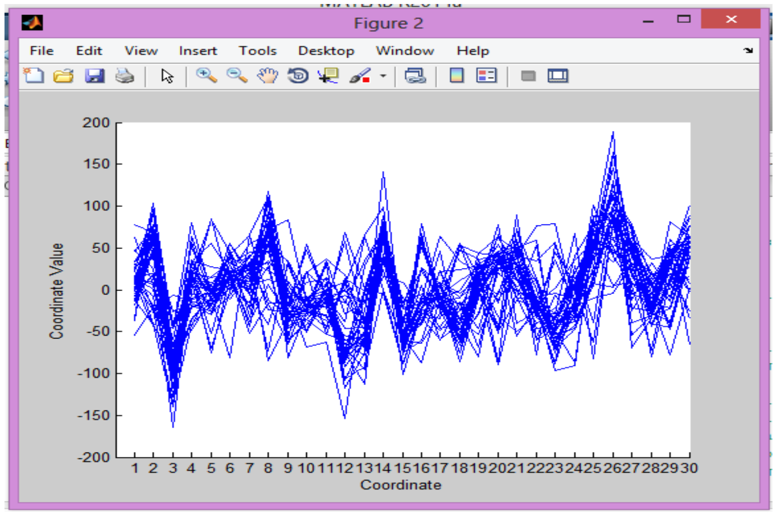
The robots in the beginning are scattered and look for the better fitness in the neighborhood in figure 8 (a). Initially all the robots take different paths to choose the brightest i.e with every iteration the results will be refined. Now in this it can be seen that all the robots take different positions in the first iteration. This blue region shows all the robots taking different path initially. This is because at this point the population is initialized randomly, they find the agent with the best fitness value and are attracted towards it.



(a) (b)

Copyright © 2019Authors

ISSN: 0971-1260
Vol-22-NO-30 Special Issue- December-2019



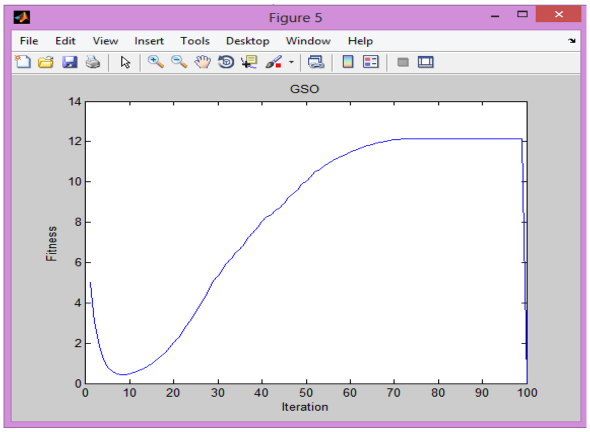
(c)

Fig 8 (a) Working of GRC (b) Converged Path (c) Final Path

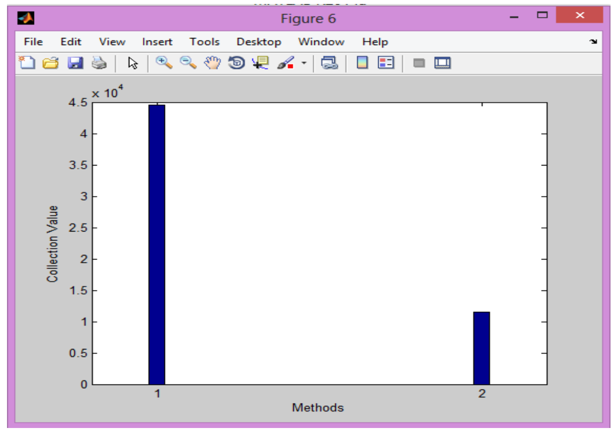
The figure 8(b) shows the path of the agents after some iterations and it can be seen that the path is now more refined. The peaks in the figure show the target i.e the garbage which is collected by the agents and dumped. The peaks also show the best fitness value among all the agents. Here, coordinate and coordinate value is plotted which gives different points in the arena where the application of GRC is working. The robots still take many paths i.e they search for the best possible path to reach the target by GSO.in the end an optimized path is obtained and also the time consumption is less. Figure 8 (c) shows the final output of GRC with the help of GSO. The agents looked for the targets i.e the garbage at different points and found the best possible path to reach the station. Initially, the glowworms take various paths to reach to the destination. Better results can be found with large number of iterations. From the number of results i.e paths, the agents find the best possible solution to reach the destinations. This path is optimized with application of GSO[12]. The randomly initialized robots are trained first to do the job. In the final graph a simple line is obtained which gives the path followed by the robots to reach to the target.

Copyright © 2019Authors

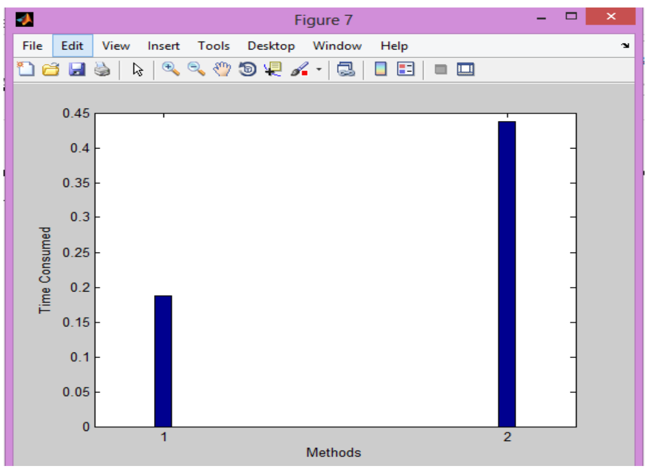
ISSN: 0971-1260
Vol-22-NO-30 Special Issue- December-2019



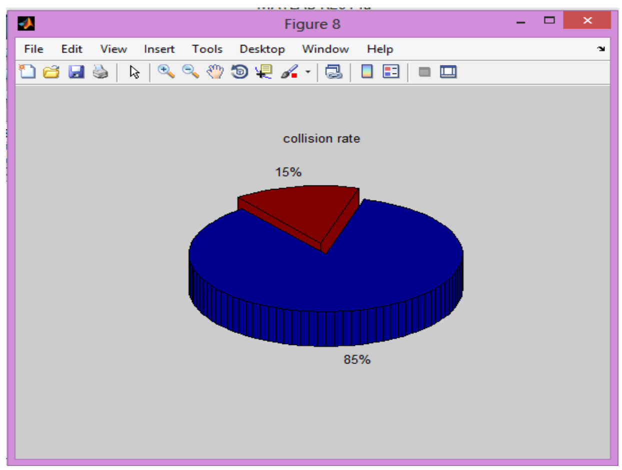
(a)



(b)



(c)



(d)

Fig 9 (a) Fitness graph of GSO (b) Comparison of GSO with PSO for garbage collection (c) Time comparison (d) Collision rate of GSO

The figure 9 (a) shows the graph for GSO for the fitness of the algorithm with the number of iterations. Initially, the fitness value is higher but then dropped suddenly as we initialize the population randomly, the gents are not able to find the brightest agent or the agent with the maximum fitness value. With the increase in number of iterations, the fitness of the algorithm increases as the research is more refined now. In the end, the dropped suddenly as the number of units to be collected are finished. More the number of iterations, more accurate will be the results.

Figure 9 (b), (c) and (d) represents the comparison between the collision rates in the GRC application by two methods: PSO and GSO. The collision rate in the PSO is much higher than the GSO. By collision, all the obstacles including walls and other robots are considered, each robot must not collide with any other robot or wall while traversing the distance towards the destination. By this collision, a lot of time is wasted and optimized results are not achieved. Lesser the number of collisions, less is the time taken to reach the destination and more optimized are the outcomes by using GSO.

Copyright © 2019Authors

ISSN: 0971-1260
Vol-22-NO-30 Special Issue- December-2019

Conclusion

In this study an optimization approach is proposed with the help of GSO, a swarm intelligence technique. This work was previously done with PSO and ACO, the results were compared to find the optimal path. In this algorithm the efficiency can be improved and can also be enhanced further with some modifications. It assures the best possible way with the help of agents i.e robots can communicate with each other and will find an optimum path to reach the target. This work also has a future scope and can also be used in real time environment so as to reduce the cost and time factor. Robots are trained and simulated to collect the garbage and deposit them in respective stations. Also a camera can be used so as to monitor the proper working of the robots.

REFERENCES

[1] A. P. Engelbrecht, Computational Intelligence, University of Pretoria, South Africa: Wiley, 2007.

[2] K. Krishnanand and D. Ghose, "Glowworm swarm based optimazion algorithm for multimodal functions with collective robotics applications," *ISSN,* no. 2, pp. 209-222, 2006.

[3] H. Nie and J. Shen, "Research on Glowworm Swarm Optimization with Ethnic Division," *JOURNAL OF NETWORKS, ,* vol. 9, no. 2, FEBRUARY 2014.

[4] G. Pessin, D. O. Sales, M. A. Dias, R. L. Klaser, D. F. Wolf, J. Ueyama, F. S. Osório and P. A. Vargas, "Swarm intelligence and the quest to solve a garbage and recycling collection problem," *© Springer-Verlag Berlin Heidelberg,* pp. 2311-2325, 2013.

[5] Y. Zhou, G. Zhou, Y. Wang and G. Zhao, "A Glowworm Swarm Optimization Algorithm Based Tribes," *An International Journal,* pp. 537-541, 2013.

[6] A. A.Katkar, "IMPROVEMENT OF SOLID WASTE COLLECTION BY USING OPTIMIZATION TECHNIQUE," *ISSN,* vol. 2, no. 4, pp. 298 - 307, 2012.

[7] M. Brambilla, E. Ferrante and M. Birattari, "Swarm robotics:A review from the swarm engineering," IRIDIA, 2012.

[8] X.-m. T. Yan-fei Zhu, "Overview of swarm Intelligence," *IEEE,* vol. 9, pp. 400-403, 2010.

[9] K. N. Krishnanand and . D. Ghose, "Glowworm swarm optimization for simultaneous capture of

Copyright ⓒ 2019Authors

multiple local optima of multimodal functions," *Springer,* vol. 3, p. 87–124, 2009.

[10] K. Krishnanand and D. Ghose, "Detection of Multiple Source Locations using a Glowworm Metaphor with Applications to Collective Robotics," *IEEE,* p. 14, 2005.

[11] S.-C. Chu, H. C. Huang, J. F. Roddick and J.-S. Pan, "Overview of Algorithms for Swarm Intelligence," *Springer-Verlag Berlin Heidelberg,* pp. 28-41, 2011.

[12] M. P. Asha Gowda Karegowda, "A survey of applications of glowworm swarm optimization algorithm," *IC2IT,* pp. 39-42, 2013.

[13] Yarinezhad, R., & Sarabi, A. "A New Routing Algorithm for Vehicular Ad-hoc Networks based on Glowworm Swarm Optimization Algorithm." *Journal of AI and Data Mining*, *7*(1), 69-76,2019

Copyright © 2019Authors