Solving Freeze Tag Problem In Swarm Robotics Using Genetic Algorithm And Dijikstra's Algorithm

Sathya Priya K
Research Scholar,School of Computer Science Engineering and Applications,Bharathidasan University,Trichy -23, TamilNadu, India

Abstract-*The Freeze-Tag Problem is a situation in which there is "n" number of robots;out of which n-1 ones are "snoozing" and one is "wakeful". Any wakeful robot can move and stir another robot by touching it. The objective is to minimize the time for awakening all the resting robots.In this work,genetic algorithm and dijikstra's algorithm are implemented and compared to the above problem. The goal is to limit the expected time to awaken all the robots.*

Keywords-Genetic Algorithm,dijikstra's algorithm, Selection Process

I. INTRODUCTION

The Freeze-Tag Problem problem seeks to find the fastest ordering to awaken a set of inactive nodes. Solutions are represented bydifferent awakening schedules, representations of this ordering, that describe how the work is completed in awakening all principals in the problem, be they robots or nodes on a network. Previous research has tackled this problem primarily using greedy choice strategies. These strategies are fairly good; there is certainly room to look into the varying factors of a solution and its creation in order to attain better results. This research set out to do this using Genetic Algorithms. To accomplish this, the algorithm treats solutions as chromosomes in a gene pool, testing their fitness and genetic viability compared to one another as well as providing means for genetic variation through recombination and mutation. This yields improved solutionsfrom the initial generation, and the algorithm processes many generations before returning a result. Using a Genetic Algorithm, here evaluated awakening schedules for the FTP and compared them to solutions given by greedy algorithms and a center-of-mass algorithm. The next section describes the FTP in detail. The third section runs through the Genetic Algorithm as it applies to solving this problem, followed by this system and data. This research end remarks concerning potential improvements to Genetic Algorithms applied to the FTP

II. PROBLEM DEFINITION

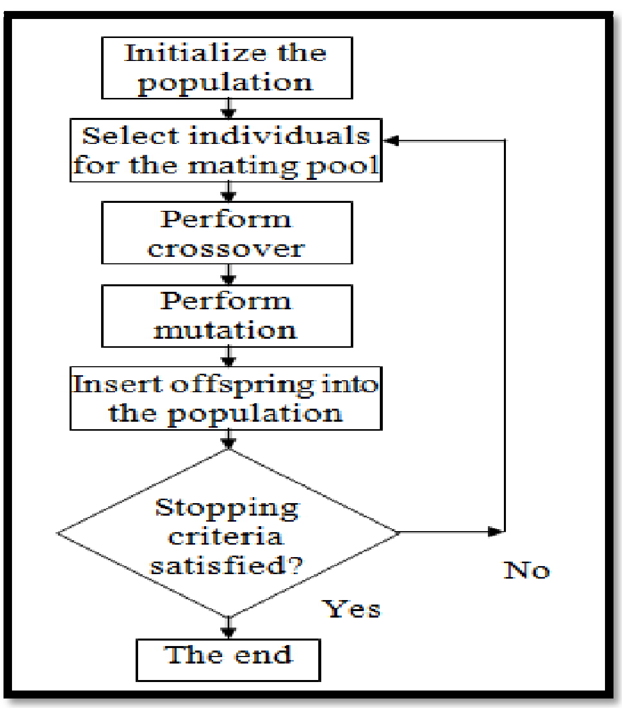
The genetic algorithm , improve upon the awakening schedules of prior applications of the FTP. Due to its evolutionary processes and genetic recombination, the algorithm is able to make non-greedy decisions and arrive at a faster solution. Those schedules that are slower and, thus, undesirable are deemed unfit and fail to survive on to future generations. Also unlike greedy choice methods, the mutation phase allows for seemingly negative connections in the awakening schedule to be made that may potentially result in quicker overall solutions to the problem. The most essential portions of the genetic algorithm are the recombination and cross-over stage and the mutation stage. These evolutionary simulators are the driving force of increasing the fitness of a generation (the former) and aiding in the search throughout all neighborhoods instead of just one (the latter). Further improving upon these algorithms could result in even better solutions to the FTP. Perhaps better conditions for choosing how to construct a child schedule from two parents could be offered, or a smarter search through potential swaps within the mutation function could yield a more intelligent genetic alteration. Alterations to the population size of a generation or the conditions with which evolutionary simulation ends may also provide further improvements to solutions. Other optimizations should also increase the performance of the genetic algorithm. Such implementations, as their work on the TSP problem indicates, would narrow the gap between potential lower bounds to the FTP and the solutions given by a genetic algorithm.

The Dijkstra's algorithm, it is very comparable to Prim's algorithm for minimum spanning tree. Like Prim's MST, make a *SPT (shortest path tree)* with given resource as root. Keep two sets, one set contains vertices included in shortest path tree, and other set includes vertices not yet included in shortest path tree. At every step of the algorithm, find a highest point which is in the other set (set of not yet included) and has least distance from source. As a result, all the asleep robots at the same depth are awakened simultaneously through parallel awakening.

III. PROPOSED WORK

Genetic algorithms require that the initial generation be comprised of viable solutions to the problem toward which they are applied. Thus, with this initial generation of solutions (chromosomes), the evolutionary algorithms can simulate genetics as they work towards improvements of the N given solutions, the goal being to find a very good solution due to evolution over generations. To this end, our algorithm needed a set of solutions to the FTP at the start. The initial generation was seeded with N awakening schedules given by such algorithms as the greedy-fixed, greedy-dynamic, or even random, unbalanced awakening schedules. Before our algorithm initiated its genetic process, these solutions were placed into the generation structure, a data type that contained an array of chromosomes. The chromosomes were arrays of n (n to reflect the n robots; N represents the number of solutions in a generation) nodes representing the awakening schedule. Each node represented a robot and contained the ID of the robot that awakened it, the make span of the awakening schedule from its position down, and the ID of the one or two robots awakened from this position (both the now awakened robot and its contact proceed from this position to awaken others). After the initial generation was created, the algorithm simulated the following steps in order to reach the best solution it can reach via improvements on the current generation: evaluation, fitness and selection, reproduction and cross-over, and mutation. The following is a basic outline of the steps of the algorithm:

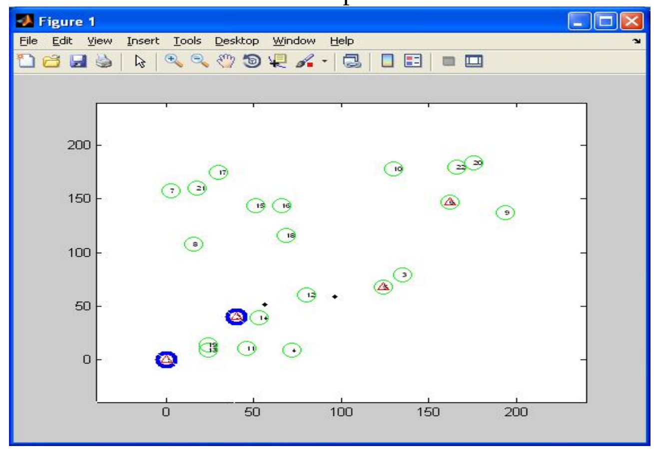
The GA quality originates from the certainly parallel hunt of the arrangement space that it performs by means of a populace of competitor arrangements and this populace is controlled in the simulation.A wellness work is utilized to assess people, and conceptive achievement shifts with wellness. A powerful GA portrayal (i.e., changing over an issue space into qualities) and significant wellness assessment are the keys of the accomplishment in GA applications.

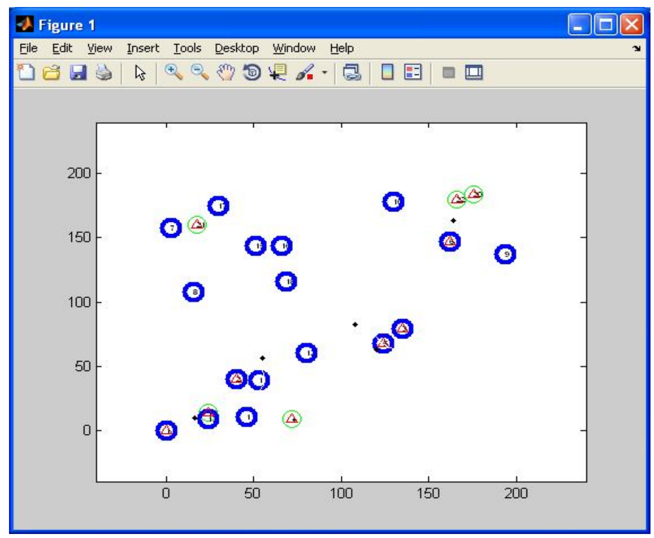


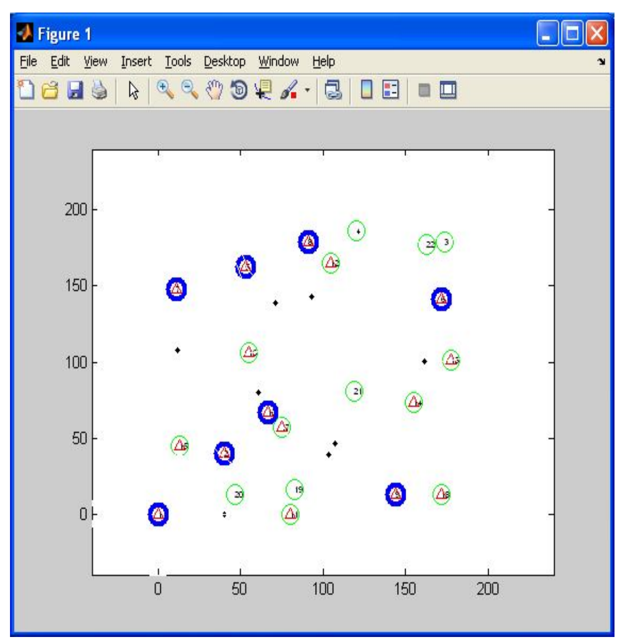
IV. RESULT AND ANALYSIS

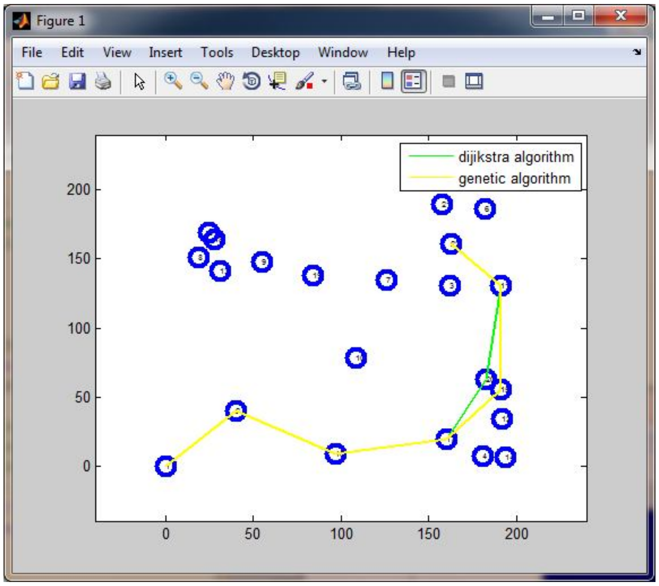
The MATLAB implementations in given bellow

Selection process

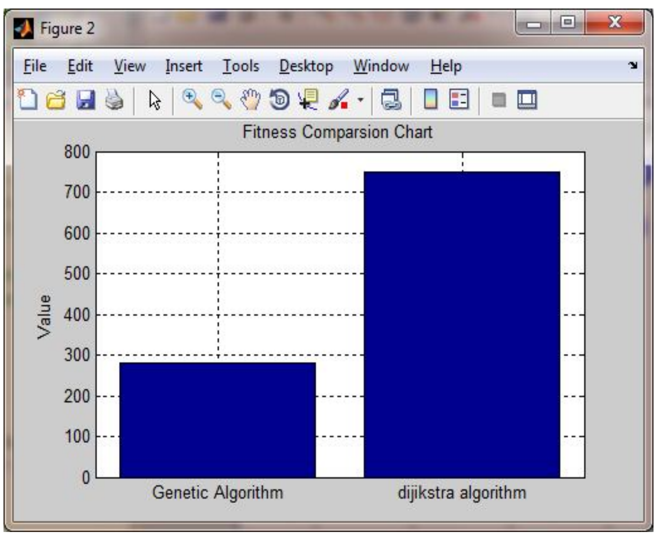




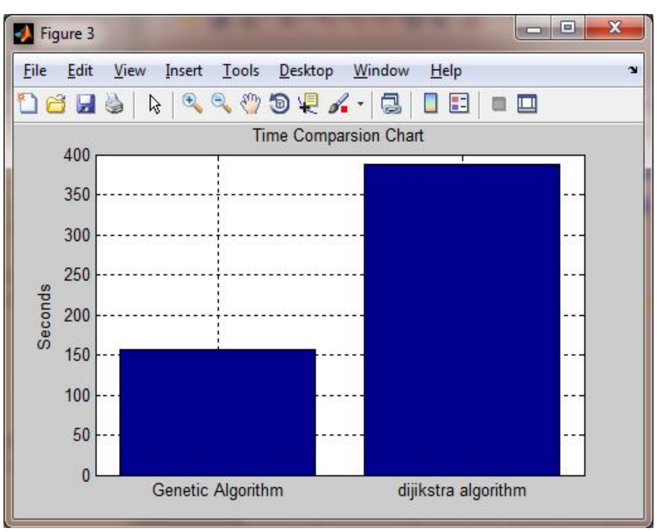




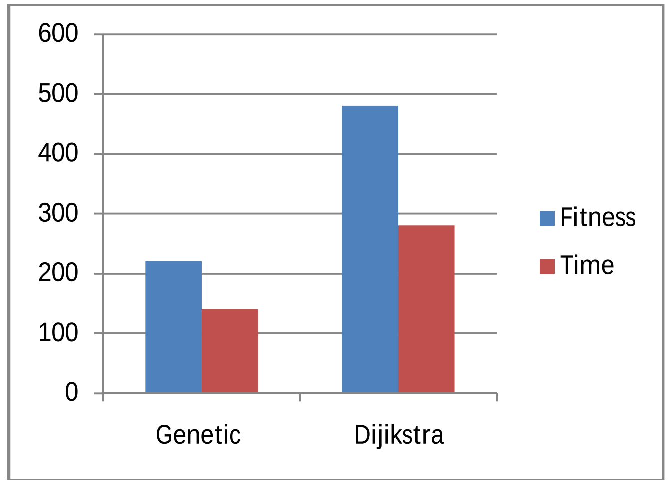
Fitness value comparision chart:



Time comparision chart:



Genetic algorithm and dijikstra's algorithm comparision chart:



V. CONCLUSION

In this implementation work, solving the freeze tag problem by using 22 nodes and using genetic algorithm and dijikstra's algorithm.The genetic algorithm gives better performance when compared to dijikstra's algorithm.The genetic algorithm is reduces the minimizing time and minimizing the fitness value.The genetic algorithm is well suited for Solving freeze tag problem .

FUTURE WORK

As a future work, different selection method and technique can be used to solve freeze tag problems.

REFERENCES

[1] Aarts, Emile and Jan Kaerel Lenstra. Local Search in Combinatorial Optimization. John Wiley and Sons, Ltd. 1997.

[2] Arkin, E. M., Michael A. Bender, Sandor P. Fekete, Joseph S. B. Mitchell, and Martin Skutella. "The Freeze-Tag Problem: How to Wake Up a Swarm of Robots." In Proc. 13th ACM-SIAM Sympos. Discrete Algorithms. pp. 568 – 577, 2002.

[3] Arkin, E. M., Michael A. Bender, Dongdong Ge, Simai He, and Joseph S. B. Mitchell. "Improved Approximation Algorithms for the Freeze-Tag Problem." Proceedings of the fifteenth annual ACM symposium on Parallel algorithms and architectures. pp. 295 – 303, 2003.

[4] Julstrom, Bryant A. and Gunther R. Raidl. "A Permutation-Coded Evolutionary Algorithm for the Bounded-Diameter Minimum Spanning Tree Problem." 2003 Genetic and Evolutionary Computation

Conference's Workshops Proceedings, Workshop on Analysis and Desgn of Representations. pp 2-7, 2003.

[5] Raidl, Gunther R. "An Efficient Evolutionary Algorithm for the DegreeConstrained Minimum Spanning Tree Problem." Proceedings of the 2000 Congress on Evolutionary Computation CEC00. 2000.

[6] Sztainberg, Marcelo O., Esther M. Arkin, Michael A. Bender, and Joseph S. B. Mitchell. "Analysis of Heuristics for the Freeze-Tag Problem." Proc. Scandinavian Workshop on Algorithms, Vol. 2368 of Springer-Verlag LNCS, pp. 270 – 279, 2002.

[7] Whitley, Darrel. "A Genetic Algorithm Tutorial." Statistics and Computing (4). pp. 65 – 85, 1994. [8] TSPLIB.

[8] J. Gottlieb, B. A. Julstrom, F. Rothlauf, and G. R. Raidl. Pr¨ufer numbers: A poor representation of spanning trees for evolutionary search. In L. Spector, E. Goodman, A. Wu, W. Langdon, H.-M. Voigt, M. Gen, S. Sen, M. Dorigo, S. Pezeshk, M. Garzon, and E. Burke, editors, Proceedings of the 2001 Genetic and Evolutionary Computation Conference, pages 343–350. Morgan Kaufmann, 2000.

[9] C. Jordan. Sur les assemblages des lignes. J. Reine Angew. Math., 70:185–190, 1869.

[10] B. A. Julstrom. Encoding rectilinear Steiner trees as lists of edges. In G. Lamont, J. Carroll, H. Haddad, D. Morton, G. Papadopoulos, R. Sincovec, and A. Yfantis, editors, Proceedings of the 16th ACM Symposium on Applied Computing, pages 356–360. ACM Press, 2001.

[11] G. Kortsarz and D. Peleg. Approximating shallow-light trees. In Proceedings of the 8th Symposium on Discrete Algorithms, pages 103–110, 1997.

[12] G. Kortsarz and D. Peleg. Approximating the weight of shallow Steiner trees. Discrete Applied Mathematics, 93:265–285, 1999.

[13] C. C. Palmer and A. Kershenbaum. Representing trees in genetic algorithms. In D. Schaffer, H.-P. Schwefel, and D. B. Fogel, editors, Proceedings of the First IEEE Conference on Evolutionary Computation, pages 379–384. IEEE Press, 1994.

[14] R. C. Prim. Shortest connection networks and some generalizations. Bell System Technical Journal, 36:1389–1401, 1957.

[15] G. R. Raidl. An efficient evolutionary algorithm for the degree-constrained minimum spanning tree problem. In C. Fonseca, J.-H. Kim, and A. Smith, editors, Proceedings of the 2000 IEEE Congress on Evolutionary Computation, pages 104–111. IEEE Press, 2000.

[16] G. R. Raidl and B. A. Julstrom. Edge-sets: An effective evolutionary coding of spanning trees. IEEE Transactions on Evolutionary Computation, 2003. To appear.

[17] K. Raymond. A tree-based algorithm for distributed mutual exclusion. ACM Transactions on Computer Systems, 7(1):61–77, 1989.

[18] F. Rothlauf, D. Goldberg, and A. Heinzl. Network random keys – a tree network representation scheme for genetic and evolutionary algorithms. Evolutionary Computation, 10(1):75–97, 2002.