Incorporating Strategy Adoption into Genetic Algorithm Enabled Multi-Agent Systems

Yasinthara Madushani
Department of Physics
University of Colombo
Colombo, Sri Lanka
Email: 2015s15202@stu.cmb.ac.lk

Dharshana Kasthurirathna
Faculty of Computing
Sri Lanka Institute of Information Technology
Malabe, Sri Lanka
Email: dharshana.k@sliit.lk

***Abstract*—Genetic Algorithm (GA) is a widely adopted optimization technique under evolutionary optimization. Inspired by the evolutionary operators of selection, crossover and mutation, Genetic Algorithms have been used to successfully solve myriad optimization problems in a wide range of domains, including in optimizing multi-agent systems. On the other hand, Evolutionary Game Theory (EGT) is used to model social-economic systems by mimicking social evolution by adopting neighborhood strategies in a stochastic manner. In this work, an extended GA is proposed for multi-agent systems, which incorporates the strategy adoption in EGT into GA enabled multi-agent systems. The proposed extended GA algorithm is applied to an example multi-robot navigation application. The proposed algorithm gives promising results in terms of the convergence time, compared to the GA based approach. Possible applications of the proposed algorithm are also discussed, while indicating potential future research directions.**

***Index Terms*—Genetic Algorithm, Evolutionary Game theory, Multi-Robot navigation**

I. Introduction

Evolutionary computation adopts the Neo-Darwinian theory of biological evolution to a computational environment. It has resulted in the creation of multiple heuristic-based optimization algorithms, based on a set of evolutionary operations inspired by the principles of biological evolution, such as natural selection and genetic inheritance. Evolutionary computation is an umbrella term that includes population-based optimization techniques such as Genetic Algorithms [1] and genetic programming [2]. Evolutionary optimization techniques have been increasingly applied in a wide range of applications, from practical applications in production and engineering to leading scientific research.

Compared to many other optimization methods, GAs use a population of individuals and this is one of the reasons for their effectiveness [3]. Evolutionary Game theory (EGT) [4], on the other hand, is used to model social and economic systems through autonomous interactions of agents. It's generally not used for optimization, rather for modeling the evolution of socio-economic systems [4]. Further, EGT employs a strategy adoption process, which is inspired by social interactions in social systems [5].

In societies of humans and other social animals, both the biological and social evolution seems to have a significant effect on optimizing preferable traits and behavior. A good example is how domestic animals such as dogs seem to 'adopt' optimal behavior from the group while passing preferable genetic traits through biological evolution [6]. Moreover, it appears that the contribution of the biological and social components in optimizing behavior may depend on the species and the specific function [7]. For instance, in humans, social or cultural evolution may play a more prominent role than any other species. Therefore, inspirations may be drawn from nature on the effect of social adoption on the GA based optimization in multi-agent systems.

Accordingly, this study proposes a novel approach to incorporate the social evolution through strategy adoption to extend the standard Genetic algorithm in a multi-agent setting. In order to test the proposed algorithm, a well-known multi-robot navigation problem is used, and it demonstrates that the inclusion of 'social interactions' into the GA, significantly improves the algorithm's performance. The proposed approach is loosely related to co-evolutionary approaches in that the evolution of a chromosome may depend on the other chromosomes in a population. However, it differs from them in that the social interaction used in the proposed approach is not the main mechanism for calculating the fitness of co-evolving species.

The remaining sections of this paper are organized as follows. The following section provides a review of the background in relation to the GA, EGT and related optimization algorithms. The next section describes the proposed algorithm with an introduction to the experimental setup and the evaluation method. The next section describes the analysis of results obtained, followed by the discussion of the results. Finally, we present a summary of the overall results, draws final conclusions and indicates possible future directions.

II. Background

A. Genetic Algorithm

Genetic algorithm is a heuristic based optimization method that finds near-optimal solutions by subjecting a population of points in a search space to a set of biologically inspired operators [8]. The 'fitness' of each member of the GA population is computed by an evaluation function that measures how well the individual performs in the problem domain. GAs primarily use three genetic operators; selection, crossover, and mutation, in order to emulate the natural selection process. One

978-1-7281-6929-3/20/$31.00 ©2020 IEEE

key feature of Genetic algorithm is that the evolutionary time-scale is much larger compared to a lifetime of a chromosome [9], [10].

1) Selection - Selection is the method of selecting potentially better individuals, to form a mating pool and let them pass their genes to the next generation [11].
2) Crossover - The crossover operator simulates the method of gene recombination, which aims to pass the most effective genes, on to subsequent generation [11].
3) Mutation - The mutation operator is used to prevent the loss of information that occurs as population converges on the fittest individuals [11].

B. Evolutionary Game Theory

Evolutionary game theory is an application of the mathematical framework of games theory [5], [12] to the modelling of social-economic systems and biological systems. It enables the understanding the selective pressures that influence the evolution of the ways of agents engaged in interactions with potential conflicts. It is the branch of science that has resulted from the adoption of game theory into evolutionary biology [13]. It is used to study how a specific strategy or set of strategies evolve over time in a population of players. An Evolutionarily Stable Strategy (ESS) is a strategy that dominates over the other strategies over time [14]. One of the key applications of EGT is to identify the ESS in a given population of autonomous players.

Evolutionary game theory asserts that the evolutionary success of a strategy of an agent depends on how it interacts with others in the same population. Thus, the evolutionary stability of a single agent cannot be measured in isolation. Instead, it must be assessed in the context of the full population in which it operates. Thus, EGT can have many applications in biology, social science and economics, as in all those domains the social interactions determine the evolutionary success of an individual agent. From a GA point of view, an organism's genetically-determined characteristics and behaviors are analogous to its strategies in a game, its fitness is comparable to its accumulated payoff, and this payoff depends on the strategies of the organisms with which it interacts. In order to accurately model the evolution of strategies, Stochastic strategy adoption processes are used. It is used to update the strategy of an agent by comparing it with a selected neighbor's strategy.

1) Stochastic Strategy Adoption: In the social evolution, players would be more inclined to adopt the apparently successful strategy and survive without getting replaced from the population. In order to model this kind of social evolution, a stochastic strategy adoption process can be applied. When employing this particular strategy adoption process, a player going through the evolutionary process is not directly replaced. Instead, its strategy could be chosen by comparing its cumulative payoff with that of a selected neighbor in a stochastic manner [5], [12].

Successful strategies are promoted through imitation and learning. Players, instead of analyzing the situation in detail, imitate the more successful ones [13]. Following is an example of a strategy adoption process [5].

$$P = \max\{0, (P_y - P_x)/[\ k > (T - S)]\ \} \tag{1}$$

In the above strategy adoption process, during each generation, all pairs of directly connected players, x and y, engage in a single round of the game, their accumulated payoff being stored as P_x and P_y respectively, representing the fitness of each player at the end of one generation. T and S are the cooperation and defection payoffs, when one player cooperates and the other defects. To update a strategy in player x, a neighbor y is selected out of all k_x neighbors, based on a probability that is proportional to its payoff difference with the node in concern. Then, the player x's strategy is updated with the select neighbor y's strategy. This process is repeated for all the players in the population iteratively until the entire population converges on the evolutionarily stable strategy.

This process has been used in Santos et al. [5] to demonstrate the evolution of cooperation in Iterated Prisoner's Dilemma games with pure strategies. In this work, a similar process is adopted to enable agents to adopt neighbors' strategies.

In strategy adoption, three types of neighborhoods are considered: Von-Neumann, Moore and Radial [15]. The Von-Neumann neighborhood considers one representative of each axial direction in the central lattice. For the Moore neighborhood, in addition to Von-Neumann, one node in each of the diagonal directions is considered. In a radial neighborhood, the nodes that fall within a certain radial boundary around the central node are cones. In this work, we employ the Radial method of determining the neighborhood, as its widely used in the literature and it's a more intuitive approach to determine the neighborhood in the application that we chose.

C. Co-Evolution

In 1994, Paredis introduced Co-evolutionary Genetic Algorithms [16], [17]. A number of co-evolutionary algorithms have recently been proposed. The algorithm operates on two sub-populations: the main sub-population, which includes individuals representing some species, and the additional sub-populations that encode certain limitations, conditions, or test sites on a solution [18].

The proposed algorithm differs from the above-mentioned work, in that it uses a neighborhood strategy adoption process to adopt fitter strategies in an agent-based system.

D. Applications of GA in multi-robot navigation

Multi-robot navigation involves finding the optimal or near-optimal collision-free path from a start location to a goal location in an environment with obstacles for multiple robots that are spatially distributed. Multiple attempts have been carried out to apply GA to this application. The work on path planning methods for mobile robots based on an adaptive genetic algorithm [19], parallel elite genetic algorithm [20], knowledge-based genetic algorithm [21], co-evolutionary genetic algorithm [22] are some examples of that. However, to the best of our knowledge, the existing work do not incorporate strategy adoption into the application of GA for optimizing multi-robot navigation.

III. Methodology

This section describes the proposed algorithm and the multi-robot navigation application that is employed to evaluate it against the GA based approach in a multi-agent setting.

Algorithm 1 Incorporating strategy adoption in a GA enabled multi-agent system

1: Initialization – Randomly generate initial population in each agent.
2: Neighbor assignment – Assign the neighborhood for each agent
3: Evaluation – Each member of the population in each agent is then evaluated and calculate a 'fitness' for that individual.
4: Adoption - Each agent adopts the fitter solutions from neighboring agents
5: Crossover – In each agent, create new solutions by combining of selected solutions.
6: Mutation – In each agent, mutation is applied by making minute changes at random to each solution.
7: Selection – In each agent, discard the less fit solutions and retain the fitter solutions in the population.
8: Repeat – Repeat the steps 4-7 until maximum allowed iterations or convergence is reached.

Adopting from standard GA, the following are the three basic hyper-parameters used for the Crossover, Mutation and Selection.

Crossover rate Crossover rate determines the number of times a crossover occurs in the chromosomes in one generation and is between 0-1.

Mutation rate Mutation rate determines how many genes mutate in one generation. It is in the range of [0:1] and it can have a significant effect of the performance of the GA implementation [23], [24].

Population size Selection of the size of the population critical, as if the size of the population is relatively small, it may result in a smaller search space, which may lead to a local optimum. However, if the size of the population is too large, this would increase the area of search and increase the computational time [25].

A. Neighbor assignment

The step 2 of the proposed algorithm refers to the assignment of the neighborhood of each agent. In a population of agents that are spatially distributed, each agent may be running a GA optimization algorithm to optimize a function. In order to adopt strategies from neighboring agents, first the neighbors of each agent needs to be assigned. For this purpose, we employ the radial neighborhood theorem to identify the neighborhood of each agent.

Depending on the neighborhood function employed, there may be hyper-parameters that can be used to adjust the neighborhood distribution. In the case of the radial neighborhood theorem based neighbor assignment, the radius of the neighborhood may be a hyper-parameter that may be tuned to fine-tune the optimization process. In the case of static agents, the neighbor assignment may be only done initially. However, if the agent placement is dynamic, the neighbor assignment operation may have to be repeated iteratively. We only consider the scenario where the agents are static, the proposed algorithm may be extended for a dynamic environment as well.

B. Adoption

The Adoption operator depicted in step 4 in the proposed algorithm consists of two operators, namely, selection for adoption and strategy adoption.

1) Selection for adoption: neighbor selection to adopt strategies is inspired by stochastic Strategy Adoption process in Evolutionary Game theory. Similar to the strategy adoption process used in Santos et al. [5], a neighbor to adopt a strategy from is selected based on a probability proportional to their average fitness. Here, average fitness of the current population in an agent is regarded as equivalent to the cumulative payoff in the EGT context. Further, a strategy in EGT is considered to be a solution or a chromosome in the GA set-up.

$$P = \max\{0, (F_y - F_x)/[\sum_{i}^{k}(F_{y_i} - F_x)]\ \} \tag{2}$$

P is the probability that a neighbor is selected. F_y and F_x are the average fitness values of populations of an agent x and y, respectively, and k is the number of neighbors of agent x. The difference of the average fitness values is divided by the sum of the differences between each neighbor and the respective node, ensuring that $0 \leq P \leq 1$. Accordingly, the probability of a neighbor being selected would be proportional to the difference of its average fitness and the selected agent's average fitness.

2) Strategy adoption: In the population representation, a segment of the population within an agent at any given time could be adopted from the neighborhood, and the remaining segment of the population could be resulting from the GA specific operators. Once combined, a population may undergo the crossover and mutation genetic operators. The extent of the population that is adopted determines the level of influence that neighboring agents' have on a population of chromosomes.

$$P_u = \ \beta_B P_b + \ \beta_S P_s \tag{3}$$

Here, P_u is the combined population and P_b is the segment of the original population that is retained based on their individual fitness and P_s is the adopted segment of the population-based on their individual fitness. The segment of chromosomes P_s is obtained by applying Stochastic strategy adoption, which is a result of the spatial interactions of an agent with its neighbors. It has to be noted that this step applies an implicit selection of solutions based on their individual fitness, both inherent and adopted, in order to keep the population stable.

The population P_u is a weighted sum of populations where β_B and β_S represent the weight of the biological and social components of evolution, respectively. Thus, $\beta_B, \beta_S \in [0,1], \beta_S + \beta_B = 1$ and $\beta_B = 1 - \beta_S$.

Accordingly, $\beta_B = 1$ would be a special case where the agents and their respective populations would be subjected to standard GA, while $\beta_B = 0$ is a special case when the agents and their respective genomes would be evolved purely in a social context, based on the principles of EGT. The distribution of β_B and β_S between 0 and 1 may result in a collection of agent-based evolutionary models and hence these can be considered as hyper-parameters of the proposed algorithm.

3) Evaluation using Multi-robot Navigation: Genetic algorithms are often applied to robotics as a method of designing sophisticated robotic controllers that enable a robot to perform complex tasks and behaviors, removing the need to manually implement a complicated robotic controller. In order to evaluate the proposed algorithm, a multi-robotic navigation problem is used where multiple robots that are spatially distributed attempt to find the optimum path to reach a particular destination.

As the navigation space of the robots, X-Y 16*16 grid matrix was created, including the robot/goal locations, and obstacles. Value 0 was used to represent an impassable path and 1 to represent the possible path. The location of the end goal was defined with the value 4. Assuming this robot could rotate 45 degrees (main 4 directions and sub 4 directions). At the initialization, the robots were placed in random positions, and obstacle coordinates were marked.

The total population size was set to 200 and the maximum number of iterations were fixed initially. Number of robots were fixed to 100 and the crossover rate and the mutation rate were set to 0.5 and 0.05, respectively. The chromosome length of each chromosome or path was set to 192. Each robot in the navigational space would evolve its solution population independently and parallelly with each other. The initial population of each robot consists of completely random solutions or paths created in each robot. An individual chromosome is represented by a vector, where binary values were used to indicate whether a certain point is visited in the path or not.

Algorithm 2 depicts the adoption of the proposed algorithm to the multi-robot navigation problem.

The Equation 4 gives the fitness function used in measuring the effectiveness of a particular path in reaching the destination.

$$F_j = min(\sum_{i=1}^{N-1} f(i,i+1) * s(i)), j\epsilon[1,n], s\epsilon[0,1]) \quad (4)$$

$$f(i,i+1) = \sqrt{(x_{i+1} - x_i)^2 + (y_{i+1} - y_i)^2} \quad (5)$$

Here, i is the index of the path point in a encoded chromosome and $f(i,i+1)$ is the length of two path way points with index i and $i+1$. The variable j represents a encoded chromosome in the population and (x_i, y_i) gives the position of way point i, while (x_{i+1}, y_{i+1}) represents position of the next way point. N is the total number of way points and n is population size. The variable s is set to 0 where the selected path point has obstacles and 1 where there are no obstacles. Feasibility of the path is indicated by the s and if a path is infeasible, the function would reduce the fitness of the solution.

Algorithm 2 Proposed algorithm applied for multi-robot navigation

```
1:  Robots are randomly placed in the navigation space.
2:  neighbor assignment
3:  Populations are initialized.
4:  Fitness values are calculated for each chromosome of each
    population.
5:  while termination condition not reached do
6:      for Each robot's population do
7:          Strategy adoption
8:          Crossover
9:          Mutation
10:         Selection
11:     end for
12: end while
```

In step 2 of Algorithm 2, the radial neighborhood theorem is applied by drawing a circle with the given radius around each neighbor to assign the neighbors of each robot. In order to do this, the euclidean distance from each robot to each other was calculated. The robots within the given radius were identified as being in the same neighborhood. Figure 1 depicts the neighbor assignment using the radial method.

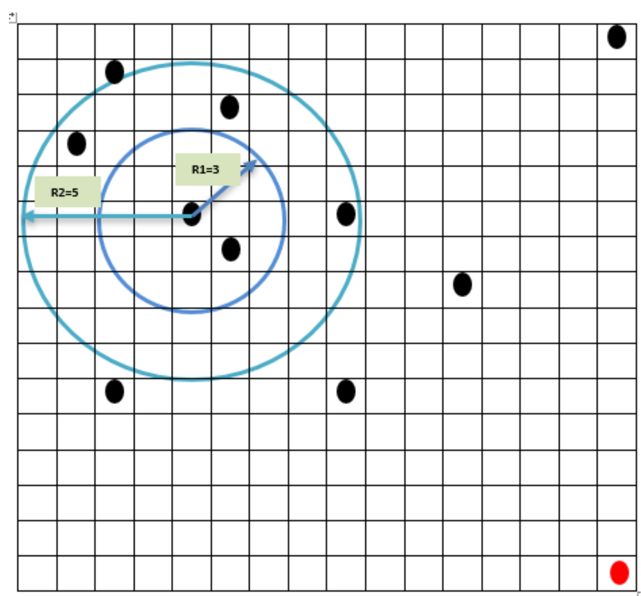

Fig. 1. Neighbor assignment

The algorithm 3 expands on the strategy adoption process that is referred in the step 7 in Algorithm 2, which is adopted to the multi-robot navigation problem.

Algorithm 3 Strategy adoption process

Input: Adoption rate

Average fitness of the population is calculated for each robot.
Select neighbor for strategy adoption
while All neighbors are not visited **do**
    Calculate the average fitness difference between each neighbor and the robot
end while
Select the neighbor for strategy adoption
Adopt the segment of the selected neighbor's chromosomes based on their individual fitness
Retain the proportion of inherent chromosomes based on their fitness.

The strategy adoption process adopts fittest chromosomes for each robot from the neighboring populations from neighbors that are selected in a stochastic manner. The segment of chromosomes in the local population to retain and the segment of chromosomes that are adopted from the selected neighbor are both determined based on each chromosome's individual fitness. Thus, the resulting population is a combination of the local population and the adopted strategies or solutions from neighbors. The combined population would then be subjected to crossover and mutation operations. Thus, the children in latter evolutionary cycles may inherit characteristics of both the genetic lineage as well as the adopted strategies from neighboring populations.

When the stochastic strategy adoption is performed, the starting point of a path or a solution may be different the original starting point, even though the solution may have the same encoded sequence of binary values. This is a result of the neighboring robot being at a different initial location, even though it might be in close proximity. However, since the initial location is always indicated by 1, the adoption of a path may result in the initial position being adjusted to the current robot. This adjustment might make certain adopted paths being infeasible. However, since their fitness may get reduced as a result, such paths may be at an evolutionarily disadvantageous position and may get eliminated in the subsequent iterations.

Two sets of experiments were conducted in order to demonstrate the feasibility and effectiveness of the proposed algorithm. In both sets of experiments, the average convergence time to reach an optimal path of all the agents in the environment, was measured. In the first experiment set, the hyper-parameter neighborhood radius was adjusted to control the neighborhood function, while keeping other hyper-parameters static. In the second experiment set, the adoption rate was varied to control the effect of the neighboring agents on a population, and its effect on the convergence time was measured. These experiments were used to compare the performance of the proposed extended GA algorithm to the standard Genetic Algorithm based approach. The following section shows the results of the experiments conducted.

IV. RESULTS

The aim of the experiments was to verify the efficiency of the proposed algorithm in comparison with the GA based approach. The proposed algorithm was implemented and run for 200 iterations with varying adoption rates and neighborhood radius values. Other hyper-parameters such as size population, chromosome length, and crossover, mutation probability were set to be static. The average convergence time was computed by averaging the average convergence time of all populations within all robots over ten iterations of the evolutionary process.

Figures 2, 3 and 4 depict the variation of the average convergence time when the radius of the neighborhood function is set to 0.2, 0.5 and 0.7, respectively.

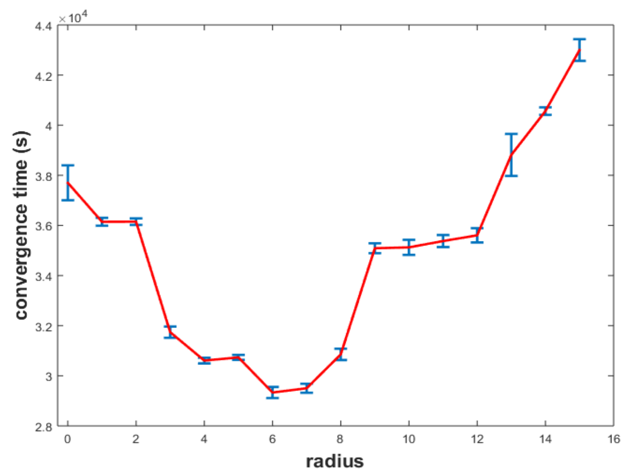

Fig. 2. The variation of the convergence time against neighbor selection radius. Adoption rate is set to 0.2.

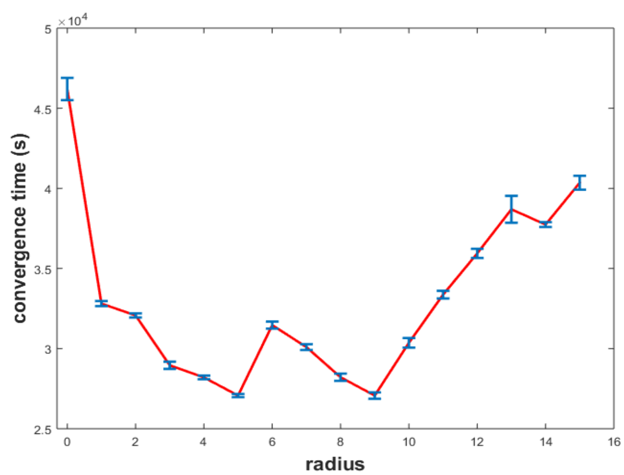

Fig. 3. The variation of the convergence time against neighbor selection radius. Adoption rate is set to 0.5.

The results indicate that for a given adoption rate, there's an optimal radius in the neighborhood function that minimizes

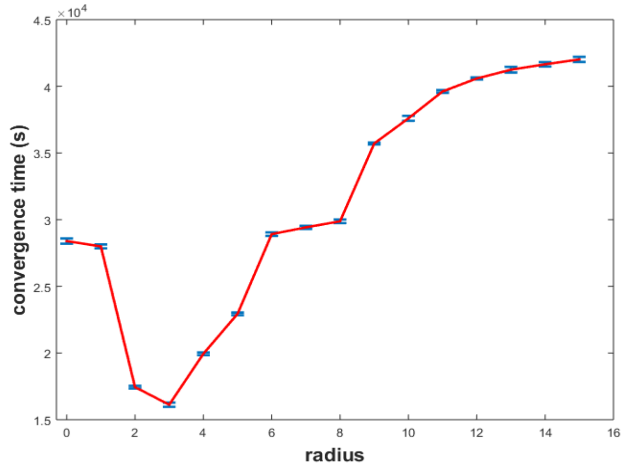

Fig. 4. The variation of the convergence time against neighbor selection radius. Adoption rate is set to 0.7.

the convergence time. Hence, it's also apparent that the membership of the neighborhood has an effect in the optimization process. The scenario when radius is set to 0 is a special case where there's no social effect on the evolution, hence it would be equivalent to a Genetic algorithm based implementation.

Next, adoption rate was varied while using the radius unit values, 2, 5 and 7 for each experimental setup. The resulted graphs are shown in Figures 5, 6 and 7.

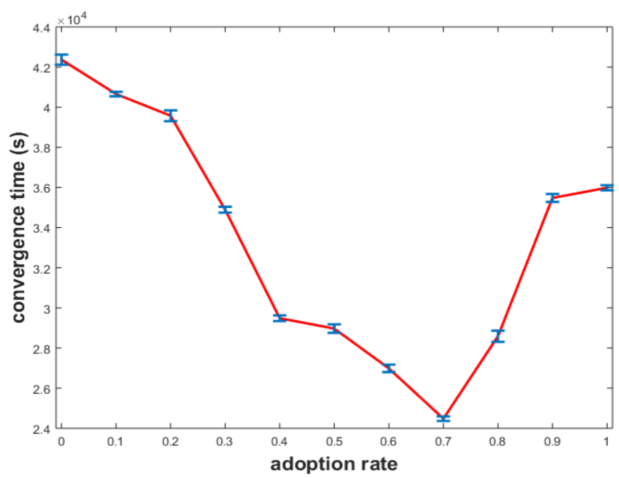

Fig. 5. The variation of the convergence time against adoption rate. Neighborhood radius is set to 2.

When the adoption rate is 0, the algorithm behaves as GA, which is a special case of the proposed algorithm, where only the GA operators are relevant. Likewise, when the adoption rate is 1, the population is reset in each iteration by strategy adoption, bearing some resemblance to the strategy adoption in EGT. Within the range $[0:1]$, there exists infinite number of optimization models that may incorporate the strategy adoption in varying degrees.

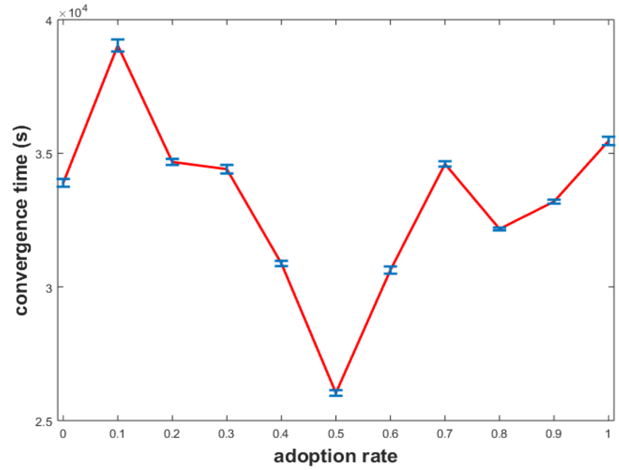

Fig. 6. The variation of the convergence time against adoption rate. Neighborhood radius is set to 5.

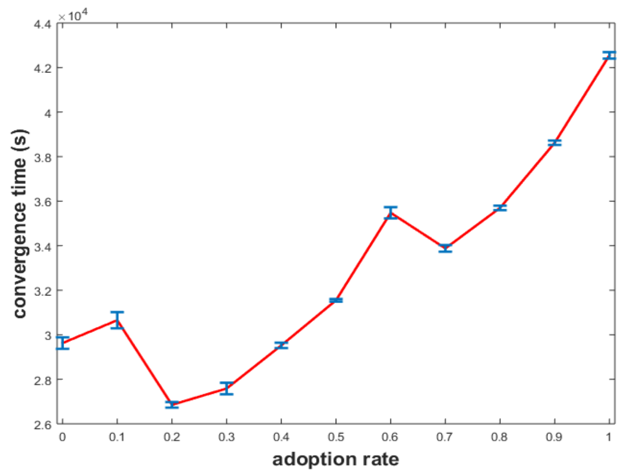

Fig. 7. The variation of the convergence time against adoption rate. Neighborhood radius is set to 7.

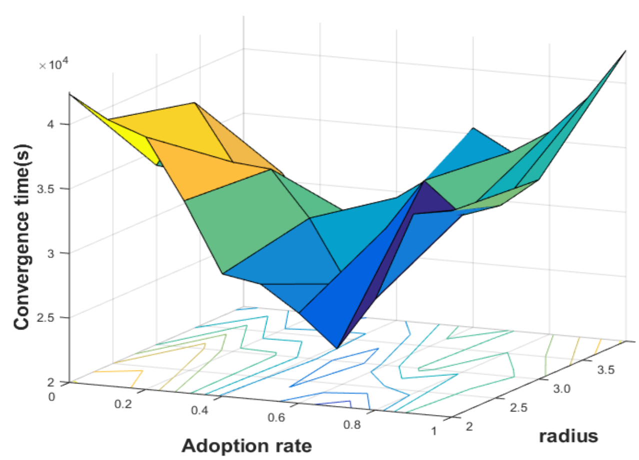

Fig. 8. The variation of the convergence time against the adoption rate and the neighborhood radius

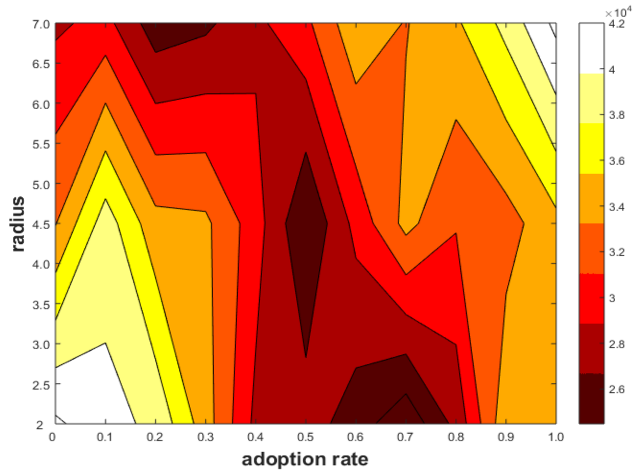

Fig. 9. Performance comparison of the neighborhood radius under variable adoption rates

TABLE I
PARAMETERS OF THE PROPOSED ALGORITHM

Parameter	***Value***
No of robots	***100***
Crossover rate	***0.5***
Mutation rate	***0.05***
Population size	***200***
Chromosome length	***192***

Figure 8 depicts the aggregated results where the variation of the convergence time is plotted against the adoption rate and the neighborhood radius. Figure 9 shows a performance comparison between the purely GA, EGT and the proposed algorithm. Table 1 shows the parameter settings of proposed algorithm. Based on the given results, it is apparent that the optimum convergence time occurs at a non-zero adoption rate. Accordingly, incorporating the stochastic strategy adoption from neighboring populations seem to improve the performance of the optimization process and leads to faster convergence of GA in a multi-agent system. Similarly, when the adoption rate is 1, which is the case where the agents operate based on an EGT inspired optimization algorithm, the convergence time increases. It's also worth noting that the specific radius and adoption rate values that minimizes the convergence time may be problem specific and may not be generalizable.

V. Discussion

In this work, we proposed an algorithm to incorporate the strategy adoption in EGT to the GA based populations in a multi-agent system. The proposed algorithm can be used to control the effect of strategy adoption through social interaction by using an adoption rate. Accordingly, GA can be regarded as a special case of the proposed algorithm. By varying the strategy adoption rate, it may be possible to obtain a collection of models that incorporate the biologically and socially inspired aspects of the evolutionary optimization in varying degrees. The results obtained for a multi-robot navigation application suggests that a non-zero strategy adoption rate lead to faster convergence in an agent-based environment, where each agent may have its own population of chromosomes.

In the robot navigation problem that was used to validate the proposed algorithm, it was observed that not only the adoption rate but also the neighborhood distribution played an important role in the evolutionary process. If the neighborhood radius is smaller, an agent is likely to select a neighbor of close proximity to adopt a strategy. Since the starting point of each path is in close proximity, this may make the adopted strategies more viable. On the other hand, if the radius is relatively larger, there are more neighbors to select from, yet a path adopted from a distant neighbor may not be viable and therefore less fit. Therefore, the optimal radius may be where these two conflicting objectives are optimized. It has to be noted that the use of the hyper-parameter of neighborhood radius may be measured in a different dimension other than spacial distance. The dimension in which the radial distance is measured may depend on the actual application of the proposed algorithm.

The proposed algorithm differs from co-evolutionary techniques as it uses the agent-based stochastic strategy adoption process to adopt chromosomes from neighboring agents. In the application that we use to evaluate the proposed algorithm, each agent has similar yet not the same objective, as the starting point of each robot is different. The agents depict autonomous behavior, similar to the EGT based social and economic models. Thus, it may be more applicable to agent-based systems that are spatially distributed, where each agent may have its own population of chromosomes.

VI. Conclusion

The objective of this work was to propose an approach and an algorithm to adopt the EGT based stochastic strategy adoption to multi-agent GA implementations. It attempts to unify the EGT based stochastic strategy adoption with GA based genetic optimization to obtain an extended optimization algorithm that takes into account the social interaction between agents. By applying the proposed algorithm to an agent-based multi-robot navigation problem, it was observed that the proposed algorithm converges faster than a purely GA based approach is demonstrated.

While the proposed algorithm uses pre-defined hyper-parameter values for the strategy adoption rate and the neighborhood radius, it may be possible to learn the optimal values for these hyper-parameters using a separate optimization process. This may be particularly useful if the problem domain is dynamic, as the optimum strategy adoption rate and the neighborhood radius may be problem specific.

While the proposed algorithm is applied to a single objective optimization problem, it may be extended to solve multi-objective optimization problems by incorporating multiple games. The 'game' in the multi-robot navigation application used in this work is played by each robot against 'nature'.

However, other strategic games such as collision avoidance games where each agent engages in a strategic interaction with each other, could be incorporated in the strategy adoption stage to optimize multiple objective functions.

Even though the proposed algorithm is applied in a multi-agent setting, it may be applied to a single population as well, by defining the neighborhood of a chromosome using a distance function to measure the proximity between individual chromosomes. Such an approach may theoretically converge with the existing work on co-evolutionary techniques.

ACKNOWLEDGMENT

The authors would like to express their heartiest gratitude to Dr. Janaka Wansapura of the Department of Physics, Faculty of Science of the University of Colombo, Sri Lanka, for his continuous guidance and support throughout this research.

REFERENCES

[1] J. H. Holland, *Adaptation in Natural and Artificial Systems*. Ann Arbor, MI: University of Michigan Press, 1975.

[2] J. R. Koza, *Genetic Programming II (Videotape): The next Generation*. Cambridge, MA, USA: MIT Press, 1994.

[3] A. Eiben and M. Schoenauer, "Evolutionary computing," *Information Processing Letters*, vol. 82, no. 1, pp. 1–6, apr 2002. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0020019002002041

[4] C. Adami, J. Schossau, and A. Hintze, "Evolutionary game theory using agent-based methods," pp. 1–26, dec 2016.

[5] F. C. Santos, J. F. Rodrigues, J. M. Pacheco, F. C. Santos, J. F. Rodrigues, and J. M. Pacheco, "Graph topology plays a determinant role in the evolution of cooperation Graph topology plays a determinant role in the evolution of cooperation," no. January, 2006.

[6] J. P. Scott and J. L. Fuller, *Genetics and the Social Behaviour of the Dog*. University of Chicago Press, 2012.

[7] K. N. Laland and V. M. Janik, "The animal cultures debate," *Trends in Ecology & Evolution*, vol. 21, no. 10, pp. 542–547, 2006.

[8] D. E. Goldberg, *Genetic Algorithms in Search, Optimization and Machine Learning*, 1st ed. USA: Addison-Wesley Longman Publishing Co., Inc., 1989.

[9] M. Tabassum and K. Mathew, "A genetic algorithm analysis towards optimization solutions," *International Journal of Digital Information and Wireless Communications*, vol. 4, pp. 124–142, 01 2014.

[10] H. Lingaraj, "A Study on Genetic Algorithm and its Applications," *International Journal of Computer Sciences and Engineering*, vol. 4, pp. 139–143, 2016.

[11] M. Amjad, S. Ikramullah, R. Kousar, R. Ahmad, M. Agha, Z. Faping, N. Anjum, and U. Asgher, "Recent research trends in genetic algorithm based flexible job shop scheduling problems," *Mathematical Problems in Engineering*, vol. 2018, pp. 1–32, 02 2018.

[12] D. Kasthurirathna and M. Piraveenan, "Topological stability of evolutionarily unstable strategies," in *2014 IEEE Symposium on Evolving and Autonomous Learning Systems (EALS)*, 2014, pp. 35–42.

[13] A. Traulsen and C. Hauert, *Stochastic Evolutionary Game Dynamics*. John Wiley Sons, Ltd, 2010, ch. 2, pp. 25–61. [Online]. Available: https://onlinelibrary.wiley.com/doi/abs/10.1002/9783527628001.ch2

[14] D. Easley and J. Kleinberg, *Networks, Crowds, and Markets: Reasoning about a Highly Connected World*. Cambridge University Press, 2010. [Online]. Available: http://www.cs.cornell.edu/home/kleinber/

[15] Y. Guanci, "Game Theory-Inspired Evolutionary Algorithm for Global Optimization," *Algorithms*, vol. 10, p. 111, 2017.

[16] K.-B. Sim, D.-W. Lee, and J. Kim, "Game theory based coevolutionary algorithm: A new computational coevolutionary approach," *International Journal of Control Automation and Systems*, vol. 2, pp. 463–474, 12 2004.

[17] F. Seredyński, A. Zomaya, and P. Bouvry, "Function optimization with coevolutionary algorithms," 01 2003, pp. 13–22.

[18] J. Derrac, S. García, and F. Herrera, "IFS-CoCo: Instance and feature selection based on cooperative coevolution with nearest neighbor rule," *Pattern Recognition*, vol. 43, no. 6, pp. 2082–2105, jun 2010.

[19] C. Liu, H. Liu, and J. Yang, "A Path Planning Method Based on Adaptive Genetic Algorithm for Mobile Robot," vol. 8, 2011.

[20] C. Tsai, H. Huang, and C. Chan, "Parallel elite genetic algorithm and its application to global path planning for autonomous robot navigation," *IEEE Transactions on Industrial Electronics*, vol. 58, no. 10, pp. 4813–4821, Oct 2011.

[21] Y. Hu and S. Yang, "A knowledge based genetic algorithm for path planning of a mobile robot," vol. 2004, 2004, pp. 4350 – 4355 Vol.5.

[22] H. Qu, K. Xing, and T. Alexander, "An improved genetic algorithm with co-evolutionary strategy for global path planning of multiple mobile robots," *Neurocomputing*, vol. 120, pp. 509–517, nov 2013.

[23] W.-Y. Lin, W.-Y. Lee, and T.-P. Hong, "Adapting crossover and mutation rates in genetic algorithms." *J. Inf. Sci. Eng.*, vol. 19, pp. 889–903, 09 2003.

[24] L. Davis, "Adapting Operator Probabilities in Genetic Algorithm," in *3rd International Conference on Genetic Algorithm*, 1995.

[25] O. Roeva, S. Fidanova, and M. Paprzycki, "Influence of the population size on the genetic algorithm performance in case of cultivation process modelling," *undefined*, 2013.

[26] H. Qu, K. Xing, and T. Alexander, "Neurocomputing An improved genetic algorithm with co-evolutionary strategy for global path planning of multiple mobile robots $," *Neurocomputing*, vol. 120, pp. 509–517, 2013. [Online]. Available: http://dx.doi.org/10.1016/j.neucom.2013.04.020

[27] Lee Jacobson, "Creating a genetic algorithm for beginners," 2012. [Online]. Available: http://www.theprojectspot.com/tutorial-post/creating-a-genetic-algorithm-for-beginners/3

[28] S. Gleeson, "Game Theory and Animal Behavior. Lee Alan Dugatkin , Hudson Kern Reeve ," *The Quarterly Review of Biology*, vol. 74, no. 4, p. 497, 1999. [Online]. Available: https://doi.org/10.1086/394204

[29] S. G. Ficici, "Solution Concepts in Coevolutionary Algorithms," Ph.D. dissertation, USA, 2004.

[30] J. Kavetha, "Coevolution Evolutionary Algorithm: A Survey," *International Journal of Advanced Research in Computer Science*, vol. 4, no. 4. [Online]. Available: www.ijarcs.info

[31] H.-P. Schwefel, *Numerical Optimization of Computer Models*. USA: John Wiley & Sons, Inc., 1981.

[32] T. Ganesan, I. Elamvazuthi, and P. Vasant, "Multiobjective design optimization of a nano-CMOS voltage-controlled oscillator using game theoretic-differential evolution," *Applied Soft Computing*, vol. 32, pp. 293–299, jul 2015.

[33] R. Storn and K. Price, "Differential evolution: A simple and efficient adaptive scheme for global optimization over continuous spaces," *Journal of Global Optimization*, vol. 23, 01 1995.

[34] M. Duan, H. Yang, S. Wang, and Y. Liu, "Self-adaptive dual-strategy differential evolution algorithm," *PLOS ONE*, vol. 14, no. 10, pp. 1–25, 10 2019. [Online]. Available: https://doi.org/10.1371/journal.pone.0222706

[35] H. Lingaraj, "A Study on Genetic Algorithm and its Applications," *International Journal of Computer Sciences and Engineering*, vol. 4, pp. 139–143, 2016.

[36] Z. Huang and Y. Chen, "An Improved Differential Evolution Algorithm Based on Adaptive Parameter," *Journal of Control Science and Engineering*, vol. 2013, p. 462706, 2013. [Online]. Available: https://doi.org/10.1155/2013/462706

[37] S. Das, P. Nagaratnam Suganthan, and S. Member, "Differential Evolution: A Survey of the State-of-the-Art," *IEEE TRANSACTIONS ON EVOLUTIONARY COMPUTATION*, vol. 15, no. 1, 2011. [Online]. Available: http://ieeexplore.ieee.org.

[38] Y. Yang, J. Ruan, B. Liu, Y. Liu, and Y. Shi, "Evolution of Cooperation with Moore Neighborhood and Self-Playing Rule," *Discrete Dynamics in Nature and Society*, vol. 2015, p. 989024, 2015. [Online]. Available: https://doi.org/10.1155/2015/989024