

Self-adaptive Hybrid Genetic Algorithm using an Ant-based Algorithm

Tarek A. El-Mihoub

Computer Engineering Department
University of Tripoli
Tripoli, Libya
tmihoub@tripoliuniv.edu.ly

Adrian Hopgood

Sheffield Business School
Sheffield Hallam University
Sheffield, UK
a.hopgood@shu.ac.uk

Ibrahim A. Aref

Computer Engineering Department
University of Tripoli
Tripoli, Libya
i.aref@ec.uot.edu.ly

Abstract—The pheromone trail metaphor is a simple and effective way to accumulate the experience of the past solutions in solving discrete optimization problems. Ant-based optimization algorithms have been successfully employed to solve hard optimization problems. The problem of achieving an optimal utilization of a hybrid genetic algorithm search time is actually a problem of finding its optimal set of control parameters. In this paper, a novel form of hybridization between an ant-based algorithm and a genetic-local hybrid algorithm is proposed. An ant colony optimization algorithm is used to monitor the behavior of a genetic-local hybrid algorithm and dynamically adjust its control parameters to optimize the exploitation-exploration balance according to the fitness landscape.

Keywords—Genetic Algorithms; Hybrid Genetic Algorithms; Self-adaptation; Ant Colony Optimization; Memetic Algorithms;

I. INTRODUCTION

The power of genetic algorithms comes from their ability to combine both exploration and exploitation in an optimal way [1]. However, although this optimal combination may be theoretically true, there are problems in practice due to using limited population sizes [2]. Incorporating a local search method within the global genetic algorithm can be a solution for combating the effect of utilizing limited population sizes [3].

The correct balance between exploration and exploitation in a hybrid genetic algorithm (HGA) depends mainly upon the fitness landscape of the problem to be solved in addition to the algorithm setup [4]. The hybrid search can adapt to a variety of fitness landscapes through adapting the control parameters associated with the genetic operators, local search control parameters and a suitable learning strategy [5].

The problem of striking a balance between the global and local search tools in a hybrid, in order to adapt the algorithm to a given problem, can be viewed as a problem of finding optimal control parameter settings. Different techniques can be used to monitor the behavior of a HGA to dynamically adapt its control parameters to improve search performance. Evolutionary self-adaptation is one possible way to implicitly determine the exploration versus exploitation trade-off.

A. Ant colony optimisation

Ant-based optimization algorithms are bio-inspired population-based optimization techniques that have been applied to solve hard optimization problems. They simulate the collective behavior of ants, which exchange information using a simple form of indirect communication mediated by pheromone formation, known as stigmergy [6]. This form of communication plays a crucial role in finding the shortest path between nest and food sources and enables adaptation to changes in the environment.

Ant colony optimization (ACO) algorithms are very effective in solving discrete optimization problems such as the travelling salesperson problem (TSP), the quadratic assignment problem, vehicle routing, sequential ordering, graph coloring, e-learning presentation problem and routing in communications networks [7]. They have also been amended in order used to solve real-parameter optimization problems [8] [9]. In the Aggregation Pheromone System (APS) [10] and its enhanced version (eAPS) [11], the pheromone trail was replaced by an aggregation pheromone, whose density was represented by a mixture of multivariate normal distributions. The ACO_R [12] and the DACO_R [13] algorithms are also proposed for optimizing continuous domains, in which a solution archive is used as a form of pheromone trail for the derivation of a probability distribution over the search space. A sampling method for discretizing the continuous search space and an incremental solution construction method based on the sampled values is incorporated in SamCo to apply ACO to continuous optimization problems [14].

B. Ant colony optimisation and genetic algorithms

Genetic algorithms and ant colony optimization algorithms can be combined to improve the combination's performance in different ways. However, most of the proposed hybrids use only three different forms of hybridization. The first set of hybrids is based on viewing the genetic algorithm as a global search method and the ant colony algorithm as a local search method. The second set is based on the ability of the genetic algorithms to adapt the control parameters of other techniques. In the last form of intergeneration, some genetic concepts and operators are incorporated into ant colony optimization algorithms.

Incorporating an ant colony optimization algorithm as a local search method within a genetic algorithm can improve the search performance. An ant colony model for continuous search spaces has been used to improve the quality of the solutions produced by a genetic algorithm [8]. The optimal results from the ACO algorithm has been used as an initial population for a genetic algorithm to avoid being trapped in a local optimum. The GA-ACO algorithm [15] is used to enhance the performance of genetic algorithm (GA) by incorporating ACO as a local search method to solve the multiple sequence alignment problems.

The search capabilities of both algorithms can be combined through executing them in parallel and exchanging solutions whenever an improved potential solution is found [16]. The same idea has been used to combine a special class of ACO entitled Adjacent Pairwise Interchange (API) and a GA to optimize continuous optimization problems [17].

Since genetic algorithms are in practice a very effective optimization technique, they have been incorporated within ant colony optimization algorithms to optimize their control parameters, which are characterized by being highly problem specific and dependent on the required solution accuracy [18]. The ASGA algorithm [19] was the first algorithm that used a genetic algorithm to evolve the control parameters of an ant-based algorithm. A genetic algorithm can be applied to optimize the control parameters of ant colony optimization algorithms in a variety of ways [20] [21] [18] [22].

Genetic models and operators can be incorporated in many ways into ant colony optimization algorithms to improve their performance. Different selection mechanisms, fitness ranking and mutation operations that are used in genetic algorithms were implemented and tested with an ant colony optimization algorithm to solve optimization problems [23] [24] [25].

II. ANT OPTIMIZATION AND HYBRID SELF-ADAPTATION

The success of a hybrid algorithm in solving a given problem efficiently depends on its success in achieving a balance between exploration and exploitation [26]. The appropriate balance of exploration and exploitation required for good performance can be achieved by finding an optimal set of the hybrid's control parameters for a given problem. The use of a mechanism to dynamically identify the effectiveness of different genetic and local operators and learning strategies for the problem currently being worked on can help to improve the hybrid's performance. Knowledge about previous and current solutions, the operators, and the learning strategies used can be utilized as a base to identify the strengths and the weakness of these operators and strategies.

The basic idea of the proposed hybrid is that, as the search progresses, the effectiveness of the genetic operators, the local search method, the duration of local search, and the learning strategies, on the performance of a hybrid genetic algorithm in dealing with the current problem can be learnt by using an ant-based algorithm as a reinforcement learning approach. The pheromone trail metaphor can be used to accumulate the experience of the past solutions on the efficiency and the effectiveness of using different operators to find a solution of the current problem.

Pheromone trail behavior can be applied to solve the problem of dynamically adjusting the probabilities of using the different genetic and local operators and learning strategies.

A population or a colony of ants collectively searches for a sequence of genetic operations, local search operator with a suitable duration, and a learning strategy, that produces an effective and efficient solution to the problem under consideration. The search space and the neighborhood notion of the problem of adapting the performance of a genetic-local hybrid to a given problem can be viewed as shown in Figure 1. Each ant performs a sequence of local moves between the different states of its search task in order to find a sequence of operations that improves the solutions in an efficient way. Each state of the problem's search space has a complementary state. For example, the complementary state of the "crossover" state is the "no crossover" state, while the complementary of the "Lamarckian" state is the "Baldwinian" state.

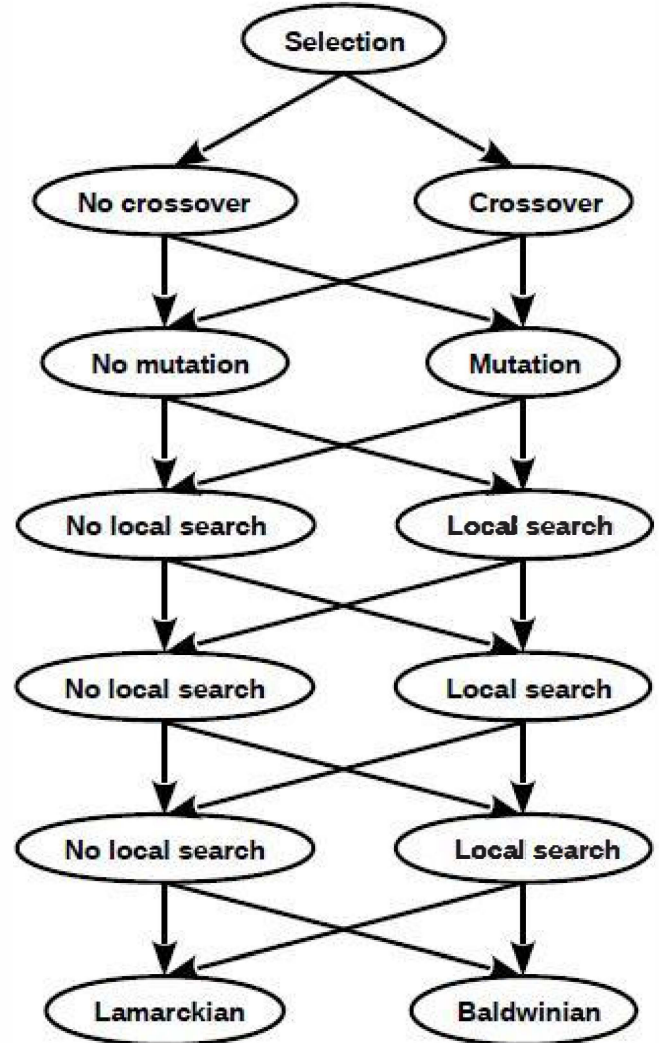


Fig. 1. The search space and the neighborhood notion

An ant moves through adjacent states starting from the selection state, which is the only state without a complementary state, and ending its tour with either the "Lamarckian" or the "Baldwinian" state. Each movement of an

ant is accompanied by performing one of the two alternatives on a solution. The path that is followed by an ant defines the sequence of genetic operations, local operation, its duration, and the learning strategy that is applied on a solution. This path is assigned a merit score, which is equal to the fitness improvement in the solution as a result of performing this sequence of operations.

At the end of the ant's tour, it releases an amount of pheromone on the edges of the path it used to build a solution based on the merit score of the tour. The density of pheromone on the paths that lead to high improvements in fitness will be higher than the others which lead to less or no improvement. This change will probably encourage other ants to follow the paths with a high density of pheromone. As a result, the sequence of genetic and local operators and learning strategies that lead to solution improvements will be preferred by most of the new candidate solutions. These preferred sequences of operations will be dynamically built based on the fitness landscape of a given problem. This mechanism can promote competition amongst the different operators and learning strategies based on its ability to improve the fitness. It can also promote cooperation between the different operators and learning strategies in order to discover more effective sequences of operations. This technique can produce a hybrid genetic algorithm that is able to adapt itself to a given problem without the need for external control.

An ant selects the next state from its adjacent states using a probabilistic decision policy. An ant decides to move from its current state to one of the available next two states, which will be referred to as the Next Do state or the Next Alternative state. The decision policy, as given in Equation 1, is based on the density of pheromone on the two branches that connect the current state to these states.

$$P_{C-NDo} = \frac{t_{C-NDo}}{t_{C-NDo} + t_{C-NAIter}} \quad (1)$$

$$P_{C-NAIter} = 1 - P_{C-NDo} \quad (2)$$

where P_{C-NDo} is the probability of moving from the current state to the Next Do state, $P_{C-NAIter}$ is the probability of moving from the current state to the Next Alternative state, t_{C-NDo} is the trail density on the edge connecting the current state to the Next Do state, and $t_{C-NAIter}$ is the trail density on the edge connecting the current state to the Next Alternative state.

Initially all the edges of the possible operations paths are assigned an equal trail density. Therefore, all the adjacent states have an equal opportunity to be visited. All the ants start from the selection state. After the genetic algorithm performs the selection operation, each ant is randomly assigned an individual from the mating pool. That ant will decide on the sequence of operations that the individual should perform. The decision is taken locally based on the current ant's state and using the decision policy given in Equation 1. The ant's tour ends by choosing one of the available learning strategies. At the end of that tour, the ant deposits an amount of pheromone

on the edges of the path it followed. The amount of pheromone deposited is made equal to the improvement in the fitness of the associated solution. This pheromone can induce the ants towards promising search regions of effective sequences of operators. The change in trail density on each edge of the followed path is given by Equation 3.

$$\Delta\tau_{(i,C-N)} = \begin{cases} \Delta fitness_i & \text{if } \Delta fitness > 0 \\ 0 & \text{otherwise} \end{cases} \quad (3)$$

where $\Delta t_{(i,C-N)}$ is the change in the trail density of the edge connecting state C to state N as a result of following the path constructed by ant i . The aim of rewarding sequences of operations that produce an improvement in fitness and not penalizing paths that reduce the performance is to encourage exploration of the search space.

The algorithm updates the pheromone at the end of the ant colony iteration (i.e. after every member of the colony has completed its tour), which is known as offline updating, to bias the search from a global perspective. The amount of pheromone on the edge connecting the C state with the N state after updating it according to the results of the ant tour is given by Equation 4.

$$t_{(i+1,C-N)} = t_{(i,C-N)} + \Delta t_{(i,C-N)} \quad (4)$$

III. EXPERIMENTS

For the purpose of evaluating the proposed Ant-based Self-Adaptive Hybrid Genetic (AntSAHG) algorithm, its performance was compared with an Evolutionary Self-Adaptive Hybrid Genetic (ESAHG) algorithm, which uses evolution to select the genetic operators, the local operator, its duration and the learning strategy that should be performed on each individual. For each operation (strategy), a digital bit is encoded into the individual's genetic structure which determines whether to perform that operation (use that strategy) or its alternative. The duration of local search is represented by two bits that specify the number of local iterations.

The quality of the solutions produced by each algorithm was used as the main measure of the algorithm's performance. The percentage of experiments that converged to the global optimum was used as an indication of the ability of the algorithms to produce high quality solutions. The performance of the two algorithms is compared in terms of the speed of finding a global optimum. These algorithms were also evaluated in terms of their ability to adapt to different fitness landscapes and population sizes.

A set of test functions has been chosen to evaluate the performance of the two self-adaptive algorithms. Three test functions have been used as a test suite. This suite of test functions includes the 20-dimensional ellipsoidal and the 20-dimensional Rastrigin.

The GA used is a simple elitist genetic algorithm with binary tournament selection, a two-point crossover operator, and a mutation operator with a probability of 0.01. The hybrids

use a local search operator, which combines the steepest descent method and Brent's method to estimate the best step size to climb to the local optimum from the current position in the basin of attraction [27]. An adaptive initial step size based on the changes in the standard deviation of the population fitness was used. The use of an adaptive step size can add an exploring role to the local search method at the early stages of search. The Baldwinian search can benefit from the adaptive initial step size since it can improve the genetic sampling ability at the early stages and can combat the hindering effect as the search approaches the fitness-convergence-state.

Population sizes of 100, 150, 200, 250 and 300 were used to optimize the test functions using the two self-adaptive hybrids. Each variable was represented by a string of 10 bits. The stopping criterion for all experiments was a maximum number of function evaluations. The value of this parameter was set to 5000 times the population size. Each experiment was repeated 50 times.

In the AntSAHG algorithm, the number of ants was set equal to the number of individuals in the genetic population. The amount of pheromone released was made equal to the fitness improvement, as given in Equation 4, in order to ensure fair comparison with the evolutionary self-adaptive technique, which uses the individual's fitness to assess the effectiveness of a control parameter in solving a given problem. However, the ant algorithm divides the whole tour into two stages. The first one is the genetic stage, where the ants decide on the genetic operators to apply. The second one is the learning stage, where the ants decide on the local operator, its duration, and the learning strategy. In order to evaluate these stages fairly, each stage is evaluated separately due to the big differences in the number of function evaluations used.

Instead of the evaporation mechanism, the AntSAHG algorithm adaptively modifies the trail density of all the possible paths to ensure that the probability of each of the alternative operations or strategies does not exceed a specific threshold. In the case of exceeding this threshold an equal amount of pheromone is added to all the possible paths. Initially all the edges of the possible operations paths are assigned an equal trail density which was set to the absolute value of the average fitness of the initial genetic population.

A. Search effectiveness and efficiency

The percentages of times each hybrid algorithm found a global optimum using different population sizes were compared. These percentages were used to evaluate the effectiveness of the two self-adaptive mechanisms in solving the test problems.

The results of the experiments conducted on the ellipsoidal test function showed that both the self-adaptive techniques were able to find the exact global optimum of the function in every experiment. This observation clearly shows the effectiveness of both algorithms in solving this type of problem, which can obstruct the self-adaptive ability of the Baldwinian search. The combination of the adaptive initial step size of local search and adaptive ability of the two hybrids can explain the improvement in the hybrids' performance. The results of comparing the speed of finding the global optimum

of the ellipsoidal function, as shown in Figure 2, show that the AntSAHG algorithm was slightly faster than the ESAHG algorithm.

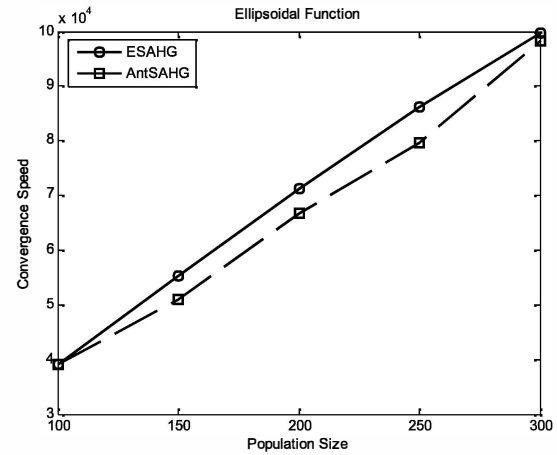


Fig. 2. The speed of the AntSAHG and ESAHG algorithms in solving the Ellipsoidal problem.

Figure 3 compares the percentage of times that the two self-adaptive hybrid algorithms found the global optimum of the Schwefel function. The graphs show that the AntSAHG hybrid algorithm was able to find the global optimum of the Schwefel function more often than the ESAHG hybrid algorithm. This can be explained by the fact that the ESAHG hybrid algorithm uses the fitness of an individual for evaluating both the quality of solutions and effectiveness of different operators and learning strategies in producing these solutions. On the other hand, the AntSAHG uses the improvements in fitness to judge the effectiveness of operators and strategies and at the same time the genetic algorithm uses the fitness to evaluate the solution's quality. The AntSAHG hybrid algorithm can easily discriminate between sequences of operators that improve the performance and sequences that do not improve it, whereas the ESAHG algorithm cannot distinguish between them.

Figure 4 compares the convergence speed with the global optimum of the Schwefel function of both self-adaptive algorithms. These graphs show that the ESAHG algorithm was much faster than the AntSAHG algorithm in finding the global optimum of this function. However, the AntSAHG was able to find that optimum more frequently than the ESAHG algorithm.

The AntSAHG algorithm outperformed the ESAHG algorithm on the Rastrigin function in terms of the percentage that converged to the global optimum using different population sizes, as depicted in Figure 5. The ESAHG was unable to find the global optimum of the Rastrigin function in most of the experiments, in contrast to the AntSAHG algorithm. This difference can be explained by the fact the evolutionary self-adaptive behavior can lead to the disappearance of useful genes of some of the control parameters and the algorithm can face some difficulties in restoring them. However, by ensuring that the probability of selecting one of the two alternatives does not exceed a threshold value in the AntSAHG algorithm, there is always a chance to select operations that do not improve the solutions'

fitnesses. This characteristic enables the AntSAHG to escape local optima and improves its ability to recover from premature convergence. The nature of the fitness landscape of the Rastrigin function, where the optima are close to each other, makes the AntSAHG algorithm able to recover from premature convergence. However, such a recovery is more difficult in the case of the Schwefel function, where the second best optimum is far from the global optimum, once the whole population has converged to a non-global optimum.

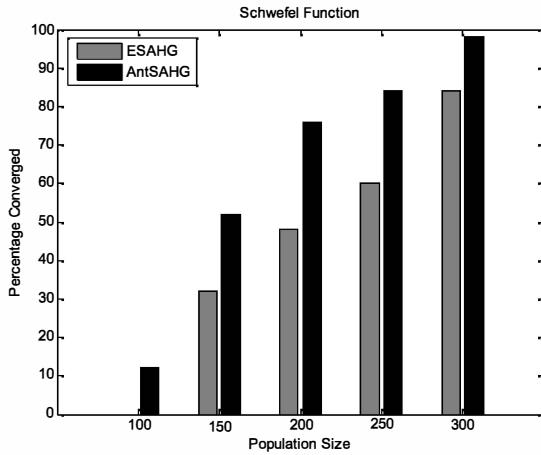


Fig. 3. Finding the global optimum of the Schwefel function.

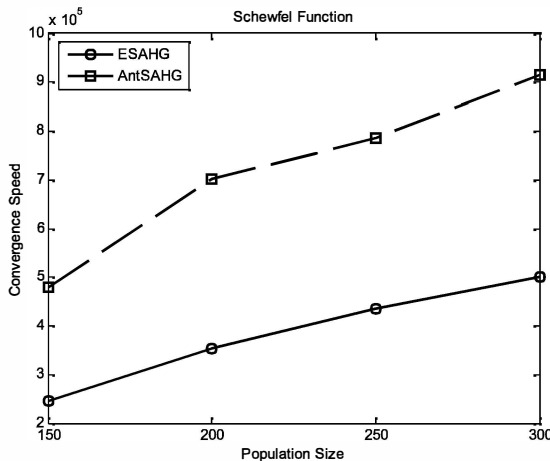


Fig. 4. The speed of solving the Schwefel function.

IV. CONCLUSION

The experiments conducted clearly show the effectiveness of the pheromone trail metaphor to utilize the experience of the past solutions for on-line learning of the effectiveness of the different combinations of operators and learning strategies in solving a given problem. The Ant-based self-adaptive mechanism was able find high-quality solutions for the test problems. It outperformed the evolutionary self-adaptive algorithm in terms of the solution quality and the convergence speed. The experiments suggest the suitability of the AntSAHG algorithm for dynamic environments.

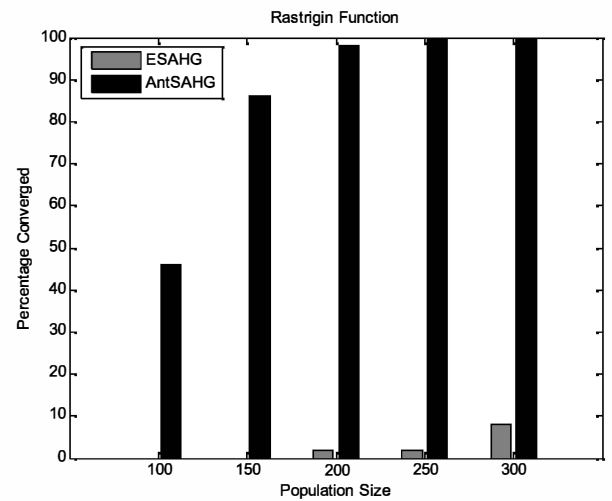


Fig. 5. Finding the global optimum of the Rastrigin function.

REFERENCES

- [1] D. E. Goldberg, *Genetic Algorithms in Search, Optimization, and Machine Learning*, Addison-Wesley, 1989.
- [2] D. Beasley, D. R. Bull and R. R. Martin, "An Overview Of Genetic Algorithms : Part 1, Fundamentals," *University Computing*, vol. 2, no. 15, pp. 58-69, 1993.
- [3] T. A. El-Mihoub, A. Hopgood, L. Nolle and A. Battersby, "Hybrid Algorithms : A review," *Engineering Letters*, vol. 3, no. 2, pp. 12-45, 2006.
- [4] T. A. El-Mihoub, A. Hopgood and I. A. Aref, "Accelerating Genetic Schema Processing Through Local Search," in *International Conference on Computer, Control, Informatics and its Application*, Jakarta, 2013.
- [5] T. A. El-Mihoub, A. Hopgood, I. Nolle and A. Battersby, "Performance of hybrid genetic algorithms incorporating local search," in *18th European Simulation Multiconference*, Magdeburg, 2004.
- [6] G. Theraulaz and E. Bonabeau, "Brief History of Stigmergy," *Artificial Life*, vol. 5, no. 2, pp. 97-116, 1999.
- [7] V. M. Dorigo and T. Stützle, "Ant Colony Optimization: Overview and Recent Advances".
- [8] G. Bilchev and I. C. Parmee, "The Ant Colony Metaphor for Searching Continuous Design Spaces," in *Lecture Notes In Computer Science Selected Papers from AISB Workshop on Evolutionary Computing*, 1995.
- [9] K. Socha, "ACO for Continuous and Mixed-Variable Optimization," in the *Fourth International Workshop on Ant Colony Optimization and Swarm Intelligence (ANTS 2004)*, 2004.
- [10] S. Tsutsui, "Ant Colony Optimisation for Continuous Domains with Aggregation Pheromones Metaphor," in the *5th International Conference on Recent Advances in Soft Computing*, 2004.
- [11] S. Tsutsui, "An Enhanced Aggregation Pheromone System for Real-Parameter Optimization in the ACO Metaphor,," in *ANTS Workshop*, 2006.
- [12] K. Socha and M. Dorigo, "Ant colony optimization for continuous domains," *European Journal of Operational Research*, vol. 185, no. 3, p. 1155-1173, 2008.
- [13] G. Leguizamón and C. Coello, "An alternative ACOR algorithm for continuous optimization problems," in *Proceedings of the Seventh International Conference on Swarm intelligence*, Berlin, 2010.
- [14] H. Xiao-Min, Z. Jun, H.-H. Chung, Y. Li and O. Liu, "SamACO: Variable Sampling Ant Colony Optimization Algorithm for Continuous Optimization," *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics*, vol. 40, no. 6, pp. 1555-1566, 2010.
- [15] Z.-J. Lee, S.-F. Su, C.-C. Chuang and K.-H. Liu, "Genetic algorithm with

ant colony optimization (GA-ACO) for multiple sequence alignment," *Applied Soft Computing*, vol. 8, no. 1, pp. 55-78, 2008.

- [16] S. Nemati, M. Basiri, N. Ghasem-Aghaee and M. H. Aghdam, "A novel ACO-GA hybrid algorithm for feature selection in protein function prediction.," *Expert Systems with Applications*, vol. 36, no. 10, 2009.
- [17] I. Ciornei and E. Kyriakides, "Hybrid Ant Colony-Genetic Algorithm (GAAP) for Global Continuous Optimization," *Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on*, vol. 42, no. 1, pp. 234-245, 2012.
- [18] D. Caertner, Clark and L, "On Optimal Parameters for Ant Colony Optimization Algorithms," in the 2005 International Conference on Artificial Intelligence, ICAI 2005, 2005.
- [19] T. White, B. Pagurek and F. Oppacher, "ASGA: Improving the Ant System by Integration with Genetic Algorithms.," in *The third Conference on Genetic Programming (GP/SGA'98)*, 1998.
- [20] H. M. Botee and E. Bonabeau, "Evolving Ant Colony Optimization," *Adv. Complex Syst*, vol. 1, no. 2, pp. 149-159, 1998.
- [21] M. L. Pilat and T. White, "Using Genetic Algorithms to Optimize ACS-TSP," in the *Third International Workshop on Ant Algorithms*, 2002.
- [22] B. Garro, H. Sossa and R. Vazquez, "Evolving ant colony system for optimizing path planning in mobile robots," in *Electronics, Robotics and Automotive Mechanics Conference*, 2007.
- [23] Y. Semet, E. Lutton and P. Collet, "Ant Colony Optimisation for E-Learning: Observing the Emergence of Pedagogic Suggestions," in *IEEE Swarm Intelligence Symposium*, 2003.
- [24] S. Tsutsui, M. Pelikan and A. Ghosh, "Performance of Aggregation Pheromone System on Unimodal and Multimodal Problems," in the 2005 IEEE CEC, 2005.
- [25] N. Zhao, S. Li and Z. Wu, "Cognitive radio engine design based on ant colony optimization," *Wireless Personal Communications*, vol. 65, no. 1, pp. 15-24, 2012.
- [26] A. A. Hopgood, *Intelligent Systems for Engineers and Scientists*, 3rd ed., CRC Press, 2012.
- [27] W. Press, S. Teukolsky, W. Vetterling and B. Flann, *Numerical Recipes In C*, Cambridge University Press, 1993.