Journal of Advanced Computer Science and Technology Research, Vol.5 No.4, December 2015, 123-132

Hybrid Cuckoo Search - Genetic Algorithm (CSGA): An Approach to Solve Some Combinatorial Problems

Sura Mazin Ali
Software Eng. Department
Baghdad College of Economic Sciences University
Baghdad, Iraq
drahmaed_tark@yahoo.com

ISSN: 2231-8852

ABSTRACT

Combinatorial problems involve finding a grouping, ordering, or assignment of a discrete, finite set of objects that satisfies given conditions. This paper presents a hybrid approach to solve some combinatorial problems using cuckoo search and genetic algorithm. The proposed approach obtains a good result compare with the classical cuckoo search and standard genetic algorithm. Two problems have been solved, Minimum Spanning Tree and Traveling Salesman Problem. The experimental results show that the suggested approach is very suitable for solving these two problems.

Keywords: **Intelligent Swarm-Based Algorithm, Cuckoo Search, Genetic Algorithm, Combinatorial Problem, Traveling Salesman Problem and Minimum Spanning Tree Problem.**

1. Introduction

In applied mathematics and theoretical computer science, combinatorial optimization is a topic that consists of finding an optimal object from a finite set of objects (Schrijver, 2013). In many such problems, exhaustive search is not feasible. It operates on the domain of those optimization problems, in which the set of feasible solutions is discrete or can be reduced to discrete, and in which the goal is to find the best solution. Some common problems involving combinatorial optimization are the traveling salesman problem ("TSP") and the minimum spanning tree problem ("MST").

Combinatorial optimization is a subset of mathematical optimization that is related to operations research, algorithm theory, and computational complexity theory. It has important applications in several fields, including artificial intelligence, machine learning, mathematics, auction theory, and software engineering.

Some research literature considers discrete optimization to consist of integer programming together with combinatorial optimization (which in turn is composed of optimization problems dealing with graph structures) although all of these topics have closely intertwined research literature. It often involves determining the way to efficiently allocate resources used to find solutions to mathematical problems (Megiddo, 2011).

However, combinatorial problems have very wide range of application and areas such as Assignment problem, Closure problem, Constraint satisfaction problem, Cutting stock problem, Integer programming, Knapsack problem, Minimum spanning tree, Nurse scheduling problem, Traveling salesman problem, Vehicle routing problem, Vehicle rescheduling problem, Weapon target assignment problem and so on.

This paper presents a hybrid tool to solve some combinatorial problems, which is called Cuckoo Search - Genetic Algorithm (CSGA). CSGA is an intelligent swarm-based algorithm which depends on two intelligent tools Cuckoo Search and Genetic Algorithm. The intelligent swarm-based algorithm will be described in section 2. Cuckoo Search details in section 3. Section 4 consist of genetic algorithm. The proposed hybrid algorithm will be presented in section 5. Section 6 contains the experiments of solving 3 problems using proposed algorithm. Section 7 consists of conclusions that are related to suggested algorithm.

2. Intelligent Swarm-Based Algorithm

Swarm-based algorithms mimic nature's methods to drive a research towards the optimal solution. A key difference between Swarm-based algorithms and direct search algorithms such as hill climbing and random walk is that Swarm-based algorithms use a population of solutions for every iteration instead of a single solution. As a population of solutions is processed in an iteration, the outcome of each iteration is also a population of solutions. If an problem has a single optimum solution, Swarm-based algorithm population members can be expected to converge to that optimum solution (Pham et al. 2006).

However, if an NP-hard problem has multiple optimal solutions, a Swarm-based algorithm can be used to capture them in its final population. Swarm-based algorithms include the Ant Colony Optimization (ACO) algorithm (Dorigo & Stutzle, 2004), the Cuckoo Search (Yang & Deb, 2009), the Genetic Algorithm (GA) (Goldberg, 1989), the Bees Algorithm (BA) (Pham et al., 2005), and the Particle Swarm Optimization (PSO) algorithm (Eberhart et al. 2001).

Common to all population-based search methods is a strategy that generates variation in the solution being sought. Some search methods use a greedy criterion to decide which generated solution to retain. Such a criterion would means accepting a new solution if and only if it increases the value of the objective function. A very successful non-greedy population-based algorithm is the ACO algorithm which emulates the behavior of real ants. Ants are capable of finding the shortest path from the food source to their nest using a chemical substance called pheromone to guide their search. The pheromone is deposited on the ground as the ants move and the probability that a passing stray ant will follow this trail depends on the quantity of pheromone laid (Pham et al. 2006).

The Genetic Algorithm is based on natural selection and genetic recombination. The algorithm works by choosing solutions from the current population and then applying genetic operators - such as mutation, crossover (Goldberg, 1989), controlled mutation and conjugation (Sadeq, 2000) - to create a new population. The algorithm efficiently exploits historical information to speculate on new search area with improved performance (Goldberg, 1989).

The successful applications of the Ant Systems in the complex engineering and management problems are certainly encouraging. At the same time, these successes act as a great inspiration to attempt to explore bees' behavior as a source of ideas and models for development of various artificial systems (Montgomery, 2005). The highly organized behavior enables the colonies of insects to solve problems beyond capability of individual members by functioning collectively and interacting primitively amongst members of the group. In honey bee colonies, this behavior allows honey bees to explore the environment in search of flower patches (food sources) and then indicates the food source to the other bees of the colony when they return to the hive. Such a colony is characterized by self-organization, adaptation and robustness (Chin et al., 2006).

Particle Swarm Optimization (PSO) algorithm is an optimization procedure based on the social behavior of groups of organization, for example the flocking of birds or the schooling of fish (Eberhart et al. 2001). Individual solutions in a population are viewed as "particles" that evolve or change their positions with time. Each particle modifies its position in search space according to its own experience and also that of a neighboring particle by remembering the best position visited by itself and its neighbors, thus combine local and global search methods (Eberhart et al. 2001).

3. Cuckoo Search Algorithm

CS is a heuristic search algorithm which has been proposed recently by Yang and Deb (Yang & Deb, 2009). The algorithm is inspired by the reproduction strategy of cuckoos. At the most basic level, cuckoos lay their eggs in the nests of other host birds, which may be of different species. The host bird may discover that the eggs are not its own and either destroy the egg or abandon the nest all together. This has resulted in the evolution of cuckoo eggs which mimic the eggs of local host birds. To apply this as an optimization tool, Yang and Deb used three ideal rules (Yang & Deb, 2009), (Zheng & Zhou, 2012) :

(1) Each cuckoo lays one egg, which represents a set of solution co-ordinates, at a time and dumps it in a random nest;

(2) A fraction of the nests containing the best eggs, or solutions, will carry over to the next generation;

(3) The number of nests is fixed and there is a probability that a host can discover an alien egg. If this happens, the host can either discard the egg or the nest and this result in building a new nest in a new location. Based on these three rules, the basic steps of the Cuckoo Search (CS) can be summarized as the pseudo code shown as in Fig. 1.

Cuckoo Search via Levy Flight Algorithm
Input: Population of the problem;
Output: The best of solutions;

```
    Objective function f(x), x = (x1, x2, ...xd)^T
    Generate initial population of n host nests xi
                                  (i = 1, 2, ..., n)
    While (t <Max Generation) or (stop criterion)
         Get a cuckoo randomly by Levy flight
         Evaluate its quality/fitness Fi
         Choose a nest among n(say,j)randomly
         If (Fi > Fj) replace j by the new solution;
         A fraction(pa) of worse nests are abandoned and new ones are
          built;
         Keep the best solutions (or nests with quality solutions);
         Rank the solutions and find the current best;
         Pass the current best solutions to the next generation;
    End While
```

Fig (1) Basic Cuckoo Search Algorithm

When generating new solution $x^{(t+1)}$ for, say cuckoo i, a Levy flight is performed

$x^{(t+1)}_i = x(t)_i + \alpha \oplus Levy(\beta)$ ……. (1)

where $\alpha > 0$ is the step size which should be related to the scales of the problem of interests. In most cases, we can use $\alpha = 1$. The product $\oplus$ means entry-wise walk while multiplications. Levy flights essentially provide a random walk while their random steps are drawn from a Levy Distribution for large steps

$Levy \sim u = t^{-1-\beta}$ $(0 < \beta \leq 2)$ ……… (2)

this has an infinite variance with an infinite mean. Here the consecutive jumps/steps of a cuckoo essentially form a random walk process which obeys a power-law step-length distribution with a heavy tail. In addition, a fraction *pa* of the worst nests can be abandoned so that new nests can be built at new locations by random walks and mixing. The mixing of the eggs/solutions can be performed by random permutation according to the similarity/difference to the host eggs.

4. Genetic Algorithms

Genetic Algorithms (GA) are search algorithms based on the mechanics of natural selection and natural genetics. They combine survival of the fittest among string structures with a structured yet randomized information exchange to form a search algorithm with some of the innovative flair of human search. In every generation, a new set of artificial creatures (strings) is created using bits and pieces of the fittest of the old; an occasional new part is tried for good measure, while randomized, genetic algorithms are no simple random walk. They efficiently exploit historical information to speculate on new search points with expected improved performance (Goldberg, 1989).

Genetic algorithms have been developed by John Holland in the 1970s (Holland, 1975), his colleagues, and his students at the University of Michigan. The goals of their research have been (1) to abstract and rigorously explain the adaptive processes of natural systems, and (2) to design artificial software systems that retain the important discoveries in both natural and artificial science systems (Goldberg, 1989). The simple form of GA involves three types of operators : selection, crossover and mutation (Michell, 1998):

Selection : This operator selects chromosomes in the population for reproduction. The fitter the chromosome, the more times it is likely to be selected to reproduce. There are several selection methods used in this operator such as roulette wheel, sigma scaling, Boltezman, elitism, rank tournament, steady-state …etc (Sadeq, 2000).

Crossover : This operator randomly chooses a locus and exchanges the subsequences before and after that locus between two chromosomes to create two offspring. For example, the string 10000100 and 11111111 could be crossed over after the third locus in each to produce the two offspring 10011111 and 11100100. The crossover operator roughly mimics biological recombination between two single-chromosome organisms. The position of classical crossover operator is either one or two positions. In (Sadeq, 2000) there is a survey of several types of crossover operator such as crowding, fitness sharing function, restricted mating, spatially restricted mating, dominance, reordering, inversion, segregation, migration, translocation, duplication, deletion, pre-selection, partially matched, order crossover, cycle crossover, very greedy crossover …etc. Also in (Sadeq, 2000) conjugation operator is proposed as an operator in GA instead of crossover operator.

Mutation : This operator randomly flips some of the bits in a chromosome. For example, the string 00000100 might be mutated in its second position to yield 01000100. Mutation can occur at each bit position in a string with some probability, usually small (Sadeq, 2000).

Figure (2) shows the classical approach of GA steps.

1. Generate randomly chromosomes.
2. Calculate the fitness function value of each chromosome.
3. Repeat
4. Select pair of chromosomes.
5. Crossover the pair to produce the two offspring.
6. Mutate the two offspring.
7. Until (Goal or Max. generation).

Fig.2: Pseudo code of the Genetic Algorithm

5. Proposed Approach: Cuckoo Search Based on Genetic Algorithm

The diversity of population in the cuckoo search is less than in the genetic algorithm. It depends on the random solution rather than on the behavior of algorithm. While the diversity in the genetic algorithm is good via the crossover in the first degree with mutation in the second degree. Therefore, these two important operators will be use in the cuckoo search algorithm to increase the diversity of this algorithm. The increasing of diversity in the cuckoo search algorithm will be increase the performance of this technique, get more perfect solutions and decrease the time in several cases. The proposed approach mixes between Cuckoo Search Algorithm and Genetic Algorithm to increase the performance of Cuckoo Search Algorithm firstly, and to increase the diversity in it secondly. The proposal uses the main operators of Genetic Algorithm (crossover and mutation) in the Cuckoo Search Algorithm as a tool to increase the diversity in the Cuckoo Search algorithm. Furthermore, the proposal approach suggested using several types of crossover operators such as (single-point crossover, crowding crossover, segregation and migration) and conjugation operator. In another step the proposal approach uses mutation operator if the neighborhood sites resulting from crossover or conjugation operator are not enough. Figure (3) shows the pseudo code of the Cuckoo Search-Genetic Algorithm.

The two evolutionary operators, crossover and mutation will provide a good diversification for the solution population. Crossover and mutation work only when the new solution is worse than the current solution, therefore these two operators will be work as solution for local minimum problem. Furthermore, these two operators provide a new two solutions (cuckoos) for the population, causing more diversity for the population of solutions.

```
Cuckoo Search – Genetic Algorithm with Levy Flight Selection
Input: Population of the problem;
Output: The best of solutions;
   Objective function f(x), x = (x1, x2, ...xd)^T
   Generate initial population of n host nests xi
                                (i = 1, 2, ..., n)
   While (t <Max Generation) or (stop criterion)
       Get a cuckoo randomly by Levy flight
       Evaluate its quality/fitness Fi
       Choose a nest among n(say,j)randomly
       If (Fi > Fj) replace j by the new solution;
       Else
              Begin
                     Get another cuckoo randomly by Levy flight;
                     Make Crossover and Mutation Operator (for the random two cuckoos);
                     Reproduce new two solutions;
                     Put the two new solutions in the current population;
              end
       A fraction(pa) of worse nests are abandoned and new ones are built;
       Keep the best solutions (or nests with quality solutions);
       Rank the solutions and find the current best;
       Pass the current best solutions to the next generation;
   End While
```

Fig.3 : Pseudo code of the Proposal of basic Cuckoo Search Genetic Algorithm

In addition to the single-point crossover, the suggested proposal approach uses several types of crossover operator such as (crowding crossover, segregation and migration) and conjugation operator. In the following point these operators will be illustrated.

1- Crowding Operator : De Jong (1975), experimented with a “crowding” operator in which a newly formed offspring replaces the existing individual most similar to itself. This prevents too many similar individuals “crowds” from being in the population at the same time.
2- Segregation Operator : To see how segregation works, we simply imagine the process of gamete formation when we have more than one chromosome pair in the genotype. Crossover occurs as before; however, when we go to form a gamete and randomly select on of each of the haploid chromosomes, this random selection process, known as segregation, effectively disrupts any linkage that might exist between genes on different chromosomes (Holland, 1975).
3- Migration Operator : Grosso (Holland, 1975) maintained a biological orientation in his study of migration operator. The migration operator complemented roughly some bits of selected individuals in the population based on the application.
4- Conjugation Operator : in medicine we can define conjugation operator as (Jane & Alxander, 1982), (Jawetz, 1998) “the formation of the intermediate bridge (conjugation tube), which will transfer the DNA of the donor cell to the recipient cell.” This operator suggested in (Sadeq, 2000).

6. Cuckoo-Genetic Algorithm: A Tool to Solve Combinatorial Problems

In this paper, two combinatorial problems have been solved using the proposed Cuckoo Search - Genetic Algorithm which are Minimum Spanning Tree and Traveling Salesman Problem. As experimental results, this paper compares its approach with Genetic Algorithm and Cuckoo Search alone.

6.1 Minimum Spanning Tree (MST)

Given undirected graph G with positive edge weights (connected), the goal is finding a minimum weight set of edges that connects all of the vertices. This problem is fundamental with diverse applications such as Network design (telephone, electrical, hydraulic, TV cable, computer, road), Approximation algorithms for NP-hard problems such as (traveling salesperson problem and Steiner tree), Indirect applications such as (maximum bottleneck paths, image registration with entropy, Learning salient features for real-time face verification, reducing data storage in sequencing amino acids in a protein), Cluster analysis and so on (Cormen et al. 2001).

Several experiments have been executed to find MST for the different tress within 20, 40, 60, 80 and 100 nodes. Table (1) illustrates samples of experimental results.

Table (1): Experimental Results of MST Problem

No. of Nodes	Optimal MST	Method	Avg. of MST	Avg. Time (Sec.)	Best Iteration
20	71	GA	79	2.2	38
		CS	88	1.3	44
		CSGA (Crossover)	73	2.1	39
		CSGA (Conjugation)	83	2.3	52
		CSGA (Crowding)	81	2.4	37
		CSGA (Segregation)	92	2.3	39
		CSGA (Migration)	90	2.2	37
40	127	GA	140	4.1	51
		CS	152	3.2	44
		CSGA (Crossover)	134	3.7	40
		CSGA (Conjugation)	150	3.8	46
		CSGA (Crowding)	144	3.7	47
		CSGA (Segregation)	148	3.9	45
		CSGA (Migration)	146	3.8	48
60	231	GA	241	7.2	90
		CS	256	5.7	81
		CSGA (Crossover)	238	6.2	88
		CSGA (Conjugation)	265	6.6	98
		CSGA (Crowding)	253	6.5	86
		CSGA (Segregation)	255	6.4	87
		CSGA (Migration)	262	6.3	92
80	352	GA	270	10.7	112
		CS	287	7.9	127
		CSGA (Crossover)	362	8.2	115

		CSGA (Conjugation)	290	8.4	128
		CSGA (Crowding)	271	8.3	126
		CSGA (Segregation)	272	8.5	128
		CSGA (Migration)	277	8.7	133
100	486	GA	512	13.2	150
		CS	534	11.6	171
		CSGA (Crossover)	498	12.1	165
		CSGA (Conjugation)	542	12.3	172
		CSGA (Crowding)	521	12.5	167
		CSGA (Segregation)	523	12.4	159
		CSGA (Migration)	531	12.6	169

6.2 Traveling Salesman Problem

The traveling salesman problem is a classical optimization problem. Optimization problems involve finding a maximum or minimum value of a mathematical function, usually subject to some sort of constraints expressed as mathematical function (Thompson & Thompson, 1987), (Shapiro, 1987). The traveling salesman problem is easy to describe: a salesman must visit a series of cities. Each city should be visited only once. After a final city is visited, the salesman returns to the starting city. The distance between each city is known. What is the shortest possible tour the salesman can make? Several experiments have been executed on traveling salesman problem using 5, 10, 15…., and 30 nodes. The most results of the proposed algorithm are near to optimal solutions especially using crossover operator, the other operators are not good like it. Table (2) illustrate sample of experimental results of TSP.

Table (2): Sample of Experimental Results of TSP

No. of Nodes	Optimal Value	Method	Avg. Value of TSP	Avg. Time (Sec.)	Best Iteration
10	875	GA	898	1.9	35
		CS	910	1.7	40
		CSGA (Crossover)	882	1.8	36
		CSGA (Conjugation)	929	2.4	48
		CSGA (Crowding)	917	2.5	33
		CSGA (Segregation)	915	2.6	34
		CSGA (Migration)	921	2.4	31
20	1358	GA	1389	3.9	47
		CS	1395	3.7	41
		CSGA (Crossover)	1379	3.6	38
		CSGA (Conjugation)	1402	4.1	44
		CSGA (Crowding)	1390	3.7	43
		CSGA (Segregation)	1394	3.9	42
		CSGA (Migration)	1398	3.8	46
30	3107	GA	3133	6.9	85

		CS	3139	5.9	77
		CSGA (Crossover)	3119	6.4	83
		CSGA (Conjugation)	3148	6.2	94
		CSGA (Crowding)	3142	6.2	81
		CSGA (Segregation)	3151	6.1	83
		CSGA (Migration)	3141	6.7	87

7. Conclusions

Cuckoo search - Genetic Algorithm (CSGA) is a hybrid intelligent algorithm that is based on two important algorithms, Cuckoo Search and Genetic algorithm. The proposed algorithm in this paper is a good tool to solve combinatorial problems. The presented algorithm uses the operators of genetic algorithm as a tool to improve the solutions in the cuckoo search algorithm. The experiments in this paper used several types of genetic algorithm operators (single-point, crowding crossover, segregation and migration) crossover operator and conjugation operator. In this paper two combinatorial problems have been solved using proposed algorithm with previous operators. In both problems (TSP and MST) the proposed CSGA within crossover operator is the best operator as results, while all the others operators are less than this operator.

References

Ahmed T. Sadeq, "Conjugation & Controlled Mutation: New Operators in Genetic Algorithms", Ph. D. dissertation, University of Technology, Baghdad, Iraq, 2000.

Alexander Schrijver, "**Combinatorial Optimization**", Technical Report, University of Amsterdam, Nethelands, 2013.

Bev & Bell Thompson, "Neurons, Analog Circuits & the Traveling Salesperson", AI Expert Journal,, July, 1987.

Chin S. Ch., Malcolm Y. H., Appa I. S. and Kheng L. G., "A Bee Colony Optimization Algorithm To Job Shop Scheduling", Proceeding of the Winter Simulation Conference, IEEE, 2006.

De Jong, K. A., "An Analysis of Behavior of a Class of Genetic Adaptive Systems", Ph. D. Thesis, University of Michigan, 1975.

Dorigo M. and Stutzle T., "Ant Colony Optimization", MIT Press, Cambridge, 2004.

Eberhart, R., Y. Shi, J. Kennedy, "Swarm Intelligence", Morgan Kaufmann, San Francisco, 2001.

Goldberg D. E., "Genetic Algorithms in Search, Optimization and Machine Learning ", Addison-Wesley, Longman, 1989.

H. Zheng and Y. Zhou, "A Novel Cuckoo Search Optimization Algorithm Base on Gauss Distribution", Journal of Computational Information Systems 8: 10, 4193–4200, 2012.

Holland, J. H., "Adaptation in Natural and Artificial Systems", University of Michigan Press, 1975.

Jane K. and Alxander H., "Genetic Engineering : Principles and Methods", Plenum Press, U.S.A., 1982.

Jawetz, Melnick and Adelberg, "Medical Microbiology", Prentice Hall Int., 1998.

Melanie Michell, "An Introduction to Genetic Algorithms", MIT Press, London, England, 1998.

Montgomery, E. J. (2005). Solution biases and pheromone representation selection in ant colony optimisation (Doctoral dissertation, School of Information Technology, Bond University).

Nimrod Megiddo, "**Discrete Optimization**", Elsevier Pub., 2011.

Pham D.T., Ghanbarzadeh A., Koc E., Otri S., Rahim S., and Zaidi M. "The Bees Algorithm - A Novel Tool for Complex Optimization Problems", Proceeding of the 2nd Virtual International conference On Intelligent Production Machines and Systems, D.T. Pham, E. E. Eldukhri and A. J. Soroka (eds), Elsevier (Oxford) (2006), ISBN-10:08-045157-8.

Pham D.T., Ghanbarzadeh A., Koc E., Otri S., Rahim S., and Zaidi M. "The Bees Algorithm", Technical Note, Manufacturing Engineering Center, Cardiff University, UK, 2005.

Shapiro, S., "Encyclopedia of Artificial Intelligence", U.S.A., 1987.

Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, and Clifford Stein. *Introduction to Algorithms*, Second Edition. MIT Press and McGraw-Hill, 2001. ISBN 0-262-03293-7. Chapter 23: Minimum Spanning Trees, pp. 561–579.

X. S. Yang and S. Deb, "Cuckoo search via Lévy flights". World Congress on Nature & Biologically Inspired Computing (NaBIC 2009). IEEE Publications. pp. 210–214, December, 2009.

Yang XS, Deb S, (2009) Cuckoo search via Lévy flights, Proceedings of World Congress on Nature & Biologically Inspired Computing, India, pp 210-214.