

Enhanced Cuckoo Search Algorithm with SPV Rule for Quadratic Assignment Problem

Mahmoud M. Ismail
Department of Decision
Support, Faculty of Computers
and Informatics, Zagazig
University, Egypt

Ibrahim M. Hezam
Department of computer,
Faculty of Education, Ibb
University
Ibb city, Yemen

Emad El-Sharkawy
Faculty of information systems
and computer science,
October 6 University , Egypt.

ABSTRACT

Cuckoo Search (CS) algorithm has exhibited good performance across a wide range of application problems. In this paper, a Modified Cuckoo Search (MCS) algorithm is presented to solve the Quadratic Assignment Problem (QAP), which is a NP-hard problem and is one of the most interesting and challenging combinatorial optimization problems in the research community. To handle the discrete variables of the Quadratic Assignment Problems, the smallest position value (SPV) rule is used to enable the continuous inter-species cuckoo search to be applied to most types of sequencing problems. In the computational experiments, we evaluate the performance of our approach on widely known instances from the literature. In these experiments, we compare the proposed algorithm against the best proposals from the related literature and we conclude that our algorithm is able to report high-quality solutions.

Keywords

Cuckoo Search Algorithm; Meta-heuristics; Optimization; SPV rule, Quadratic Assignment Problem.

1. INTRODUCTION

The quadratic assignment problem (QAP) was introduced by Koopmans and Beckman in 1957 in the context of locating "indivisible economic activities"[1]. The objective of the problem is to assign a set of facilities to a set of locations in such a way as to minimize the total assignment cost. The assignment cost for a pair of facilities is a function of the flow between the facilities and the distance between the locations of the facilities [2-4].

It is possible to formulate some classic problems of combinatorial optimization, such as the traveling salesman, maximum clique and graph partitioning problems as a QAP. The QAP belongs to the class of NP-complete problems and is considered one of the most difficult combinatorial optimization problems. Exact solution strategies for the QAP have been unsuccessful for large problem (approximately $N \geq 25$). The Mathematical formulation of the problem as follows [3-5]:

$$\min_{\phi \in S} \sum_{i=1}^n \sum_{j=1}^n a_{ij} b_{\phi(i)\phi(j)} + \sum_{i=1}^n c_{i\phi(j)} \quad (1)$$

There are many different interpretations for the matrices, one of which is:

- a_{ij} represents the flow from the facility i to the facility j ,
- b_{ij} represents the distance from the location i to the location j ,

- c_{ij} represents the cost of the placing facility i to the location j

According to the computational complexity theory, there exists no exact algorithms to solve NP-hard problems in polynomial time unless $P=NP$.

Optimization problems are the problem in finding the best solution from among the set of all feasible solutions [6]. Optimization problem in engineering and industry are typically non-linear. This problem is complex, often highly non-linear constraints imposed by physics and design code. As a result, the design problem is often NP-hard (Non-deterministic Polynomial-time hard) and need highly sophisticated tools [6-7]. Find the optimal solution to optimization problems is usually challenging. A recent trend is to use nature-inspired metaheuristics algorithms to find the solution for non-linear optimization problems. The nature-inspired algorithm is developed with the idea of nature such as fish, bees, flower and firefly. In general, there are two types of the nature-inspired algorithms, heuristic and metaheuristic [7]. Heuristic means to discover a good solution to an optimization problem by trial and error process. The quality of the solution can be found in a reasonable amount of time but it does not promise the optimal solution to be achieved [8] whereas metaheuristic is the advanced development of heuristic algorithm. Meta means beyond or higher-level and generally the performance is better than heuristics. In addition, metaheuristic uses some random tradeoff and local search. Two essential characteristics in metaheuristics are exploitation and exploration [7]. Exploitation intends to use the information from the current best solution. This process searches the best candidates around the neighborhood of current best solution whereas, exploration means to generate diverse solution as possible to explore the search space on a global scale.

Nowadays, various evolutionary metaheuristic optimization algorithms have been developed and designed to handle complex optimization problems. Among these algorithms, CS is proposed to look for a global optimal solution in combinatorial and complex optimization problems.

This paper is organized as follows: after introduction, section 2 briefly introduces cuckoo search algorithm. Section 3 the proposed algorithm is introduced. While the results are discussed in section 4. Finally, conclusions are presented in section 5.

2. CUCKOO SEARCH ALGORITHM

The Cuckoo search algorithm is a Meta heuristic search algorithm, which has been proposed recently by Yang and Deb [9,16], it was based on the following idealized rules:

(1) Each cuckoo lays one egg at a time, and dumps it in a randomly chosen nest.

(2) The best nests with high quality of eggs (solutions) will carry over to the next generations. (3) The number of available host nests is fixed, and a host can discover an alien egg with a probability $p_a \in [0,1]$.

In this case, the host bird can either throw the egg away or abandon the nest so as to build a completely new nest in a new location. The main steps of cuckoo search algorithm are summarized in algorithm 1. When generating new solutions $x_i(t+1)$ for the i th cuckoo, the following Lévy flight is performed:

$$x_i^{(t+1)} = x_i^{(t)} + \alpha \oplus \text{Levy}(\lambda) \quad (2)$$

Where $\alpha > 0$ is the step size, which should be related to the scale of the problem of interest. The product \oplus means entry-wise multiplications. We consider a Lévy flight in which the step-lengths are distributed according to the following probability distribution

$$\text{Levy}(u) = t^{-\lambda}, 1 < \lambda \leq 3 \quad (3)$$

This has an infinite variance. 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 [9,14-17].

Algorithm 1: Cuckoo search algorithm

Define Objective function $f(x)$, $x = (x_1, x_2, \dots, x_d)$

Initial a population of n host nests x_i ($i = 1, 2, \dots, d$)

while ($t < \text{MaxGeneration}$) or (stop criterion);

Get a cuckoo (say i) randomly and generate a new solution by Lévy flights;

Evaluate its quality/fitness; F_i

Choose a nest among n (say j) randomly;

if ($F_i > F_j$),

Replace j by the new solution;

end

Abandon a fraction (P_a) of worse nests

[and build new ones at new locations via Lévy flights];

Keep the best solutions (or nests with quality solutions);

Rank the solutions and find the current best;

end while

Post process results and visualization;

End

3. THE PROPOSED ALGORITHM (MCS) FOR QUADRATIC ASSIGNMENT PROBLEM

Cuckoo search succeeded in proving its superior performance, compared with PSO and GA (to solve multimodal functions) with its better strategy in exploring the solution space. This strategy is enhanced by Lévy flights, which has an important role in controlling the balance between intensification and diversification. And the reduced number of parameters enables CS to be more versatile [9]. The strength of CS is the way how to exploit and explore the solution space by a cuckoo. This cuckoo can have some ‘intelligence’ so as to find much better solutions. So we can control the

intensification and diversification through the cuckoo’s mobility in the search space. The proposed improvement considers the smallest position value (SPV) rule is used to enable the continuous inter-species cuckoo search. The main idea of our proposed algorithm is that the modified CS seeks good solutions found using local search in areas specified by Lévy flights. The weakness of local search is that it is likely to be trapped in a local optimum. This can easily be strengthened by using our modified CS that requires the displacements by zones and not by solutions, which should minimize the probability of falling into local optima. One of the objectives in extending CS to solve the Quadratic Assignment Problem (QAP) is to keep its main advantages and integrate these advantages into the discrete version of modified CS.

In this section, we will introduce the proposed algorithm for solving Quadratic Assignment Problem. The main steps of the improved cuckoo search algorithm are summarized in algorithm 2.

Algorithm 2: Cuckoo Search Algorithm for Quadratic Assignment Problem

Step 1: Set n size of nest and initial a population of n host nests $\{X_i^0, i = 1, 2, \dots, n\}$ m the number of permutations

Step2: Generate initial solution randomly

$$\{X_i^0, i = 1, 2, \dots, n\} \text{ where } X_i^0 = [x_{i1}^0, x_{i2}^0, \dots, x_{im}^0]$$

Step3: Apply the SPV rule to find the sequence

$$\phi_i^0 = [\phi_{i1}^0, \phi_{i2}^0, \dots, \phi_{im}^0] \text{ of cuckoo } \{X_i^0, i = 1, 2, \dots, n\}$$

Evaluate fitness for every individual and determine the best individual with the best objective value;

Step 4: while ($t < \text{MaxGeneration}$) or (stop criterion); Do;

Step 5: Get a cuckoo X_s randomly and generate a new solution,

Step 6: Apply the SPV rule to find the sequence

$$\phi_i^s = [\phi_{i1}^s, \phi_{i2}^s, \dots, \phi_{im}^s] \text{ for } i=1, \dots, n \text{ of cuckoo}$$

$\{X_i^s, i = 1, 2, \dots, n\}$ Evaluate fitness for every individual and determine the best individual with the best objective value F_s ;

Step 7: choose another individual X_j randomly;

Step 8: Apply the SPV rule to find the sequence

$$\phi_i^j = [\phi_{i1}^j, \phi_{i2}^j, \dots, \phi_{im}^j] \text{ of cuckoo } \{X_i^j, i = 1, 2, \dots, n\}$$

Evaluate fitness for every individual and determine the best individual with the best objective value F_j ;

Step 9: if $F_s > F_j$

Step 10: Replace j by the new solution;

Step 11: Abandon a fraction (P_a) of worse nests [and build new ones at new locations];

Step 12: Apply the SPV rule to find the sequence

$$\phi_i^k = [\phi_{i1}^k, \phi_{i2}^k, \dots, \phi_{im}^k] \text{ for } i=1, \dots, n. \text{ Evaluate its}$$

quality/fitness; $F_i^k, i = 1, 2, \dots, n$

Step 13: Keep the best Sequence solutions (or nests with quality solutions);

Step 14: Rank the solutions and find the current best;

Step 15: Update the generation number $t=t+1$;

Step 16: Post process results and visualization;

4. EXPERIMENTS AND RESULTS

Numerous examples have been done to verify the weight of the proposed algorithm. The standard particle swarm optimization (PSO), simulated annealing (SA) [10] and the improved cuckoo search (MCS) algorithms are tested on some instances (benchmarks) of QAP taken from the publicly available electronic library QAPLIB of QAP problems [11]. Most of the instances included in QAPLIB have already been solved in the literature and their optimality results can be used to compare algorithms. twenty one instances are considered with sizes ranging from 10 to 256. The numerical value in the name of an instance represents the size of the instance. All the experiments were performed on a Windows 7 Ultimate 64-bit operating system; processor Intel Core to Duo CPU 2.20 GHZ; 4 GB of RAM and code was implemented in MATLAB. The values of parameters of the proposed algorithm are selected, based on some preliminary trials. The selected parameters in MCS are those values that gave the best results concerning both the solution quality and the computational time. In each case study, 30 independent runs of the algorithms with these parameters are carried out. Tables 1 summarize the experiments results.

Table 2 presents the computational results of the proposed algorithm on seven QAPLIB instances. The column 'SD' denotes the standard deviation which takes the value 0.00 shown in bold when all solutions found have the same length over the 30 runs.

We compare our proposed algorithm with one recently reported by [12]. Table 3 summarizes the comparison of error rate of the proposed algorithm with bat algorithm and particle swarm optimization.

In Tables 4, the experimental results of the Proposed algorithm are compared with the both methods ACO and PSO. The results of these two methods are directly

summarized from original paper [13]. It can be seen clearly from Tables 4 that MCS outperforms the other two algorithms (ACO and PSO) in solving all the six tested QAP instances. The proposed MCS algorithm obtains four best solutions while ACO and PSO only obtains one best solution among six QAP instances.

Table 3. Comparison of Error rate of the proposed algorithm with [12]

Instance	BAT[12]	PSO[12]	MCS
Bur26a	1.90	3.26	0.00
Esc32e	0.00	40	0.00
Lipa20a	3.8	4.2	0.00
Lipa30a	3.3	3.6	0.12

5. CONCLUSION

The quadratic assignment problem is one of the most studied combinatorial optimization problems with various practical applications. In this paper, the continuous cuckoo search Algorithm was converted into discrete cuckoo search algorithm based on SPV to solve the Quadratic Assignment Problems. One of the advantages of the modified MCS is that it is relatively independent of cuckoos in the search process for the best position. So, it is more likely to find good solutions in areas unexplored by metaheuristics, which take the best position found so far as a starting point for other much better solutions. The performance of the proposed algorithm is tested against a set of benchmarks of QAP from the public QAPLIB Library. The results of the tests show that the proposed algorithm is superior to some other metaheuristics.

Table 1. The comparison results of the proposed algorithm and other algorithms

Instance	Optimal Sol.	PSO[10]		SA[10]		OB-GA [10]		MCS	
		optimal	Error%	optimal	Error%	optimal	Error%	optimal	Error%
0	135028	135028	0.000	135028	0.000	135028	0.000	135028	0.000
tai20a	703482	704190	0.101	703482	0.000	703482	0.000	703482	0.000
tai30a	1818146	1868871	2.790	1849696	1.735	1863722	2.506	1818146	0.000
tai40a	3139370	3233491	2.998	3212692	2.335	3215382	2.421	3139370	0.000
tai50a	4941419	5088574	2.978	5041058	2.016	5067904	2.559	4941419	0.000
tai60a	7208572	7438398	3.188	7381442	2.398	7422318	2.965	7208572	0.000
tai80a	13557864	13831255	2.016	13852100	2.170	13814562	1.893	13557864	0.000
tai100a	21125314	21603991	2.266	21499478	1.771	21501554	1.780	21125314	0.000
tai150b	498896643	508956543	2.016	502651565	0.572	506448890	1.513	498896643	0.000
tai256c	44759294	44819486	0.134	44814014	0.122	45023121	0.589	44759294	0.000
sko100a	152002	155462	2.276	154210	1.452	153090	0.715	152002	0.000

sko100b	153890	156007	1.376	154262	0.241	155030	0.740	153890	0.000
sko100c	147862	150978	2.107	149542	1.136	149948	1.410	147862	0.000
sko100d	149576	152346	1.852	151746	1.450	150828	0.837	149576	0.000

Table 2. Computational results of the proposed algorithm for another 7-benchmark instances for QAPLIB

Instance	Optimal	Best	Worst	Average	SD	Time/s
Chr12a	9552	9552	9552	9552	0.00	1.24
Chr15b	7990	7990	7990	7990	0.00	3.71
Chr20a	2192	2192	2192	2192	0.00	4.58
Bur26a	5426670	5426670	5426670	5426670	0.00	6.54
Kra30a	88900	88900	88900	88900	0.00	8.21
Esc32a	130	130	130	130	0.00	9.81
Lipa40a	31538	31538	31539	31538.03	0.17	41.51

Table 4. Comparison of experimental results of the proposed algorithm with [13]

Instance	ACO[13]			PSO[13]			MCS		
	AvgCost	Std.	Time	AvgCost	Std.	Time	AvgCost	Std.	Time
nug5	50.0	0	258.47	50.2	0.6325	103.81	50	0.00	0.1
chr12a	16557	1661.6	1048.0	13715	2098.0	736.05	9552	0.00	1.24
tail2a	256180	3066.5	968.16	254230	5809.9	653.14	224416	0.00	2.01
dre30	1849.6	82.1998	1514.4	1592.0	118.47	1040.5	508	0.00	9.3
tho40	302840	3603.3	1612.3	286670	5318.3	1233.3	240516.37	2.87	45.1
tai50a	5626356	15225	2045.3	5587622	52893	1602.4	4941503	21.24	98.7

6. REFERENCES

- [1] Koopmans, T. C., and Beckmann, M. 1957. Assignment problems and the location of economic activities. *Econometrica: journal of the Econometric Society*, 53-76.
- [2] Finke, G., Burkard, R. E., and Rendl, F. 1987. Quadratic assignment problems. *North-Holland Mathematics Studies*, 132, 61-82.
- [3] Burkard, R. E. 1984. Quadratic assignment problems. *European Journal of Operational Research*, 15(3), 283-289.
- [4] Burkard, R. E. 2013. Quadratic assignment problems, pp. 2741-2814. Springer New York.
- [5] Shawky, L. A. E. F., Metwally, M. A. E. B., and Zaid, A. E. N. H. 2015. Quadratic Assignment Problem: A survey and Applications. *International Journal of Digital Content Technology and its Applications*, 9(2), 90.
- [6] Rao, S. S., & Rao, S. S. 2009. *Engineering optimization: theory and practice*. John Wiley & Sons.
- [7] Glover F. W., & Kochenberger, G. A. (Eds.). 2006. *Handbook of metaheuristics (Vol. 57)*. Springer Science & Business Media.
- [8] Blum, C., & Roli, A. 2003. Metaheuristics in combinatorial optimization: Overview and conceptual comparison. *ACM Computing Surveys (CSUR)*, 35(3), 268-308.
- [9] Gandomi, A. H., Yang, X. S., & Alavi, A. H. 2013. Cuckoo search algorithm: a metaheuristic approach to solve structural optimization problems. *Engineering with computers*, 29(1), 17-35.
- [10] Mamaghani, A. S., & Meybodi, M. R. 2012, October. Solving the Quadratic Assignment Problem with the modified hybrid PSO algorithm. In *Application of Information and Communication Technologies (AICT), 2012 6th International Conference on* pp. 1-6. IEEE.
- [11] Burkard, R. E., Karisch, S. E., & Rendl, F. 1997. QAPLIB—a quadratic assignment problem library. *Journal of Global optimization*, 10(4), 391-403.
- [12] Shukla, A. 2015, May. A modified bat algorithm for the Quadratic Assignment Problem. In *2015 IEEE Congress on Evolutionary Computation (CEC)* .pp. 486-490. IEEE.
- [13] Liu, H., Abraham, A., & Zhang, J. 2007. A particle swarm approach to quadratic assignment problems. In *Soft Computing in Industrial Applications* (pp. 213-222). Springer Berlin Heidelberg.
- [14] Hezam I. M., M. Abd-ElBaset and I. Selem, 2015. Cuckoo Search Algorithm for Stellar Population Analysis of Galaxies, *International Journal of Information Technology and Computer Science*, vol. 7, pp.29-33.
- [15] Abdel-Baset, M., & Hezam, I. 2016. Cuckoo Search and Genetic Algorithm Hybrid Schemes for Optimization Problems. *Appl. Math*, 10(3), 1185-1192.
- [16] Abdel-Basset, M., Hessin, A. N., & Abdel-Fatah, L. 2016. A comprehensive study of cuckoo-inspired algorithms. *Neural Computing and Applications*, 1-17.
- [17] Abdel-Baset, M., & Hezam, I. M. (2016). Solving Linear Least Squares Problems Based on Improved Cuckoo Search Algorithm. *Math. Sci. Lett.* 5, No. 2, 199-202