International Journal of
Soft Computing and Software Engineering (JSCSE)

Vol.4, No.6, 2014
Published online: Jun 25, 2014

e-ISSN: 2251-7545
DOI: 10.7321/jscse.v4.n6.1

# Discrete Cat Swarm Optimization for Solving the Quadratic Assignment Problem

*1Mohammed ESSAID RIFFI, 2Abdelhamid BOUZIDI
*Laboratory of MATIC, Computer Science Department.*
*Faculty of Sciences, Chouaïb Doukkali University*
*EL JADIDA, MOROCCO*
*Email : 1said@riffi.fr, 2mr.abdelhamid.bouzidi@gmail.com*

***Abstract.*** **A discrete cat swarm optimization is a metaheuristic based on natural behavior of cats, each cat has two modes that are the seeking mode, and the tracing mode. The seeking mode is when a cat is at rest, that's how a cat spends most of its life time. The tracing mode is when a cat is hunting. This paper proposes a new discrete cat swarm optimization algorithm to solve the quadratic assignment problem, as one of the known combinatorial optimization problems. This problem is attributed to NP-Hard class. In order to test the performance of the algorithm described herein, we will resolve some instances of the quadratic assignment library problem.**

**Keywords**: *Cat Swarm Optimization, CSO, Quadratic assignment problem, QAPLIB, tracing mode, seeking mode.*

* Corresponding Author:
Mohammed ESSAID RIFFI,
Lab. Of MATIC, Dept. of Computer Science, Faculty of Sciences,
Chouaib Doukkali University, EL JADIDA, MOROCCO,
Email: said@riffi.fr Tel: +212-670-471153

## 1. Introduction

The quadratic assignment problem was introduced in 1957 by C. Koopmans and BECKMAN [1], to solve assignment problems and the location of economic activities [2]. Noticing the large number of area application where this problem appears, such as localization applications [3], Scheduling Problem [4][5] ,hospital layout [6],computer panel design [7] [8], it ignited interest for many scientific researchers.

For solving this NP-hard combinatorial optimization problem, a set of exacts algorithms [9-10], heuristics like Simulated Annealing [11], Tabu Search [12-13], Harmony Search [14], and Metaheuristic based on the natural behavior of swarms are designed to overcome the difficulties presented by conventional methods, like Genetic Algorithms [15-16], Ant Colony Optimization [17], and Particle Swarm Optimization [18].

In this research paper, we will demonstrate how we adapted the cats swarm optimization algorithm for solving the quadratic assignment problem beginning with a presentation of the problem studied. Then we will describe the metaheuristic used, its origin, parameters, and methods. Next, we will propose an adaptation of the cats swarm optimization algorithm in order to solve the quadratic assignment problem and we will present the results for the implementation of this metaheuristic in the standard instances of the library QAPLIB [19,26]. Finally, we will give a conclusion.

## 2. Formulation of the problem:

The quadratic assignment problem consists of place **n** objects $O = \{o_1,o_2,\dots,o_n\}$ are intercommunicating in **n** positions $P = \{p_1,p_2,\dots,p_n\}$. Mathematically it is to realize a bijection $\mathbf{o_i}$ elements of the set **O**, on those $\mathbf{p_j}$ of the set **P**, with: $(i, j) \in [1, n]$ and **Card(O) = Card(P) =n**. A flow

International Journal of
Soft Computing and Software Engineering (JSCSE)

Vol.4, No.6, 2014
Published online: Jun 25, 2014

e-ISSN: 2251-7545
DOI: 10.7321/jscse.v4.n6.1

matrix C, where each element $\mathbf{c_{ij}}$ indicates the flow between an object $\mathbf{o_i}$ and an object $\mathbf{o_j}$. And a distance matrix D where each element $\mathbf{d_{kl}}$ gives the distance between a position $\mathbf{p_k}$ and a position $\mathbf{p_l}$ with $(\mathbf{k},\mathbf{l}) \in [\mathbf{1},\mathbf{n}]$.

The objective is to minimize the function $F(\varphi)$ :

$$\boxed{F(\varphi) = \sum_{i=1}^{n}\sum_{j=1}^{n}(c_{ij} * d_{\varphi(i)\varphi(j)})}$$

The solution can be found through many ways. For our case, we will use the instances of QAPLIB library, where the solution is presented by a permutation vector $\varphi$, of n elements:

$$\varphi = \{\varphi_1, \varphi_2, \dots, \varphi_n\}$$

The optimal solution will be the optimal permutation vector $\varphi$ where $\mathbf{F(\varphi)}$ has the optimum value. The optimum function values solution are given in QAPLIB web site [26].

## 3. Cat Swarm Optimization:

The Cat Swarm optimization (CSO) algorithm was realized in 2006 CHU & TSAI[20], and used by other researchers to solve Continuous Optimization for a single objective problem[20-21], and also multi-objective[22-24]. This algorithm was used in 2013 by A.BOUZIDI and M.E.RIFFI [25], to solve a classical combinatorial optimization problem, which is the traveling salesman problem.

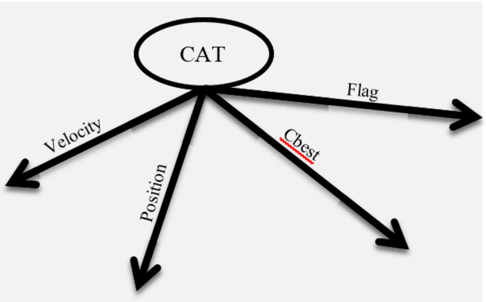


**Figure 1 : Characteristics of Cats**

Every cat is characterized by its position which represents the solution, velocity, best position, and its flag, that shows the mode of cat, if it's in tracing or searching mode. The mode is changed by the mixer ratio.

## 4. CSO to solve the Quadratic Assignment Problem:

To solve the Quadratic Assignment Problem (QAP), we need to find the minimum permutation vector into distance matrix **D**. This vector will be presented as the position of the cat, and the description of every cat in swarm is as following:

**Position** : permutation vector $\varphi$
**Cbest** : Best permutation vector found by this cat
**Velocity** : Present the set of permutation to be applied to the position of the cat
**Flag** : used to determinate if the cat is in SM or TM.

After setting the operators, we'll go to the handling part, and the use of characteristics.

As already indicated, each cat has two modes, the searching **(SM)** and the tracing mode **(TM)** that are combined by the mixer ratio **(MR)**.

### 4.1. Searching Mode (SM):

It shows that the cat $_i$ is at rest, the parameters used in this mode are:

**SMP**: Seeking memory pool.
**CDC**: seeking range of the selected dimension.
**SRD**: counts of dimension to change
**SPC**: self-position consideration.

The process of seeking mode is described as follows:

e-ISSN: 2251-7545
DOI: 10.7321/jscse.v4.n6.1

**Begin**

Make j copies of cat $_k$
**if** (SPC==true)
includes cat $_k$ as a candidate and j=SMP-1
**ELSE**
j= SMP
**EndIf**
**For** each copy
Select a number of dimensions based on CDC
Update their value using SRD percent of their current value
Evaluate the fitness of each copy
**For** each cat
$$P_k = \frac{|FS_k - FS_{min}|}{FS_{max} - FS_{min}}$$
Randomly select a new position for cat $_k$
**EndFor**
**END**

### 4.2. Tracing mode TM:

This is the hunting mode, when every cat$_i$ moves quickly to trace his path by updating its velocity, and after that moves according to its velocity position. The description of the process is as follows:

**Begin**
//update_velocity
$\mathbf{V'_i = w*V_i + r_1 * c_1 * (X_{best} - X_i)}$ //(1)
//update_position
$\mathbf{X_i = X_i + V_i}$ //(2)
**END**

Where, in equation (1):

- **X** best: is the best solution / position of the cat who has the best fitness value.
- **V $_i$**: The old speed value (current value).
- **V' $_i$**: the new value of the velocity obtained by the equation.
- **$c_1$**: is a constant.
- **$r_1$**: a random value in the range [0, 1]

And in equation **(2)**:

- **X' $_k$**: The new value of the position of the cat $_i$
- **X $_k$**: The actual position of cat $_i$
- **V $_k$**: The velocity of cat $_i$

Some concepts about operation will be introduced.

**Definition 1**: a movement of cat $_k$ is a swap applied to solution/position of this cat.

**Definition 2:** Addition between position **X** and a velocity **V** (X+V), is applying the swap in **V** to position **X**, the result is a new position.

**Definition 3**: Addition between two velocities **v** and **v'** ( **v**+ **v'**), is a new velocity containing all the couple of swaps of **v** and **v'**.

**Definition 4**: the result of the subtraction between two positions **x** and **x'** is a velocity **v**, it's the opposite of addition:

$$\mathbf{x + v = x' \Leftrightarrow x' - x = v}$$

International Journal of
Soft Computing and Software Engineering (JSCSE)

Vol.4, No.6, 2014
Published online: Jun 25, 2014

e-ISSN: 2251-7545
DOI: 10.7321/jscse.v4.n6.1

**Definition 5**: a multiplication is performed between a float value and velocity, the result is a velocity. The different possible cases according to the real **k** are:

- If **k = 0 :** **k * v = 0**
- If **(k>0 & k<=1) :** Then **r * v = ($i_k$,$j_k$)[k : 0 → (c*|v|)]**
- If **k>1**: then we separate. Decimal and integer part, **k = n +x**. Where **n** is the integer part of **r**, and **x** corresponds to the decimal parts. We will then return each party to the previous cases.
- If **k<0: k * v= (-k)* ¬v.** Now **(-k) >0**, and you will consider one of the previous cases.

### 4.3. CSO Algorithm:

To combine the two modes of CSO algorithm, the mixture ratio (MR) indicates the rate of mixing the seeking mode and the tracing mode. The general process of Cat Swarm Optimization Algorithm is:

```
BEGIN
  Create N cats
  Analyze the position, velocity, and flag of every cat
  Repeat
     Evaluate the position of each cat and keep the position
  of the cat with the best fitness value.
     If flag of catk is SM
          Apply catk into SM Process
     ELSE
          Apply catk into TM Process
     End If
     Reinitialize the flag of each cat
  Until ( Condition is satisfied )
END
```

## 5. Results and Discussion:

In this part we will check the validity of the proposed adaptation for the CSO algorithm to resolve QAP problem. For this benchmark **instance** of QAPLIB library is selected for the simulations. The experiments were performed on a PC with processor Intel(R) Core(TM) 2 Duo CPU T5800@ 2.00GHZ and 3.00 MB of RAM.

The table 2 presents the result of applying CSO to some instances of QAPLIB the best Optimum found in the library [26], For Each **instance** runs by a generation of a random number of cats. After reaching to the result **opt obtained** taken in ten iterations, the **time** is the average execution time got by the Sum time of every execution divided by ten, and calculate the average **Err**, the error is got by:

$$\%err = \frac{\mathbf{optimum_{obtained}} - \boldsymbol{Optimum}}{\mathbf{Optimum}} \times \mathbf{100}$$

This table 1 show the values of parameters used:

**Table 1.** Parameter values that has been used

| | |
|---|---|
| SMP | 5 |
| CDC | 0.8 |
| MR | 0.3 |
| C1 | 2.05 |
| R1 | [0,1] |
| W | 0.729 |

**Table 2.** Table of Results

| **Instance** | **opt** | **Opt Obtained** | **Time (s)** | **Err** |
|---|---|---|---|---|

e-ISSN: 2251-7545
DOI: 10.7321/jscse.v4.n6.1

| | | | | |
|---|---|---|---|---|
| **Bur26a** | 5426670 | 5426670 | 15 | 0 |
| **Bur26b** | 3817852 | 3817852 | 20 | 0 |
| **Bur26c** | 5426795 | 5426795 | 24 | 0 |
| **Bur26d** | 3821225 | 3821225 | 69 | 0 |
| **Bur26e** | 5386879 | 5386879 | 05 | 0 |
| **Bur26f** | 3782044 | 3782044 | 21 | 0 |
| **Bur26g** | 10117172 | 10117172 | 36 | 0 |
| **Bur26h** | 7098658 | 7098658 | 22 | 0 |
| **Chr12a** | 9552 | 9552 | 00 | 0 |
| **Chr12b** | 9742 | 9742 | 01 | 0 |
| **Chr12c** | 11156 | 11156 | 03 | 0 |
| **Chr15a** | 9896 | 9896 | 19 | 0 |
| **Chr15b** | 7990 | 7990 | 01 | 0 |
| **Chr15c** | 9504 | 9504 | 05 | 0 |
| **Chr18a** | 11098 | 11098 | 41 | 0 |
| **Chr18b** | 1534 | 1534 | 01 | 0 |
| **Chr20a** | 2192 | 2196 | 46 | 0 |
| **Chr20b** | 2298 | 2298 | 417 | 0 |
| **Chr20c** | 14142 | 14142 | 05 | 0 |
| **Chr22a** | 6156 | 6156 | 14 | 0 |
| **Chr22b** | 6194 | 6194 | 56 | 0 |
| **Chr25a** | 3796 | 3796 | 140 | 0 |
| **Els19** | 17212548 | 17212548 | 7 | 0 |
| **Esc16a** | 68 | 68 | 00 | 0 |
| **Esc16b** | 292 | 292 | 00 | 0 |
| **Esc16c** | 160 | 160 | 00 | 0 |
| **Esc16d** | 16 | 16 | 00 | 0 |
| **Esc16e** | 28 | 28 | 00 | 0 |
| **Esc16f** | 0 | 0 | 00 | 0 |
| **Esc16g** | 26 | 26 | 00 | 0 |
| **Esc16h** | 996 | 996 | 00 | 0 |
| **Esc16i** | 14 | 14 | 00 | 0 |
| **Esc16j** | 8 | 8 | 00 | 0 |
| **Esc32a** | 130 | 130 | 00 | 0 |
| **Esc32b** | 168 | 168 | 00 | 0 |
| **Esc32c** | 642 | 642 | 00 | 0 |
| **Esc32d** | 200 | 200 | 01 | 0 |
| **Esc32e** | 2 | 2 | 00 | 0 |
| **Esc32g** | 6 | 6 | 00 | 0 |
| **Esc32h** | 438 | 438 | 03 | 0 |
| **Esc64a** | 116 | 116 | 30 | 0 |
| **Esc128** | 64 | 64 | 979 | 0 |
| **Had12** | 1652 | 1652 | 00 | 0 |
| **Had14** | 2724 | 2724 | 00 | 0 |
| **Had16** | 3720 | 3720 | 02 | 0 |
| **Had18** | 5358 | 5358 | 02 | 0 |
| **Had20** | 6922 | 6922 | 01 | 0 |
| **Kra30a** | 88900 | 88900 | 27 | 0 |
| **Kra30b** | 91420 | 91420 | 104 | 0 |
| **Kra32** | 88700 | 88700 | 90 | 0 |
| **Lipa20a** | 3683 | 3683 | 04 | 0 |
| **Lipa20b** | 27076 | 27076 | 00 | 0 |
| **Lipa30a** | 13178 | 13178 | 49 | 0 |
| **Lipa30b** | 151426 | 151426 | 00 | 0 |
| **Lipa40a** | 31538 | 31538 | 733 | 0 |

e-ISSN: 2251-7545
DOI: 10.7321/jscse.v4n6.1

| | | | | |
|---|---|---|---|---|
| **Lipa40b** | 476581 | 47 6581 | 00 | 0 |
| **Lipa50a** | 62093 | 62093 | 2233 | 0 |
| **Lipa50b** | 1210244 | 1210244 | 00 | 0 |
| **Lipa60b** | 2520135 | 2520135 | 00 | 0 |
| **Lipa70b** | 46032200 | 46032200 | 00 | 0 |
| **Lipa80b** | 7763962 | 7763962 | 00 | 0 |
| **Lipa90b** | 12490441 | 12490441 | 00 | 0 |
| **Nug12** | 578 | 578 | 00 | 0 |
| **Nug14** | 1014 | 1014 | 01 | 0 |
| **Nug15** | 1150 | 1150 | 00 | 0 |
| **Nug16a** | 1610 | 1610 | 00 | 0 |
| **Nug16b** | 1240 | 1240 | 01 | 0 |
| **Nug17** | 1732 | 1734 | 16 | 0 |
| **Nug18** | 1930 | 1930 | 05 | 0 |
| **Nug20** | 2570 | 2570 | 09 | 0 |
| **Nug21** | 2438 | 2438 | 05 | 0 |
| **Nug22** | 3596 | 3596 | 06 | 0 |
| **Nug24** | 3488 | 3488 | 08 | 0 |
| **Nug25** | 3744 | 3744 | 07 | 0 |
| **Nug27** | 5234 | 5234 | 22 | 0 |
| **Nug28** | 5166 | 5166 | 132 | 0 |
| **Nug30** | 6124 | 6124 | 961 | 0 |
| **Rou12** | 235528 | 235528 | 01 | 0 |
| **Rou15** | 354210 | 354210 | 01 | 0 |
| **Rou20** | 725522 | 725522 | 988 | |
| **Scr12** | 31410 | 31410 | 00 | 0 |
| **Scr15** | 51140 | 51140 | 02 | 0 |
| **Scr20** | 110030 | 110030 | 07 | 0 |
| **Sko42** | 15812 | 15812 | 834 | 0 |
| **Sko49** | 23386 | 23418 | 5271 | 0.14 |
| **Tai12a** | 224416 | 224416 | 01 | 0 |
| **Tai12b** | 39464925 | 39464925 | 01 | 0 |
| **Tai15a** | 388214 | 388214 | 33 | 0 |
| **Tai15b** | 51765268 | 51765268 | 01 | 0 |
| **Tai17a** | 491812 | 491812 | 03 | 0 |
| **Tai20a** | 703482 | 708584 | 96 | 0 |
| **Tai20b** | 122455319 | 122455319 | 19 | 0 |
| **Tai25a** | 1167256 | 1167256 | 1465 | 0 |
| **Tai25b** | 344355646 | 344355646 | 473 | 0 |
| **Tai30b** | 637117113 | 637117113 | 352 | 0 |
| **Tai35b** | 283315445 | 283315445 | 915 | 0 |
| **Tai40b** | 637250948 | 637250948 | 3052 | 0 |
| **Tai64c** | 1855928 | 1855928 | 97 | 0 |
| **Tho30** | 149936 | 149936 | 640 | 0 |
| **Tho40** | 240516 | 240516 | 779 | 0 |

## 6. Conclusion:

Recently, a set of difficult combinatorial problems has led many researchers to solve it by several methods, especially methods based on the population approach. In this research paper, we used the algorithm Cat Swarm Optimization, a recent algorithm, which has shown its performance in the table2 that contains the results obtained by its application to this discrete problem without hybridization. We really hope that it will be applied to other combinatorial problems, not just single-objective ones but also for multi-criterion discreetly problems. Finally, we will be very interested to answer your

e-ISSN: 2251-7545
DOI: 10.7321/jscse.v4.n6.1

questions, not only to improve the meta-heuristics, but also to solve difficult optimization problems that surround us.

## REFERENCES:

[1] Koopmans, T.C., and M. Beckman . Assignment Problems and the Location of Economic Activities.Econometric 25, 53-76, 1957.
[2] Cela E. . "The Quadratic Assignment Problem: Theory and Algorithms Kluwer", 1998.
[3] Koopmans, T.C., and M. Beckman . "Assignment problems and the location of economic activities". Econometric, vol. 25, no.1, pp.53-76, 1957.
[4] Carlson, R.C. and G.L. Nemhauser . "Scheduling to minimize cost". Operations Research, vol. 14, 52-58, 1966.
[5] Geoffrion, A.M., and G.W. Graves. "Scheduling Parallel Production Lines with Changeover Costs: Practical Applications of a Quadratic Assignment/LP Approach". Operations Research , vol. 24, pp. 595-610, 1976.
[6] Elshafei, A.N. "Hospital Layout as a Quadratic Assignment Problem". Operations Research Quarterly, vol. 28, pp.167-179, 1977.
[7] Brixius, N.W. and K.M. Anstreicher . "The Steingberg Wiring Problem". To appear in, The Sharpest Cut, M. Grtschel, ed., SIAM, 2001.
[8] Resende, M.G.C., K.G. Ramakrishnan, and Z. Drezner. "Computing Lower Bounds for the Quadratic Assignment Problem with an Interior Point Algorithm for Linear Programming". Operations Research, vol. 43, pp.781-791, 1995.
[9] Christo"des N, Benavent E. "An exact algorithm for the quadratic assignment problem". Operations Research, vol. 37, pp:760-8.
[10] Bazara MS, Sherali MD. "Benders' partitioning scheme applied to a new formulation of the quadratic assignment problem". Naval Res. Logist. Quart, vol.27; pp.29-41, 1980.
[11] Wilhelm MR, Ward TL. "Solving quadratic assignment problems by simulated annealing". IEEE Trans, vol.19, pp.107-19, 1987.
[12] Skorin-Kapov J. "Tabu search applied to the quadratic assignment problem". ORSA J, Comput. Vol.2, pp.33-45, 1990.
[13] Taillard E. "Robust tabu search for the quadratic assignment problem". Parallel Comput, vol.17,pp.443-55,1991.
[14] T. Amudha, "Improving the Solution Quality of Quadratic Assignment Problems using Harmony Improvised," vol. 3, no. 8, pp. 1–8, 2012..
[15] Fleurent C, Ferland JA. "Genetic hybrids for the quadratic assignment problem". DIMACS Series in Discrete Mathematics and Theoretical Computer Science, vol. 16, Providence. RI: American Mathematical Society, pp. 173-87, 1994.
[16] Y. Wu and P. Ji, "Solving the Quadratic Assignment Problems by a Genetic Algorithm with a New Replacement Strategy," pp. 225–229, 2008.
[17] Vittorio Maniezzo and Alberto Colorni. "The ant system applied to the quadratic assignment problem". Knowledge and Data Engineering, IEEE Transactions on , vol.11; no.5, september/october 1999.
[18] Mamaghani, Ali Safari, and Mohammad Reza Meybodi. "Solving the Quadratic Assignment Problem with the modified hybrid PSO algorithm." Application of Information and Communication Technologies (AICT), 2012 6th International Conference on. IEEE, pp.1-6, 2012..
[19] Burkard, Rainer E., Stefan E. Karisch, and Franz Rendl. "QAPLIB–a quadratic assignment problem library." Journal of Global optimization, vol.10, no.4, pp.391-403, 1997.
[20] S. C. Chu, and P. W. Tsai, "Computational Intelligence Based on the Behaviour of Cat", International Journal of Innovative Computing, Information and Control, vol.3, no.1, pp.163-173, 2007.
[21] B. Santosa and M. K. Ningrum, "Cat Swarm Optimization for Clustering". International Conference of Soft Computing and Pattern Recognition, pp. 54–59, 2009.
[22] M. Carle and A. Martel, "The CAT Metaheuristic for the Solution of Multi-Period Activity-Based Supply Chain Network Design Problems," 2010.

Vol.4, No.6, 2014
Published online: Jun 25, 2014

e-ISSN: 2251-7545
DOI: 10.7321/jscse.v4.n6.1

[23] Hwang, Jong-Ching, et al. "CSO algorithm for economic dispatch decision of hybrid generation system." Proceedings of the 10th WSEAS international conference on applied informatics and communications, and 3rd WSEAS international conference on biomedical electronics and biomedical informatics. World Scientific and Engineering Academy and Society (WSEAS), pp. 81–86, 2010.

[24] Hwang, Jong-Ching, et al. "CSO and PSO to solve optimal contract capacity for high tension customers." Power Electronics and Drive Systems, 2009. PEDS 2009. International Conference on. IEEE, pp. 246–251, 2009.

[25] A. Bouzidi and M. E. Riffi, “Discrete Cat Swarm Optimization to Resolve the Traveling Salesman Problem”, IJARCSSE International Journal of Advanced Research in Computer Science and Software Engineering , vol. 3, no. 9, pp. 13–18, 2013.

[26] R.E. Burkard, S.E. Karisch and F. Rendl, “A Quadratic Assignment Problem Library”, http://www.opt.math.tu-graz.ac.at/qaplib/, [Online], Available: April 1996, [Accessed: 27 Feb. 2014].