Volume 3, Issue 9, September 2013 ISSN: 2277 128X
International Journal of Advanced Research in Computer Science and Software Engineering
Research Paper
Available online at: www.ijarcsse.com

Discrete Cat Swarm Optimization to Resolve the Traveling Salesman Problem

Abdelhamid Bouzidi, Mohammed Essaid Riffi
Laboratory Of Matic, Mathematics & Computer Science Dept
Faculty Of Sciences, Chouaïb Doukkali University
El Jadida, Morocco

Abstract: This research paper presents an adaptation of the cat swarm optimization (CSO) to solve the traveling salesman problem (TSP). This evolutionary algorithm appeared in 2007 by Chu and Tsai for optimization problems in the continuous case. To solve TSP, which is a discrete problem, we will describe the various operators and operations performed in two different modes of this algorithm, which is the searching mode (TM) and the tracing mode (SM). At the end, we will demonstrate the success of the proposed terms.

Keywords: Swarm intelligent, Swarm optimization, Discrete optimization, Evolutionary computing, Cat, Traveling salesman problem.

I. INTRODUCTION

The traveling salesman problem (TSP) [1], studied since the 19th century, is one of the known problems in the operational research .The importance of this problem appears in many application areas such as telecommunications, electronics, logistics, transportation, astronomy, industry, the scheduling problem among others. It's very difficult to solve this type of problem because it belongs to NP-complete problems class. But now, we are able to solve it with the emergence of heuristic. For example, local search ([4]-[5]), simulated annealing ([6]-[7]), and tabu ([8]-[9]). And by the emergence of meta-heuristic. For example, genetic algorithm (GA) ([10]-[13]), ant algorithm ([14]-[19]), particle swarm optimization ([20]-[24]), bee colony optimization ([26]-[28]).Now, with the content of this article we can solve this problem using the CSO algorithm. To prove the performance and convergence of the proposed algorithm, we will use TSPLIB instances (Library of TSP instances). In this research paper, the first section is about the presentation of the CSO [2]-[3] algorithm, the second contains a description of the TSP. The third is an adaptation of the CSO algorithm for TSP. The fourth section is concerned with the test results by using instances of TSPLIB. Finally, comes the conclusion in the last section.

II. CAT SWARM OPTIMIZATION

CSO [2-3] is an evolutionary optimization algorithm that models the natural behavior of cats. This behavior is represented in two modes:

- Searching mode (TM): Cats move slowly when resting.
- Tracing mode (SM): Cats move quickly when hunting.

These two modes are linked to each other by the mixing ratio (MR), which determines if a cat is going to be in a searching mode or in a tracing mode. Each cat is characterized by its position that represents the solution, its velocity and the flag that determines whether the cat is in tracing mode or in searching mode.

III. ADAPTATION OF CSO TO RESOLVE TSP:

First, we will define operators/the characteristic of each cat that we will use in the CSO algorithm.

Let **G(X , E)** a graph, with **X** set of Nodes, and **E** set of edges. For the TSP the nodes present the cities and the each edge of **E** present the path between every two cities from **X**. Let $\mathbf{x_i}$, $\mathbf{x_j}$ two city of X. The operators are:

1. **Position**: This is the solution represented by the Hamiltonian path
2. **Velocity**: the set of pair $(\mathbf{x_i}, \mathbf{x_j})$, which represents the permutation to be applied to the position, Let |v| the number of a pair $(\mathbf{x_i}, \mathbf{x_j})$. The velocity is defined by :

$$\mathbf{v=(i_k,j_k)[k:0 \rightarrow |v|]}$$

3. **Flag**: This allows us to determine if the cat is in searching mode or in tracing mode

Our objective, is finding the cat that have the best fitness. The fitness present the distance of the position (distance total of the Hamiltonian path).

A. Operation used in CSO algorithm:

This section is devoted to the description of the algebraic operations used in the tracing mode for the discrete case. It's similar to the discrete particle swarm optimization (PSO) algorithm for the traveling salesman problem, defined by Clerc [25] in 2004.

These operations will be performed on the velocity and the position of every cat in the tracing mode.

Before we begin, let **x** and **x'** two position / path of the salesman, and a velocity **v** represents all permutations to performed.

1) Opposite of velocity:

Is defined by: $\neg v=(i_k,j_k)[k : |v| \rightarrow 0]$,with: $v + \neg v = \emptyset$

2) Addition:

This is done between a position **x** and velocity **v**, in order to have a new position **x'**.

$$x + v = x'$$

Adding operation which translates the movement, represents the set of permutations to be applied to the position **x** to get a new position **x'**.

3) Subtraction (position - position):

This operation is performed between two positions to give a velocity.

$$x' - x = v$$

This is the opposite of the addition operation ($x'- x = v \Leftrightarrow x + v = x'$). In this case, by two positions **x** and **x'**, we will be solving all permutations performed on **x**, to obtain **x'**. These pairs of permutations **v** is the velocity.

4) Multiplication:

This operation is performed between a real **r** and velocity $v=(i_k,j_k)[k :0 \rightarrow |v|]$, the result is a velocity. The different possible cases according to the real **r** are:

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

B. Seeking mode:

1) Presentation:

This sub-mode is used to model each cat during his rest period, where it will be looking carefully around his surrounding for his next catch. As the continuous case, we are using four factors, except we're going to adapt them to the discrete case.

The four factors are:

- SMP: number of cats in search mode
- CDC: size selected to carry the mutation
- SRD: the first position to choose randomly
- SPC: a Boolean value indicating whether a cat can be selected for the in tracing modeor not

2) Description:

Seeking mode is described as follows:

Step 1: put j copies of the present position of the cat k, with j = SMP. If the value of SPC is true or j = SMP-1, and retain the cat as one of the candidates.

Step 2: Generate a random value of SRD

Step 3: If the fitness (FS) are not equal, calculate the probability of each candidate by equation **(a)**, the default probability value of each candidate is 1.

Step 4: Perform mutation and replace the current position.

$$Pi = \frac{|FSi - FSmax|}{\text{FSmax}-\text{FSmin}} \qquad \textbf{(a)}$$

C. Tracing mode:

1) Presentation

This is the hunting mode. Where, every cat traces its path, according to its own velocity.

2) Description

The action of every cat in the tracing mode can be described as follows:

Step 1: update the velocities of each $\textbf{cat}_k$ according to equation **(b)**.

Step 2: check if the velocities are of the highest order.

Step 3: update the position of k cat according to equation **(c)**.

$$V'_k = w*V_k + r_1 * c_1 * (X_{best} - X_k) \qquad \textbf{(b)}$$

Where:

X_{best}: is the best solution / position of the cat who has the best fitness value.

V_k : The old speed value (current value).

V'_k: the new value of the velocity obtained by the equation (2).

c1: is a constant.

r1: a random value in the range **[0, 1]**.

$$X'_k = X_k + V_k \qquad \textbf{(c)}$$

Where:

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 $_i$

D. Discussion about adding the "w" parameter:

In the continuous case, the value of the inertia parameter w is 1. But with this value for the traveling salesman problem, we can't usually reach the optimum, just rarely (in up to 2/10 iterations), and if we reach it, it requires a very large number of iterations (more than 100 iterations) in a very important execution time which takes sometimes a lot of days, as a simple or a hard instance of TSPLIB. Now, by changing the value of inertia, we have succeeded in obtaining the results in Table 1, which shows the performance of the algorithm by this modification of w.

E. Complete Mode:

Full mode CSO is composed of a searching mode and a tracing mode. These two modes are combined by the mixing ratio (MR), which will determine which mode each cat will have. The CSO algorithm is described as following

1) Creating N Cats with a null speed. The position $\mathbf{x_i}$ of each cat is randomly generated and initialized. $\mathbf{x_{best}}$ position that has the best fitness in the swarm.
2) Initialized the flag of each cat by MR.
3) Calculates the fitness of each cat, and update $\mathbf{x_{best}}$.
4) Changing the position of each cat with respect to its mode indicated by its flag
5) Updates the value of the flag of each cat according to MR.
6) Checks the stop condition. If yes, complete the program. If not, repeat 3), 4) and 5).

The flowchart of the algorithm is as follows:

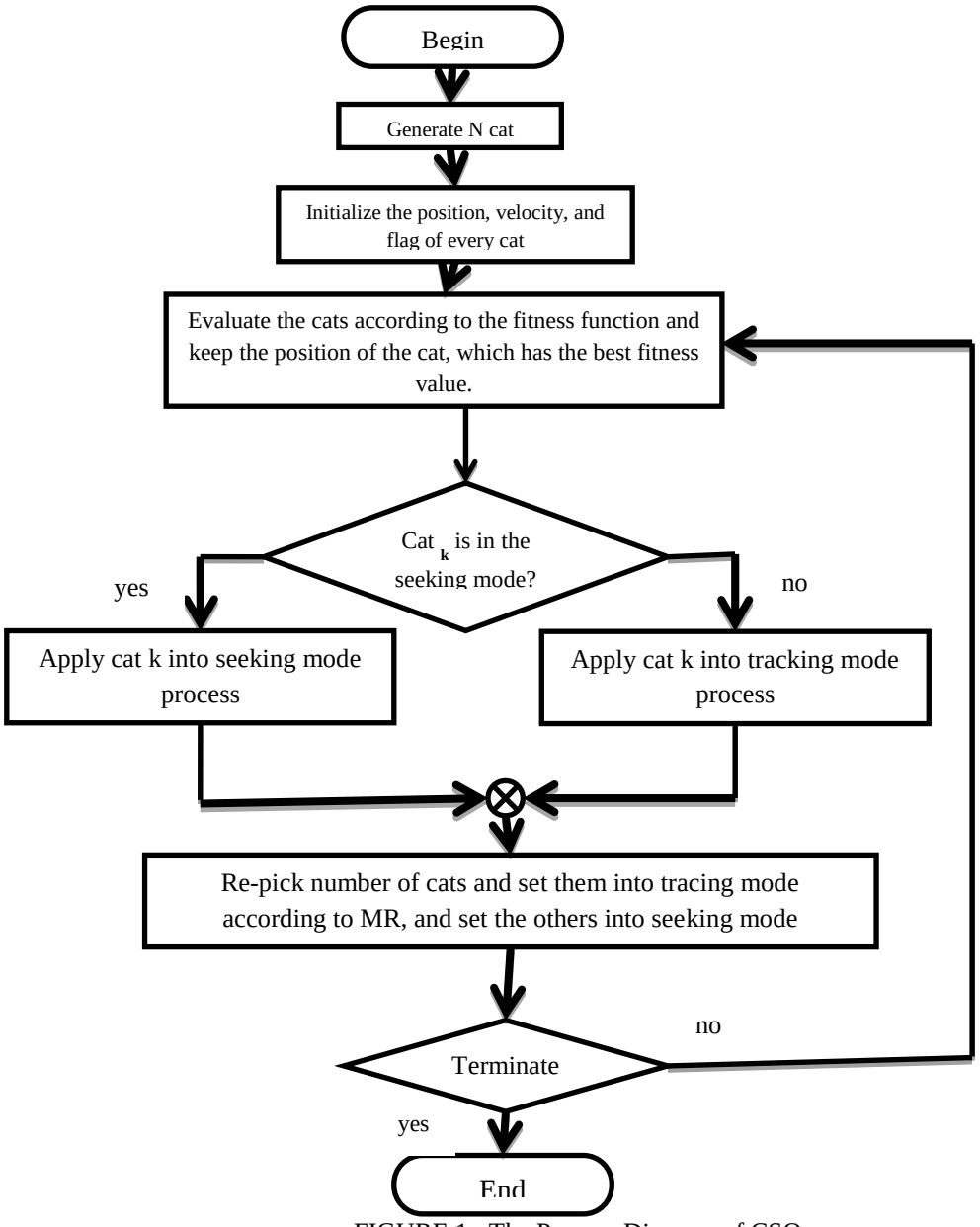

FIGURE 1: The Process Diagram of CSO

IV. EXPERIMENTS AND COMPUTATIONAL RESULTS

To check the validity of the proposed adaptation for the CSO algorithm in the discrete case, some instances of the TSPLIB library [21] are selected for the simulations. The experiments were performed on a PC with processor Intel(R) Core(TM) 2 Duo CPU T5800@ 2.00GHZ 2.00GHz and 3.00 MB of RAM. Each instance runs for 100 times.

This table contains the values of the used parameter.

Table 1: value of uses parameters

SMP	Number of cities
CDC	3
MR	33%
C1	2.05
R1	[0,1]
W	0.729

Table 2 shows the numerical results. The second column is the number of nodes **Nbr node**. The third represents the best result ***Opt*** given by the documentation TSPLIB. The fourth column is the best result ***BestR*** obtained by using the CSO algorithm. The fifth is the worst results ***WorstR*** for the selected instance. The percentage of error ***Err*** is calculated:

$$Err = \frac{\frac{(BestR+WorstR)}{2} - \mathbf{Opt}}{\mathbf{Opt}}$$

Table 2 : Numericals results obtained

Instance	*Nbr node*	*Optimum*	*BestR*	*WorstR*	*Err*
a280	280	2579	2579	2783	3.9550
berlin52	52	7542	7542	7693	1.0011
bier127	127	118282	118282	122128	1.6258
ch130	130	6110	6110	6394	2.3241
ch150	150	6528	6528	6782	2.0786
eil51	51	426	426	427	0,0799
eil76	101	538	538	549	1,02230
eil101	101	629	629	636	0,5564
gil262	262	2378	2378	2670	6.1396
kroA100	100	21282	21282	21717	1,0220
kroB100	100	22141	22141	23014	1,9715
kroC100	100	20749	20749	21669	2,2170
kroD100	100	21294	21294	22878	3,7195
rat99	99	1211	1211	1298	3,5921
rd100	100	7910	7910	8165	1.6119
st70	70	675	675	682	0,5185

For CDC, we have tested some values to a many instance of TSPLIB. The result obtained by changing the values CDC and mark number of being arrived to the best solution in teen iteration is:

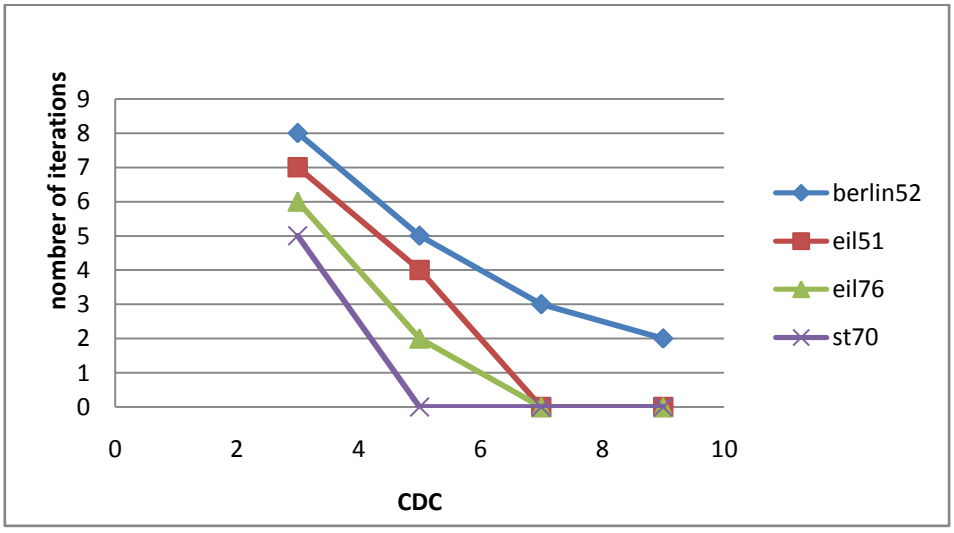

FIGURE 2: Graph represente by changing CDC values,the nbr to finding the best solution for berlin52 instance

According to the graph obtained in figure 2, we can clearly see that the best value to choose is **CDC = 3**.

V. CONCLUSION:

In this article we presented the adaptation of the algorithm cat swarm optimization, which has never been tested in the discrete case. The results obtained by testing on TSPLIB instances have demonstrated the performance of this algorithm. In the future we will like to adapt and apply CSO also for multi-objective discrete problems.

REFERENCES

[1] David L. Applegate, Robert E. Bixby, Vasek Chavtal, and William J. Cook, *The Problem*, Princeton University Press, United Kingdom,5-13,2006.

[2] S. C. Chu, and P. W. Tsai, "Computational Intelligence Based on the Behaviour of Cat", International Journal of Innovative Computing, Information and Control, 3 (1, pp.163-173), 2007.

[3] Meysam Orouskhani, Yasin Orouskhani, and Mohammad Teshnehlab, "A New Adaptive and Dynamic Strategy in Cat Swarm Optimization", *The AAth UK Workshop on Comptational Intelligence*, University of Manchester, pp.49-54,2011

[4] Johnson, D.S., McGeoch, L.A.: *The traveling salesman problem: A case study in local optimization*. In: Aarts, E.H.L., Lenstra, J.K. (eds.) Local Search in Combinatorial Optimization. Wiley (1997)

[5] Olaf Mersmann, Bernd Bischl, Jakob Bossek, Heike Trautmann, Markus Wagner, Frank Neumann ,"Local Search and the Traveling Salesman Problem: A Feature-Based Characterization of Problem Hardness", *Lecture Notes in Computer Science* ,pp. 115-129 ,2012.

[6] Emile H.L. Aarts, Jan H.M. Korst,"Boltzmann machines for travelling salesman problems",*European Journal of Operational Research*, Vol 39, pp 79–95, 6 March 1989.

[7] James R.A Allwright, D.B Carpenter,"A distributed implementation of simulated annealing for the travelling salesman problem",*Parallel Computing*,Vol 10,pp 335-338, May 1989

[8] Michel Gendreau, Gilbert Laporte, Frederic Semeta, "A tabu search heuristic for the undirected selective travelling salesman problem", *European Journal of Operational Research*,Vol 106, pp 539–545, 16 April 1998

[9] James R.A Allwright, D.B Carpenter, "A Tabu Search Approach for the Prize Collecting Traveling Salesman Problem", *Electronic Notes in Discrete Mathematics*,Vol 41,pp 261-268, 5 June 2013

[10] S.-M. Chen and C.-Y. Chien, "Solving the traveling salesman problem based on the genetic simulated annealing ant colony system with particle swarm optimization techniques", *Expert Systems with Applications*, vol. 38, no. 12, pp. 14439–14450, Nov. 2011.

[11] C. Moon, J. Kim, G. Choi, and Y. Seo, "An efficient genetic algorithm for the traveling salesman problem with precedence constraints," *European Journal of Operational Research*, vol. 140, no. 3, pp. 606–617, Aug. 2002.

[12] Y. Nagata and D. Soler, "A new genetic algorithm for the asymmetric traveling salesman problem," *Expert Systems with Applications*, vol. 39, no. 10, pp. 8947–8953, Aug. 2012.

[13] J. Yang, C. Wu, H. P. Lee, and Y. Liang, "Solving traveling salesman problems using generalized chromosome genetic algorithm," *Progress in Natural Science*, vol. 18, no. 7, pp. 887–892, Jul. 2008

[14] Jun-man, K., and Yi, Z.: "Application of an Improved Ant Colony Optimization on Generalized Traveling Salesman Problem", *Energy Procedia*, 2012, 17, pp. 319-325

[15] Cheng, C.-B., and Mao, C.-P.: "A modified ant colony system for solving the travelling salesman problem with time windows", *Mathematical and Computer Modelling*, 2007, 46, pp. 1225-1235

[16] Ghafurian, S., and Javadian, N.: "An ant colony algorithm for solving fixed destination multi-depot multiple traveling salesmen problems", *Applied Soft Computing*, 2011, 11, pp. 1256-1262

[17] Dong, G., Guo, W.W., and Tickle, K.: "Solving the traveling salesman problem using cooperative genetic ant systems", *Expert Systems with Applications*, 2012, 39, pp. 5006-5011

[18] Bai, J., Yang, G.-K., Chen, Y.-W., Hu, L.-S., and Pan, C.-C.: "A model induced max-min ant colony optimization for asymmetric traveling salesman problem", *Applied Soft Computing*, 2013, 13, pp. 1365-1375.

[19] Tsai, C.: "A new hybrid heuristic approach for solving large traveling salesman problem3, *Information Sciences*, 2004, 166, pp. 67-81.

[20] Goldbarg, E. F. G., Goldbarg, M. C., & Souza, G. R. De. (2008). *Particle Swarm Optimization Algorithm for the Traveling Salesman Problem*, (September).

[21] Shi, X. H., Liang, Y. C., Lee, H. P., Lu, C., & Wang, Q. X. (2007). "Particle swarm optimization-based algorithms for TSP and generalized TSP". *Information Processing Letters*, *103*, pp. 169–176.

[22] Fan, H. (2010). "Discrete Particle Swarm Optimization for TSP based on Neighborhood", *10*, pp. 3407–3414.

[23] Labed, S., Gherboudj, A., & Chikhi, S. (2012). "A Modified Hybrid Particle Swarm Optimization Algorithm For Solving The Traveling Salesmen",Vol 39, No. 2, pp. 132–138.

[24] Tasgetiren, M. F., Suganthan, P. N., & Pan, Q.-Q. (2007). A discrete particle swarm optimization algorithm for the generalized traveling salesman problem. Proceedings of the 9th annual conference on Genetic and evolutionary computation - GECCO '07, pp. 158-165.

[25] Clerc, M. "Discrete Particle Swarm Optimization, Illustrated by Traveling Salesman Problem". *In New Optimization Techniques in Engineering*. Springer-Verlag, Berlin , 2004.

[26] Li-Pei Wong, Low M.Y.H., Chin Soon Chong, "Bee Colony Optimization with local search for traveling salesman problem", *Industrial Informatics*, pp 1019-1025, 2008.

[27] James R.A Allwright, D.B Carpenter, "A Bee Colony Optimization Algorithm for Traveling Salesman Problem ", *Modeling & Simulation*, pp 818-823, May 2008.

[28] Girsang A.S.,Chun-Wei Tsai, Chu-Sing Yang, "A Fast Bee Colony Optimization for Traveling Salesman Problem", *Innovations in Bio-Inspired Computing and Applications (IBICA)*, pp 7-12, Sept 2012.

