

# A Bi-Objective Cat Swarm Optimization Algorithm for Set Covering Problem

Broderick Crawford, Ricardo Soto, Hugo Caballero  
and Eduardo Olgúin

**Abstract** In this paper, we study a classical problem in combinatorics and computer science, Set Covering Problem. It is one of Karp's 21 NP-complete problems, using a new and original metaheuristic, Cat Swarm Optimization. This algorithm imitates the domestic cat through two states: seeking and tracing mode. The OR-Library of Beasley instances were used for the benchmark with additional fitness function, thus the problem was transformed from Mono-objective to Bi-objective. The Cat Swarm Optimization finds a set solution non-dominated based on Pareto concepts, and an external file for storing them. The results are promising for further continue in future work optimizing this problem.

**Keywords** Multiobjective problems · Evolutionary algorithm · Swarm optimization · Cat swarm optimization · Multiobjective cat swarm optimization · Pareto dominance

## 1 Introduction

Optimization problems require complex and optimal solutions because they relate to distribute limited basic resources. To resolve these problems it means improving the lives of poor people directly and enabling the growth of businesses, for example:

---

B. Crawford · R. Soto · H. Caballero (✉)  
Pontificia Universidad Católica de Valparaíso, Valparaíso, Chile  
e-mail: hcaballec@gmail.com

B. Crawford · E. Olgúin  
Universidad San Sebastián, Santiago Metropolitan Region, Chile

B. Crawford  
Universidad Central de Chile, Santiago Metropolitan Region, Chile

R. Soto  
Universidad Autónoma de Chile, Temuco, Chile

R. Soto  
Universidad Científica Del Sur, Lima, Peru

© Springer International Publishing Switzerland 2016  
R. Silhavy et al. (eds.), *Artificial Intelligence Perspectives in Intelligent Systems*,  
Advances in Intelligent Systems and Computing 464,  
DOI 10.1007/978-3-319-33625-1\_44

resources related to social welfare, reaction by natural disasters, medical distribution capabilities. For these reasons the optimization generates a wide area of research in the sciences of Operations Research and Computer.

In the last decades bio-inspired algorithms have called the attention of researchers, in particular the heuristic Particle Swarm Optimization, which is based the behavior of some species: Bugs, fish, felines. These species use the collective intelligence to reach specific objectives guided by some community member. This paper is focused on studying the heuristic based on the behavior of domestic cats to solve a classical problem the Set Covering Problem (SCP).

## 2 Multi Objective

Decision problems involves multiple evaluation criteria and generally they are in conflict. To resolve a multi objective problem it required to optimize multiple criteria simultaneously. Exists a wide variety of cases in our society, for example: vehicle route optimization, environmental problems, allocation of medical resources. The solution to multi-objective optimization problem it is presented by a set of feasible solutions, and the best of them define a set of non-dominated solutions, this set we will call Front. Formally the multi objective problem is defined as:

$$\min z(x) = [z_1(x), z_2(x), z_3(x), z_4(x), \dots, z_M(x)] \quad (1)$$

The goal consists in minimizing a function  $z$  with  $M$  components with a vector variable  $x = (x_1, \dots, x_n)$  in a universe  $U$ , i.e., A solution  $u$  dominates  $v$  if  $u$  performs at least as well as  $v$  across all the objectives and performs better than  $v$  in at least one objective.

### 2.1 Objective Space

The dimensions of the target area corresponding to the number of functions to optimize. In this single-objective problem is one-dimensional space, since each decision vector corresponds to only a scalar number. In multi-objective problems, this is multi-dimensional space, where each dimension corresponds to each objective function to be optimized.

### 2.2 Pareto Dominance

If we have two candidate solutions  $u$  and  $v$  from  $U$ , vector  $z(u)$  is said to dominate vector  $z(v)$  denoted by:  $z(u) < z(v)$ , if and only if,

$$z_i(u) \leq z_i(v), \quad \forall i \in \{1, \dots, M\} \tag{2}$$

$$z_i(u) \leq z_i(v), \quad \exists i \in \{1, \dots, M\} \tag{3}$$

If solution  $u$  is not dominated by any other solution, then  $u$  is declared as a Non Dominated or Pareto Optimal Solution. There are no superior solutions to the problem than  $u$ , although there may be other equally good solutions (3).

### 3 Set Covering Problem

SCP is defined as a fundamental problem in Operations Research and often described as a problem of coverage of  $m$ -rows  $n$ -columns of a binary matrix by a subset of columns to a minimum cost [1]. It is one of Karp’s 21 NP-complete problems. This is the problem of covering the rows of an  $m$ -row,  $n$  column, zero-one  $m \times n$  matrix  $a_{ij}$  by a subset of the columns at minimal cost. Formally, the problem can be defined as:

Defining  $x_j = 1$  if column  $j$  with cost  $c_j$  is in the solution and  $x_j = 0$  otherwise

$$\text{Minimize } Z = \sum_{j=1}^n c_j x_j \quad j \in \{1, 2, 3, \dots, n\} \tag{4}$$

Subject to:

$$\sum_{j=1}^n a_{ij} x_j \geq 1 \quad i \in \{1, 2, 3, \dots, m\} \tag{5}$$

$$x_j \in \{0, 1\} \tag{6}$$

This definition contains a one fitness function, there is just one objective to be optimized. We study the case for two objective functions, using meta heuristic Cat Swarm Optimization (CSO) and using position vector of ones and zeros. A complete case study of SCP using CSO was done **Pontificia Universidad Católica de Valparaíso** [2].

#### 3.1 Set Covering Problem Bi Objective (SCPBO)

This work focuses on solving the SCP with two fitness functions, i.e., textit  $M = 2$ . To ensure the fitness functions have opposed criteria the second cost vector will be transposed the first, therefore the definition will be:

$$c_2 = (c_1)^t \tag{7}$$

$$\min z(x) = [z_1(x), z_2(x)] \quad (8)$$

$$\text{Minimize } Z_1 = \sum_{j=1}^n c_j^1 x_j \quad j \in \{1, 2, 3, \dots, n\} \quad (9)$$

$$\text{Minimize } Z_2 = \sum_{j=1}^n c_j^2 x_j \quad j \in \{1, 2, 3, \dots, n\} \quad (10)$$

Subject to:

$$\sum_{j=1}^n a_{ij} x_j \geq 1 \quad i \in \{1, 2, 3, \dots, m\} \quad (11)$$

## 4 Cat Swarm Optimization CSO

Some species of felines shows similar behavior when they are hunting, commonly hunt in packs, some of them remain on alert and others run after their prey. They work with a common purpose, the prey. The CSO was introduced was in 2016 original version of CSO by Chu, Tsai, and Pan. They observed the behavior of the cats and modeled their behavior. Based on their studies they suggested that cats have two modes of behavior:

- (a) Seeking mode: Cat spends most of the time when they are awake on resting. While they are resting, they move their position carefully and slowly.
- (b) Tracing mode: cats change their position according to its own velocities for every dimension.

These states have been mathematically modeled y both sets of cats are used to reach a goal, that is, hunt prey. The position of each cat represent a solution set, has position and velocity for each dimension and a fitness value. Additionally a flag is used to identify whether the cat is in seeking mode or tracing mode. [3, 4] have shown that the CSO performs better than PSO with respect to convergence speed and residual mean square error, but it requires higher computation time.

### 4.1 Algorithm

The CSO algorithm works with a set of parameters that configure the behavior of the pack:

- NC: Number of population or pack cats
- MR: Mixture Rate that defines number of cats mode, this parameter must be chosen between 0 and 1. Define what percentage of cats are in seeking mode and tracing mode

Both group works with a reaches its optimal solution. The flowchart of this algorithm is shown in Fig. 1 and the and a description of the actions are outlined below.

- Randomly initialize the position of cats in D-dimensional space i.e.  $X_{id}$  representing position of  $i$  th cat in  $d$  th dimension.
- Randomly initialize the velocity of cats i.e.  $V_{id}$ .
- According to MR, cats are randomly picked from the population and their flag is set to seeking mode, and for others the flag is set to tracing mode.
- Evaluate both objective function for each cat.
- Store the position of the cats representing non-dominated solutions in the archive.
- If  $i$ th cat is in seeking mode, apply the cat to the seeking mode process, otherwise apply it to the tracing mode process. Check the termination condition, if satisfied, terminate the program. Otherwise repeat steps c to e.

## 4.2 Seeking Mode

The seeking mode corresponds to a global search technique in the search space of the optimization problem. A term used in this mode is seeking memory pool (SMP). It is the number of copies of a cat produced in seeking mode.

There are four essential factors in this mode: seeking memory pool (SMP), seeking range of the selected dimension (SRD), counts of dimension to change (CDC), and self-position considering (SPC).

- SMP is used to define the size of seeking memory for each cat. SMP indicates the points explored by the cat. This parameter can be different for different cats.
- SRD declares the mutation ratio for the selected dimensions.
- CDC indicates how many dimensions will be varied.
- SPC is a Boolean flag, which decides whether current position of cat

The steps involved in this mode are:

- Create  $T$  ( $=SMP$ ) copies of  $j$  th cat i.e.  $Y_k d$  where  $(1 \leq k \leq T)$  and  $(1 \leq d \leq D)$ .  $D$  is the total number of dimensions.
- Apply a mutation operator to  $Y_k$ .
- Evaluate the fitness of all mutated copies.
- Update the contents of the archive with the position of those mutated copies which represent non dominated solutions.
- Pick a candidate randomly from  $T$  copies and place it at the position of  $j$ th cat.

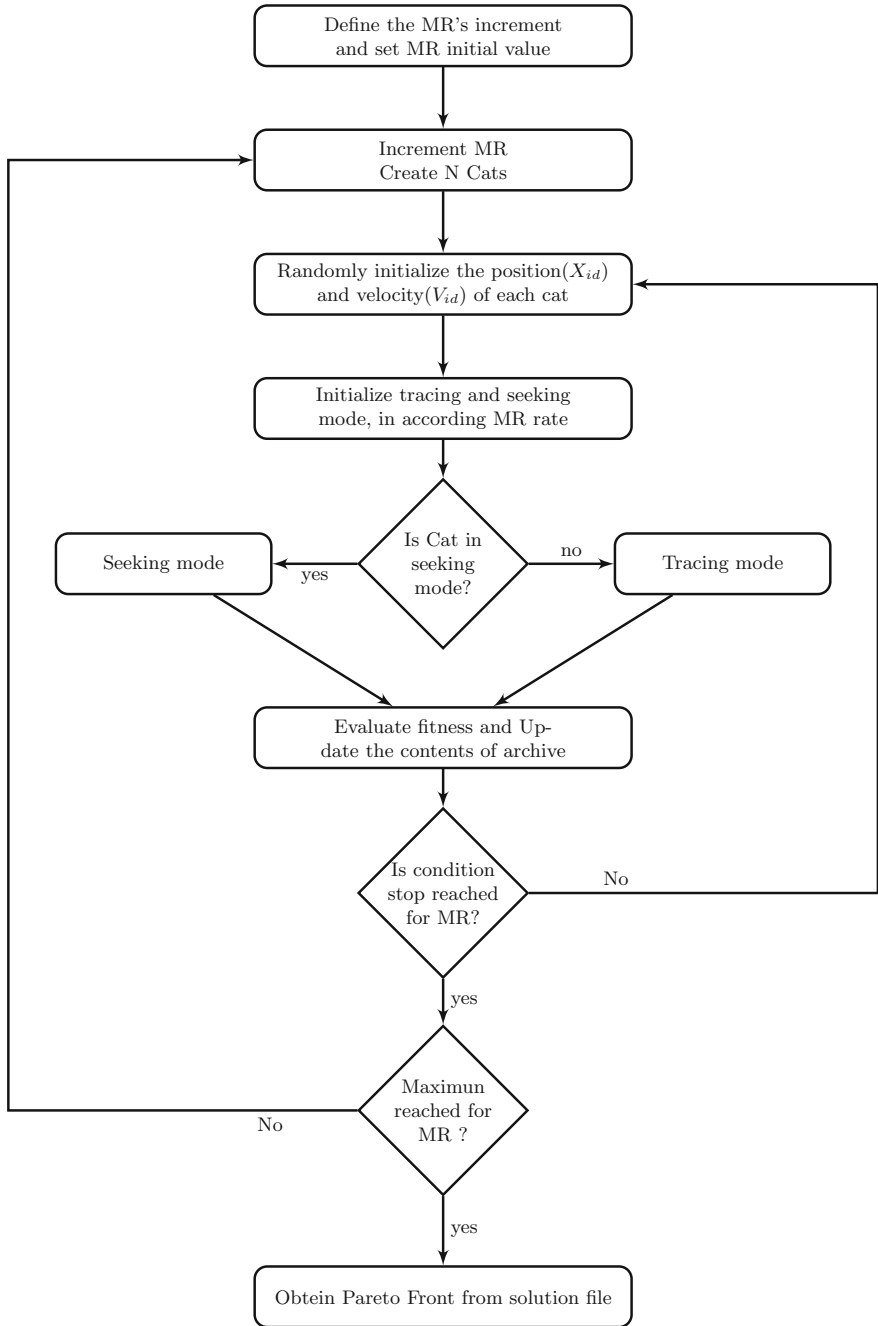


Fig. 1 Experimental workflow

### 4.3 Tracing Mode [4]

The tracing mode corresponds to a local search technique for the optimization problem. In this mode, the cat traces the target while spending high energy. The rapid chase of the cat is mathematically modeled as a large change in its position. Define position and velocity of  $i$ th cat in the  $D$ -dimensional space as  $X_i = (X_{i1}, X_{i2}, X_{i3} \dots X_{iD})$  and  $V_i = (V_{i1}, V_{i2}, V_{i3} \dots V_{iD})$  where  $(1 \leq d \leq D)$  represents the dimension. The global best position of the cat swarm is represented as  $X_g = (X_{g1}, X_{g2}, X_{g3} \dots X_{gD})$ . The steps involved in tracing mode are:

- (a) Compute the new velocity of  $i$ th cat using (13)

$$V_{id} = w * V_{id} + c * r * (X_{gd} - X_{id}) \tag{12}$$

where

- $w$  = is the inertia weight
- $c$  = is the acceleration constant
- $r$  = is a random number uniformly distributed in the range  $[0, 1]$

- (b) Compute the new position of  $i$ th cat using

$$X_{id} = X_{gd} - X_{id} \tag{13}$$

- (c) If the new position of  $i$ th cat corresponding to any dimension goes beyond the search space, then the corresponding boundary value is assigned to that dimension and the velocity corresponding to that dimension is multiplied by  $-1$  to continue the search in the opposite direction.
- (d) Evaluate the fitness of the cats.
- (e) Update the contents of the archive with the position of those cats which represent no dominated vectors.

### 4.4 Flow Chart Diagram

The Fig. 1 shows the flow of processes to solve SCP-BO. The parameter that produces a greater quantity of solutions, is MR. It was determined experimentally, from 0.5 until 0.9. For this experiment we define an increment of 0.1

- (a) Initiate  $MR_p$  in min value (= 0.5)
- (b) Create the cat swam, N cats working to solve the problem
- (c) Define, randomly the position and velocity for each cat
- (d) Distribute the swarm in tracing and seeking mode based on MR

- (e) Check if the cat is feasible solution (Eq. 11). if the cat satisfies the restriction compute the fitness (Eqs. 9 and 10) and compare with the non dominated solutions in the archive
- (f) Update de solution file
- (g) If number iteration less than the max iteration continue work, goto step c
- (h) If  $MR_p$  less than max value MR, increment  $MR_p$  and go to step b
- (i) Calculate the pareto front from non domination file

## 5 Experimental Results

The BCSO was evaluated using the next features:

- (a) Using 4 of 65 Instances for set covering from OR-Library of Beasley [5]
- (b) MacBook Pro (13-inch, Mid 2012), CPU MacBook Pro (13-inch, Mid 2012), 16 GB 1333 MHz DDR3, OS X Yosemite, version 10.10.5
- (c) IDE: BlueJ version 3.1.5 (Java version 1.8.0\_31)

The working conditions of the process were:

- (a) 1.500 iterations for each varying from  $MR = 0.5$  until  $MR = 0.7$ , using an increment 0.1
- (b) 30 times each Beasley instance
- (c) The Optimal Pareto Front for each instance was obtained varying MR from 0.1 until 0.99 and determined by the union of fronts obtained MR
- (d) The parameters used BCO was obtained from [2, 4] and show in Tables 1 and 2

**Table 1** Parameter values CSO

Name	Parameter	Value	Obs
Number of cats	C	30	
Mixture ratio	MR	0.5	
Seeking memory pool	SMP	20	–
Probability of mutation	PMO	1	–
Counts of dimensions to change	CDC	0,001	–
Inertia weight	w	1	–
Factor $c_1$	$c_1$	1	–



**Table 2** Experimental results: MAX, MIN, PROM, DESV in sec

INST	HYPER	SPREAD	MAX	MIN	PROM	DESV
scp41	0,6223	0,85	132,441	109,345	118,84	7,48
scp42	0,6845	1,03	156,556	121,211	143,1	10,03
scp43	0,7261	0,91	135,301	115,309	125,22	6,53
scp44	0,5804	1,02	154,609	129,51	140,53	6,38
scp45	0,7426	1,11	134,763	105,963	119,49	9,28
scp46	0,5435	1,21	140,833	114,68	134,02	7,41
scp47	0,5172	1,05	147,812	126,058	136,42	7,26
scp48	0,7319	0,92	135,586	114,344	120,57	7,09
scp49	0,6029	1,00	159,194	135,4	148,21	7,13
scp51	0,6156	1,24	270,516	247,489	256,56	7,63
scp52	0,6378	0,87	282,612	259,742	270,77	7,24
scp53	0,6611	0,99	257,966	203,538	229,88	17,42
scp54	0,8511	1,05	259,181	212,809	241,01	15,83
scp55	0,5872	1,14	234,381	205,496	225,25	9,034
scp56	0,7223	1,07	265,601	218,673	238,11	14,73
scp57	0,6036	1,14	259,252	234,426	245,85	8,5
scp58	0,6242	0,98	270,754	242,436	254,9	9,502
scp59	0,5338	1,07	243,131	209,511	227,58	11,92
scp61	0,5992	0,93	103,339	81,946	94,31	7,73
scp62	0,6673	1,04	100,748	79,064	91,99	8,47
scp63	0,6873	1,01	96,555	77,817	86,94	6,07
scp64	0,6361	1,34	103,206	78,183	90,4	9,28
scp65	0,6696	0,99	101,831	78,088	90,66	7,24
scpa1	0,7834	0,91	506,951	463,377	482,7	14,94
scpa2	0,5462	1,21	618,465	513,501	559,59	34,73
scpa3	0,5631	1,04	517,718	474,45	496,63	13,89
scpa4	0,6269	1,12	526,053	469,132	502,76	17,68
scpa5	0,7679	1,12	529,614	445,58	488,48	27,49
scpb1	0,6986	1,13	108,345	100,11	104,23	2,613
scpb2	0,6071	1,12	106,523	100,768	103,81	2,14
scpb3	0,7764	1,13	106,237	102,402	104,37	1,35
scpb4	0,5971	1,21	108,057	101,961	105,19	2,58
scpb5	0,7009	0,97	351,324	307,065	330,22	13,41
scpc1	0,6049	1,23	360,421	340,318	350,72	9,88
scpc2	0,6412	1,13	345,952	333,604	342,16	6,17
scpc3	0,6157	1,01	368,203	330,924	349,43	10,17
scpc4	0,5932	1,01	409,037	374,056	384,78	11,75
scpc5	9,6481	0,99	981,574	894,662	932,12	26,51

## 6 Conclusion

There are not published results on a SCP Bi Objective. We think the Pareto Front is quite promising considering just we varied only MR. We also believe varying the population of cats with the best value of MR, we will improve the results. In this first phase of our research we only work with MR parameters, however we think that using adaptive techniques for parameters: seeking memory pool (SMP), seeking range of the selected dimension (SRD), counts of dimension to change (CDC), and self-position considering, we improve our results.

The next step:

- (a) To use adaptive techniques for CSO parameters
- (b) To obtain a Pareto optimal front using GA and PSO algorithm

in this work an analysis of quality of results should be done using the metric of comparison. The hypervolume considers aspects of convergence and diversity in a particular front, which would be a good metric to evaluate our results [6].

**Acknowledgments** The author Broderick Crawford is supported by grant CONICYT/FONDECYT/REGULAR/1140897 and Ricardo Soto is supported by grant CONICYT/FONDECYT/INICIACION/11130459

## References

1. Caprara, A., Toth, P., Fischetti, M.: Algorithms for the set covering problem. *Ann. Oper. Res.* **98**(1–4), 353–371 (2000)
2. Crawford, B., Soto, R., Berrios, N., Johnson, F., Paredes, F.: Binary cat swarm optimization for the set covering problem, pp. 1–4 (2015)
3. Chu, S.-C., Tsai, P.-W.: Computational intelligence based on the behavior of cats. *Int. J. Innov. Comput. Inf. Control* **3**(1), 163–173 (2007)
4. Pradhan, P.M., Panda, G.: Solving multiobjective problems using cat swarm optimization. *Exp. Syst. Appl.* **39**(3), 2956–2964 (2012)
5. Beasley, J.E.: A lagrangian heuristic for set-covering problems. *Nav. Res. Logist. (NRL)* **37**(1), 151–164 (1990)
6. Knowles, J., Corne, D.: The pareto archived evolution strategy: a new baseline algorithm for pareto multiobjective optimisation. In: *Proceedings of the 1999 Congress on Evolutionary Computation, 1999. CEC 99., vol. 1. IEEE* (1999)