International Journal of Recent Technology and Engineering (IJRTE)
ISSN: 2277-3878, Volume-8, Issue-1S4, June 2019

Multi-objective Optimization using Cricket Chirping Algorithm

Jonti Deuri, S. Siva Sathya

Abstract--- ***Utmost real world optimization problems are typically multi-objective with complex constraints. Now-a-days the heuristics and meta-heuristics approaches are becoming more powerful to solve these optimization problems considerably to direct approach. Cricket Chirping Algorithm (CCA) is a metaheuristics approach developed based on the chirping behavior of cricket for solving optimization problem. In this paper the cricket chirping algorithm for single objective optimization is extended for solving multi-objective optimization problems by adopting the aggressive behavior of cricket. The proposed Multi-objective Optimization using Cricket Chirping Algorithm(MOCCA) is first validated with a subset of benchmark test functions and compared with the other metaheuristics algorithm like MOPSO, NSGA-II and SPEA-II. The performance, efficiency and robustness of the proposed algorithm are experimented and statistically analyzed and it is proved the superiority of the MOCCA to other methods in terms of diversity and convergence of the solution. It can provide optimal or near optimal solutions for a wide range of problems.***

Keywords--- ***Multi-objective Optimization, Cricket Chirping Algorithm, Meta-heuristics, Optimization Problem.***

I. INTRODUCTION

The Multi-Objective Optimization (MOO) problems consist of multiple objectives with complex and highly non-linear constraints. Sometimes the multiple objectives may conflict with each other i.e. an improvement of one objective may lead to plummeting of another or sometimes the true optimal solutions may not exist and some approximations are needed. Often, there is no single optimal solution, but rather a set of alternative solutions. Even single objective global optimization problem is also not easy to find the global optimality if it consists of highly nonlinear design functions. In recent days, Meta-heuristic algorithms are becoming more powerful algorithm to solve these types of problems [1].The best Trade-off among the objectives is a set of solutions which is called Pareto optimal solutions and these are non-dominated with each other when consider all the objectives. This set is called Pareto optimal set and the corresponding objectives vector is called Pareto optimal Front [2]. The main goal of multi-objective optimization problem is to find a set of representative solution which are uniformly distributed and as close to the true Pareto front as possible. The fitness functions can be evaluated either by using weighted sum approach or Pareto ranking approach. Several MOO algorithms fall into either of the two approaches. But they basically differ in the fitness function evaluation procedure. There are many powerful algorithms for multi-objective optimization with successful application [3]-[6]. In addition metaheuristics approaches emerge as a major player for multi-objective global optimization. They often harness the successful characteristics of nature, especially biological, physical, chemical and many more new algorithms are emerging with many applications [7]-[8].

The Cricket Chirping Algorithm (CCA) has been developed by JontiDeuri and S. Siva Sathya [9] for solving single objective optimization problem by adopting the cricket's chirping behaviour. In this paper, CCA is extended to solve multi-objective optimization problems and formulate a Multi-Objective Cricket Chirping Algorithm (MOCCA). The MOCCA uses the concept of Pareto dominance for solving MOO problems and a set of solution that balances the objectives are obtained. Different metrics such as Generational Distance, Spacing, and Maximum Spread are used to analyse and validate the performance of MOCCA.

The remaining part of the paper is organized as follows: Section II describes the literature review of this field. The preliminary CCA is discussed in Section III then the proposed multi objective optimization using CCA is elaborated in Section IV. The results and finding are shown in Section V following the comparison with its counterpart in Section VI. The statistical analysis is shown in chapter VII and finally the conclusion and future works note on Section VIII followed by the references.

II. LITERATURE REVIEW

Over the last few decades, population based meta-heuristics algorithms are becoming more popular for solving multi-objective optimization problems. Evolutionary approaches and swarm optimizations are two most powerful methods in population based meta-heuristics [10]. The simple genetic algorithm was extended by Grefenstette's GENESIS program to include multiple objective functions that is called Vector Evaluated Genetic Algorithm (VEGA) [11]-[12]. In VEGA the populations are divided into N equal subpopulation assigning a fitness based on different objective functions. In 1993, Srinivas and Deb developed a Non-dominated Sorting Genetic Algorithm (NSGA) based on several layers of classifications of the individuals. The populations are ranked on the basis of non-domination and all non-dominated individuals are classified into one category and shared with their dummy fitness values to maintain the diversity of the population. The NSGA was improved by Dev et al. [13] called NSGA-II that is more efficient to NSGA. It uses elitism and a crowded comparison operator but does not use an external memory and no additional parameter for diversity.

Manuscript received June 10, 2019.

Jonti Deuri, Department of Computer Science, Pondicherry University, Pondicherry, India. (e-mail: jontideuri@gmail.com)

S. Siva Sathya, Department of Computer Science, Pondicherry University, Pondicherry, India. (e-mail: ssivasathya@gmail.com)

Published By:
Blue Eyes Intelligence Engineering
& Sciences Publication

Niched Pareto Genetic Algorithm (NPGA) was proposed by Horn et. al. [14] using tournament selection scheme based on Pareto dominance [15] and was improved by Erickson et al. where Pareto rankings are used but keeps tournament selection that was named as NPGA-2. There is no external memory is used and elitism mechanism is same as NPGA. When both competitors were either dominated or non-dominated (i.e., there is a tie), the result of the tournament was decided through fitness sharing. Multi-Objective Genetic Algorithm (MOGA) was proposed by Fonseca and Fleming[16] where the rank of a certain individual corresponds to the number of chromosomes in the current population by which it is dominated. Knowles and Corne introduced the Pareto Archived Evolution Strategy (PAES) [17] which consists of (1+1) evolution strategy, i.e. a single parent can generates only single offspring, and combined with an archive to record the non-dominated solutions previously found. Strength Pareto Evolutionary Algorithm (SPEA) was proposed by E. Zitzler and L. Thiele [18] that uses archive containing non-dominated solution. At each generation all non-dominated population members are copied to the archive and any dominated individuals or duplicates (regarding the objective values) are removed from the archive during the update operation. If the size of the updated archive exceeds a predefined limit, further archive members are deleted by a clustering technique which preserves the characteristics of the non-dominated front. Zitzler, et. al. [19] improved the SPEA, which was known as SPEA-II, in terms of three functions schemes i.e., fitness assignment scheme, which takes for each individual into account how many individuals it dominates and it is dominated by, a nearest neighbour density estimation technique, which allows a more precise guidance of the search process and a new archive truncation method that, guarantees the preservation of boundary solutions. The main difference from SPEA to SPEA-II is in archive updating operation. MOEA based on decomposition (MOEA/D) was developed byQ. Zhanget. al. [20] based on conventional aggregation approaches in which an MOO problem is decomposed into a number of scalar objective optimization problems (SOPs), also called sub-problems.

In 1995 Eberhart and Kennedy developed a population-based stochastic optimization technique called Particle swarm optimization (PSO) inspired by the social behaviour of bird flocking or fish schooling. PSO was extended to multi-objective optimization by Moore and Chapman [21]. The majority of the currently proposed MOPSO approaches define the concept of leaders. Each particle might have a set of different leaders from which just one can be selected in order to update its position. Such a set of leaders is usually stored in a different place from the swarm, which is called external archive; this is a repository in which the non-dominated solutions found so far are stored. The solutions contained in the external archive are used as leaders when the positions of the particles of the swarm have to be updated. Furthermore, the contents of the external archive are also usually reported as the final output of the algorithm. X. Hu and R. Eberhart [22] proposed a Dynamic Neighbourhood PSO by using Dynamic Neighbourhood strategy, new particle updating and one dimensional optimization to deal with multi-objective. The Dynamic Neighbourhood PSO was modified by using extended memory to store the global optimal solution and extend the original method [23] using PSO with a secondary repository of particles that is later used by other particles to guide their own flight and mutation operator that enriches the exploratory capabilities of the algorithm[24][25]. The disadvantage of original method was the multi-frontal problem which was overcome in the extended version.

Although, population-based search algorithms produce convenient results, there is no any single heuristic algorithm that could provide superior performance than others in solving all optimizing problems. In other words, an algorithm may solve some problems better and some problems worse than others [26]. Hence, proposing new high performance heuristic algorithms are welcome.

III. BASIC CRICKET CHIRPING ALGORITHM

Cricket is an insect similar to grasshopper with flattened body that makes a sound which is known as chirping. Many animals use acoustic signals for intra-specific (within species) communication. For example, birds sing, frogs croak, and crickets chirp etc. The cricket uses this chirping mainly for mating and aggression. Based on this chirping behaviour of cricket a new algorithm “Cricket Chirping Algorithm (CCA)” was proposed by J. Deuri and S. S. Sathya [9], [27]. The chirping of cricket is categorized as calling chirp when the cricket chirps for mating, courtship chirp when the male cricket attempt to mate with the female, copulatory chirp after successful mating and aggressive chirp when the cricket combat with other cricket.

In CCA each cricket is assumed to be a solution in the search space and is characterized by its position in the search space. Out of the total cricket population, few of them are determined by the user are designated as female population. By nature, only the male crickets can chirp and its chirping rate is based on the outside temperature. The male cricket may chirp for mating or aggression. Based on their chirping rate in certain temperature the cricket moves to a new position by emitting a mating song and mate with females crickets and produce offspring. The offspring represents a new position of the cricket. By emitting an aggressive song, they fight with other male crickets and the winner cricket reach new positions in the search space. The cricket which has the highest fitness will be selected as winner cricket. So for simplicity, Crickets are assumed to be in two states: they might chirp for mating and for aggression.

- First when the male cricket produces calling chirps for mating, they emit a peculiar sound and the female crickets are attracted and they move towards female cricket. After mating they produce offspring, which means they are taken to new positions in the search space.
- Second, when the cricket chirps for aggression, they emit an aggressive chirp and other male crickets are warned and make combat. All crickets may not be chirping for aggression.

Published By:
Blue Eyes Intelligence Engineering
& Sciences Publication

- For simplicity, we can use a simple representation that the probability of chirping for aggression is P_{agg} which is in between [0, 1]. When a cricket chirps for aggression, it is assumed that they randomly walk to another male cricket and fight. The winning cricket takes the place of the solution and removes the loser cricket. For more details of CCA with algorithm and flowchart, interested reader can refer [9][27][28][29].

IV. PROPOSED MULTI-OBJECTIVE CRICKET CHIRPING ALGORITHM (MOCCA)

This section describes the design and implementation of MOCCA which uses the Pareto approach for extending the CCA to solve MOO problems.

The main differences noticed in CCA for MOO Problems are as follows:

- Instead of choosing female cricket from the population, allow the male cricket to find/search female cricket. Their chirping rate will increase as the temperature increases and based on their chirping rate female cricket gets attracted.
- When the cricket chirps for aggression, it fights with other male crickets and the winner is chosen based on six aggression levels[30]:

1. *Mutual Avoidance:* In this level no aggressive interaction takes place. The winner is decided mutually.
2. *Pre-Established Dominance:* In this level one cricket attacks and the other retreats.
3. *Antenna fencing:* In antenna fencing crickets lashes with their antenna. It is an enthusiastically inexpensive signal that carries mostly motivational information about resource value.
4. *Mandible spreading (Unilateral):* One cricket shows broadly spread mandibles, which indicate that it is superior to the other.
5. *Mandible Spreading (bilateral):* In this level, both crickets display their spread jawbones. Mandible spreading indicates the strength of the cricket.
6. *Wrestling:* In this level a thoroughgoing fight where the crickets may repeatedly disengage, combat and bite other body parts and reengage mandibles to show their strength.

The fight can be settled at any of the levels (1)-(6) by an opponent. The looser retreating, upon which the established winner typically produces the rivalry song together with characteristic body tremulous (jerking).

Based on these behaviours of cricket the fitness calculation of two crickets is implemented. In Multi-Objective Cricket Chirping Algorithm (MOCCA) an external repository is used to store the non-dominated solutions (Pareto Front).

The pseudocode is shown in TABLE I and the detailed stepwise procedure is given in the next section.

Table I: Algorithm for multi-objective cricket chirping algorithm with pareto based

Algorithm for Multi-Objective Cricket Chirping algorithm (MOCCA)
1. Inputs: *N*: Number of cricket population, *M_gen*: Maximum number of generation, T: environment temperature, P_{agg}: the probability of aggression, *nrep*: size of the repository, *ngrid*: number of grids.
2. Problem definition:*d*: dimension of the cricket in search space, *x*: the position of the cricket, *M*: number of objectives, *ub*: upper bound, *lb*: lower bound.
3. Initialization:initialize cricket population in the search space randomly; $x(i,d) = lb_{(1,d)} + (ub_{(1,d)} - lb_{(1,d)}) \cdot rand$
4. Initialize the repository : *rep=[ ]*
5. Evaluate the objective function: $f_{(i,m)}$
6. Calculate the fitness: $fitness_{(i)} \leftarrow calculate\_fitness(x_{(i,d)}, f_{(i,m)})$
7. While stopping criteria is not met:
a. Allow the cricket for mating chirp:
i. $female \leftarrow Search\_female(x_{(i,d)})$
ii. $offsprings \leftarrow mating(female, x_{(i,d)})$
iii. *rep←best(offsprings, parents) //Choose the best one among parents and offspring*
b. Allow for aggression chirp on aggression rate A_r *winner←fight() //Warn the other male cricket for fighting*
End while
8. Return the optimal solution

V. GENERAL FRAMEWORK OF MOCCA

A. Input

N: Size of the cricket population,
M_gen: Maximum number of generations,
T: Environment's Temperature,
P_{agg}: the probability of aggression,
nrep: size of the repository,
ngrid: number of grids.

B. Output

rep: An external repository '*rep*' is used to store Pareto front

C. Initialization.

Initialize the cricket population in the search space randomly

$$x(i,d) = lb(1,d)+(ub(1,d)-lb(1,d))\cdot rand \quad (1)$$

Where d is the dimension of the decision variables and $i=1,2,\ldots N$, lb is the lower bound and $ub$is the upper bound of the variable in the search space.

Initialize the repository *'rep'* which stores the non-dominated crickets in the initial *x*.

D. Evaluate the Objective Function.

Calculate the objective function value of each objective for all the crickets. For cricket *i*, the value of each objective $f(i, m)$, where $m = 1, 2, \ldots, M$, is calculated.

E. Fitness Calculation.

This procedure involves calculating the strength of each cricket. The fitness calculation of the crickets is shown in TABLE II.

Published By:
Blue Eyes Intelligence Engineering
& Sciences Publication

F. Mating Phase.

In this phase, the cricket will search for female crickets and mate with them. After successful completion of mating, they produce the offspring.

The mating process is done similar to the crossover process of the genetic algorithm. The mating process produces offspring and selects the best one among the offspring and parents. Then the non-dominated cricket is stored in the repository *'rep'*.

Table II: Algorithm for fitness calculation of cricket

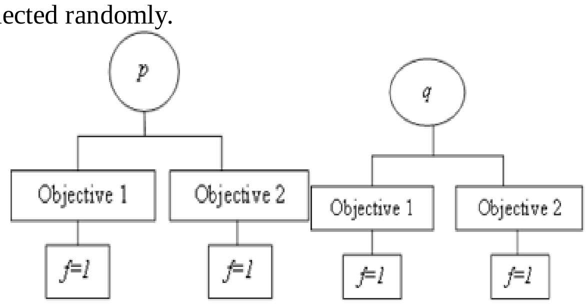
Calculate_fitness()

For each cricket (*i=1 to N*)
For each cricket (*j=i+1 to N*)
For each objective (*k=1 to M*)
If ($x_{(i,k)}<x_{(j,k)}$) and ($x_{(i,k)}!=x_{(j,k}$
greater(i)=greater(i)+1; less(j)= less(j)+1;
else if ($x_{(i,k)}==x_{(j,k)}$)
equal(i)= equal(i)+1; equal(j)= equal(j)+1;
else
greater(j)=greater(j)+1; less(i)= less(i)+1;
end
fit(i)=[greater(i)+ equal(i)+ less(i)];
fit(j)=[greater(j)+ equal(j)+ less(j)];
if (fit(i)>= fit(j))
strength(i)= strength(i)+1;
End
End
End

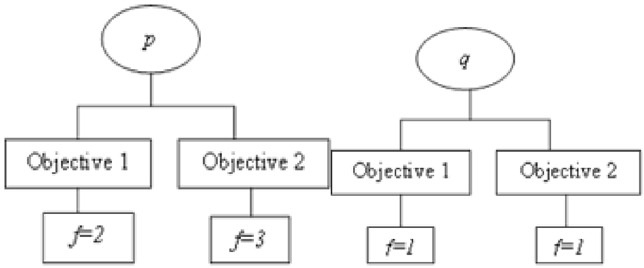
G. Aggression Phase.

The male cricket gets into the aggression phase with probability rate A_r. The two crickets fight with each other and the winner is selected based on the six aggression levels. The levels are described as follows:

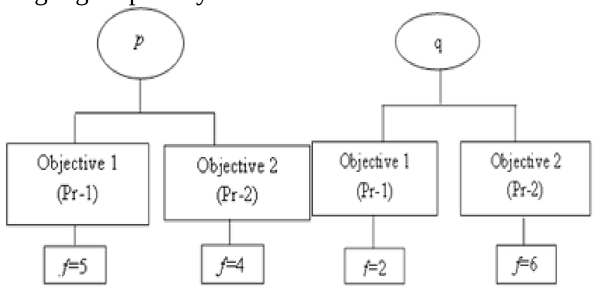
1) Level 1-Mutual Avoidance. When the values of each objective of a solution p are equal to the corresponding values of each objective of the solution q, then anyone solution (cricket individual) will be selected as the winner. For example, p and q have two objectives and the fitness (f) of both solutions is equal. In this case, any solution p or q is selected randomly.



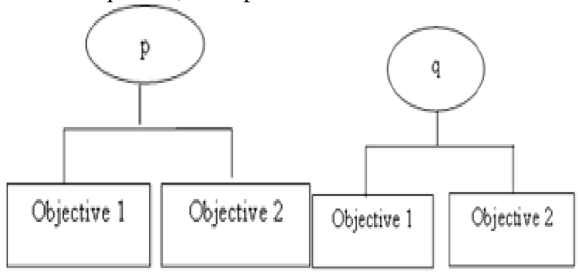
2) Level 2-Antennal Fencing. When the values of each objective of a solution p are greater than the corresponding values of each objective of the solution q then p wins. For example, in the following figure all the objectives of p are greater than q, so p will win.



3) Level 3: Pre-Established Dominance. For this phase fix a priority (P_r) for each objective and based on priority and objective value choose the winner. For example, the priority is assigned high in objective 2. Here higher value is having high priority. In this case, the solution q will win since q is having the highest value of objective 2 that is having higher priority.

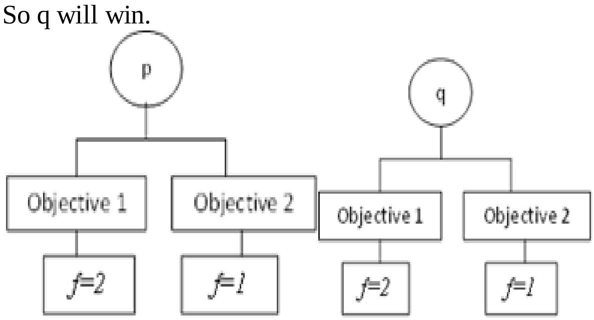


4) Level 4: Mandible Spreading (Unilateral). In this phase, one solution p that satisfies the constraints, whereas another one q is not, then p will win.



Satisfying Constraints Not satisfying Constraints

5) Level 5: Mandible Spreading (Bilateral). In this case, both the solutions are satisfying constraints, but a number of constraints satisfying solution will win. Check the number of constraints satisfied by each solution. For example, from the figure number of constraints satisfied by q is more than p. So q will win.



No. of Satisfying Constraints=2No. of Satisfying Constraints=3

6) Level 6: Wrestling. At this level, the cricket will fight with each other but not exploited in this research.

Published By:
Blue Eyes Intelligence Engineering
& Sciences Publication

H. Retaining of Non-Dominated Solution

An external repository or archive is used to store the records of non-dominated solutions. It consists of an archive controller and an adaptive grid. The function of addition or deletion of a solution to the archive is controlled by the archive controller. The main purpose of the adaptive grid is to produce well-distributed Pareto fronts. The reason for choosing an adaptive grid is its computational cost that is easy and lower than niching.

VI. EXPERIMENTAL RESULTS AND ANALYSIS

This section first describes the standard benchmark test functions that are considered for experiments.

The different performance metrics used for performance measurement and the implementation results of MOCCA are described in the following sections.

A. Multi-Objective Test Functions

There are a large number of standard test functions that are available for MOO problems.

To validate the proposed MOCCA, a subset of a few widely used functions is selected that is convex, non-convex and discontinuous. The test functions without constraints are given in TABLE III and the test problems with constraints are given in TABLE IV.

Table III: Test problem without constraint

Problem	N	Variable Bounds	Objective Functions	Characteristics of Pareto Front
ZDT1	30	[0,1]	$f_1(x)=x_1$ $f_2(x)=g(x)[1-\sqrt{x_1/g(x)}]$ $g(x)=1+9((\sum_{(i=2)}^{n} x_i))/((n-1))$	Convex
ZDT2	30	[0,1]	$f_1(x)=x_1$ $f_2(x)=g(x)[1-x_1/g(x)]^2$, $g(x)=1+9(\sum_{(i=2)}^{n} x_i)/(n-1)$	nonconvex
ZDT3	30	[0,1]	$f_1(x)=x_1$, $f_2(x)=g(x)\left[1-\sqrt{x_1/g(x)} - \frac{x_1}{g(x)} Sin(10\pi x_1)\right.$ $g(x)=1+9\left(\sum_{i=2}^{n} x_i\right)/(n-1$	Noncontiguous Convex
ZDT4	10	[-5,5]	$f_1(x)=x_1$, $f_2(x)=g(x)\left[1-\sqrt{x_1/g(x)}\right]$ $g(x)=1+10(n-1)+\sum_{i=2}^{n}[x$	Continuous Non-convex
ZDT6	10	[0,1]	$f_1(x)=1-\exp(-4x_1)sin^6(6$, $f_2(x)=g(x)[1-(f_1(x)/g(x)$ $g(x)=1+9\left[\left(\sum_{i=2}^{n} x_i\right)/(n-\right.$	Non-convex
SCH	1	$[-10^3,10^3]$	$f_1(x)=x^2$, $f_2(x)=(x-2)^2$	Connected Convex

Table IV: Test Problem with Constraints

Problem	N	Variable Bounds	Objective Functions	Characteristics of Pareto Front
TNK	2	$x_i \in [0, \pi]$ $i = [1, 2]$	$f_1(x) = x_1$, $f_2(x) = x_2$ $g_1(x) = x_2 + x_2^2 - 1 - 0.1cos(16 \arctan(x_1/x_2)) \geq 0$ $g_2(x) = (x_1 - 0.5)^2 + (x_2 - 0.5)^2 \leq 0.5$	Discrete
BNH	2	$x_1 \in [0, 5]$ $x_2 \in [0, 3]$	$f_1(x) = 4x_1^2 + 4x_2^2$, $f_2(x) = (x_1 - 5)^2 + (x_2 - 5)^2$ $g_1(x) = (x_1 - 5)^2 + x_2^2 \leq 25$ $g_2(x) = (x_1 - 8)^2 + (x_2 + 3)^2 \geq 7.7$	Continuous convex
OSY	6	$x_1 \in [0,10]$ $x_2 \in [0, 10]$ $x_3 \in [1, 5]$ $x_4 \in [0, 6]$ $x_5 \in [1, 5]$ $x_6 \in [0, 10]$	$f_1(x) = -[25(x_1-2)^2+(x_2-2)^2+(x_3-1)^2+(x_4-4)^2+(x_5-1)^2]$ $f_2(x) = x_1^2+x_2^2 + x_3^2 + x_4^2 + x_5^2 + x_6^2$ $g_1(x) = x_1 + x_2 - 2 \geq 0$, $g_2(x) = 6 - x_1 - x_2 \geq 0$ $g_3(x) = 2 - x_2 + x_1 \geq 0$, $g_4(x) = 2 - x_1 + 3x_2 \geq 0$ $g_5(x) = 4 - (x_3 - 3)^2 - x_6 \geq 0$, $g6(x) = (x_5 - 3)^2 + x_6 - 4 \geq 0$	Continuous non-convex
CONSTR	2	$x_1 \in [0.1, 1]$ $x_2 \in [0, 10]$	$f_1(x)=x_1$, $f_2(x)=(1-x_2)/x_1$ $g_1(x)=9x_1+x_2 \geq 6$ $g_2(x)=9x_1-x_2 \geq 1$	

B. Performance Metrics

Any algorithm is validated by using a set of performance metrics. The metrics used for validating SOO problems may not correctly evaluate the performance of MOO problem. Hence there exists a separate set of performance metrics exclusively designed for validating MOO problems. The important ones that are used to evaluate the performance of the proposed MOCCA are given below.

1) Generational Distance (GD). The most commonly used performance metric is Generational distance. It is measured as the extent to which the actual Pareto Front and the obtained Pareto Front are distant from each other. It is mathematically computed as shown in (2).

$$GD = \frac{1}{n}\sqrt{\sum_{i=1}^{n} dist_i^2} \quad (1)$$

In equation (2), n indicates the cardinality of solutions in the generated Pareto-Front. $dist_i$ signifies the Euclidean distance between solution *i* in the actual Pareto front and its closest neighbor in the generated Pareto front. The lesser is the value of *GD*, the better will be the convergence.

Published By:
Blue Eyes Intelligence Engineering
& Sciences Publication

2) Spacing (SP). The primary intention of spacing metrics is to determine the extent to which the solutions are equally spread along the generated Pareto front.

It can be mathematically defined as in equation 3:

$$SP = \sqrt{\frac{1}{n}\sum}, \qquad \bar{d} \ (2)$$

In equation (3), n and $dist_i$ hold similar meaning as that of (2). When this metric holds a small value, it signifies a more uniform spread of the solutions.

3) Maximum Spread (MS). The maximum spread (MS) metric exhibits the extent to which the actual Pareto Front encloses the generated Pareto-optimal front. This is identified in accordance with the hyper boxes constructed by the optimal function values from the actual Pareto optimal front and the generated Pareto-optimal front. It is mathematically expressed as in (4).

$$MS = \left[\frac{1}{m}\sum_{i=1}^{m}\left[\frac{min\left(f_i^{max},F_i^{max}\right)-min\left(f_i^{min},F_i^{min}\right)}{F_i^{max}-F_i^{min}}\right]^2\right] \qquad (3)$$

In Equation (4), m indicates the number of objectives. f_i^{max} and f_i^{min} signify the respective maximum and minimum of associated i^{th} objective in the generated Pareto-front, respectively, and F_i^{max} and F_i^{min} are the maximum and minimum of the i^{th} objective in the true Pareto-front. When maximum spread metric characterizes a larger value, it depicts that the spread of the solutions is better.

C. Parameter Settings

Generally, meta-heuristics approaches need parameter settings for better performance. The different parameters used in MOCCA, MOPSO, SPEA-II, and NSGA-II are given in TABLE V.

Table V: Parameters Used In MOCCA and The Other Algorithms

PARAMETERS	MOPSO	SPEA-II	NSGA-II	MOCCA
Population Size	100	100	100	100
External Archive Size	100			100
No. of adaptive grid	7			7
Inertia weight	0.4			
c_1 and c_2	[0,1]	-	-	-
Aggression rate	-	-	-	0.50

D. Experimental Results

To illustrate the validity of the proposed MOCCA, a number of experiments are conducted over test functions for both multi-objectives optimizations with constraint and without constraints. Fig.1-Fig.6 shows the non-dominated Pareto front produce by MOCCA for MOO without constraints and Fig.7- Fig.10 show the Pareto front produce by MOCCA for MOO with constraints.

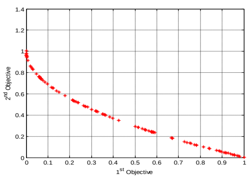


Fig. 1: Pareto front for ZDT1

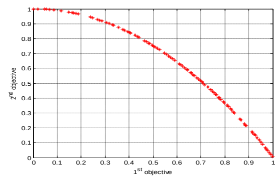


Fig. 2: Pareto front for ZDT2

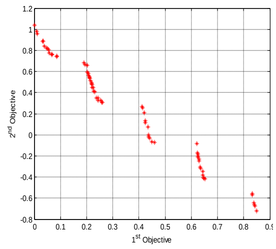


Fig. 3: Pareto front for ZDT3

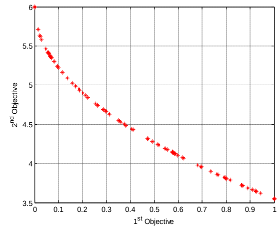


Fig. 4: Pareto front for ZDT4

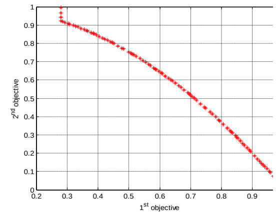


Fig. 5: Pareto front for ZDT6

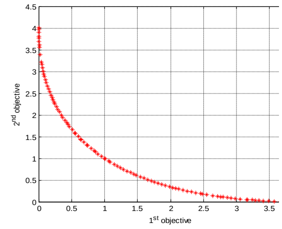


Fig. 6: Pareto front for SCH

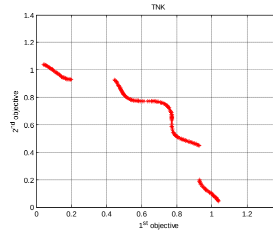


Fig. 7: Pareto front for TNK

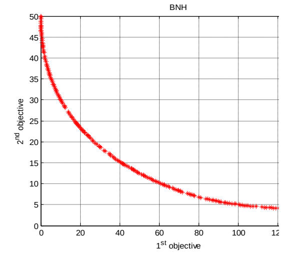


Fig. 8: Pareto front for BNH

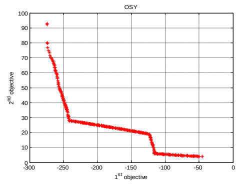


Fig. 7: Pareto front for OSY

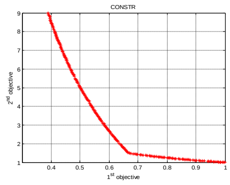


Fig. 8: Pareto front forCONSTR

VII. COMPARISON AND STATISTICAL ANALYSIS

A. Comparison with Other Algorithms

Having obtained the Pareto solutions for the various benchmark functions using MOCCA, it has to be compared with another such algorithm for its performance efficiency. Hence MOCCA is compared with a set of standard MOO algorithms such as SPEA-II, NSGA-II, and MOPSO. The parameter values used in all MOO methods are given in TABLE V and three performance metrics are considered for evaluating the experimental results and these are briefly explained in Section VI (B).

Each algorithm is executed 50 times for every method and the statistical results of the performance metrics SP, MS and GD are reported in TABLE VI, TABLE VIIand TABLE VIIIrespectively.

From the statistical analysis, it is shown that the MOCCA performs better compared to its counterpart considering the performance metrics GD, SP, and MS.

Table VI: Comparison of MOCCA with Other Algorithms Regarding the mean of SP

Problem	MOPSO	SPEA-II	NSGA-II	MOCCA
ZDT1	0.312	0.02671	0.03128	**0.0011**
ZDT2	0.3167	0.1067	0.0187	**0.00406**
ZDT3	0.03299	0.0129	0.00856	**0.00299**
ZDT4	0.5776	0.03109	0.0189	**0.0058**
ZDT6	0.29	0.04899	0.01589	**0.00369**

Table VII: Comparison of MOCCA with other Algorithms Regarding The Mean of Ms

Problem	MOPSO	SPEA2	NSGA-II	MOCCA
ZDT1	0.9982	0.89986	0.9992	**1**
ZDT2	0.9862	0.8906	**1**	**1**
ZDT3	0.88927	0.98902	0.99018	**1**
ZDT4	0.9852	0.93058	0.9988	**0.99998**
ZDT6	0.8485	0.98905	**1**	**1**

Table VIII: Comparison of MOCCA with Other Algorithms Regarding the mean of GD

Problem	MOPSO	SPEA2	NSGA-II	MOCCA
ZDT1	0.0645	0.01809	0.01925	**0.001**
ZDT2	0.0523	0.1523	0.0053	**0.0013**
ZDT3	0.07853	0.03553	0.00607	**0.00238**
ZDT4	0.2609	0.23284	0.20244	**0.03001**
ZDT6	0.0513	0.0765	0.04394	**0.0017**

B. Statistical Analysis

To test the significance of the results produced by MOCCA, a statistical analysis using ANOVA has been carried out on the basis of SP, MS, and GD produced by various algorithms that are shown in TABLEVI, VII and VIII respectively. The analysis is done by considering a 5% significance level over the performance metrics produced by the algorithms corresponding to the test problems for the three different methods i.e. MOPSO, SPEA-II, NSGA-II and MOCCA.

1) ***ANALYSIS I***. In this analysis the hypothesis is set as follows:

- **Null hypothesis H0.** There is no significant difference in the SP among the methods MOPSO, SPEA-II, NSGA-II, and MOCCA.
- **Alternative hypothesis H1.** There is a significant difference in the SP among the methods MOPSO, SPEA-II, NSGA-II, and MOCCA.

The one way ANOVA Test is done and the results found in the experiment are shown in TABLE IX. Here, p values (Sig.= 0.000) is less than 0.05 that strongly oppose the null hypothesis. So it is concluded that there is a significant difference of the SP among the methods MOPSO, SPEA-II, NSGA-II, and MOCCA.

Table IX: Anova Test Over the Methods MOCCA with MOPSO, SPEA2, NSGA2 based ON SP

ANOVA
SP

	Sum of Squares	df	Mean Square	F	Sig.
Between Groups	.306	3	.102	10.558	.000
Within Groups	.154	16	.010		
Total	.460	19			

2) ***ANALYSIS II***. In this analysis the hypothesis is set as follows:

- **Null hypothesis H0:** There is no significant difference in the MS among the methods MOPSO, SPEA-II, NSGA-II, and MOCCA.
- **Alternative hypothesis H1:** There is a significant difference in the MS among the methods MOPSO, SPEA-II, NSGA-II, and MOCCA.

Table X: Anova Test over the Methods MOCCA with Mopso, SPEA-II, NSGA-II, based ON MS

ANOVA
MS

	Sum of Squares	df	Mean Square	F	Sig.
Between Groups	.017	3	.006	3.283	.048
Within Groups	.028	16	.002		
Total	.044	19			

The one way ANOVA test is performed and the results found in the experiment are shown in the TABLE X. From the ANOVA test, it is found that the p values (Sig. =0.048) are less than the significance value 0.05. So it is concluded that there is a significant difference in the MS among the methods MOPSO, SPEA-II, NSGA-II, and MOCCA.

3) ***ANALYSIS III***.In this analysis the hypothesis is set as follows:

- **Null hypothesis H0:** There is no significant difference in the GD among the methods MOPSO, SPEA-II, NSGA-II, and MOCCA.
- **Alternative hypothesis H1:** There is a significant difference in the GD among the methods MOPSO, SPEA-II, NSGA-II, and MOCCA.

Table XI: Anova Test Over the Methods MOCCA with MOPSO, SPEA-II, NSGA-II BASED ON GD

ANOVA
GD

	Sum of Squares	df	Mean Square	F	Sig.
Between Groups	.031	3	.010	1.782	.191
Within Groups	.093	16	.006		
Total	.124	19			

Published By:
Blue Eyes Intelligence Engineering
& Sciences Publication

The one way ANOVA test is done and the results found in the experiment are shown in the TABLE XI. From the results, it is shown that p values (Sig. =0.191) are greater than 0.05 (5% significance level). Therefore, the null hypothesis H0 is accepted. So it is concluded that there is no significant difference of the GD among the methods MOPSO, SPEA-II, NSGA-II, and MOCCA.

VIII. CONCLUSION AND FUTURE WORK

In this paper, the CCA is extended for solving MOO problems called MOCCA. The MOCCA is differing from the basic CCA in two terms. First, the male cricket is allowed to search the female cricket in the search space and secondly, when the male cricket chirps for aggression the winner is selected depends on the seven aggression levels. A different fitness calculation method is also developed and an external archive is used to retain the non-dominated solutions. The MOCCA is implemented and experimented with some of the standard benchmark test problems with constraint and without constraints and compared with three popular techniques i.e. MOPSO, SPEA-II and NSGA-II. The experiment result shows better results compared to its counterparts in terms of generational distance, spacing and maximum spread. The performance of the methods is statistically analyzed by using one way ANOVA test based on the SP, MS, and GD. Though the MOCCA shows better result corresponding to GD, there is no significant difference among the methods over the performance metrics GD, but shows significant difference among the methods over the performance metrics SP and MS. As a future direction MOCCA could be used or apply for different real life problems.

REFERENCES

1. T. P. Bagchi, “Multiobjective Optimization,” pp. 136–146, 1999
2. S. Huband, S. Huband, P. Hingston, and L. Barone, “A Review of Multi-objective Test Problems and a Scalable Test Problem Toolkit A Review of Multiobjective Test Problems and a Scalable Test Problem Toolkit,” vol. 10, no. 2006, pp. 477–506, 2011.
3. F. Dey and A. Caflisch, “Fragment-based de novo ligand design by multi-objective evolutionary optimization. Supporting Information,” J. Chem. Inf. Model., vol. 48, no. 3, pp. 679–690, 2008.
4. H. Mori and T. Yoshida, “An efficient multi-objective memetic algorithm for uncertainties in distribution network expansion planning,” 2009 IEEE Power Energy Soc. Gen. Meet. PES ’09, no. 2, pp. 1–6, 2009.
5. V. Ebrahimipour and M. Sheikhalishahi, “Application of multi-objective particle swarm optimization to solve a fuzzy multi-objective reliability redundancy allocation problem,” 2011 IEEE Int. Syst. Conf. SysCon 2011 - Proc., vol. 1, no. 2, pp. 326–333, 2011.
6. R. K. Jena, “Artificial Bee Colony Algorithm based Multi-Objective Node Placement for Wireless Sensor Network,” Int. J. Inf. Technol. Comput. Sci., vol. 6, no. 6, pp. 25–32, 2014.
7. Hojjati, Ali, Mohsen Monadi, AlirezaFaridhosseini, and MiraliMohammadi. "Application and comparison of NSGA-II and MOPSO in multi-objective optimization of water resources systems." Journal of Hydrology and Hydromechanics 66, no. 3: 323-329(2018).
8. Jiao, Ruwang, et al. "Dynamic constrained multi-objective evolutionary algorithms with a novel selection strategy for constrained optimization." Proceedings of the Genetic and Evolutionary Computation Conference Companion. ACM, 2018.
9. Deuri, Jonti, and S. Siva Sathya. "Cricket chirping algorithm: an efficient meta-heuristic for numerical function optimisation." International Journal of Computational Science and Engineering 16.2, 162-172,2018.
10. M. Shukla, “Review of Multi-objective Optimization using Genetic Algorithm and Particle Swarm Optimization,” pp. 72–74.
11. L. M. P. M. Srinivas, “Genetic Algorithm: A Survey,” Evol. Comput. IEEE, 1994.
12. D. Goldberg, “Genetic Algorithms in Search, Optimization, and Machine Learning,” vol. 9, 2005.
13. K. Deb, S. Agrawal, S. Pratab, and T. Meyarivan, “A fast elitist non- dominated sorting genetic algorithm for multi-objective optimization: NSGA-II.,” Parallel Probl. Solving from Nat. VI Conf., 2000.
14. J. Horn, N. Nafpliotis, and D. E. Goldberg, “A Niched Pareto Genetic Algorithm for Multiobjective Optimization,” pp. 82–87, 1994.
15. J. Horn and N. Nafpliotis, “Multiobjective optimization using the niched pareto genetic algorithm,” Cec, vol. 1, pp. 82–87, 1994.
16. C. M. Fonseca and P. J. Fleming, “MULTI-OBJECTIVE GENETIC ALGORITHMS pp. 2–6.
17. J. Knowles and D. Corne, “The Pareto archived evolution strategy: A new baseline algorithm for Pareto multi-objective optimisation,” Proc. 1999 Congr. Evol. Comput. CEC 1999, vol. 1, pp. 98–105, 1999.
18. E. Zitzler and L. Thiele, “Multiobjective evolutionary algorithms: a comparative case study and the strength Pareto approach,” Evol. Comput. IEEE Trans., vol. 3, no. 4, pp. 257–271, 1999.
19. E. Zitzler, M. Laumanns, and L. Thiele, “SPEA2 : Improving the Strength Pareto Evolutionary Algorithm,” pp. 1–21, 2001.
20. Q. Zhang, S. Member, and H. Li, “MOEA / D : A Multiobjective Evolutionary Algorithm Based on Decomposition,” vol. 11, no. 6, pp. 712–731, 2007.
21. J. Moore and R. Chapman, “Application of particle swarm to multiobjective optimization,” Dep. Comput. Sci. Softw. Eng. Dep. Auburn Univ., pp. 1–4, 1999.
22. C. A. CoelloCoello and M. S. Lechuga, “MOPSO: A proposal for multiple objective particle swarm optimization,” Proc. 2002 Congr. Evol. Comput. CEC 2002, vol. 2, pp. 1051–1056, 2002.
23. X. Hu and R. Eberhart, “Multiobjective optimization using dynamic neighborhood particle swarm optimization,” Congress on Evolutionary Computation, vol. 2. pp. 1677–1681, 2002.
24. X. H. X. Hu, R. C. Eberhart, and Y. S. Y. Shi, “Particle swarm with extended memory for multiobjective optimization,” Proceedings of the 2003 IEEE Swarm Intelligence Symposium. SIS’03 (Cat. No.03EX706). 2003.
25. C. a C. Coello, G. T. Pulido, and M. S. Lechuga, “Handling multiple objectives with particle swarm optimization,” Evol. Comput. IEEE Trans., vol. 8, no. 3, pp. 256–279, 2004.
26. D.H. Wolpert, W.G. Macready, No free lunch theorems for optimization, IEEE Transactions on Evolutionary Computation 1 (1997) 67–82
27. J. Deuri and S. S. Sathya, “A Novel Cricket Chirping Algorithm for Engineering Optimization Problem,” vol. 9, no. 6, pp. 397–402, 2015.

28. Deuri, Jonti, and S. Siva Sathya. "Impact of Parameter Tuning on the Cricket Chirping Algorithm." International Journal of Intelligent Systems and Applications 9.9 (2017): 58.
29. Deuri, Jonti, and S. Siva Sathya. "Multilevel Thresholding for Image Segmentation Using Cricket Chirping Algorithm." Bio-Inspired Computing for Image and Video Processing. Chapman and Hall/CRC, 2018. 31-58.
30. K. Lorenz and N. Tinbergen, "Behavior of the House Cricket, Achetadomesticus," no. 1, 2010.