I.J. Intelligent Systems and Applications, **2017, 9, 58-68**
Published Online September 2017 in MECS (http://www.mecs-press.org/)
DOI: 10.5815/ijisa.2017.09.07

Impact of Parameter Tuning on the Cricket Chirping Algorithm

Jonti Deuri
Department of Computer Science, Pondicherry University, Kalapet-605014, India
E-mail: jontideuri@gmail.com

S. Siva Sathya
Department of Computer Science, Pondicherry University, Kalapet-605014, India
E-mail: ssivasathya@gmail.com

Received: 17 May 2017; Accepted: 12 June 2017; Published: 08 September 2017

***Abstract*—**Most of the man-made technologies are nature-inspired including the popular heuristics or meta-heuristics techniques that have been used to solve complex computational optimization problems. In most of the meta-heuristics algorithms, adjusting the parameters has important significance to obtain the best performance of the algorithm. Cricket Chirping Algorithm (CCA) is a nature inspired meta-heuristic algorithm that has been designed by mimicking the chirping behavior of the cricket (insect) for solving optimization problems. CCA employs a set of parameters for its smooth functioning. In a meta-heuristic algorithm, controlling the values of various parameters is one of the most important issues of research. While solving the problem, the parameter values have a potential to improve the efficiency of the algorithm. The different parameters used in CCA are tuned for better performance of the algorithm through experiments conducted on a set of sample benchmark test functions and then, the fine-tuned CCA is compared with some other meta-heuristic algorithms. The results show the optimal choice of the various parameters to solve optimization problems using CCA.

***Index Terms*—**Metaheuristic Algorithm, Parameter Tuning, Optimization Problem, Cricket Chirping Algorithm, Test Function, Nature-inspired Algorithm.

I. Introduction

Nowadays meta-heuristics algorithms are among the most powerful algorithms to solve complex optimization problems. When these heuristic or meta-heuristics search algorithms are used to solve a particular problem, we need some steps to map the original problem context with the problem-solving framework like the specification of the representation and evaluation of the fitness function. In most of the metaheuristics algorithms, parameter tuning by hand is a common practice. Generally, one parameter is tuned at a time and repeated for simultaneous tuning of more parameters. However, it leads to a huge amount of experiments and may cause some sub-optimal choices. Obtaining an optimal or near optimal solution of the algorithm depends on the parameter values. Therefore, finding good parameter values is important, even if the process requires a lot of additional resources. Determining the best parameter values is a hard and challenging task since there is not much knowledge about the effects of parameters on the algorithm's performance [1]. The procedure of parameter setting can be categorized into two types: parameter tuning (before the run) and parameter control (during the run). In parameter tuning, parameter values are defined in advance and do not change during the execution of the algorithm. In parameter control, parameter values are changed during the algorithm run and can be deterministic, adaptive or self-adaptive. There has been a large amount of work dedicated to finding the optimal parameters of Meta-heuristics Algorithms [2, 3]. The Genetic Algorithm (GA) is the oldest meta-heuristic algorithm developed by Holland [4], which is inspired by Darwin's theory of evaluation and has been employed in solving various types of the optimization problem. De Jong put a significant effort into finding parameter values for a traditional GA and recommended values for the probabilities of single-point crossover and bit mutation, which were good for a number of numeric test problems [5, 6]. Yuan and Gallagher [7] proposed an approach by combining the Meta-EA [8] with a method called Racing [9], which is based on the statistical analysis of algorithm's performance with different parameter settings [10]. The Particle Swarm Optimization (PSO) is another one most popular meta-heuristics algorithm which was developed by [11] based on the social behavior and movement of organisms in a bird flock or fish school. For enhancing its performance developed an automatic parameter tuning technique was developed by G. S. Tewolde et. al.[12]. Many other nature-inspired algorithms such as Ant Colony Optimization (ACO), Simulated Annealing (SA), Harmony Search (HS), Big Bang Big Crunch algorithm (BBBC), Artificial Bee Colony (ABC), Gravitational Search Algorithm (GSA), Cuckoo Search (CS) algorithm, Firefly Algorithm (FA), Bat Algorithm (BA) etc. are some of the most popular optimal meta-heuristic algorithms for solving optimization problem [13-23]. Different authors have fine-tuned the parameters for different problems such as ABC was tuned for energy efficiency optimization in massive MIMO systems, Gaussian noise elimination on digital images [24-26], GA for real world transportation problem, Energy-Minimizing Vehicle Routing Problem, fire tube boiler[27-29], Shuffled Frog Leaping Algorithm Applied to Optimizing Water Distribution Networks [30, 31], Self-Tuning PID(Proportional Integral Derivative) for PMSM (Permanent Magnet Synchronous Motor) Vector Control based on improved KMTOA (kinetic-molecular theory optimization algorithm) [32], PID controller parameter tuning for Superheated Steam Temperature of Power Station Boiler [33], improved PSO tuned PID and Sliding Mode (SMC) classical controllers for the motion control problem of the robotic manipulator [34] etc.

Cricket Chirping Algorithm (CCA) is a new meta-heuristic algorithm developed based on the chirping behavior of cricket [35, 36]. Adjustment of different parameters of the meta-heuristics algorithm is usually a time-consuming task which is mostly done by a trial and error approach. In this study, the performance of CCA tests with different values of the parameters like temperature, aggression rate, crossover rate and female selection. The rest of the paper is organized as follows: Section II shows the details of the cricket chirping algorithm. In Section III the impact of different parameters is analyzed in some benchmark test problems and the comparison of experimental results with its counterpart are discussed in Section IV and the article is concluded with future directions in Section V followed by the references.

II. Cricket Chirping Algorithm

In this section, the details of CCA are explained. First, the natural behavior of cricket is described briefly, followed by the development of CCA, its ability to solve the optimization problem and the movement of the cricket in the search space.

A. Cricket's Behavior in Nature

Crickets are insects that make a sound in summer night that is called as chirping which is scientifically referred to as stridulation. Their body structure and long antennae including jumping hind legs are similar to a grasshopper. Acoustic communication among animal utilizes sound to signal information from one individual to another. Many animals use acoustic signals for intraspecific (within species) communication. For example, birds produce a song, frogs croak, crickets chirp etc. In crickets, generally male cricket produces the chirp by rubbing their forewings (elytra) against each other. The song or chirp of cricket is categorized into different types based on their chirping behavior [37- 39].

Calling song: The calling song is produced to attract female cricket for mating which is fairly loud and this is the chirping that is most commonly heard during summer nights.

Courtship song: The courtship song is like a scraping noise of low intensity that is made when a male attempts to mate with a female.

Copulatory song: After a successful mating they produce a copulatory song for a brief period.

Aggressive song: This song is also called rivalry or triumph song. An aggressive song detects the near presence of male cricket.

Though the cricket chirping is divided into different types, mostly crickets chirp for mainly two reasons: i) for mating and ii) for aggression. They produce Calling Chirp for mating with female crickets and Aggressive Chirp to fight with other male crickets. Fig. 1 shows the behavior of Mating chirp and Aggressive chirp. When the cricket chirps for mating, it produces offspring and when the cricket chirps for aggression, it fights with other male cricket and the winner cricket will survive in the environment.

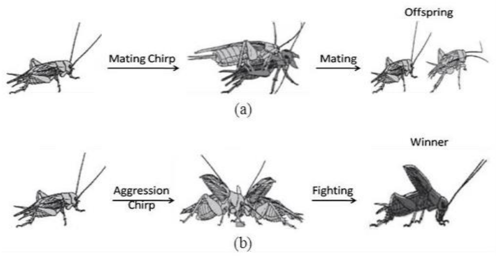


Fig.1. Crickets behavior: (a) Calling Chirp and (b) Aggressive Chirp

B. Cricket Chirping Algorithm

Each cricket is assumed to be a solution and is characterized by its position in the search space. Out of the total cricket population, few of them as determined by the user are randomly 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. When they chirp for mating, based on their chirping rate at certain temperature the cricket moves to a new position and mate with females and produce offspring. The offspring represents a new position (solution) of the cricket. By emitting an aggressive song, they fight with other male crickets and the winning cricket will survive 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 crickets produce calling chirps for mating, they emit a peculiar sound and the female crickets are attracted and they move towards female cricket. They are allowed to mate and after mating, they produce offspring, which means they are taken to new positions in the search space. The mating process is similar to the crossover process of the genetic algorithm.

Second, when the cricket chirps for aggression, they emit an aggressive chirp and other male crickets are warned and female crickets will move away. All crickets may not be chirping for aggression. For simplicity, we can use a simple representation that the probability of chirping for aggression is A_r, which is in between [0, 1]. When 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.

The fitness of the male cricket is calculated based on their attractiveness and replace the position of low fit cricket with high fit cricket. The aim is to use the new and potentially better solutions (cricket) to replace a not so good solution. The attraction is based on the loudness of the chirping sound. The chirping sound is calculated based on the environment temperature and the cricket moves to the new position.

C. Movement of Cricket

For the cricket movements, the rules to update the positions x_i and velocities v_i in a d-dimensional search space have to be defined. An American physicist and naturalist, Amos Emerson Dolbear developed a relationship between air temperature and the chirping rate of cricket [40]. The chirping of crickets is related to temperature, as well as age and mating success. Dolbear expressed the relationship using (1) or (2) which provides a way to estimate the temperature T_c in degrees Celsius and degrees Fahrenheit T_f from the number of chirps per unit time N_c:

$$Tc = 10 + \frac{Nc - 40}{7} \tag{1}$$

Or,

$$T_f = 50 + \frac{Nc - 40}{4} \tag{2}$$

The chirping rate is derived by using Dolbear's law in a certain temperature *Tc*,

$$Nc = (Tc - 10) * 7 + 40 \tag{3}$$

Since the chirping rate is the number of chirps per unit time, it is assumed to be the frequency of the cricket's chirp. From the frequency calculate the velocity of the cricket as follows:

$$v_i = Nc * \lambda \tag{4}$$

Here, λ is the wavelength, the gap between one chirp to another chirp which is uniformly drawn. From the velocity, we calculate the step size (*st*) by using (5).

$$st_i = v_i + (x_i - x^*) * \alpha \tag{5}$$

Where α=0.01 is a constant value which is used to control the movements of the cricket within a bounded space and x_i is the current position and x^* is the best position ever encountered by the cricket. Then the cricket will move to the new position by using the following formula:

$$x_{i+1} = x_i + st_i \tag{6}$$

Equations (1)-(6) are used when the cricket chirps for mating and they change their step size according to the chirping rate at a certain temperature. The mating process is similar to the crossover operator of the genetic algorithm.

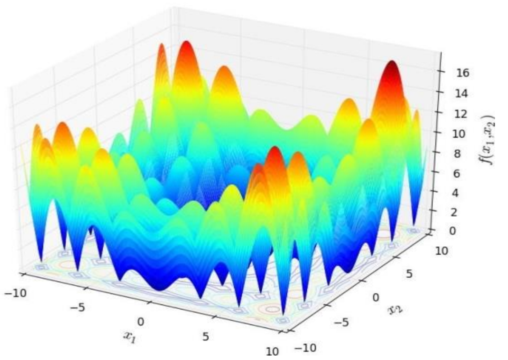
Table 1. Algorithm of CCA

Algorithm for CCA( )
Begin
Input: *fc (x): Objective function; n: Total Number of crickets;* T_c *: Temperature;* A_r*: Aggression Rate;* C_r*: Crossover Rate; k: No. of female crickets* F_s.
1. Randomly Initialize the crickets position x_i ;
2. Calculate the fitness of the crickets;
3. Choose k crickets as female crickets F_s ;
4. While (stopping criteria not met)
a. Chirp for mating //Call procedure calling-chirp();
b. Mate with female crickets // call procedure mating();
c. Allow the male crickets to chirp for aggression //Call procedure aggr-chirp()
d. Compute the fitness of the crickets;
e. Rank the crickets and find the current best;
5. End while
6. Return the global best cricket at termination
End
Procedure calling-chirp( )
Begin
For every male cricket
• Calculate the N_c and v_i using (3) and (4);
• Calculate the st_i using (5);
• Move each cricket to the new position using (6)
• Return crickets in the new position;
End For
End
Procedure mating ( )
Begin
• Randomly choose a cut point in both male cricket M_i and female cricket F_s ;
• Exchange the genetic materials of both M_i and F_s with reference to their cut points to produce two new offspring; // Similar to crossover in GA//
• Return the best of the two offspring and the parents as the new cricket (position);
End
Procedure aggr-chirp()
Begin
if rand> A_r //rand is a random number within [0 1]
• Choose a random cricket. //Conduct a tournament between them.
• Fight with other male crickets.
• Return the winner cricket.
End
End

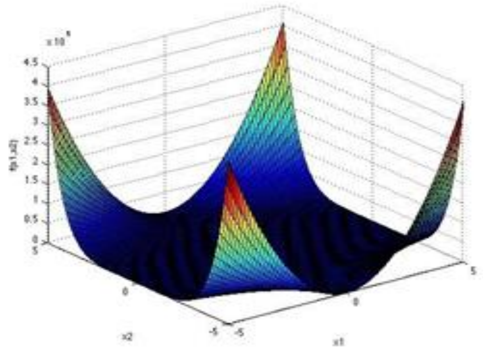
In the aggressive Chirp phase, the male cricket is allowed for aggression with aggression rate A_r, it is assumed that when the cricket chirps for aggression, they move to a new position using a random walk to another male cricket and make combat. The fittest cricket will be the winner and survive in the search space. Table 1 shows the pseudo code of the Cricket Chirping Algorithm.

III. IMPACT OF VARIOUS PARAMETERS OF CCA

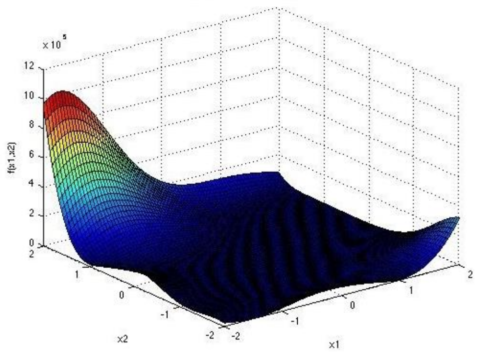
Generally, meta-heuristic algorithms have several parameters to be fine-tuned. CCA has parameters like Temperature *(T_c)*, Aggression Rate *(A_r)*, Crossover Rate *(C_r)* and Female Selection *(F_s)*. This section analyses the impact of these parameters on the performance of the CCA. The values of each parameter are varied by keeping other parameters fixed. To fix the parameters for CCA the performance of different parameter values are studied on the benchmark mathematical function, namely Alpine, Beale, Goldstein and Price, Rastrigin, Sphere, and Tripod function. Brief descriptions of these functions are given below and the 3D view for a two-dimensional graph representation of each function is shown in Fig.2.



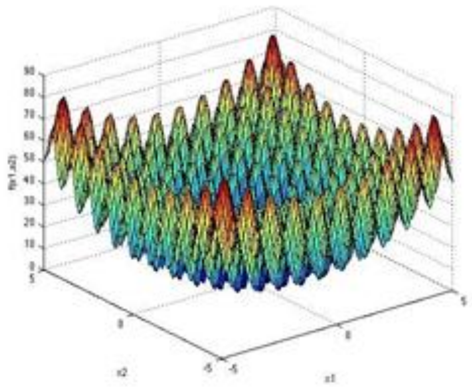
(a) Alpine 1 Function



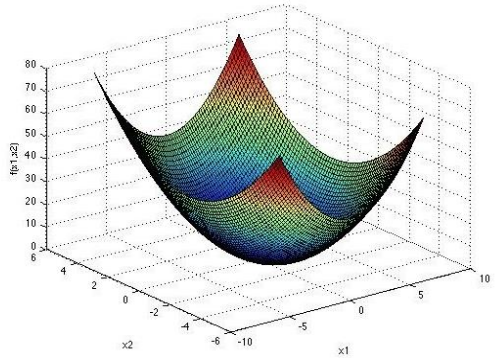
(b) Beale Function



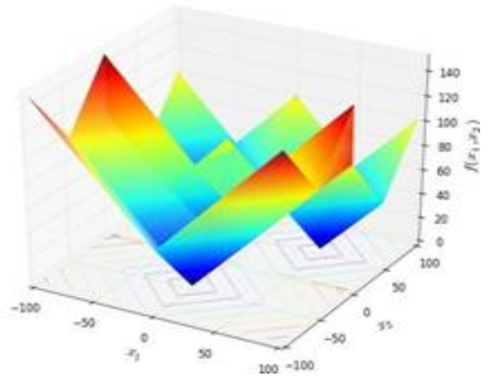
(c) Goldstein & price Function



(d) Rastrigin Function



(e) Sphere Function



(f) Tripod Function

Fig.2. Two-dimensional graph representation of the test functions

A. Test Functions

Alpine1Functions: This is a multimodal minimization problem defined as follows:

$$f(x)=\sum_{i=1}^{d}\left|x_i \sin(x_i)+0.1x_i\right| \tag{7}$$

Here, *d* represents the number of dimensions and x_i *[−10, 10]* for *i = 1, ., d.* The Global optimum is $f_{min}=0$ for $x_i=0$, *i = 1, ., n.*

Beale Function (Hedar, N.D.): It is a continuous, differentiable, non-separable, non-scalable, unimodal function. The function is defined as follows:

$$f(x)=(1.5-x_1+x_1x_2)^2+(225-x_1+x_1x_2^2)^2+(2625-x_1+x_1x_2^3)^2 \tag{8}$$

Goldstein & Price Function: The function Goldstein & Price returns the value:

$$f(x)=\left[1+\left(x_0+x_1+1\right)^2\left(19-14x_0+3x_0^2-14x_1+6x_0x_1+3x_1^2\right)\right]\times \left[30+\left(2x_0-3x_1\right)^2\left(18-32x_0+12x_0^2+48x_1-36x_0x_1+27x_1^2\right)\right] \quad (9)$$

With domain $-2\ |x_i| \leq 2$ and the global minimum $f_{min} = 3$ at the point (0,−1).

Rastrigin function: The Rastrigin function has several local minima. It is highly multimodal, but the locations of the minima are regularly distributed. It is shown in the figure in its two-dimensional form. The function is given below:

$$f(x)=10n+\sum_{i=1}^{d} x_i^2-10\cos(2\pi x_i) \quad (10)$$

The range is $-5.12 \leq x_i \leq 5.12$ and global minimum $f_{min}=0$ at the point (0,.......,0)

Sphere Function: It is a continuous, convex and unimodal function. This function has d local minima except for the global one.

$$f(x)=\sum_{i=1}^{d} x_i^2 \quad (11)$$

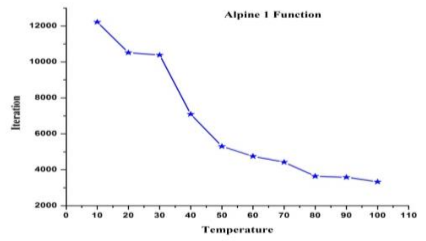
Where, $x_i \in [-5.12,\ 5.12]$ for all $i = 1, .\ \ , d$. The Global Minimum $f(x*) = 0$, at $x* = (0, ..., 0)$.

Tripod Function: It is a semi-continuous problem. The global minimum is $f_{min} = 0$ on (0, -50). It is theoretically easy, this problem is, in fact, difficult for a lot of algorithms that are trapped in the two local minima. Here, d represents the number of dimensions and $x_i \in [-100,100]$ for i=1,..,.., d.

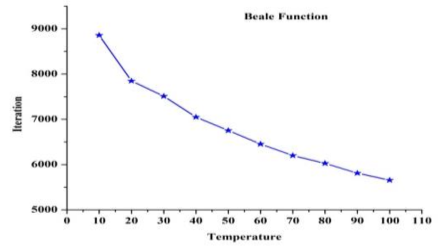
$$f(x)=\begin{bmatrix} p(x_2)*(1+p(x_1)) \\ abs\left(x_1+50*p(x_2)*\left(1-2*p(x_1)\right)\right) \\ +abs\left(x_2+50*\left(1-2*p(x_2)\right)\right) \end{bmatrix} \quad (12)$$

B. Impact of Temperature (T_c)

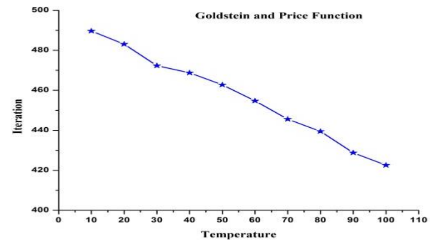
The cricket's chirp depends on the outside temperature. Generally, the higher the temperature of environment the higher the chirping rate. Here the cricket is allowed to chirp in different temperatures. The temperature is taken as 10, 20, 30, 40, 50, 60, 70, 80, 90 and 100. The program is run for 100 times for every temperature value to find the global optimal value and the average number of iterations is calculated. In Fig.3(a)-(f) the effect of the different temperature values is shown.



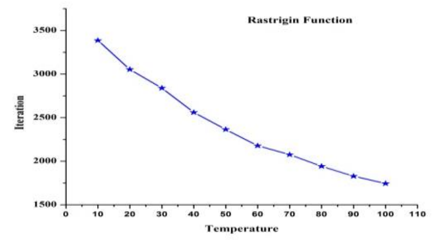
(a)



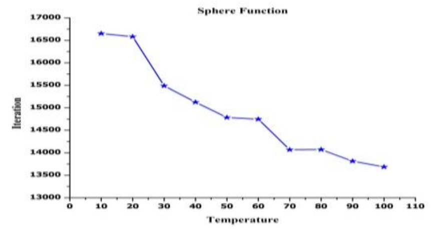
(b)



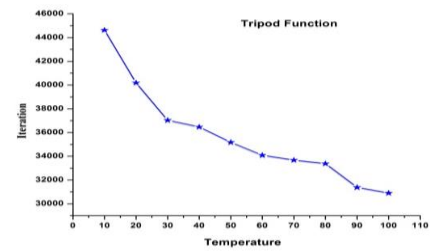
(c)



(d)



(e)



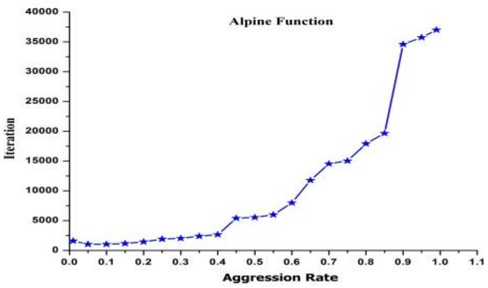
(f)

Fig.3. The number of iterations to find the global optimal value of the test Functions in different Temperature

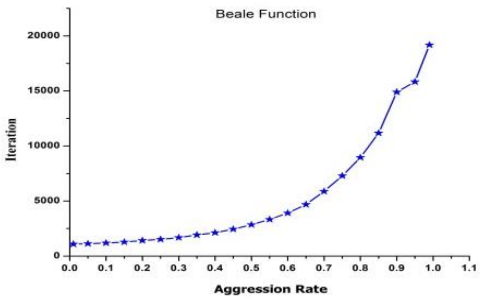
From the graph shown in Fig. 3, it is visible that when the temperature is increased the number of iterations needed to find the optimal value is reduced for all functions. So we fixed the temperature between 90 to 100.

C. Impact Of Aggression Rate (A_r)

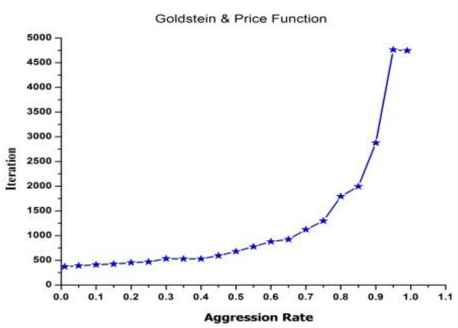
When the cricket wants to fight, it makes the aggressive chirp. Since not all the crickets chirp for aggression, it is needed to choose the aggression rate *(A_r)*. Having analyzed, the performance of CCA is better at a higher temperature. This experiment analyzed the impact of different aggression rate A_r at the temperature (T_c=100). The probability of aggression rate is varied like 0.01, 0.05, 0.10, 0.15, 0.20, 0.25, 0.30, 0.35, 0.40, 0.45, 0.50, 0.55, 0.60, 0.65, 0.70, 0.75, 0.80, 0.85, 0.90, 0.95 and 0.99. The program is run 100 times for all test functions with every A_r value and the average iterations to find the global minima are calculated. Fig. 4(a)-(f) shows how the aggression rate affects the results in different functions.



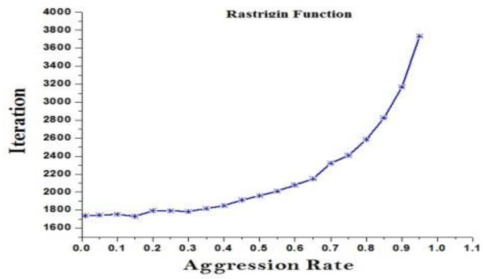
(a)



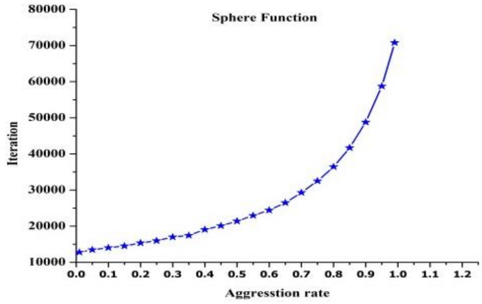
(b)



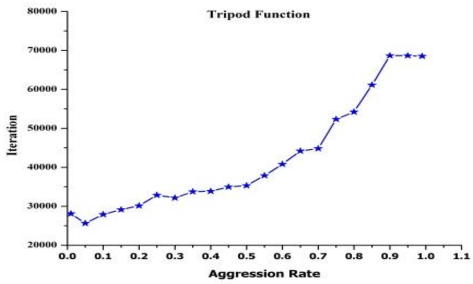
(c)



(d)



(e)



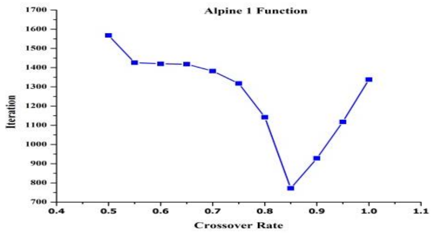
(f)

Fig.4. The number of iterations to find the global optimal value of the test Functions in different Aggression rate

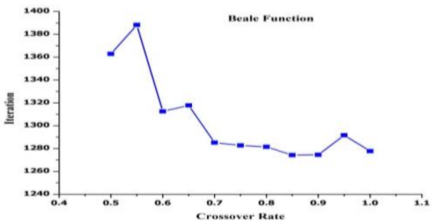
From the graph, it is shown that in the lower value of aggression rate, the number of iterations needed for optimization is less. CCA is showing better results at aggression rate 0.05, 0.10, and 0.15. Based on the results, [0.05 to 0.25] is considered as the aggression rate for low dimensional problems.

D. Impact of Crossover Rate (C_r)

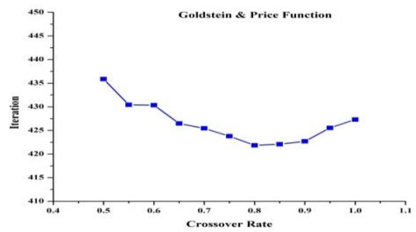
After making the calling song, the female and male crickets undergo the mating process. The crossover is done using different crossover rates *(C_r)*. Generally, crossover rate is high in GA. So the program is tested for crossover rate 0.50, 0.55, 0.60, 0.65, 0.70, 0.75, 0.80, 0.85, 0.90, 0.95 and 0.99 having temperature rate 100 and aggression rate 0.15. The performances of different crossover rate in each test functions are shown in Fig.5 (a)-(f). The process is same as in Section III (B) and Section III(C).



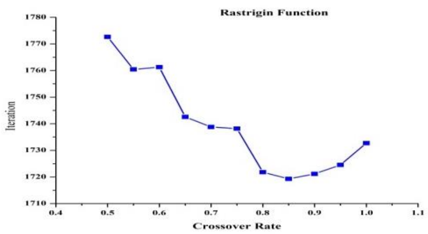
(a)



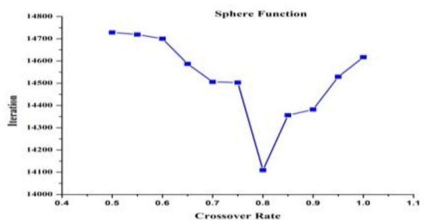
(b)



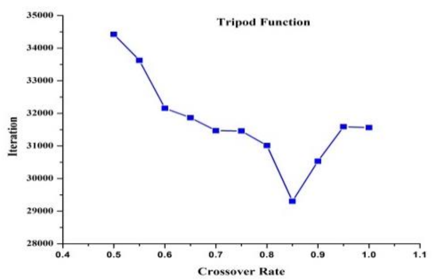
(c)



(d)



(e)



(f)

Fig.5. The number of iterations to find the global optimal value of the test Functions in different Crossover Rate

From the figure, it is shown that the higher crossover rate from 0.75 to 0.85 is showing good performance. So the crossover rate *(C_r)* would be fixed in 0.80.

E. Female Selection (F_s)

In CCA, only the male cricket chirps for mating. To perform the mating operation the male crickets have to choose the female crickets. The selection of female (F_s) crickets may be done in different ways. The CCA is tested for the following female cricket selection methods.

Random Female Selection: In this method, 50% of the total crickets are randomly chosen as female crickets and allowed to mate with male crickets randomly.

Best Fit selection: In the best fit selection process the highest fit cricket is selected as female cricket and makes the crossover process in two ways. First, the female cricket is allowed to mate with all the male crickets and second, allow mating only with one male cricket which is randomly chosen.

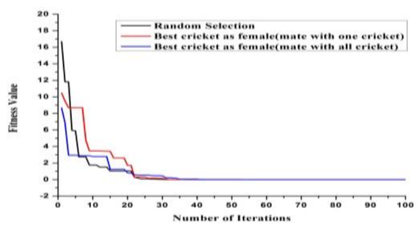
Lowest fit Selection: In the lowest fit process the worst cricket is selected as female cricket and allowed to mate. Since this selection does not converge to the optimal fitness value, it is not considered for female selection.

The program is run for 100 iterations for the Random Selection method and Best Fit Selection method (both mating with one cricket and all cricket). The fitness value of selection methods for all the test functions is run 100 iterations and the graph of convergence to the optimal solution is shown in Fig.6(a)-(f). It is clearly observed from the graph that the Best Fit Selection scheme mating with all male crickets, offers better results in terms of convergence or speed.

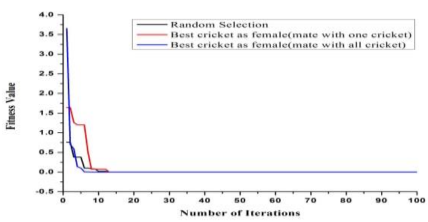
From the experiment conducted, the best CCA parameter found are listed in Table 2. Using these CCA parameter values, the CCA Algorithm is compared with other meta-heuristic Algorithm in the next section.

Table 2. Parameter values of CCA

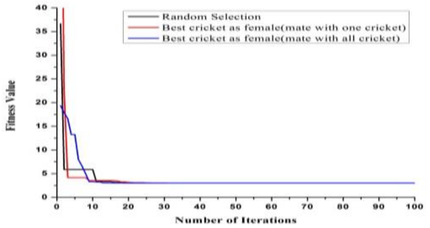
Parameter Name	Value
Temperature	100
Aggression rate	0.15
Crossover rate	0.80
Female Selection	Best cricket as female and mate with all cricket



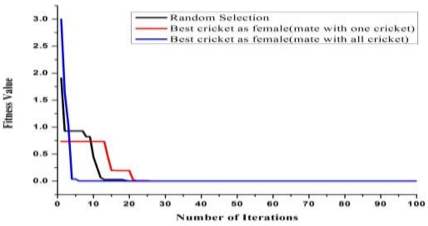
(a) Alpine 1 Function



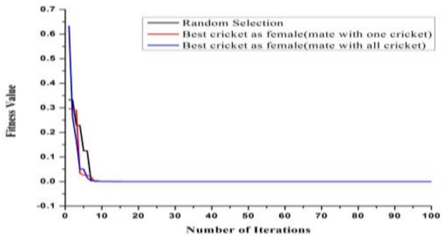
(b) Beale Function



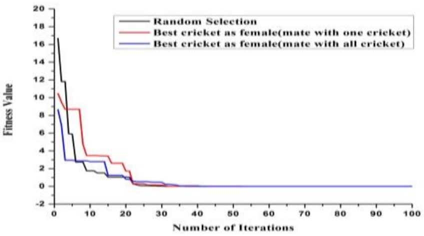
(c) Goldstein & Price Function



(d) Rastrigin Function



(e) Sphere Function



(f) Tripod Function

Fig.6. The fitness value of test Functions with different female selection methods

IV. Comparison of CCA with other Meta-Heuristic Algorithms

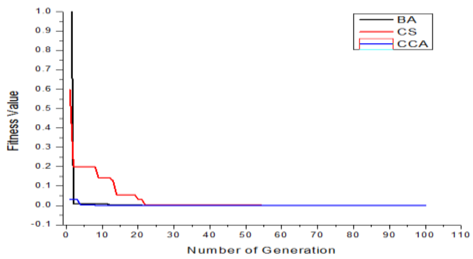
After fine-tuning the optimal parameter values for CCA, the algorithm is compared with the popular meta-heuristic algorithms to show the betterment of CCA. Bat Algorithm (BA) and Cuckoo Search (CS) Algorithm are considered in comparison with CCA as they are recent and popular algorithms. GA and PSO have not been compared here since the BA and CS show better results compared to them [21, 22]. The BA was developed based on the echolocation behavior of bats. The bat flies in search of its prey with a velocity v_i at position (solution) x_i with varying frequency. It has loudness A_i and pulses emission rate r_i. In BA, the balance between exploration and exploitation is controlled by tuning algorithm-dependent parameters and uses a frequency-tuning technique to control the dynamic behavior of a swarm of bats. For BA the loudness and pulse are set to 0.5 and the minimum and maximum frequency is taken as 0 and 2.

The CS algorithm was developed based on the breeding behavior of cuckoo in combination with the Levy flight behavior of some birds. Some species of Cuckoo birds lay their eggs in communal nests. If a host bird discovers the eggs are not their own, they will either throw these alien eggs away or simply abandon its nest and build a new nest elsewhere. CS has three idealized rules. First, each cuckoo lays one egg at a time, and dump its egg in the randomly chosen nest. Second, the best nests with the high quality of eggs will carry over to the next generations. And Third, the number of available host nests is fixed and the egg laid by a cuckoo is discovered by the host bird with probability $p_a \in [0,1]$. In the CS algorithm, the discovery rate of the alien egg is taken as 0.25.

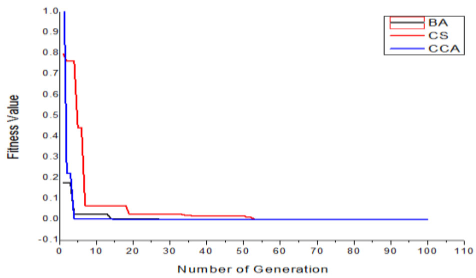
The CCA, BA, and CS are run for every test function using the above-mentioned parameter values. The convergence toward the optimal value of each algorithm for all functions in 100 generations are shown in Fig.7 (a)-(f). The fitness value of each algorithm for the benchmark test functions in 100 generations is shown in Table 3. The CCA shows better fitness values compared to the BA and CS.

Table 3. Comparison of fitness value of CCA with BA and CS in 100 generations

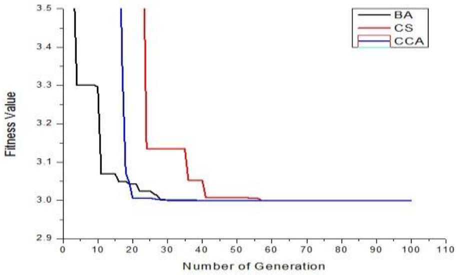
Function Name(f(x))	*Fitness Value (f_{min})*		
	BA	**CS**	**CCA**
Alpine 1	1.17E-08	2.02E-08	**2.52E-21**
Beale	1.63E-08	5.66E-07	**8.93E-17**
Goldstein	3	3	3
Rastrigin	2.37E-06	0.000322	**7.11E-15**
Sphere	7.17E-06	4.67E-09	**7.03E-23**
Tripod	0.000146	0.008835	**3.87E-11**



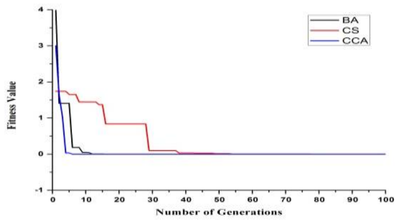
(a) Alpine Function



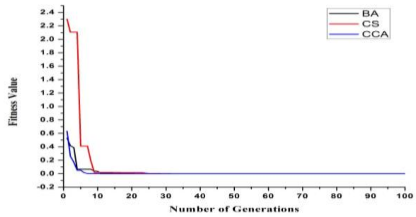
(b) Beale Function



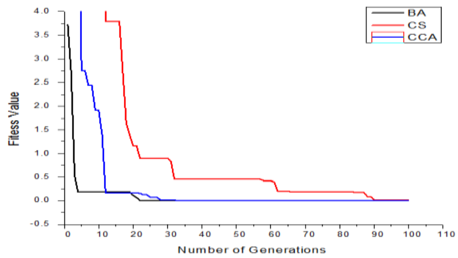
(c) Goldstein & price Function



(d) Rastrigin Function



(e) Sphere Function



(f) Tripod Function

Fig.7. Fitness value of BA, CS, and CCA for each test function

V. CONCLUSIONS

This paper analyzes the impact of the various parameters used in CCA. The parameters *viz.*, environmental Temperature T_c, Aggression Rate A_r, Crossover Rate C_r and Female Selection F_s have an effective contribution in the performance of CCA. When comparing the initial and final set of parameters, it is found that the final set provides better results than the initial parameter configuration for the problem under study. As per the analysis of the experiments, the higher the temperature, the higher the fitness value of the crickets. The cricket produces high-frequency sound in high temperatures. But, in low aggression rate, it shows better performance for low dimension problems. In female selection, the best fit female selection converges faster compared to other female selection schemes. The values obtained through various experimental settings could be fixed as the standard parameters for the CCA algorithm in future. The comparison with its counterparts also shows that CCA performs better than others and hence it could be used as a good optimization technique. In future, CCA can be extended to solve multi-objective optimization problem and it can be applied in different areas.

ACKNOWLEDGMENT

This work is supported by the University Grants Commission (UGC) under the Rajiv Gandhi National Fellowship (RGNF) scheme, India.

REFERENCES

[1] Nannen, Volker, Selmar Smit, and Agoston Eiben. "Costs and benefits of tuning parameters of evolutionary algorithms." *Parallel Problem Solving from Nature–PPSN X* (2008): 528-538. doi: 10.1007/978-3-540-87700-4_53

[2] Fallahi, M., S. Amiri, and M. Yaghini. "A parameter tuning methodology for metaheuristics based on design of experiments." *International Journal of Engineering and Technology Sciences* 2, no. 6 (2014): 497-521.

[3] Eiben, Agoston E., Zbigniew Michalewicz, Marc Schoenauer, and James E. Smith. "Parameter control in evolutionary algorithms." In *Parameter setting in evolutionary algorithms*, pp. 19-46. Springer Berlin Heidelberg, 2007. doi 10.1007/978-3-540-69432-8_2.

[4] Holland, John H. *Adaptation in natural and artificial systems: an introductory analysis with applications to biology, control, and artificial intelligence*. MIT press, 1992.

[5] DeJong, K. A. "Analysis of the Behavior of a Class of Genetic Adaptive Systems. Dept." *Computer and Communication Sciences, University of Michigan, Ann Arbor* (1975).

[6] Grefenstette, John J. "Optimization of control parameters for genetic algorithms." *IEEE Transactions on systems, man, and cybernetics* 16, no. 1 (1986): 122-128. doi: 10.1109/TSMC.1986.289288

[7] Yuan, Bo, and Marcus Gallagher. "A hybrid approach to parameter tuning in genetic algorithms." In *Evolutionary Computation, 2005. The 2005 IEEE Congress on*, vol. 2, pp.1096-1103. IEEE, 2005. doi: 10.1109/CEC.2005. 1554813

[8] Back, Thomas. *Evolutionary algorithms in theory and practice: evolution strategies, evolutionary programming, genetic algorithms*. Oxford university press, 1996.

[9] Maron, Oded, and Andrew W. Moore. "The racing algorithm: Model selection for lazy learners." In *Lazy learning*, pp. 193-225. Springer Netherlands, 1997. doi 10.1007/978-94-017-2053-3_8.

[10] Birattari, Mauro, Thomas Stützle, Luis Paquete, and Klaus Varrentrapp. "A racing algorithm for configuring metaheuristics." In *Proceedings of the 4th Annual Conference on Genetic and Evolutionary Computation*, pp. 11-18. Morgan Kaufmann Publishers Inc., 2002.

[11] Poli, Riccardo, James Kennedy, and Tim Blackwell. "Particle swarm optimization." *Swarm intelligence* 1, no. 1 (2007): 33-57.doi: 10.1007/s11721-007-0002-0

[12] Tewolde, Girma S., Darrin M. Hanna, and Richard E. Haskell. "Enhancing performance of PSO with automatic parameter tuning technique." In *Swarm Intelligence Symposium, 2009. SIS'09. IEEE*, pp. 67-73. IEEE, 2009. doi: 10.1109/SIS.2009.4937846

[13] Dorigo, Marco, Vittorio Maniezzo, and Alberto Colorni.

"Ant system: optimization by a colony of cooperating agents." *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)* 26, no. 1 (1996): 29-41. doi: 10.1109/3477.484436

[14] Wong, Kuan Yew. "Parameter tuning for ant colony optimization: a review." In *Computer and Communication Engineering, 2008. ICCCE 2008. International Conference on*, pp. 542-545. IEEE, 2008.

[15] Kirkpatrick, Scott, C. Daniel Gelatt, and Mario P. Vecchi. "Optimization by simulated annealing." *science* 220, no. 4598 (1983): 671-680.

[16] Geem, Zong Woo, Joong Hoon Kim, and G. V. Loganathan. "A new heuristic optimization algorithm: harmony search." *Simulation* 76, no. 2 (2001): 60-68.

[17] Erol, Osman K., and Ibrahim Eksin. "A new optimization method: big bang–big crunch." *Advances in Engineering Software* 37, no. 2 (2006): 106-111.

[18] Karaboga, Dervis, and Bahriye Basturk. "A powerful and efficient algorithm for numerical function optimization: artificial bee colony (ABC) algorithm." *Journal of global optimization* 39, no. 3 (2007): 459-471. doi: 10.1007/s10898-007-9149-x

[19] Karaboga, Dervis, and Bahriye Basturk. "On the performance of artificial bee colony (ABC) algorithm." *Applied soft computing* 8, no. 1 (2008): 687-697. doi.org/10.1016/j.asoc.2007.05.007

[20] Rashedi, Esmat, Hossein Nezamabadi-Pour, and Saeid Saryazdi. "GSA: a gravitational search algorithm." *Information sciences* 179, no. 13 (2009): 2232-2248. doi:10.1016/j. ins.2009.03.004

[21] Yang, Xin-She, and Suash Deb. "Cuckoo search via Lévy flights." In *Nature & Biologically Inspired Computing, 2009. NaBIC 2009. World Congress on*, pp. 210-214. IEEE, 2009. doi: 10.1109/NABIC.2009.5393690

[22] Yang, Xin-She. "A new metaheuristic bat-inspired algorithm." *Nature inspired cooperative strategies for optimization (NICSO 2010)* (2010): 65-74. doi 10.1007/978-3-642-12538-6_6

[23] Yang, Xin-She. "Firefly algorithms for multimodal optimization." In *International symposium on stochastic algorithms*, pp. 169-178. Springer Berlin Heidelberg, 2009. doi: 10.1007/978-3-642-04944-6_14

[24] Bouchibane, F. Z., and M. Bensebti. "Parameter tuning of Artificial Bee Colony algorithm for energy efficiency optimization in massive MIMO systems." *Detection Systems Architectures and Technologies (DAT), Seminar on*. IEEE, 2017 doi: 10.1109/DAT.2017.7889188 .

[25] Kockanat, Serdar, and Nurhan Karaboga. "Parameter tuning of artificial bee colony algorithm for gaussian noise elimination on digital images." In *Innovations in Intelligent Systems and Applications (INISTA), 2013 IEEE International Symposium on*, pp. 1-4. IEEE, 2013.

[26] Akay, Bahriye, and Dervis Karaboga. "Parameter tuning for the artificial bee colony algorithm." In *International Conference on Computational Collective Intelligence*, pp. 608-619. Springer Berlin Heidelberg, 2009.

[27] Chebbi, Olfa, and Jouhaina Chaouachi. "Effective parameter tuning for genetic algorithm to solve a real world transportation problem." *In Methods and Models in Automation and Robotics (MMAR), 2015 20th International Conference on, pp. 370-375. IEEE,* 2015. doi: 10.1109/MMAR.2015.7283904.

[28] Cooray, P. L. N. U., and Thashika D. Rupasinghe. "Machine Learning-Based Parameter Tuned Genetic Algorithm for Energy-Minimizing Vehicle Routing Problem." *Journal of Industrial Engineering* 2017 (2017). doi.org/10.1155/2017/3019523

[29] Hassanein, Osama I., Ayman A. Aly, and Ahmed A. Abo-Ismail. "Parameter tuning via genetic algorithm of fuzzy controller for fire tube boiler." *International Journal of Intelligent Systems and Applications* 4, no. 4 (2012): 9.

[30] Mora-Melia, Daniel, Pedro L. Iglesias-Rey, F. Javier Martínez-Solano, and Pedro Muñoz-Velasco. "The Efficiency of Setting Parameters in a Modified Shuffled Frog Leaping Algorithm Applied to Optimizing Water Distribution Networks." *Water* 8, no. 5 (2016): 182. doi: 10.3390/w8050182

[31] Veček, Niki, Marjan Mernik, Bogdan Filipič, and Matej Črepinšek. "Parameter tuning with Chess Rating System (CRS-Tuning) for meta-heuristic algorithms." *Information Sciences* 372 (2016): 446-469.

[32] Yi, Lingzhi, Chengdong Zhang, and Genping Wang. "Research of Self-Tuning PID for PMSM Vector Control based on Improved KMTOA." *International Journal of Intelligent Systems and Applications* 9, no. 3 (2017): 60.

[33] Xiao, Benxian, Jun Xiao, Rongbao Chen, and Yanhong Li. "PID Controller Parameters Tuning Based-on Satisfaction for Superheated Steam Temperature of Power Station Boiler." *International Journal of Information Technology and Computer Science (IJITCS)* 6, no. 7 (2014): 9.

[34] Kapoor, Neha, and Jyoti Ohri. "Improved PSO tuned Classical Controllers (PID and SMC) for Robotic Manipulator." *International Journal of Modern Education and Computer Science* 7, no. 1 (2015): 47.

[35] Deuri, Jonti, and S. Siva Sathya. "A novel cricket chirping algorithm for engineering optimization problem." *Advances in Natural and Applied Sciences* 9, no. 6 SE (2015): 397-403.

[36] Jonti Deuri, S. S. Sathya, Cricket chirping algorithm: an efficient metaheuristic for numerical function optimization, International Journal of Computational Science and Engineering, in press.

[37] Alexander, Richard D. "Aggressiveness, territoriality, and sexual behavior in field crickets (Orthoptera: Gryllidae)." *Behaviour* (1961): 130-223.

[38] Brown, William D., Adam T. Smith, Brian Moskalik, and Josh Gabriel. "Aggressive contests in house crickets: size, motivation and the information content of aggressive songs." *Animal Behaviour* 72, no. 1 (2006): 225-233. doi.org/10.1016/j.anbehav.2006.01.012

[39] Mays, David L. "Mating Behavior of Nemobiine Crickets: Hygronemobius, Nemobius, and Pteronemobius (Orthoptera: Gryllidae)." *Florida Entomologist* (1971): 113-126.DOI: 10.2307/3493557

[40] Dolbear, A. E. "The cricket as a thermometer." *The American Naturalist* 31, no. 371 (1897): 970-971.

Authors' Profiles

Jonti Deuri was born in Lakhimpur District of Assam, India in 1988. She is currently pursuing Ph.D. degree in the Department of Computer Science and Engineering at Pondicherry University, Puducherry, India. She received Master Degree on Computer Application from Dibrugarh University, Assam, India. Her research interests focus on Nature-inspired Algorithm, Swarm Intelligent and Image Processing.

Dr. S. Siva Sathya, is an Associate Professor in the Department of Computer Science and Engineering, Pondicherry University, Puducherry, India. She completed her B.Tech degree in Computer Science and Engineering from Pondicherry Engineering College and M.Tech and Ph.D. degree in Computer Science and Engineering from Pondicherry University. Her research interest includes Evolutionary Algorithms, Bioinformatics, Intrusion Detection and Response, Image Processing and Computer based Music Analysis. She has a number of research papers in International journals and conferences. She also works for women's safety with modern technology by implementing mobile apps for women safety.

How to cite this paper: Jonti Deuri, S. Siva Sathya, "Impact of Parameter Tuning on the Cricket Chirping Algorithm", International Journal of Intelligent Systems and Applications(IJISA), Vol.9, No.9, pp.58-68, 2017. DOI: 10.5815/ijisa.2017.09.07