

Comparison of Advanced Optimization Algorithm for Task Scheduling in Cloud Computing

S. Gunasekaran¹, S. Sonialpriya²

¹Coimbatore Institute of Engineering and Technology, Narasipuram Road, Narasipuram, Tamilnadu, India

²Coimbatore Institute of Engineering and Technology, Narasipuram Road, Narasipuram, Tamilnadu, India

Abstract: In this paper, the algorithmic concepts of the Cuckoo-search (CK), Particle Swarm Optimization (PSO), Differential Evolution (DE), and Artificial Bee Colony (ABC), algorithms have been analyzed. The mathematical optimization problem solving successes of the mentioned algorithms have also been compared statistically by testing more than 20 different benchmark functions. Pragmatic results reveal that the problem solving success of the CK algorithm is very close to the DE algorithm. The run-time complexity and the required function-evaluation number for acquiring global minimiser by the DE algorithm are normally smaller than the relationship algorithms. The performances of the CK and PSO algorithms are statistically closer to the performance of the DE algorithm than the ABC algorithm. The CK and DE algorithms deliver more robust and precise results than the PSO and ABC algorithms.

Keywords: Cuckoo search algorithm, Particle swarm optimization, Differential evaluation algorithm, Artificial bee colony algorithm.

1. Introduction

The standard optimization methods commonly used in scientific applications consist of hessian matrix based methods (Min-Jea et al. 2009) and gradient based methods (Haupt 1995). In practice, it is necessary many of the classical optimization methods to fulfil with the structure of the objective function intended to be solved. However, if the derivative of the objective function cannot be calculated, it gets hard to search the optimal solution via classical optimization means (Rashedi et al. 2009). It can be shown by mathematical methods that a solution obtained by using classical techniques is globally optimum (Nowak and Cirkpa 2004). But however, it is general to use Metaheuristic algorithms in solving non-differentiable nonlinear-objective functions the solution of which is either impossible or extremely tricky by using the classical optimization techniques (Karaboga and Akay 2009a; Nowak and Cirkpa 2004; Clerc and Kennedy 2002; trelea 2003; Dorigo et al. 1996; ong et al 2006; Storn and Price 1997; yang and Deb 2010; Das et al 2011; Yang 2005, 2009; Zhang et al 2007). While a method, which can be used to show that a solution obtained by metaheuristic algorithm is the optimum solution, as not been proposed by the researchers yet, the solution obtained via these method are referred to as sub-optimal solutions (Rashedi et al.2009; ong et al.2006; Storn and Price 1997; Yang and Deb 2010). The Metaheuristic algorithms that are most broadly used in scientific applications are Genetic Algorithm (Deb et al.2002), particle swarm optimization algorithm (PSO) (Clerc and Kennedy 2002; Trelea 2003; Yoshida et al.2000; Juang 2004), Differential Evolution algorithm (DE) (Stron and Price 1997; Ferrante and Ville 2010; Price and Storn 1997; Storn 1997; Liu and Lampinen 2005; Ali and Torn 2004; Shahryar et al.2008; Das and Suganthan 2009; Kaelo and Ali 2006; Swagatam et al.2009;Janez et al. 2007), Artificial bee colony algorithm (Karaboga and Akay 2009a; Karaboga and Basturk 2007a,b; Janez et al.2007; Karaboga 2009; Fei et al. 2009), Cuckoo Search Algorithm (Yang and Deb 2009; Deb et al 2002),

Gravitaional search Algorithm(Rashedi et al. 2009;Esmat et al. 2010,2011;Duman et al. 2010;Chaoshun and Jianzhong 2011)Harmony search Algorithm (Gee metal.2001;Lee and Geem 2004;Mahadevi et al.2007) and their derivatives.

2. Review of the CK, PSO, DE and ABC Algorithm

The common random-walk based system-equation of metaheuristic algorithms is given as follow (Mersha and Dempe 2011):

$$V < -X_i + s\delta x \quad (1)$$

In Eq. δx denotes the step-size and X subtends to a random solution of the related objective function where $x \in R^n$. $s \in R$ value is the scaling factor selected for δx values. To develop the X pattern, the decision rule defined in Eq.2 is used where f is the objective function (storm and price 1997);

$$F(v) < f(X_i) \rightarrow X_i = v \quad (2)$$

Here, $X_i = [x_j / j=1,2,3,\dots,n]$ vector shows the i^{th} pattern of the pattern matrix, $P = [X_i]$, Which includes n dimensional m random solution where $i=1,2,3,\dots,a,b,r_1,r_2,r_3,r_4,r_5,\dots,m$.

The Metaheuristic algorithms change radically in terms of the methods they use to establish s and δx values of Eq. 1 (Karboga and Akay 2009a; Storn and Price 1997; Clerc and Kennedy 2002; Terlea 2003; Dorigo et al. 1996; Das et al.2011). The process to be used to conclude the mentioned s and δx values directly specifies the development strategy of the x_j attributes. The simplest method followed to conclude δx s to use another arbitrarily particular solution. For the situation of $r_1 \neq r_2 \neq r_3$, Let X_{r_1} and X_{r_2} correspond to the accidental solutions of the related goal function. In this case, if $\delta x_{r_1, r_2}$ is defined as,

$$\delta x_{r_1, r_2} = X_{r_1} - X_{r_2} \quad (3)$$

Volume 4 Issue 3, March 2015

www.ijsr.net

Thus, the generalized system-equation of the randomized mutation operator of the DE/rand/I algorithm (Storm and price 199; price an storm 1997) is achieved as,

$$V \leftarrow x_{r3} + s (x_{r1} - x_{r2}) \quad (4)$$

The differences in use of the interconnected system-equations are quite important in comparison of metaheuristic optimization algorithms. The DE algorithm uses a significant effective adaptive-mutation strategy together with the related system-equations (storm and price 1997; Ferrante and Vilie 2010; Price and storm 1997; storm 1999; Das and Suganthan 2009). This case is the most significant variation of the DE algorithm from the standard generic algorithm. Two detailed survey papers on DE based optimization algorithms are given in (Ferrante and Ville 2010; Das and Suganthan 2009)

In the DE/rand/I mutation strategy, s scaling factor is exposed by F. In this case, the generalized system-equation of the DE algorithm can be defined by Eq.5 where $F^* \in \mathbb{R}^n$,

$$V \rightarrow x_{r3} + F^* (x_{r1} - x_{r2}) \quad (5)$$

In the original ABC algorithms, s scaling factor is shown by Φ (Karaboga and Akay 2009a). In this case, the generalized system-equation of the ABC algorithm can be defined by eq. 6 where $\Phi^* \in \mathbb{R}^n$,

$$V \rightarrow x_{r1} + \Phi^* (x_{r1} - x_{r2}) \quad (6)$$

In eqs 5 and 6, * denotes the Hadamard multiplication operator. In lots of engineering problem, the dimension of the problem is given as $n \gg 1$. On condition that $h \in \mathbb{Z}^+$ and $h \in [1, n]$ is a random integer number, if a random solution of any problem is shown by $X = [x_1, x_2, x_3 \dots x_h \dots x_n]$, eqs. 5 and 6 can be generalized to the differential solution.

2.1 Cuckoo search algorithm

The Cuckoo search (CK) algorithm (Yang and Deb 2009, 2010), which is a residents based stochastic global search algorithm, has been proposed by Yung and Deb in (2009). In the CK algorithm, a model corresponds to a nest and relationship each individuals attribute of the model corresponds to a cuckoo-egg. The common system-equation of the CK algorithm is based on the common system-equation of the random-walk algorithms, which is given in Eq.6;

$$X_{g+1,i} = X_{g,i} + \alpha * \text{levy}(\lambda) \quad (7)$$

Where g indicates the number of the existing generation ($g=1, 2, 3, \dots$, maxcycle and maxcycle denotes the encoded maximum generation number). In this algorithm, the first Value of the j^{th} attributes of the i^{th} pattern $P_{g=0,i} = [x_{g=0,j,i}]$, have been determined by using Eq.7,

$$X_{g=0,j,i} = \text{rand.} (up_i - low_i) + low_i \quad (8)$$

Where low_i and up_i are the lower and upper search-space restricted of j^{th} attributes, respectively. The CK algorithm controls the border conditions in each computation steps. Therefore, when the value of an attribute excess the authorized search space limits, then the value of the

connected attribute is efficient with the value of the closer limit value to the related attribute. Before starting to iterative search process, the CK algorithm detects the most winning pattern as x_{best} pattern. The iterative growth phase of the pattern matrix start with the detection step of the Φ by using Eq.8;

$$\Phi = \left(\frac{\tau(1+\beta) \sin(\pi \frac{\beta}{2})}{\tau(\frac{1+\beta}{2}) \beta^{\frac{1-\beta}{2}}} \right)^{1/\beta} \quad (9)$$

2.2 Differential Evolution Algorithm

The Differential evolution (DE) algorithm is a population-based, heuristic evolutionary optimization algorithm residential for the result of real-valued numerical optimization problems (Storn and Price 1997; Price and Storm 1997; Storm 1999; Price et al. 2005). It is a very successful global search algorithm with a quite easy mathematical formation. This algorithm uses the mutation, crossover, and range strategies of the generic algorithm. The mainly important difference of the DE algorithm from the standard genetic algorithm. The most important difference of this algorithm from the standard genetic algorithm is the strong mutation strategies it has (Storn and Price 1997; Price and Storn 1997). The DE algorithm can use two kinds of crossover schemes (i.e., exponential and binomial) (Storn and Price 1997; Ferrante and Ville 2010; Price and Ville 2010; Price and storn1997; Storn 1999; Liu and Lampinen 2005; Shahryar et al. 2008). In the benchmark DE algorithm, five different mutation strategies can be used with one of the two different crossover methods (Shahryar et al.2008; Das and Suganthan 2009; Kaelo and Ali 2006; Swagatam et al.2009; Janez et al.2007; Price et al.2005; Vesterstrom and Thomsen 2004; Bin et al. 2010). Therefore, the benchmark DE algorithm has ten different options to classify the algorithmic structure to be used. The standard mutation strategies implemented in the original DE algorithm (Storn and Price 1997; Price and Storn 1997; Price et al. 2005; Das and Suganthan 2009; Swagatam et al.2009) are given in Eq.9;

DE/rand/1	$v = x_{r1} + F. (x_{r2} - x_{r3})$
DE/rand/2	$v = x_{r1} + F. (x_{r2} - x_{r3}) + F. (x_{r4} - x_{r5})$
DE/best/1	$v = x_{\text{best}} + F. (x_{r1} - x_{r2})$
DE/best/2	$v = x_{\text{best}} + F. (x_{r1} - x_{r2}) + F. (x_{r3} - x_{r4})$
DE/target-to-best/1	$v = x_i + F. (x_{\text{best}} - x_i) + F. (x_{r1} - x_{r2})$

(10)

There are several mutation algorithms developed to be used with the DE algorithm. The most general one used in the literature is the DE/rand/I/bin algorithm which basically population (Storn and Price 1997; Price and Storn 1997; Price et al.2005). The weighted difference of two of the three arbitrarily selected solutions is added to the third solution to obtain the donor solution, Which is then mutated with a 4th arbitrarily selected solution, and if the objective function value of the obtained mutant solution provides a better solution than the objective function value of the selected original 4th solution, the mutant solution is selected instead of the mentioned 4th solution within the next population. The detailed performed test results have exposed that DE/rand/I/bin is excessively susceptible to the size of

population, total number of iterations, crossover value (CR) and the weighting value (F).

2.3 Particle swarm optimization algorithm

The particle swarm optimization (PSO) algorithm is also population-based, stochastic and multiagent parallel global-search technique (Del Valle et al. 2008; Clerc and Kennedy 2002; Trelea 2003; Yoshida et al. 2000; Juang 2004; Sousa et al. 2004). Unlike the genetic algorithm and the DE algorithm, the PSO algorithm has no crossover and mutation operators. The PSO algorithm is based on the arithmetic modelling of different combined behaviours of the living creatures that display difficult social behaviours. In the PSO algorithm, while a pattern (i.e., particle) is budding a new state, both the cognitive component of the relative particle and the social component generated by the swarm are used. This state enables the PSO algorithm to successfully develop the local solutions into global optimum solutions. However, the PSO algorithm is significantly affected by the initial values of the parameters used in the Weighting of the cognitive and social components and the weighting strategy of the velocity vector. The Lbest and Gbest topologies are the mostly used topology in the standard PSO algorithm. In this paper a efficient implementation of the PSO algorithm With Lbest topology, which is known as PSO-2007, has been used (Bin et al. 2010)

The success of the PSO algorithm in finding the global optimum depends particularly on the initial values of the control parameters (c_1 , c_2 , ω) of the PSO algorithm, the size of swarm value, and the maximum iteration number. In the tests complete in this paper, control parameter values of the PSO algorithm have been selected as the same with the values given in (Karaboga and Akay 2009a); $c_{1\text{initial}}=1.80$, $c_{2\text{initial}}=1.80$, and $\omega = 1 + \frac{\text{rand}}{2}$ as in (Eberhart and Shi 2001).

2.4 Artificial bee Colony Algorithm

The Artificial Bee Colony (ABC) algorithm is also population-based numeric optimization algorithm. It is based on the simplified numerical models of the food searching behaviours of the bee-swarms. In this algorithm, any arbitrary solution of the problem corresponds to a source of nectar. Here one employed bee assigned to each nectar sources (i.e. the population Value size). The employed bee of a nectar source that has run out of nectar turns into a scout bee again. The amount of the nectar in a nectar source is expressed with the objective function value of the related nectar source. Therefore, the ABC algorithm targets to place the nectar source that has the maximum amount of nectar.

In the initial step of the ABC algorithm, initial-nectar sources are arbitrarily generated by using the Eq.7. While starting the calculations, Value of the failure variable in which the number of failures to develop a nectar source is unseen made as $\text{failure}_i = 0$ (Karaboga and Akay 2009a). Following the creation of initial nectar resources, the ABC algorithm starts to search for the solution of the numeric optimization problem using the basic tools. The employed bee tries to increase the nectar source to which it is assigned using the

other nectar sources as well. If the employed bee finds a superior nectar source to use it instead of the old one. This process is modelled in Eq.10;

$$V_{r1,j} = X_{r1,j} + \Phi_{r1,j} \cdot (X_{r1,j} - X_{r2,j}) \quad (11)$$

Where $\Phi_{i,j}$ is a arbitrary number generated in the range of [-1 1]. $X_{r1,j}$ and $X_{r2,j}$ specify the j th parameters of the $r1$ th and $r2$ th pattern (i.e., nectar source in this algorithm) respectively.

If the v_{r1} value has a superior objective function value than the X_{r1} value, the X_{r1} value is updated as $X_{r1} := v_{r1}$ and the crash variable become $\text{failure}_{r1} = 0$. If the v_{r1} value does not have a superior objective function value then X_{r1} , the employed bee continued to go to the X_{r1} source, and since the X_{r1} solution cannot be improved, the failure_{r1} value that is the development meter related to the nectar source X_{r1} increases by one unit.

Using the objective function value of all nectar sources, the probability values, p_i , to be used by the onlooker bees are obtained by using Eq.11;

$$P_i = \frac{\text{fitness}_i}{\sum_{i=1}^N \text{fitness}_i} \quad (12)$$

As the fitness_i value is increases, the number of employed bees that will select this region of nectar source will enhance. The ABC algorithm selects the nectar sources to be visited by the bees using the roulette selection technique used in the genetic algorithms; a arbitrary number within the range of [0 1] is generated for each nectar source, if the P_i value is higher than the generated arbitrary number, the onlooker bees search for new nectar sources to develop the nectar source X_i using the Eq.11. If a X_i sources has a failure_i value higher than a assured threshold value, that X_i sources is left, and the employed bee assigned hereto goes to a arbitrary nectar source generated newly.

Table 1: The Benchmark function used to examine the numerical optimization problem solving successes of the CK, DE, and ABC algorithms and features of these functions.

FNC	NAME	LOW	UP
F1	ACKLEY	-32	32
F2	BOHACHECKSKY	-100	100
F3	BOOTH	-10	10
F4	BRANIN	-5	10
F5	GRIEWANK	-600	600
F6	LANGERMANN	0	10
F7	PERM	-4	4
F8	RASTRIGIN	-5.12	5.12
F9	TRID	-100	100
F10	ZAKHAROV	-5	10

Table 2. The MeanOpt values of the CK, PSO, DE, and ABC algorithms.

	CK	PSO	DE
F1	0.0000000000000044	0.0000000000000080	0.000000000000000044
F2	0.0000000000000000	0.0000000000000000	0.0000000000000000
F3	0.0000000000000000	0.0000000000000000	0.0000000000000000
F4	0.3978873577297382	0.3978873577297382	0.3978873577297382
F5	0.0000000000000000	0.0092269113767599	0.0011086678563541
F6	-1.0809384421344377	-1.0809384421344377	-1.0809384421344377
F7	0.0000528002823306	0.0013823741086214	0.0051435833427266
F8	1.2828840539586595	28.0080704889407630	15.3906006782917850
F9	-210.000000000028700	-210.000000000002300	-210.000000000036400
F10	0.0000000000000000	0.0000000000000000	0.0000000000000000

Table 3: The BestOpt values of the CK, PSO, DE, and ABC algorithm.

	CK	PSO	DE
F1	0.0000000000000044	0.0000000000000080	0.0000000000000044
F2	0.0000000000000000	0.0000000000000000	0.0000000000000000
F3	0.0000000000000000	0.0000000000000000	0.0000000000000000
F4	0.3978873577297382	0.3978873577297382	0.3978873577297382
F5	0.0000000000000000	0.0000000000000000	0.0000000000000000
F6	-1.0809384421344377	-1.0809384421344377	-1.0809384421344377
F7	0.0000000085822797	0.0000088339092326	0.0047965266340616
F8	0.0003806178377204	13.9294167236679410	7.9596724567463761
F9	-210.000000000036400	-210.0000000000027300	-210.0000000000036400
F10	0.0000000000000000	0.0000000000000000	0.0000000000000000

3. Conclusion

In this paper, the mathematical optimization problem solving successes of the CK, PSO, DE, and ABC algorithms have been compared statistically. Statistical study pleased that the problem solving achievement of the CK and DE algorithms are quite better than the PSO and ABC algorithms. The PSO algorithm is successful in the solution of many benchmark functions, but its famous stability problem restricts the success rate of this algorithm against the CK and DE algorithms. Since the PSO algorithm maintains its stochastic

behaviour ability better than the ABC algorithm while searching for the global optimum value, it provides more successful answer than the ABC algorithm. The ABC algorithm basically has a search strategy, which is considerably similar to the standard DE algorithm. However, the ABC algorithm has a very victorious decision mechanism that decides which areas within the search space require to be surveyed in more details.

References

- [1] Pinar Civicioglu, Erkan Besdok (2011) "A conceptual comparison of the Cuckoo-search, Particle swarm optimization, Differential evolution and Artificial bee colony algorithm" *Artif Intell Rev* DOI 10.1007/s10462-011-9276-0.
- [2] Akay B, Karaboga D (2010) "A modified artificial bee colony algorithm for real-parameter optimization" *Inf Sciv* (in press, online version).
- [3] Bin X, Jie C, Zhi-Hong P, Feng P(2010) "An adaptive hybrid optimizer based on particle swarm and differential evolution for global optimization" *Sci China Inf Sci* 53(5):980-989.
- [4] Clerc M, Kennedy J (2002) "The particle swarm-explosion, stability, and convergence in a multidimensional complex space." *IEEE Trans Evol Comput* 6(1):58-73.
- [5] Des S, Sugantham P (2009) "Differential evolution: a survey of the state-of-the-art." *IEEE Trans evol Comput* 15(1):4-31.
- [6] Dorigo M, Bonabeau E, Theraulaz G (2000) "Ant algorithms and stigmergy." *Future Gener Comput Syst* 16(8):851-871.
- [7] Haupt R (1995) "Comparison between genetic and gradient-based optimization algorithm for solving electromagnetic problems" *IEEE Trans Magn* 31(3):1932-1935.
- [8] Horst R, Pardalos PM, Thoai NV (2000) "Introduction to global optimization." Kluwer Academic Publishers, Dordrecht, The Netherland.
- [9] Horst R, Pardalos PM, Thoai NV (2000) "Introduction to global optimization." Kluwer Academic Publishers, Dordrecht, The Netherland.
- [10] Janez B, Borko B, Saso G et al (2007) "performance comparison of self-adaptive and adaptive differential evolution algorithms" *Soft Comput* 11(7):617-629.
- [11] Juang C (2004) "A hybrid of genetic algorithm and particle swarm optimization for recurrent network design." *IEEE Trans Syst Man Cybern part B Cybern* 34(2); 997-1006.
- [12] Kaelo p, Ali MM (2006) "A numerical study of some modified differential evaluation algorithm." *Eur J Oper Res* 169(3); 1176-1184.
- [13] Karaboga D, Akay B (2009a) "A comparative study of artificial bee colony algorithm." *Appl Math Comput* 214(12); 108-132.
- [14] Karaboga D, Akay B (2009b) "A survey: algorithm simulating bee swarm intelligence." *Artif Intell Rev* 31(1-4); 61-85.
- [15] Karaboga D, Basturk B (2007a) "Artificial bee colony (abc) optimization algorithm for solving constrained

optimization problems.” Lecture Notes Comput Sci 4529; 789-798.

- [16] Yang XS (2009) “Firefly algorithms for multimodel optimization.” Lecture Notes Comput Sci 5792; 169-178.
- [17] Trelea IC (2003) “The particle swarm optimization algorithm: convergence analysis and parameter selection.” Inf Process Lett 85(6):317-325.

Author Profile



S. Gunasekaran received the B.E. and M.E. degrees in Computer Science and Engineering from Kumaraguru College of Technology in 1997 and 2003, respectively. During 2007-2010, he started Phd research in Computer Science and Engineering at Anna University, Coimbatore, India. His area of interest are Cloud computing, MANET, Data Mining and Data Warehousing. His professional Experience is about 10 years. Presently, working as Head of the Department at Coimbatore Institute of Engineering and Technology TamilNadu, India.



S. Sonialpriya received the B.E graduation in Computer Science and Engineering from Coimbatore Institute of Engineering and Technology in 2009-2013 respectively and currently pursuing Master of Engineering in Computer Science and Engineering at Coimbatore Institute of Engineering and Technology, TamilNadu, India.