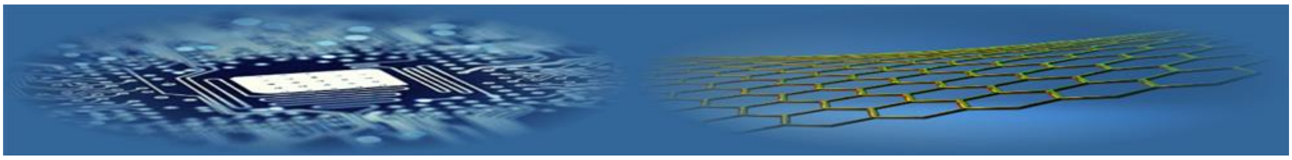

Comparison of Cuckoo Search, Tabu Search and TS-Simplex algorithms for unconstrained global optimization

Ghania Khensous[1, 2*], Belhadri Messabih[2], Abdellah Chouarfia[2], Bernard Maigret[3]

[1]Université des Sciences et de la Technologie d'Oran USTO-MB, Oran, Algérie
[2]Ecole Normale Supérieure d'Oran ENSO, Oran, Algérie
[3]CNRS, LORIA, Vandoeuvre-Les-Nancy, France

[*]Corresponding author: gh.khensous@yahoo.fr

Received 17 December 2016, www.cmnt.lv

Abstract

Metaheuristics Algorithms are widely recognized as one of the most practical approaches for Global Optimization Problems. This paper presents a comparison between two metaheuristics to optimize a set of eight standard benchmark functions. Among the most representative single solution metaheuristics, we selected Tabu Search Algorithm (TSA), to compare with a novel population-based metaheuristic: Cuckoo Search Algorithm (CSA). Empirical results reveal that the problem solving success of the TSA was better than the CSA. However, the run-time complexity for acquiring global minimizer by the Cuckoo Search was generally smaller than the Tabu Search. Besides, the hybrid TSA-Simplex Algorithm gave superior results in term of efficiency and run-time complexity compared to CSA or TSA tested alone.

Keywords

Metaheuristic Algorithms
CSA
TSA
Global Optimization
Nature Inspired Algorithms

1 Introduction

Global Optimization has been an active area of research for several decades since optimization problems are inherent in nearly every research area, ranging from engineering to the natural sciences such as Biology or Chemistry. It is also an active research topic in many other areas such as Mathematics, Business, and the Social Sciences [1]. As many real-world optimization problems become more complex, better optimization algorithms were needed.

In all optimization problems, the goal is to find the minimum or the maximum of the objective function. Therefore, the aim of optimization is to obtain the relevant parameter values allowing an objective function the generation of the minimum or maximum value. Thus, unconstrained optimization problems can be formulated as the minimization or the maximization of D-dimensional function [2]:

$$Min\ (or\ Max)\ f(x),\ x = (x_1,\ x_2,\ x_3, \ldots x_D) \qquad (1)$$

The challenge of developing new methods, baptized Metaheuristics, which are better able to solve difficult problems, still attracts the interests of current researchers. Metaheuristic optimization is therefore a field of growing interest since a single metaheuristic optimization algorithm, which can solve all optimization problems of different types and structures, does not exist.

The metaheuristic optimization algorithms use two basic strategies while searching for the global optimum: exploration and exploitation [3]. The exploration process succeeds in enabling the algorithm to achieve the best local solutions within the search space, whereas the exploitation process expresses the ability to reach the global optimum solution around the obtained local solutions.

A metaheuristic algorithm must have some characteristics such as [4]: it must be able to reach rapidly the global optimum solution; the total calculation amount and the run-time required to reach the optimum must be acceptable for practical applications. The algorithmic structure of a metaheuristic has also to be simple enough to allow its easy adaptation to different problems. Besides, it is desired that the metaheuristics have very few algorithmic control parameters excluding the general ones like total Number of iterations or the size of the population (for the population based optimization algorithms).

There are a wide variety of metaheuristics and a number of properties allowing their classification. One classification dimension is single solution vs. population-based [5]: Single solution approaches focus on modifying and improving a single candidate solution such as Simulated Annealing (SA) and Tabu Search Algorithm (TSA). Whereas population-based approaches maintain and improve multiple candidate solutions such as Genetic Algorithms (GA) and Cuckoo Search Algorithm (CSA).

Several comparisons of the efficiency of metaheuristic algorithms have been published [6-11]: It has been shown that TSA represents one of the most efficient heuristic techniques to find good quality solutions in a short running time compared to population-based algorithms such as GA or Ant Colony Optimization (ACO) [12, 13]. It has been shown also that CSA gave superior results compared to GA, Particle Swarm Optimization (PSO) and Artificial Bee Colony (ABC) [4, 14-15].

In this paper, we applied therefore CSA and TSA for optimizing eight standard test functions with diverse properties: modality, separability, and valley landscape to analyze their effectiveness in terms of solution quality and runtime. We then compared the both metaheuristics to the novel algorithm combining TSA and Nelder-Mead Simplex minimizer.

This paper is organized as follows: Section 2 describes the principles of the applied algorithms: CSA, TSA and Simplex algorithm as well as the test functions. In Section 3, we analyze and compare the results obtained in terms of run-time and solution quality. Section 4 concludes this paper.

2 Material and Method

2.1 CUCKOO SEARCH ALGORITHM

CSA is a novel population based stochastic search metaheuristic proposed by Yang and Deb in 2009 [16-18]. It is inspired by a natural mechanism; the parasitic breeding behavior of some cuckoo species that lay their eggs in the nests of host birds. Therefore, a pattern corresponds to a nest and similarly each individual attribute of the pattern corresponds to a cuckoo egg and the latter represents a new solution. In each computation steps, the new and potentially improved solutions replace the worse solutions (eggs in the nests).

CSA can be briefly described using the following three rules [17-19]:

1. Each cuckoo lays one egg at a time and dumps it in a randomly chosen nest.
2. Best nests with high quality of eggs will be passed to the next generations.
3. The number of available host nests is fixed, and a host bird can discover a foreign egg with a probability pa. In this case, the host bird can throw the egg away or abandon the nest and build a new one in a new location.

We have chosen this population based stochastic global search metaheuristic algorithm because it has been shown that CSA is superior with respect to GA, PSO and ABC [12-14]. Besides, several studies indicate that Cuckoo Search is a powerful algorithm and successful results have been achieved in various applications such as manufacturing optimization [20], physically - based runoff - erosion model [21] Query Optimization [22], Training Artificial Neural Networks [23] and PCB (Printed Circuit Boards) Drill Path Optimization [19] as well as Performing Phase Equilibrium Thermodynamic Calculations [15].

Although this metaheuristics is novel, many improvements are proposed in the literature such as ICS (acronyms of Improved Cuckoo Search) which is proposed to enhance the accuracy and the convergence rate of this algorithm [24]. In this version, a proper strategy for tuning the cuckoo search parameters is used instead of keeping these parameters constant.

Another modified cuckoo search algorithm is also presented in [25]: the authors implemented a CSA version where the step size is determined from the sorted, rather than only permuted fitness matrix.

In exploring the search space, Yang and Deb discovered that the performance of the CSA could be significantly improved by using Lévy Flights instead of simple random walk [18] since Lévy Flight can maximize the efficiency of resource searches in uncertain environments. For this reason, we have selected this version of the CSA algorithm. In the other hand, this CSA version has outperformed both GA and PSO for all the test functions used in [16-17].

The different steps of the CSA implemented in our work (the minimization of test functions) can be summarized in the following flow chart [16-19].

In its original version, CSA is proposed for continuous problems; however, it can be extended for combinatorial discrete optimization problems [19, 26]. It can also be combined with others metaheuristics such as TSA [22], Scatter Search [23] and Greedy Randomized Adaptive Search Procedure (GRASP) [27].

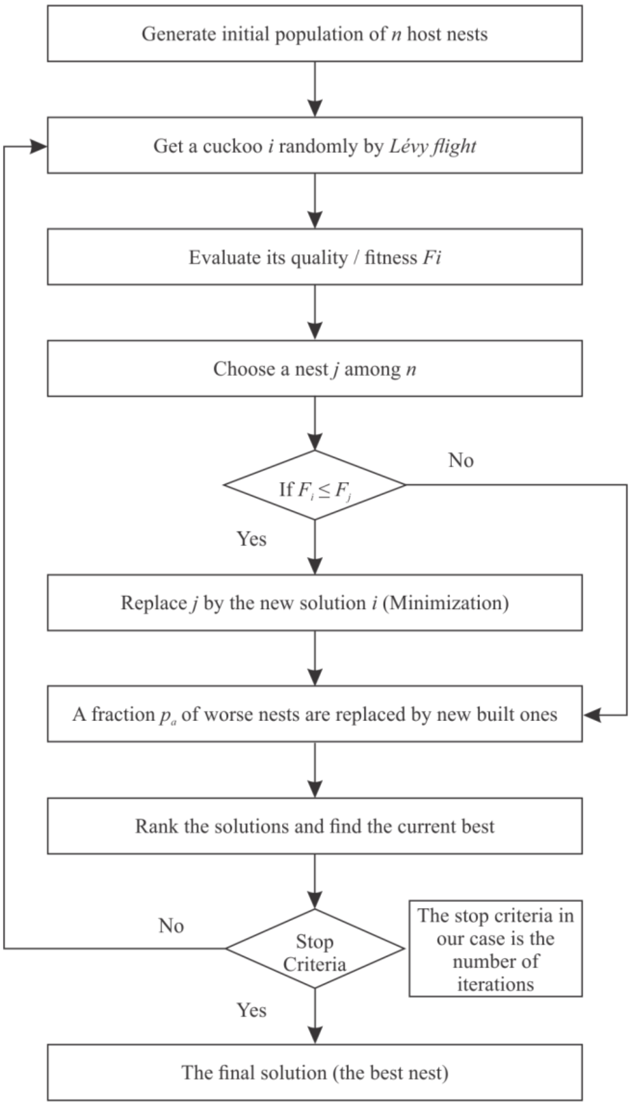

FIGURE 1 Flow Chart of CSA

2.2 TABU SEARCH ALGORITHM

TSA was first proposed by Fred Glover in 1986 [28]. It is inspired by human memory. It is so called because it avoids returning to recently visited solutions. At each iteration, the best neighbor is selected as a current solution. To avoid cycles, i.e.; the infinite repetition of a sequence of movements, the L latest movements are forbidden (L is the length of the tabu list, which is a short-term memory. It contains the best conformations already visited). Then, the selected movements must be the best ones and not in the tabu list.

Although it might seem simple to reject a solution to a discrete combinatorial problem if it appears in the tabu list, this is not the case for continuous problems.

As for other metaheuristics, a random candidate solution within a neighborhood can be defined. If this solution has an objective value higher than the current solution (minimization), the decision whether to accept it or not is based on the content of the tabu list. However, rather than checking if the solution is already tabu it should be checked if the solution is within a certain distance of a solution in the tabu list [29]. A TSA with this property is called Enhanced Continuous Tabu Search (ECTS) [30].

ECTS is proposed for the global optimization of multi-minima functions, it results from an adaptation of combinatorial TSA that aims to follow Glover's basic approach as closely as possible. In order to cover a wide domain of possible solutions, this algorithm first performs the diversification: it locates the most promising areas, by fitting the size of the neighborhood structure to the objective function and its definition domain. For each located promising area, the algorithm continues the search by intensification within one promising area of the solution space.

The flow chart presented on Figure 2 outlines the different steps of the ECTS used in our work.

We have chosen this variant for its advantages [30]: first, its principle is rather basic, directly inspired from combinatorial Tabu Search. Secondly, the authors tested and compared the efficiency of ECTS to other published versions of Continuous Tabu Search and to some alternative algorithms like Simulated Annealing. The results revealed that ECTS showed a good performance for functions having a large number of variables.

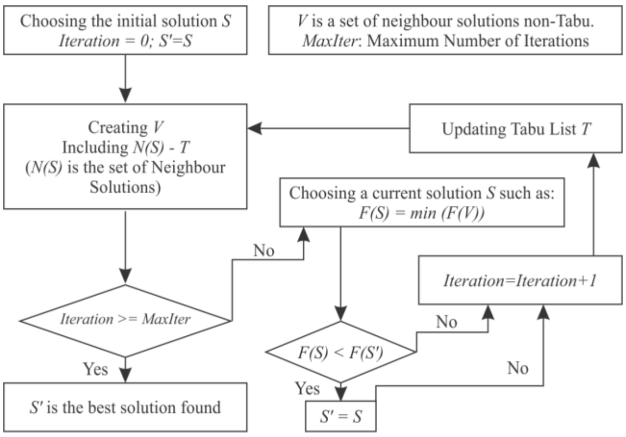

FIGURE 2 Flow Chart of TSA

Among the neighbourhood search methods, TSA is considered as one of the most prominent, being widely used and providing a powerful approach for solving a large range of optimization problems [31]. TSA, which also has the advantage that only function values are used, (differentiability and continuity being not required), is characterized by the use of "memories" during the search [32]. Additionally, TSA needs fewer parameters to be adjusted than the SA algorithm. Unlike other metaheuristics, TSA is not trapped in local minimum [29].

TSA is subject to several developments such as: Directed Tabu Search (DTS), which is a continuous TSA [33]. The Memory Models are also introduced in order to improve Tabu Search with real continuous variables [34].

2.3 SIMPLEX ALGORITHM

The Nelder-Mead minimization method [35] is based on the comparison of function values at *n+1* vertices of a general simplex. The simplex adapts itself via Reflection, Expansion as well as Contraction operations by replacing the vertex with the highest value by another point with lower value.

Figure 3 illustrates the principle of the Nelder–Mead Simplex algorithm.

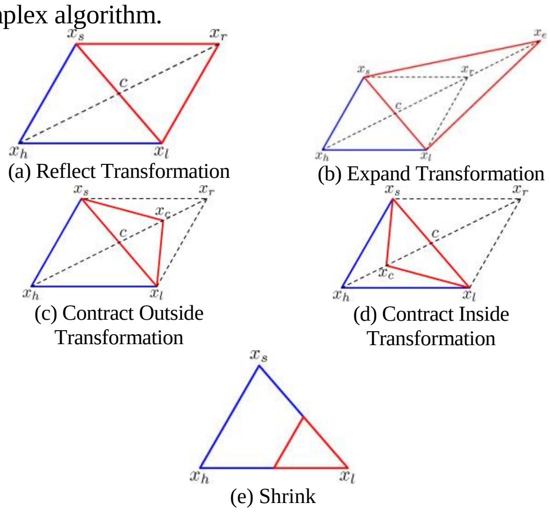

FIGURE 3 Nelder–Mead Simplex Algorithm's Principle

We have opted for the Nelder–Mead Simplex algorithm because it is a classical very powerful local descent algorithm, making no use of the objective function derivatives [36].

An overview of the algorithm is outlined in Figure 4 [37-38]:

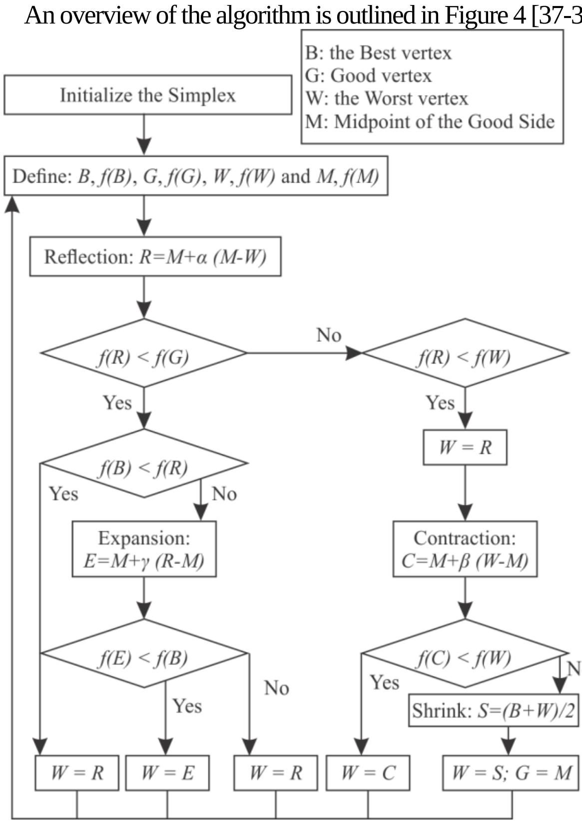

FIGURE 4 Flow Chart of Nelder-Mead Simplex Algorithm

2.4 HYBRID TSA-SIMPLEX ALGORITHM

In spite of the numerous advantages of the Tabu Search, it might not find a near-optimal solution for some problems, especially continuous ones [29], but it can find several good starting points for local search. For this reason, and in order to improve TSA effectiveness, we have applied the Nelder-Mead minimization to the TSA solutions. Therefore, the initial simplex is composed of the TSA solutions.

We have opted for the combination of these algorithms because many comparisons are made in this way. These comparisons concluded that hybrid search algorithms gave superior results compared with any of the algorithms tested individually [39-41].

2.5 TEST FUNCTIONS

Test functions are important to validate new optimization algorithms and to compare the performance of various algorithms. There are many test functions in the literature [42-45], but there is no standard list or set of benchmark functions to be followed.

In order to make sure whether the tested algorithms can solve certain types of optimization efficiently, test functions should have diverse properties. So, we select a list of eight test problems usually used for checking the properties of the optimizers.

The details of the continuous test functions used in our work are summarized in the following table 1.

TABLE 1 Test Functions

Beale Function (Non-Separable, Non-Scalable and Unimodal)	
$f_1(x,y) = (1.5-x+xy)^2 + (2.25-x+xy^2)^2 + (2.625-x+xy3)^2$	Search Space: [-4.5;4.5] Global Minimum: $f_1(3\ ;0.5)=0$
De Joung (or DJ) Function (Separable, Scalable, Unimodal)	
$f_2(x,y) = x^2+y^2$	Search Space: [-5.12;5.12] Global Minimum: $f_2(0\ ;0)=0$
Goldstein and Price (or GP) Function (Non-Separable, Non-Scalable and Multimodal)	
$f_3(x,y)=[1+(x+y+1)^2\times(19-14x+3x^2-14y+6xy+3y^2)]\times [30+(2x-3y)^2\times(18-32x+12x^2+48y-36xy+27y^2)]$	Search Space: [-2;2] Global Minimum: $f_3(0\ ;-1)=3$
Himmelblau Function (Non-Separable, Non-Scalable, and Multimodal)	
$f_4(x,y) = (x^2+y-11)^2 + (x+y^2-7)^2$	Search Space: [-6;6] Global Minimum: $f_4(3\ ;2)=0$
Matyas Function (Non-Separable, Non-Scalable and Unimodal)	
$f_5(x,y) = 0.26(x^2+y^2) - 0.48xy$	Search Space: [-10;10] Global Minimum: $f_5(0\ ;0)=0$
Rastrigin Function (Separable, Scalable and Multimodal)	
$f_6(x,y) = 20 + (x^2-10\cos(2\pi x)) + (y^2-10*\cos(2\pi y))$	Search Space: [-5.12;5.12] Global Minimum: $f_6(0\ ;0)=0$
Rosenbrock Function (Non-Separable, Scalable and Unimodal)	
$f_7(x,y) = 100(x^2-y)^2 + (x-1)^2$	Search Space: [-10;10] Global Minimum: $f_7(1\ ;1)=0$
Step Function (Separable, Scalable and Unimodal)	
$f_8(x,y) = (x+0.5)^2 + (y+0.5)^2$	Search Space: [-100;100] Global Minimum: $f_8(0.5;0.5)=0$

These functions have diverse properties in terms of modality, separability and valley landscape: According to [46], the modality of a function corresponds to the number of ambiguous peaks in the function landscape. If these peaks are encountered during an exploration process, there is a tendency that the algorithm may be trapped in one of such peaks. This will have a negative impact on the search process, since it can direct the search away from the true optimal solutions. So, a function with more than one local optimum is called multimodal. These functions are used to test the ability of an algorithm to escape from any local minimum.

Another test problem is formulated by separable and non-separable functions [46]. The dimensionality of the search space is an important issue with the problem. In general, separable functions are relatively easy to optimize, when compared with their inseparable counterpart, because each parameter of a function is independent of the other parameters. If all the variables are independent, then we can perform a sequence of n (n being the number of independent variables) independent optimization processes.

Finally, a valley occurs when a narrow region of little change is surrounded by areas of steep descent [46] (this region attracts the minimizers). The progress of a search process of an algorithm may be slowed down significantly on the floor of the valley. Functions with flat surfaces pose a difficulty for the algorithms, as the flatness of the function does not give the algorithm any information to direct the search process towards the minima.

3 Experiment

To verify the reliability of the CSA, TSA and Nelder-Mead Simplex algorithms, several well-known test functions as shown in table 1 are considered.

The parameters of the CSA, TSA and Simplex algorithms used in our experiments are given in the table 2 below.

TABLE 2 The parameters of the CSA, TSA and Simplex Algorithms

Algorithm	Parameters
CSA	Number of Nests = 25 Discovery Rate = 0.25
TSA	Tabu List Length = 10 Neighborhood Size = 10
Simplex	Alpha = 1.0 Beta = 0.5 Gamma = 2.0

We have executed each algorithm for 1000, 10000 and 100000 iterations. The table 3 shows the run times of the algorithms CSA, Simplex and TSA for 1000 iterations.

TABLE 3 The Run-time of CSA, TSA and Simplex Algorithms

Function	Algorithm	Run-time (seconds) for 1000 Iterations
f_1	CSA	0.181127
	TSA	999.355000
		0.004000
f_2	CSA	0.174693
	TSA	999.476000
	Simplex	0.002000
f_3	CSA	0.153440
	TSA	999.185000
	Simplex	0.0000000
f_4	CSA	0.384832
	TSA	999.486000
	Simplex	0.002000
f_5	CSA	0.137662
	TSA	999.496000
	Simplex	0.001000
f_6	CSA	0.258002
	TSA	999.363000
	Simplex	0.003000
f_7	CSA	0.422266
	TSA	1001.631000
	Simplex	0.003000
f_8	CSA	0.203946
	TSA	999.305000
	Simplex	0.001000

From this table, we can note that the simplex algorithm is the best in terms of running time and the TSA is the slowest algorithm. These findings remain true even for 10000 and 100000 iterations.

We have then executed each algorithm ten times for each benchmark function. The following table 4 shows the experimental results of the comparative performances in terms of best, worst and average solutions between the Cuckoo Search and Tabu Search Algorithms.

TABLE 4 Best, Worst and Average Solutions of CSA and TSA

Test function	CSA			TSA		
	Best	*Worst*	*Average*	*Best*	*Worst*	*Average*
f_1	0.00099507	0.1729	0,0420371609	0.003130	9.819423	1,3011135
f_2	0,0003587200	0,0032968000	0,0012535720	0,0000010000	0,0001410000	0,0000360909
f_3	3,0121	4,9408	3,93157	3,000088	97,698648	38,0309893
f_4	0,007139	0,10514	0,0340206	0,000055	0,021961	0,0043994
f_5	0,0000652940	0,0066796000	0,0024487004	0,0003460000	0,0355960000	0,0172967000
f_6	0,0966590000	1,9902000000	0,9387889000	0,0001740000	0,0189600000	0,0041338000
f_7	0,0332870000	1,4028000000	0,5478952000	0,0124640000	3,8449870000	1,0535761000
f_8	0,12119	2,5204	0,87182667	0,0019300000	0,0195940000	0,0111310909

From the table 4 above, we can note that TSA is more efficient since TSA solutions are better than the CSA ones for six functions. However, TSA is slower than CSA.

Besides, we can note that CSA is better for Non-Separable, Non-Scalable and Unimodal functions (Beale and Matyas functions).

In order to improve the TSA run-time, we have executed the hybrid algorithm TSA-Simplex: we first execute the TSA for only 10 iterations. We have then applied the Nelder-Mead minimization (1000 iterations) to the TSA solutions. Therefore, the initial simplex is composed of the TSA solutions.

The following table 5 shows the results of this combination.

TABLE 5 TSA-Simplex Results

Test function	Best	Worst	average	Runtime average
f_1	0,0000000000	0,1651646790	0,0337103523	9,90962
f_2	0,0000000000	0,00077629	0,000077629	9,90236
f_3	3,0000000000	30,1994131088	14,2549998587	9,9053
f_4	0,0000000000	0,0062818	0,00155021	9,90369
f_5	0,0000000000	0,01367587	0,00136759	9,90418
f_6	0,0000000000	0,00019708	0,000022294	9,90489
f_7	0,0000000000	3,13065288	0,77918326	9,90416
f_8	0,0000000000	0,00650235	0,00103485	9,90568

The table 5 above shows obviously that the hybrid algorithm finds the exact solution to all the benchmark functions even in acceptable run-time. These findings remain true even when minimizing functions with 3, 4 and 5 dimensions (3, 4 and 5 variables instead of 2) for all benchmarks used in our work.

4 Conclusion

In this study, we have selected three metaheuristic algorithms: CSA, TSA and Simplex method for the test of eight difficult optimization functions with diverse properties: modality, separability, and valley landscape to analyze their effectiveness in terms of solution quality and runtime. The functions were systematically optimized by the different metaheuristics and the results were tracked and compared.

The results show clearly that TSA is more reliable than CSA since the best TSA solutions are better than the CSA ones in 6 functions (f_2, f_3, f_4, f_6, f_7 and f_8) among 8 (see table 4), whereas CSA is faster than TSA (see table 3). Note also that CSA gives good results for Non-Separable, Non-Scalable and Unimodal functions (f_1, f_5).

To improve the runtime of the TSA, we have combined it with simplex algorithm (as it has the best run-time). The hybrid algorithm is the more reliable as it successfully optimized all functions and found the global minima for each one within reasonable run-time.

With regard to the future, we believe that the application of these metaheuristics to solve real-world problems such as molecular docking is promoting [47-51]. On the other hand, the CPU time could be drastically reduced by using a parallel version of these metaheuristics [52-54] that could be easily implemented on GPUs (Graphical Processing Units).

References

[1] Grebner C 2012 New Tabu Search Algorithms for the Exploration of Energy Landscapes of Molecular Systems *WÜRZBURG*

[2] Tuba M, Subotic M, Stanarevic N 2011 Modified cuckoo search algorithm for unconstrained optimization problems *Proceedings of the European Computing Conference, Paris*

[3] Rashedi E, Nezamabadi-pour H, Saryazdi S 2009 GSA: A Gravitational Search Algorithm *Information Sciences* **179** 2232–48

[4] Civicioglu P, Besdok E 2011 A conceptual comparison of the Cuckoo-search, particle swarm optimization, differential evolution and artificial bee colony algorithms *Artif Intell Rev* DOI 10.1007/s10462-011-9276-0

[5] Layeb A 2009 *Introduction aux métaheuristiques* Université Mentouri de Constantine

[6] Bajeh A O, Abolarinwa K O 2011 Optimization: A Comparative Study of Genetic and Tabu Search Algorithms *International Journal of Computer Applications* (0975-8887) **31**(5)

[7] Kannan S, Slochanal S M R, Padhy N P 2005 Application and Comparison of Metaheuristic Techniques to Generation Expansion Planning Problem *IEEE Transactions on Power Systems* 20(1)

[8] Madić M, Marković D, Radovanović M 2013 Comparison of MetaHeuristic Algorithms for Solving Machining Optimization Problems *Facta Universitatis Series: Mechanical Engineering* **11**(1) 29-44

[9] Silberholz J, Golden B 2010 *Comparison of Metaheuristics* University of Maryland

[10] Azimi Z N 2004 Comparison of Metaheuristic Algorithms for Examination Timetabling Problem *J. Appl. Math. & Computing* **16**(1-2) 337-54

[11] Arora T, Gigras Y 2013 A Survey of Comparison between Various Metaheuristic Techniques for Path Planning Problem *International Journal of Computer Engineering & Science*

[12] Pirim H, Bayraktarn E, Eksioglu B 2008 Tabu Search: A Comparative Study *Wassim Jaziri* ISBN 978-3-902613-34-9, 278 I-Tech, Vienna, Austria

[13] Pappas C, Srinivasan R K, Barnes J W, Sarkar S 2004 *A Comparison of Heuristic, Meta-Heuristic and Optimal Approaches to the Selection of Conservation Area Networks* University of Texas at Austin

[14] Sangita Roy, Sheli Sinha Chaudhuri 2013 Cuckoo Search Algorithm using Lèvy Flight: A Review *I.J. Modern Education and Computer Science* **12** 10-5

[15] Seif-Eddeen K Fateen, Bonilla-Petriciolet A 2014 On the Effectiveness of Nature-Inspired Metaheuristic Algorithms for Performing Phase Equilibrium Thermodynamic Calculations *The Scientific World Journal*

[16] Yang X-S, Deb S 2010 Engineering Optimisation by Cuckoo Search *Int. J. Mathematical Modelling and Numerical Optimisation* **1**(4) 330–43

[17] Yang X-S, Deb S 2009 Cuckoo search via Lévy flights *Proc. of World Congress on Nature & Biologically Inspired Computing (NaBIC 2009), December 2009, India. IEEE Publications, USA* 210-4

[18] Yang X-S 2010 *Nature-Inspired Metaheuristic Algorithms Second Edition* Luniver Press

[19] Wei Chen Esmonde Lim, Kanagaraj G, Ponnambalam S G 2014 PCB Drill Path Optimization by Combinatorial Cuckoo Search Algorithm *Scientific World Journal* Article ID 264518 http://dx.doi.org/10.1155/2014/264518

[20] Syberfeldt A, Lidberg S 2012 Real-World Simulation-Based Manufacturing Optimization using Cuckoo Search *WSC Simulation Conference*

[21] Santos C A G, Freire P K M M, Mishra S K 2012 Cuckoo Search via Lévy Flights for Optimization of a Physically - Based Runoff - Erosion Model *Journal of Urban and Environmental Engineering (JUEE)* **6**(2) 123-31

[22] Mukul Joshi, Praveen Ranjan Srivastava 2013 Query Optimization: An Intelligent Hybrid Approach using Cuckoo and Tabu Search *International Journal of Intelligent Information Technologies* **9**(1) 40-55

[23] Al-Obaidi A T S 2013 Improved Scatter Search Using Cuckoo Search *IJARAI - International Journal of Advanced Research in Artificial Intelligence* **2**(2)

[24] Ehsan Valian, Shahram Mohanna, Saeed Tavakoli 2011 Improved Cuckoo Search Algorithm for Global Optimization *International Journal of Communications and Information Technology, IJCIT*

[25] Tuba M, Subotic M, Stanarevic N 2012 Performance of a Modified Cuckoo Search Algorithm for Unconstrained Optimization Problems *WSEAS Transactions on Systems* E-ISSN: 2224-2678 **11**(2)

[26] Gherboudj A 2013 *Méthodes de résolution de problèmes difficiles académiques* Université de Constantine **2**

[27] Hongqiang Zheng, Yongquan Zhou, Qifang Luo: 2013 A hybrid Cuckoo Search Algorithm-GRASP for Vehicle Routing Problem *Journal of Convergence Information Technology (JCIT)* **8**(3)

[28] Glover F 1990 *Tabu Search: A Tutorial* University of Colorado

[29] Schulz P G 2006 *Creative Design in Optimization Metaheuristics Applied to Multi-modal Continuous Functions* Kongens Lyngby

[30] Chelouah R, Siarry P 2000 Tabu Search applied to global optimization *European Journal of Operational Research* **123** 256-70

[31] Py Y 2008 Local Search Techniques: Focus on Tabu Search *Wassim Jaziri Ed.* ISBN 978-3-902613-34-9 278

[32] Morales L B, Garduño–Juárez R, Aguilar–Alvarado J M, Riveros-Castro F J 2000 A Parallel Tabu Search for Conformational Energy Optimization of Oligopeptides *J Comput Chem* **21** 147–56

[33] Hedar A, Fukushima M 2006 Continuous Optimization: Tabu Search directed by direct search methods for nonlinear global optimization *European Journal of Operational Research* **170** 329–49

[34] Connor D A M 2006 Memory Models for Improving Tabu Search with Real Continuous Variables *Proceedings of the Sixth International Conference on Hybrid Intelligent Systems*

[35] Nelder J A, Mead R 1965 A Simplex method for function minimization *The Computer Journal*

[36] Chelouah R, Siarry P 2005 A hybrid method combining continuous tabu search and Nelder–Mead simplex algorithms for the global optimization of multiminima functions *European Journal of Operational Research* **161** 636-54

[37] Mathews J H, Fink K K 2004 *Numerical Methods Using Matlab* Prentice-Hall Inc. Upper Saddle River, New Jersey USA, ISBN: 0-13-065248-2

[38] Tomick J J 1995 *On convergence of the Nelder-Mead Simplex algorithm for unconstrained stochastic optimization* The Pennsylvania State University

[39] Hou T J, Wang J M, Xu X J 1999 A Comparison of Three Heuristic Algorithms for Molecular Docking *Chinese Chemical Letters* **10**(7) 615–8

[40] Koduru P, Das S, Welsh S M 2006 A Particle Swarm Optimization-Nelder Mead Hybrid Algorithm for Balanced Exploration and Exploitation in Multidimensional Search Space *Proceedings of the ICAI - International Conference on Artificial Intelligence, Las Vegas, Nevada, USA*

[41] Mashinchi M H, Orgun M A, Pedrycz W 2011 Hybrid optimization with improved tabu search *Applied Soft Computing* 1993-2006

[42] Molga M, Smutnicki C 2005 Test functions for optimization needs

[43] Andrei N 2008 An Unconstrained Optimization Test Functions Collection *Advanced Modeling and Optimization* **10**(1)

[44] Yang X S 2010 *Test Problems in Optimization* Department of Engineering, University of Cambridge, UK

[45] Adorio E P 2005 *MVF - Multivariate Test Functions Library in C for Unconstrained Global Optimization* Department of Mathematics U.P. Diliman

[46] Jamil M, Yang X S 2013 A Literature Survey of Benchmark Functions For Global Optimization Problems *Int. Journal of Mathematical Modelling and Numerical Optimization* **4**(2), 150–94

[47] Huang S Y, Zou X 2010 Review Advances and Challenges in Protein Ligand Docking *International Journal of Molecular Sciences*

[48] Kitchen D B, Decornez H, Furr J R, Bajorath J 2004 Docking and Scoring in Virtual Screening for Drug Discovery: Methods and Applications *Nature Reviews*

[49] Sousa S F, Fernandes P A, Ramos M J 2006 *Protein–Ligand Docking: Current Status and Future Challenges* Université de Porto-Portugal

[50] Huang S Y, Grinter S Z, Zou X 2010 Scoring functions and their evaluation methods for protein–ligand docking: recent advances and future directions *Phys. Chem. Chem. Phys.* **12** 12899–908

[51] Wardono B, Fathi Y 2004 A tabu search algorithm for the multi-stage parallel machine problem with limited buffer capacities *European Journal of Operational Research*

[52] Fiechter C N 1994 A parallel tabu search algorithm for large traveling salesman problems *Discrete Applied Mathematics*

[53] Matsumura T, Nakamura M, Tamaki S 2000 *A Parallel Tabu Search and Its Hybridization with Genetic Algorithms* University of the Ryukyus, Japan

AUTHORS

	**Khensous Ghania, 17.01.1981, Oran - Algeria** **Current position, grades:** Ph.D. Computer Sciences student at the USTO-MB University (Université des Sciences et de la Technologie d'Oran- Mohamed BOUDIAF); Assistant Lecturer at Ecole Normale Supérieure d'Oran (ENSO), Algeria. **Scientific interest:** Optimization, Bioinformatics, Artificial Intelligence.
	**Messabih Belhadri, 1958, Oran - Algeria** **Current position, grades:** Senior Lecturer at the USTO-MB University, Oran- Algeria. Ph.D. degree in Computer Sciences from the University of Paris 7, France, in 1997.
	**Chouarfia Abdellah, 1955, Mostaganem - Algeria** **Current position, grades:** Professor at the USTO-MB University, Oran- Algeria. Ph.D. degree in software engineering from the University Paul Sabatier of Toulouse France in 1983.
	**Maigret Bernard, 1944, Nancy - France** **Current position, grades:** Emeritus Research Director at CNRS - France. Ph.D. degree in theoretical physique from the University of Paris, France in 1976.