American International Journal of Research in Science, Technology, Engineering & Mathematics

Available online at http://www.iasir.net

ISSN (Print): 2328-3491, ISSN (Online): 2328-3580, ISSN (CD-ROM): 2328-3629

AIJRSTEM is a refereed, indexed, peer-reviewed, multidisciplinary and open access journal published by International Association of Scientific Innovation and Research (IASIR), USA

(An Association Unifying the Sciences, Engineering, and Applied Research)

The Behavior of Mathematical Functions of Xcos (X) and Tanh (X): Designing a New Meta-Heuristic Algorithm

Afsaneh Farajpour Khanaposhtani

MSc Mathematics, Karaj Branch, Islamic Azad University, Karaj, Iran

Abstract: Nowadays meta-heuristic methods have been extensively utilized for obtaining satisfactory results in terms of combinatory optimization. This extensiveness of utility is due to the increased complexity of problems and inability of current mathematical solutions for determination of an optimal point while consuming a rational amount of resources. Meta-heuristic methods are usually developed through investigation of optimizations found in nature and taking inspirations from them. Among these algorithms, it can be referred to the genetic algorithm, ant algorithm and simulated annealing. The pattern proposed by this paper is developed through investigation of the interesting behavior of xCos (x) and tanh (x) functions in loops and proposes a method for finding neighborhood in continuous functions. This, method is more efficient in terms of precision and speed compared to the simulated annealing algorithm and cloud based simulated annealing algorithm. The superiority of the proposed algorithm over the aforementioned ones has been proven through comparison of performance of these algorithms in terms finding the optimal points of seven well known continuous functions.

Keywords: meta-heuristic, neighborhood search, continuous optimization, simulated annealing

I. Introduction

An algorithm is a description of steps that are implemented in computer based programs in a better way in order to find an approximation of an optimal point. Designing an algorithm can have several purposes including obtaining a local optimal point, obtaining a universal optimal point, findings the entire universal optimal points in addition to finding entire local and universal optimal points [7]. Algorithms proposed in the context of combinatorial optimization can be divided into two categories including complete algorithms and approximation algorithms. Complete algorithms guarantee the finding of an optimal point in a limited time for the entire instances of an optimization problem with a finite size [2].

Approximation algorithms were officially proposed in 1960s for production of near-optimal solutions. These algorithms are utilized for optimization problems that cannot be solved efficiently with the existing calculation methods. With emergence of the NP-completeness theory in the beginnings of 1970s, this scientific domain became more important since a great need was felt for production of near-optimal solutions for NP-hard optimization problems for overcoming computational intractability. Algorithms of that time were able to provide a near-optimal solution in a limited time only for a single problem and also, for other problems they were hardly able to produce a suboptimal solution [6].

Approximation methods are generally based on two basic principles: innovative constructive heuristics and local search methods. Innovative constructive heuristics are related to the processes of production of primary results prior to execution of another operation. Local search can be considered as the heuristic mechanism in which the neighbors of the current solution are investigated as potential substitutes for the current solution. If a neighbor was accepted, the procedure continues towards the new solution and the neighbors of the new current solution are investigated [4].

Some scientific sources have divided heuristic technics into two families of specific heuristics and meta-heuristics [11]. Specific heuristics are utilized for solvation of specific problems or instances of them while meta-heuristic ones are general-purpose algorithms that are able to be utilized for almost every optimization problem.

II. Meta-heuristic algorithms

During the past 30 years, new types of approximation algorithms have emerged which are basically aimed at combination of heuristic methods in larger contexts in order to efficiently explore the search space. Currently, these methods are known as meta-heuristic methods. This phrase was for the first time used by Glover (1986) and is a combination of two Greek words of Meta and Heuristic. The prefix of Meta means higher or above and heuristic is defined as finding. Prior to public acceptance of the phrase of meta-heuristic, the term Modern Heuristics was used to refer to these methods [1]. In fact a meta-heuristic method is an innovative approach for

solvation of a very general stratum of problems. In addition, it is an efficient combination of purpose functions or innovative methods based on purpose functions [13].

The main advantage of utilizing meta-heuristic methods lies in existence of limited hypotheses in formulation of the model while it can't be realized with mathematical solutions. Some optimization problems cannot be unambiguously formulated through mathematical-analytic solutions. In fact, in these cases the purpose function is like a black box. There are no analysis rules for the goal in optimization of black boxes [11] (diagram 1).

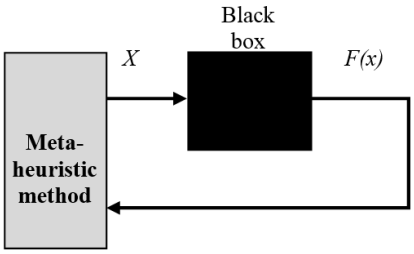

Fig 1, black box scenario for purpose function

Several meta-heuristic algorithms have been proposed which are of some high efficiencies for some specific problems while for other problems may face problems in terms of coming close to the optimal solution. In fact, with respect to the structure of problem and type of its complexity, a meta-heuristics expert will select an algorithm for solvation of the problem and adjusts the parameters related to that algorithm through making use of the scientific approach of design of experiments (DOE). Table 1 shows the most important proposed meta-heuristic methods.

Many of the meta-heuristic methods are inspired from physical or natural processes. Ant colony optimization (ACO), evolutionary algorithms (EAs) and the simulated annealing (SA) are examples of such algorithms. ACO and EAs are inspired from nature and SA is inspired from the process of cooling of metals in strong metals industry.

Table 1, the most important meta-heuristic methods

algorithm	developer	year
Metropolis-Hastings algorithm	Hastings	1970
genetic algorithm	Holland	1975
genetic programming	Smith	1980
simulated annealing	Kirkpatrick et al.	1983
tabu search	Glover	1986
artificial immune system	Farmer et al.	1086
memetic algorithm	Moscato	1989
ant colony algorithm	Dorigo	1992
MOGA for multiobjective optimization	Fleming	1993
NSGA for multiobjective optimization	Fonseca	1994
particle swarm optimization	Kennedy and Eberhart	1995
CMA-ES	Hansen and Ostermeier	1996
Storn and Price	differential evolution	1997
Rubinstein	cross entropy method	1997
harmony search	Geem et al.	2001
NSGA-II for multiobjective optimization	Deb et al.	2002
bee colony optimization	Nakrani and Tovey	2004
Glowworm swarm optimization	Krishnanand and Ghose	2005
Artificial Bee Colony Algorithm (ABC)	Karaboga	2005
honey-bee mating optimization	Haddad et al.	2006
Intelligent Water Drops	Hamed Shah-Hosseini	2007
firefly algorithm	Yang	2008
Monkey Search	Mucherino and Seref	2008
cuckoo search	Yang and Deb	2009
bat algorithm	Yang	2009
X algorithm.	Metaheuristic Solutions	2011

III.Production of solution in meta-heuristics

In each meta-heuristic method, the manner of determination of neighbors of the current solution, method of evaluation of neighbors and the manner of selection of the substitute for the current optimal solution are different. Among different optimization processes, repeated solution production is a very high importance. The most important step in a repeating process is production of the new solution (S') according to the current solution (S) and testing whether the step needs to be repeated or not. Local search technics are repeated processes in which the

neighboring of N(s) is defined for each S solution and the next solutions of the S' are selected from the N(s) solutions [5].

Xen Yao [14] carried out a series of studies and states that simulated annealing (SA) uses a larger number of neighbors and is better than algorithms which use fewer neighbors.

It has been shown that the actual performance of SA is highly dependent on choice of parameters. However, still not many researches have been performed regarding parameters related to neighboring especially in the entire simulated annealing process. Leo performed a series of computational experiments and investigated the effect of size of neighborhood on the process of simulated annealing in the problem of flow-shop scheduling. He proposed an optimized process of simulated annealing with a variable neighborhood size. This method displayed a better performance than the conventional method of simulated annealing in all contexts [8]. Later Lou, Yoan and Xang [9] proposed a method based on cloud theory which allows for a better local search and obtaining better solutions.

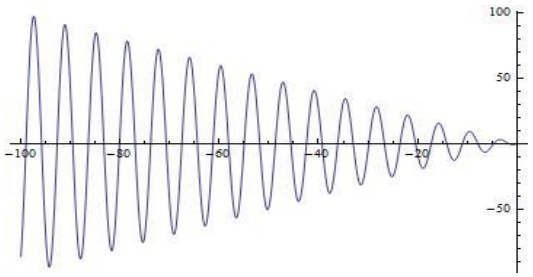

Fig 2, the curve of the xCos(x) function in the interval [0, -100]

IV.Functions used for production of solution

In the method proposed in this article, the two functions of xCos(x) and tanh(x) play a very important role in terms of finding neighbor solutions. In the following, important features of these functions are investigated.

a) xCos(x)

This function has an axis of x=0. fig 2 shows this function in the interval of [0, -100]. As the value of x closes to zero, the range of fluctuation of function value reduces and this characteristic has been used for making the algorithm more convergent.

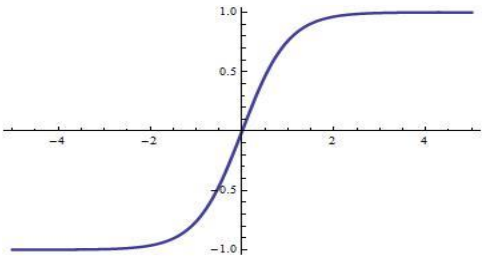

Fig 3, the range of fluctuation of function value reduces

b) tanh(x)

This function has an axis of (0, 0) and a range of (-1, 1). If a constant like A is multiplied by this function, the range becomes (-a, a). This characteristic is used for production of solutions with limited distance from the current solution.

V.Introduction of the proposed algorithm

This algorithm is based on a population of solutions and its most important part is its formulations used for production of optimal solutions from existing solutions. Another important feature of this algorithm is that it has a specific number of repetitions as an input parameter that results in more stability in terms of time based performance of the algorithm. In other words, a very small standard deviation in algorithm's ending times in several executions.

Initialization of parameters

Number of repetitions (**K**)

Number of solutions yielded from the first formula ($\mathbf{n_1}$)

Number of solutions yielded from the second formula ($\mathbf{n_2}$)

Initiation

1 -> i

$F^* \leftarrow +\infty$ (the best value for the purpose function to the time)

Receive upper bound (**U**) and lower bound (**L**) for solutions

Add a number of $\mathbf{n_1+n_2}$ random solutions

Complete the loop between **i=1** and **i= 0.001** with a pace of $-\frac{0.999}{K}$

Repair the ill solutions
Evaluate solutions and find the value of purpose function for each solution
The best value of purpose function among the values of →**F**
The solution related to the best value of purpose function →**S**
If $\mathbf{F^T > F}$, then $\mathbf{F \rightarrow F^T}$ and $\mathbf{S \rightarrow S^T}$

Produce a number of n_1 solutions through the first formula ($iCos(100i) \times R \times S^*$)

Produce a number of n_2 solutions through the second formula ($0.05tanh(iCos(100i)) \times R \times (U - L) + S^*$)

End of the loop
Print $\mathbf{F^T}$ and $\mathbf{S^T}$
End

VI.Testing the performance of the algorithm

In order to investigate the performance of the proposed algorithm, seven well known functions in continuous optimization have been used. These functions were used by Lou, Yoan and Xang [9] for investigation of efficiency of the algorithm of cloud based simulated annealing. These functions are shown in table 2 along with their investigation range. In addition, a three dimensional diagram for these functions is shown in diagrams 4 to 10.

Table 2, characteristics of functions used for testing the performance of the algorithm

Function	Function Criterion	Function Range
Sphere model function	$F_1 = \sum_{i=1}^{5} x_i^2$	$\lvert x_i \rvert \le 30$
Hyper-ellipsoid function	$F_2 = \sum_{i=1}^{10} i^2 x_i^2$	$\lvert x_i \rvert \le 1$
Generalized Rastrigin's function	$F_3 = \sum_{i=1}^{30} \left(x_i^2 - 10\cos(2\pi x_i) + 10 \right)$	$\lvert x_i \rvert \le 2.56$
Branin function	$F_4 = \left(x_2 - \frac{5.1}{4\pi^2} x_1^2 + \frac{5}{\pi} x_1 - 6 \right)^2 + 10\left(1 - \frac{1}{8\pi}\right)\cos x_1 + 10$	$\lvert x_i \rvert \le 5$
Griewangk's function	$F_5 = \frac{1}{4000}\sum_{i=1}^{30} x_i^2 - \prod_{i=1}^{30} \cos\left(\frac{x_i}{\sqrt{i}}\right) + 1$	$\lvert x_i \rvert \le 256$
DeJong's f2 function	$F_6 = 100(x_1^2 - x_2)^2 + (1 - x_1)^2$	$\lvert x_i \rvert \le 2.56$
Six hump camel back function	$F_7 = 4x_1^2 - 2.1x_1^4 + x_1^6/3 + x_1 x_2 - 4x_2^2 + 4x_2^4$	$\lvert x_i \rvert \le 5$

Among the functions under investigation, F1, F2 and F3 are continuous and single peaked. In other words, there exists only one best solution for them. However, F4 is continuous and multi-peaked which means it has several optimal solutions. The aforementioned four functions are used for investigation of proposed algorithm's searching ability. Functions F5 and F6 are nonlinear and have several minimums. F5 has a local minimum in $x_i = \pm k\pi\sqrt{i}$, $i = 1, 2, ..., 256,$ and $k = 1, 2, ..., 256$

While F6 has also several minimum points in proximity of $x_1 = x_2 = 1$. The values of these functions will not be linearly changed with the change of value of x. Therefore, these functions are used for investigation of ability of the proposed algorithm in terms of avoiding being trapped in local minimum points and getting free from early convergence. F7 is a special non-linear function having several minimum points. Not only its values are small, but also they are close to each other. In addition, this function has an infinite number of local minimum points which can easily impede the process of searching. The points of (0.089842, -0.712656) and (-0.089842, 0.712656) are the universal minimum points of this function. In this regard, the function of F7 will have the value of -1.0316285. Since finding an optimal solution for this function is extremely hard, it has been used for testing the algorithm's ability for getting close to the universal optimal point and testing the reliability of performance of the algorithm as well. The figure of the function is shown in diagram 10.

In order to make the proposed algorithm comparable with the SA and CSA algorithms, tests similar to those administered for the aforementioned algorithms were designed and developed for the proposed algorithm. The mutual characteristics of these experiments included programming of the algorithm through the MATLAB software, 100 algorithm repetitions for each function with a computer device equipped with a 3.0 GHz CPU and 1Gigabytes of Ram. The value of algorithm's parameters included K=300 and n1= 10 and n2= 20. Results of experiment are shown in table 4 along with the results obtained by Lou, Yoan and Xang.

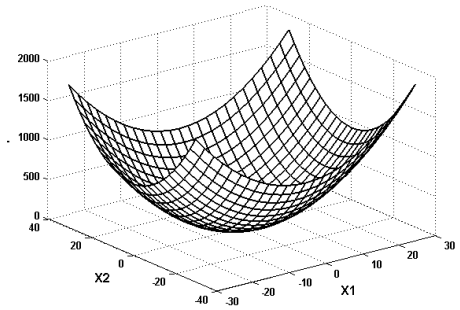

Fig 4, the F1 function with assumption of 3 variables

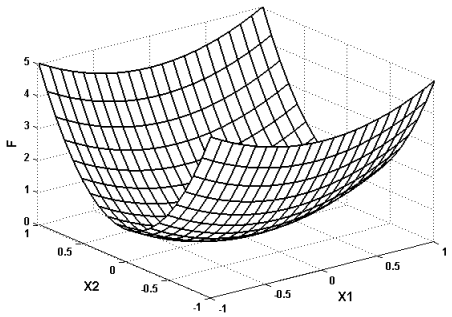

Fig 5, the F2 function with assumption of 3 variables

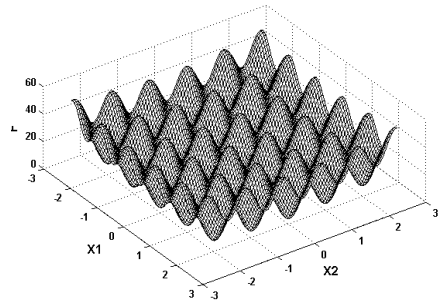

Fig 6, the F3 function with assumption of 3 variables

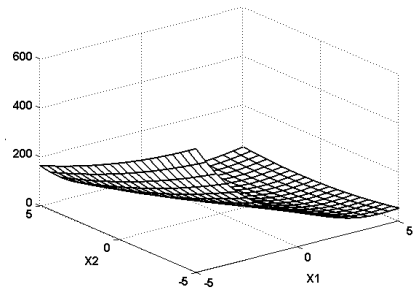

Fig 7, the F4 function

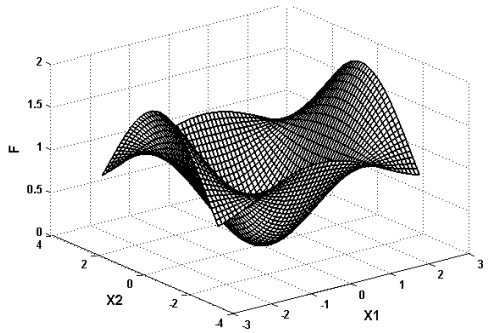

Fig 8, the F5 function with assumption of 3 variables

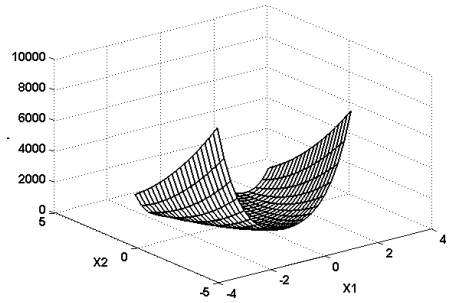

Fig 9, the F6 function

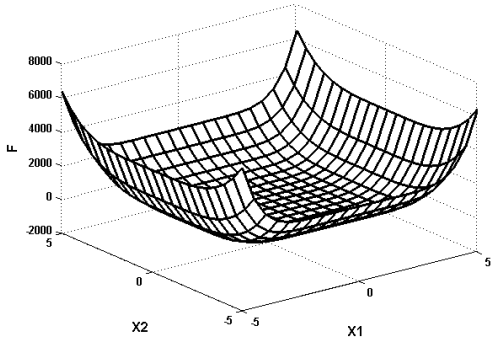

Fig 10, the F7 function

As you can see, in terms of precision and speed, the proposed algorithm is more efficient than the two previously developed algorithms. In order to perform a statistical analysis on results, the main hypothesis of the study has been stated as follows:

The performance of the proposed algorithm is better than CSA and SA algorithms. In addition, subsidiary hypotheses related to the efficiency of the proposed algorithm are as follows:

1- it gives a better solution for the F1 function
2- it gives a better solution for the F2 function
3- it gives a better solution for the F3 function
4- it gives a better solution for the F4 function
5- it gives a better solution for the F5 function
6- it gives a better solution for the F6 function
7- it gives a better solution for the F7 function
8- it consumes less time for finding a solution for F1 function

9- it consumes less time for finding a solution for F2 function
10- it consumes less time for finding a solution for F3 function
11- it consumes less time for finding a solution for F4 function
12- it consumes less time for finding a solution for F5 function
13- it consumes less time for finding a solution for F6 function
14- it consumes less time for finding a solution for F7 function

In order to test the subsidiary hypotheses, the t-test was used for comparison of means of the two populations with ambiguous variances. The first population was consisted of the entire results yielded by the CSA algorithm and the second population was consisted of the results yielded by the proposed algorithm. By performing the necessary and related statistical tests through the Excel 2007 software, certain results were obtained which are shown in table 3 in summary.

Table 3, results of statistical tests

Subsidiary hypotheses	P-Value	results	Subsidiary hypotheses	P-Value	results
1	3.44E-69	positive	8	1.92E-117	positive
2	6.29E-31	positive	9	2.33E-119	positive
3	1.07E-60	positive	10	1.00E+00	negative
4	1.47E-09	positive	11	8.00E-195	positive
5	5.02E-20	positive	12	8.92E-156	positive
6	3.86E-14	positive	13	3.97E-137	positive
7	7.17E-28	positive	14	4.11E-148	positive

As you can see, expect for the hypothesis number 10, the rest of hypotheses have been entirely accepted at a confidence level higher than 99%. Hypothesis number 10 is related to the time efficiency of the proposed algorithm in function F3 and this hypothesis can also be verified and accepted through making certain changes and adjustments in parameters. On the other hand, Wang, Linn and Huang (2010) carried out a study named as "solving the problem of six humped camel through the evolutionary thermodynamic algorithm in order to solve function F7. They adjusted their algorithm in a way that 100 solutions were produced in each loop and each loop was repeated 15 times. Their results are shown in table 4.

Table 4, results of evolutionary thermodynamic algorithm

F-min-val	X1	X2	Step
-1.031627816006554	-0.089933	0.7123900	1997
-1.031628310766583	0.0897910	-0.712526	2197
-1.031628417907029	-0.089903	0.7127110	2155
-1.031628453473775	0.0898420	-0.712655	2289
-1.031628452765956	-0.089842	0.7126470	1847
-1.031621705742450	0.0910380	-0.712344	2539
-1.031628453371039	-0.089846	0.7126540	5000
-1.031628445575145	0.0898690	-0.712683	2399
-1.031628115974580	-0.090134	0.7127050	1885
-1.031628443823104	0.0898920	-0.712659	2152

By performing certain analyses, it can be concluded that the proposed algorithm in this paper is not significantly different than the algorithm proposed by Wang, Linn and Huang in terms of producing a near-optimal solution. Table 5, shows the details related to this comparison. It should be noted that Wang, Linn and Huang have not mentioned the consumed amount of time for production of solution. However, with respect to existing evidence, it seems that their algorithm consumes more time compared with the proposed algorithm.

Table 5, comparison of models

	Best value	Worst value	Average value	values Standard
Wang, Linn and Hunag's algorithm	-1.031628453	-1.031621706	-1.031627662	2.10297E-06
Proposed algorithm	-1.031628453	-1.031623874	-1.031627917	7.5611E-07

Major factors for determining the efficiency of meta-heuristic algorithms are convergence diagrams by the use of which, meta-heuristics experts are able to make judgments regarding the performance of algorithm in terms of early convergence or adequate number of repetitions. Diagrams 11 and 12 are two samples of convergence diagrams of the proposed method. Since the initial solutions of algorithm are selected randomly from the solution space, the bests of these solutions usually show the value of purpose function as an undesirably high value. This fact causes the convergence diagram to initiate from a high value and overall, it results in significant increase of the scale of diagram in vertical plane in a way that optimizations in the next step are viewed too small that cannot be detected on the diagram.

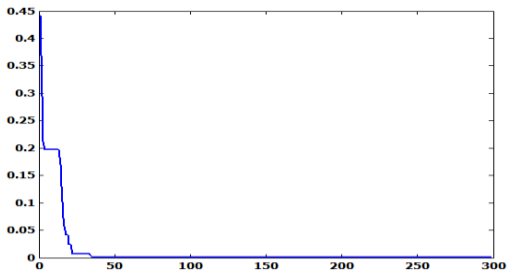

Fig 11, algorithm's convergence diagram for F6 function

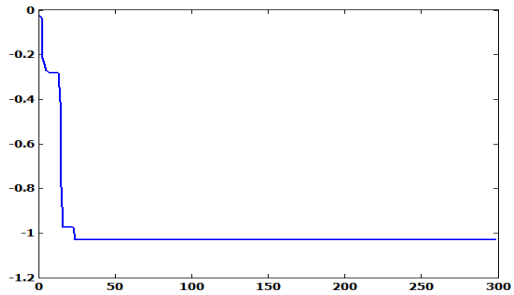

Fig 12, algorithm's convergence diagram for F7 function

VII. Conclusions and Suggestions

Our proposed algorithm includes the following advantages:
Speed in execution, simplicity of implementation and closeness of ultimate solutions to universal optimal points. Running the proposed algorithm for seven famous and well-known mathematical functions indicated that compared to simulated annealing and cloud based simulated annealing algorithms, the proposed algorithm as a complete superiority. Some of the advantages of the proposed meta-heuristic method include:

- simplicity of algorithm structure
- high speed in execution of algorithm since the execution algorithm includes only one loop
- ability to avoid local optimal points
- ability for extensive searching of the solution space
- reliability of algorithm
- having underlying convergence mechanisms for solution production

The current algorithm can also be used for other continuous functions and the results can be compared with those obtained by other algorithms. This algorithm is also able to be developed for discrete problems. One of the most extensive contexts for other researchers to work on can be implementation of this algorithm in terms of problems related to optimization of actual combinations since it seems that previous problems which have been solved by previous algorithms can have better solutions if solved with this algorithm.

VIII. References

[1] Blum, C., & Roli, A. (2003). Metaheuristics in combinatorial optimization: Overview and conceptual comparison. *ACM Comput. Surv., 35*(3), 268-308. doi: 10.1145/937503.937505
[2] Blum, C., & Roli, A. (2008). Hybrid Metaheuristics: An Introduction. In C. Blum, M. J. e. B. Aguilera, A. Roli & M. Sampels (Eds.), *Hybrid Metaheuristics* (pp. 1-30). Berlin: Springer-Verlag.
[3] Borenstein, Y., & Poli, R. (2006). *Structure and metaheuristics*. Paper presented at the Proceedings of the 8th annual conference on Genetic and evolutionary computation, Seattle, Washington, USA.
[4] Burke, E. K., & Kendall, G. (2005). Introduction. In E. K. Burke & G. Kendall (Eds.), Search Methodologies: *Introductory Tutorials in Optimization and Decision Support Techniques* (pp. 5-18). New York: Springer.
[5] Ferland, J. A., Hertz, A., & Lavoie, A. (1996). An Object-Oriented Methodology for Solving Assignment-Type Problems with Neighborhood Search Techniques. *Operations Research, 44*(2), 347-359.
[6] Gonzalez, T. F. (2007). Basic Methodologies. In T. F. Gonzalez (Ed.), *Handbook of Approximation Algorithms and Metaheuristics* (pp. 1.1-1.17). Boca Raton, FL, USA: Chapman and Hall/CRC.
[7] Hendrix, E. M. T., & G.-Tóth, B. (2010). Goodness of optimization algorithms *Introduction to Nonlinear and Global Optimization* (Vol. 37, pp. 67-90): Springer New York.
[8] Liu, J. (1999). The impact of neighbourhood size on the process of simulated annealing: Computational experiments on the flowshop scheduling problem. [doi: DOI: 10.1016/S0360-8352(99)00075-3]. *Computers & Industrial Engineering, 37*(1-2), 285-288.
[9] LV, P., Yuan, L., & Zhang, J. (2009). Cloud theory-based simulated annealing algorithm and application. *Engineering Applications of Artificial Intelligence, 22*, 742–749.
[10] Metaheuristics. (2011), 2011, from www.metaheuristic.com/metaheuristic_optimization.php
[11] Talibi, E.-G. (2009). METAHEURISTICS: FROM DESIGN TO IMPLEMENTATION
[12] Wang, X. (2010). Solving Six-Hump Camel Back Function Optimization Problem by Using Thermodynamics Evolutionary Algorithm.
[13] Weise, T. (2007). Global Optimization Algorithm: Theory and Application
[14] Xin, Y. (1991). Simulated Annealing with Extended Neighborhood. *International Journal of Computer Mathematics, 40*(3-4), 169-189.