Available online at www.sciencedirect.com

ScienceDirect

Procedia Computer Science 124 (2017) 151–157

Procedia Computer Science

www.elsevier.com/locate/procedia

4th Information Systems International Conference 2017, ISICO 2017, 6-8 November 2017, Bali, Indonesia

Killer Whale Algorithm: An Algorithm Inspired by the Life of Killer Whale

Totok R. Biyanto[a,*], Matradji[a], Sonny Irawan[c], Henokh Y. Febrianto[d], Naindar Afdanny[a], Ahmad H. Rahman[a], Kevin S. Gunawan[a], Januar A. D. Pratama[a], Titania N. Bethiana[b]

[a]*Department of Engineering Physics, Sepuluh Nopember Institute of Technology, Surabaya 60111, Indonesia*
[b]*Department of Chemical Engineering, Sepuluh Nopember Institute of Technology, Surabaya 60111, Indonesia*
[c]*Department of Petroleum Engineering, Universiti Teknologi PETRONAS, Malaysia*
[d]*Department of Industrial Engineering, Pusan National University, South Korea*

Abstract

This paper proposed a new algorithm inspired by the life of Killer Whale. A group of Killer Whale called *Matriline* that consist of a leader and members. The leader's duty searches prey position and the optimum direction to chase the prey, meanwhile chase the prey is performed by the members. Optimum direction means minimum direction and maximum velocity. Global optimum is obtained by comparing the results of member's actions. In this algorithm, if value of objective function of members more than leader, hence the leader must find out another new potential prey. In order to obtain the performance of proposed algorithm, it is necessary to test the new algorithm together with other algorithm using known mathematical function that available in Comparing Continuous Optimizers (COCO) especially Black Box Optimization Benchmarking (BBOB). Optimization results show that the performances of purposed algorithm has outperformed than others algorithms such as Genetic Algorithm (GA), Imperialist Competitive Algorithm (ICA) and Simulated Annealing (SA).

Peer-review under responsibility of the scientific committee of the 4th Information Systems International Conference 2017.

Keywords: Optimization; Killer Whale; Algorithm; Benchmark.

* Corresponding author. Tel.: +62-822-307-18944
E-mail address: trb@ep.its.ac.id

1877-0509 © 2018 The Authors. Published by Elsevier B.V.
Peer-review under responsibility of the scientific committee of the 4th Information Systems International Conference 2017
10.1016/j.procs.2017.12.141

1. Introduction

Global optimization fundamentally is a method to find out the solution of optimization problems that contain many local optimum, in many areas of applications [1]. In order to solve optimization problem, many stochastic algorithms have been purposed such as GA [2], ICA [3] and SA [4]. Class of stochastic optimization techniques can be classified as pure random algorithm such as GA, SA, Particle Swarm Optimization (PSO) [5], Duelist Algorithm (DA) [6], Rain Water Algorithm (RWA) [7], Whale Optimization Algorithm (WOA) [8]; and organized stochastic algorithms such as ICA [3], Grey Wolf Optimizer (GWO) [9], Ant Colony Optimization [10] and Artificial Bee Colony (ABC) [11]. None of algorithms in the literature has capability to memorize pattern of optimization results.

In this paper, a new algorithm based on the life of the Killer Whale was proposed. The basic philosophy of algorithm is the movement patterns of Killer Whale in pursuit of prey and social structure of Killer whale. Moreover, the novelty of this algorithm is incorporating memorize capability of Killer Whale in the proposed algorithm [12]. The entire algorithms have own advantages and disadvantages. The relationship between the levels of complexities of the algorithm with the time consumptions to solve optimization problems are proportional. This phenomena usually called No Free Lunch (NFL) theorem [13]. As mentioned before, incorporating memorize capability in the algorithm will affect to time consumptions per iteration, however this algorithm require less iteration. This feature is useful to solve very complex optimization problem or huge optimization variables.

2. Killer whale algorithm

This section will discuss the proposed optimization algorithm based on mimics the leadership hierarchy and hunting mechanism of Killer Whale as well as memorize capability for each *Matriline* in mathematical model.

2.1. Inspiration

Killer Whales (*Orcinus Orca*) is the marine mammals as the highest peak of the food chain in marine ecological system or the apex marine predator. Killer Whale has three kinds of body morphological forms, such as type A, type B and type C, which type A has largest body shape than the other types of Killer Whales [14].

Basically, Killer Whales as apex marine predator are classified into two types of specialization depend on hunting patterns i.e. Fish-Feeding Residents and Mammal-Hunting Transients. Fish-Feeding Residents is Killer Whale with hunting pattern in the same area, meanwhile, the Mammal-Hunting Transients will hunt follow the prey migration season. The prey scanning is carried out using echolocation vocalizations. Killer Whales have 3 types of sounds namely Clicks, Whistles and Pulsed Calls [14].

2.2. Implementation into the form of algorithms based on inspiration

This section will discuss implementation of hutting pattern of Killer Whale in mathematical model.

- Foraging Geometry

According to [12], in order to implement of echolocation of Killer Whale, the mathematical model that represents Killer Whales in foraging of prey are utilized as search movement-agents to find best solution of objective function.

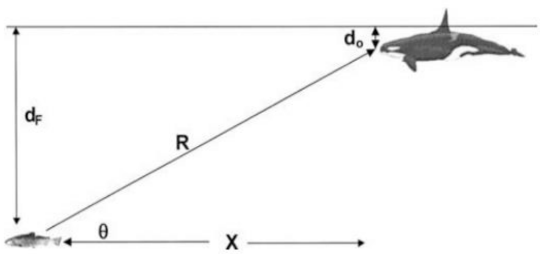

Fig. 1. Foraging Geometry of Killer Whale at depth d_o in pursuit of a Prey at range R and a depth d_F [12].

The following notations are used in Fig. 1: d_F and d_o are the depth of the prey and the Killer Whale's sonar, respectively. R is the slant range between the prey and the Killer Whale, X is the horizontal range, and θ is the angle between the slant and horizontal range [12]. The angle θ can be determined by the equation.

$$\theta = \sin^{-1}\left(\frac{d_F - d_O}{R}\right) = \tan^{-1}\left(\frac{d_F - d_O}{x}\right) \tag{1}$$

- Velocity of Movement

Every search agents require a velocity to move from recent location to prey location, in term of magnitude and direction of movement. Therefore, the mathematical model to represent the movement of Killer Whale to reach prey can be formulated as follows:

$$\begin{cases} \vec{v_i} \leftarrow \vec{v_i} + \vec{U}(0,\emptyset_1)\otimes(\vec{p_i} - \vec{x_i}) + \vec{U}(0,\emptyset_2)\otimes(\vec{p_g} - \vec{x_i}), \\ \vec{x_i} \leftarrow \vec{x_i} + \vec{v_i}\cdot t \end{cases} \tag{2}$$

Each search agents consist of three D-dimensional vectors, where D is the dimensionality of the search space that contain of the current position $\vec{x_i}$, the previous best position $\vec{p_i}$, the velocity $\vec{v_i}$ and the time t. The current position $\vec{x_i}$ can be considered as a set of coordinates describing a point in space. On each iteration of the algorithm, the current position is evaluated as a problem solution. If the current position better than previous position, therefore the current position will represent as previous position in next iteration $\vec{p_i}$. The value of the best objective function will represent as best position for previous iteration $pbest_i$ and it will compare with the next iteration value of objective functions to obtain better position that will be stored as $\vec{p_i}$ and $pbest_i$ at the next iterations. New positions are calculated by changing $\vec{v_i}$.

- Cluster for Search Space

The optimization problem consist of N-dimensional space and $[-x,x]$ boundary. Optimization techniques will be assigned to find out the optimized variables within $-x$ to x for each N. In Killer Whale Algorithm, as group of search agents, the range of optimization can be clustered become some range based on number of search agents. Each search agents will seek local optimum value within the cluster from the centroid of the cluster. Clustering process in this algorithm requires matrix input of M points and K initial cluster centres in N dimensions. Number of points in cluster L is denoted by $NC(L)$, $D(I,L)$ is the Euclidean distance between point I and cluster L, the general procedure is to search for a K-partition with locally optimal within-cluster sum of squares by moving points from one cluster to another. The clustering process can be adopted using method available in [15] and can be explained as follows:

Step 1. Each point I (I = 1,2,…,M), identified the closest and Second closest cluster centres $IC1(I)$ and $IC2(I)$ respectively. Denote point I to cluster $IC1(I)$.

Step 2. Update the centre of clusters to be the mean of points contained within them.

Step 3. All clusters be appropriate to the live set, initially.

Step 4. Utilize the optimal-transfer (OPTRA) stage:
Consider each point I (I = 1,2,…,M) in turn. If cluster L (L = 1,2,…,K) is updated in the last quick-transfer (*QTRAN*) stage, therefore, it belongs to the live set throughout current stage. Otherwise, it is not include in the live set if there are no updated in the last *M OPTRA* steps. Let point I be in cluster $L1$. If $L1$ include in the live set, perform Step 4a; otherwise, carry out Step 4b.

Step 4a. Calculate the quantity of minimum, $R2 = [NC(L) * D(I,L)^2] / [NC(L)+1]$, overall clusters $L(L \neq L1, L = 1,2,\ldots,K)$. Let $L2$ clustered with the smallest $R2$. If this value greater than or equal to $[NC(L1) * D(I,L1)^2] / [NC(L1)-1]$, reallocation is not required and $L2$ is the new $IC2(I)$. (Should be noted that the value $[NC(L1) * D(I,L1)^2] / [NC(L1)-1]$ is memorized and will save at the same value for point I until cluster $L1$ is renewed). Otherwise, point I is assigned to cluster $L2$ and $L1$ is the updated $IC2(I)$. Centre of clusters are renewed to be the averages of points assigned to them if restructuring has taken place. The two involved clusters in the transfer of point I at this particular step are now in the live set.

Step 4b. This step is similar as Step 4a, except that the minimum *R2* is calculated over cluster in the live set lonely.

Step 5. Break if the live set is blank. Otherwise, continue to Step 6, after a single pass through the data set.

Step 6. Perform the QTRAN stage:
Take each point I (I = 1,2,…,M) in turn. Put $L1 = IC1(I)$ and $L2 = IC2(I)$. It is not required to check the point I if both the clusters $L1$ and $L2$ have same values in the last M steps. Calculate the values $R1 = [NC(L1) * D(I,L1)^2] / [NC(L1)\text{-}1]$ and $R2 = [NC(L2) * D(I,L2)^2] / [NC(L2)+1]$. (As mention before, $R1$ is memorize and will save the same values until cluster $L1$ is renewed). If $R1$ less than $R2$, point I rests in cluster $L1$. Otherwise, over change $IC1(I)$ and $IC2(I)$ and renew the centre of clusters $L1$ and $L2$. The two area of clusters are recognized for their involvement in a relocation at this step [15].

- Flowchart of Algorithms

The initialization process is performed to determine the initial parameters such as the number of *Matriline*, dimensions of the objective function, the global optimum which is the minimum or maximum value, lower bound and upper bound of the optimized variables, number of the cluster as well as the number of iterations for clustering process. The number of *Matriline* will be classified as a leader and members.

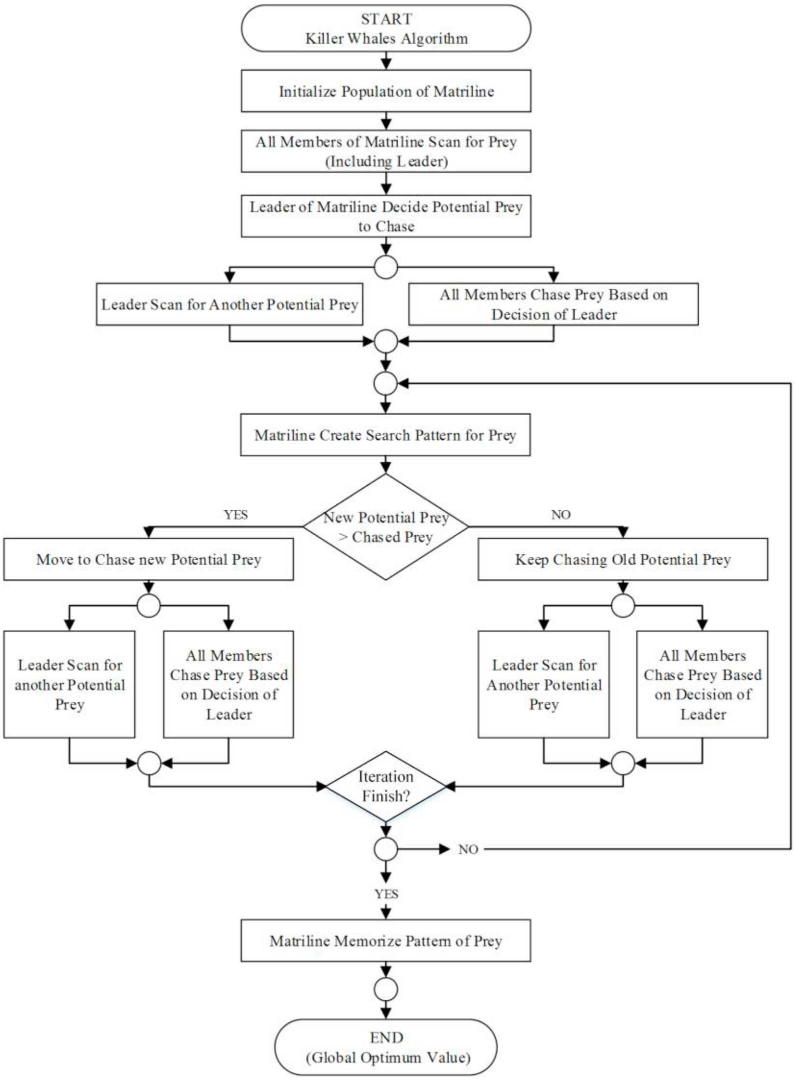

Fig. 2. Killer Whale Algorithm Flowchart

Clustering method is utilized to obtain the global optimum value of objective function rapidly and avoid fall in the local optimum value. The search area of each cluster will be traced by leader and members, searching method begins at the point of centroid of each cluster. Leader will move to the other side of cluster if the value of objective function lower than the members. If leader shift to another cluster, the members will move to a new cluster. This process is performed using the principle of the number of clusters divided with the number of iterations. For example, there are 4 clusters with 20 iterations for search agents, therefore each cluster will carry out 5 iterations.

3. Experimental results

The performance of Killer Whale Algorithm can be measured using some real-parameter in COCO especially BBOB. BBOB is a standard function to examine a new algorithm proposed by N. Hansen, et al. [16]. In this paper used 6 benchmark functions with noise. The purpose of benchmark is to prove the capability of proposed algorithm to solve optimization problem. Furthermore, the performance of proposed algorithm are compared to the famous algorithms such as GA [2], ICA [3] and SA [4]. The experiment results exhibit that Killer Whale Algorithm is able to reach global optimum under lesser number of iterations than GA, ICA and SA (Fig. 3).

3.1. Noisy benchmark function

The purpose of benchmark using noisy function is to examine the algorithms to find out global optimum value in known value of objective functions. Gaussian noise (F_{GN}), Uniform noise (F_{UN}), and Seldom Cauchy noise (F_{CN}) were utilized in the benchmark functions as shown in Table. 1 [16].

Table 1. Noisy Benchmark Function

Fun.	Test Problem	Objective Function	Search Range	Dim.
f_1	Sphere with moderate Gaussian noise	$f_1(x) = f_{GN}\left(\sum_{i=1}^{D} x_i^2, 0.01\right)$	[-5 to 5]	5, 20
f_2	Sphere with moderate Uniform noise	$f_2(x) = f_{UN}\left(\sum_{i=1}^{D} x_i^2, 0.01\left(0.49 + \frac{1}{D}\right), 0.01\right)$	[-5 to 5]	5, 20
f_3	Sphere with moderate Seldom Cauchy noise	$f_3(x) = f_{CN}\left(\sum_{i=1}^{D} x_i^2, 0.01, 0.05\right)$	[-5 to 5]	5, 20
f_4	Rosenbrock with moderate Gaussian noise	$f_4(x) = f_{GN}\left(\sum_{i=1}^{D-1} (100(x^2 - x_{i+1})^2 + (x_i - 1)^2), 0.01\right)$	[-5 to 5]	5, 20
f_5	Rosenbrock with moderate Uniform noise	$f_5(x) = f_{UN}\left(\sum_{i=1}^{D} (100(x^2 - x_{i+1})^2 + (x_i - 1)^2), 0.01\left(0.49 + \frac{1}{D}\right), 0.01\right)$	[-5 to 5]	5, 20
f_6	Rosenbrock with moderate Seldom Cauchy noise	$f_6(x) = f_{CN}\left(\sum_{i=1}^{D} (100(x^2 - x_{i+1})^2 + (x_i - 1)^2), 0.01, 0.05\right)$	[-5 to 5]	5, 20

3.2. Parameter setting

Parameter settings that used in initialization step for all algorithms as follows. The population and iterations that used in GA are 100 and 100, respectively. The GA's mutation and crossover probabilities are set at 0.05 and 0.8. The number of colonies in ICA is set at 100, with initial number of imperialists of 8, revolution rate of 0.4 and number of decades of 100. In SA, the number of cooling factor is 0.8, the number of Boltzmann constant is 1 and the number of maximum rejections is 100. Parameter setting of Killer Whale Algorithm are 100 population, 10 leaders and 100 iterations.

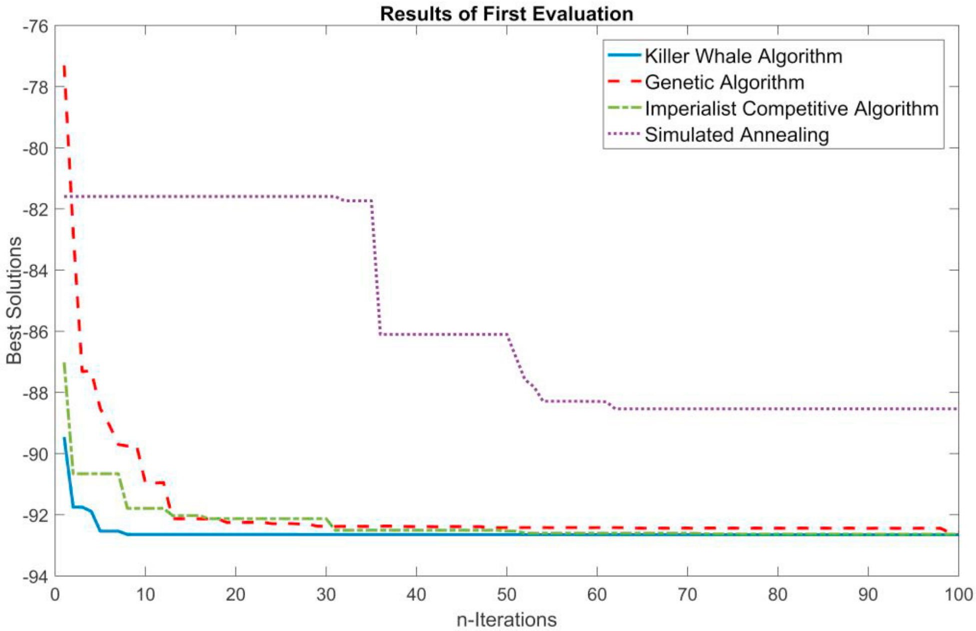

Fig. 3. Comparison of best solutions between algorithms

3.3. Results of benchmark using 5 and 20 dimensions

The statistical results of benchmark of the algorithms using 5 and 20 dimensions are tabulated in Tables 2 and 3. The optimization results show that Killer Whale Algorithm is able to solve optimization problem with characteristics noisy at low or high dimensions of optimized variables. In general, the proposed method provide robust performance and able to surpass the performance of three compared algorithms.

Table 2. Results in 5 Dimension

Fun		GA	ICA	SA	KWA
f_1	Min	-92.5800	-92.6369	-88.4881	**-92.6500**
f_2	Min	-92.6185	-92.5763	-88.4136	**-92.6500**
f_3	Min	-82.4908	-82.6480	**-88.0047**	-85.7252
f_4	Min	848.0670	-129.8190	2.2091e+04	**-134.6677**
f_5	Min	276.3204	-116.6629	2.2556e+04	**-134.3457**
f_6	Min	231.7008	-113.7259	2.2130e+04	**-125.1652**

Table 3. Results in 20 Dimension

Fun		GA	ICA	SA	KWA
f_1	Min	-12.6199	-52.2025	-63.8754	**-92.6495**
f_2	Min	-11.5006	-58.3282	-88.0677	**-92.6498**
f_3	Min	-10.8461	-49.0257	-78.1589	**-82.7811**
f_4	Min	4.3482e+04	1.2555e+04	2.2310e+04	**-119.2875**
f_5	Min	5.4096e+04	1.3248e+04	5.1639e+04	**-117.7796**
f_6	Min	4.6363e+04	2.0891e+4	2.2145e+04	**-107.6472**

4. Conclusion

In this paper, a new algorithm inspired by the life of killer whale was proposed. A group of Killer Whale called *Matriline* that consist of a leader and members. The leader's duty searches prey position and the optimum direction to chase the prey, meanwhile prey hunting is performed by the members. Global optimum is obtained by comparing the results of member's actions. Testing was performed using the 6 noisy benchmark function that contain Gaussian noise, Uniform noise and Seldom Cauchy noise. The optimization results show the performance of proposed algorithm surpass others compared algorithms and robust.

Acknowledgements

The authors gratefully thank to Sepuluh Nopember Institute of Technology (ITS) Surabaya and Lembaga Penelitian dan Pengabdian kepada Masyarakat ITS (LPPM ITS) for providing the facilities for conducting this research.

References

[1] T. Back, Evolutionary Algorithms in Theory and Practice: Evolution Strategies, Evolutionary Programming, Genetic Algorithms, New York: Oxford University Press, 1996.

[2] D. Goldberg, Genetic Algorithms in Search Optimization and Machine Learning, New York: Addison-Wesley, 1989.

[3] E. A. Gargari and C. Lucas, "Imperialist competitive algorithm: an algorithm for optimization inspired by imperialistic competition," in *IEEE Congress on Evolutionary Computation*, Singapore, 2007 .

[4] S. Kirkpatrick, "Optimization by simulated annealing: Quantitative studies," *Journal of Statistical Physics,* vol. 34, no. 5–6, p. 975–986, 1984.

[5] J. Kennedy and R. Eberhart, "Particle Swarm Optimization," in *EEE International Conference on Neural Networks,* Path, Australia, 1995.

[6] T. R. Biyanto, H. Y. Fibrianto, G. Nugroho, A. M. Hatta, E. Listijorini, T. Budiati, and H. Huda, "Duelist Algorithm: An Algorithm Inspired by How Duelist Improve Their Capabilities in a Duel," in *International Conference in Swarm Intelligence*, Bali, Indonesia, 2016.

[7] T. R. Biyanto, M. N. Syamsi, H. Y. Fibrianto, N. Afdanny, K. S. Gunawan, A. H. Rahman, J. A. D. Pratama and A. I. Abdillah, "Optimization of Energy Efficiency and Conservation in Green Building Design Using Duelist, Killer-Whale and Rain-Water Algorithms," in *International Conference of Applied Science and Technology for Infrastructure Engineering*, Surabaya, Indonesia, 2017.

[8] S. Mirjalili and A. Lewis, "The Whale Optimization Algorithm," *Advances in Engineering Software,* vol. 95, p. 51–67, 2016.

[9] S. Mirjalili, S. M. Mirjalili and A. Lewis, "Grey Wolf Optimizer," *Advances in Engineering Software,* vol. 69, pp. 46-61, 2014.

[10] M. Dorigo, M. Birattari and T. Stutzle, "Ant colony optimization," *IEEE Computational Intelligence ,* vol. 1, no. 4, pp. 28-39, 2006.

[11] D. Karaboga and B. Basturk, "A powerful and efficient algorithm for numerical function optimization: artificial bee colony (ABC) algorithm," *Journal of Global Optimization,* vol. 39, no. 3, p. 459–471, 2007.

[12] Whitlow W. L. Au, et al., "Echolocation signals of free-ranging Killer whales (Orcinus orca) and modeling of foraging for Chinook salmon (Oncorhynchus tshawytscha)," *The Journal of the Acoustical Society of America,* pp. 901-909, 2004.

[13] D. H. Wolpert and W. G. Macready, "No free lunch theorems for optimization," *IEEE Transactions on Evolutionary Computation,* vol. 1, no. 1, pp. 67-82, 1997 .

[14] W. F. Perrin and B. Wursig, Encyclopedia of marine mammals, Academic Press, 2009.

[15] J. A. Hartigan and M. A. Wong, "Algorithm AS 136: A k-means clustering algorithm," *Journal of the Royal Statistical Society, Series C (Applied Statistics),* vol. 28, pp. 100-108, 1979.

[16] N. Hansen, et al., "Real-parameter black-box optimization benchmarking 2010: Experimental setup," Institut National de Recherche en Informatique et Automatique, Paris, 2010.