*International Journal of Science and Advanced Technology (ISSN 2221-8386)* Volume 4 No 9 September 2014
http://www.ijsat.com

# A Modified Optimization Algorithm Inspired by Wild Dog Packs

**Essam Al Daoud**
Computer Science Department, Zarqa University,
Zarqa, Jordan.
essamdz@zu.edu.jo

*Abstract*—**The difficulties associated with using mathematical optimization problems have contributed to the development of alternative solution[1]s. The formal methods often fail in solving NP-hard applications of the large size. To overcome this problem, several metaheuristic algorithm have been suggested. In this paper a modified optimization algorithm inspired by wild dog packs is suggested, the new approach solved the drawback of the wild dog packs optimization by using a set of harmony search whenever a local minimum is detected. Benchmark comparisons among the algorithms are presented for eight functions. The suggested algorithm outperforms various other mataheuristics algorithms for several configurations and dimensions.**

***Keywords- harmoney search; local minima; metheuristic; wild dog packs.***

## I. INTRODUCTION

Optimization techniques are search methods, where the goal is to find a solution to the continuous optimization problems, the combinatorial problems and the NP-Hard problems in general. There are many real world optimization problems such as vehicle rescheduling problem, vehicle routing problem, weapon target assignment problem, integer programming, nurse scheduling problem, constraint satisfaction problem, assignment problem, closure problem, cutting stock problem, knapsack problem, linear programming and traveling salesman problem [1]. Because of their difficulty and enormous practical importance, a large number of solution techniques for attacking NP-hard integer and combinatorial optimization problems have been proposed. However many of the practical problems cannot be solved by using the formal optimization methods, therefore several meta-heuristic approaches have been used to solve the continuous and the discrete optimization problems, we cannot expect them to find the best solution all the time, but expect them to find the good enough solutions or even the optimal solution most of the time, and more importantly, in a reasonably and practically short time. Modern meta-heuristic algorithms are almost guaranteed to work well for a wide range of tough optimization problems such as tabu search (TS), simulated annealing (SA), genetic algorithm (GA), particle swarm optimization (PSO), artificial neural network (ANN), ant colony optimization (ACO), harmony search (HS) and wild dog optimization (WDO). Two major directions of any metaeuristic algorithms are: exploitation (intensification) and exploration (diversification). exploration means to try several solutions and explore the search space on the global scale, while exploitation means to exploit a good solution in a local region. On the other hand, metaheuristic algorithms can be divided into population based methods and trajectory based methods. Population based methods can identify a promising region very quickly, whereas trajectory methods are better at exploring a certain region [2, 3].

## II. LITERATURE REVIEW

Since the 1970s, many meta-heuristic algorithms that combine rules and randomness imitating natural phenomena have been developed to overcome the computational drawbacks of existing numerical algorithms (i.e., sensitivity to initial values, complex derivatives, and the large amount of enumeration memory) when solving difficult and complex engineering optimization problems. In 1983 IBM researchers introduced simulated annealing algorithm Based on the way nature performs an optimization of the energy of a crystalline solid when it is annealed to remove defects in the atomic arrangement .Simulated annealing uses the objective function of an optimization problem instead of the energy of a real material .Starting with an initial high temperature value, the annealing algorithm proceeds by choosing an adjustable solution parameter at random and changing its value by a random amount .The temperature and the amount of change in the objective function are then used to determine if the new solution is accepted. High temperatures allow solution to escape local minima. In 1986 Glover improved The Tabu search algorithm by keeping track of information and decisions used previously during the search. The information is used to guide the move from the current solution $i$ to the next solution $j$ . The role of the memory will be to restrict the choice to some subset by forbidding for instance moves to some neighbor solutions. Particle swarm optimization (PSO) was proposed by Kenned and Eberhart in 1995 [4] and modified by Shi and Eberhart [5]. In PSO, the velocity vector and the position vector are updated as follows:

$$v_i(t+1) = cv_i(t) + a_1 n_1(p_i - x_i(t)) + a_2 n_1(b - x_i(t)) \quad (1)$$

$$x_i(t+1) = x_i(t) + v_i(t+1) \quad (2)$$

where c, $a_1$, and $a_2$ are constants, $n_1$ and $n_2$ are uniform random numbers in the range [0,1], b is the global best position that is discovered by the whole swarm, and $p_i$ is

[1] This research is funded by the Deanship of Research and Graduate Studies in Zarqa University /Jordan

the best position of the i[th] particle [6, 7]. Ant colony optimization (ACO) is another metahubristic algorithm proposed by Marco Dorigo in 1992 [8]. ACO uses the heuristic information and the pheromone to guide the searching process. The probability with which ant *k*, currently at node *i* and the next node is *j*:

$$p_{ij}^{k} = \frac{\tau_{ij}^{\alpha}\eta_{ij}^{\beta}}{\sum_{l \in N_i^k} \tau_{il}^{\alpha}\eta_{il}^{\beta}} \quad (3)$$

The pheromone is updated as following

$$\tau_{ij} = \tau_{ij} + \sum_{k=1}^{m} \Delta\tau_{ij}^{k} \quad (4)$$

where

$$\Delta\tau_{ij}^{k} = \begin{cases} 1/C & \text{if the arc belongs to tour k} \\ 0 & \text{otherwise} \end{cases} \quad (5)$$

and $\alpha$ and $\beta$ are two parameters which determine the relative influence of the pheromone trail and the heuristic information, $\eta_{ij} = 1/d_{ij}$ is a heuristic value that. In 2001 Geem *et al.* [9] developed a New Harmony search (HS) meta-heuristic algorithm that was conceptualized using musical process of searching for a perfect state of harmony. This harmony in music is analogous to find the optimality in an optimization process. In music improvisation process musician plays different notes of different musical instrument and find the best combination of frequency for best tune. Similarly in HS method also best combination of available solutions is selected and objective function is optimized.

**Algorithm 1**: Harmony Search
Generate initial harmonics (real number arrays)
**while** ( t < *iter* )
    Generate new harmonics by accepting best harmonics
    Adjust pitch to get new harmonics (solutions)
    **if** (rand>*hm*),
        choose an existing harmonic randomly
    **else if** (rand>*pa*),
        adjust the pitch randomly within limits
    **else**
        generate new harmonics via randomization
    Accept the new harmonics (solutions) if better

Where *pa* is the pitch adjusting rate, and *hm* is the harmony memory accepting rate [10 -13]. In 2014 Al Daoud *et al.*[14] introduced a new efficient algorithm based on wild dog behavior. Wild dog optimization uses the self competing to find and update alpha. Algorithm 2 illustrates the wild dog optimization

**Algorithm 2**: Wild dog pack optimization (WDPO)
Generate n dogs randomly and choose the best as alpha
**while** ( t< *iter* )
    Evaluate dogs Locations
    If iteration % q = 0
        Update the Parameters using self competition
    Select the new dogs Locations
    Evaluate Dogs
    If no improvement for v iterations
        Use Hoo procedure to escape from the local minimum

## III. A Modified Wild Dog Algorithm

The main disadvantage of the original wild dog pack optimization is it does not balance well between the exploitation and the exploration strategies. Wild dog algorithm spends more time in exploitation than exploration. While the standard harmony search is more efficient in exploration than exploitation. Therefore the suggested algorithm combines the advantages of the both algorithms, once a local minimum is detected the algorithm will switch automatically to a new version of the harmony search with a suitable parameter to escape from the local minimum efficiently. Algorithm 3 is the modified Wild dog pack optimization (MWDPO).

**Algorithm 3**: Modified Wild dog pack optimization
Generate n dogs randomly and choose the best as alpha
**while** ( *t* < *iter* )
  if *flag* = 0
      Evaluate dogs Locations
      if iteration % *q* =0
          Update the Parameters using self competition
      Select the new dogs Locations
      Evaluate Dogs
      if no improvement for v iterations
          *flag*=1
  else
      if (rand < $tol_1$)
          Update a dog randomly
      else if (rand < $tol_2$)
          adjust the alpha dog by *seq*
      else
          exclude a dog randomly and initialize a new one
      after k sub-iterations switch *flag* to 0

In the suggested algorithm the set *seq* is a set of numbers such that the next number is equal to the previous number * 10. For each iteration the alpha dog is adjusted randomly according to the set *seq*.

## IV. Benchmark Problem

To compare the efficiency of the famous metaheuristic methods eight different functions are selected, which has several features such as difficulty, reparability, regularity, continuity, and multimodality. The selected functions can be extended to arbitrary dimensions *d* [15, 16]:

1- ***Sphere***:

$$f_1(x) = \sum_{i=1}^{d} x_i^2 \quad (6)$$

2- ***Rosenbrock***:

$$f_2(x) = \sum_{i=1}^{d-1} [100(x_{i+1} - x_i^2)^2 + (1 + x_i)^2] \quad (7)$$

3- ***Ackley***:

$$f_3(x) = -20\exp\left(-2.0\sqrt{\frac{1}{30}\sum_{i=1}^{d} x_i^2}\right) - \exp\left(\frac{1}{30}\sqrt{\sum_{i=1}^{d}\cos(2\pi x_i)}\right) + 20 + e \quad (8)$$

4- ***Griewank***:

$$f_5(x) = \frac{1}{4000}\sum_{i=1}^{d} x_i^2 - \prod_{i=1}^{d}\cos(\frac{x_i}{\sqrt{i}}) + 1 \quad (9)$$

5- ***Schwefel***:

$$f_6(x) = \sum_{i=1}^{d}|x_i| + \prod_{i=1}^{d}|x_i| \quad (10)$$

6- ***Step***:

$$f_4(x) = \sum_{i=1}^{d}( \lfloor x_i^2 + 0.5 \rfloor )^2 \quad (11)$$

7- ***Rotated hyper-ellipsoid***:

$$f_7(x) = \sum_{j=1}^{d}\sum_{i=1}^{j} x_i^2 \quad (12)$$

8- ***Rastrigin***:

$$f_8(x) = 10d + \sum_{i=1}^{d}(x_i^2 - 10\cos(2\pi x_i)) \quad (13)$$

## V. EXPERIMENTAL RESULTS

In this section, three algorithms are implemented and compared namely self-adaptive harmony search (SAHS) [12], wild dog pack optimization (WDPO) [14] and the suggested algorithm: modified wild dog pack optimization (MWDPO). The same parameters are used as suggested by [14], but the new algorithm parameters are $tol_1$=0.08, $tol_2$=0.99, *seq* ={3, 0.3, 0.03}, *k*=500 and *v*=100. All the results are averaging of 30 runs. The experiements are implemented for dimensions *d*=30 and *d*=100. The number of the function evaluation 500000 and 50000. Tables 1-4 show that the suggested algorithm is outperform all the previous algorithms form most of the functions, dimensions and the number of the evaluation.

TABLE I. TABLE TYPE STYLES COMPARES MWDPO, SAHS AND WDPO BY USING DIMENSION 30 AND 500000 FUNCTION EVALUATIONS

| function | SAHS | WDPO | MWDPO |
|---|---|---|---|
| **Rosenbrock** | 6.3643e-002<br>(1.9570e-002) | 4.2135e-028<br>(4.1298e-026) | **1.0442e-032**<br>(1.2763e-028) |
| **Sphere** | 7.6724e-012<br>(1.7946e-012) | 2.6963e-315<br>(7.6182e-316) | **3.8112-318**<br>(1.7243-317) |
| **Ackley** | 2.0308e-004<br>(5.7756e-005) | 1.5099e-014<br>(3.7810e-015) | **2.5643-020**<br>(1.1128-016) |
| **Griewank** | 4.2386e-010<br>(1.9681e-010) | **0**<br>(0) | **0**<br>(0) |
| **Schwefel** | 5.6850e-006<br>(2.6129e-006) | 4.8970e-139<br>(1.0203e-138) | **5.6116e-140**<br>(1.3002e-139) |
| **Step** | **0**<br>(0) | **0**<br>(0) | **0**<br>(0) |
| **Rotated-h-e** | 1.1240e+001<br>(8.0782e-000) | 9.5172e-320<br>(1.3780e-320) | **2.3410e-322**<br>(0.3241e-321) |
| **Rastrigin** | 7.1162e+000<br>(5.2029e-001) | 6.2341e+000<br>(4.9925e-000) | **3.0011e-012**<br>(1.3225e-011) |

TABLE II. TABLE TYPE STYLES COMPARES MWDPO, SAHS AND WDPO BY USING DIMENSION 100 AND 500000 FUNCTION EVALUATIONS

| function | SAHS | WDPO | MWDPO |
|---|---|---|---|
| **Rosenbrock** | 2.9130e+001<br>(7.4963 e+000) | 2.8220e+001<br>(3.8701e+000) | **7.5321e-003**<br>(1.3341e-001) |
| **Sphere** | 2.2626e-001<br>(9.5596e-002) | 6.4645e-306<br>(1.3992e-304) | **1.4415e-308**<br>(9.1294e-308) |
| **Ackley** | 8.2945e-003<br>(3.4291e-003) | 9.3259e-014<br>(7.2402e-013) | **1.0034e-016**<br>(1.3716e-014) |
| **Griewank** | 9.4160e-005<br>(1.1619e-005) | 1.1102e-016<br>(5.4497e-013) | **2.3981e-018**<br>(1.2981e-017) |
| **Schwefel** | 1.7854e-002<br>(3.2936e-003) | 1.3011e-058<br>(2.1560e-057) | **7.4351e-065**<br>(1.7863e-063) |
| **Step** | 4.1200e+001<br>(7.4370e+000) | 4.1333e+000<br>(1.0023e+000) | **6.0023e-003**<br>(9.8827e-004) |
| **Rotated-h-e** | 7.6734e+001<br>(9.7645e+000) | 7.2618e-045<br>(2.2514e-041) | **2.2640e-061**<br>(8.9023e-060) |
| **Rastrigin** | 3.9200e+001<br>(1.8324e+001) | 5.1810e+001<br>(8.4351e+000) | **1.3719e-003**<br>(9.7431e-003) |

TABLE III. TABLE TYPE STYLES COMPARES MWDPO, SAHS AND WDPO BY USING DIMENSION 30 AND 50000 FUNCTION EVALUATIONS

| function | SAHS | WDPO | MWDPO |
|---|---|---|---|
| **Rosenbrock** | 3.6953e+001<br>(8.9989 e+000) | 6.7795e+000<br>(6.8722e-001) | **2.8861e-001**<br>(7.8430e-001) |
| **Sphere** | 6.7442e+000<br>(1.2769e+000) | **2.7434e-037**<br>(6.0890e-036) | 1.0912e-036<br>(2.1145e-035) |
| **Ackley** | 1.8987e+000<br>(1.1000e+000) | 2.6544e-005<br>(7.9923e-006) | **4.8241e-007**<br>(9.7342e-007) |
| **Griewank** | 9.5488e-001<br>(1.2459e-001) | 1.3870e-014<br>(8.3260e-013) | **3.5100e-020**<br>(9.2118e-018) |
| **Schwefel** | 1.0534e-001<br>(4.8074e-002) | 2.4618e-014<br>(1.8713e-014) | **4.5618e-022**<br>(6.7545e-021) |
| **Step** | 1.6000e-001<br>(5.7910e-002) | **0**<br>(0) | **0**<br>(0) |
| **Rotated-h-e** | 1.1002e+002<br>(4.2402e+001) | 5.7546e-024<br>(5.2541e-022) | **1.4320e-028**<br>(3.2124e-022) |
| **Rastrigin** | 7.6706e+001<br>(8.0748e+000) | 1.1369e+002<br>(4.5061e+001) | **1.5462e-000**<br>(3.5170e-000) |

TABLE IV. TABLE TYPE STYLES COMPARES MWDPO, SAHS AND WDPO BY USING DIMENSION 100 AND 50000 FUNCTION EVALUATIONS

| function | SAHS | WDPO | MWDPO |
|---|---|---|---|
| **Rosenbrock** | 8.5773e+001 (4.8637e+000) | 8.5009e+001 (9.4571e+00) | **6.3580e-000** (3.422e-000) |
| **Sphere** | 5.0306e+001 (1.9145e+001) | **1.5448e-032** (4.018e-031) | 3.2101e-031 (8.122e-030) |
| **Ackley** | 1.4494e+001 (9.8972e-001) | 1.5230e+001 (1.892e+000) | **7.3412e-001** (9.201e-000) |
| **Griewank** | 7.6855e+000 (1.5228e+000) | 2.8208e-05 (6.752e-005) | **1.0021e-06** (2.4512e-005) |
| **Schwefel** | 3.2853e+001 (4.9568e+000) | 7.7974e-001 1.9016e-001 | **2.1348e-008** 3.4545e-007 |
| **Step** | 3.7633e+002 (2.7528e+001) | 1.6333e+001 (2.865e+000) | **5.6202e-001** (1.3310e-001) |
| **Rotated-h-e** | 7.6699e+003 (5.6183e+002) | 3.8721e+001 (5.963e+000) | **8.6591e-003** (0.4461e-003) |
| **Rastrigin** | 7.9321e+002 (3.6942e+001) | 3.1823e+002 (7.334e+001) | **4.4771e+000** (2.5811e-000) |

## VI. CONCLUSION

In this paper, three metaheuristic algorithms are implemented and compared: SAHS, WDPO and MWDPO. A brief description of each method is presented along with an algorithm to facilitate their implementation. Eight benchmark continuous optimization functions were used. The suggested method was found to perform better than other algorithms in terms of solution quality and optimization iterations.

## ACKNOWLEDGMENT

This research is funded by the Deanship of Research and Graduate Studies in Zarqa University /Jordan

## REFERENCES

[1] H. Al-Tabtabai and P. A. Alex, "Using genetic algorithms to solve optimization problems in construction,". Eng Constr Archit Manage, vol. 6, no. 2, pp.121–32, 1999.

[2] J. Holland, "Adaptation in natural and artificial systems," Ann Arbor, MI: University of Michigan Press, 1975.

[3] T. Hegazy, "Optimization of construction time-cost trade-off analysis using genetic algorithms," Can J Civil Eng, vol. 26, pp. 685–97, 1999.

[4] R. Eberhart and J. Kennedy, "A new optimizer using particle swarm theory," Proceedings of the sixth international symposium on micro machine and human science, Nagoya, Japan, pp. 39–43. 1995.

[5] J. Kennedy and R. Eberhart, "Particle swarm optimization," Proceedings of IEEE International Conference on Neural Network, Piscataway, NJ, pp. 1942–1948, 1995.

[6] G. Evers and M.B. Ghalia, "Regrouping Particle Swarm Optimization: A new Global Optimization Algorithm with Improved Performance Consistency Across Benchmarks," IEEE International Conference on Systems, Man and Cybernetics, San Antonio, pp. 3901–3908, 2009.

[7] X.C. Zhao, "A perturbed particle swarm algorithm for numerical optimization," Appl. Soft Comput. Vol. 10, pp. 119–124, 2010.

[8] V. Ganapathy, T. Tang, S. Parasuraman,"Improved Ant Colony Optimization for Robot Navigation," Proceeding of the 7th International Symposium on Mechatronics and its Applications (ISMA10), Sharjah, UAE, April 20-22, 2010.

[9] Z. W. Geem, "Music-Inspired Harmony Search Algorithm: Theory and Applications," Studies in Computational Intelligence, Springer, 2009.

[10] O. M. Alia and M. Rajeswari, "The variants of the harmony search algorithm: an Overview," Artif. Intell. Rev. vol. 36, pp. 49–68, 2011.

[11] M. G. Omran and M. Mahdavi, "Global-best harmony search," Applied Mathematics and Computation, vol. 198,pp. 643–656, 2008.

[12] C. M. Wang and Y. F. Huang, "Self adaptive harmony search" algorithm for optimization, Expert Systems with Applications, vol. 37, pp. 2826–2837, 2010.

[13] P. Chakraborty, G. G. Roy, S. Das and D. Jain, A. Abraham, "An improved harmony search algorithm with differential mutation operator," Fundam Inform. Vol. 95, pp.1–26, 2009.

[14] E. Al daoud, R. Alshorman and F. Hanandeh, "A New Efficient Meta-Heuristic Optimization Algorithm Inspired by Wild Dog Packs" International Journal of Advanced Science and Technology, vol.71, pp.1-17, 2014.

[15] N. Andrei, "An Unconstrained Optimization Test Functions Collection," Advanced Modeling and Optimization, vol.10 , pp. 147-161, 2008.

[16] M. O. Ali, S. P. Koh, K. H. Chong and F.D.Yap, "Hybrid Artificial Immune System-Genetic Algorithm optimization based on mathematical test functions," IEEE Student Conference on Research and Development, pp 256 – 261, 2010.