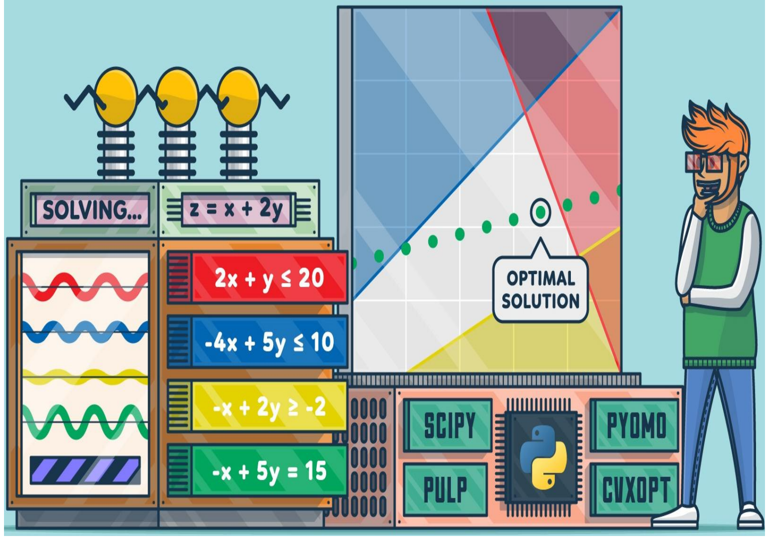
# Implementation of Wolf Pack Algorithm for unconstrained optimization

Yash Shethwala
Praveen Tamrakar

Under Guidance of **Dr. Theodore Pavlic**

# Introduction

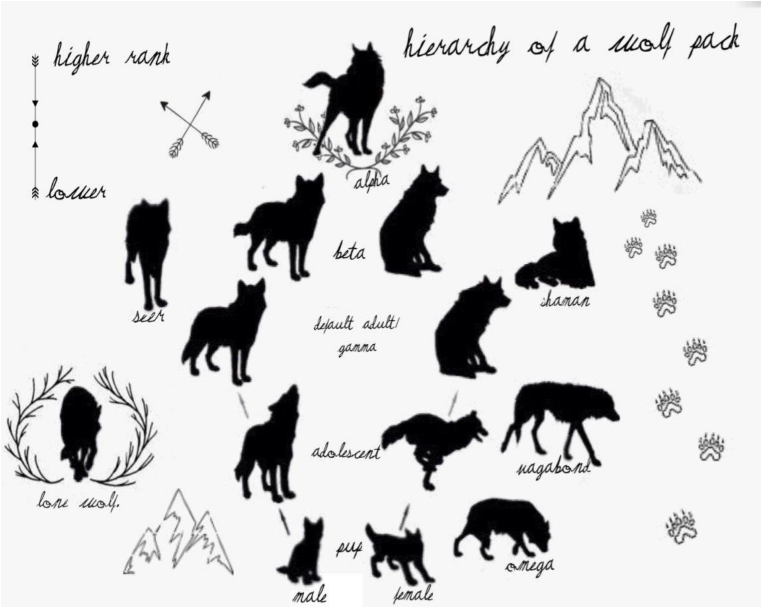
- Global optimization becomes problem due to great search space and high dimensions adding up.
- The animals in nature have developed the powerful swarm intelligence which gave us new ideas to solve the optimization problem.
- Many intelligent optimization algorithms made by researcher to tackle the problem are ACO, PSO, ABC and AFSA.
- General unconstrained optimization can be formulated as below



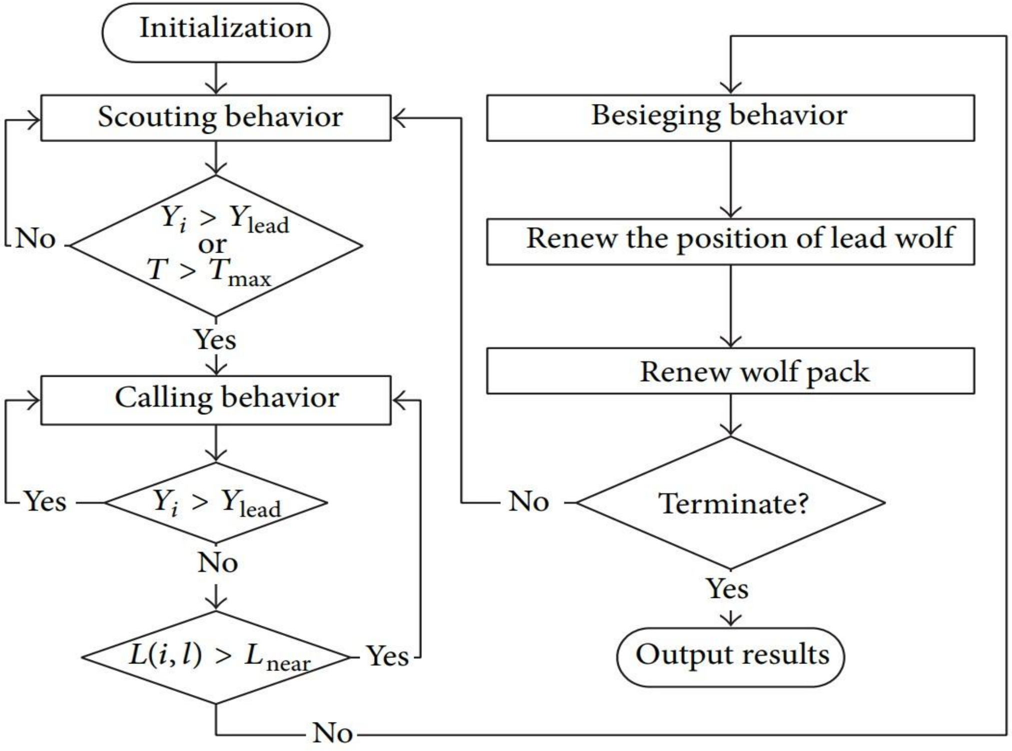
$$\text{min or max } f(X), \quad X = (x_1, x_2, x_3, \ldots\ldots, x_n)$$

# Wolf pack

- A Harsh living environment and constant evolution for centuries have created their rigorous optimization system and subtle hunting behavior gave us ideas to solve the problem.
- They have social work division. There are three division in their groups
  - Lead wolf - smartest and most ferocious
  - Elite wolves - Act as a scout.
  - Ferocious wolves - Hunter
- They works as a group to find a prey, hunt and distribute the food among the group.

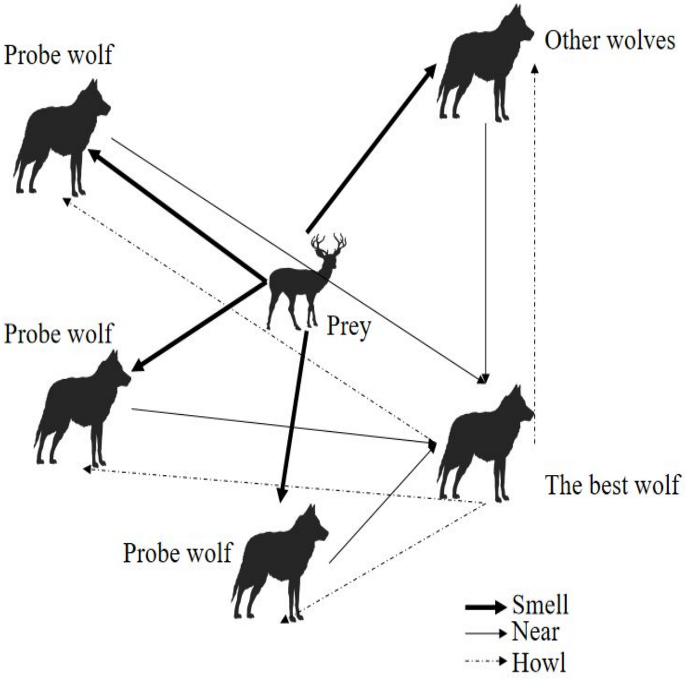


# Flow chart of the wolf pack algorithms



# Generating the Lead wolf

- Lead wolf is responsible for commanding the wolves and constantly making decision by evaluating surrounding situation and perceiving information from other wolves.
- The lead wolf will be decided on the basis of highest smell concentration $Y=f(X)$ where, $f(X)$ is the objective function and $X_i=(x_{i1}, x_{i2}..., x_{in})$ is the position of $i^{th}$ wolf
- It will continuously compare with the other wolves, if the lead wolf will be replaced if some other wolf will find better $Y$.



# Scouting Behavior

- The scouting behaviour is basically interpreted as exploration step of optimization.
- The wolves except for lead wolf will take $step_a$ in $h$ different direction and check if max$\{Y_{i1}, Y_{i2}..., Y_{ip}\} > Y_{0i}$ and update its position $X$.
- This step will be repeated until $Y_i > Y_{lead}$ or the maximum number of iterations $T_{max}$ is achieved for that behavior.

$$x_{id}^{p} = x_{id} + \sin\left(2\pi \times \frac{p}{h}\right) \times step_a^d, \quad p = \{1, 2\ldots, h\}$$

# Calling behavior

- Lead wolf will howl and summon the ferocious wolf for the hunt.
- The lead wolf is mapped as position of the prey, causing the ferocious wolves gather around the position of the lead wolf by taking step length of $step_b$.
- This behaviour will continue until $L(i,l)<L_{near}$ when no new leader is formed
- If $Y_i > Y_{lead}$, the lead wolf will be replaced by wolf *i* and all the wolf will take scouting behavior.

$$x_{id}^{k+1} = x_{id}^{k} + step_b^d \cdot \frac{\left(g_d^k - x_{id}^k\right)}{\left|g_d^k - x_{id}^k\right|}$$

# Besieging behavior

- Besieging is a procedure in which wolves move with step length of $step_c$ towards prey for the hunt.
- Except the lead wolf all the wolves take this step to kill the prey.
- Prey position will be the position of the lead wolf.
- The wolf will update its position only if $Y_{ik} > Y_{i0}$.

$$x_{id}^{k+1} = x_{id}^{k} + \lambda \cdot step_c^d \cdot \left|G_d^k - x_{id}^k\right|$$

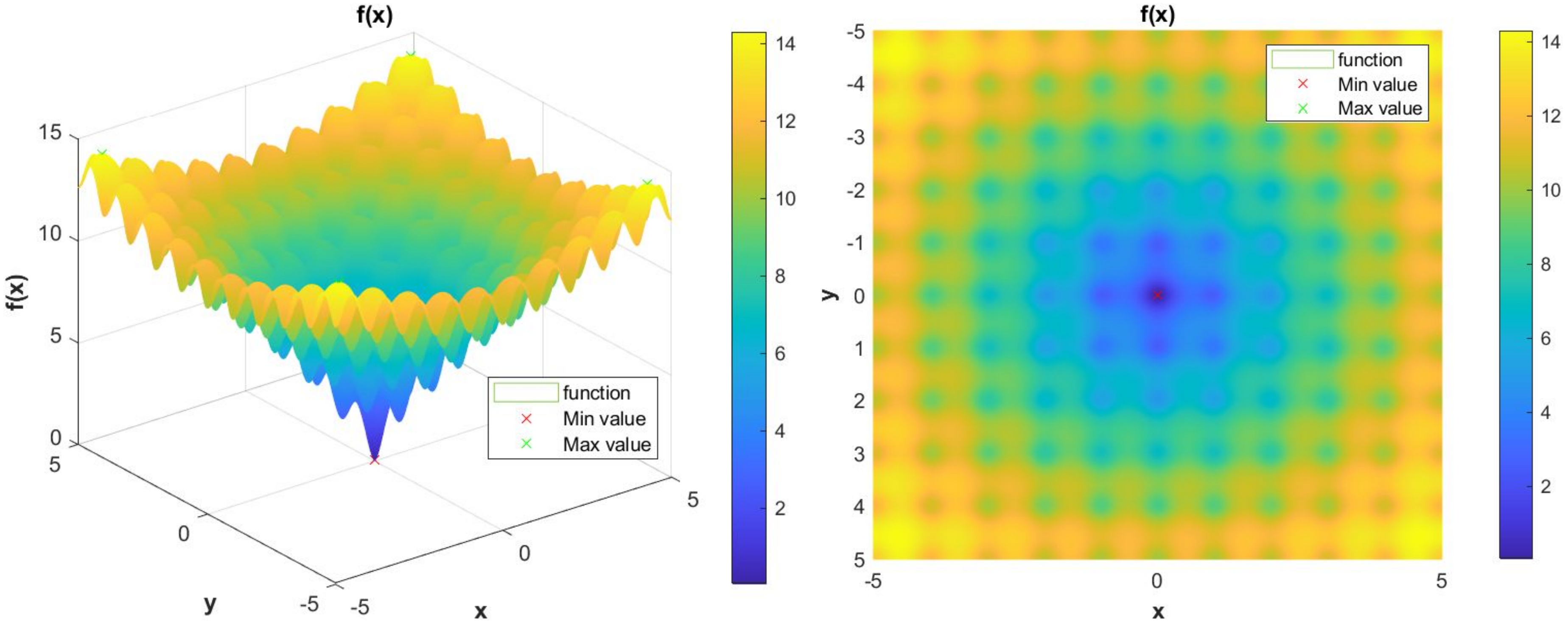
# Renew the wolf

- After the catching the prey, its divided among the wolves unequally.
- Stronger wolves get more food than weaker wolves.
- Simulation of this behaviour is done by making dead R wolves while generating the new R wolves near the position of the lead wolf.
- The position of the wolves will be updated using given equation
- Select R randomly as an integer in the range [*n*/(2·*β), n*/*β*], where *β* is the population renewing proportional coefficient.

$$x_{id} = g_d \cdot (1 + rand), \quad i = \{1,2\ldots, R\}$$

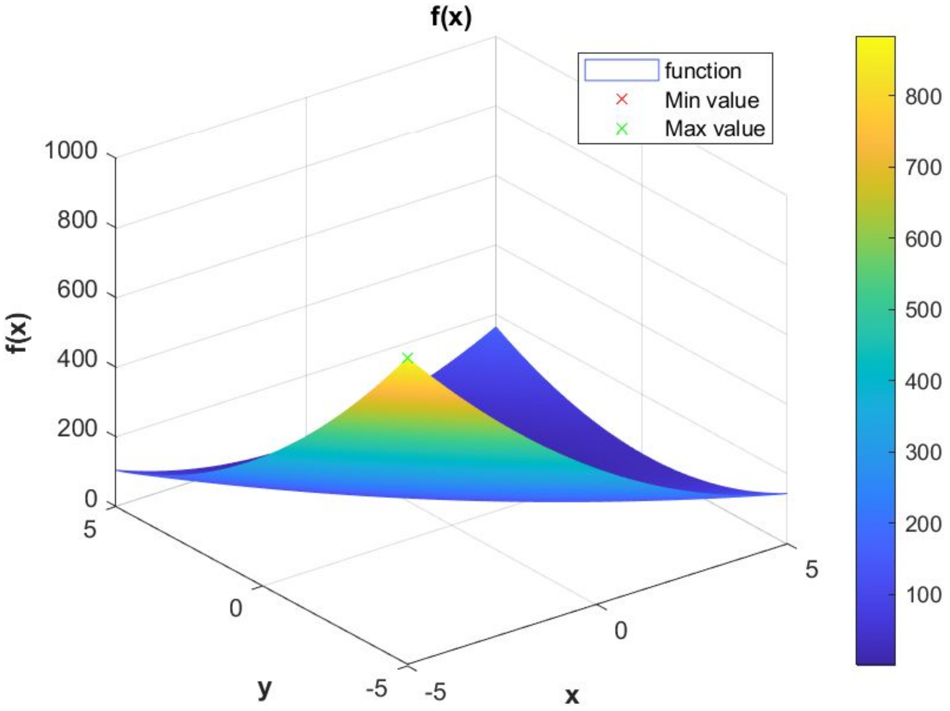
# Test functions

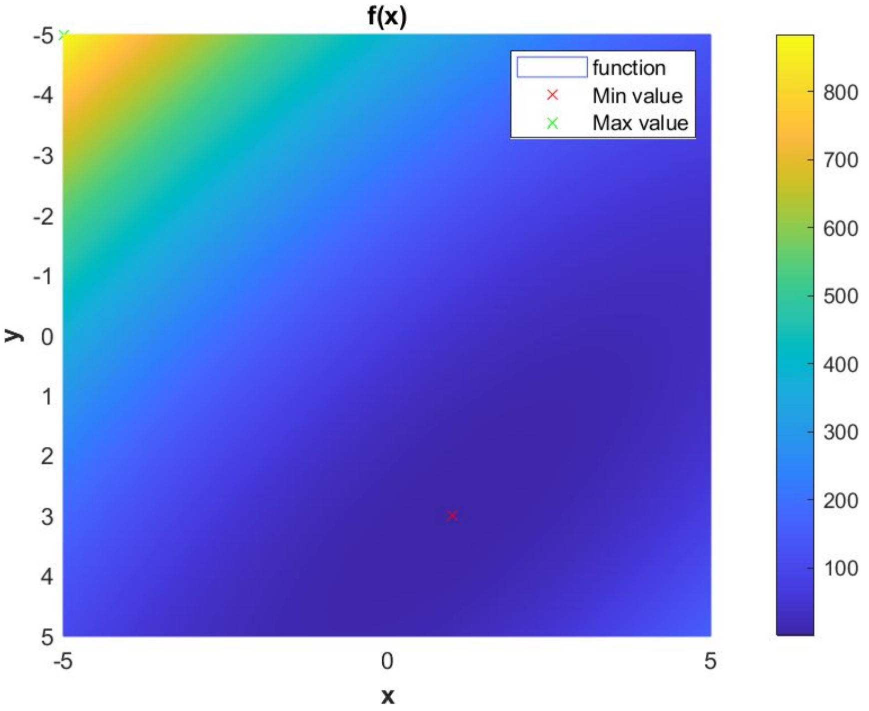
- We have experimented with 3 off-the-shelf test function which can be seen in next slides.
- Those functions are Auckley function, Booth function and Easom function.
- The auckley function is a multimodal function with optima $f(0,0) = 0$
- The booth function is with optima $f(1,3) = 0$
- The Easom function has its optima $f(\pi,\pi) = -1$



## Auckley Function

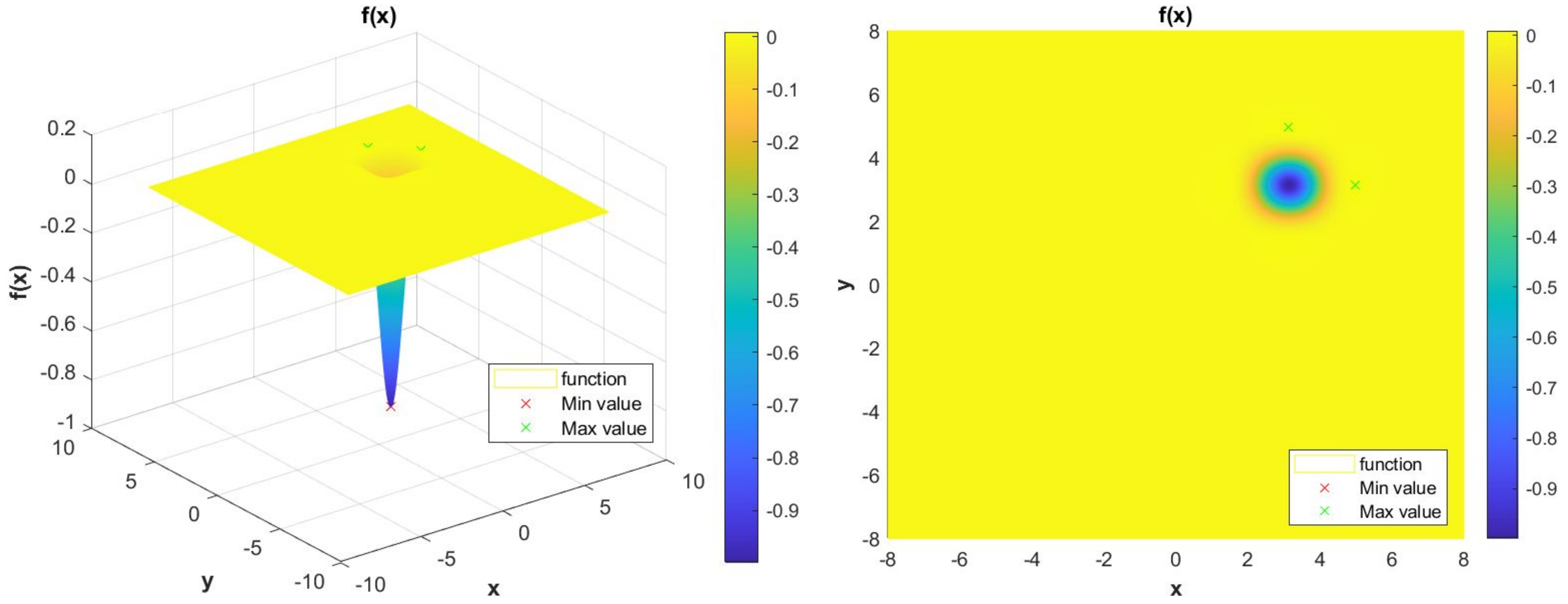
$$f(x,y) = -20e^{-20\sqrt{0.5(x^2+y^2)}} - e^{-0.5(\cos 2\pi x + \cos 2\pi y)} + e + 20$$





Booth Function

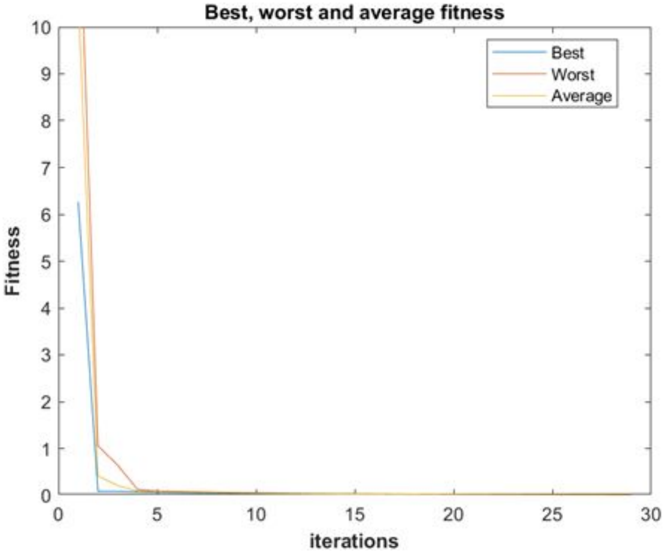
$$f(x, y) = (x + 2y - 7)^2 + (2x + y - 5)^2$$



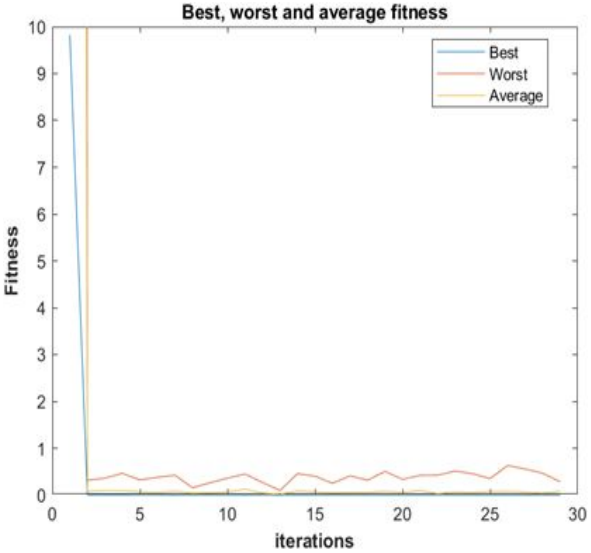
# Easom Function

$$f(x,y) = -\cos(x)\cos(y)\,e^{-((x-\pi)^2+(y-\pi)^2)}$$

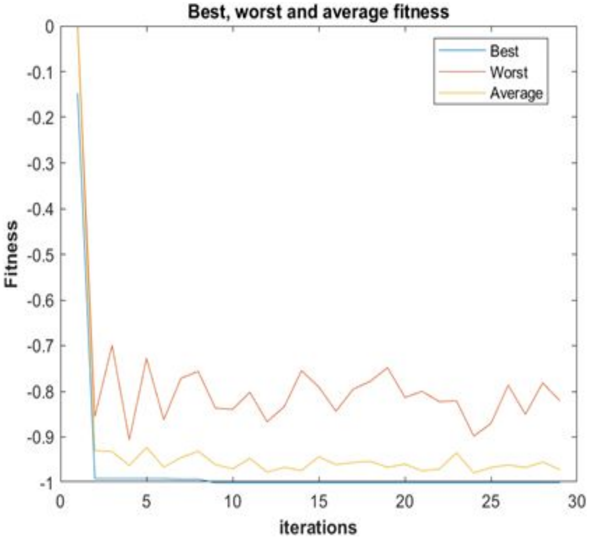
# Results



Auckley Function



Booth Function



Easom Function

# Conclusion

- The number of wolves is directly related to the rate of accuracy, but it is indirectly proportional to the rate of convergence.
- We need to select optimal value of $L_{near}$ as It will take a long time or may not even converge with really smaller value of $L_{near}$. Also, the larger value will take longer time to converge. However, the smaller value of the hyper-parameter $L_{near}$ ,will increase exploitation.
- The higher the value of S, the faster the convergence, however it may oscillate around the prey during besieging behavior. On the other hand, if we maintain the value of S smaller, it will take significantly longer to converge and finish the besieging behavior.
- WPA algorithm is a great optimization method, providing quick and accurate results.

# References

[1] C. Grosan and A. Abraham, “A novel global optimization technique for high dimensional functions,” International Journal of Intelligent Systems, vol. 24, no. 4, pp. 421–440, 2009.

[2] F. Kang, J. Li, and Z.Ma, “Rosenbrock artificial bee colony algorithm for accurate global optimization of numerical functions,” *Information Sciences*, vol. 181, no. 16, pp. 3508–3531, 2011.

[3] Y. Yang, Y.Wang, X. Yuan, and F. Yin, “Hybrid chaos optimization algorithm with artificial emotion,” Applied Mathematics and Computation, vol. 218, no. 11, pp. 6585–6611, 2012.

[4] W. S.Gao andC. Shao, “Pseudo-collision in swarm optimization algorithm and solution: rain forest algorithm,” Acta Physica Sinica, vol. 62, no. 19,Article ID 190202, pp. 1–15, 2013.

[5] Y. Celik and E. Ulker, “An improved marriage in honey bees optimization algorithm for single objective unconstrained optimization,” The Scientific World Journal, vol. 2013,Article ID 370172, 11 pages, 2013.

[6] M. Dorigo, Optimization, learning and natural algorithms [Ph.D. thesis], Politecnico di Milano, Milano, Italy, 1992.

# References

[7] D. Karaboga, “An idea based on honeybee swarm for numerical optimization,” Tech. Rep. TR06, Computer Engineering Department, Engineering Faculty, Erciyes University, Kayseri, Turkey, 2005.

[8] J. Kennedy and R. Eberhart, “Particle swarm optimization,” in Proceedings of the IEEE International Conference on Neural Networks, pp. 1942–1948, Perth, Australia, December 1995.

[9] X.-L. Li, Z.-J. Shao, and J.-X. Qian, “Optimizing method based on autonomous animats: Fish-swarm Algorithm,” System Engineering Theory and Practice, vol. 22, no. 11, pp. 32–38, 2002.

[10] C.-G. Liu, X.-H. Yan, and C.-Y. Liu, “The wolf colony algorithm and its application,” Chinese Journal of Electronics, vol. 20, no. 2, pp. 212–216, 2011.

[11] J. A. Ruiz-Vanoye, O. D´ıaz-Parra, F. Coc´on et al., “Meta-Heuristics algorithms based on the grouping of animals by social behavior for the travelling sales problems,” International Journal of Combinatorial Optimization Problems and Informatics, vol. 3, no. 3, pp. 104–123, 2012.

[12] Hu-Sheng Wu, Feng-Ming Zhang, "Wolf Pack Algorithm for Unconstrained Global Optimization", Mathematical Problems in Engineering, vol. 2014, Article ID 465082, 17 pages, 2014. https://doi.org/10.1155/2014/465082

“ Thank You