

# Improved Query Processing in Web Search Engines Using Grey Wolf Algorithm

Nishant Pal<sup>1</sup>, Akshat Chawla<sup>2</sup>, A.Meena Priyadharsini<sup>3</sup>

<sup>1,2</sup>UG Student, <sup>3</sup>Assistant Professor

<sup>1,2,3</sup>Department of CSE, SRM University, Kattankulathur.

\*Corresponding Author Email: [nishant\\_pal@srmuniv.edu.in](mailto:nishant_pal@srmuniv.edu.in)

## Abstract

In Information systems working at a large scale where retrieval of information is an essential operation for example search engines etc. The users are not only concerned with the quality of results but also the time they consume for querying the data. These aspects lead to a natural tradeoff in which the approaches that lead to an increase in data have a similar larger response time and vice-versa. Hence, as the requirement for faster search query processing time along with efficient results is increasing, we need to identify other ways for increasing efficiency. This work proposes an application of the meta-heuristic algorithm called Grey Wolf Optimization (GWO) algorithm to improve Query Processing Time in Search Engines. The GWO algorithm is an alter ego of the way in which the grey wolves are organised and their hunting techniques. There are four categories of grey wolves in a single pack of grey wolves which are alpha, beta, delta, and omega respectively. They are used to work in a simulating hierarchy. These help achieve better search results at decrease query response timings.

## 1. Introduction

Since the internet is full of large amount of data, web search engines require operations such as crawling and indexing of a large number of web pages and arrive at the results in the least possible time when queried by users. Users' queries are processed by physical servers that act as query processing nodes. Web search engines therefore are composed of a large number of these query processing nodes, along with infrastructures for thermal cooling, telecommunication, power supply etc. which are hosted in huge datacenters containing thousands of petabyte of data. This complex infrastructure is required necessarily to decrease the time required for processing the search queries so that users can get the query results in minimum time. Since every day some thing or the other is happening in the world, the information content on the internet is increasing day by day. Web search engines require new strategies to tackle with the increasing data and the demand for faster results. This work proposes the application of GREY WOLF Optimisation algorithm for the reduction of query execution time.

Grey Wolf algorithm follows the process of hunting in a hierarchy just like the grey wolves do. The GWO algorithm represents the way in which the grey wolves categorise themselves and perform the hunting process using a particular strategy. In a pack of wolves there are four types of wolves which are represented as alpha wolves, beta wolves, delta wolves, and omega wolves. The  $\alpha$  wolves are the strongest of all the wolves in the pack, they lead the remaining wolves and also take important decisions along with tracking of other wolves in the group for the maintenance of social equality. The  $\beta$  wolves are less strong than the alpha wolves, they follow the alpha wolves in hunting process and when alpha wolves die or get injured, the beta wolves take their place.

The  $\delta$  wolves control the remaining pack of  $\omega$  wolves and provide coordination by transferring important information about  $\omega$  wolves

to  $\alpha$  and  $\beta$  wolves. The  $\omega$  wolves are the lowest in the hierarchy and comprise the remaining pack of wolves. The paper shows that Grey Wolf Optimization algorithm allows lesser query processing time than the existing particle swarm optimisation that is used in processing search queries.

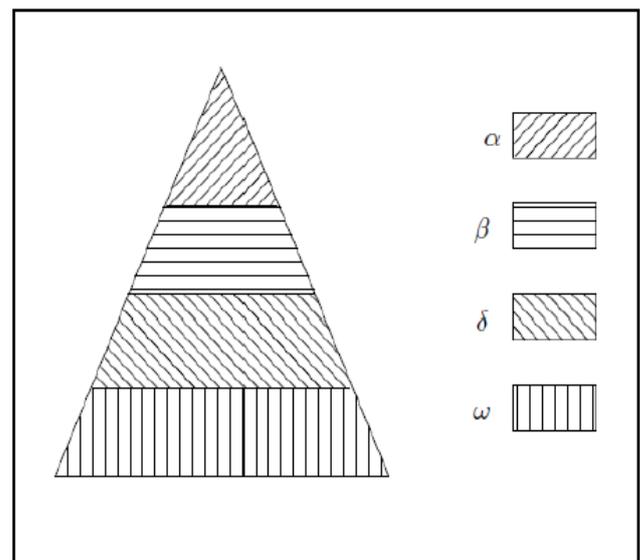


Fig. 1: Social Hierarchy of Grey Wolf

## 2. Related Work

- Zhihui Du proposed energy efficient scheduling for interactive services. The reducing of energy consumption, helped in directly reducing the operational cost of service providers. The flaw was

ignoring other factors such as the monetary cost of accessing resources.

- **Saurabh Bilgaiyan** proposed analysis of task scheduling in Cloud Computing using Evolutionary and Swarm-based Algorithms. By making the use of cloud computing, the scheduling of the task was done on the resources so as to provide maximum performance in minimum time but the drawback in this was cost for the resources was not taken care of.
- **Mahdi Mahjour-Bonab** proposed uses Particle Swarm Optimisation algorithm to optimize queries in grid structured databases. They made repetitions on different particles to make the query function on various websites to obtain optimum results which were later compared. The problem with this was this lead to exhaustive search.
- **Seyedali Mirjalili** proposed a new meta-heuristic called Grey Wolf Optimizer (GWO) which is evolved from the hunting behavior of grey wolves. There are mainly four types of grey wolves namely alpha, beta, delta, and omega. The topmost wolf is the alpha wolf in hierarchy. The author mainly implemented steps of hunting, searching, encircling, and attacking prey. The results were obtained and then compared with Particle Swarm Optimization (PSO), Gravitational Search Algorithm (GSA), Differential Evolution (DE), Evolutionary Programming (EP), and Evolution Strategy (ES). By comparing the results of all algorithm it was concluded that grey wolf algorithm gives best performance of all techniques.
- **Craig Macdonald** proposed to predict response time for Online Query Scheduling. Author said that execution time of queries should be made known to users for different dynamic scheduling servers in advance so that one can use the best server available to user and reduce the execution time. Its drawbacks include more time consumption in processing queries and difficulty in determining the execution time in advance.
- **Xiao mingyao** proposed in his research paper Embedded database query optimization algorithm based on particle swarm optimization that Particle Swarm Optimization in embedded query processing has shown considerable results in reducing query processing time but have not been tested for large databases. In large databases it has been found out that Particle Swarm Optimization needs to take care of specific details based on the database to give a good performance.

### 3. Existing System

In existing system, the algorithm used for query optimization is Particle swarm algorithm for optimisation.e., a heuristic global algorithm, which works on the principle of swarm behaviour. It is derived from the various studies and research done on the movement of birds and fishes. The algorithm is used in many varied applications allowing efficient utilization of the algorithm in many scenarios with the requirement of changes in only a few parameters. While searching for food the birds (searching agents) are scattered to allow them to move in all directions to search for the optimum (food). One among the many birds is present which has the clearest information about the location of the optimum that satisfies the user's given criterion. This bird is located and is used to find the optimum solution.

### 4. Proposed System

In the proposed system, the algorithm used for query optimization is Grey Wolf Algorithm. In GWO, the wolf which represents the solution with the best fitness value is categorized as the alpha ( $\alpha$ ) wolf. The wolves with second best and third best fitness values are called beta ( $\beta$ ) wolf and delta ( $\delta$ ) wolf respectively. The other remaining candidate solutions are referred to as omega ( $\omega$ ) solutions. In the GWO, the hunting process (optimization) is led

by  $\alpha$ ,  $\beta$ , and  $\delta$  wolves. The  $\omega$  wolves follow these three leading wolves to locate the optimum solution.

### A. Pseudocode for GWO algorithm

```

Initialize the grey wolf population  $P_i$  ( $i = 1, 2, \dots, n$ )
Initialize the values of  $a$ ,  $A$ , and  $C$ 
Calculate the fitness of each search agent
 $P\alpha$ =the most efficient search agent
 $P\beta$ =the second most efficient search agent
 $P\delta$ =the third most efficient search agent
while ( $t < \text{Max number of iterations}$ )
for each search agent
Update the current search agent's position by equation (3.7)
end for
Update the values of  $a$ ,  $A$ , and  $C$ 
Calculate all search agents' fitness values.
Update the values of  $P\alpha$ ,  $P\beta$ , and  $P\delta$ 
 $t=t+1$ 
end while
return  $P\alpha$ 

```

Following are some important facts that should be taken into consideration while estimating the performance of GWO algorithm theoretically:

- The hierarchical social behaviour of grey wolves helps to locate the best solutions that have occurred during the iterations.
- The encircling mechanisms helps to identify a circular shaped neighbourhood region around the solutions. This can also be applied to hyper spheres.
- The parameters  $A$  and  $C$  benefit the GWO algorithm by allowing the candidate solutions to have random unique radii of hyper-spheres.
- The hunting methods allows the candidate wolves to estimate the location of prey.
- $a$  and  $A$  vector parameters ensure efficient exploration and exploitation.
- As the values of parameters  $a$  and  $A$  vary a gentle transition occurs that causes the exploration process to be converted into exploitation process in GWO algorithm.
- As the value  $A$  decreases, half of the iterations are utilized for exploring the prey where  $|A| \geq 1$  and the other half are utilized for exploiting the location of the prey where  $|A| < 1$ .
- The only main parameters to be adjusted in GWO are ( $a$  and  $C$ ).

### 5. Methodology

The main steps of grey wolf hunting process are as follows:

- Encircling the prey.
- Hunting
- Attacking the prey
- Searching next prey.

#### A. Encircling the prey

During the process of hunting, the grey wolves try to gather around the prey and encircle it. The following equations describe how the wolves behave, mathematically-

$$\vec{D} = |\vec{c} \cdot \vec{X}_p(t) - \vec{x}(t)|$$

$$\vec{x}(t+1) = \vec{X}_p(t) - \vec{A} \cdot \vec{D}$$

The current iteration is marked by  $t$ , coefficient vectors are  $C$  and  $A$ , prey's position vector is denoted by  $X_p$ , and  $X$  represents the

location vector of grey wolf. The vectors **C** and **A** are calculated using the following formulas:

$$\vec{A} = 2\vec{a} \cdot \vec{r}_1 - \vec{a}$$

$$\vec{C} = 2 \cdot \vec{r}_2$$

As the iterations occur the values of **a** are varied linearly between 2 and 0 provided the random vectors **r**<sub>1</sub>, **r**<sub>2</sub> lie in [0,1].

**B. Hunting**

Grey wolves are dexterous at recognizing the location of prey and encircle them. The alpha wolves lead the hunt. Occasionally the beta and delta wolves may also take part in the hunting process. However, in an abstract search space we have no clue about the prey’s position(optimum).For mathematically simulation of the hunting behavior of grey wolves, we assume that the Alpha ,Beta and Gama wolves have better knowledge about the position of the optimum(pre) and thus they are better situated near the prey. As a result we have determined the best three solutions, the other candidate solutions or the omega wolves follow these three wolves by updating there values according to the best solutions that has been found. The following formulas signify this concept.

$$\vec{D}_\alpha = |\vec{C}_1 \cdot \vec{X}_\alpha - \vec{X}|, \vec{D}_\beta = |\vec{C}_2 \cdot \vec{X}_\beta - \vec{X}|, \vec{D}_\delta = |\vec{C}_3 \cdot \vec{X}_\delta - \vec{X}|$$

$$\vec{X}_1 = \vec{X}_\alpha - \vec{A}_1 \cdot (\vec{D}_\alpha), \vec{X}_2 = \vec{X}_\beta - \vec{A}_2 \cdot (\vec{D}_\beta), \vec{X}_3 = \vec{X}_\delta - \vec{A}_3 \cdot (\vec{D}_\delta)$$

$$\vec{X}(t + 1) = \frac{\vec{X}_1 + \vec{X}_2 + \vec{X}_3}{3}$$

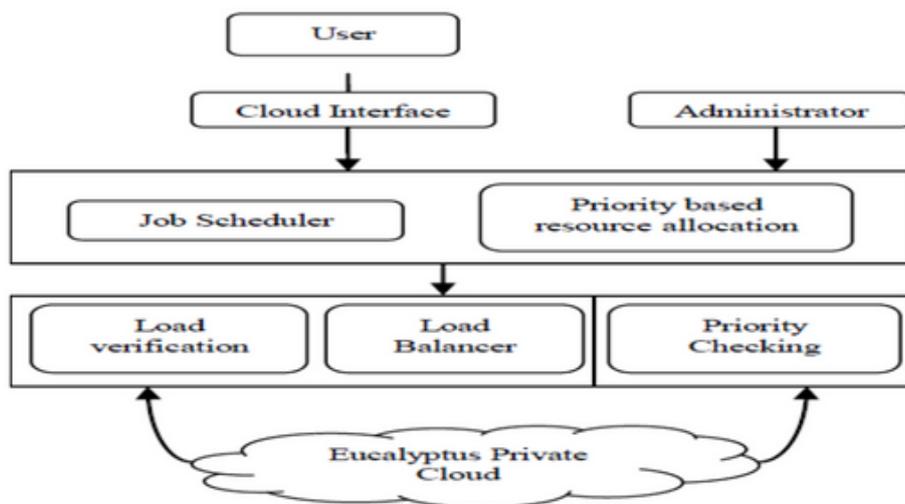
**C. Attacking prey (exploitation)**

The hunting process of the grey wolves come to an end when the prey stops moving. In order to mathematically explain this step of approaching prey we lower the value of **a** which causes a decrease in the changing ranges of **A**. In other words **A** is a random value which lies in the interval [-a,a] where **a** is varied between 2 to 0 during the iterations. When random values of **A** lies in the interval [-1,1], the search agent’s next move can occupy any position

between its current position and the position of the prey. The proposed operators help the GWO algorithm to update the location of the search agents to the locations corresponding to alpha, beta, and delta; and hence move towards the prey and attack it thus location the optimum solution. However, these operators make the GWO algorithm prone to getting stagnated at the local optima. In order to provide efficient exploration we require more operators.

**D. Search for prey (exploration)**

Grey wolves mostly carry out the search operation depending on the alpha, beta, and delta wolves’ position. They segregate in all directions to search for prey and after determination of the location they congregate in a single group to attack the prey. In order to mathematically explain the diverging concept, we keep the values **A** random values such that |A|>1 which makes the search agents to segregate in all directions, allowing them to explore the search space fully to catch prey. The values of **C** vector are random and lie in the interval [0, 2]. It provides randomized weight values for prey in order to emphasize stochastically (**C**>1) or deemphasize (**C**<1) the effect of prey in determining the distance. This helps GWO to work in a more random manner throughout optimization, which allows efficient exploration and local optima avoidance. It is important to mention here that **C** does not decreases linearly in contrast to **A**. It is essentially required that **C** must provide random values so that the exploration is emphasized in both initial and final iterations. The **C** vector makes it possible to avoid the situation of local optima stagnation. The **C** vector can be also acts as an alternative to nature putting obstacles in the path of approaching wolves. The nature puts obstacles in the path of hunting wolves in order to prevent them from reaching the prey quickly and conveniently. This is exactly the function of vector **C**. Depending on the wolf’s position, it can assign a random weight to the prey such that it becomes difficult for wolf to reach it or vice versa. The epitome of the search process consists of initialising a population of grey wolves (candidate solutions) consisting of random number of wolves in the beginning. During the iterations, alpha, beta, and delta wolves capture the position of the prey which is most probable through estimation. Finally, the GWO algorithm is consummated by achieving an end criterion.



**Fig. 2:** Architecture System

The user enters the search query in the search engine .This query is sent to the cloud where the server is hosted .The server runs the Gwo algorithm to search for the best required results depending on the number of results the user wants. The 22 benchmark functions

are used to get best fitness among the results .The results are sent to the load verification and balancer which removes the unwanted and spam websites. Depending upon the queries location ,the result that has the highest fitness favourable to that location from

which the query arrived is chosen. The Eucalyptus Private Cloud in this example provides the dataset of search queries with corresponding location and number of visitors which is used by the Gwo algorithm to get the optimum result.

GWO	PSO
0	4.24E-17
3.07E-243	1.31E-09
3.42E-216	0.894981
25.0925	17.4573
0	0
4.49E-05	0.0149869
0	1.69E-16
-8164.71	-8304.92
0	32.8336
7.99E-15	2.93E-09
0	0
0.0130216	7.15E-20
1.01E-06	3.82E-18
0.998004	0.998004
-1.03163	-1.03163
0.397887	0.397887
3	3
-0.300479	-0.300479
-3.32199	-3.322
-10.1532	-10.1532
-10.4029	-10.4029
-10.5363	-10.5364
11	6
17	13

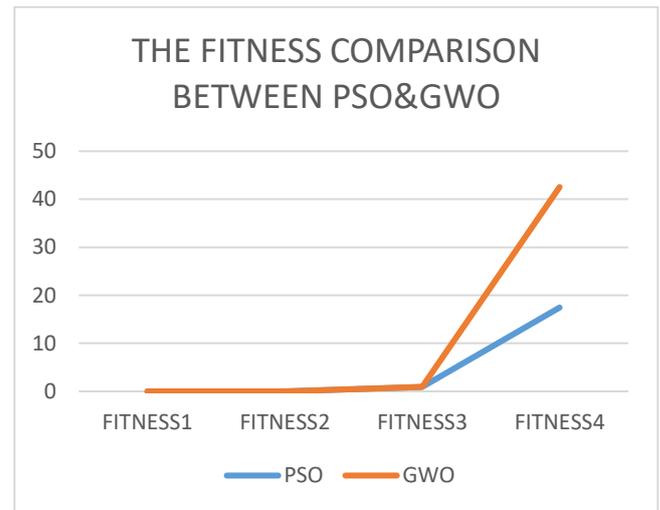
Table 1: Characteristics of Benchmark Functions

No.	Function Name	Range	Dimen sion	$f_{min}$	Type
1	Sphere	[-100,100]	30	0	U
2	Schwefel 2.22	[-10,10]	30	0	U
3	Schwefel 1.2	[-100,100]	30	0	U
4	Rosenbrock	[-30,30]	30	0	U
5	Step	[-100,100]	5	0	U
6	Quartic	[-1.28,1.28]	30	0	U
7	SumSquares	[-10,10]	30	0	U
8	Schwefel	[-500,500]	30	-12569.5	M
9	Rastrigin	[-5.12,5.12]	30	0	M
10	Ackley	[-32,32]	30	0	M
11	Griewank	[-600,600]	30	0	M
12	Penalized	[-50,50]	30	0	M
13	Penalized2	[-50,50]	30	0	M
14	Foxholes	[-65,65]	2	1	FDM
15	Six hump camel back	[-5,5]	2	-1.031	FDM
16	Branin	[-5,5]	2	0.398	FDM
17	GoldStein-Price	[-2,2]	2	3	FDM
18	Hartman3	[1,3]	3	-3.86	FDM
19	Hartman6	[0,1]	6	-3.32	FDM
20	Shekel5	[0,10]	4	-10.1532	FDM
21	Shekel7	[0,10]	4	-10.4028	FDM
22	Shekel10	[0,10]	4	-10.5363	FDM

### 6. Conclusion

An efficient application of meta-heuristic Grey Wolf Algorithm is proposed to eliminate the waiting time. Our proposed algorithm reduces the total query execution time and resource consumption. The proposed algorithm sets some conditions such as all the

queried outputs are independent to each other. Furthermore, it automatically categorises the query result. From the graph, we observed that our proposed algorithm is helps achieve faster search query results and hence emerges as a strong competitive algorithm in Search Systems design providing better and more efficient results than existing particle swarm optimization algorithm.



### References

- [1] Zhihui Du\*, Hongyang Sun†, Yuxiong He‡, Yu He\*, David A. Bader§, Huazhe Zhang “Energy-Efficient Scheduling for Best-Effort Interactive Services to achieve high response quality”
- [2] Saurabh Bilgaiyan, Santwana Sagnika, Madhabananda Das School of Computer Engineering.I.” An Analysis of Task Scheduling in Cloud Computing using Evolutionary and Swarm-based Algorithms”
- [3] Mahdi Mahjour-Bonab1 and Javad Sohafi-Bonab“Query Optimization in Grid Databases Using with Particle Swarm Optimization”
- [4] Mirjalili, Seyedali, Seyed Mohammad Mirjalili, and Andrew Lewis. "Grey wolf optimizer."
- [5] Craig Macdonald1, Nicola Tonello2, Iadh Ounis1 “Learning to Predict Response Times for Online Query Scheduling”
- [6] Xiao mingyao,Li Xiongfei “Embedded database query optimization algorithm based on particle swarm optimization”
- [7] M. N. Alpdemir, A. Mukherjee, and N. W. Paton, “Servicebased Distributed Querying on the Grid”. ICSSOC 2003, First International Conference, Trento, Italy, 2003, PP.467-482.
- [8] N. Bruno, and S. Chaudhuri, “Exploiting Statistics on Query Expressions for Optimization”. In SIGMOD 2002, Proceedings ACM SIGMOD International Conference on Management of Data, June 3-6 2002.
- [9] S. Shivle, and H. J. Siegel, “Mapping of Subtasks with Multiple Versions in a Heterogeneous Ad Hoc Grid Environment” Third International Symposium on Parallel and distributed computing, 2004,PP.380-387
- [10] Z. Zhou, “Using Heuristic and Genetic Algorithms for Large-Scale Database Journal of Information and Optimization.” Computing Science, Vol. 2, No. 4, 2007, PP.261-280.
- [11] J. Wang, J. Horng, Y. Hsu, and B. Liu, “A Genetic Algorithm for Set Query Optimization in Distributed Database Systems.”, IEEE, 1996, PP.1977-1982.
- [12] A. Goyal, L. Vasiliu, and B. Sapkota, “Use of AI Query Optimization of Relational Database.”, in 18th IEEE International Conference on tools with Artificial Intelligence (ICTAI06), 2006.
- [13] J. Kennedy, and Eberhart, Swarm Intelligence, Morgan Kaufman, 2001.
- [14] Z. J. Li, X. D. Liu,X. D. Duan, and C. R. Wang, “Optimal Solution for Grid Resource Allocation Using Particle Swarm Optimization”, Third International Conference on Multimedia and Ubiquitous Engineering, 2009, PP.339-346.
- [15] V. Zadorozhny, A. Gal, and L. Raschid, “AreNA: Adaptive Distributed Catalog Infrastructure Based on Relevance

- Network.”, In the 31st VLDB Conference, Trondheim, Norway, 2005.
- [16] PlanetLab, PlanetLab Platform, <http://www.planet-lab.org>.
- [17] L. Shuo, and H. K. Karimi, “Grid Query Optimizer to Improve Query Processing in Grids”, *Future Generation Computer Systems*, Vol.24, 2008,PP.342-353.
- [18] M. Lovbjerg, “Improving Particle Swarm optimization by Hybridization of Stochastic Search Heuristics and Self-Organized Criticality”, M.S. thesis, supervisor: T.Krink, university of Aarhus, May 2002.
- [19] M. Clerc, and J. Kennedy, "The Particle Swarm Explosion, Stability and Convergence in A Multidimensional Complex Space", in *IEEE Transaction on Evolutionary computation*, Vol.6, Feb 2002, PP.58-73.
- [20] J. Riget, and S. Vesterstrom, “A Diversity- Guided Particle Swarm Optimizer The ARPSO”, in *Journal of Evalife Technical Report*, Vol.2, 2002.
- [21] Y. Shi, and R. Eberhart, “Empirical Study of Particle Swarm Optimization”, In *Proceedings of Congress on Evolutionary Computation*, 1999, PP.1945-1950.
- [22] Z. Jirong, “A Modified Particle Swarm Optimization Algorithm”, *Journal of Computers*, Vol.4, No.12, 2009, PP.1231-1236.
- [23] Shu\_Chuan, C. Jui\_Fang, and P. Jui\_fang, “A Parallel Particle Swarm Optimization Algorithm With Communication Strategies”. *IJCSI International Journal of Computer Science Issues*, Vol. 9, Issue 6, No 2, November 2012.
- [24] T. Padmapriya and V. Saminadan, “Distributed Load Balancing for Multiuser Multi-class Traffic in MIMO LTE-Advanced Networks”, *Research Journal of Applied Sciences, Engineering and Technology (RJASET) - Maxwell Scientific Organization*, ISSN: 2040-7459; e-ISSN: 2040-7467, vol.12, no.8, pp:813-822, April 2016.
- [25] S.V.Manikanthan and K.Baskaran “Low Cost VLSI Design Implementation of Sorting Network for ACSFD in Wireless Sensor Network”, *CiiT International Journal of Programmable Device Circuits and Systems*, Print: ISSN 0974 – 973X & Online: ISSN 0974 – 9624, Issue : November 2011, PDCS 112011008.