Chapter 1: INTRODUCTION

Ground water exploration is very challenging problem as there is shortage of surface water in urban area as well as rural areas. Water is used for industrial, domestic and agricultural use but with the excessive growth of population, this resource is depleting. Under ground water is a best substitute in these areas. Water is one of the mankind's most vital resources. There is a conventional underground water survey method i.e. called as "Destructive surveying" (like drilling, monitoring wells, and excavation) that sometimes consumes massive manpower and yet can't precisely build up a system of ground water levels. Generally we use "Ground penetrating Radar" to find out the possibility but it's not efficient in some cases. Before digging, if we can find out the possibility of ground water in that particular region then we can easily reduce the manpower. To find out groundwater accurately and also to determine the quantity and quality of the water, various techniques must be used. Researchers have used various techniques in past years like Biogeography based optimization (BBO), particle swarm optimization (PSO), etc. But these techniques fail to give optimum solution to this real time problem.

1.1. Motivation

Under Groundwater exploration is covered in our research work. Around 70.9% of earth's surface covered with water. The presence of human culture can't be overemphasized without preservation of water. Ground water constitutes only 0.6% of all the water on this earth planet, 97.4% accounts for sea water and 2 % for snow on the poles. Generally ground water is obtained below the surface of earth. Thus, Ground water is basically a one of the important commodity for human living. As it is used for industrial, domestic and agricultural use but with the increase in population its resources are depleting. Therefore we need a system or any mechanism to preserve water which depicts the percentage of water beneath the earth's surface. Geoscientist needs a system to understand the potentiality of water at certain regions on earth. In the field of remote sensing, there are various systems exists that finds out the existence of ground water but their efficiency is very low.

Grey wolf Optimizer is a new metaheuristic algorithm comes under natural computation technique. It was introduced by Seyedali Mirjalili, Seyed Mohammad Mirjalili and Andrew Lewis in the year 2014[3] in their research paper titled ―Grey Wolf Optimizer. This algorithm is motivated by grey wolves (Latin name of grey wolf is 'Canis Lupus'). This algorithm impersonates the chasing system and initiative pecking order. This metaheuristic algorithm has shown good results in various other applications like travelling tournament problem. This motivated us to apply it for ground water detection integrated with case based reasoning.

1.2. Related Work

Exploration of Under Groundwater is a field of research which is being solved by several geoscientists through their various traditional techniques such as Ground Penetrating Radar (GPR). But it is a new field of research for computer scientists.

Few works are being done around there. Swarm intelligence algorithm such as Particle Swarm Optimization (PSO), Biogeography Based Optimization (BBO)[4], Cuckoo Search[1] has already been applied for exploration method. They had used case based reasoning for prediction mechanism. The main drawback of above mentioned problem is in some cases the output shows discrepant result.

In case of biogeography based optimization and particle swarm optimization, the one difficult part of research is to figure it out the sufficient amount of cases. It gives poor accuracy when search space is huge. However, other techniques were not able to give great result. Additionally these strategies were not able to handle the huge search space.

1.3. Problem Statement

The aim of this work is to implement Groundwater Possibility Retrieval System to detect the groundwater possibility at various geographical regions using the technique of Grey Wolf Optimizer integrated with case based reasoning. Grey wolf Optimizer is a new metaheuristic algorithm which came up in last year. Though lots of metaheuristic algorithm exists like Biogeography based Optimization(BBO), Particle swarm Optimization(PSO), Cuckoo Search(CS) but the main advantage of grey wolf optimizer is that its search space is broad in nature. Because of its fresh introduction it scarcely has any footprints in any application. There are two techniques which are integrated together i.e. Grey Wolf Optimizer and Case base reasoning and then both of them are applied to identify the vicinity of ground water in a particular region as queried by user. Case based Reasoning is an approach that learns from past and can imitate the human behavior learning. The problem statement can be stated as:

"Detecting groundwater possibility at various geological portions using Grey Wolf Optimizer integrated with Case based reasoning."

Therefore we can utilize our proposed work in any real life issue where groundwater discovery is important.

1.4. Scope of the work

Our approach is to use *Grey Wolf Optimizer in Groundwater Exploration* to able to perform some very important targets successfully completed.

Expert has given us data containing the main attributes which are necessary to know about the groundwater possibility. The attributes are generally geology, landform, land use, soil, lineament, and slope. Essentially the force qualities are contracted down to High, Moderate and Low. Depending on their value experts say the presence of groundwater in that region.

For new case, Case base reasoning finds out the probable solution from the case base. These cases then reduced to find out the best solution using Grey Wolf Optimizer. Hence Scope of the work is as follows:

1. Adapting Grey wolf optimizer for ground water detection integrated with case base reasoning.
2. Apply it to the data set which has been given by expert.
3. Compare the results of Grey Wolf Optimizer with Cukoo search.

It is a bio-inspired groundwater potentiality that detects the presence of groundwater without even manual intervention. Geologist, by using our system can understand the percentage of groundwater of a remote place without even going to such places.

1.5. Organization of thesis

The remainder part of thesis is described in the following sections:

Section 2: This section gives the brief description of Computational intelligence that contains Biogeography based optimization (BBO), particle swarm optimization(PSO), and Cuckoo Search(CS).

Section 3: This section gives brief introduction about the case based reasoning (CBR). The component and features of CBR with its life cycle are discussed. The different techniques of case based reasoning are discussed.

Section 4: This section gives a detailed description of Grey wolf optimizer. It is the main algorithm that is used for ground water exploration. In this chapter, the behavior of Grey wolves that are apex predators has been introduced like how they search for the prey, etc.

Section 5: This section gives detailed description of combination of grey wolf optimizer along with case based reasoning (CBR). This section gives the proposed algorithm of Grey wolf Optimizer in ground water detection possibility. Various geological attributes are discussed which plays crucial role in the framework of ground water detection.

Section 6: This section provides the result of the ground water detection mechanism. It shows some layout of our software that is developed for groundwater potentiality in a particular region and our software has shown correct results in almost all cases as provided by the user.

Section 7: In this section, the conclusion of the thesis work and the future scope of the work are presented.

Section 8: This section gives the references details of the thesis.

Chapter 2: COMPUTATIONAL INTELLIGENCE

This chapter presents brief description of different computational intelligence algorithm that has been used for ground water detection.

2.1 Particle Swarm Optimization

Particle swarm optimization(PSO) is a populace based stochastic improvement procedure introduced by Dr. Eberhart and Dr. Kennedy in 1995, enlivened by social conduct of bird rushing or fish schooling.

PSO offers numerous similitude's with evolutionary computation approaches, for example, Genetic Algorithms (GA). The framework is instated with a populace of random solutions and scans for optima by upgrading eras. Be that as it may, not at all like Genetic Algorithm, PSO has no development operators, for example, mutation and crossover. In PSO, the potential results, called particles, fly through the issue space by taking after the present ideal particles.

Every particle stays informed concerning its coordinates in the problem space which are connected with the best solution (fitness) it has accomplished as such. (The fitness value is likewise put away.) This worth is called pbest. Another "best" esteem that is followed by the particle swarm optimizer is the best esteem, got so far by any particle in the neighbors of the particle. This area is called lbest. At the point when a particle takes all the populace as its topological neighbors, the best esteem is a global best and is called gbest.

The particle swarm optimization idea comprises of, at every time step, changing the speed of (accelerating) every particle toward its pbest and lbest areas (neighborhood adaptation of PSO).

Acceleration is weighted by a random term, with separate random numbers being created for acceleration toward pbest and lbest areas.

In recent years, PSO has been effectively connected in numerous exploration and application areas. It is exhibited that PSO shows signs of improvement results in a speedier, less expensive path compared with other methods.

Another reason that PSO is appealing is that there are couples of parameters to change. One form, with slight varieties, functions admirably in a wide assortment of uses. Particle swarm optimization has been utilized for methodologies that can be used over an extensive variety of applications, and in addition for particular applications concentrated on a particular prerequisite. PSO is additionally utilized for detecting ground water. As in case of ground water detection, after the clustering of geographical attributes using some case based algorithm, ranking of best and optimum solution is done by particle swarm optimization. This is how we can use particle swarm optimization for detecting ground water[5].

2.1.1. PSO Algorithm for ground water detection

1. Initialize the fitness value of each particle is evaluated and stored.

2. Number of waves is equivalent to number of iterations.

3. When new one's(wave) arrives then evaluates the fitness value of particle in the new wave and further compares it with particle at the corresponding location in the previous one.

4. If the fitness value of current one is better than the best one then set *pbest* as best value(fitness) and discard other ones which are not closer to fitness function.

5. Finally choose the particle with fitness value that is best amongst all as the *gbest*.

6. Process continues till fitness function value is fully specified no of iterations is completed.

7. End.

2.2 Biogeography Based Optimization

Biogeography is the study of the geological circulation of organic living beings(biological organisms). The attitude of the researcher is that we can gain from nature[2]. This spurs the use of biogeography to optimization problems. The arithmetic of biogeography as the premise for the improvement of another field: Biogeography based Optimization has been included. Researchers have introduced the idea of integration of Biogeography based optimization(BBO)[4] with case based reasoning(CBR) retrieval to build up a data retrieval framework. Utilizing BBO incorporated with case based technique; we will recover the data in regards to groundwater probability in a given region. Biogeography is the study of geographic appropriation of natural life forms. The study of Biogeography can be followed to the work of century patriots such as Algred Wallace [2005] and Charles Darwin[1995].

Biogeography based optimization is an application of biogeography to optimize problem. It is designed according to the migration and displacement of species between the islands. It is practically equivalent to general problem solutions. Geological regions that are appropriate as living arrangements for organic species are known as high habitat suitability index (HSI) [T. Wesche, G. Goertler and W. Hubert (1987)]. Highlights that correspond with HSI incorporate such components as rainfall, differences of vegetation, assorted qualities of topographic features, land range, and temperature. The variables that portray habitability are known as SIV(suitability index variable). Suitability index variable can be viewed as the independent variables, and habitat suitability index can be viewed as the dependent ones. High habitat suitability index habitats have a tendency to have a substantial no of species, while those with a low habitat suitability index have a little no of species. Habitats with a high habitat suitability index have numerous species that emigrate to close-by habitats, basically by virtue of the expansive number of species that they have. Habitats with a high habitat suitability index have a low migration rate in light of the fact that they are as of now soaked with species; high habitat suitability index habitats are much static in their species conveyance than low ones. Hence By the same token, high habitat suitability index habitats have a high emigration rate; the huge to emigrate to neighboring habitats number of species on high habitat suitability index islands has numerous open doors. A decent solution closely resembles an island with a high habitat suitability index, and a poor solution speaks to an island with a low habitat suitability index. High habitat suitability index solutions oppose change more than low HSI. By the same token, high HSI have a tendency to impart their elements to low HSI. Poor solutions acknowledge a ton of new components from good solution. This expansion of new elements to low habitat suitability index solutions may raise the nature of those solutions. This well known approach is called as BBO introduced by Dan Simon 2008[2].

In order to find out possibility of ground water, a combination of Biogeography based optimization with Case base Reasoning plays a decisive role. In this algorithm, clustering is done by case based reasoning where number of problem solutions is equivalent to number of habitats (classification of case base) [4]. Then according to the problem, isolated habitat and habitat suitability index is defined by assuming similarity threshold (attributes matched). Finally, solution of best species is then proceeding to an input species as optimum solution. This is how we can use BBO for detecting ground water.

2.2.1. BBO Algorithm for ground water detection

1. Classify the case base into habitats where Number of habitats is equal to number of problem solutions.

2. Define Isolated habitats in which the input one is placed.

3. Define Habitat Suitability Index(HSI).

   Suppose *a* as the similarity threshold

   For each habitat

     For each species

       If (HIS = Maximum (*a*))

   Similar Emigrate species to be put in Isolated habitat.

     End

   End

4. Remove less valuable species. Now apply case retrieval method on isolated ones to retrieve the best match to the input ones.

5. Finally Solution of best ones is transferred to an input one as its solution class.

6. End.

2.3 Cuckoo Search

Cuckoo Search is based upon the conduct of specific types of cuckoo which lay their eggs in the homes (nests) of different winged animals in a parasitic way. In the event that the egg adjust the qualities of the host settles then the eggs will survive generally the egg will be tossed. The procedure of advancing to best lay parasitic eggs is the principle point of Cuckoo search. In Cuckoo search the arrangement is spoken to by a nest. The algorithm is essentially having two operations. In the first operation another home (nest) is created. The new home is then assessed. The new home is at that point contrasted and the present home picked indiscriminately. This arbitrariness is finished by Lévy flight. The present best homes i.e. quality arrangements are kept and the most exceedingly awful homes are tossed. In the second operation the worst homes are expelled with probability P and from the quality arrangements locate the best arrangement [1].

2.3.1. Cuckoo Search Algorithm for ground water detection

1. Produce objective function of each put away cases for the situation base f(x), x={x1, x2,… … xd}. Take issue case data from the client (say i).

2. Introduce Count_case_base to 1 and Introduce total_cases.

3. While (count_case_base <= total_cases)

   {

Consider the present case (home) from the case base (Say j).

Assess Similarity (Fi, Fj) = i

Store the likeness estimation of Fj in another database named Temp_case_base

Count_case_base = count_case_base + 1

}

4. Store the cases in final_Case_base having the most extreme qualities from temp_case_base
5. Keep all these greatest qualities (best arrangements)
6. Ignore_Cases = Total_cases – final_case_base
7. Portions (ignore_cases) which are most noticeably awful homes are surrendered.
8. Rank the arrangements/ cases in final_case_base

{

Discover a relationship between the issue cases and the cases in final_Case_base

The greatest estimation of connection capacity is positioned as best arrangement.

}

9. The case with most noteworthy connection quality is considered.
10. Postprocess results

{

Apply all conceivable propositional rationale condition.

Contingent upon the condition the result can be {High, Moderate or Low}

} End.

Chapter 3: GREY WOLF OPTIMIZER

This work proposes another rising meta-heuristic called Grey Wolf Optimizer (GWO) motivated by grey wolves (Canis lupus). The Grey wolf Optimizer algorithm impersonates the administration progression and chasing system of grey wolves in nature. There are four sorts of grey wolves, for example, alpha, beta, delta, and omega are utilized for reenacting the initiative chain of command. Furthermore, the three fundamental strides of chasing, hunting down prey, enclosing prey, and assaulting prey, are actualized. The technique is then benchmarked on twenty nine test parameters, and the outcomes are checked by a relative study with, Gravitational Search Algorithm (GSA), Evolutionary Programming (EP), Particle Swarm Optimization (PSO), Differential Evolution (DE) and Evolution Strategy (ES)[3].

Fig 1. Grey Wolves

3.1. INSPIRATION

Fundamentally Grey wolf (Canis lupus) fits in with Canidae group. They are considered as pinnacle predator, which implies that they are at the highest point of the natural food chain. Generally grey wolves like to live in a pack. The gathering size is 5–12 overall. Exceptionally compelling is that they have an extremely strict social prevailing progression as demonstrated in Fig 2[3].

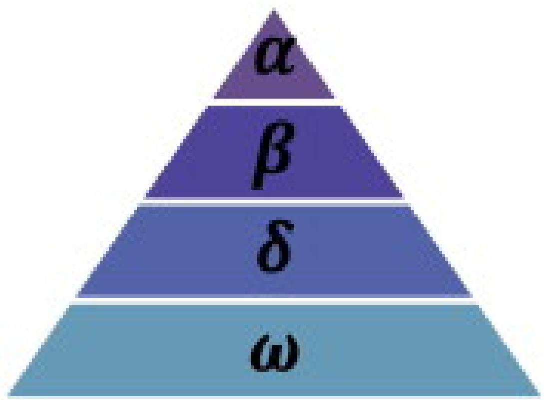

Fig 2. Hierarchy of Grey Wolf

In the gathering of Grey wolves, the pioneers can be male and a female, called Alphas. Alphas are leaders. The alpha is for the most part in charge of settling on choices about chasing, resting spot, time to wake, etc. The alpha's choices are managed to the pack. In any case, a just conduct has likewise been seen, in which an alpha takes after alternate two-timers. In the group, the whole pack recognizes the alpha by holding their tails down. The alpha ones are additionally called the prevailing wolf since his/her requests ought to be trailed by the pack. The alpha wolves are just permitted to mate in the group. Interestingly, the alpha is not so much the most grounded individual from the pack but rather the best regarding dealing with the pack. This demonstrates that the association and control of a pack is considerably more vital than its quality.

Accordingly, Alpha wolves are the prevailing two-timer. Thus they are at the top in the chain of command of dark wolves.

The second level in the chain of command of wolves is Beta. The betas are subordinate ones that help the alpha in choice making or other pack exercises. The beta wolf can be either female or male, and he or she is most likely the best possibility to become the alpha. They obtain the position of alphas in the event that one of the alphas passes away or turns out to be extremely old. The beta wolf ought to regard the alpha, however orders the other lower-level wolves too. It assumes the part of a consultant to the alpha and the discipliner for the pack. The beta fortifies the alpha's orders all through the pack and offers input to the alpha.

The most reduced in the progression of grey wolf is omega. The omega assumes the part of substitute. Omega wolves dependably need to submit to the various overwhelming wolves. They are the last ones that are permitted to eat. It may appear the omega is not a vital individual in the pack, but rather it has been watched that the entire pack face inside battling and issues if there should arise an occurrence of losing the omega. This is because of the venting of viciousness and dissatisfaction of all wolves by the omega. This helps fulfilling the whole pack and keeping up the predominance structure. At times the omega is additionally the sitters in the pack[3].

The third position in the pecking order goes to Delta. On the off chance that a wolf is not an alpha, beta, omega, then he/she is called as subordinate. Delta wolves need to submit to alphas and betas, however they overwhelm the omega. Scouts, sentinels, seniors, seekers, and overseers have a place with this classification. Scouts are in charge of viewing the limits of the domain and cautioning the pack in the event of any peril. Sentinels ensure and guarantee the well being of the pack. Seniors are the accomplished wolves who used to be either alpha or beta. Seeker's help the alphas and betas when chasing prey and giving food to the pack. At long last, the guardians are in charge of watching over the frail, sick, and injured wolves.

In the social order of wolves, group chasing is another intriguing social conduct of grey wolves. As per Muro et al., the fundamental periods of grey wolf chasing are as follows [3]:

- Tracking, pursuing, and drawing closer the prey.
- Pursuing, circling, and pestering the prey until it quits moving.
- Attack towards prey.

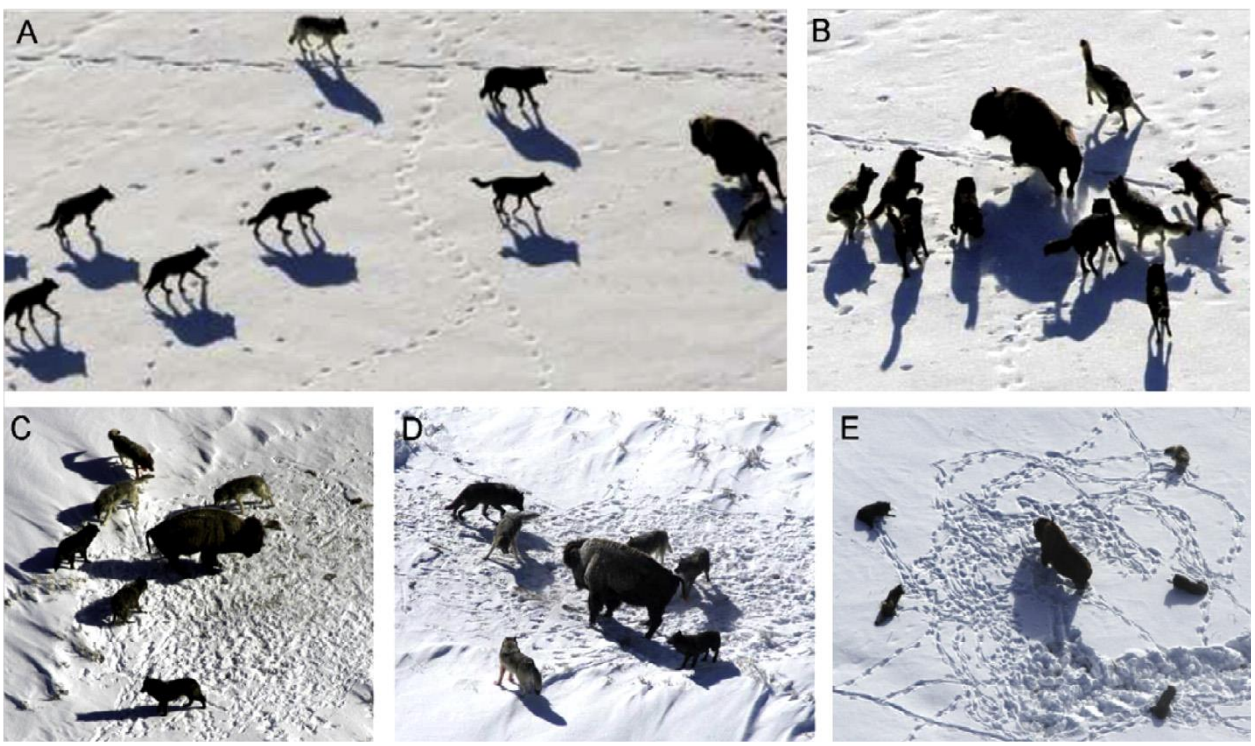

Fig 3. Hunting Behaviour of Grey Wolves

These above steps are shown in Fig 3[3]. In this work this chasing method and the social pecking order of grey wolves are numerically displayed keeping in mind the end goal to outline GWO and perform enhancement.

3.2. Mathematical model and algorithm

In this section the mathematical models of the social hierarchy, tracking, circling, and assaulting prey are provided.

3.2.1. Social hierarchy

In the social progressive system of wolves when planning GWO, we consider the fittest arrangement as the alpha (α). Thus, the second and third best arrangements are named beta (β) and delta (δ) separately. Whatever is left of the applicant arrangements are thought to be omega (ω). In the GWO technique, the chasing (hunting) is guided by α, β, and δ. The ω wolves take after these three wolves[3].

3.2.2. Encircling prey

As mentioned above, during the chase, wolves encircle prey [3]. In the mathematically model encircling conduct the following equations are proposed:

$$\mathbf{D = | C. X_p (t) - A. X (t)|}$$

$$\mathbf{X(t+1) = X_p(t) - A. D}$$

where t demonstrates the current iteration, A and C are coefficient vectors,

Xp is the position vector of the prey, and X demonstrates the position vector of a grey wolf.

The vectors A and C are evaluated as follows:

$$\mathbf{A = 2a \cdot r_1 \cdot a}$$

$$\mathbf{C = 2 \cdot r_2}$$

where components of a are straightly diminished from 2 to 0 through the span of iterations and r1, r2 are arbitrary vectors in [0, 1].

To see the impacts of initial two mathematical statements, a two-dimensional position vector and a portion of the conceivable neighbors are given in Fig. 4(A). As can be found in this figure, a grey wolf can overhaul its position(X, Y) as indicated by the position of the prey (X*, Y*). Better places around the best agents can be come to regarding the present position by conforming the estimation of A and C vectors. For example, (X*–X, Y*) can be come to by setting A=(1,0) and C=(1,1). The possible upgraded positions of a grey wolf are illustrated in Fig. 4(B). Note that the arbitrary vectors r1 and r2 permit wolves to achieve any position between the points showed in Fig. 3. So a grey wolf can overhaul its location inside the area around the prey in any irregular area by using first two equations.

The same idea can be stretched out to an inquiry space with m dimensions and the wolves will proceed in hyper-solid (or hyper-cubes) around the best solution got as such.

3.2.3. Hunting

Grey wolves can perceive the area of prey and enclose them. The chase is normally guided by the alpha wolves. The beta and delta may likewise take part in chasing occasionally. On the other hand, in an abstract search space we don't have clue about the area of the ideal (prey). With a specific end goal to mathematically simulate the chasing conduct of grey wolves, we assume that the alpha (best solution), beta, and delta have better learning about the potential area of prey. Hence, we spare the initial three best solutions got so far and oblige the other agents (counting the omegas) to redesign their positions as per the position of the best hunt agents. The following formulas are proposed in this context.

$$\mathbf{D_\alpha = |\ C_1 \ .\ X_\alpha - X\ |,}$$

$$\mathbf{D_\beta = |\ C_2 \ .\ X_\beta - X\ |,}$$

$$\mathbf{D_\delta = |\ C_3 \ .\ X_\delta - X\ |,}$$

$$\mathbf{X_d = X_\alpha - A_1 \ .\ (D_\alpha),}$$

$$\mathbf{X_e = X_\beta - A_2 \ .\ (D_\beta),}$$

$$\mathbf{X_f = X_\delta - A_3 \ .\ (D_\delta),}$$

$$\mathbf{X\ (t+1) = (\ X_d + X_e + X_f)\ /\ 3,}$$

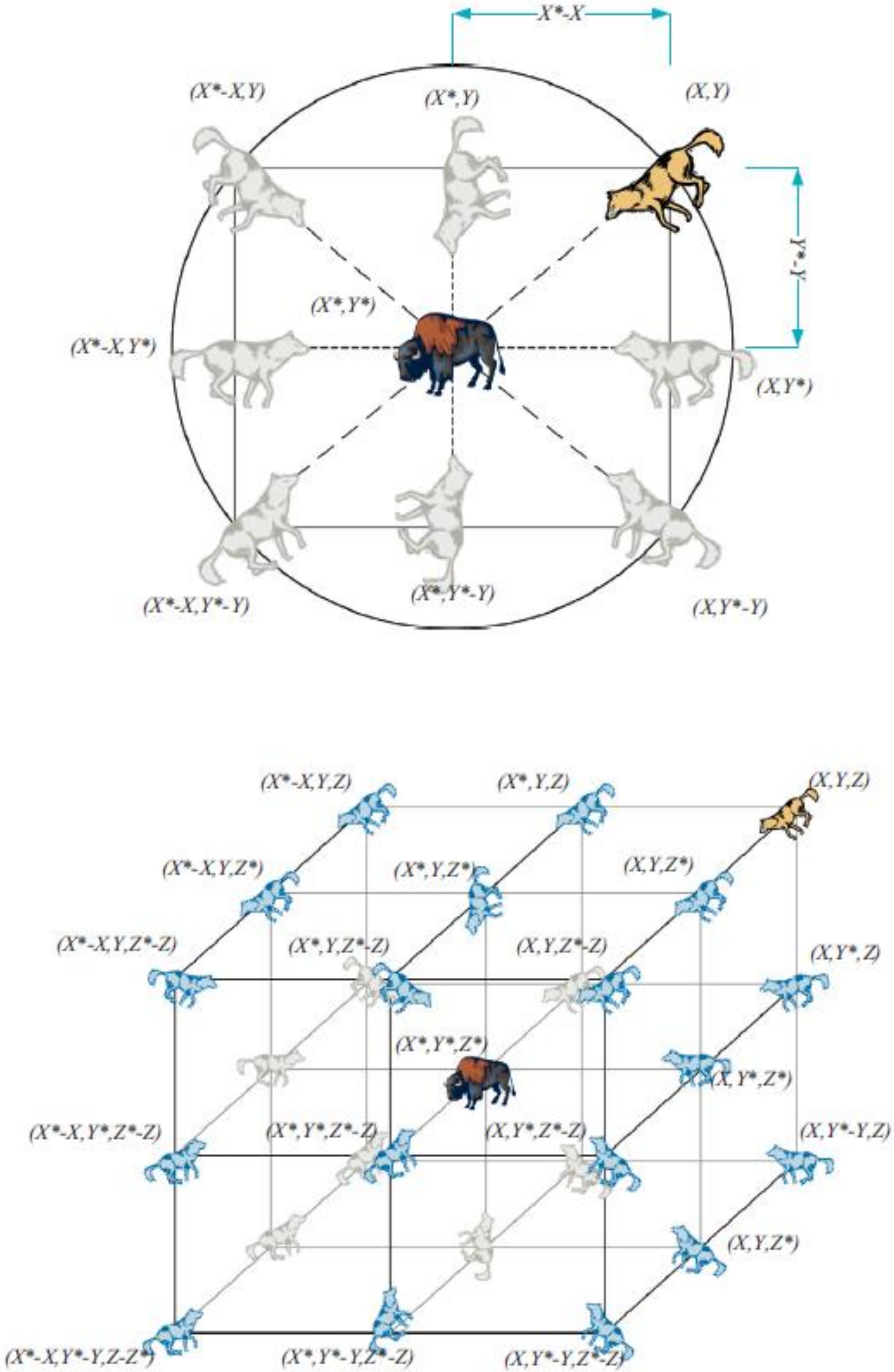

Fig 4. 2D & 3D position vectors and their possible next locations.

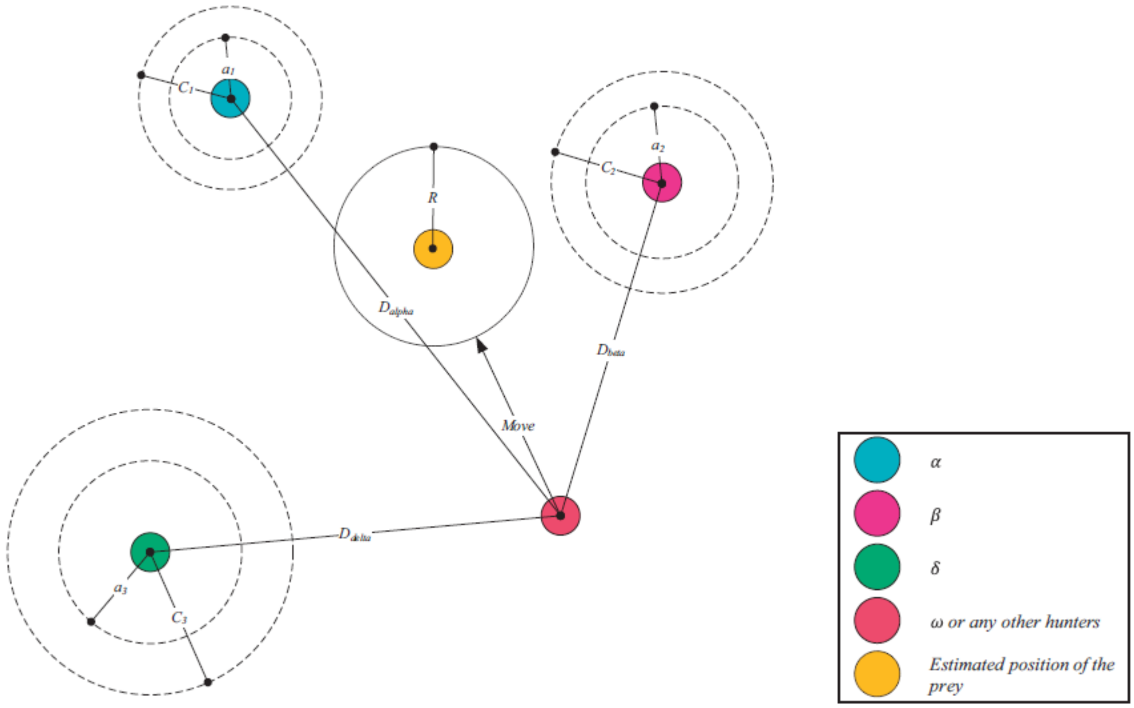

Fig 5. Position updating in GWO.

Fig. 4 and Fig. 5[3] shows how a search agent upgrades its position as per alpha, beta, and delta in a two dimensional search space. It can be watched that the last position would be in an arbitrary place inside of a circle which is characterized by the location of alpha, beta, and delta wolves in the pursuit space. However, alpha, beta, and delta appraise the position of the prey, and other grey wolves overhaul their positions arbitrarily around the prey.

3.2.4. Attacking prey (exploitation)

As said over the grey wolves complete the chase by assaulting the prey when it quits moving. According to mathematical model, drawing nearer the prey we diminish the estimation of a. Note that the fluctuation range of A is additionally diminished by a. However, A will be an arbitrary value in the interval [-2a, 2a] where a is diminished from 2 to 0 through the span of iterations. When random values of A are in [-1, 1], the next location of a search agent can be in any position between its present position & position of the prey.

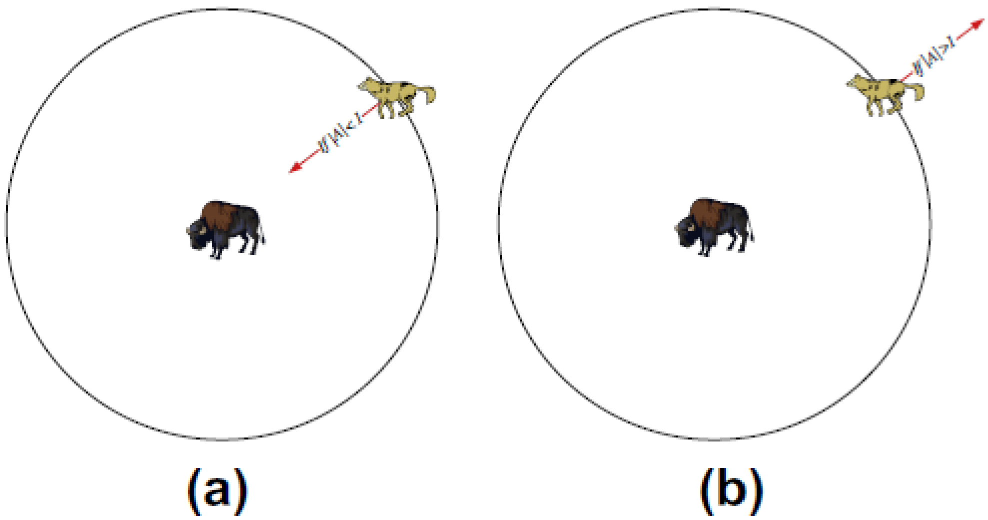

Fig 6. Attacking prey versus searching for prey.

Fig. 6(a)[3] demonstrates that |A| < 1 powers the wolves to assault towards the prey. With the operators proposed as such, the GWO algorithm permits its search agents to redesign their position taking into account the area of the alpha, beta, and delta; and assault towards the prey. Thus, the GWO algorithm is inclined to stagnation in neighborhood solutions with these operators. Circling mechanism shows investigation to some degree, yet GWO needs more emphasize to underscore investigation [3].

3.2.5. Search for prey (exploration)

Grey wolves search as per the current position of the alpha, beta, and delta wolves. They separate from one another to hunt down prey and join to assault prey. In the mathematical model, we use A with arbitrary values more than 1 or not exactly -1 to oblige the search agent to veer from the prey.

This stresses investigation and permits the GWO algorithm to pursuit all inclusive. Fig. 6(b) likewise demonstrates that $|A| > 1$ strengths the grey wolves to veer from the prey to ideally locate a fitter prey. Another segment of GWO that supports investigation is C. The C vector contains irregular values in [0, 2]. This segment gives irregular weights to prey keeping in mind the end goal to stochastically underline ($C > 1$) or deemphasize ($C < 1$) the impact of prey in characterizing the separation in. This helps GWO to demonstrate a more arbitrary conduct all through optimization, favoring investigation and neighborhood optima avoidance. It merits saying here that C is not directly diminished as opposed to A.We intentionally oblige C to give arbitrary qualities at all times to underscore investigation not only during initial one but also final iteration.

This part is extremely useful if there should arise an occurrence of nearby optima stagnation, particularly in the last iterations. The C vector can be additionally considered as the impact of obstacles to drawing nearer prey in nature. As a rule, the obstacles in nature show up in the chasing ways of wolves and drawing closer prey. This is precisely what the vector C does.

According to the position of a wolf, it can arbitrarily give the prey a weight and make it harder and more remote to reach for wolves, or vice versa[3].

To entirety up, the search procedure begins with making an arbitrary populace of grey wolves in the GWO algorithm. Through the span of iterations, alpha, beta, and delta wolves gauge the likely position of the prey. Every competitor solutions redesign its separation from the prey. The parameter a is diminished from 2 to 0 to underline investigation and abuse, respectively. Candidate solutions have a tendency to veer from the prey when $|A| > 1$ and merge towards the prey when $|A| < 1$. At last, the GWO algorithm is ended by the fulfillment of an end standard.

Chapter 4: CASE BASED REASONING

4.1. Case Based Reasoning

Case-based reasoning is a problem solving model that in numerous regards is generally not the same as other significant Artificial Intelligence approaches. This section depicts the ideas, segments and elements of CBR. CBR is authored by prominent information scientists, Professors S.K.Pal and S.Shiu. The classic definition of CBR was coined by Riesbeck and Schank[7]:

"A case-based reasnor solves problems by using or adapting solutions to old problems"

Hence, case based reasoning is method which tackles problem case in view of the cases put away for the case base. This method coordinates the problem case with case base and figure out an answer.

As opposed to depending exclusively on broad information of a problem space, or making relationship along generalized connections between problem descriptors and conclusions, CBR has the capacity use the particular learning of already experienced, solid problem cases. Another problem is settled by discovering a comparative past case, and reusing it in the new problem circumstance. A second imperative difference is that CBR additionally is a way to deal with incremental, sustained learning, since another experience is held every time an issue has been solved, making it instantly accessible for future issues. The CBR field has developed quickly in the course of the most recent couple of years, as seen by its expanded offer of papers at major conferences.

4.2. Components and Features of Case-Based Reasoning

The structure of a case-based reasoning framework is normally contrived in a way that reflects separate stages: retrieval and adaptation. However at the highest level case-based reasoning can be seen as secret elements as delineated in figure 7,8 that joins the reasoning system and the accompanying outside facets[7]:

- The input specification or problem case.
- The output that characterize a recommended solution to the problem.
- The memory of past cases, the case base, that are referred by the reasoning method.

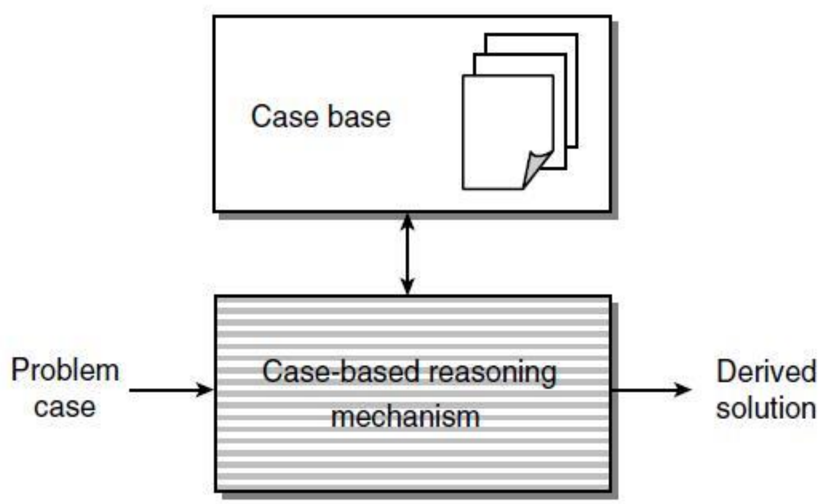

Fig 7. CBR System

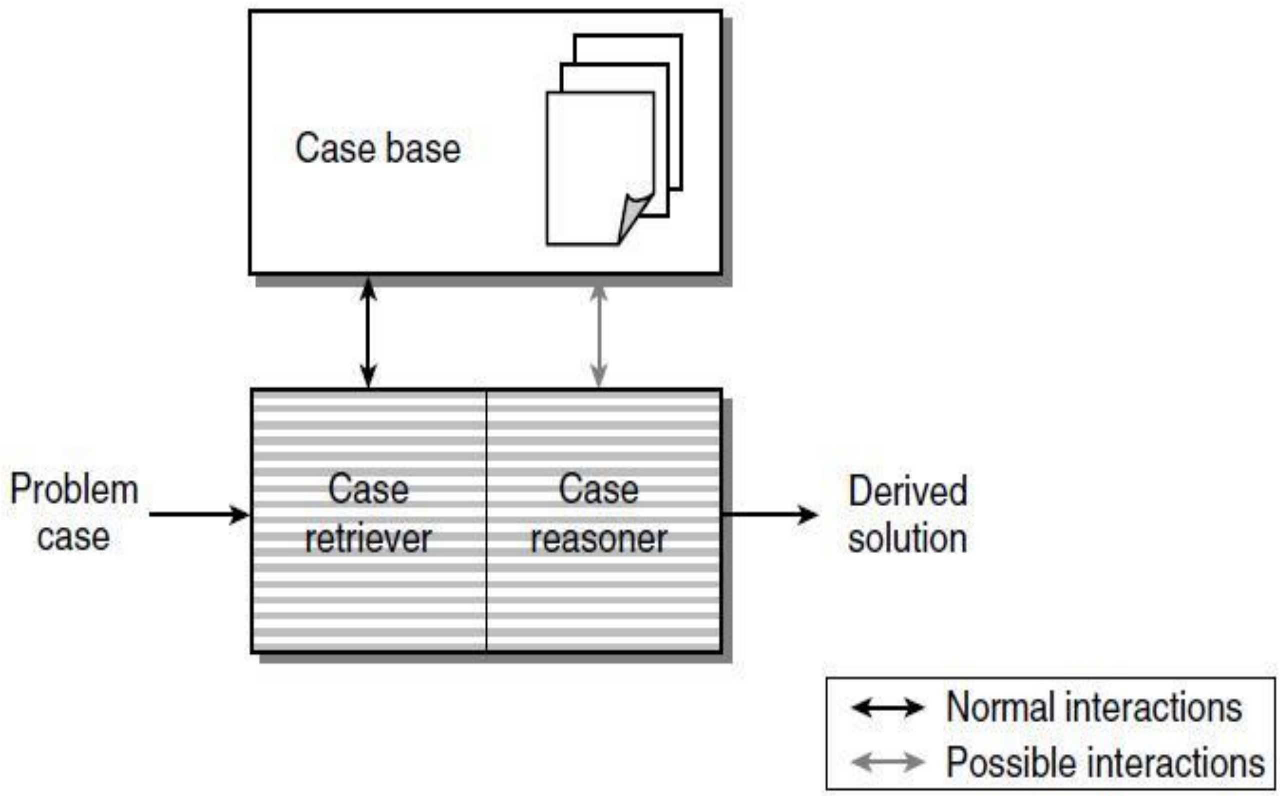

Fig 8. Two Major components of CBR[7].

In most case-based reasoning frameworks, the case-based reasoning mechanism, referred to as the issue solver or reasoner, has an inner structure separated into two noteworthy parts: the case retriever and case reasoner. The case retriever's task is to locate the proper cases for the case base while the case reasoned uses the case retrieved to discover a solution for the problem given. This reasoning process includes both deciding the contrasts between the cases recovered and the present case, and adjusting the solution for mirror these distinctions suitably. The reasoning procedure might include recovering extra cases or segment of cases from the case base.

4.3. CBR Life Cycle

The problem-solving life cycle in a CBR system consists essentially of the following four parts as shown in figure 9[7]:

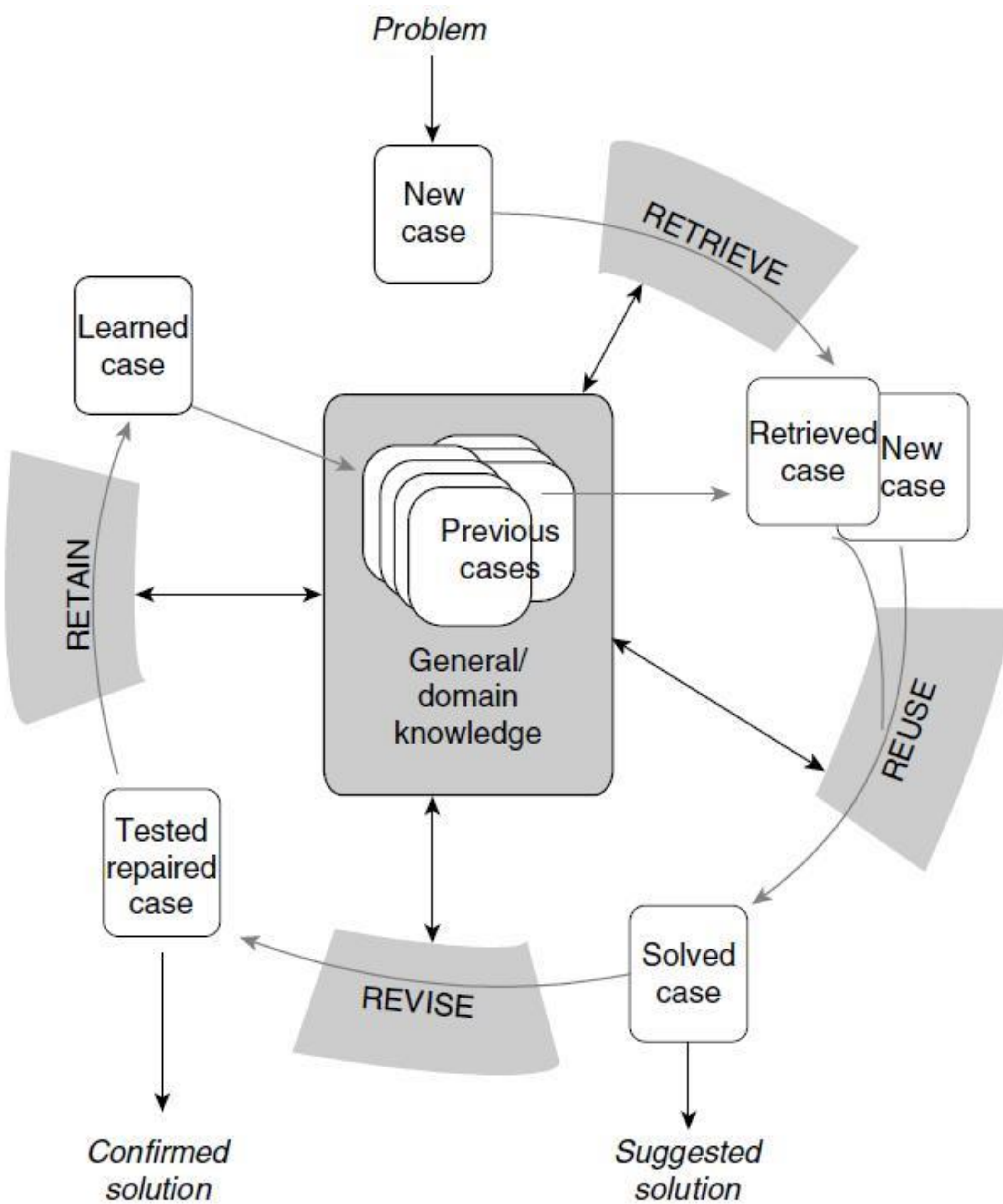

Fig 9. CBR Cycle

Case-based reasoning has been formalized for purposes of computer reasoning as a four-step process[8]:

- **Retrieve:** comparative already experienced cases (e.g., problem solution outcome triples) whose problem is judged to be comparable as given a target problem, recover from memory cases applicable to comprehending it. A case comprises of a problem, its solution, and, regularly annotations about how the solution was inferred. Let us assume Fred needs to get ready blueberry hotcakes. Being a learner cook, the most significant experience he can review is one in which he effectively made plain hotcakes. The methodology he took after for making the plain pancakes, together with legitimization for choices made along the way, constitutes Fred's retrieved case.
- **Reuse:** the cases by duplicating or coordinating the solutions from the cases recovered. Map the solution from the past case to the objective problem. This may include adjusting the solution as expected to fit the new circumstance. In the hotcake case, Fred must adjust his recovered answer for incorporate the expansion of blueberries.
- **Revise:** adjusting the solution(s) recovered trying to tackle the new issue. Having mapped the past solution to the objective situation, test the new solution in this present reality and, if essential, reconsider (revise). Assume Fred adjusted his hotcake arrangement by adding blueberries to the batter. In the wake of blending, he finds that the batter has turned blue – an undesired impact. This proposes the accompanying revision: defer the expansion of blueberries until after the batter has been spooned into the pan.

- **Retain:** new solution once it has been affirmed or approved. After the solution has been effectively adjusted to the objective problem, store the subsequent experience as another case in memory. Fred, as needs to be, records his recently discovered technique for making blueberry hotcakes, in this way improving his arrangement of put away encounters, and better setting him up for future pancake making requests.

In numerous practical applications, the reuse and revise stages are at times hard to recognize, and a few researchers utilize a single adaptation level that replaces and consolidates them. Hence, adaptation in case based reasoning frameworks is still an open inquiry on the grounds that it is a muddled procedure that tries to control case arrangements. Normally, this requires the advancement of a causal model between the problem space and the solution space of the related cases.

4.4. Application of CBR

Case based reasoning solves real-world problems. Thus this section provides an overview how case based reasoning solves real life problems using four different techniques.

4.4.1. CBR using nearest neighbour

Nearest neighbor strategies are maybe the most generally utilized innovation as a part of Case based reasoning since it is given by the lion's share of Case based reasoning tools. Nearest neighbor algorithm all work in a comparable manner. The likeness of the problem (target) case to a case for the case library for every case property(attribute) is resolved. This measure may be multiplied by a weighting variable. At that point the aggregate of the comparability of all attributes is estimated to give a measure of the similarity of that case in the library to the objective case. Algorithm like this is utilized by most case based reasoning devices to perform nearest neighbor retrieval. Similarities are typically standardized to fall inside of a scope of zero to one (where zero is absolutely unique and one is a definite match) or as a rate similarity where 100% is a precise match. The utilization of nearest neighbor is all around represented by the Wayland framework.

4.4.2. CBR using induction

Induction techniques are regularly utilized as a part of Case based reasoning since much portion of the all the more intense financially accessible Case based reasoning devices give this facility (e.g. Kate from AcknoSoft; ReCall from ISoft; CBR-Works from TecInno; and ReMind from Cognitive Systems). Induction algorithms, for example, ID3, assemble choice trees from case histories. The Induction techniques recognize designs amongst cases and partition the cases into clusters. Every cluster contains cases that are comparative(similar). A prerequisite of induction is that one objective case highlight is characterized (i.e. the component that the algorithm will impel). Basically the Induction techniques are being utilized as classifiers to cluster comparative cases together. It is expected (normally effectively) that cases with comparable problem descriptions will refer to comparative problems and subsequently comparable solutions. It is utilized as a part of Troubleshooting CFM 56-3 motors on Boeing 737s.

4.4.3. CBR using fuzzy logic

Fuzzy logics are a method for formalizing the typical handling of fuzzy linguistic terms, like, excellent, good, fair and poor, which are connected with contrasts in a attribute depicting a feature. Any number of linguistic terms can be utilized. Fuzzy logics inherently represents thoughts of similarity, since good is closer (more comparable) to excellent than it is to poor[8].

For CBR, a fuzzy inclination function can be utilized to ascertain the similarity of a single attribute of a case with the relating attribute of the objective. For example, in Fig. 10, a difference of 1 unit in the values of an attribute would be considered excellent, a difference of 2 would be good, 3 would be fair and 4 would be poor. The result of using fuzzy preference functions is a vector, called the fuzzy preference vector. The vector contains a fuzzy preference value for each attribute. The values in this vector can be combined, through weighted aggregation, to produce a robust similarity value. The use of fuzzy preference functions allows for smooth changes in the result when an attribute is changed unlike the large changes that are possible when step functions are used. A fuzzy preference function is used to transform a quantifiable value for each attribute into a qualitative description of the attribute that can be compared with the qualitative description of other attributes.

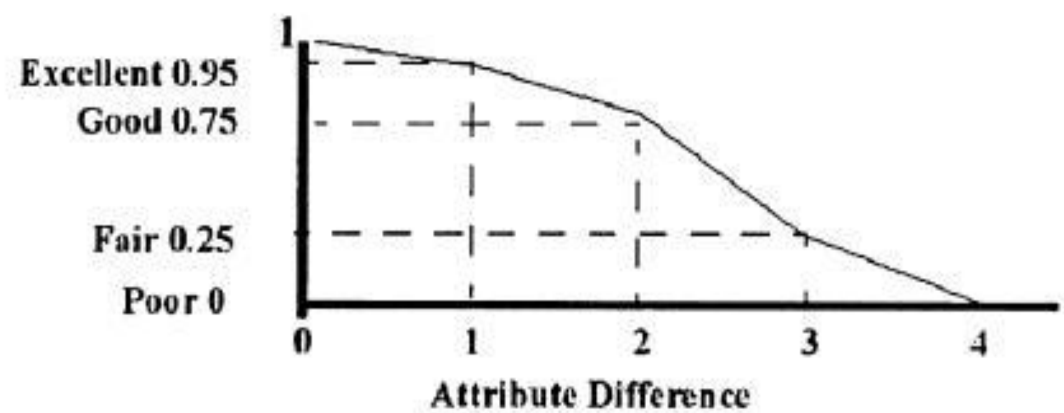

Fig 10. A fuzzy preference function

Thus, a fuzzy preference function allows a comparison of properties that are based on entirely different scales such as cost measured in cents per pound and spectral curve match measured in reflection units[7].

Chapter 5: Combination of Grey wolf Optimizer and Case Based Reasoning

This chapter deals with our proposed algorithm for exploration of groundwater possibility in any particular area. Our algorithm has shown quiet efficient results in exploration mechanism. Our experts had given some areas values and our proposed algorithm detected the presence/absence of groundwater in that particular area.

Basically Groundwater is located beneath the ground surface in soil spores spaces and in fractures of rock formation. Groundwater is essentially soil dampness, permafrost stationary water in low porousness bedrock and profound geothermal or oil development water. Under Groundwater existence depends on natural resources. Existence of various resources is necessary to find the potentiality of groundwater. In recent findings a lot of swarm intelligence techniques have been devised for detection of groundwater. Both Biogeography Based Optimization and Particle Swarm Intelligence had been used in these detection process but there some pitfalls in these algorithms. Grey wolf optimizer is able to remove all those pitfalls and came with quiet efficient results in detection process. GWO being a meta heuristic algorithm have a balanced layout in both randomization and intensification. It maintains a good balance of intensive local search (as its search space is extensive in nature) and then exploration of full search which leads to find the problem in most optimized way. But in case of BBO and PSO both searching space is narrowed down and thus the searching is concentrated on local optima.

Groundwater possibility varies with natural resources. The detection is quite variable thus several conditions are needed to be considered during their detection mechanism such as geology, lineament, land use. Soil, slope, and land form . For example geology (one of the most important attribute for groundwater detection) has high significance in detection whereas soil has least importance.

But GWO discussed almost all possible conditions where possibility of groundwater can be high, low and moderate. Our proposed algorithm combined both the strength of Case based reasoning and even GWO. Basically for the initial search we have used Case based using nearest neighbour to match the user input data with the stored database.

5.1. Geographical Attributes

The dataset used in our experiment consists of groundwater observations made by domain experts to predict the groundwater possibility. Experts have given us values of attributes of various regions. It is not concentrated to any particular area[1].

ATTRIBUTES	**VALUES**
GEOLOGY	Sedimentary, Younger alluvium, Older alluvium, Igneous, Metamorphic
LANDFORM	Floodplain, Intermontanevalley, Pediment, Alluvialfans, Bajada, Pediplain, Buriedpediment, AlluvialPlain, Deltaic Plain, Riverterraces, Oldmeander etc.
SOIL	Sandyloam, Sandygravel, Coarsesand, Clayloam, Alluvialsand, Gravelsand, Gravel Sand Pebbles, Sand, Rocky etc.
LANDUSE	Agricultural land, Forest, Cultivated Land, Fallowland, Waterbody, Wasteland, Swampy land, Build up, Urban, Grass, Shrubs, Mixed vegetation.
SLOPE	Gentle, Steep.
LINEAMENT	Absent, Present.

Table 1. Six attributes used in dataset

From the above table 1, it is noticed that generally 6 attributes are considered. Namely they are geology, landform, soil, land use, slope and lineament. Each attribute has their significance in detection. Our expert has ranked the attribute. The ranking denotes which attributes are most important gets Rank 1 that indicates this attribute existence or value has a great impact on the existence of groundwater in that region whereas rank 6 indicates that this attribute has least significance in groundwater potentiality. Thus the ranking is as shown in table 2.

RANK	**ATTRIBUTE**
1	**GEOLOGY**
2	**LINEAMENT**
3	**LANDFORM**
4	**SLOPE**
5	**LANDUSE**
6	**SOIL**

Table 2. Ranking of attributes

Each of the attributes is described in the following section[4]:

- **Geology:** is the science that contains the investigation of the Earth and the procedures by which it is molded and changed.
- **Lineament:** A lineament is a linear component in a landscape which is an outflow of a fundamental geological structure, for example, a fault. Ordinarily a lineament will contain a fault aligned valley, a progression of fault or fold-aligned hills, a straight coastline or to be sure a mix of these components. Crack zones, shear zones

and volcanic interruptions, for example, dykes can likewise offer ascent to lineaments.

- **Landform:** A landform or physical component of the landscape. Landform components additionally incorporate seascape and ocean water body interface elements, for example, peninsulas, bays, oceans etc, including sub-watery territory components, for example, submersed mountain ranges, volcanoes, and the considerable sea basins.
- **Land use**: Land use is the human use of land. Land use is the modification of natural environment or building something new in areas like fields, pastures and settlements.
- **Soil:** *S*oil contains various mineral constituents. It is a natural body consisting of various layers. Basically their morphological, physical, chemical and mineralogical characteristics vary from their parent materials.
- ***S*lope:** In geographical term slope is known as gradient. Gradient tells you how steep the route is. One normal approach to define the normal steepness or slope(gradient) between two points is to express the difference in elevation distance (ED) separated by the horizontal distance (HD):

$$\mathbf{GRAD} = ED / HD$$

5.2. Integration of Case Based Reasoning

In our application CBR has immense importance for retrieval of similar cases from the stored database. In this Case based reasoning system framework, land parameters and their corresponding solutions i.e. the possibility of groundwater (High, Moderate and Low) are put away as cases for the case base. By the trait significance, six attributes are chosen as inputs. The possibility of groundwater is chosen as the output choice attribute of the case. The cases are gathered from master knowledge [4]. The case is spoken to by the ordered pair C= <F,D> where F is the geographical parameter of case C i.e. F= { Geology, Landform, Soil, Lineament, Slope, Land type} and D is the decision attribute i.e. the existence groundwater in the particular region[1]. It is shown in table 3.

The case searching and matching is a key step in case retrieving and it directly influences the retrieval efficiency and accuracy. The case retrieval in essence is to find the most similar case in the case base to target case. The K-nearest neighbour is widely used for its advantage of clear physical concept and simple calculation for case searching. A sum of similarities is calculated according to the similarity between each feature in problem case and the cases in case base. Similarity index value is determined by the below mentioned equation.

$$\text{Similarity Index } (F_i, F_j) = \sum f(F_i, F_j) \text{ X } w_i \text{ for i=1 to n.}$$

Where Similarity (F_i,F_j) represents the similarity value of F_i and F_j. F_i is the problem case, F_j is the case stored in the case base, n is the number of attributes and w_i is weight of attribute.

GEOLOGY	LANDFORM	SOIL	LAND USE	SLOPE	LINEAMENT	SOLUTION
Sedimentary	Flood Plain	Sandy loam	Agricultural land	Gentle	Absent	High
Sedimentary	Intermountain Valley	Sandy gravel	Forest	Steep	Present	High
Sedimentary	Pediment	Coarse sand	Cultivated	Gentle	Present	Moderate
Sedimentary	Flood Plain	Sandy loam	Agricultural land	Gentle	Present	High
Younger Alluvium	Flood Plain	Sandy loam	Agricultural land	Gentle	Present	High

Table 3. Cases from the Case Base.

5.3. Steps of algorithm(GWD)

Begin

1. Set the objective function of each stored cases in the case base F(x) where x = 1,2,….136. // total cases as provided by experts = 136
2. Take input value from user and Initialize the count_case_base to 1.
3. Initialize total cases to 115. // we consider 115 cases out of 136 cases as given.
4. While (count_case_base <= total_case)

   { Consider the current case from the case base

   Find out the Similarity Index (F_i, F_j) = $\sum f(F_i, F_j) X w_i$ for i=1 to n. // we used K-nearest neighbor algorithm(KNN) to find the match between the set of cases and problem case in the case base.

   Where F_i= Problem Case

   F_j= Case stored in the case base

   N= No. of attribute(n=6)

   If (F_i=F_j) then f returns 1 otherwise 0.

   Then store the all the cases with similarity value 1 in the Temp_case_base.

   Count_case_base = Count_case_base + 1.

   }
5. Store the cases in the final case base from the temp case base.
6. For ranking the solution/cases in the final case_base.

   Initialize the GWO population i.e. final_case_base X_i (where i=1,2,3……m)
7. Initialize a, A and C. // A and C are coefficient vector that depends upon the value of a, r_1 and r_2(random vector). The value of A & C is identified by these equations i.e

$A = 2a.r_1 - a$ and $C = 2.r_2$ // $a \in [2,0]$ and $r_1, r_2 \in [0,1]$

8. Evaluate the fitness of each case X_i. // fitness function value of each case base is identified by these equations

$$D_i = | C.X_p(t) - X_i(t)| \quad \text{and} \quad X_i(t+1) = X_p(t) - A.D$$

Where t is current iteration, X_p is problem case and X is member of final case base

X_α = the best solution from the case base.

X_β = the second best solution from the case base.

X_δ = the third best solution from the case base.

9. While (t< Max number of iterations)

    For each case X_i

    $X_1 = X_\alpha - A_1. (D_\alpha)$,

    $X_2 = X_\beta - A_2. (D_\beta)$,

    $X_3 = X_\delta - A_3. (D_\delta)$

    $X_i(t+1) = (X_1 + X_2 + X_3)/3$ // Update the current case base.

    End for

    Update a, A and C.

    Evaluate the fitness of all case.

    Update X_α, X_β, X_δ // according to the fitness function value of each case

    t = t+1

    end while

10. return the best solution X_α.

    End

5.4. Detailed Description of algorithm

- In this step, fundamentally we need to figure out number of cases should be considered. Despite the fact that objective function must be figured out for every case for the problem base. In this way we have considered a sum of 115 cases for our case base. The ignored cases are required for validation process. A client input is taken. Clients are obliged to give their problem case to the framework. Subsequently client needs to enter the estimations of 6 attributes. Then the system shows up for each attribute and in that menu all the values are shown.
- A user input is taken and a counter is needed for staying informed regarding cases for the case base. In this manner we initialize a counter name count_case_base to 1. This counter essentially indicates the present case in the case base.
- In this step, we consider the aggregate cases(total) required for our processing purpose. In this way we instate a counter named total_case to 115.
- A while condition is utilized. This loop is ended until every one of the cases for the case base is not encountered. Here, we need to find out the value of Similarity index(F_i, F_j) where F_i stands for problem case and F_j stands for case stored in case base. We have used K-nearest neighbor(KNN) algorithm(using Euclidean distance) to find the match between set of cases in the case base and problem case. If the value of problem case and case stored is equal then store the similarity value in Temp case base. W_i is basically weight of attributes as shown in figure 11. In this figure, X axis represents the attributes and Y axis represents the weight of each attributes. According to the figure, it

understands that Geology attribute has more importance than others and Soil attribute has low significance.

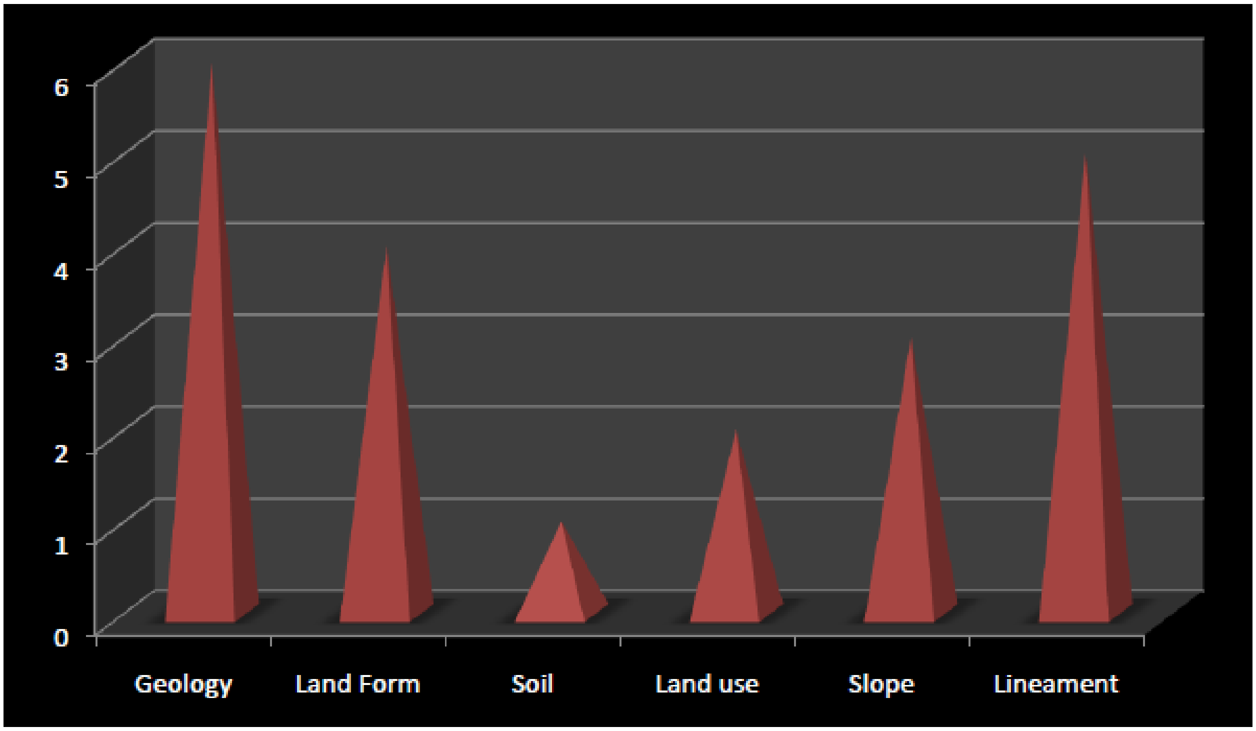

Fig 11. Weight of each attribute.

In the below mentioned figures, It shows the various attributes values. Figure 12 shows the Geology values in which Y axis represents the range of attributes values and X axis represents the value of attributes. It has attributes like Sedimentary, Younger alluvium, Older alluvium, Igneous, Metamorphic. Figure 13 shows the Landform attributes. It has attributes like Floodplain, Intermontanevalley, Pediment, Alluvialfans, Bajada, Pediplain, Buriedpediment, AlluvialPlain, Deltaic Plain, Riverterraces, Oldmeander etc. Figure 14 shows the Soil attributes. It has attributes like Sandyloam, Sandygravel, Coarsesand, Clayloam, Alluvialsand, Gravelsand, Gravel Sand Pebbles, Sand, Rocky etc. Figure 15 shows the Land use attributes. It has attributes like Agricultural land, Forest, Cultivated Land, Fallowland, Waterbody, Wasteland, Swampy land, Build up, Urban, Grass, Shrubs, Mixed vegetation.

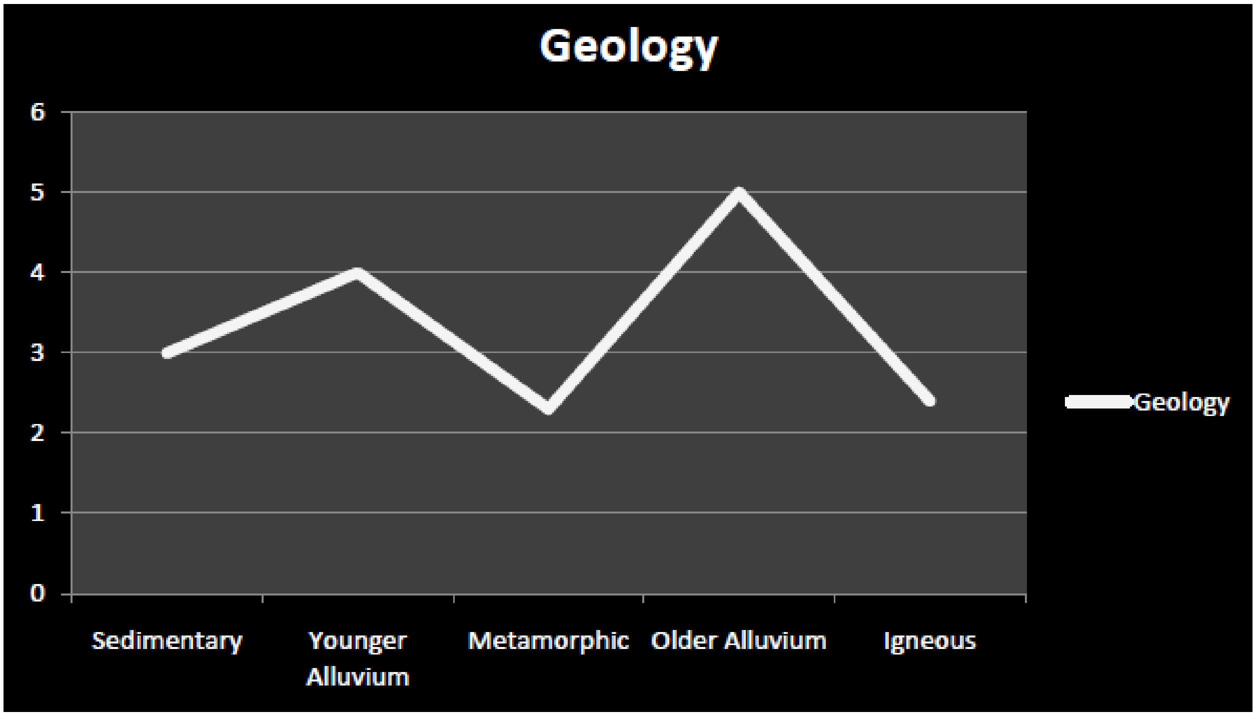

Fig 12. Geology attribute Values.

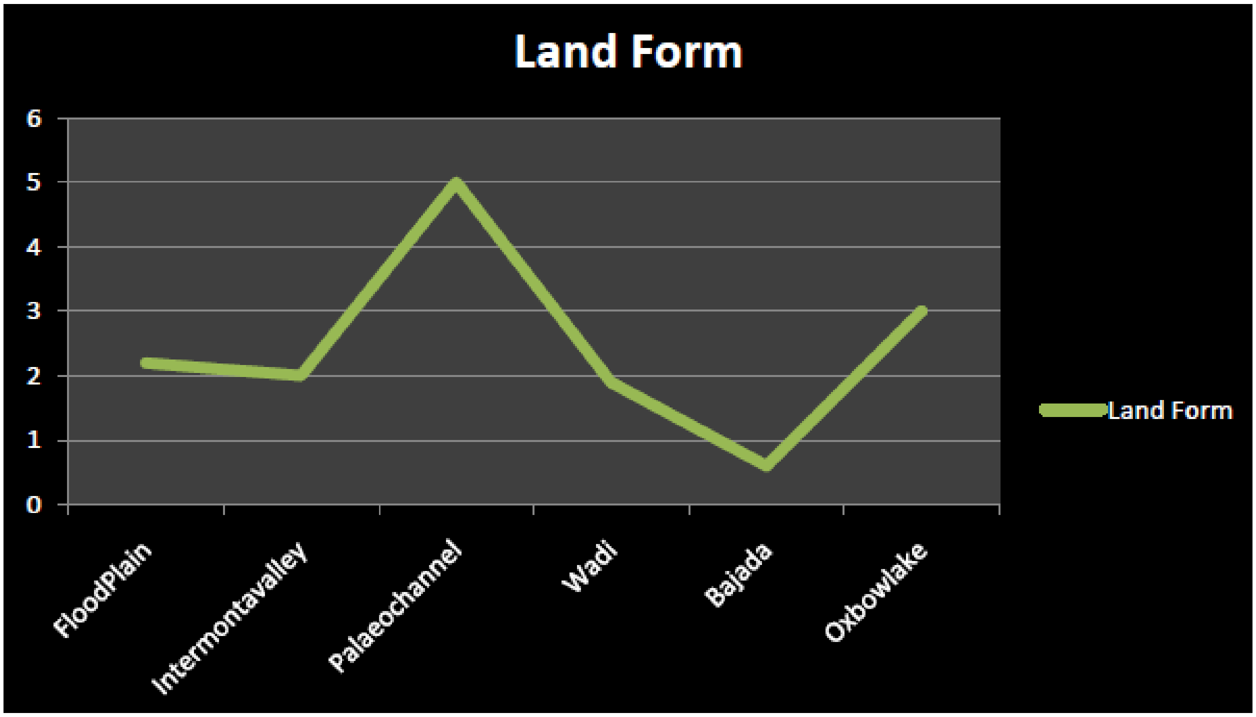

Fig 13. Land Form attribute values.

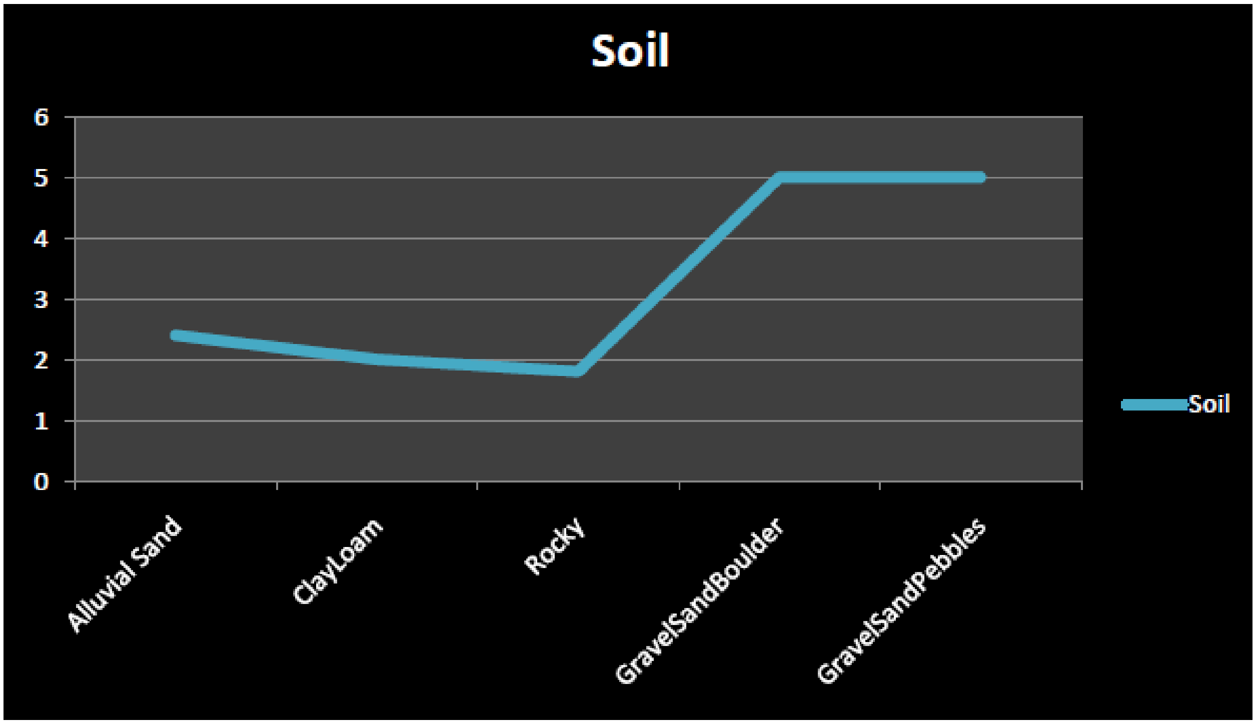

Fig 14. Soil attribute values.

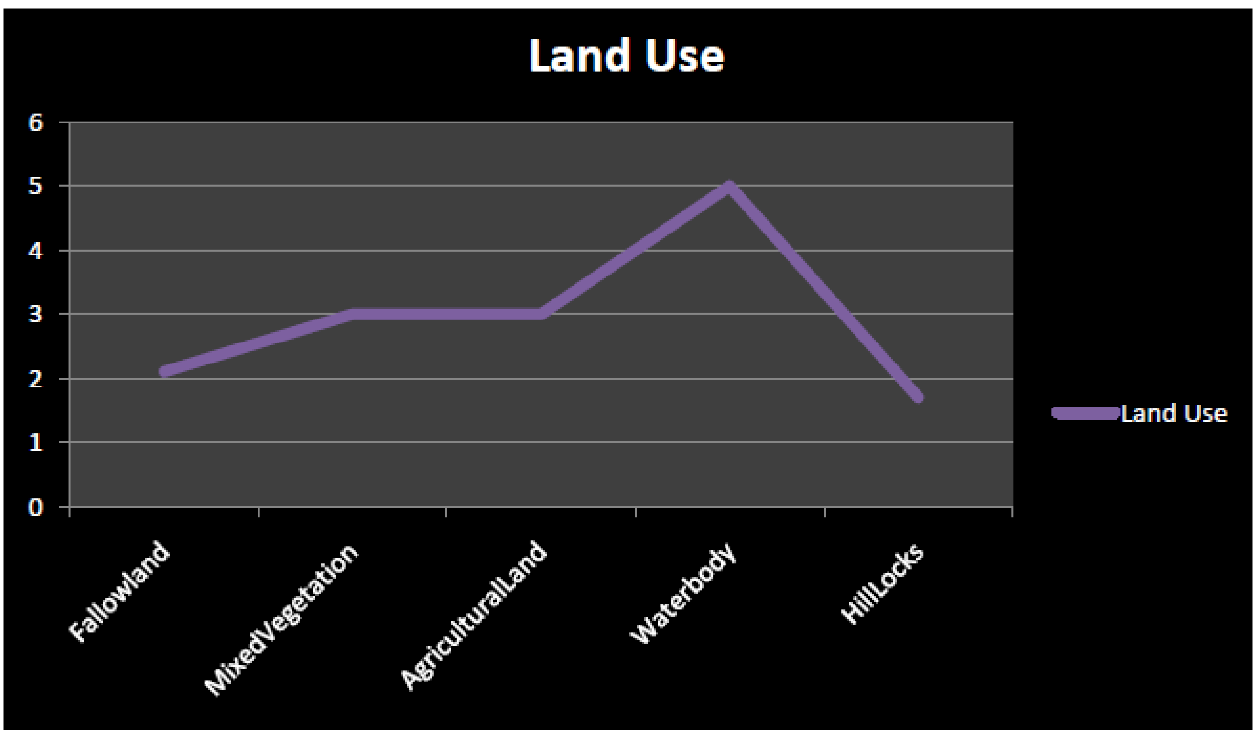

Fig 15. Land use attribute values.

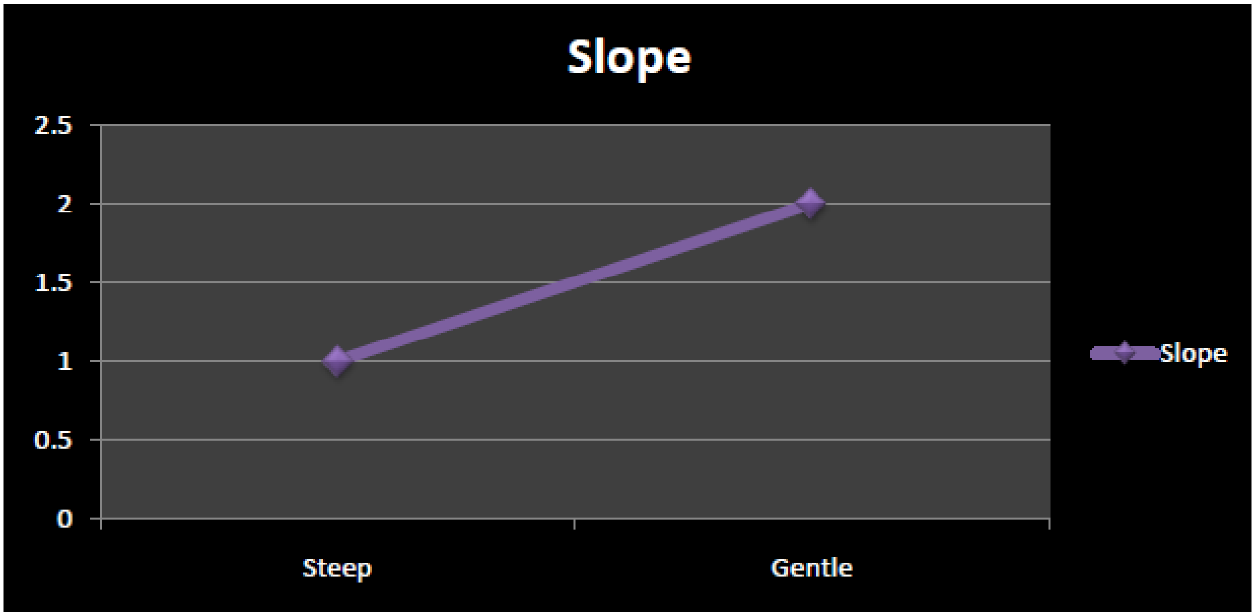

Fig 16. Slope attribute values.

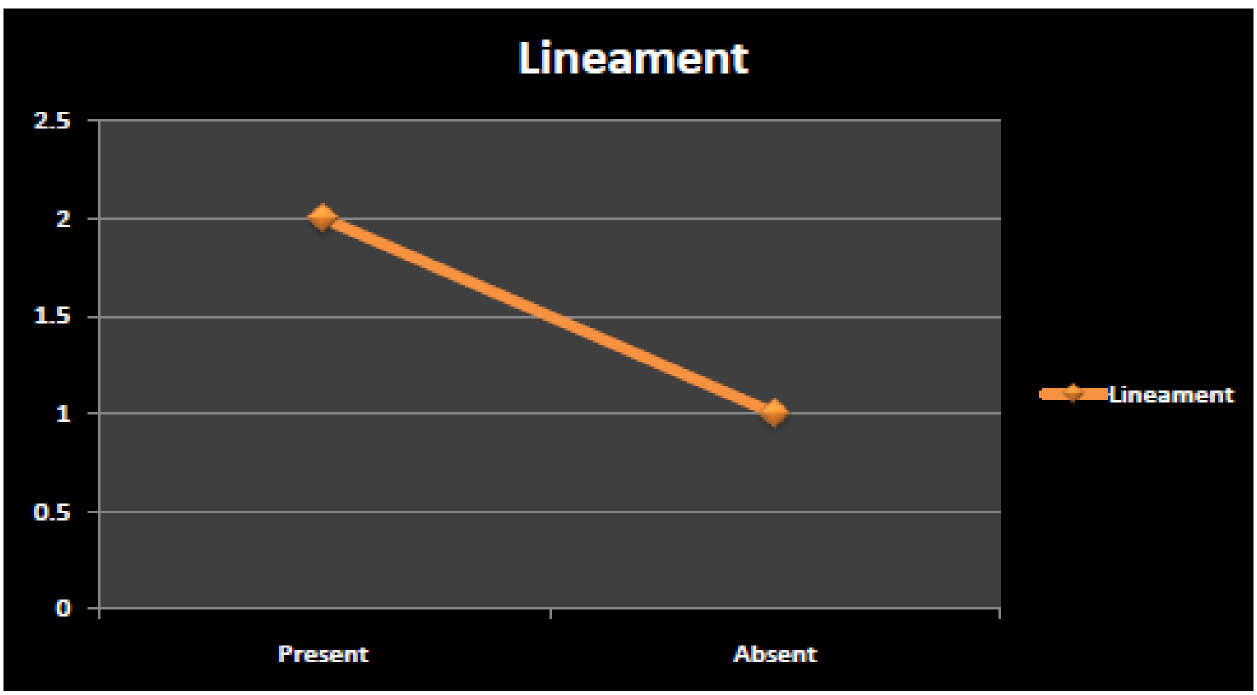

Fig 17. Lineament attribute values.

Figure 16 shows the Slope. It can be steep or gentle.

Figure 17 shows the Lineament.

- Each case is identified by their Similarity index value that is stored in the temp case base. Then increment the count case base by 1.
- Store the cases in the final case base from the temp case base. From this content, we are required to find out the best possible solution from the database. Hence we use Grey wolf Optimizer.
- Initialize the grey wolf population i.e. final case base X_i where i= 1,2,….m.
- Initialize the values of a, A and C i.e. random parameters that assists the ranking procedure. A and C are coefficient vectors that depends upon the value of a ϵ [2,0] , r_1 & r_2 ϵ [0,1].

$$A = 2a.r_1 - a \quad \text{and} \quad C = 2r_2$$

- Evaluate the fitness value of each case base.

  $D = | C.X_p(t) - X_i(t)$ and $X_i(t+1) = X_p(t) - A.D$ where t is current iteration, X_p is problem case and X_i is the final case base.

  After calculating the fitness values, we need to assign X_α for the best solution from the case base, X_β for the second best solution, X_δ for the third best solution according to the maximum value of fitness function value.
- While loop is used in which we check for each case base and then update the current case base. Then update the values for each iteration in order to ignore the unwanted cases by using this equation i.e. $X(t+1) = (X_1 + X_2 + X_3)/3$

  where $X_1 = X_\alpha - A_1. (D_\alpha)$,

  $X_2 = X_\beta - A_2. (D_\beta)$,

  $X_3 = X_\delta - A_3. (D_\delta)$

  A_1, A_2, A_3 are the coefficient vectors of alpha, beta and delta respectively.

- Therefore, we are required to rank the best solution from the case base according to the value of fitness of each case base.
- From the above calculated fitness value, Grey wolf optimizer recognizes the best solution.
- End of program.

Chapter 6: RESULTS AND DISCUSSION

We will discuss the results of underground water exploration in this chapter. Our algorithm is able to provide competitive result. We have also compared it with various traditional techniques like Particle Swarm Optimization (PSO), Biogeography Based Optimization (BBO) etc. It is applied to various regions and in almost all regions it has generated correct output. The algorithm proposed in the previous chapter has been implemented in Matlab 8.0.

6.1. Comparison of Grey wolf Optimizer with other algorithms

In order to obtain the comparison, some benchmark functions have been used to test the performance and efficiency of optimization algorithm. Regardless of the straightforwardness, we have picked these test functions to have the capacity to contrast our outcomes with those of the current metaheuristic techniques. These benchmark functions are the moved, pivoted, extended, and consolidated variations of the established functions which offer the best multifaceted nature among the present benchmark functions. The algorithm is then compared with Biogeography Based Optimization (BBO), Particle Swarm Optimization (PSO) and Cuckoo Search.

There are various benchmark functions such as Sphere function, Griewank's function, Ackley's function, Rastrigin's function and Weierstras's function as described in [3]:

- **Sphere Function**

$$\mathbf{F(x)} = \sum_{i=1}^{d} \mathbf{x_i^2} \qquad x_i \; [-5,5]$$

Where f_{min} = 0 and dimension of function is 10.

- **Griewank's Function**

$$F(x) = \frac{1}{4000} \sum_{i=1}^{d} x_i^2 - \prod_{i=1}^{d} \cos(x / \sqrt{i}) + 1, \qquad x_i [-5,5]$$

Where f_{min} = 0 and dimension of function is 10.

- **Ackley's Function**

$$F(x) = -a \exp\left(-b \sqrt{\frac{1}{d} \sum_{i=1}^{d} x_i^2}\right) - \exp\left(\frac{1}{d} \sum_{i=1}^{d} \cos(c x_i)\right) + a + \exp(1)$$

Where f_{min} = 0 and dimension of function is 10 and x_i [-5,5]

- **Rastrigin's Function**

$$F(x) = 10d + \sum_{i=1}^{d} [x_i^2 - 10 \cos(2\prod x_i) \qquad x_i [-5,5]$$

Where f_{min} = 0 and dimension of function is 10.

- **Weierstras's Function**

$$F(x) = \sum_{n=0}^{\inf} a^n \cos(b^n \prod x) \qquad x_i [-5,5]$$

Where f_{min} = 0 and dimension of function is 10.

Table 4. Comparison of GWO with PSO, BBO and Cuckoo Search

FUNCTIONS	PSO	BBO	Cuckoo Search	GWO
Sphere	100 ± 81.65	100.2 ± 188.57	32.756 ± 53.25	43.83 ± 69.86
Griewank's	155.91 ± 13.176	160.89 ± 151	109.12 ± 40.50	91.8 ± 95.55
Ackley's	172.03 ± 32.769	214.1 ± 74.17	173.72 ± 28.25	61.43 ± 68.688
Rastrigin's	314.3 ± 20.066	615.5 ± 670.8	103.54 ± 37.55	123.12 ± 163.99
Weierstras's	83.45 ± 101.11	358.1 ± 168.3	59.24 ± 19.36	102.14 ± 81.25

In table 4, the numbers are in the format: average number of evaluations, so 43.83 ± 69.86 means that mean of function evaluation is ~44 with a standard deviation of ~70.

According to the above results, we can see that Grey wolf Optimizer is more efficient than other meta heuristic algorithms. These results show the brilliant performance of grey wolf optimizer in terms of exploiting, exploration the optimum and local minima avoidance simultaneously.

6.2. Grey Wolf Optimizer : Dataset

The dataset has been taken from the expert in DRDO. We proposed an algorithm gives the output as possibility of ground water in a particular area or region. In our dataset, many regions are considered by our expert. As user input its requirement our detector detects whether such region will have {high, moderate, low} groundwater. Thus software is a total solution for groundwater detection possibility provided all cases are put away in the case base. The case base provided by our expert we have separated 115 cases for our case base and some for validation purpose as shown in table 5. Grey wolf Optimizer is a recent technique and it has not shown its footprints in detector mechanism. The main advantage of this algorithm is search space is extensive in nature. Thus our proposed algorithm combines all the strength of this metaheuristic algorithm. Its searching is more intensified than other swarm intelligence thus it shows more accurate results than other techniques. The validation dataset has same configuration as the case base.

GEOLOGY	LANDFORM	SOIL	LANDUSE	SLOPE	LINEAMENT	STATUS
2.3	0.1	5	4	1	2	Moderate
2.3	1	5	4	1	2	Moderate
3	2.2	2.1	2.4	1	1	High
3	2.2	2.1	2.4	2	1	High
3	2.2	2.1	3	2	1	High
3	2.2	2.1	5	2	1	High
2.4	1	5	2.2	1	1	Low

Table 5. Validation dataset of groundwater

6.3. Implementation

The user interface is developed in Matlab 8.0 where user can enter the attribute values and find out the percentage of possibility of ground water.

Upon execution of our system the first window which appears is shown in figure 18.

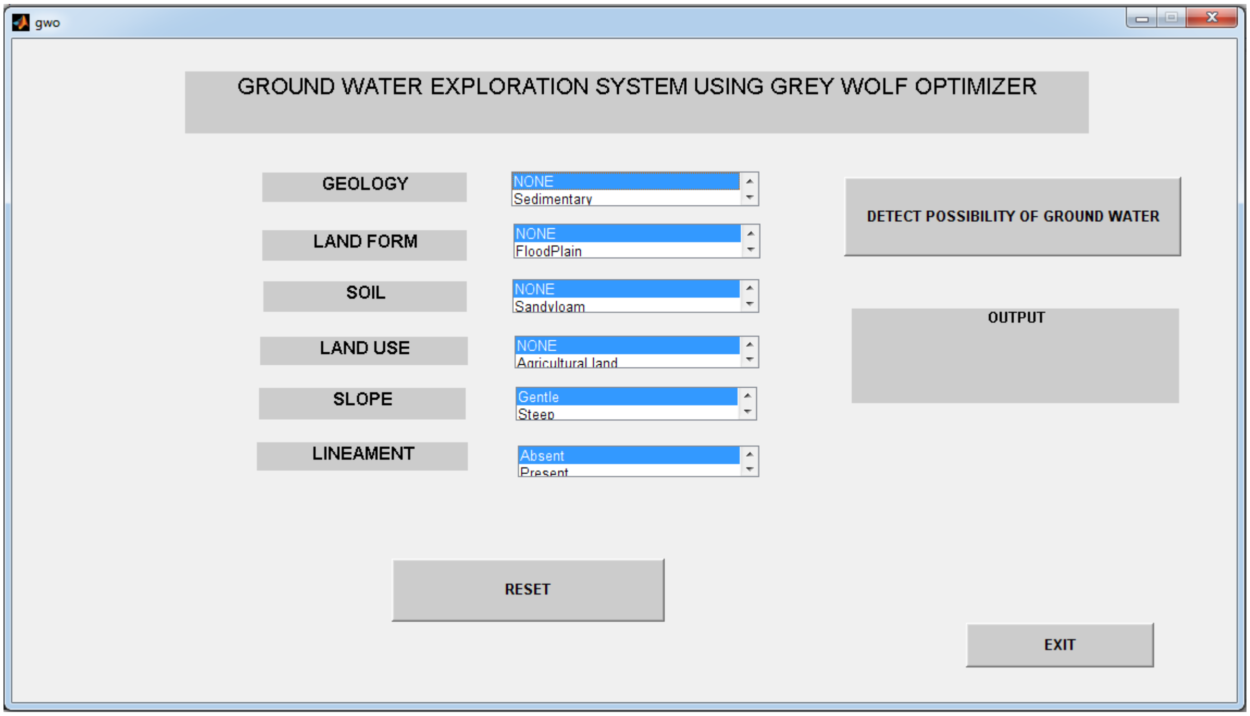

Fig 18. First window appear at the time of execution of software

This represents all the geographical attributes such as geology, land form, soil type, land use, slope, lineament which plays very crucial role in the detection of groundwater.

When the "Detect possibility of Ground water" is pressed then in the text bar "Output" value is displayed whether the possibility is high, low or moderate.

High Possibility: 85% and above

Moderate Possibility: 60 to 84 %

Low Possibility: less than 60 %

In this graphical user interface, RESET button is used to reset all the attribute values to their initial condition and OUTPUT text box will remain empty as no query is entered by the user.

The purpose of RESET button is shown in figure 19.

When exit button is clicked then the GUI is closed and it goes back to matlab window.

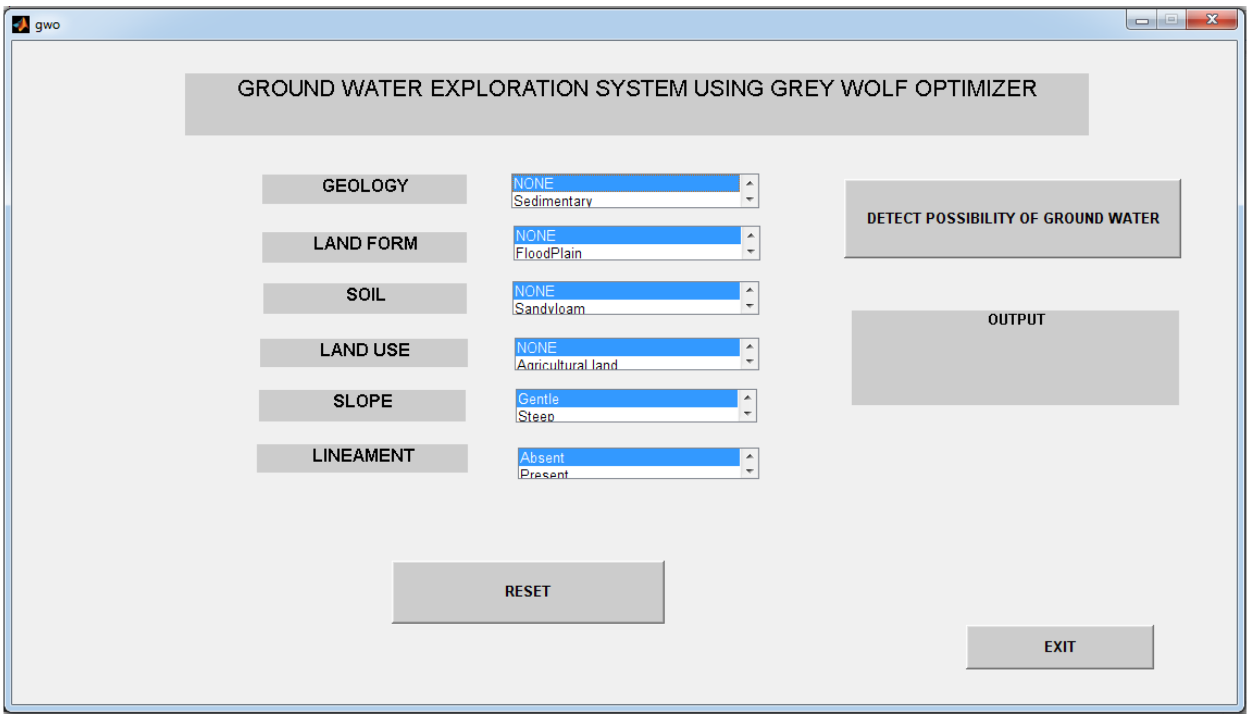

Fig 19. Purpose of RESET button.

6.4. Results

We took a query that is entered by the user. All the query attributes are mentioned in the table 6.

GEOLOGY	**Younger Alluvium**
LAND FORM	**Intermontervalley**
SOIL TYPE	**Coarse Sand**
LAND USE	**Cultivated Land**
LINEAMENT	**Present**
SLOPE	**Gentle**

Table 6. User Query

The result of above query shows in the figure 20.

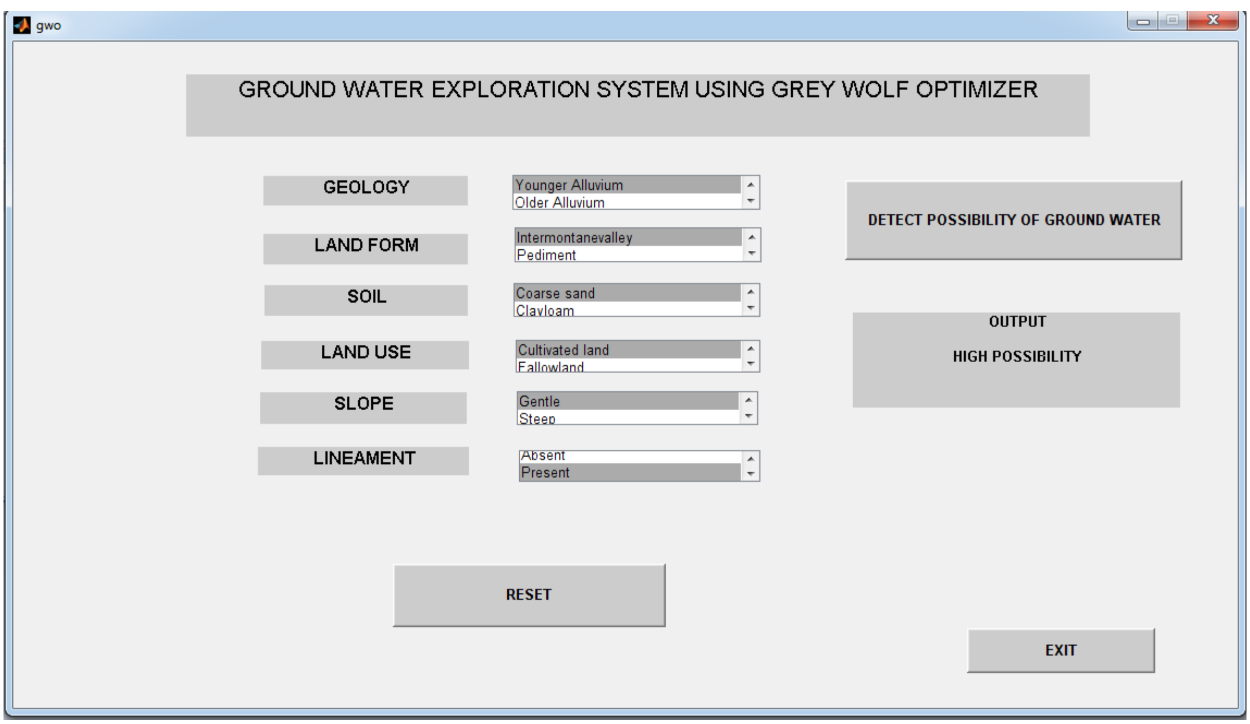

Fig 20. Result of user's query

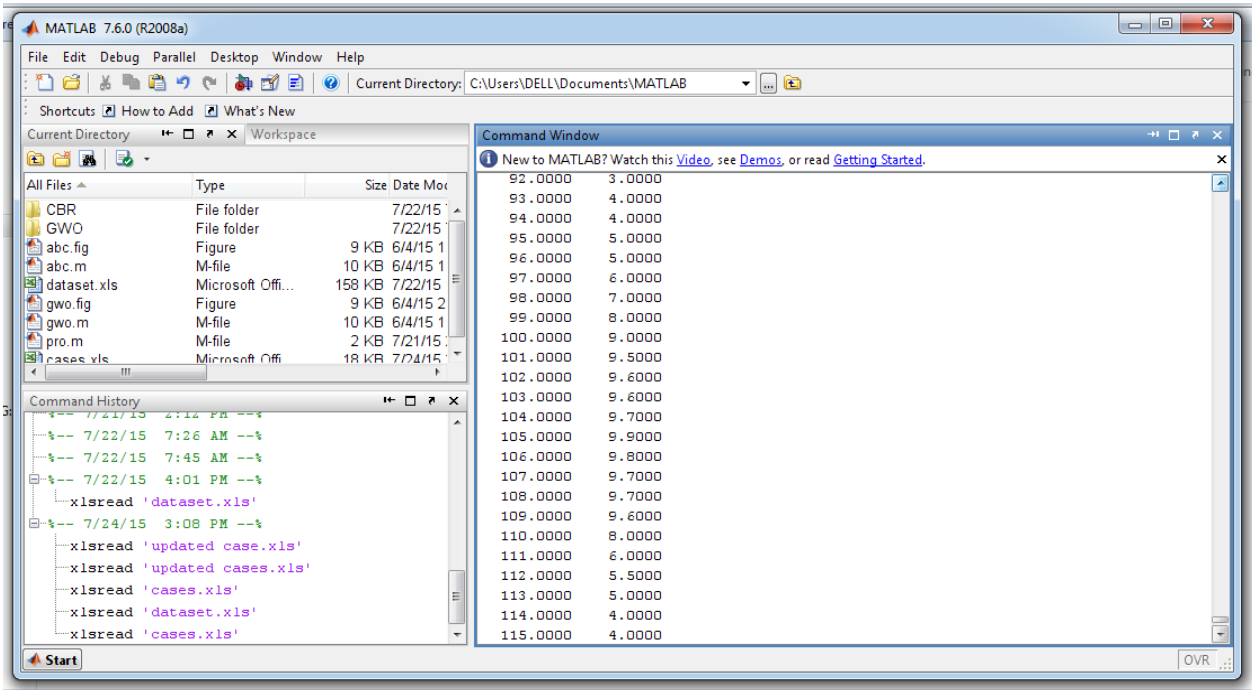

Fig 21. Cases stored in final case base

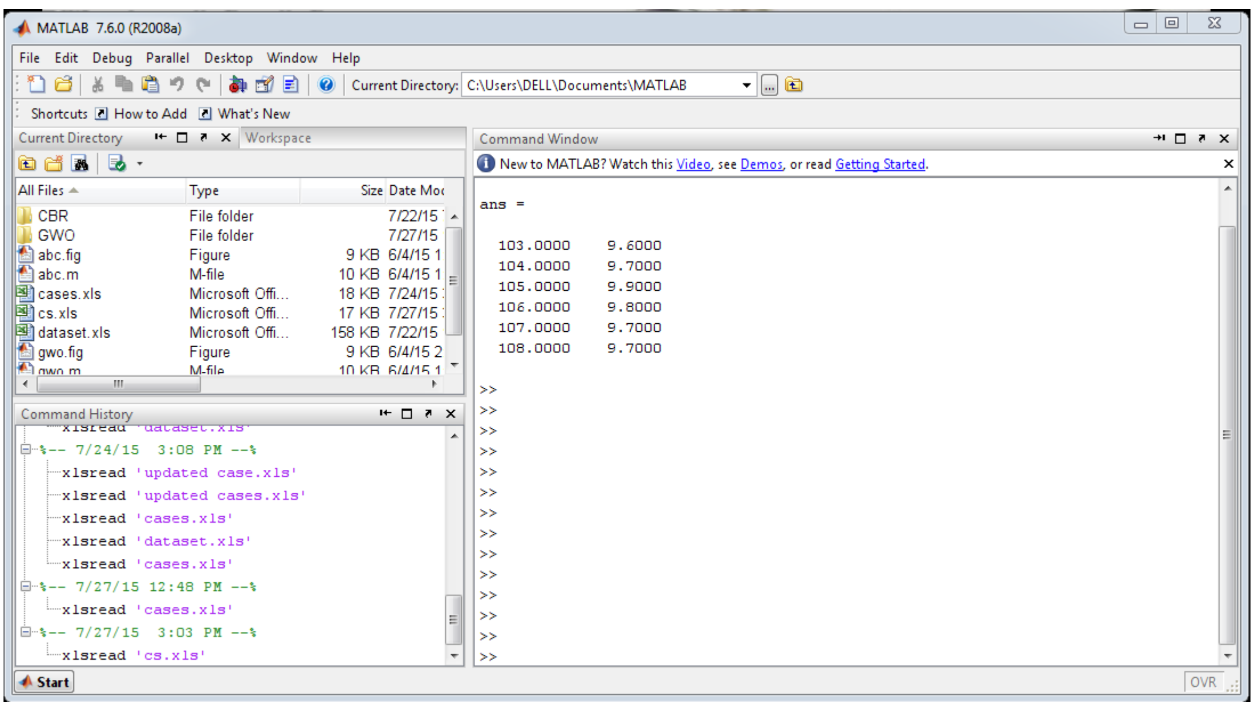

Fig. 22. Cases after ignoring irrelevant cases.

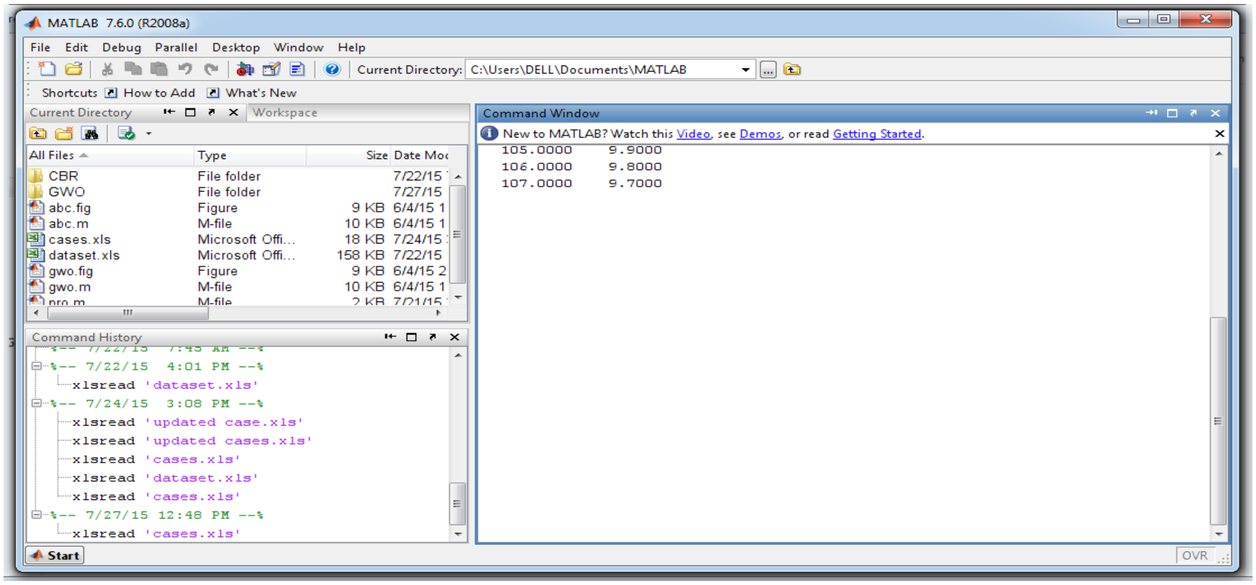

Fig. 23. Final three cases.

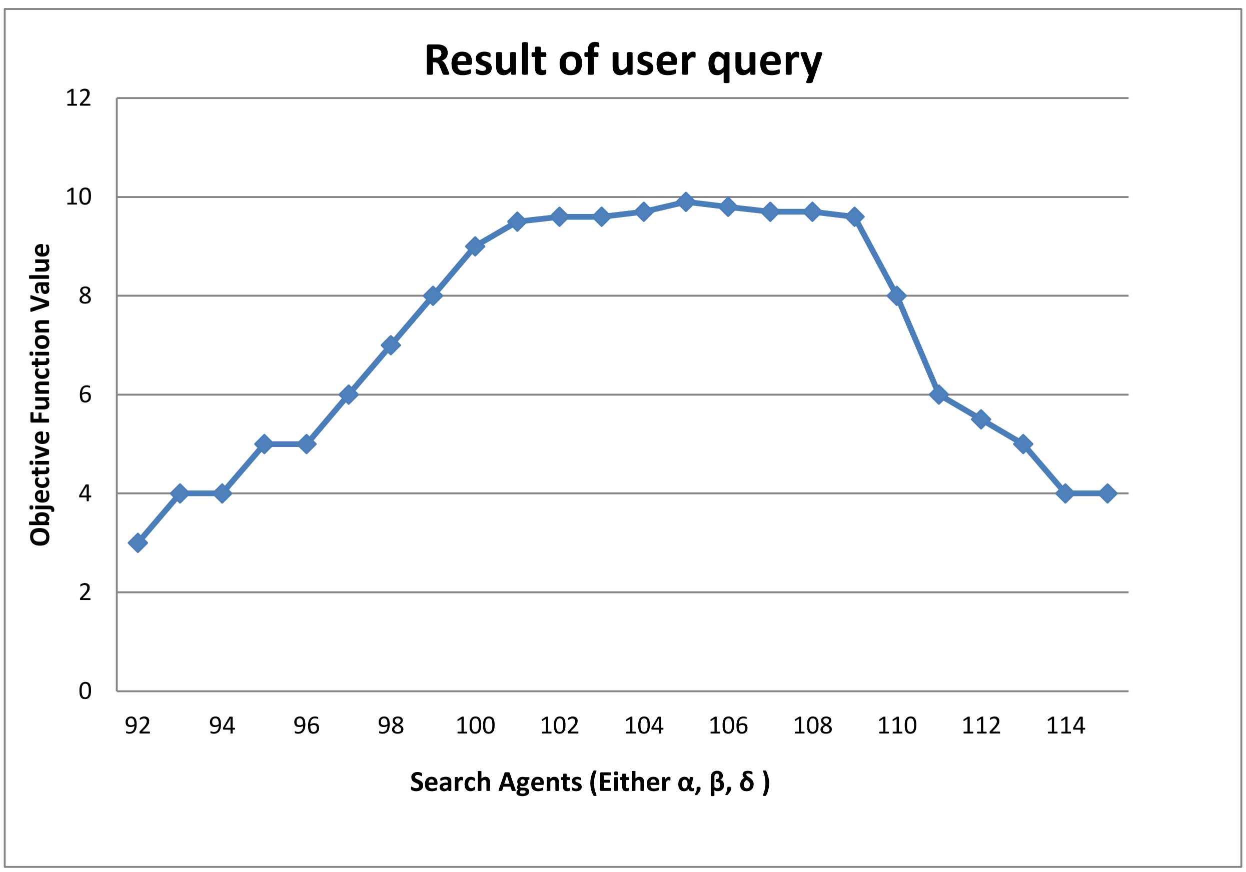

Fig 24. Objective Function vs Search Agent(GWO)

According to the query entered by the user, we first obtain the objective function value for each search agent. In figure 21, cases stored in the final case base is shown. After ignoring irrelevant cases, remaining cases are shown in figure 22. Final cases which are having high fitness value are shown in figure 23.

In Figure 24, X axis represents all the search agents (which are looking for prey) and Y axis represents the objective function value (fitness value) for each search agent.

According to the plotted graph, we have observed that highest objective function value(fitness value) is 9.9~10. And around 6 search agents have highest objective function value.

Among the six search agents which are having highest fitness value, best solution is obtained by the ranking of these search agents. Ranking is done by Grey wolf optimizer according to fitness value. Hence, search agent 105 seems to be alpha(α) i.e. best solution obtained. The result of the user query is high.

$$\text{Total case taken} = 115$$

$$\text{Ignored cases} = 115 - 6 = 109$$

$$\text{Ignored case \%} = (109 \times 100) \div 115$$

$$= 94.78\ \%$$

Hence, around 94.78 % are the ignored case that means our algorithm neglected these search agents which can’t give optimum result to the user query.

6.5. Comparison of GWO with Cuckoo Search

Comparison is done in terms of % of ignored case for the same query as entered by the user. In case of Cuckoo Search, we considered 10 cases which are having highest objective function among 115 cases to find the optimum result. Thus from the current query, ignore cases are 105 (115-10) and total case are 115.

Ignored case (CS) % = (105×100) ÷ 115 = 91.3 %

But as in case of Grey wolf Optimizer (GWO), the ignored case % is 94.78% (as calculated above).

Cuckoo search (CS) has to find the best possible case from the rest 8.7 % cases while Grey wolf Optimizer (GWO) has to find the best possible case from the remaining 5.22 % cases. This shows that Grey wolf optimizer is much efficient than Cuckoo search.

Chapter 7: CONCLUSION AND FUTURE SCOPE

7.1. Conclusion

The proposed algorithm of groundwater detection is another area of natural computation that we have implemented. We have implemented an algorithm for groundwater detection that is quiet efficient. This metaheuristic algorithm had been proved to be better searching algorithm than other traditional technique. Thus our proposed framework is able to detect the presence of groundwater at inaccessible areas through remote sensing. Thus traditional methods difficulties and disadvantages are overcome. As in grey wolf optimizer, search space is extensive in nature and it works according to the behavior of grey wolves. This GWO algorithm shows the hunting behavior of grey wolves. In our algorithm, we consider the alpha, beta, and delta wolf as three different positions in the single case base. We got case base from experts. Our system predicted the presence of groundwater in a region as provided by user. Our system aimed at answering whether the region is having low, high or moderate groundwater possibility. We have developed a framework which is equipped for deciding groundwater content at different depth without doing any physical work like drilling wells etc. Utilizing the framework, it is as of now possible to endeavor a local hydro land outline to direct the conveyance of groundwater stores both in plain and inside and out with high viability, as needs be guaranteeing a strong choice of the most valuable areas for water supply exhausts. We have used case based reasoning along with grey wolf optimizer to detect the ground water possibility. Till date, There are so many techniques that have been applied to remote sensing satellite image that have efficiently classified the geospatial features of the terrain despite its discrepant uncertainties.

This framework has critical influence in military applications amid the season of fights to find the groundwater plausibility in closed off domains like regions over the fringe where improvement of troops majorly depends on such information as water is a central need of survival. The result shows that GWO was able to provide highly competitive results compared to various traditional techniques like PSO, BBO etc.

7.2. Future Scope

As our system is giving optimum results, we can improve its detecting mechanism efficiency if huge amount of case base is provided by experts. One of the difficult parts is to find the sufficient amount of cases. From these cases, fewer ones will give poor result with less accuracy. If this framework can be connected practically to manage thousands cases, the proficiency will most likely results great. Future work should be possible by trying different things with other categorical datasets of distinctive locales to investigate the productive results that can be appropriate in different fields. We can also use rule induction algorithm of machine learning which quiet similar with CBR along with case base reasoning for case base retrieval. Also we can develop the binary and multi objective versions of Grey wolf optimizer.

REFERENCES

[1] Daya Gupta, Bidisha Das, V. K. Panchal, "Applying Case Based Reasoning in Cuckoo Search for the Expedition of Groundwater Exploration", Springer, December 2012.

[2] Dan Simon, "Biogeography-based optimization," IEEE Transactions on Evolutionary Computation, vol. 12, December 2008.

[3] Seyedali Mirjalili, Seyed Mohammad Mirjalili, Andrew Lewis, "Grey Wolf Optimizer", Springer, January 2014.

[4] V.K.Panchal, Harish Kundra, Amanpreet Kaur, "Biogeography based groundwater exploration", 2010 International Journal of Computer Application, IJCA, 0975-8887, vol. 1, no. 8, 2010.

[5] V.K.Panchal, Harish Kundra, Navpreet Kaur, "A novel Approach to Integration of waves of swarms with Case Based Reasoning to detect Groundwater potential". 8th Annual Asian Conference & Exhibition of geospatial Information Technology & Application, Map Asia, Singapore, 2009.

[6] Daya Gupta, Lavika Goel, V.K.Panchal, "Embedding Expert Knowledge to Hybrid Bio-Inspired techniques- An Adaptive Strategy towards Focussed Land Cover feature Extraction", International Journal of Computer Science & Information Security, (IJCSI), vol.8, no.2, 2010.

[7] C.K. Riesbeck, R. C. Schank, "Inside case based reasoning", L. Erlbaum Associates Inc. Hillsdale, 1989, New Jersey, ISBN: 0898597676.

[8] I. Watson, "Applying case based reasoning: Techniques for enterprise system", The Morgan Kaufmann Series in Artificial Intelligence, 1997, California, USA.

[9] C.J.Price, I. Pegler, “Deciding parameters values with case based reasoning”, In I. Watson (eds), Progress in Case Based Reasoning, Lecture Notes in Artificial Intelligence,1020 Springer, 1995, Berlin.

[10] Sonal Kumar, Daya Gupta, V.K.Panchal, Shashi Kumar, “Enabling web services for classification of satellite images”, In International Conference on Semantic Web & Web Services, SWWS, July 13-16, 2009, Orlando.

[11] An Introduction to Remote Sensing & Geographic Information System, Yale University, Genocide Studies Project, Remote Sensing & Geographic Information System Research.

[12] Farid Melgani and Lorenzo Bruzzone, “Support Vector Machines for Classification of Hyperspectral Remote-Sensing Images”, IEEE transactions on Geoscience and Remote Sensing, Vol.42.No.8, Aug 2004.

[13] Gregoire Mercier and Marc Lennon, “Support Vector Machines for Hyperspectral Image Classification with Spectral-based kernel”, IEEE 2003.

[14] Peijun Du, Junshi Xia, Jocelyn Chanussot , Xiyan He, “Hyperspectral Remote Sensing image Classification based on the Integration of Support Vector Machine and Random Forest”, IEEE 2012

[15] Grey Wolf optimizer Toolbox, http://www.mathworks.com/matlabcentral/fileexchange/47258-grey-wolf-optimizer-toolbox

[16] MATLAB, http://www.mathworks.com/.