

Design and Development of Grey Wolf Metaheuristic Technique with Application to Combinatorial Optimization Problem

Thesis submitted in partial fulfillment of the requirements for the award of degree of

Master of Technology

in

Computer Science and Applications

Submitted By

Seema

(Roll No. 601403025)

Under the supervision of:

Dr. Vijay Kumar

Assistant Professor

Thapar University, Patiala



COMPUTER SCIENCE AND ENGINEERING DEPARTMENT

THAPAR UNIVERSITY

PATIALA – 147004

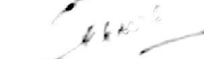
June 2016

Certificate

I hereby certify that the work which is being presented in the thesis entitled, "*Design and Development of Grey Wolf Metaheuristic Technique with Application to Combinatorial Optimization Problem*", in partial fulfillment of the requirements for the award of degree of Master of Engineering in *Computer Science and Application* submitted in Computer Science and Engineering Department of Thapar University, Patiala, is an authentic record of my own work carried out under the supervision of *Dr. Vijay Kumar* and refers other researcher's work which are duly listed in the reference section.

The matter presented in the thesis has not been submitted for award of any other degree of this or any other University.

Signature:



(Seema)

This is to certify that the above statement made by the candidate is correct and true to the best of my knowledge.



(Dr. Vijay Kumar)
Assistant Professor,
CSED

Countersigned by



(Dr. Maninder Singh)

Head
Computer Science and Engineering Department
Thapar University
Patiala



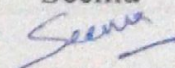
(Dr. S. S. Bhatia)
Dean (Academic Affairs)
Thapar University
Patiala

Acknowledgement

First of all I would like to thank the Almighty, who has always guided me to work on the right path of the life. It is a great privilege to express my gratitude and admiration towards my respected supervisor **Dr. Vijay Kumar**, Assistant Professor, Computer Science & Engineering Department, Thapar University, Patiala. He has been an esteemed guide and great support behind achieving this task. This work would not have been possible without the encouragement and able guidance of him. I also thank my supervisor for his time, patience, discussions and valuable comments. His enthusiasm and optimism made this experience both rewarding and enjoyable. I am truly grateful to him for extending his total co-operation and understanding whenever I needed help and guidance from him. I am also heartily thankful to **Dr. Maninder Singh**, Associate Professor and Head, Computer Science & Engineering Department and **Dr. Sanmeet Kaur**, PG coordinator, for motivation and providing uncanny guidance and support throughout the preparation of the thesis report.

I will be failing in my duty if I do not express my gratitude to **Dr. S. S. Bhatia**, Senior Professor and Dean of Academic Affairs, for making provisions of infrastructure such as library facilities, computer labs equipped with net facilities, immensely useful for the learners to equip themselves with the latest in the field.

I am also thankful to the entire faculty and staff members of Computer Science and Engineering Department for their direct-indirect help, cooperation, love and affection, which made my stay at Thapar University memorable. Last but not least, I would like to thank my family for their wonderful love and encouragement, without their blessings none of this would have been possible.

Seema

(601403025)

Abstract

Classical algorithms are unable to find the optimal solution for many real life problems. Metaheuristic algorithms were developed to solve these problems. These are popular due to their easy implementation and flexible nature. These algorithms are inspired from behaviour of bird/animals, collective intelligence of swarms, logical behaviour of physical processes occurred in the nature. Grey Wolf Algorithm (GWA) is one of the recent developed meta-heuristic techniques and has been used for solving many optimization problems.

The main aim of this dissertation is to develop an improved version of classical Grey Wolf Algorithm (GWA). GWA is inspired from social leadership and hunting mechanism of grey wolves. The two control parameters namely A and C are greatly affect the performance of GWA. The improvement has been done in GWA for further exploration and exploitation. The proposed algorithm is tested on thirteen benchmark functions and a comparative analysis is made with other well-known metaheuristic algorithms.

List of Figures

Figure No	Figure Descriptions	Page No
Figure 1.	Classification of Metaheuristics.....	3
Figure 2.	Flowchart of Grey Wolf Algorithm (GWA).....	10
Figure 3.	Movement of Grey Wolves in Oval Path.....	30
Figure 4.	Flowchart of Modified Grey Wolf Algorithm(MGWA).....	31
Figure 5.	Pseudo code of the Modified Grey Wolf Algorithm(MGWA).....	32

Chapter 1

Introduction

In optimization problems, there is a need to find out the best values of the variables or good configuration of the parameters in order to find optimal solutions. In some of such problems the solutions are encoded in real valued variables while in others discrete variables are used to encode the solutions. Combinatorial Optimization problems (CO) belong to this category where solutions are encoded with discrete variables. To solve CO problems, complete or approximate methods can be used. In complete methods, there is guarantee of finding optimal solution in some bounded time but these methods may take exponential time for NP-Hard problems. So, approximate methods were used to solve these problems. Here, there is no guarantee to find the optimal solution but approximate methods tend to find solutions in reduced time as compare to complete methods. Approximate methods can be further classified into constructive and local search methods. Constructive methods are generally fast than the local search. However, they produce inferior quality solutions and generate solutions from the scratch. In local search methods start with some initial population and improve this population. The metaheuristic methods belong to this category of approximate methods.

The metaheuristic concept was introduced by Glover in 1986. Metaheuristics refer to those heuristic techniques that can be applied to wide range of problems. These are general techniques which can be applied to any problem as these are not problem dependent. These are techniques for better exploration of the search space. There is need for suitable balance between intensification and diversification in these techniques. The name of the few metaheuristics techniques are Ant colony optimization (ACO)[1], particle swarm optimization (PSO)[2], Tabu Search[3], Simulated Annealing[4], Genetic Algorithms[5] etc. Metaheuristic techniques are successfully used in many areas like engineering, finance and product management etc. to solve complex problems.

1.1 Definition

Metaheuristic can be defined as “the framework which is problem independent and that provides solution to optimization problems”. In metaheuristic the subordinate heuristic is guided by different strategies and for better exploration and exploitation of the search space. These are best applied in situation when there is insufficient or incomplete information about the problem and computation capability is limited [6].

“A metaheuristic is formally defined as an iterative generation process which guides a subordinate heuristic by combining intelligently different concepts for exploring and exploiting the search space, learning strategies are used to structure the information in order to find efficiently near-optimal solutions”[7].

“A metaheuristic is a set of concepts that can be used to define heuristic methods that can be applied to wide set of different problems. In other words metaheuristic can be seen as a general algorithmic framework which can be applied to different optimization problems with relatively few modifications to make them adapted to a specific problem”[8].

1.2 Characteristics

The main characteristics of the metaheuristics are as follows [6]:-

- These are not problem specific.
- Their nature is non-deterministic type.
- These are approximate methods.
- Temporary deterioration of solution may be accepted.

1.3 Classification

At present many metaheuristics techniques are available for solving optimization problems. These can be classified into various categories on the basis of different parameters or choices. These metaheuristic techniques have been classified into four different classes according to different dimensions given as under [6, 9]:-

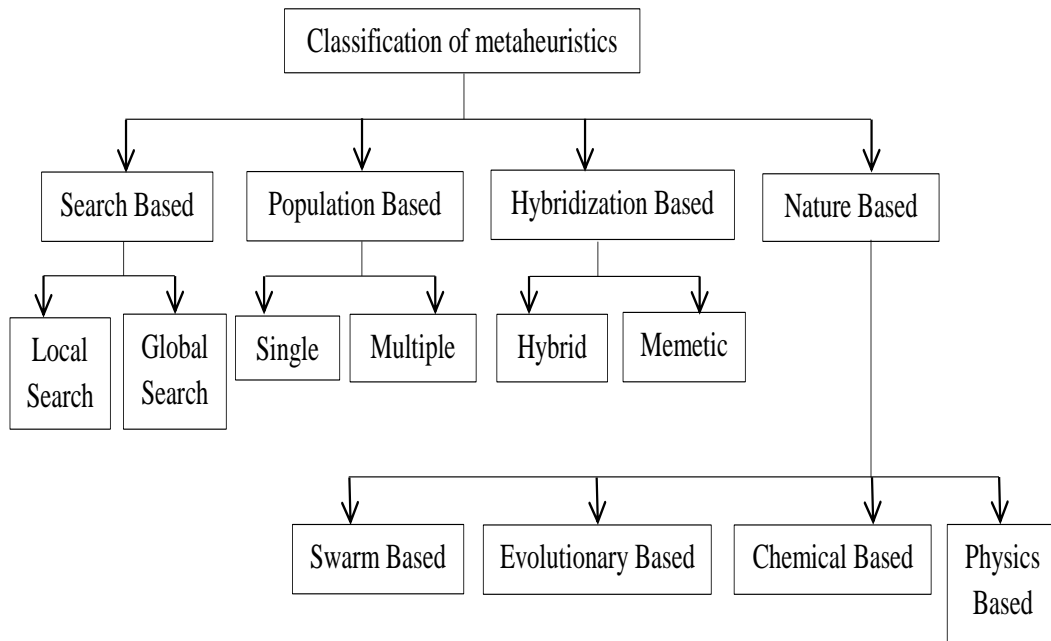


Figure 1. Classification of Metaheuristics

Based on the above classification these metaheuristic techniques are described as follows:-

1.3.1 Local Search vs. Global Search

Depending on the search strategy we can classify our metaheuristics as the one that belongs to local search algorithms and the other belonging to global search algorithms. In local search metaheuristics algorithms the starting point is from a candidate solution and then slowly with each passing iteration movement towards neighborhood solution is made. The neighborhood relation should be defined in the search space in these algorithms. One of the popular local search algorithms in optimization is Hill Climbing. In this algorithm we start with any random solution to a problem. Now we change one element of the solution and check if it is better than the previous solution or not. If it is better than the previous solution then the current state is updated and this process goes on until we found the goal state. Some of the examples of local search metaheuristics algorithms are Iterated Local Search [10], Variable

Neighborhood Search [11] etc. The advantage of these algorithms is that they use little memory space and they are able to find out reasonable solutions in large state spaces too. However the disadvantage of these algorithms is getting stuck in local optima. This was the main reason for moving towards global search algorithms.

Global search metaheuristics are generally population based metaheuristics techniques. In global search algorithms, best solutions are found out in global search space. There can be multiple local optima present in search space and among them the best one will be chosen out by the algorithm. These algorithms are more effective than the local search algorithms because of their capability of proper balance between exploration and exploitation. The termination of these algorithms is the stopping criterion defined by the user. These algorithms are robust in nature and give better results than local search algorithm. However, these algorithms may get suffer from premature convergence. Genetic Algorithms (GAs) belongs to the category of global search metaheuristics algorithms. Other examples of global search metaheuristics are Ant Colony Optimization, Particle Swarm Optimization and Evolutionary Computation etc.

1.3.2 Single Solution vs. Population based

Another way of classifying metaheuristic techniques is based on candidate solution. If we are improving and finding one candidate solution only then it is single solution based metaheuristic technique but in population based metaheuristics we have set of candidate solutions. Variable Neighborhood Search (VNS) is one such single solution based metaheuristic technique. The neighborhood solutions are found by this algorithm and a move in that direction is made if they are better. Whether we have continuous or discrete parameters in our problem, this algorithm can be applied irrespective of it. Any type of problem can be solved by this algorithm like linear, non-linear, integer, mixed integer etc. Apart from VNS we have some other single candidate solution algorithms such as Simulated Annealing, Guided Local Search and Iterated Local Search etc.

In population based metaheuristic, there are a set of candidate solutions. The advantage of population based metaheuristic techniques is that we have greater

exploration. The multiple candidate solutions help each other and share information with each other in order to move towards more promising areas. Hence there are less chances of getting stuck in local optima as compare to single solution based metaheuristics. Particle Swarm Optimization (PSO) is one such algorithm belonging to population based metaheuristic techniques category. The algorithm mimics the intelligent behavior of the swarms in group belonging to biological population. Other population based metaheuristic techniques apart from PSO are Ant Colony Optimization, Social Cognitive Optimization, Artificial Bee Colony and Penguin Search Optimization Algorithm etc. These population based metaheuristics techniques are gaining ground than the single solution based metaheuristics because of their better performance and ability to find global optima efficiently along with good convergence capability.

1.3.3 Hybrid and Memetic Algorithm

Another classification of the metaheuristic algorithms is into hybrid and memetic algorithms. After the pure metaheuristic techniques researchers started following many new approaches of combining different algorithms belonging to metaheuristics. The aim was to get improved exploration and exploitation capability of the algorithms. The combination of the algorithms is powerful tool for solving large and complex problems. Such approaches were termed as hybrid approaches. In order to solve any hard optimization problem the choice of complementary algorithm greatly affects the performance in good manner. The complimentary algorithm may belong to already existing metaheuristic techniques or it may belong to different area too. So there are many different parameters of hybridizing the algorithms. For example metaheuristics can be hybridized with other metaheuristics or hybridize metaheuristic with some algorithm specific to given problem or hybridize metaheuristic with technique of other areas like artificial intelligence, operational research etc. Another way of classifying them can be based on the level of the hybridization between the algorithms. Based on the strength of their combination these are divided into two classes. In one class, high level weak coupling algorithms are present where individual identity of the algorithm is maintained. In second class low level strong coupling algorithms are present. These algorithms depend on each other and share individual components. There can be manifold

possibilities of hybridizing these algorithms. Hybridizing algorithms open gates of further research and more chances to improve the algorithm significantly.

Memetic algorithms refer to those algorithms whose structure is defined by the evolutionary framework along with some local components. These are result of the Darwinians and Dawkins inspired concepts and philosophy. Local search methods can be used in these algorithms to avoid the problem of premature convergence. A meme is that unit which can be transferred from one mind to another. The mode of the transfer is via imitation. The main idea behind their development is that in some cases genetic algorithms do not perform better. So, on those particular problem areas memetic algorithms are used. The development of the memetic algorithms is classified into three generations. In first generation of the memetic algorithms population based global search methods were hybridized with cultural evolutionary stage. The second generation memetic algorithms are termed as multi-meme algorithms. In these algorithms it is assumed genotype is encoded in the form of memetic material. For local refinement we use decoded meme of each individual and transfer occurs by normal process of inheritance. In third generation of memetic algorithms we are referring to self-generating memetic algorithms where memes to be used are not known in priori.

1.3.4 Nature inspired Metaheuristics

Those metaheuristics which are inspired by the nature are known as nature inspired metaheuristics. Nature is surviving several changes in the environment over a million of years. The adaptation of nature is amazing and can be used as an inspiration for developing efficient and adapting algorithms. According to the Darwin theory of evolution only the fittest species in the changing environment can survive [12]. Hence this pressure of selection helps to improve the system for better by selecting quality species. At each year new algorithms are coming inspired by the nature. Improvements in the existing algorithms are also taking place by employing these nature inspired metaheuristic techniques in them. These nature inspired metaheuristics algorithms are classified into four categories as depicted in Figure 1(mentioned in Section 1.3)

In swarm based metaheuristics the behavior of the swarm is mimicked. The swarm can be group of animals or birds. Example of swarm based metaheuristics includes Cuckoo Search [13], Particle Swarm Optimization (PSO) [2], Grey Wolf Algorithm (GWA) [14], firefly algorithm [15] etc. To understand the working of swarm based metaheuristics consider PSO algorithm. A scenario of birds searching for food is taken. At each iteration the flock of bird is getting closer towards the target i.e. food. What the birds are doing is to follow the best bird position in the direction of food. Now in PSO each bird is taken as single solution and it is termed as a particle. There is group of particles and these particles move around in the search space. Particles share information with each other and move towards more promising areas of the search space. Each particle has its own velocity and fitness value. At each iteration of the algorithm the fitness value of the particle is compared and updated if it is better than the previous value and accordingly particle position is also updated. Two values are maintained Pbest and Gbest. Pbest is the best value obtained by the particle so far and Gbest is the best value obtained in the neighborhood until now. Particle is move around in search space based on these two values. The choice of the parameter greatly affects the performance of the algorithm. The local search ability and global search ability of PSO depends on the value of the inertial weights. It is better to change the value of inertial weight from small to large than keeping it static or fixed because changing the inertial weights from small to large results into better PSO performance. So careful selection of the parameters is needed while using this algorithm for better performance.

In evolutionary based metaheuristics algorithm the principles of evolution are applied. The whole process of evolution is modeled mathematically in these algorithms. Genetic Algorithm (GA) is one of the popular evolutionary based metaheuristic techniques. In this algorithm every optimization parameter is coded in terms of gene. Chromosomes are made up of these genes and the genes correspond to the parameters. Depending on the problem chromosome can be binary string, real numbers or a list of elements in the database. Every chromosome corresponds to a solution and the set of such solutions is termed as population. The fittest ones will be selected for mating from the population. In mating process genes from different parents are used to create new child termed as crossover. Now these newly created children are put into the population

and whole process is repeated again. This will keep going until we reach to convergence or reach maximum number of generations. There are many variations of Genetic Algorithms (GAs) such as simple Genetic Algorithms, steady state Genetic Algorithms, Genetic Algorithms with multiple populations and Genetic Algorithms with crowding and sharing techniques etc. Depending on the type of the problem we apply one of them. Genetic Algorithms (GAs) are capable of handling any type of fitness landscapes, real or discrete parameters and are very robust.

In physics based metaheuristics algorithms, the laws of physics are employed in the algorithm. The movement of the agents is according to the physical rules in the search space. Gravitational Search Algorithm (GSA) is a popular metaheuristic technique belonging to this category. This algorithm mimics the gravitational law given by Newton. Calculation of the fitness value and computation of masses of the agents is performed. After that acceleration is computed by considering all the forces that are taking place on the agent. Once acceleration is found out the velocity is calculated. The position of the agent is computed and algorithm is kept on running until stopping criterion is achieved. This algorithm has been used for solving many optimization problems.

In chemical based metaheuristics algorithms the whole process of chemical reaction is mimicked. When a chemical reaction takes place the molecules react with each other and they come to low energy state in the end. Chemical Reaction Optimization (CRO) is well known algorithm in the category of chemical based metaheuristics algorithms. In this algorithm, potential energy of molecules is used as a measure for objective function value and kinetic energy is used as an estimation of the molecular structure change. The whole concept of energy that it can neither be created nor be destroyed is used in the algorithm. Conversion of kinetic energy to potential energy and vice versa is implemented via different steps. This algorithm implements complete procedure of a chemical reaction and assures convergence to lower states.

Inspired by the swarm based metaheuristics algorithms the base of the proposed approach in this thesis is Grey Wolf Algorithm (GWA). Grey Wolf Algorithm (GWA) is a recently developed population-based metaheuristic technique. It is developed by Mirjalili et al. in 2014. It is inspired from social and hunting behavior of grey wolves.

The grey wolves are living in pack of 5-12 wolves. A social hierarchy is present in the pack of grey wolves. The leaders of the pack are alpha (α) wolves [16]. They are considered as best solutions and they are having best knowledge of the position of the prey. The beta (β) and delta (δ) wolves help the alpha wolves in decision making process. Both of these (β and δ) can be considered as second and third fittest solutions. The remaining grey wolves are treated as omega (ω) wolves. The ω wolves follow commands of other wolves in hierarchy. The omega wolves are considered as remaining candidate solutions. The encircling behavior of grey wolves can be mathematically modeled as follows [14]:

$$D_1 = |C_c * Z_{Prey}(t_c) - Z(t_c)| \quad (1)$$

$$Z(t_c+1) = Z_{Prey}(t_c) - A_c * D_1 \quad (2)$$

Here Z and Z_{Prey} represent the position of grey wolf and prey respectively. D_1 represents distance between grey wolf and prey. t_c is current iteration and $Z_{Prey}(t_c)$ represents the position of prey at t_c . A_c and C_c are control coefficients that are responsible to take decision for grey wolves. The following equations are used for encircling the prey.

$$A_c = 2 * av * rv_1 - av \quad (3)$$

$$C_c = 2 * rv_2 \quad (4)$$

The value of av is linearly decreased from 2 to 0 during the course of iteration. rv_1 and rv_2 are random variables in the range of $[0,1]$. Hunting of prey is guided by alpha followed by beta and gamma wolves. The rest of wolves update their positions according to positions of alpha, beta, and gamma wolves.

$$D_{1\alpha} = |C_{c1} * Z_\alpha - Z|, D_{1\beta} = |C_{c2} * Z_\beta - Z|, D_{1\delta} = |C_{c3} * Z_\delta - Z| \quad (5)$$

$$Z_1 = Z_\alpha - A_c * D_{1\alpha}, Z_2 = Z_\beta - A_c * D_{1\beta}, Z_3 = Z_\delta - A_c * D_{1\delta} \quad (6)$$

$$Z = (Z_1 + Z_2 + Z_3) / 3 \quad (7)$$

The wolves are attacking on prey for hunting. The exploitation capability of wolves is used for this purpose. The control coefficient A_c is used to decide whether wolf has to explore or exploit. When A_c coefficient is greater than 1, wolves move towards the prey and attack the prey. Otherwise they will search the prey. Therefore, the value of av is linearly decreased from 2 to 0. The algorithm has few parameters to adjust and it is easy to implement. There is balance between exploration and exploitation depending on the

coefficient vector value of A_c . This algorithm has been tested on 13 benchmark functions and has shown good performance rate. The flowchart of GWA is given below in Figure 2

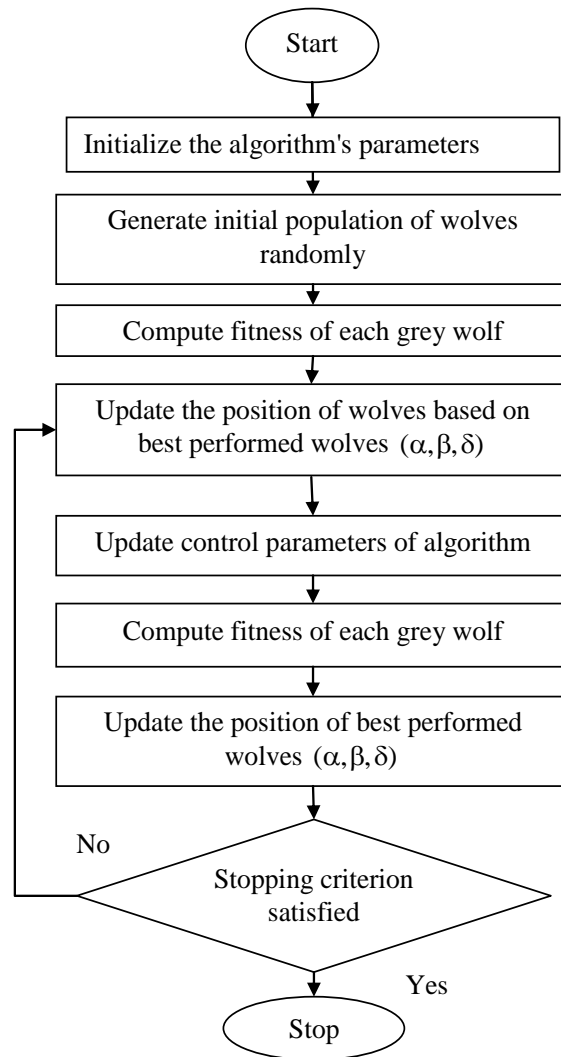


Figure 2. Flowchart of Grey Wolf Algorithm (GWA)

1.4 Applications

Metaheuristics are commonly used for solving combinatorial optimization problems. A single metaheuristic technique is not sufficient enough to solve all the problems. So, depending upon the type of the problem we use different metaheuristic techniques. The applicability of these techniques is described by describing all the problems that has been solved by them.

I. Scheduling Problems

Scheduling problems involve the allocation of the resources to different jobs. The resource can be labor, space or equipment and job can be any activity, customer or task. These problems can be taken as assignment problems. In these problems various decisions needs to be taken like how many resources to which job should be given and when to be given in order to find optimal solution. Metaheuristics techniques are used for solving scheduling problems in reasonable amount of time with better performance.

II. Space Allocation Problems

Nowadays organizing space in large organizations is typical issue. Space allocation problems refers to the problems of how space will be allocated between different departments, employees etc. Many constraints can be present like room size, people that need to be allocated in near rooms, far rooms etc. These problems of this kind are solved by metaheuristics techniques.

III. Clustering Problems

In Clustering Problems objects that are similar to each other are kept in same cluster. The objects that are different to each other belong to different clusters. These problems come under the unsupervised learning problems. The problem is to find out the best clusters and the criteria of best cluster can be many. So, metaheuristics techniques are used to solve these problems effectively.

IV. Travelling Salesman Problem

In travelling salesman problem as the size of the problem increases the search space constituting candidate solutions increase exponentially. The aim is to find best optimal path covering all the cities only once. The solution for this kind of problem is found out by using metaheuristics techniques.

V. Radio Network Design (RND) Problem

In RND problem target is to cover the geographical area with minimum number of radio antennas for saving cost expenditure. This problem is a practical telecommunication area problem and its solution can be found out by using metaheuristics techniques.

VI. Quadratic Assignment Problem

In this problem there are some n set of locations and n set of facilities. The distance between each pair of location and flow between each pair of facilities is specified. The aim is to best assign all the facilities to different location by keeping the sum of distance multiplied by facilities minimum. This problem can be solved by using metaheuristics techniques.

1.5 Thesis Organization

The organization of thesis is structured into six chapters. The brief description of each chapter is given below:-

In Chapter 1 a brief description about metaheuristics is given. It includes definition, their characteristics and classification. Under classification part, these metaheuristics are divided into four different categories. Further description of these techniques is given according to the mentioned categories. In last Grey Wolf Algorithm (GWA) is discussed that serves as a base for the proposed approach.

Chapter 2 summarizes all the metaheuristics techniques developed so far. A description of metaheuristics techniques is presented with their year wise development. Their methodologies and working is discussed starting from the beginning to the end.

In the beginning of Chapter 3, motivation behind the proposed approach is given. Thereafter, the problem statement is given. A brief description of the objectives that were kept in mind while developing the novel approach is given in the end of the chapter.

Chapter 4 provides a new methodology that has been proposed to address the major issues that were found out in classical Grey Wolf Algorithm (GWA). A flowchart for understanding the working of the proposed algorithm is given and pseudo code is also mentioned for better clearance of the algorithm.

In Chapter 5 a comparative study of the proposed algorithm with other metaheuristics techniques is presented. The proposed approach has been tested on thirteen benchmark functions. The results on unimodal and multi-modal benchmark test functions are shown separately in tables. A noisy study has also been made and convergence characteristic of the proposed algorithm is shown in figures.

Chapter 6 consists of concise discussion about how the proposed algorithm is better than the classical Grey Wolf Algorithm (GWA). A brief conclusion is given. In last, the further research that can be performed is stated under the future scope section.

Chapter 2

Literature Survey

This chapter presents a survey of all the metaheuristic algorithms that have been used extensively for solving the optimization problems. These metaheuristic techniques are discussed one by one in the order of their development.

Holland proposed genetic algorithm in 1975[5]. It is an adaptive algorithm having many variants until now. The correlation of search variables is not an issue in this algorithm. It mimics the nature inspired selection mechanisms. Mainly two things are required in a genetic algorithm. One is the fitness function and second is the genetic representation of all the solutions present in the domain. Representation in the form of array of bits is very common and tree or graph representations can also be used. The first step in the algorithm is of initialization and it starts with some random generated population. It can also be started with some population seeded into particular areas of the search space. Second step is the process of selection. In selection, the fittest candidate solutions are selected for the generation of new population. The fitness of the solutions is evaluated using fitness function. The fitness function is problem dependent. Fitness function used for one problem may not be suitable for other problem. For certain problems where it is hard to find fitness function, simulation is used. The next step after selection process is applying set of operators like crossover and mutation to generate new generation of the population. In crossover two sets of parents are needed while in mutation only one set of the parent is required to impart the changes in the population. After that in last step, algorithm is terminated if maximum iterations are reached or desired population of good candidate solutions is found.

In 1983 Kirkpatrick et al. developed Simulated Annealing [4] metaheuristic algorithm. It is based on the mechanisms of thermodynamics where metal cool down and anneal. Metal is heated at high temperature and then it is allowed to cool down. If it is cooled down quickly a high energy state is acquired but if it is cooled down slowly a perfect equilibrium is reached. The energy state of the metal that is heated corresponds to

the objective function that needs to be minimized. Both discrete and continuous variables problems can be handled by the Simulated Annealing. This algorithm is easy to implement. It takes less CPU time and has low memory requirements. It can be used to solve any optimization problem. For example travelling salesman problem can be solved by using this algorithm. The required good optimal solutions can be easily found by using this algorithm. Even in the presence of the noisy data this algorithm performs well.

Dorigo in 1992 proposed this new metaheuristic named as Ant Colony Optimization [1]. This algorithm is inspired by the behavior of the ants hunting for food. In the beginning ants have no idea about where the food is. They wander randomly. When any ant finds food then it will leave markers on the whole path so that other ants could reach the food source as well. When other ants follow the same path then the path gets populated by their markers and gets stronger. Each time ants bring food they release a chemical substance named as pheromones which help them to find even much shorter paths. Hence we are able to find out the optimal paths by mimicking the behavior of the ants in the algorithm. In this algorithm agents are termed as ants that search optimal solutions i.e. food. For applying Ant Colony Optimization firstly convert the problem in the form of a weighted graph problem. The ants tend to find out the best paths in the graph as they do in actual natural scenario. The stopping criteria of the algorithm can be maximum iteration or stagnation. This algorithm has been used widely for solving various problems specially those where the topologies keep on changing like in computer networks or artificial intelligence simulations of workers etc.

Kennedy and Eberhart in 1995 invented a new metaheuristic algorithm and termed it as Particle Swarm Optimization (PSO) [2]. This algorithm is inspired by the behavior of the swarms in nature like birds etc. We consider the scenario where birds are searching for the food. In each iteration the movement is more close to the target which is food. The best way to reach quickly towards the food is to follow the bird that is nearest in the direction of the food. PSO follows the same strategy. Here there exists swarm of particles. The particles are the solutions. These particles move into the search space for finding optimal solution i.e. food. There are two values in PSO which determine the movement of any particle. One is the best position value of the particle in neighborhood

and other is the best value obtained by any particle so far in the swarm. The position value is calculated by using some fitness function. So the potential solutions (particles) move around the search space by following optimal solutions (best particle position). PSO shows many similarities with GA (genetic algorithm). Both algorithms start with some random generated set of candidate solutions and use fitness value to evaluate the population. The updating, selection process is done by using some random technique. However PSO differs from GA in many ways. PSO does not use any operators like mutation operator, crossover operator etc. Particles in the PSO have memory and the updating mechanism is according to the velocity. The information sharing mechanism is also different in both algorithms. PSO can be used as an optimizer also. The parameters to be adjusted are few and this algorithm is easy to implement. Many complex problems of optimization have been solved by using this algorithm.

Harmonic search [17] was developed by Geem et al. in 2001. This algorithm mimics the improvisation method that Jazz musicians typically follow. Musicians find out the best harmony by playing notes. Now every musician is considered as a decision variable and notes they generate is the values they find out. The best harmony corresponds to the global optima. As the harmony gets improved time by time the same way our solution vectors gets improved by the time. Harmonic search uses stochastic derivative mechanisms. Hence it can be applied easily when we have discrete variables where other gradient methods are not suitable for applying. For example if any decision variable has three candidate value then the stochastic derivation of the objective function with respect to the decision variable yield selection probability like 30% for first, 50% for second and 20% for the third candidate value. This probability is updated on each iteration and the cumulative probability is 100%. There are many advantages of using Harmonic Search algorithm like it can be applied on both continuous as well as discrete functions. There is no need of initializing the variables. The algorithm shows no divergence and has novel stochastic derivation mechanism.

Krishnanand and Ghose invented Glowworm Swarm Optimization (GSO) [18] in 2005. It is a swarm based metaheuristic technique which mimics the behavior of the glowworms. Glowworms have capability of glowing at different intensities. In GSO

algorithm the intensity of the agent glow is approximately equivalent to the objective function value that needs to get optimized. Brighter glowworms tend to attract the glowworms of less intensity. The setting of the dynamic decision range is vital part of the algorithm. Distant glowworms effect is neglected in two conditions. Either the glowworm has sufficient number of neighbors or range is exceeded beyond the perception of the glowworm. This range setting part makes this algorithm different from the firefly algorithm where automatic subdivision of the groups of the firefly occurs. The algorithm can have cognitive limits that help it for having convergence into higher function values. Multiple optima can be calculated simultaneously for problems belonging to multimodal functions.

Artificial Bee Algorithm [19] was proposed by Karaboga in 2005. It is a swarm based metaheuristic algorithm for solving optimization problems. It is inspired by the intelligent behavior of the honey bees that shows collective and self-organizing traits. This algorithm considered three important factors. These are employed honey bees, unemployed honey bees and food source. Both employed and unemployed honey bees search for the food source. Bees move in multi-dimensional search space. The position of the food is changed if they find out food with more nectar. Bees remember the position of food having high nectar and continuously keep on searching for much better food. Now the algorithm considered bees as agents and the food source as optimal solution. Initially the bees or agents will find some solution and then after some iteration they will find optimal solution. Colony size and maximum cycle numbers are the only parameters that need to be adjusted. This algorithm has been used for solving many problems like image segmentation, structural optimization etc.

This algorithm was proposed by Hamed Shah Hosseini in 2007. It is swarm based metaheuristic technique inspired by the nature. The flow of the water drops in a river is observed. The flow of the river is resultant flow created by the swarm of the water drops. The river is an environment that has been modified by the swarm of the water drops. The path which a river follows is made up of several twists and turn and it reaches its destination following this path. The water drops are pulled by the gravitational force of the earth and it makes them run into the direction of the destination. Now there are some

traits of the water drops constituting river in the nature. Consider the scenario where a water drop has to move from one point to other point in a river bed. Every water drop has some velocity associated with it. If the speed of the water drop is high then it will transfer more soil from the river bed from one location to other location. Also if a terrain is having less soil content then the speed of the water drops will be more in such area. When there exist multiple paths for a river then the swarm of the water drops chooses the path having less soil content as it will be less resistant. Hence in this way an optimal path towards the destination can be found out. This phenomenon is mimicked in the Intelligent Water Drops Algorithm [20]. For using this algorithm the problem is represented into the form of a graph. This graph acts as an environment for the intelligent water drops and they flow on this graph. The intelligent water drops compete and cooperate with each other to find best path towards the optimal solution. The various parameters needs to be adjusted for using this algorithm like velocity of the water drops, soil etc. This algorithm shows good convergence and good optimal solutions of many optimization problems can be found out by using it.

Developed by Atashpaz Gargari in 2007, this algorithm is based on socio-political environment on the concept of the imperialism. Imperialism general meaning is the extension of the power of any government beyond its boundary. Countries try to dominate each other and want to extend their power. They may employ direct control or can indirectly control the other country by taking control of their markets, economy etc. The benefit is increased power and resource usage of that country. This behavior has resulted into the competition among countries. The one which is powerful will survive and the weak will fall apart. This whole idea or phenomenon is mimicked into the Imperialist Competitive Algorithm [21]. This algorithm starts with some random population. This population corresponds to the countries in the world and divided into two categories. In one category all the imperialist are present that are more powerful countries. In other category all the colonies are present that are less powerful countries which belongs to the imperialists. The power of the empire is taken inversely proportional of the cost value and it depends on the power of the imperialist country and its colonies. Now the competition among the empires takes place. The empires having more power and the one which is strong will survive. The weak empires will be

collapsed. The movement of the colonies is accordingly and goes hand by hand. When only the powerful empires are left then the convergence to single empire takes place. This algorithm uses two operators named as assimilation and revolution. Using assimilation all the countries come closer to the imperialist and using revolution better exploration of the search space takes place by random movement in the position of the countries. This algorithm has been used for solving various problems belonging to different domains of engineering and many variants of the algorithm are also proposed until now. Recently this algorithm has been declared as separate subject and it has been divided into two branches named as speech coding and acceptance of the failure. A different viewpoint on this algorithm is taking place and it will bring many changes on the way how it solves the problem.

Wierstra et al. proposed Natural Evolution [22] Strategies in 2008. Mainly these strategies are used for solving black box problems. These strategies belong to numerical optimization family algorithms. Search distribution parameters are updated iteratively according to the higher value of the fitness. The basic strategy is to first generate search points using some parameterized distribution. Then calculate the fitness value of each search point. The fitness function structure can be found out by using the parameters of the distribution in the algorithm. For example if we use Gaussian distribution then the parameters to be used will be mean and the covariance matrix that will help in finding the structure of the fitness function. Evaluation of the gradient is performed by the natural evolution strategy. After that gradient ascent step is applied that will normalize the update along with natural gradient. This step is important as it will prevent the premature convergence and other undesired effects which can arise due to the parameterization process. The whole process keeps on going until a stopping criterion is reached. In all the natural evolution strategies the only difference lies in the kind of distribution and gradient method used. According to the search space, select the distribution to be used. For example use full convergence matrix if dealing with low dimensionality. In case of high dimensionality, heavy tailed distributions are used and try to limit the covariance towards the diagonal only. These optimization strategies can be used to solve black box problems. There are numerous implementations of the above discussed natural evolution strategies which have been employed successfully for solving many problems.

The Firefly Algorithm [15] was proposed by Yang in 2008. This algorithm is also one of the nature inspired metaheuristic technique where the behavior of the fireflies is mimicked. Fireflies flashes light and the main purpose of their flashing is to attract other flies. Any firefly can get attracted to any other firefly. The brightness level of the firefly will determine which firefly will be attracted to whom. It is assumed that fireflies with less brightness will be attracted by the fireflies of higher brightness. The intensity of the brightness of fireflies is decreased as the distance between them increases. If any firefly fails to find out any firefly of higher brightness then it will move around randomly. The objective function value can be associated with the brightness level. Intensity variation and the brightness formulation are the two factors that need to be kept in mind in this algorithm. For further research this algorithm can be applied with some other algorithm for better performance.

Monkey algorithm is swarm based metaheuristic algorithm which was developed by Mucherino and Seref in 2008. This algorithm is inspired by the monkey's climbing process in the nature. The first step for monkey's climbing mountains is to climb. Mathematically it can be viewed as finding of local optima in a step by step process. After the first step monkey will be on the top of a mountain. Now what it will do is to have a look around on its surroundings. When it finds a mountain higher than the current mountain then automatically it will jump to higher mountain. This can be interpreted as discovering new solution in the neighborhood from the current solution. The current solution will be replaced if and only if the new neighborhood solution is better than the current solution. After that each monkey will change its direction either forward or backward based on the position of the pivot. The pivot position can be found out by taking center of all monkey's position. The purpose of this movement is to search out more new solutions from the search space. This process will be done iteratively and each monkey will be able to find highest mountain peak. The optimal solution will be found out in this way. Monkey Search Algorithm[23] can be applied to vast variety of problems irrespective of the fact that objective function can be differentiated or not. For problems of higher dimension this algorithm is suitable as the population size is unaffected by the dimension. In case of multi-modal and nonlinear problems this algorithm has been used extensively. The algorithm is stopped in either of two conditions when maximum

iterations are reached or there is no change in the optimal value for successive K generations.

Gravitational Search Algorithm [24] was developed by Rashedi in 2009. This algorithm is based on gravitational law of gravity. It is a population based metaheuristic technique that involves mass interaction. There are four parameters involved in this algorithm. These are position mass, inertial mass, active gravitational mass and passive gravitational mass. Search agents are considered as collection of the masses. After calculating the fitness value of each agent, calculate their mass. Now acceleration of the agent can be computed by considering all the forces which are taking place on the agent. Afterwards calculate velocity of the agent by making use of the computed acceleration. Then position of the agent is calculated. The algorithm is applied until the stopping criterion is reached. This algorithm is gaining popularity because of its robust nature. Better exploration and exploitation of the search space is achieved by using this algorithm. However this algorithm is complex and takes more computational time. There is much advancement and modifications proposed on this algorithm over the past few years for much better performance.

Cuckoo Search [13] was proposed by Yang and Deb in 2009. It is based on cuckoo's behavior of how they lay their eggs in hosts nest and how these eggs are hatched by the hosts. This algorithm is nature driven metaheuristic technique. Cuckoos birds are known for their beautiful voices and they have unique reproduction behavior too. They lay their eggs in the host nest. They have ability to mimic the color and pattern of the host eggs. Also they hatch their eggs earlier than the host eggs and then put their eggs in host nest. If the host finds out that the eggs are not its own then it can abandon the nest or throw away those eggs. Now in Cuckoo Search algorithm it is assumed that each cuckoo is laying one egg at a time and then it randomly put its egg in host nest. The host nest containing other eggs that are other solutions. The nest which is having best solutions will be carried to next generations. Hence we are replacing the solutions with the new solutions. There is a limit on the number of available host nest. The probability of host discovering the alien eggs is given. In case alien eggs are identified then these eggs are thrown away or get abandoned by the host. The search of food is done in a

random way in the nature by the animals. A levy flight is common way of searching the food. In levy flight, we are calculating step lengths. This can be applied to Cuckoo Search Algorithm. This algorithm has been used in solving many optimization problems.

Ali Husseinzadeh Kashan in 2009 proposed League Championship Algorithm (LCA) [25]. This algorithm mimics the championship environment where many teams are playing in a league. Teams are consisting of players playing with each other. The outcome of the game is either win or loss. The whole match is modeled artificially in the algorithm. The algorithm has artificial teams with artificial league. The number of weeks the league is played is equivalent to the number of the iterations in the algorithm. Now a team can win with its better formation or good player strength. Here in the algorithm we have player strength as the fitness value and the formation is same as the solution corresponding to each team. Every team tries to build new formations for each week in every match. Hence new solutions are generated in each contest. League Championship Algorithm is population based metaheuristic technique where there is focus on the relative comparison between individuals which leads to win portion more towards the better solutions than the win portion of the weaker solutions. Hence the search direction is automatically moves more towards the winner. There are four equations in LCA for the generation of the new solutions i.e. new formations in each match. Truncate distribution is used to stop the algorithm from premature convergence. There is proper balance between intensification and diversification in the algorithm.

Josue Cuevas et al. in 2009 proposed Virus Optimization Algorithm [26]. This algorithm imitates the behavior of the viruses attacking a living cell. Viruses have tendency to grow and immune system is responsible to keep a check on their growth. Viruses enter into living bodies by infection. Once they get entered they will start replicating and alter the genetic material of the host cell. More viruses will be produced and ultimately host cell will die. In the algorithm solution space is taken as cell itself and global optima can be found inside the cell. Many viruses can coexist within a host cell. Each such virus represents a solution in the solution space. The viruses are divided into two categories:- strong and common which corresponds to the exploration and exploitation capability of the algorithm. Strong viruses will be having high objective

function value than the common viruses and they will replicate faster than them. Common viruses will be doing the exploration work and the strong viruses will be killing the cell or we can say that they will be doing exploitation work. Mainly three types of work is done by the algorithm:- initialization ,replication and updation. Initialization occurs according to the fitness value and in replication new solutions are created. In updation work, a check on the population size is maintained. There are various parameters involved in the algorithm such as number of initial solutions, growth rate of viruses and new solutions, number of strong solutions etc. The stopping condition of the algorithm can be maximum number of the iterations or the maximum number of the replications. This algorithm is population based metaheuristic technique and it has been used to solve many optimization problems like continuous domain problems etc. One advantage of this algorithm is that it can be easily parallelized and thus it can be implemented on computers easily.

Bat algorithm [27] was proposed by Yang in 2010. This algorithm is inspired by the bats behavior of finding their prey and how they differentiating between things during the dark. There exist a wide variety of the bats. The bats don't have eyes to see. Instead they make use of echolocation which is kind of a sonar system for finding their prey, finding their home location and for the movement in right direction. The bats produce loud sound impulse and they listen to the echo that is produced as a result from the surrounding objects in order to detect direction of movement, speed and location of the object. Depending on the species this sound pulse can be a constant frequency signal or short modulated frequency signal. In the Bat Algorithm agents are taken as virtual bats that are flying with some random velocity at certain position with some frequency. The frequency is changed when a prey is found by the bat. Intensification of the search occurs by using random walk and goes till the point the best is obtained. A balance between exploration and exploitation can be found by tuning the parameters in the algorithm. This algorithm has been used in solving engineering design problems, in classification of gene expression data, in fuzzy systems and many others. There exist many variants of this algorithm until now and it is one of the popular metaheuristic techniques available today

Inspired by the galaxy in the universe this algorithm was proposed by Hamed Shah-Hosseini in 2011. In this algorithm starts with some initial solution and then the movement around the search space is organized as spiral with some particular steps and rotations in each dimension. The better solution can be found out by the gradual opening of the galaxy arm space. Hence the optimal solutions can be found out by imitating arms of the galaxies. The solutions that are obtained from the arms of the galaxy can be further tuned up by using local search algorithm in it. This algorithm has been used successfully in multilevel thresholding. For example in case of multilevel threshold the initial solution space is made up of some thresholds. The algorithm moves spirally from this initial solution which can be consider as center of the galaxy. Now the spiral movement of the arms of galaxy around the core will search out some new solutions. Then a local search algorithm is applied to find good solution from the obtained new solution. This new better solution can be considered as new core of the galaxy. The whole process is kept on going until we reach the stopping criteria. Galaxy based Search Algorithm [28] is nature inspired metaheuristic for finding optimal solution in a search space in case of continuous optimization problems.

Spiral Optimization [29] is a new metaheuristic technique for solving two dimensional continuous optimization problems. This method was proposed by Tamura and Yasuda in 2011. It is a multipoint method that is inspired by the spiral phenomenon in the nature. This phenomenon can be observed into a whirling current, low pressure, nautilus shell etc. in the nature. The focused spirals are approximated to the logarithmic spirals in the algorithm. The behavior in metaheuristics can be observed by the discrete processes which generate spirals in the logarithmic spirals. The points are rotated around the center iteratively in order to get the solutions during the convergence process and the points are attracted by the center. The exploration is done in the early stage of the convergence. On the later stages of the convergence process much better solutions are search out. So exploitation is performed in the later stages of the convergence process. This algorithm has few parameters to adjust. Recently this algorithm has been modified for solving n dimensional continuous optimization problems. The rotation matrices were defined in n dimensional space for making the two dimensional spiral into n dimensional

model. This algorithm has shown promising results and it has been used for solving various optimization problems.

This algorithm was proposed by the Civicioglu in 2012. This algorithm imitates the Brownian movement of the organisms used for migrating from one place to another place. Over the year the climatic conditions vary in the nature. So organisms move from one place to other for finding better capacity rich food areas. The species which is migrating corresponds to a superorganism that is made up of many individuals. So this superorganism starts its movement from one place to another by using Brownian movement. Now if the new area fulfills all the demands of the superorganism then it will stay there otherwise movement to a more fertile area is made by the superorganism. In Differential Search Algorithm [30], it is assumed that the random solutions belonging to the population corresponds to the migrating behavior of the superorganism. The global minimum is reached by the last one among them. This artificial superorganism may settle down on a random location if it is suitable for some time and further migration is made from that particular location. Finally the superorganism moves to the global minimum value. This algorithm is population based metaheuristic technique and has used for solving many optimization problems.

Artificial Cooperative Search [31] was introduced by Civicioglu in 2013 for the purpose of solving real valued optimization problems. This algorithm is swarm intelligence based algorithms where two superorganisms interact with each other with the purpose of achieving global minimum value. Superorganisms tend to move from one area to another area in search of better food due to climatic changes in the environment. In many superorganisms how various decisions are taken is not clear like the direction of movement, migration time etc. Two superorganisms will interact with each other if they have relationship like as that of parasite/host etc. This can result into cooperation and many other phenomenons between these superorganisms. In Artificial Cooperative Search Algorithm the solutions constitute the superorganism and are of two types. One is alpha and other is beta. They itself consists of sub-superorganisms and the number of individuals in a sub-superorganism corresponds to the dimension of the problem. The predator and prey relationship exist between the two superorganisms. The predator will

be pursuing prey superorganism in the process of achieving global minima and they exhibit cooperation between them as they interact. This algorithm takes the benefit of the social interaction between the organism and it is used for solving various numerical optimization problems.

Seyedali Mirjalili et al. in 2014 proposed Grey Wolf Optimizer [14]. This metaheuristic technique mimics the social and hunting behavior of the grey wolves. Grey wolves live in a pack from 5-12. They live in social hierarchy as alpha, beta, gamma and omega. The alpha wolves are the leader of the pack. Beta wolves follow alpha and gamma follows beta wolves. The omega wolves follow the order of all the other wolves. The best solution in the algorithm corresponds to the alpha wolves and second best to beta and third best solution as gamma wolves. The rest candidate solutions are termed as omega wolves. All the other wolves update their position based on the position of the alpha, beta and gamma wolves. The wolves firstly will search for the prey then encircle it and after that attack it. There is proper balance between exploitation and exploration in the algorithm. There are few parameters to adjust and it is simple to implement. This algorithm is used for solving many optimization problems and it is one of the recent developed metaheuristic techniques.

Multi-verse Optimizer [32] is proposed by Seyedali Mirjalili et al. in 2015. This algorithm is inspired by the cosmic concepts of the universe. The concept of black hole, white hole and worm hole has been mathematically modeled to perform the task of exploitation, exploration and local search. This algorithm is driven from the multi-verse theory of the universe which states that there occurred more than one big bang in the universe and each bang resulted into a new universe. Hence there exist other universes too and all of them interact with each other and have their own physical laws. Now each of these universes expands and reaches to suitable situation by the interaction of white hole, black hole and warm hole. In MVO for the exploration purpose white and black hole are used and for exploitation purpose warm holes are used. Every solution corresponds to universe and every variable in the solution is an object in the universe. The inflation rate of every solution is according to the fitness value. Higher probability of inflation rate means high probability of white hole and less probability of black hole.

Objects are received by black holes and send by the white holes between the universes. The best universe attracts objects of other universes by worm holes. This algorithm shows good convergence and there is proper balance between exploration and exploitation. This metaheuristic technique is one of the latest among all and has good performance rate.

Chapter 3

Problem Statement

This chapter includes motivation behind the proposed algorithm and describes the problems that have been figured out after studying the classical Grey Wolf Algorithm. The objectives that are needed to be attained for successful completion of research work are also discussed.

3.1 Motivation

The motivation behind the proposed algorithm is described in the form of key points that are given below:-

- Study shows that there is no confirmation regarding the best solution obtained by the metaheuristics to be the optimal one.
- There can be problem of premature convergence.
- Stagnation is another issue in these metaheuristics techniques.

Inspired from these facts the proposed algorithm is developed after a brief study of the classical Grey Wolf Algorithm (GWA).

3.2 Problem Statement

After studying the classical Grey Wolf Algorithm (GWA) metaheuristic technique, many shortcomings of the algorithm were figured out. The problem was to improve this classical metaheuristic technique for its better performance. The issue of how this improvement can be made in the classical Grey Wolf Algorithm (GWA) was the main focus of the problem.

3.3 Thesis Objectives

1. To study existing metaheuristic techniques for optimization problems.

2. To study and implement classical Grey Wolf Algorithm (GWA) for various optimization problems.
3. To develop a novel Grey Wolf Algorithm for improving the performance of classical Grey Wolf Algorithm (GWA).
4. To validate the proposed algorithm on the benchmark test functions.

This chapter includes a new proposed approach inspired by the classical Grey Wolf Algorithm (GWA). In the proposed algorithm there is a perfect balance between exploration and exploitation [33].

4.1 Proposed Approach

In proposed approach, a novel position updating concept is incorporated in GWA which provides better exploration and exploitation capability. The position updating concept is drawn from basic mathematical concepts. The original GWA use the circle path concept. In GWA, the wolves are moving around the prey in circle. However, we used the oval path concept in modified GWA. The oval path concept provides better exploration and exploitation as shown in Figure 3. The red wolf α is responsible for exploitation. This is considered as best solution. Other wolves are responsible for exploration. These are considered as other candidate solutions.

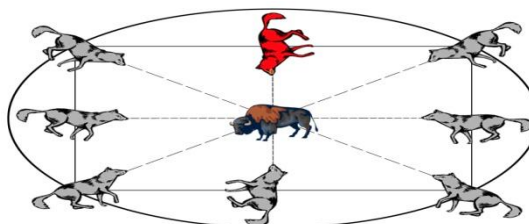


Figure 3. Movement of Grey Wolves in Oval Path

The position updating concept is incorporated in GWA. For this, first select K best grey wolves from population. The Euclidean distance between best grey wolf Z_α and other wolves is computed. This is given below.

$$Z_{i,new}(t_c+1) = Z_\alpha(t_c) + R_{i,best} * U(-2,2) \quad (8)$$

$$Z_{\alpha,new}(t_c+1) = Z_\alpha(t_c) * U(-2,2) \quad (9)$$

Here, $R_{i,best}$ is the Euclidean distance between the best grey wolf (Z_α) and the other wolves ($Z_i, i = 1, \dots, K$). The term $U(-2,2)$ returns a uniformly distributed random number

in the interval $[-2, 2]$. The proposed approach exploits the search space if $U(-2, 2)$ return a number near to 1, otherwise exploration takes place. The positions of grey wolves are updated depend upon the best grey wolf solution Z_α and computed as follows.

$$Z_i(t_c+1) = \begin{cases} Z_{i,new}(t_c+1) & \text{if } \text{fit}(Z_{i,new}) < \text{fit}(Z_i) \\ Z_i(t_c) & \text{, Otherwise} \end{cases} \quad (10)$$

The flowchart of MGWA is given in Figure 4.

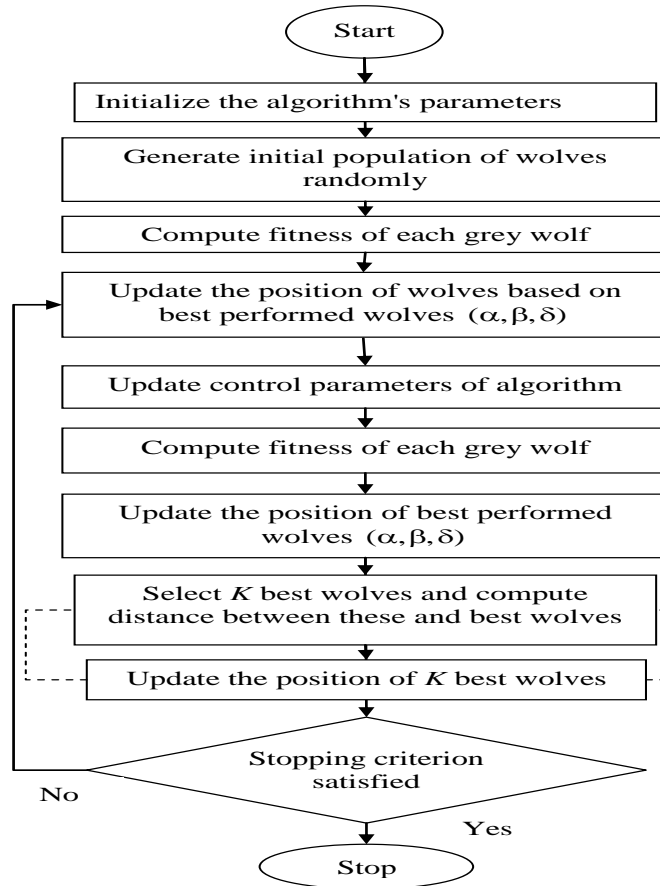


Figure 4. Flowchart of Modified Grey Wolf Algorithm

4.2 Computational Procedure of MGWA

For the better understanding of the proposed Modified Grey Wolf Algorithm (MGWA) a pseudo code is given in Figure 5. The various steps are summarized as follows:-

Step 1 : Initialize the wolves population and various parameters involved in the algorithm.

Step 2 : Apply the procedure of Grey Wolf Algorithm (GWA)

Step 3 : Select K best wolves and calculate the distance between best wolves and these selected wolves.

Step 4 : Accordingly update the position of selected K wolves.

Step 5 : Stop the algorithm if stopping criterion is met.

Modified Grey Wolf Algorithm

Input:

Population Size, n ;
Control Coefficient, av ;
Maximum number of iterations, MAX_ITER ;

Output:

Global best solution, Z_α

begin

Generate initial population of n grey wolves Z_i ($i = 1, 2, \dots, n$)

Set iteration counter $t = 0$

Compute the fitness of each grey wolf

Identify the first three best grey wolves based on fitness i.e. Z_α , Z_β and Z_δ

while ($t < MAX_ITER$)

for each grey wolf **do**

 Compute the control coefficients A_c and C_c by Eqs. 3 and 4

 Update the position of the current wolf using Z_α , Z_β and Z_δ by Eq.7

end for

 Update the coefficient av

 Compute the fitness of all grey wolves

 Update the value of Z_α , Z_β and Z_δ based on fitness

for each grey wolf **do**

 Select K best grey wolves among the Z_i .

 Find out the Euclidean distance between i th agent and the best agent Z_α

 Find out the new position of these K agents and the best agent from Eqs. 8 and 9

 Update their positions based on Eq. 10

end for

end while

 return the best solution, Z_α

end

Figure 5. Pseudo code of the Modified Grey Wolf Algorithm (MGWA)

Chapter 5

Experimental Results

This chapter includes experimentation results performed on proposed technique. The Comparative analysis shows that proposed method provides better results. Noisy study is also performed and results have been described in tabular form. Finally the convergence characteristic of proposed algorithm is given in the last.

5.1 Benchmark Test Functions

The proposed algorithm (MGWA) has been tested on thirteen benchmark functions. The effectiveness of the proposed approach is demonstrated on these thirteen benchmark functions. These are classified into two categories: unimodal and multimodal. Table 1 and 2 describe the characteristics of seven unimodal and five multimodal benchmark test functions in terms of range, dimension and optimal value. On each benchmark function the algorithm is run 20 times. For every benchmark test function the optimal value is 0 except for F_8 . On F_8 it is $-418.98 \times n$ where n refers to the dimension of the function. In the Table 2 , $-418.98 \times n$ value is denoted by x.

Table 1 Unimodal Benchmark Test Functions

Function Name	Benchmark Test Function	Range	F_{\min}
Sphere Function	$F_1(Z) = \sum_{i=1}^D z_i^2$	[100,100]	0
Schwefel 2.22 Function	$F_2(Z) = \sum_{i=1}^D z_i + \prod_{i=1}^D z_i $	[-10,10]	0
Schwefel 1.2 Function	$F_3(Z) = \sum_{i=1}^D \left(\sum_{j=1}^i z_j \right)^2$	[100,100]	0
Schwefel 2.21 Function	$F_4(Z) = \max_i \{ z_i , 1 \leq i \leq D \}$	[100,100]	0
Rosenbrock Function	$F_5(Z) = \sum_{i=1}^{D-1} \left[100(z_{i+1} - z_i^2)^2 + (z_i - 1)^2 \right]$	[-30,30]	0

Step Function	$F_6(\mathbf{Z}) = \sum_{i=1}^D (z_i + 0.5)^2$	[100,100]	0
Quadratic Noi-seU Function	$F_7(\mathbf{Z}) = \sum_{i=1}^D iz_i^4 + \text{rand}(0,1)$	[1.28,1.28]	0

Table 2 Multi-modal Benchmark Test Functions

Function Name	Benchmark Test Function	Range	F_{\min}
Schwefel 7 Function	$F_8(\mathbf{Z}) = \sum_{i=1}^D -z_i \sin(\sqrt{ z_i })$	[-500,500]	X
Rastrigin's Function	$F_9(\mathbf{Z}) = \sum_{i=1}^D [z_i^2 - 10 \cos(2\pi z_i) + 10]$	[5.12,5.12]	0
Ackley1 Function	$F_{10}(\mathbf{Z}) = -20 \exp\left(-0.2 \sqrt{\frac{1}{D} \sum_{i=1}^D z_i^2}\right) \dots$ $-\exp\left(\frac{1}{D} \sum_{i=1}^D \cos(2\pi z_i) + 20 + e\right)$	[-32,32]	0
Griewank's Function	$F_{11}(\mathbf{Z}) = \frac{1}{4000} \sum_{i=1}^D z_i^2 - \prod_{i=1}^D \cos\left(\frac{z_i}{\sqrt{i}}\right) + 1$	[-600,600]	0
Penalty1 Function	$F_{12}(\mathbf{Z}) = \frac{\pi}{D} \left\{ 10 \sin(\pi x_i) + \sum_{i=1}^{D-1} (x_i - 1)^2 \dots \right.$ $\left. \dots [1 + 10 \sin(\pi x_{i+1})] + (x_D - 1)^2 \right\}$ $+ \sum_{i=1}^D u(z_i, 10, 100, 4)$ $x_i = 1 + \frac{z_i + 1}{4}$ $u(z_i, a, k, m) = \begin{cases} k(z_i - a)^m & z_i > a \\ 0 & -a < z_i < a \\ k(-z_i - a)^m & z_i < -a \end{cases}$	[-50,50]	0
Levy 13 Function	$F_{13}(\mathbf{Z}) = 0.1 \left\{ \sin^2(3\pi z_1) + \sum_{i=1}^D (z_i - 1)^2 \dots \right.$ $\left. \dots [1 + \sin^2(3\pi z_i + 1)] + \dots \right.$ $\left. \dots (z_D - 1)^2 [1 + \sin^2(2\pi z_D)] \right\} +$ $\sum_{i=1}^D u(z_i, 5, 100, 4)$	[-50,50]	0

5.2 Parameter Setting

The parameters value used for the experimentation purpose plays a vital role in the performance of the selected algorithm. Three metaheuristics algorithms are used in the experimental study. The parameters values used in these algorithms are given below in the Table 3. For PSO the values of the constants c_1 and c_2 is taken as 2. W_{max} and W_{min} refers to the maximum and minimum inertial weight. For GWA and MGWA av is the control coefficient value and V_{max} stands for the maximum velocity.

Table 3 Parameter values

Algorithm	Population Size	Maximum Iterations	Search agents	c_1	c_2	av	W_{max}	W_{min}	V_{max}
GWA	n	500	30	-	-	[2,0]	-	-	-
MGWA	n	500	30	-	-	[2,0]	-	-	-
PSO	n	500	30	2	2	-	0.9	0.2	6

5.3 Performance Comparison

The performance of proposed approach (MGWA) is compared with Particle Swarm Optimization (PSO) and Grey Wolf Algorithm (GWA) metaheuristic techniques. The results are evaluated on the basis of mean and standard deviation value. Table 4 and 5 show the results on unimodal and multimodal benchmark functions respectively. On F_1 the mean value obtained from GWA is better than PSO and the best result is obtained on MGWA. For function F_2 also, MGWA outperforms the other two algorithms. On F_3 the best result is obtained from MGWA while GWA and PSO give second and third best results. The similar results are obtained on F_4 also. From F_1 to F_4 same values are obtained on each algorithmic run. On F_5 GWA shows better results than PSO while MGWA gives best results among all. For function F_6 , MGWA results are better than GWA but best results are given by PSO. On F_7 the best result is obtained from MGWA while GWA and PSO give second and third best results. F_8 results are similar to F_7 . MGWA shows best results on F_9 and F_{11} while second and third best results are obtained by GWA and PSO. On F_{10} also MGWA outperforms the other two algorithms. It is

observed that the same results are obtained on each algorithmic run of MGWA on F_9 , F_{10} and F_{11} . On F_{12} , PSO shows best results while GWA and MGWA gives second and third best results. Similar results are obtained on F_{13} also. The results on these thirteen benchmark functions clearly reveal that MGWA outperform other two algorithms.

Table 4 Results on Unimodal Benchmark Test Functions

	PSO	GWA	MGWA
F_1	1.33E-04 (1.16E-04)	5.00E-28 (4.68E-28)	0.00E+00 (0.00E+00)
F_2	2.54E-02 (3.52E-02)	1.20E-16 (6.47E-17)	0.00E+00 (0.00E+00)
F_3	7.01E+01 (3.02E+01)	1.83E-05 (2.48E-05)	0.00E+00 (0.00E+00)
F_4	1.11E+00 (3.08E-01)	8.50E-07 (7.33E-07)	0.00E+00 (0.00E+00)
F_5	8.85E+01 (6.70E+01)	2.71E+01 (6.83E-01)	1.76E+01 (1.17E+01)
F_6	6.29E-04 (1.20E-03)	7.45E-01 (4.28E-01)	5.48E-01 (4.28E-01)
F_7	1.66E-01 (5.27E-02)	2.10E-03 (1.20E-03)	1.24E-04 (1.35E-04)

Table 5 Results on Multi-modal Benchmark Test Functions

	PSO	GWA	MGWA
F_8	-4.63E+03 (1.32E+03)	-6.18E+03 (9.07E+02)	-1.94E+09 (6.15E+09)
F_9	5.71E+01 (1.11E+01)	3.31E+00 (5.28E+00)	0.00E+00 (0.00E+00)
F_{10}	1.57E-01 (3.71E-01)	1.04E-13 (1.70E-14)	8.88E-16 (0.00E+00)
F_{11}	9.60E-03 (7.30E-03)	5.00E-03 (1.00E-02)	0.00E+00 (0.00E+00)
F_{12}	5.20E-03 (2.32E-02)	4.60E-02 (2.13E-02)	-1.20E+02 (1.11E+01)
F_{13}	3.30E-03 (5.20E-03)	6.93E-01 (2.52E-01)	-1.11E+02 (9.38E+00)

5.4 Noisy Study

The effect of noise degrades the performance of the metaheuristic techniques. A study of PSO, GWA and MGWA metaheuristic algorithms has been done on thirteen noisy benchmark functions. A noisy benchmark function can be defined as:-

$$F_{\text{noisy}}(Z) = F(Z) + N(0,1) \quad (10)$$

where $N(0,1)$ refers to the Normal Distribution having mean value 0 and standard deviation value as 1. The aim of this study is to test the performance of proposed approach (MGWA) in presence of noise. Separate results for both unimodal and multimodal noisy benchmark functions are shown in the Table 5 and Table 6. On unimodal noisy benchmark functions MGWA performs better than other two except on F_3, F_4 and F_6 . On all these three functions, PSO gives best result while MGWA gives better results than GWA. On multi-modal functions except on F_{10} and F_{11} the proposed approach MGWA gives best results. On F_{10} and F_{11} functions best results are shown by PSO followed by MGWA and GWA respectively.

Table 6 Results on Unimodal Benchmark Test Functions

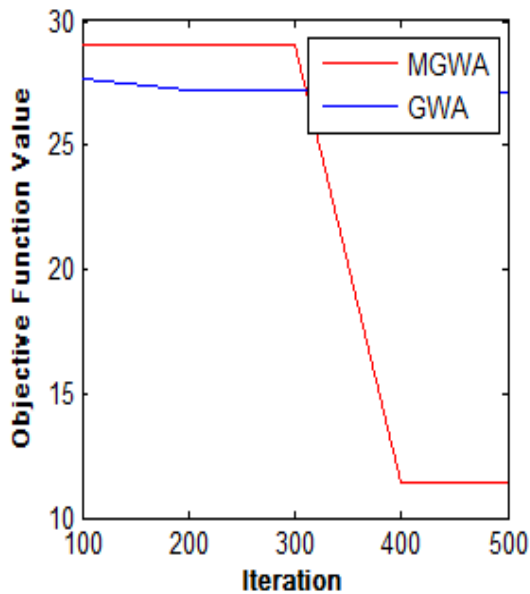
	PSO	GWA	MGWA
F_1	-3.85E-01 (7.69E-01)	-3.30E+00 (4.37E-01)	-3.92E+00 (2.79E-01)
F_2	2.16E+00 (8.89E-01)	-2.68E+00 (5.62E-01)	-3.88E+00 (2.41E-01)
F_3	1.12E+02 (4.85E+01)	-1.39E+01 (3.11E+00)	-2.18E+01 (1.29E+00)
F_4	5.96E+00 (2.81E+00)	-1.88E+00 (4.99E-01)	-4.04E+00 (3.13E-01)
F_5	9.91E+01 (9.45E+01)	2.53E+01 (5.20E-01)	2.50E+01 (2.50E+01)
F_6	-3.32E-01 (8.61E-01)	2.55E-01 (6.96E-01)	2.11E+00 (2.57E+00)
F_7	-9.74E-01 (6.72E-01)	-3.19E+00 (3.78E-01)	-3.74E+00 (4.55E-01)

Table 7 Results on Multi-modal Benchmark Test Functions

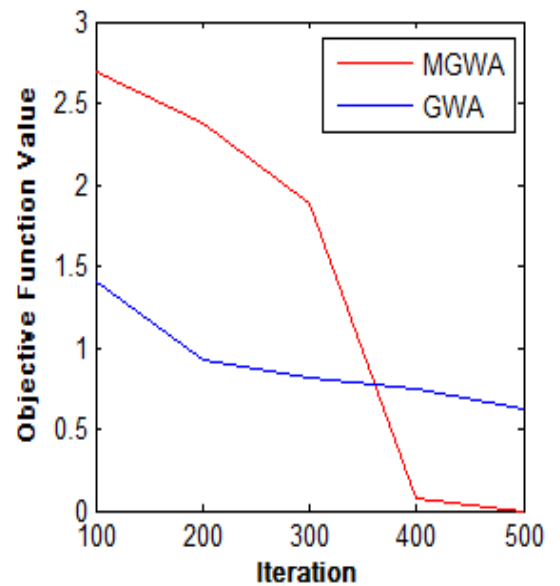
	PSO	GWA	MGWA
F ₈	-4.78E+03 (1.33E+03)	-5.93E+03 (1.02E+03)	-1.10E+09 (3.36E+09)
F ₉	6.11E+01 (1.02E+01)	3.70E-01 (5.48E+00)	-4.01E+00 (3.76E-01)
F ₁₀	7.61E+00 (4.54E+00)	-3.11E-02 (4.11E+00)	-4.01E+00 (2.52E-01)
F ₁₁	7.17E+00 (2.72E+00)	-2.37E+00 (4.55E-01)	-3.88E+00 (2.64E-01)
F ₁₂	-8.67E+01 (9.18E+00)	-1.07E+02 (9.26E+00)	-1.19E+02 (8.73E+00)
F ₁₃	-8.00E+01 (1.31E+01)	-9.68E+01 (1.11E+01)	-1.17E+02 (1.06E+01)

5.5 Convergence Characteristics

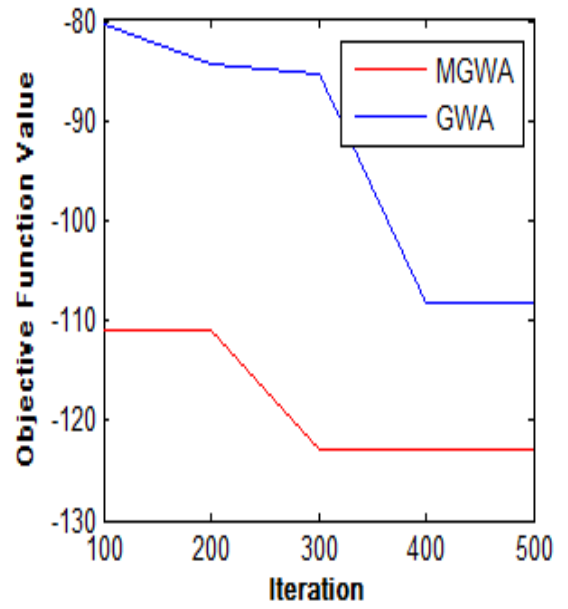
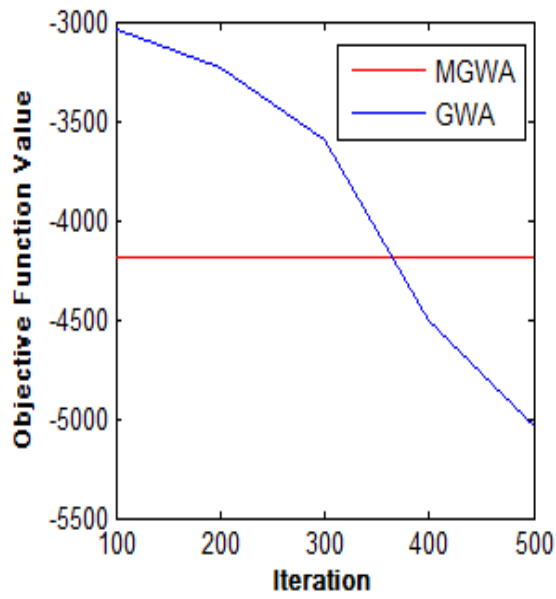
The convergence characteristic of MGWA is shown in the Figure 6 below on different benchmark functions. It has been observed that MGWA has better convergence than GWA.



(a)



(b)



(c)(d)

Figure 6. Convergence characteristics of GWA and MGWA (a) on F_5 (b) on F_6 (c) on F_8 (d) on F_{13}

Chapter 6

Conclusion and Future Scope

This chapter provides conclusion followed by future scope of the proposed approach.

6.1 Conclusion

- A brief study on existing metaheuristic techniques is performed for understanding their working methodology.
- As the base approach, classical Grey Wolf Algorithm (GWA) metaheuristic technique is examined in detail and implemented.
- Inspired by the problems that can arise in classical GWA, a novel approach (MGWA) is developed.
- Validation of the proposed algorithm is made by testing it on thirteen benchmark functions.
- A comparative analysis is performed with other metaheuristic techniques in order to judge the performance of proposed approach.
- Experimentation results confirmed that the proposed approach is better than the other metaheuristic techniques.

6.2 Scope for Future Work

In future work, the proposed approach can be further extended by hybridizing it with other metaheuristic techniques. There is possibility of getting much better results if hybridization methodology is used. The novel approach can be used for solving problem like tension/compression spring, pressure vessel design, welded beam, optical buffer design etc. This approach can also be applied on clustering problems which are one of the most frequently encountered problems in practical environment. In the field of artificial intelligence, this approach can be used for training the neural networks. There is wide range of applicable areas where the proposed approach can be used and tested.

References

- [1] A.Colomi , M.Dorigo and V.Maniezzo.Distributed Optimization by Ant Colonies , Proceedings of the First European Conference on Artificial Life, Paris, France, Elsevier Publishing, 134-142, 1991
- [2] J.Kennedy and R.Eberhart. Particle Swarm Optimization,Proceedings of IEEE International Conference on Neural Networks, 1942–1948, 1995
- [3] F.Glover. Future Paths for Integer Programming and Links to Artificial Intelligence, Computers and Operations Research 13 (5): 533–549, 1986
- [4] S.Kirkpatrick, C.D.Gelatt and M.P.Vecchi. Optimization by Simulated Annealing ,Science 220 (4598): 671–680, 1983
- [5] J.Holland. Adaptation in Natural and Artificial Systems. Cambridge, MA: MIT Press. ISBN 978-0262581110, 1992
- [6] C.Blum and A.Roli. Metaheuristics in combinatorial optimization: Overview and conceptual comparison. ACM Computing Surveys (CSUR), 35(3), 268-308, 2003
- [7] I.H.Osman and G.Laporte. Metaheuristics: A bibliography. Annals of Operations research, 63(5), 511-623, 1996
- [8] M. Dorigo. Metaheuristics network website 2000,2009
- [9] M.Birattari, L.Paquete, T.Strutzle and K.Varrentrapp. Classification of Metaheuristics and Design of Experiments for the Analysis of Components Tech. Rep. AIDA-01-05, 2001
- [10] H.R Lourenço, O.Martin and T.Stützle. Iterated Local Search: Framework and Applications, Handbook of Metaheuristics, 2nd. Edition. Kluwer Academic Publishers, International Series in Operations Research & Management Science 146: 363–397, 2010

- [11] N. Mladenović and P.Hansen. Variable neighborhood search, *Computers and Operations Research* 24 (11): 1097–1100, 1997
- [12] A.R.Wallace. *Darwinism: an exposition of the theory of natural selection with some of its applications*. Cosimo, Inc., 2007
- [13] X.S.Yang and S.Deb. Cuckoo search via Lévy flights, *Nature & Biologically Inspired Computing, NaBIC 2009. World Congress on. IEEE*, 2009
- [14] S.Mirjalili, S.M.Mirjalili and A.Lewis. Grey wolf optimizer, *Advances in Engineering Software* 69 : 46-61,2014
- [15] X.S.Yang. Firefly algorithm, stochastic test functions and design optimization, *International Journal of Bio-Inspired Computation* 2(2): 78-84, 2010
- [16] L.D.Mech. Alpha status, dominance, and division of labor in wolf packs, *Canadian Journal of Zoology*, 77(8):1196-1203, 1999
- [17] K.S.Lee and Z.W.Geem. A new metaheuristic algorithm for continuous engineering optimization: harmony search theory and practice, *Computer methods in applied mechanics and engineering* 194(36): 3902-3933, 2005
- [18] K.N.Krishnanand and D.Ghose. Glowworm swarm optimization for simultaneous capture of multiple local optima of multimodal functions, *Swarm intelligence* 3(2): 87-124, 2009
- [19] D.Karaboga. Artificial bee colony algorithm, *Scholarpedia* 5(3): 6915, 2010
- [20] H.Shah-Hosseini. The intelligent water drops algorithm: a nature-inspired swarm-based optimization algorithm, *International Journal of Bio-Inspired Computation* 1(1-2): 71-79, 2009
- [21] E.Atashpaz-Gargari and C.Lucas. Imperialist competitive algorithm: an algorithm for optimization inspired by imperialistic competition, *Evolutionary Computation, 2007.CEC 2007. IEEE Congress on. IEEE*, 2007

- [22] D.Wierstra, T.Schaul, T.Glasmachers, Y.Sun, J.Peters, and J.Schmidhuber. Natural evolution strategies, *Evolutionary Computation*, 2008. CEC 2008 (IEEE World Congress on Computational Intelligence), IEEE Congress on. IEEE, 2008.
- [23] R.Zhao and W.Tang. Monkey algorithm for global numerical optimization, *Journal of Uncertain Systems* 2(3) : 165-176,2008
- [24] E.Rashedi, H.Nezamabadi-Pour and S. Saryazdi. GSA: a gravitational search algorithm, *Information sciences* 179(13): 2232-2248, 2009
- [25] A.H.Kashan. An efficient algorithm for constrained global optimization and application to mechanical engineering design: League championship algorithm (LCA)," *Computer-Aided Design* 43(12): 1769-1792, 2011.
- [26] J.R.C.Juarez, H.J. Wang, Y.C. Lai and Y.C Liang. Virus Optimization Algorithm (VOA): A Novel Metaheuristic for Solving Continuous Optimization Problems
- [27] X.S.Yang. A new metaheuristic bat-inspired algorithm, *Nature inspired cooperative strategies for optimization (NICSO 2010)*, Springer Berlin Heidelberg, 65-74,2010
- [28] H.Shah-Hosseini. Principal components analysis by the galaxy-based search algorithm: a novel metaheuristic for continuous optimization, *International Journal of Computational Science and Engineering* 6.1(2): 132-140, 2011
- [29] K.Tamura and K.Yasuda. Primary study of spiral dynamics inspired optimization, *IEEJ Transactions on Electrical and Electronic Engineering* 6.S1: S98-S100, 2011
- [30] P.Civicioglu. Transforming geocentric cartesian coordinates to geodetic coordinates by using differential search algorithm, *Computers & Geosciences* 46: 229-247, 2012
- [31] P.Civicioglu. Artificial cooperative search algorithm for numerical optimization problems, *Information Sciences* 229: 58-76, 2013

- [32] S.Mirjalili, S.M.Mirjalili and A.Hatamlou. Multi-verse optimizer: a nature-inspired algorithm for global optimization, *Neural Computing and Applications*: 1-19, 2015.
- [33] E.Alba and B.Dorronsoro. The exploration/exploitation tradeoff in dynamic cellular genetic algorithms, *Evolutionary Computation, IEEE Transactions on*9(2), 126-142, 2005

List of Publications

Accepted:

- [1] Seema and V. Kumar, "Modified Grey Wolf Algorithm for Optimization Problems," in *International Conference on Inventive Computation Technologies ICICT, 2016*

Video Link

<http://youtu.be/F2noNSLZnEA>

Plagiarism Report

m.tech thesis

ORIGINALITY REPORT

14%	8%	12%	1%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	www.cs.uoi.gr Internet Source	1%
2	Lecture Notes in Computer Science, 2013. Publication	1%
3	Lecture Notes in Computer Science, 2014. Publication	<1%
4	www.asap.cs.nott.ac.uk Internet Source	<1%

