Supervised By:

**Syed Qamar Askari**

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| LITERATURE REVIEW 1.0  Global Optimization Using Meta-Heuristics | |  |  | | --- | --- | | Faiza Shanawar | 15140070 | | Haider Ali | 15140101 | | Mohsin Qamar | 15140104 | | Usama Imran | 15140098 | |

Contents

[1. Introduction: 2](#_Toc6332175)

[1.1. Mathematical Optimization: 2](#_Toc6332176)

[1.1.1. Types: 2](#_Toc6332177)

[1.1.2. Applications: 3](#_Toc6332178)

[1.2. Meta-Heuristics: 3](#_Toc6332179)

[1.3. Related Work: 3](#_Toc6332180)

[1.4. Benchmark Functions: 4](#_Toc6332181)

[1.4.1. Types: 4](#_Toc6332182)

[1.4.2. Overview: 4](#_Toc6332183)

[2. Literature Review: 6](#_Toc6332184)

[2.1. Genetic Algorithm 6](#_Toc6332185)

[3. References 9](#_Toc6332186)

***Document Information***

|  |  |
| --- | --- |
| User |  |
| Project | Meta-Heuristics for Global Optimization |
| Document Version | 1.0 |
| Document ID | AR-02 |
| Identifier | Literature Review 1.0 |
| Status | Draft |
| Authors(s) | Usama Imran |
| Approver(s) | Syed Qamar Askari |
| Issue Date |  |

# Introduction:

## Mathematical Optimization:

Mathematical Optimization can be defined as a technique of finding a maximum or minimum value of a function of several variables subject to set of constraints. More generally, optimization includes finding "best available" values of some objective function given a defined domain (or input), including a variety of different types of objective functions and different types of domains.



Graph of a paraboloid given by

z = f(x, y) = −(x² + y²) + 4. The global maximum at (x, y, z) = (0, 0, 4) is indicated by a blue dot.

### Types:

Optimization is classified into following types:

* Discrete optimization
* Continues optimization
* Single Objective Optimization
* Multi Objective Optimization
* Many Objective Optimization
* Combinatorial optimization
* Unconstrained optimization
* Constrained optimization

### Applications:

Optimization techniques are being used in real world problems some of their applications are as follows:

* Feature Selection
* Automatic Clustering
* Time Scheduling
* Wireless Sensor Network Optimization
* Vehicle Routing Problem
* Watermarking
* Bioinformatics
* Circuit Designing
* Game Strategy Planning
* Power Supply Management

## Meta-Heuristics:

Heuristic is a Greek word which means “to solve”. It pertains to trial-and-error method of problem solving used when an exact algorithmic approach is impractical. Main characteristic of meta-heuristics is that they are problem independent. Meta-heuristics give us a way to solve complex problems that are not solvable in polynomial time (NP-Hard Problems). Although they don’t give exact solution of a particular problem, meta-heuristics provide guidelines that can give best solution available

## Related Work:

Following are the lists of optimization Algorithms:

|  |  |
| --- | --- |
| **Algorithm** | **Inspiration** |
| Genetic Algorithm | Darvin’s Theory of Evolution, solutions are subjected to crossovers and mutation same as proposed by Darvin. |
| Gray Wolf Optimization | This algorithm is inspired by pack of wolves that are search for hunt. |
| Particle Swarm Optimization | This algorithm is inspired by flock of birds. |
| Gravitational Search Algorithm | This algorithm is inspired by Newton’s Law of Gravitation. |
| Whale Optimization | This algorithm is inspired by Whale hunting. |
| Dragonfly Optimization | This algorithm is inspired by dragonflies and their hunting behavior. |
| Centipede Optimization | This algorithm is inspired by centipede’s hunting behavior |
| Bat Optimization | This algorithm is inspired by bat’s echolocation |
| Ant Colony Optimization | This algorithm is inspired by ant’s colony |
| Artificial Bee Colony Optimization | This algorithm is inspired by Bees |

## Benchmark Functions:

After the algorithms are proposed, they are tested on some evaluation functions which are known as benchmark functions. Algorithms are subjected to benchmark functions their performance is checked through comparison with already existed algorithms.

### Types:

Following are the types of benchmark functions:

1. Unimodal
2. Multimodal
3. Differentiable
4. Non-differentiable
5. 1- Dimensional
6. 2-Dimentional
7. 3-Dimensoinal
8. N-Dimensional
9. Convex
10. Non-convex
11. Parametric
12. Separable

### Overview:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ***Function name*** | ***Graph*** | ***Equation*** | ***Nature*** | ***Range*** |
| Ackley N. 2 Function |  | f(x,y)=−200e−0.2√x2+y2 | N-dimensional, unimodal, convex, differentiable | [-6,6] |
| Booth Function |  | f(x,y)=(x+2y−7)2+(2x+y−5)2 | 2-dimensional, continuous, convex, differentiable, non-separable, unimodal convex | [−10,10] |
| Brent Function |  | f(x,y)=(x+10)2+(y+10)2+e−x2−y2 | 2-dimensional continuous convex differentiable non-separable unimodal convex | [−10,10] |
| Drop-Wave Function |  | f(x,y)=−1+cos(12√x2+y2) (0.5(x2+y2)+2) | 2-dimensional continuous unimodal non-convex | [−5.2,5.2] |
| Exponential Function |  | f(x)=f(x1,...,xn)= −exp(−0.5n∑i=1x2i) | n-dimensional continuous differentiable non-separable unimodal convex | [−1,1] |
| Leon Function |  | f(x,y)=100(y−x3)2+(1−x)2 | 2-dimensional continuous differentiable non-separable unimodal non-convex | [0,10] |
| Deckkers-Aarts Function |  | (x,y)=105x2+y2−  (x2+y2)2+10−5(x2+y2)4 | 2-dimensional continuous differentiable non-separable multimodal non-convex | [−20,20] |
| Styblinski-Tank Function |  | f(x)=f(x1,...,xn)=  12n∑i=1(x4i−16x2i+5xi) | n-dimensional continuous multimodal non-convex | [−5,5] |
| Bartels Conn Function |  | f(x,y)=  |x2+y2+xy|+|sin(x)|+|cos(y)| | 2-dimensional non-separable multimodal non-convex non-differentiable | [−500,500] |
| Schwefel 2.20 Function |  | f(x)=f(x1,...,xn)=n∑i=1|xi| | 2-dimensional continuous differentiable non-separable unimodal non-convex | [−100,100] |
| Egg Crate Function |  | f(x,y)=x2+y2+25(sin2(x)+sin2(y)) | 2-dimensional continuous differentiable separable multimodal non-convex | [−5,5] |
| Shubert Function |  | f(x)=f(x1,...,xn)=  n∏i=1(5∑j=1cos((j+1)xi+j)) | n-dimensional continuous differentiable non-separable multimodal non-convex | [-10,10] |
|  |  |  |  |  |

# Literature Review:

## Genetic Algorithm

**Genetic Algorithm** is one of the most famous evolutionary algorithms. GA is inspired by Darwin’s theory of evolution. There is selection of population called “Chromosomes” and then they are subjected to crossover and mutation processes (Same as biological process). There are multiple applications of GA like: product designing, automotive designing, capacitated vehicle routing problem. [1]



On the other hand, **Gravitational Search Algorithm** is Physics based algorithm which is, as the name suggests, inspired from Newton’s law of Gravitation. In this algorithm, every solution is treated as an object. And the object’s fitness is determined by the mass of that object, greater the mass, greater would be the fitness. The object of higher fitness attracts the object of lower fitness by following the rules of physics. Hence, this algorithm also has many useful applications. GSA is used in Economic Load Dispatch Problem, Energy Management System, Feature Subset Selection, Training the Neural Networks, Unit Commitment Problem [2].



**Particle Swarm Optimization** is Swam based intelligence algorithm which is inspired by swarm of fish, and birds. In this algorithm, every solution is treated as particle and there runs a swarm and all the particles moves around the global best solution. This algorithm is used in: multimodal optimization problems, production scheduling, power system operations, cryptarithmetics and many more. [3]



**Gray Wolf Optimization** is also well known meta-heuristic algorithm inspired by the pack of wolves which are hunting their prey. In this algorithm, each solution is treated as wolf and the highest fitness wolf is considered as “alpha”, second highest fit wolf is known as “beta”, third one is called “delta” and all the others are known as “omega”. All wolves follow the alpha wolf and they get the global best solution which is called “prey”. [4]



**Simulated Annealing** is also well known meta-heuristic algorithm. It mimics the annealing process in material processing when a metal cools and freezes into a crystalline state with minimum energy. The annealing process involves the careful control of temperature and its cooling schedule. [5]

**Soccer League Competition Algorithm** is also an optimization algorithm inspired by the optimization of football league competitions. All teams play 2 matches with other respective team. Total matches depend upon the total number of teams competing in the tournament by (M\*(M-1))/2 (where M is the total no of teams). Each team wants to top the table at the end of each iteration. Teams which consists high performed or high fitness players has more probability to win matches against opponent teams. The team fitness is calculated by the average total fitness of the players. Each team has 11 fixed players and 11 substitute players. Every team has a SP (Star Player) and the tournament has an SSP (super star player) which has best fitness among team and best fitness among the whole tournament players respectively. The winning and losing team applying different strategies to perform better in next matches. Winning team fixed players try to imitate SP (star player) of the team and SSP (super star player) of the team. Substitutes of the winning team tries to improve their performance by making their fitness at least at the average of fixed players of the team. On the other hand, fixed players of losing team tries to improve their performance by changing position of players. The losing team substitutes pairs are being entered by a certain probability to make winning probability chances. At the end of the tournament, best teams buy players with best fitness and average and weak players are bought by weak team. SSP is the optimal and SP is the local optima of the solution. [6]

**Social Evolution Algorithm** is inspired by human’s interactions and beliefs. The individuals interact and share information to its neighbor. This Algorithm have three phases: initialization phase, evaluation phase and interaction phase. In this Algorithm, von Neumann Neighborhood architecture is adopted for building the neighborhood. Each Individual’s fitness and probability is calculated. The individuals Evaluate the neighbor based on co-operation factor (controlled parameter) and ability and productivity of the neighbor, then interact with the identified neighbor. The individual will not interact with any random solution in the society instead, they may interact more with the random neighbor in the von Neumann neighborhood architecture because of affinity and trust worthiness, but they are free to explore the society based on NCF (controlled parameter). Also, once the individual is selected for the interaction, the individual solution interacts with the selected individual for all the dimensions of the problem. Once the interaction is performed, individual evaluate the quality of interaction (QI). If the quality of interaction is inferior, interaction’s indecisive factor IDF is evaluated to decide on the interaction as negative or indecisive. All the indecisive interactions will undergo a second opinion process. In the second opinion process, the individual can consult an expert either from the neighborhood or from the society or a non-existing individual with the average capabilities to further evaluate the indecisive interaction before adopting the change to emerge and evolve. After the interaction phase, evaluate the fitness of the updated solutions and compare with the respective original solution to consider the best for next generation. Before the above process is repeated until a termination condition (maximum cycle number), calculate the probabilities of the individuals, average solution the best in the society for the next generation. [7]

Hence, there are many other algorithms that are inspires by either nature, physics or some rules. More nature inspired algorithms are: Whale Optimization, Dragonfly Algorithm, Moth-Flame Optimization Algorithm, Whirlpool optimization etc.

# References

|  |  |
| --- | --- |
| [1] | P. a. K. H. Kán, "Automated interior design using a genetic algorithm," pp. 1-10, 2017. |
| [2] | H. A. Nazmul SIDDIQUEa, "APPLICATIONS OF GRAVITATIONAL SEARCH ALGORITHM IN," *JOURNAL OF CIVIL ENGINEERING AND MANAGEMENT,* p. 10, 2016. |
| [3] | M. L. Settles, "An Introduction to Particle Swarm Optimization," 2005. |
| [4] | Seyedali Mirjalili a,⇑, Seyed Mohammad Mirjalili b, Andrew Lewis a, "Grey Wolf Optimizer," *Advances in Engineering Software,* p. 46–61, 2014. |
| [5] | S. Kirkpatrick, C. D. Gelatt, Jr., M. P. Vecchi, "Optimization by Simulated Annealing," *Science,* vol. 220, no. 4598, pp. 671-681, 1983. |
| [6] | N. Moosavian, "Soccer League Competition Algorithm," *International Journal of Intelligence Sciences,* 2014. |
| [7] | R. S. P. a. Gursaran, "Social Evolution: An Evolutionary Algorithm," *Advances in Intelligent Systems,* 2014. |