GE-International Journal of Engineering Research

Vol. 4, Issue 3, March 2016 IF- 4.721 ISSN: (2321-1717)

Website: www.aarf.asia Email : editor@aarf.asia , editoraarf@gmail.com

THE PREDATOR - PREY APPROACH TO FUNCTION OPTIMISATION

[1] Errol Xavier Lobo [2] Archana Pandita,

[1, 2,] Department of Computer Science and Engineering, Birla Institute Of Technology, Offshore Campus, Ras Al Khaimah, UAE

ABSTRACT

Genetic algorithms are powerful tools for solving certain problems. A Predator-Prey approach resembles natural ecology wherein species groups compete against each other, according to a natural hierarchy. Predation in biology occurs when a predator devours another living animal (prey) to use the energy and nutrients for growth, maintenance and reproduction. Such predation also can be introduced to Genetic Algorithms for the solving of constrained multi-objective function optimisation problems. This project is aimed at studying and implementing the Predator-Prey Approach to constrained, multi-objective Function Optimisation and the effects of variations introduced into the approach. The use of predation together with classical GA operators of Crossover, Mutation, Dominance and Inversion is also investigated.

Keywords—*Genetic Algorithms, Predator Prey approach, optimization function, lattice, fitness function, elitist ,dominance, Predation with Crossover, Mutation, Inversion and Diversity Preservation*

I. INTRODUCTION

Genetic algorithms are search algorithms based on the mechanics of natural selection and natural genetics. They combine survival of the fittest among strings structures with a structured yet randomised information exchange to form a search algorithm with some of the innovative flair of human search. In every generation, a new set of artificial creatures (strings) is created using bits and pieces of the fittest of the old; an occasional new part is tried for good measure. While randomised, genetic algorithms are no simple random walk. They efficiently exploit historical information to speculate on new search points with expected improved

A simple Genetic Algorithm

An evolutionary algorithm at its simplest form can be stated as in [2]:

Initiate current population arbitrarily
WHILE the termination criterion is not fulfilled
create empty temporary population
WHILE temporary population does not fulfil
select parents
cross parents with probability Pc
IF crossing has occurred
 - Mutate one of the descendants with probability Pm
- evaluate descendants
-add descendant to the temporary population
OTHERWISE
-add parents to the temporary population
END IF
END WHILE
increase generations counter
establish the temporary population as new current population
END WHILE

II. THE PREDATOR – PREY APPROACH

Biologically, predation occurs when one of the animals (the predator) devours another living animal (the prey) to use the energy and nutrients in the body of the prey for growth, maintenance or reproduction. Using the predator-prey idea, a model was proposed, adapting the predator-prey concept to Genetic Algorithms.

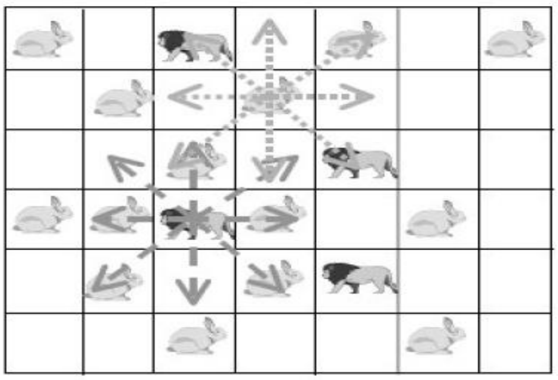


Figure 1: Predator - Prey Interaction

The Basic Algorithm

The basic Predator-Prey Approach proposed by Laumanns et al. is described below [3].

1. Initialize set of preys randomly between the variable limits.
2. Place these preys on the vertices of undirected connected graph.
3. Place predators randomly on the vertices of the graph.
4. Assign each predator with one objective function in a manner so that each objective is assigned to at least one predator.
5. Evaluate preys around each predator and select the worst prey. (For example, in a minimization problem, the worst prey will be the one which is having the largest value of the objective function which was assigned to that predator.)
6. The selected preys will be swallowed by the predators, meaning that the worst prey will be deleted and will be replaced by an offspring.
7. Create an offspring by mutating a randomly picked prey around the worst prey which was chosen by the predator.
8. Then predators will now take a random walk to the vertex which is a neighbor of the current position of the predator.
9. This completes one generation of the predator-prey algorithm. Repeat Steps 5 to 8 for the next generation.

With more generations, the prey population is hoped to reach near the true Pareto-optimal front. The break criterion is either a maximum number of generations or the fact that the objective functions to be optimised have an error below a certain threshold.

To implement the Predator-Prey Approach to constrained, multi-objective Function Optimisation problems and investigate the effects of introducing variations into the approach, together with the combination of predation with the classical GA operators of Crossover, Mutation, Dominance and Inversion.

This paper seeks to study the effects of introducing variations into the basic approach and then combining predation with the classical GA operators. Some of these variations include:

Effects of varying lattice sizes

1. Variations in fitness function
2. Variations in Prey Replacement
3. Effects of Elitist Strategies
4. Effects of Dominance
5. Using Recombination, Mutation and Inversion with Predation
6. Strategy for adequate distribution of solutions

The research is confined to investigating the basic Predator-Prey approach, its variations and combination with other GA operators. It does not seek to exploit the parallelism inherent in the Predator – Prey approach. It does not also consider other natural phenomena like migration or pack behaviour.

III EXPERIMENTAL WORK

Optimisation Function: Throughout the research, the constrained multi-objective optimisation problem whose Pareto-optimal solutions was to be determined was taken to be:

Minimise F(x) = [objective1(x); objective2(x)]

x

subject to $0 < x < 2$

where,

$$\text{objective1}(x) = (x + 2)^2 - 10$$

and

$$objective2(x) = (x - 2)^2 + 20$$

The problem has only one decision variable and two objectives to be minimised. The plot of the two objectives is below.

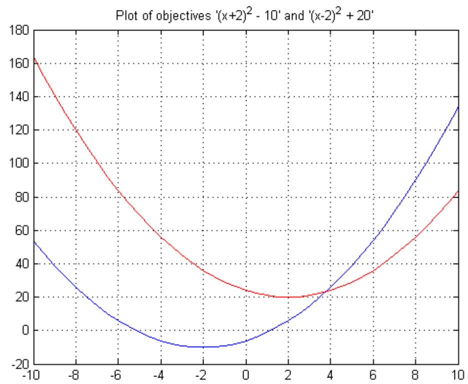


Figure 2: Plot of objectives $(x+2)^2$ - 10 and $(x-2)^2$ + 20

One can observe that the two objectives have their minima at x = -2 and x = 2 respectively. However in a multi-objective optimisation problem, any solution in the range -2 <= x <= 2 is equally optimal. The objective is to find a set of solutions in that range such that none of the objective functions can be improved without degrading some of the other objective values. Such a solution is said to be Pareto optimal.

The Predator-Prey Approach: A MATLAB Program implementing the basic Predator – Prey Approach was then written and successfully run. The approach has been described in Chapter 2 of this report. The results obtained from the basic Predator – Prey approach were recorded.

Some important details of the implementation are

Number of Prey	15
Number of Predators	2
Lattice Size	6x6
Probability of Mutation	0.1

Important Parameters for initial experiments of the Predator Prey Approach

Effects of Varying Lattice Sizes: While keeping the other parameters constant, the effects of varying lattice sizes on the approach was investigated. The program was run several times with lattice sizes of 6x6, 7x7, 8x8, 9x9, 10x10, 11x11 and 12x12.

Effects of Varying Fitness Functions: While the weakest prey was always determined using the fitness function corresponding to the Predator's objective function, a variation in the program was found by also calculating an aggregate fitness based on the formula

$$f = f1 + f2 - | f1 - f2 |$$

The aggregate fitness was used in subsequent variations for purposes such as determining the strongest prey corresponding to aggregate fitness around the predator.

Elitist Strategies: The use of elitist strategies was incorporated into the Predator-Prey approach. Elitism implies the use of the strongest prey (corresponding to some fitness) around the predator to produce prey for the next generation. Devoured prey is the weakest prey corresponding to predator objective function. For selection of Elite prey four variations were employed. The Lattice size was taken to be 8x8.

1. Variation A: Strongest prey corresponding to predator objective function without mutation
2. Variation B: Strongest prey corresponding to predator objective function with mutation
3. Variation C: Strongest prey corresponding to aggregate fitness function without mutation
4. Variation D: Strongest prey corresponding to aggregate fitness function with mutation.

Random Prey Replacement Strategy A random prey replacement strategy was then investigated. After the prey to be devoured was selected, a new prey was randomly generated in the specified limits and made to replace the devoured prey. This examined the assumption that a randomly generated prey may be more fit than the weakest prey.

Use of Dominance The use of Dominance in the Predator-Prey approach was investigated under three variations. The first of these has been taken from [3].

1. Variation A: If only an offspring is found to be better than the worst prey, the worst prey will be replaced. The evaluation will be based on the *domination* criteria. If the offspring *weakly* dominates (best in at least one objective function) all existing preys, thereby meaning that no prey in the existing population strongly dominates (best in all objective functions) the offspring, then that offspring is fit for that population. If the created offspring is fit, then the worst prey will be replaced by the offspring. When the offspring is not found to be fit, the worst prey will remain in the population and the predator will take a random walk.
2. Variation B: The approach A was modified to check the dominance of the prey using aggregate fitness instead of fitness corresponding to single objective functions
3. Variation C: The new prey whose dominance was to be examined was taken as a mutation of the best individual in the neighbourhood of the predator, instead of the former approaches that used a mutation of the weakest prey.

In this final modification, each prey is assumed to have an influencing region which is defined by a hyper-cube around it on the objective space. The offspring is not accepted if it is created within the influencing region of any existing prey. An influencing region of ± 0.01 was taken for the runs.

Predation with Crossover, Mutation, Inversion and Diversity Preservation A combined elitist strategy of Predation, Crossover, Mutation, Inversion and Diversity Preservation was then run for the test function. An algorithm for this combined strategy is presented below.

1. Initialize set of preys randomly between the variable limits.

2. Place these preys on the vertices of undirected connected graph.

3. Place predators randomly on the vertices of the graph.

4. Assign each predator its objective function.

5. Evaluate preys around each predator and select the worst prey.

6. Create two offspring by applying a crossover operation between the first and the second best preys in the neighborhood of the predator. Randomly choose one of the two offspring and mutate it to create the child solution.

7. Child acceptance criteria:

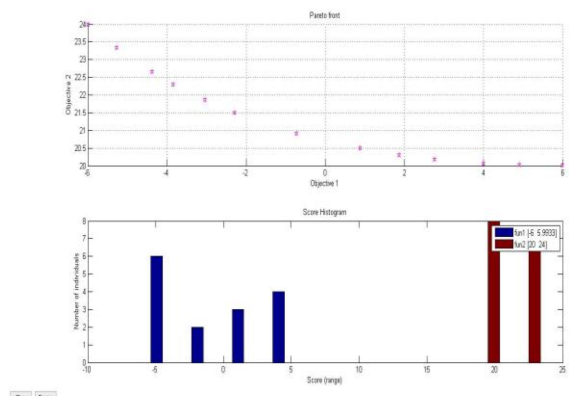
a. If the child solution is not within the influencing region of any existing prey, it replaces the worst prey. Predator also moves to the position of the worst prey.

b. Else the child is not accepted and a new child is created by Step 6. The creation of new child and its acceptance test are continued a maximum of 20 iterations, after which the inversion operator is applied to the most recent solution to give new prey.

8. This completes one generation of the predator-prey algorithm. Repeat Steps 5 to 7 for the next generation.

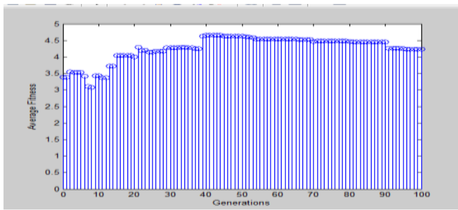
IV RESULTS

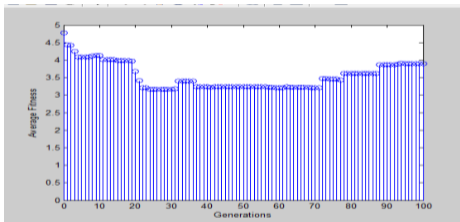
The Traditional GA Approach The solutions obtained from one run of the MATLAB Program using the traditional GA approach are given below.



The Predator Prey Approach

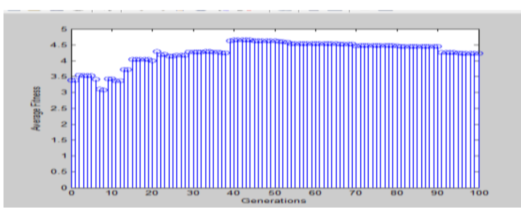
The graphs below show the average fitness across generations obtained from the basic Predator – Prey approach. While an improvement in average fitness was observed in many cases, there was also complete degradation observed in some runs. An example of such degradation is shown in Case 2 below.



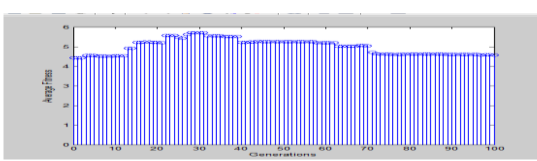


Effects of Varying Lattice Sizes

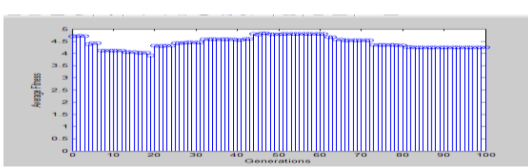
When the lattice sizes were varied, it was observed that lattice size impacted the performance of the approach. Smaller lattice sizes showed faster changes in the average fitness between generations. Larger lattice sizes showed slower changes in the average fitness between generations. There was also a greater possibility of degradation with smaller lattice sizes.



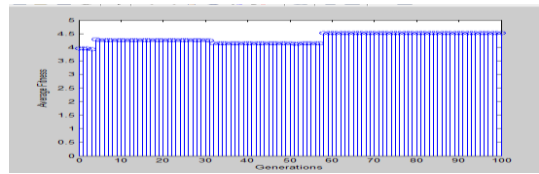
6x6 Lattice Size



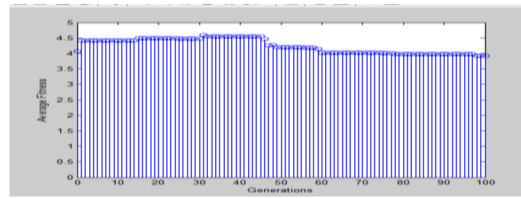
7x7 Lattice Size



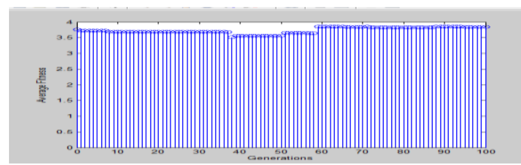
8x8 Lattice Size



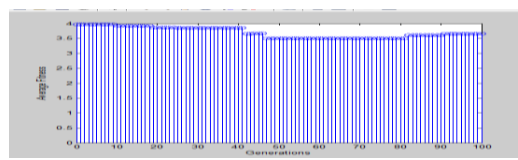
9x9 Lattice Size



10x10 Lattice Size



11x11 Lattice Size



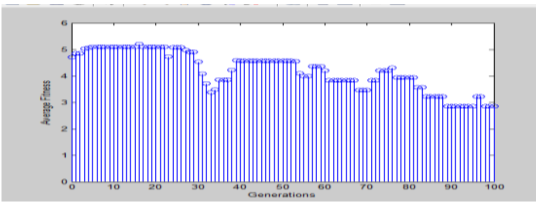
12x12 Lattice Size

Elitist Replacement Strategies

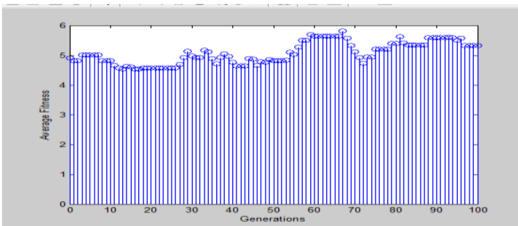
The observations made for the Elitist Replacement Strategies are tabulated below.

1. Variation A: Both positive and negative results observed. In the long run, population diversity is greatly affected
2. Variation B: Both positive and negative results observed. Population Diversity is preserved preserved by use of mutation.
3. Variation C: Population diversity is greatly affected. In most cases, there is improvement in the aggregate fitness.
4. Variation D: In most cases, there is improvement in the aggregate fitness. Population Diversity is preserved by use of mutation.

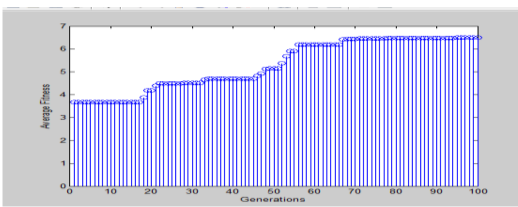
Sample outputs for these strategies are shown ahead.



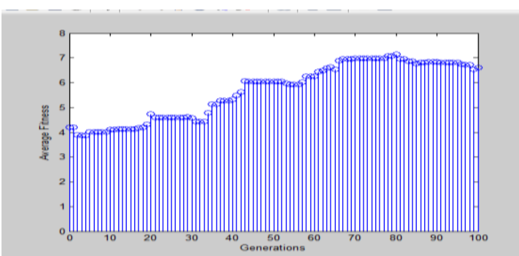
Elitist Replacement by Predator Fitness without Mutation



Elitist Replacement by Predator Fitness with Mutation



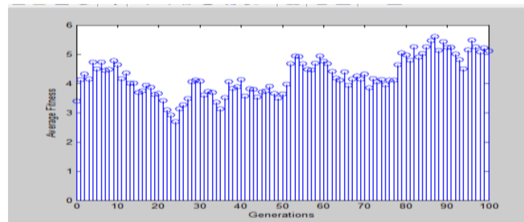
Elitist Replacement by Aggregate Fitness without Mutation

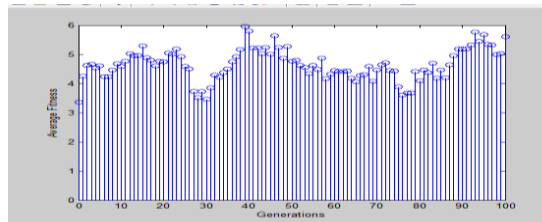


Elitist Replacement by Aggregate Fitness with Mutation

Random Prey Replacement Strategy

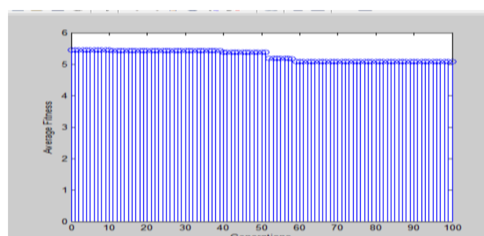
Such a technique gave no certainty of improvement. The average fitness graphs are shown below. The oscillations in average fitness are evident.



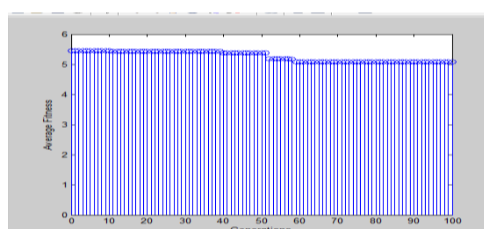


Use of Dominance

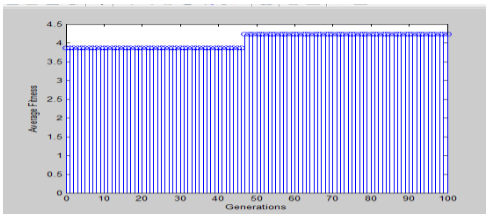
The effects of the use of dominance under the three strategies described in the previous section are tabulated below.
Variation A: Negative results in most cases.
Variation B: Extremely slow convergence to a slightly improved solution set from the initial set. In many cases, the improvement recorded was poor.
Variation C: Gradual improvements in the average fitness of population, though the rate of optimisation was still slow.
Sample outputs for these strategies are shown ahead



Variation A



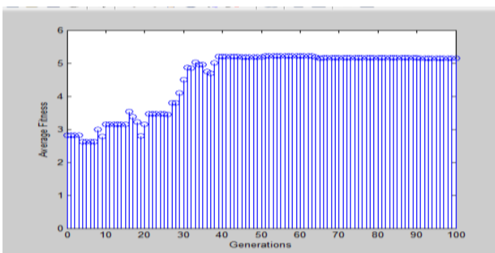
Variation B

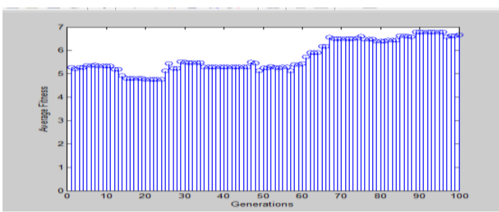


Variation C

Use of Recombination and Mutation

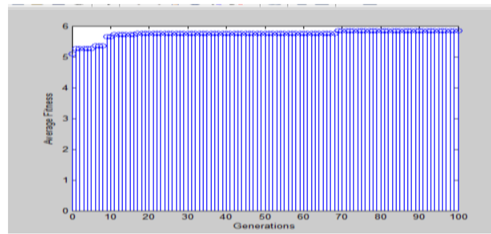
The use of Crossover and Mutation gave positive results but the diversity of population appeared to be adversely affected. Two sample average fitness graphs are shown below.

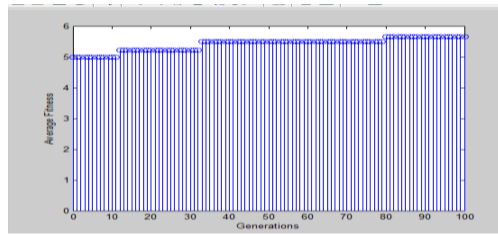




Diversity Preservation

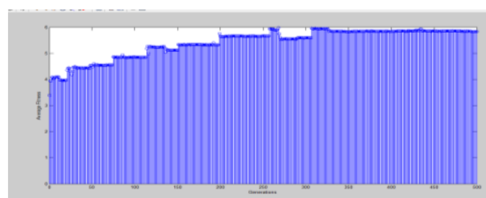
With the use of Diversity Preservation, a positive influence on the adequate distribution of prey was observed. However, the mechanism affects the speed or rate of convergence of the approach negatively. The two graphs illustrate this.

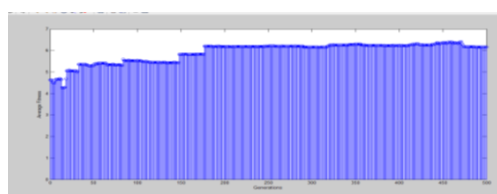




Predation with Crossover, Mutation, Inversion and Diversity Preservation

The combination of Predation with the GA operators of Crossover, Mutation and Inversion together with the mechanism for ensuring adequate distribution of prey showed positive results. Although inter-generational average fitness decreased in the interim, there was always an overall improvement observed in average fitness of the population. However, the rate of convergence was slower because of the diversity preservation mechanism.





V. CONCLUSION

A systematic investigation of the Predator-Prey approach to solving constrained, multi-objective optimisation problems has been carried out successfully. The project has reviewed the variations to the basic Predator-Prey approach and the use of Predation with the classical GA operators has been implemented. The Predator-Prey approach has been found to successfully converge to the Pareto optimal front for the constrained, multi-objective optimisation problems used and some of the variations in the basic approach provided faster and more reliable convergence. The use of predation can indeed be considered an enhancement to Genetic Algorithms.

V. REFERENCES

[1] D. E. Goldberg, *Genetic Algorithms in Search, Optimization and Machine Learning*. Pearson

[2] M. G. J. D. Pablo Seonane, "Approach for solving multimodal problems using Genetic Algorithms with Grouped into Species optimized with Predator-Prey," *Special Issue on Distributed Computing and Artificial Intelligence*.

[3] K. Deb, U.Bhaskara Rao, *Investigating predator-Prey Algorithms for Multi-Objective Optimization*. 2005

[4] A. Ghosh and S. Dehuri, "Evolutionary Algorithms for Multi-Criterion Optimisation: A Survey," *International Journal of Computing & Information Sciences*, 2004.

[5] X. Li, "A real-coded predator-prey genetic algorithm for multiobjective optimisation".

[7] D. E. Goldberg, “Genetic Algorithms in Search, Optimization and Machine Learning” in Addison-Wesley, Reading, 1989.

[8] M. Gestal Pose, *Computación evolutiva para el proceso de selección devariables en espacios de búsqueda multimodales*. PhD Thesis, 2010.

[9] H.Blom, C.Küch, K.Losemann, *PEPPA: a project for evolutionary predator prey algorithms*. GECCO '09. 2009.

[10] D.Kalyanmoy, U.Bhaskara, *Investigating predator-Prey Algorithms for Multi-Objective Optimization*. Department of Mechanical Engineering Indian Institute of Technology Kanpur. 2005.

[11] K. Deb, U.Bhaskara Rao, *Investigating predator-Prey Algorithms forMulti-Objective Optimization*. 2005

[12] Xingyi Zhang; Ye Tian; Yaochu Jin "A Knee Point-Driven Evolutionary Algorithm for Many-Objective Optimization", *Evolutionary Computation, IEEE Transactions on,* On page(s): 761 - 776 Volume: 19, Issue: 6, Dec. 2015

[13] D. Simon, "A probabilistic analysis of a simplified biogeography-based optimization algorithm", *Evol. Comput.*, vol. 19, no. 2, pp. 167-188, 2011