EVOLUTIONARY METHODS FOR DESIGN, OPTIMIZATION AND CONTROL
T. Burczynski and J. Périaux (Eds.)

MODIFIED PREDATOR-PREY (MPP) ALGORITHM FOR CONSTRAINED MULTI-OBJECTIVE OPTIMIZATION

Souma Chowdhury, George S. Dulikravich and Ramon J. Moral
Department of Mechanical and Materials Engineering, MAIDROC Lab.,
Florida International University, 10555 West Flagler St., EC 3474, Miami, Florida 33174, U.S.A.
Email: schow003@fiu.edu; dulikrav@fiu.edu
web page:http://maidroc.fiu.edu

Abstract. In this work, an evolutionary multi-objective optimization algorithm based on the dynamics of predator-prey interactions existing in nature is presented. This algorithm is comprised of a relatively small number of predators and a much larger number of prey, randomly placed on a two dimensional lattice with connected ends. The predators are partially or completely biased towards one or more objectives, based on which each predator kills the weakest prey in its neighborhood. A stronger prey created through evolution replaces this prey. The prey remain stationary, while the predators move around in the lattice during the course of generations. Modifications of the basic predator-prey algorithm have been implemented in this study regarding the selection procedure, apparent movement of the predators, mutation strategy, dynamics of the Pareto convergence and a capability of handling equality and inequality constraints. The final modified algorithm was tested on standard constrained and unconstrained multi-objective optimization problems.

Key words: constrained optimization, evolutionary algorithms, multi-objective optimization, predator-prey algorithm.

1 INTRODUCTION

In 1998, Hans Paul Schwefel and co-workers[1] proposed a new optimization algorithm to search for Pareto-optimal solutions from a randomly generated initial population of candidate solutions. This algorithm imitates the natural phenomena that a predator kills the weakest prey in its neighborhood, and the next generations of prey that evolve are relatively stronger and more immune to such predator attacks. However, this initial Predator-Prey (PP) optimization algorithm seemed to have difficulty in producing well distributed non-dominated solutions along the Pareto front. Deb[2] suggested certain new features, namely, the 'elite preservation operator", the 'recombination operator' and the 'diversity preservation operator'. Further, Li[3] introduced the movement of both predators and prey and changing population sizes of prey. Some other versions of the algorithm have been presented by Grimme *et al.*[4] and Silva *et al.*[5]. The former used a modified recombination and mutation model. The latter, predominantly a particle swarm optimization algorithm, introduces the concept of predator-prey interactions in the swarm to control the balance between exploration and exploitation, hence improving both diversity and rate of convergence.

However, most of the available versions of PP find it difficult to produce a well distributed set of Pareto optimal solutions in a limited number of function evaluations. This necessitates optimization algorithms capable of producing solutions that are both optimum as well as feasible with respect to the system constraints. These system constraints can be modeled as mathematical constraint functions. Different constraint handling techniques have been developed, some of the most popular ones being penalty function methods[6,7], filtering methods and constraint domination method introduced by Deb *et al.*[8].

The Modified Predator-Prey (MPP) algorithm presented here is an evolutionary multi-objective optimization algorithm, capable of handling complex design optimization problems. It has been developed through the assimilation of special features of existing PP models, modifications of the same and addition of certain new features; the most significant one being the ability to handle both linear/non-linear equality and inequality constraints. The constraint handling method used in MPP is a combination of filtering technique and constraint dominance technique.

2 MODIFIED PREDATOR-PREY (MPP) ALGORITHM

The fundamental steps followed by MPP are outlined in the initial version developed by Chowdhury *et al.*[9]. The significant features of MPP that distinguishes it from previous versions of predator-prey algorithms are as follows:

2.1 Evolution

The generation of new solutions in each active locality (locality containing a predator) is initiated by the crossover of the strongest two local prey. The blend crossover (BLX-α), initially proposed by Eshelman and Schaffer[6] for real-coded genetic algorithms (later improved by Deb[2]), is used in this algorithm. BLX-α is defined by the following equations.

$$x_i^{(1,t+1)} = \left(1-\gamma_i\right) x_i^{(1,t)} + \gamma_i x_i^{(2,t)} \qquad (1)$$
$$\gamma_i = \left(1+2\alpha\right) u_i - \alpha$$

Here, $x_i^{(1,t)}$ and $x_i^{(2,t)}$ are the parent solutions, $x_i^{(1,t+1)}$ is the child solution and u_i is the random number between 0.0 and 1.0. A value of 0.5 is used for α as suggested by Deb[2]. This crossover child prey is then subjected to non-uniform mutation originally introduced by Michalewicz[10] and modified by Chowdhury *et al.*[9] to allow for dynamic adjustment of the extent of mutation as shown below.

$$\beta = 10^{-\left(t/t_{\max} \right)} \qquad (2)$$
$$y_i^{(1,t+1)} = x_i^{(1,t+1)} + \tau\beta\left(x_i^{(U)} - x_i^{(L)}\right)\left(1 - r_i^{\left(1-t/t_{\max}\right)^b}\right)$$

where $y_i^{(1,t+1)}$ is the child solution produced from the parent solution $x_i^{(1,t+1)}$, by mutation of the i[th] variable, $x_i^{(U)}$ and $x_i^{(L)}$ are upper and lower limits of the i[th] variable, τ takes a Boolean value -1 or 1, each with a probability of 0.5, r_i is a random number between 0 and 1, t and $t_{\max}$ are the number of function evaluations performed until then and maximum allowed number of function evaluations, respectively, (β = 1.5 determined empirically) and β is the scaling parameter. The latter two factors monitor the order of magnitude, or in other words, the extent of mutation.

Both the crossover and mutation techniques employed here establish an adaptive search, which makes the MPP algorithm more economical with respect to function evaluations.

2.2 Dominance and Constraint Handling

The constraint dominance is used as one of the qualifying criterion for new prey when compared with the surviving local prey. This method has been derived from the constraint dominance used in NSGA-II[8], which is as follows,

Solution i is said to constraint-dominate solution j if

- Solution i is feasible and solution j is not or
- Solutions i and j are both infeasible, while solution i has a smaller net constraint violation than solution j, *i.e.* ${f_{Nf+1}}^i < {f_{Nf+1}}^j$ (considering function minimization) or
- Solutions i and j are both feasible, while solution i weakly dominates solution j.

It is evident that this criterion gives feasibility preference over optimality.

2.3 Diversity Preservation

The concept of objective space hypercube is used as a qualifying criterion for new prey to assure diversity preservation. Each old local prey is considered to be at the centre of its hypercube, the size of which is dynamically updated with generations. This proves to a very promising approach to maintain diversity without posing any inhibition to Pareto convergence. In addition to this, an innovative concept of sectional convergence has been introduced to deal with this possible lack of effective variation in the prey population. Instead of the running the algorithm throughout for the same initial specified distribution of weights, there is redistribution of weights within a small biased range (<1.0) after a certain number of function evaluations. This directs the intermediate solutions to different regions of the Pareto front depending on the distribution of weights. This is essentially different from the technique used by gradient based algorithms when solving multi-objective problems since the weighted Pareto approach is initiated after partial convergence. However, the concept of sectional convergence needs to be used only in case of problems with discontinuous domains or high dimensionality.

2.4 Elitism

In MPP a secondary set is maintained, which contains the non dominated solutions from each generation. This set is known as the elite set, the size of which is preserved using the clustering technique designed by Deb[2]. After each generation, a certain number of randomly selected solutions/preys (from the main population), if found to be dominated, are replaced from the 2D lattice by randomly selected elite solutions. This new attribute boosts the speed of convergence of this algorithm. The number of such replacements should be carefully allocated. The MPP results shown in this paper have been accomplished by allowing replacement of 20% of the main population after each generation.

2.5 Constraint handling technique

Optimization algorithms when dealing with constrained multi-objective problems need to ensure that the searching mechanism employed find solutions that are both optimal and feasible. The feasible domain represents the region in the search space where every solution satisfies the problem constraints. In MPP, feasibility is accomplished and sustained by a combination of the filtering technique and the constraint dominance criterion[7]. For a general multi-objective optimization problem

$$\begin{aligned}&\text{Minimize} \quad f_i = f_i(X), \quad i = 1,2,...,Nf \\ &\text{subject to} \\ &g_i \le 0, \quad i = 1,2,3,...,p \\ &h_i = 0, \quad i = p+1, p+2,..., p+q \\ &p, q \in N\end{aligned} \tag{3}$$

where $X = (x_1, x_2, x_3, ..., x_m) \quad ,x_i \in R$.

The constraints are added up to form the $(Nf+1)^{\text{th}}$ objective

$$\text{Minimize} \quad f_{Nf+1} = \sum_{i=1}^{p} \max(g_i, 0) + \sum_{i=p+1}^{p+q} \max((h_i - \varepsilon), 0) \tag{4}$$

where ε is the tolerance for equality objectives. This additional constraint objective is treated like any other objective by the usual predator prey approach. Hence the selection pressure in the active localities is neither biased towards feasibility nor optimality. On the other hand the child prey produced has to be at least non-dominated, with respect to the other local prey, to be accepted. This brings in the constraint dominance mechanism which is biased towards achieving feasibility. Both these factors working together provide a balance between selections of prey (solutions) based on the actual objective values as well as their distances from the feasible domain.

3 RESULTS AND DISCUSSION

MPP was implemented using C++ programming language. To examine the constraint handling capability of MPP, it was tested with a well known constrained 2-objective test case TNK studied by Deb *et al.*[8]. Two standard test cases with known analytical solutions, Binh[11] multi-objective optimization problem no. 2 and the Osyczka[12] multi-objective optimization problem no. 2, have also been used. The final Pareto fronts in case of these constrained test cases were constructed of only those elite set solutions that do not violate any of the problem constraints. The user defined parameters in the algorithm pertinent to the constrained test cases are presented in Table 1.

Parameter	TNK values	Binh and Osyczka values
Population size (number of prey)	100	100
Number of predators	10	10
Elite strength	40	100
Crossover probability	1.0	1.0
Mutation probability	0.05	0.05
Number of primary iterations (sections)	0, 3	0, 6

Table 1: General MPP parameters for TNK, Binh and Ozyczka two-objectives test cases

A higher number of primary iterations and greater elite set strength were used in case of the Binh and the Osyczka problems as seen from Table 1. This is to counteract the relatively greater difficulty in covering the whole Pareto front in these two test problems. The converged Pareto fronts computed by MPP in each of these test cases are shown in Figures 1 to 6. The final Pareto fronts computed for TNK and Binh constrained multi-objective problems (Figures 1-4) are fairly accurate and well distributed. In the Binh problem, there is a significant improvement in performance when using the sectional convergence scheme (Figures 3 and 4). In case of Osyczka constrained multi-objective problems (Figures 5 and 6), though solutions converge to the global Pareto front, their distribution on the final Pareto front is not uniform, even with the sectional convergence scheme.

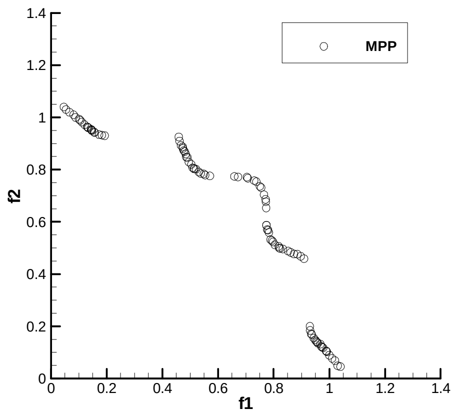

Figure 1: Constrained 2-objective test case TNK without sectional convergence

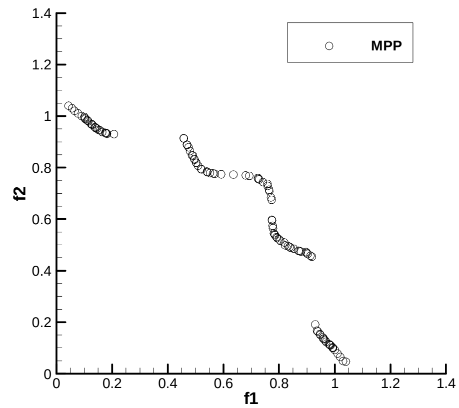

Figure 2: Constrained 2-objective test case TNK with sectional convergence

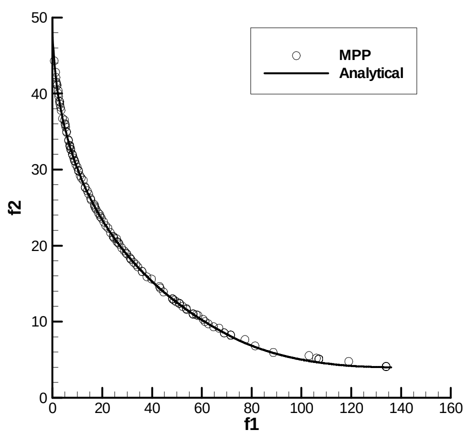

Figure 3: Constrained 2-objective test case Binh without sectional convergence

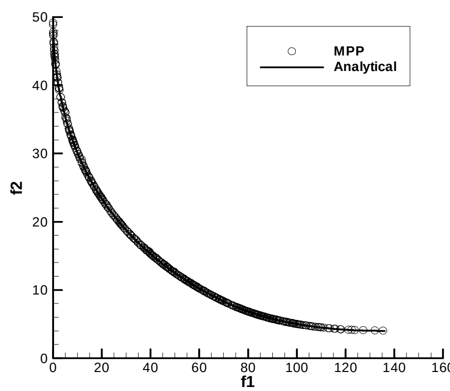

Figure 4: Constrained 2-objective test case Binh with sectional convergence

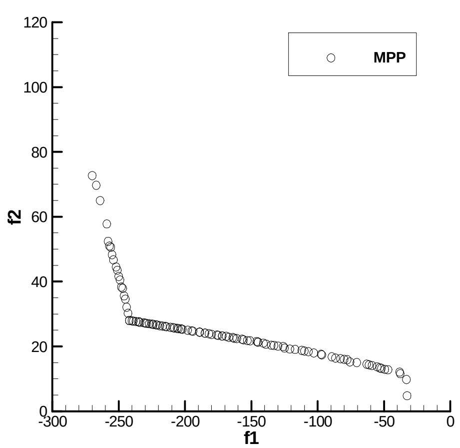

Figure 5: Constrained 2-objective test case Osyczka without sectional convergence

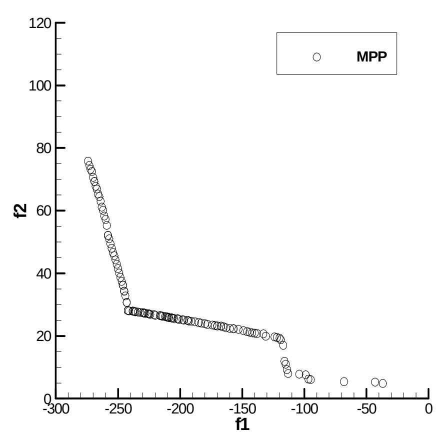

Figure 6: Constrained 2-objective test case Osyczka with sectional convergence

Overall, MPP compares well in performance, with other popular algorithms such as NSGA II[8] in solving similar constrained multi-objective problems. Nevertheless, appropriate implementation of the sectional convergence scheme is necessary for certain problems, in order to attain a reasonable spread of solutions along the final Pareto front.

4 SUMMARY

The modified predator-prey algorithm follows the basic concept of the predator-prey evolutionary strategy and combines new features, some originally developed and others derived from NSGA II algorithm. Nevertheless, MPP is quite different from the latter due to the selection mechanism, localized improvement of solutions, adaptive hypercube sizing and adaptive extent of mutation. The most promising attribute of MPP is its ability to consistently produce feasible Pareto solutions, irrespective of the number or nature (linear or non-linear) of problem constraints involved. This is accomplished without normalization of objective functions or constraint functions, or application of computationally costly penalty function methods.

5 REFERENCES

[1] M. Laumanns, G. Rudolph and H.P. Schwefel, "A spatial predator-prey approach to multi-objective optimization: A preliminary study", in *Proceedings of the Parallel Problem Solving from Nature*, V, 241-249 (1998).

[2] K. Deb, *Multi-objective Optimization Using Evolutionary Algorithms*, Chichester, UK, Wiley (2002).

[3] X. Li, "A real-coded predator-prey genetic algorithm for multi-objective optimization", *Lecture Notes in Computer Science*, Vol. 2632, pp. 69 (2003).

[4] C. Grimme and K. Schmitt, "Inside a predator-prey model for multiobjective otpimization – A second study", In *Proc. Genetic and Evolutionary Computation Conf.(GECCO 2006), Seattle WA*, ACM Press, New York, 707-714 (2006).

[5] A. Silva, E. Neves and E. Costa, "An empirical comparison of particle swarm and predator-prey optimization", *Lecture Notes in Computer Science*; 2464: 103-110.

[6] L.J. Eshelman and J.D. Schaffer, "Real coded genetic algorithms and interval schemata", in *Foundations of Genetic Algorithms 2 (FOGA 2)*, 187-202 (1993).

[7] C. Coello Coello, "Theoretical and numerical constraint-handling techniques used with evolutionary algorithms: A survey of the state of the art", *Computer Methods in Applied Mechanics and Engineering* 191(11–12), 1245–1287 (2002).

[8] K. Deb, A. Pratap, S. Agarwal and T. Meyarivan, "A fast and elitist multiobjective genetic algorithm: NSGA-II", *IEEE Transactions on Evolutionary Computation*, Vol. 6, No. 2, 182-197 (2002).

[9] S. Chowdhury, R.J. Moral and G.S. Dulikravich, "Predator-prey evolutionary multi-objective optimization algorithm: performance and improvements", *7th ASMO-UK/ISSMO International Conference on Eng. Design Optimization*, Bath, UK, July 7-8 (2008).

[10] J. Michalewicz, *Genetic Algorithms + Data Structures = Evolutionary Programs*, Berlin, Springer-Verlag (1992).

[11] T.T. Binh and U. Korn, "MOBES: A multiobjective evolution strategy for constrained optimization problems", *In The Third International Conference on Genetic Algorithms (Mendel 97)*, Brno, Czech Republic, 176-182 (1997).

[12] A. Osyczka and S. Kundu "A new method to solve generalized multicriteria optimization problems using the simple genetic algorithm", *Structural Optimization*, 10:94-99 (1995).