Modified predator-prey algorithm for constrained and unconstrained multi-objective optimisation

Souma Chowdhury and
George S. Dulikravich*

Multidisciplinary Analysis, Inverse Design,
Robust Optimization and Control (MAIDROC) Laboratory,
Department of Mechanical and Materials Engineering,
Florida International University,
10555 W. Flagler Str. Miami, Florida 33174, USA
E-mail: schow003@fiu.edu E-mail: dulikrav@fiu.edu
*Corresponding author

Ramon J. Moral

US Army Engineering Research and Development Centre,
CEERD-GM-I, 3909 Halls Ferry Road,
Vicksburg, MS 39180, USA
E-mail: ramon.moral@usace.army.mil

Abstract: Modifications of a well-known predator-prey evolutionary multi-objective optimisation 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 neighbourhood. A stronger prey created through evolution replaces this prey. In case of constrained problems, the sum of constraint violations serves as an additional objective. The prey remains stationary, while the predators move around in the lattice. Modifications have been implemented in this study regarding the selection procedure, apparent movement of the predators, mutation strategy, and dynamics of convergence to the Pareto front. Further alterations have been made making the algorithm capable of handling equality and inequality constraints. The final modified algorithm is tested on standard constrained and unconstrained multi-objective optimisation problems.

Keywords: constrained optimisation; evolutionary algorithms; multi-objective optimisation; predator-prey algorithm.

Reference to this paper should be made as follows: Chowdhury, S., Dulikravich, G.S. and Moral, R.J. (2009) 'Modified predator-prey algorithm for constrained and unconstrained multi-objective optimisation', *Int. J. Mathematical Modelling and Numerical Optimisation*, Vol. 1, Nos. 1/2, pp.1–38.

Biographical notes: Souma Chowdhury received his BTech in Mechanical Engineering from Indian Institute of Technology, Kharagpur in 2007, where he was a member of the CFD laboratory. He received his MS in Mechanical Engineering from Florida International University, where he also worked as a

Graduate Research Assistant in the Multidisciplinary Analysis, Inverse Design, Robust Optimization and Control (MAIDROC) Lab. His research interests include development and application of design optimisation techniques, large-scale numerical analysis, especially computational fluid dynamics and systems modelling. Currently, he is a PhD student at Rensselaer Polytechnic Institute.

George Dulikravich received his PhD in Aerospace Engineering from Cornell University in 1979, his MSc from University of Minnesota in 1975 and Dipl-Ing. from University of Belgrade in 1973. He worked as an NRC Associate and USRA Visiting Scientist at NASA LeRC, a Visiting Scientist at DFVLR-Goettingen, Assistant Professor at University of Texas-Austin, Associate Professor at Penn State University, and Professor at University of Texas at Arlington and at FIU. He co-authored over 390 publications and is the founder and Editor-in-Chief of *Inverse Problems in Science and Engineering* (founded in 1994). He is a fellow of the ASME, AAM and RAeS and an Associate Fellow of AIAA.

Ramon J. Moral received his BSc in 1992 from FIU, his MS from Auburn University in 1997, and PhD in 2008 from FIU, all in Mechanical Engineering. In the course of his PhD studies, he developed Multi-Objective Hybrid Optimisation software (MOHO) and a self-assembling response surface generation technique. He also performed shape optimisation of high-speed projectiles for deep penetration. He has worked as an Electronic Packaging Engineer, Manufacturing Engineer, and R&D Engineer during his six years of industrial experience at Raltron Electronics Corp., and Vectron International on microelectronics reliability, process development, CAE in packaging, and thermo-elastic FEA for product development and failure analysis.

1 Introduction

The last few decades have seen the development of optimisation algorithms inspired by the principles of natural evolution. These algorithms, often termed as evolutionary multi-objective optimisation algorithms (EMOA), utilise a set of multiple candidate solutions (population space) to follow an iterative procedure producing a final set of the best compromise solutions, the graphical representation of which is termed Pareto front (Deb, 2002). In case of single objective problems, the Pareto front reduces to a single optimal solution known as the global minimum or global maximum. Genetic algorithm, differential evolution, particle swarm and predator-prey algorithms are some of the most prominent EMOAs.

Most real world systems that demand optimised design are often subject to configurational and operational restrictions which should be taken into consideration during the process of optimisation. This necessitates optimisation algorithms capable of producing solutions that are both optimal as well as feasible with respect to the system constraints. These system constraints can be modelled as mathematical constraint functions.

In 1998, Hans Paul Schwefel proposed a new optimisation algorithm (Laummans et al., 1998) 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 neighbourhood. This prey is then replaced by a new prey that is relatively stronger and more immune to such predator attacks. However, this initial PP optimisation algorithm seemed to have difficulty in producing well distributed non-dominated solutions along the Pareto front. Since then, several modifications of the above algorithm have appeared in literature. Deb (2002) suggested an improved version of the algorithm which included certain new features, namely, the 'elite preservation operator', the 'recombination operator' and the 'diversity preservation operator'. A further modified version of the algorithm was proposed by Li (2003), where a dynamic spatial structure of the predator-prey population was used. It involved the movement of both predators and prey and changing population strength of prey. Some other versions of the algorithm have been presented by Grimme and Schmitt (2006) and Silva et al. (2002). The former used a modified recombination and mutation model. The latter, predominantly a particle swarm optimisation 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 above versions as well as some of the other popular evolutionary algorithms find it difficult to produce well distributed set of Pareto optimal solutions in a limited number of function evaluations especially when dealing with problems with more than two objectives or significantly high number of decision (design) variables. The stated can be observed from the reported results of other algorithms (results of Deb et al., 2006 and that of Deb and Rao, 2005). In most practical applications of optimisation, the calculation time for evaluating model functions dominate. This demands optimisation algorithms capable of producing dependable solutions while investing the minimum number of function evaluations possible. Moreover, the forms of the PP algorithm available in literature (as seen in Deb, 2002; Laumanns et al., 1998; Li, 2003; Grimme and Schmitt, 2006) do not have the ability to handle constraints, which form an integral part of most practical problems. Consequently, there exist very few instances of application of any form of the PP algorithm to real world problems. Nevertheless, since the *modus operandi* of the PP algorithm is significantly different from other standard EMOAs, there is sufficient basis to believe that the potentials of this algorithm have not been fully realised.

The modified predator-prey (MPP) algorithm presented here is a computationally inexpensive EMOA, capable of handling complex design optimisation 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.

2 Modified predator-prey (MPP) algorithm

2.1 Overview

Any general constrained multi-objective problem involving N_f objectives and N_V design variables can be reformulated as follows.

$$\text{minimise} \quad f_n = f_n(X), \quad \text{n} = 1, 2, ..., N_f \tag{1a}$$

where X is the vector of N_V design variables, that is,

$$X = (x_1, x_2, x_v, ..., x_{N_V}) \tag{1b}$$

and g_{ic} and h_{ic} are inequality and equality constraints, respectively, defined as

$$\begin{aligned} g_{ic} &= g_{ic}(X) \le 0, \quad ic = 1, 2, 3, ..., p \\ h_{ic} &= h_{ic}(X) = 0, \quad ic = p+1, p+2, ..., p+q \end{aligned} \tag{1c}$$

The constraints are added up to form the $(N_f + 1)^{\text{th}}$ objective in the following way,

$$\text{minimise } f_{N_f+1} = \sum_{ic=1}^{p} \max\left(g_{ic}, 0\right) + \sum_{ic=p+1}^{p+q} \max\left(\left(h_{ic} - \varepsilon\right), 0\right) \tag{2}$$

where ε is the tolerance for equality objectives. Clearly, zero is the true minimum of this combined function. It should be noted that in case of maximisation the corresponding objective function is multiplied by '–1', to convert it into a general minimisation problem. Also, a 'greater than equal to' inequality constraint is converted into a 'less than equal to' constraint by multiplying with '–1'.

The overall structure of the algorithm is presented below in sequential steps.

- Values of N_V design variables are initialised for a population of N_S candidate solutions/prey using Sobol's (1976) quasi random sequence generator and the corresponding N_f objectives are evaluated for each such candidate prey.

- The N_S prey are placed at the vertices of cells forming a two dimensional grid with connected ends hence having a toroidal nature. The grid is allowed to adjust its size dynamically according to the prey population size (in case of changing population size when working as a component of a hybrid optimisation package) maintaining the dimensions $I \times J$, where typically $J = 5$. This value of J was found to be most effective when dealing with a population of hundred or less. However, for larger populations other values of J can be tried which might even be problem-dependent for real world applications. Consequently I is chosen as the smallest integer for which $N_S < I \times J$. The few empty spaces that might occur (four in the worst case) are filled up with random clones of existing population members. Alternatively, these empty spaces can be filled with additional solutions newly generated in order to preserve diversity in the population. However, in this case the corresponding additional function evaluations need to be made.

- M_P is the number of predators placed on the same 2D unfolded toroidal grid so that they occupy random cell centres. M_P is determined as

  $$M_P = N_f \left[\frac{N_S}{20} \right) \tag{3}$$

  where $\left[r \right)$ is the lowest integer greater than r, $r \in R^+$. It was observed that for the standard test cases discussed in this paper a higher value of M_P led to premature convergence (stalling of solutions) due to excessive selection pressure (compared to evolutionary pressure), whereas a lower value of M_P demanded higher number of function evaluations without any improvement in objective function values. Thus, the optimum predator strength is dependent on the function topology.

Each predator is associated with a weighted value of the objectives as follows.

$$f = \sum_{n=1}^{Nf} w_n f_n \tag{4}$$

Here, w_n is the weight associated with the nth objective function, f_n is the nth objective function. The weights are distributed uniformly in case of two-objective problems $(0 \le w_1 \le 1,\ w_2 = 1 - w_1)$ and using Sobol's (1976) quasi random sequence generator in case of problems with more than two objectives. Objective space has not been normalised to apply the 'weighted sum of objectives' concept, and it did not pose any significant hindrance to convergence as seen from the performance of MPP in the test cases exhibited in this paper. However, in case of real problems with objective functions widely different in magnitude, each objective should be normalised with a reference objective value, or a dynamic maximum calculated from the existing population.

- Predators are randomly located in the toroidal grid. Each neighbourhood that contains a predator can be termed as an 'active locality' as shown in Figure 1. In each of these localities/cells, the value of '*f*' as defined by equation (4) corresponding to the local predator, is calculated for each prey. The weakest prey (that is*,* having the maximum value of *f*) is selected to be killed and replaced by a new prey produced by the crossover of the two strongest local prey and subsequent mutation of the crossover child.
- However, this phylogenetic child prey qualifies to be accepted only if it fulfils the following three criteria:
  1. the child is stronger than the worst local prey [based on f calculated by equation (2)]
  2. the child is non-dominated (Deb, 2002) with respect to the other three local prey
  3. the child is not within the objective space hypercube (Deb, 2002) of the other three prey of this locality.

  Ten trials are allowed to produce a qualified child that satisfies the three criteria, failing which the weakest prey is kept alive and retains its position on the grid. This is affordable, since MPP, unlike other evolutionary algorithms does not require the function evaluation of the entire population at each generation, and prey reproduction take place only in the cells that contain a predator. Nevertheless, the expense for the above qualification process increases with increasing number of simultaneous objectives.
- Upon completion of the above predator-prey interactions in each active locality, the predators are relocated randomly. A probability based relocation criterion has been introduced here, which favours a fairly even distribution of the 'number of visitations' to each cell/locality by a predator. The relocation criterion is defined as follows:

1 if $cellcount(i,j) > cellcount_{avg} + 1,$ (5.1)

the predator is not placed at this cell centre

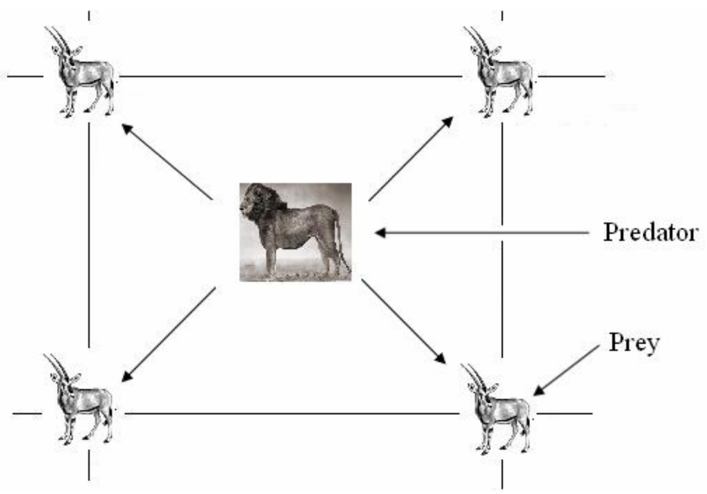
2 if $cellcount(i,j) \leq cellcount_{avg} + 1,$ (5.2)

the predator is placed at this cell centre.

Here, $cellcount(i,j)$ is the cumulative number of times predators have visited the cell (i,j) in previous generations, $cellcount_{avg}$ is the average of all $cellcount(i,j)$ and (i,j) is the randomly generated location on the 2D lattice. Laumanns et al. (1998) showed that in the original PP algorithm the torus structure leads to an expected recurrence time (visitation per each four-prey locality) per predator of $1/n$ (where $n = I \times J$) and the number of random steps required by a predator to have visited each prey is between $n \log^2 n$ and $2(n-1)^2$. Introduction of the controlled relocation in MPP improves the expected recurrence time per predator at no additional cost (except for some additional random number generation). This increases the coherence in improvement of the prey regardless of their fixed location in the toroidal grid.

- The execution of predator-prey interaction in each active locality and subsequent prey reproduction constitute a generation in the MPP algorithm. After completion of each such local generation, the non-dominated solutions in the prey population are copied to a secondary set called the 'elite set' or an 'archive'. Certain number of randomly selected elite solutions is incorporated into the main population (prey in the toroidal space) at the cost of some of the dominated solutions (dominated by at least one other prey).

Figure 1 An active four-prey locality/neighbourhood in the toroidal grid (see online version for colours)



2.2 Evolution

The generation of new solutions/prey in each active locality is initiated by the crossover of the strongest two local prey (with respect to the corresponding f value). The blend crossover (BLX-α), initially proposed by Eshelman and Schaffer (1993) for real-coded genetic algorithms (later improved by Deb, 2002), is used in the MPP. It is defined as follows.

$$\begin{aligned} x_v^{(1,t+1)} &= (1-\gamma_v)x_v^{(1,t)} + \gamma_v x_v^{(2,t)} \\ x_v^{(2,t+1)} &= \gamma_v x_v^{(1,t)} + (1-\gamma_v)x_v^{(2,t)} \\ \gamma_v &= (1+2\alpha)u_v - \alpha \end{aligned} \tag{6}$$

Here, $x_v^{(1,t)}$ and $x_v^{(2,t)}$ are the values of the vth design variable defining parent solutions/prey, $x_v^{(1,t+1)}$ is value of the vth variable defining the resulting child solution/prey and u_{nd} is the random number between 0 and 1. A value of 0.5 is used for α as suggested by Deb (2002). With BLX-α, the location of the offspring in the decision space depends upon the difference between the parent solutions (Deb, 2002). This facilitates genetic recombination that is adaptive to the existing diversity in the parent population; a desirable characteristic for Pareto convergence. However, in case of MPP algorithm this promotes localised (local to each cell) combing of the domain for better solutions depending on the total span of the local solutions. This has the potential to be more cost effective with respect to function evaluations compared to other genetic algorithms that allow crossover between randomly selected solutions from the entire population.

This crossover child prey is then subjected to non-uniform mutation originally introduced by Michalewicz (1992), and mathematically formulated as

$$y_v^{(1,t+1)} = x_v^{(1,t+1)} + \tau\left(x_v^{(U)} - x_v^{(L)}\right)\left(1 - r_v^{\left(1-t/t_{\max}\right)^b}\right) \tag{7}$$

Here, $y_v^{(1,t+1)}$ is the value of the vth variable defining the child solution/prey produced by mutation of the value of the vth design variable $x_v^{(1,t+1)}$, $x_v^{(U)}$ and $x_v^{(L)}$ are upper and lower limits of the vth design variable, τ takes a Boolean value –1 or 1, each with a probability of 0.5, r_v is a random number between 0 and 1, t and $t_{\max}$ are the number of generations already executed and the maximum allowed number of generations, respectively, while exponent b is a user defined parameter.

Non-uniform mutation favours creation of child solutions in the vicinity of the parent solution, and the probability of creating a child solution closer to the parent increases with increasing number of generations. This provides a uniformly distributed search in the earlier generations and a relatively focused search in the later ones. A modified version of this non-uniform mutation has been applied in MPP, as follows.

$$\beta = 10^{-\left(t / t_{\max} \right)}$$
$$y_v^{(1,t+1)} = x_v^{(1,t+1)} + \beta\tau\left(x_v^{(U)} - x_v^{(L)}\right)\left(1 - r_v^{\left(1 - t / t_{\max}\right)^b}\right) \quad (8)$$

Here, t and $t_{\max}$ are the number of function evaluations performed until then and maximum allowed number of function evaluations, respectively, b is a constant and β is the scaling parameter. The latter two factors monitor the order of magnitude, or in other words, the extent of mutation and hence are chosen such that the order varies between 0.01 and 0.00001.

Both the crossover and mutation techniques employed here establish a localised and adaptive search, which make the MPP algorithm more economical with respect to function evaluations. The crossover probability should be maintained at unity (that is, 100%) since localised recombination is absolutely necessary for evolution of a population which lacks global mixing of solutions. The mutation probability used is small (around 0.05) which is usual for application in evolutionary algorithms. However, specific real world problems might demand a higher or lower mutation probability, which may not be possible to predict *a priori* without definite knowledge of the function space topology.

2.3 Dominance and constraint handling

The concept of weak dominance (Deb, 2002) is applied here both as a qualification criterion for new prey as well as in case of creating and updating the elite set. Primary selection is chiefly driven by the weighted sum of objectives concept, which has inherent drawbacks such as difficulty in approximating non-convex problems and a need for normalisation of the objective functions. Such deficiencies are counteracted by controlled introduction of non-dominance based operations in the MPP algorithm. It has been shown by Knowles and Corn (2007) that converging ability of selection based on non-dominance fails to scale up with increasing number of objectives and starts performing comparably to random search for problems with seven or more objectives.

However, it should be noted that complex real world problems generally demand substantial scalability in terms of number of design variables rather than number of objectives. In predator-prey algorithm, concept of non dominance is just a filtering technique and not the chief selection criterion, but it would be difficult to predict the scalability of MPP (with respect to number of objectives) without actually applying it on a problem with large number of objectives. In case of an unconstrained optimisation problem, solution A is said to weakly dominate solution B if solution A is better than solution B in at least one objective and equal in all other objectives. However, in case of a constrained optimisation, the theory of dominance is altered to give preference to feasible solutions or relatively less infeasible solutions. The modified definition of dominance is the same as used in NSGA-II (Deb et al., 2002), which is as follows,

Solution A is said to constraint-dominate solution B if

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

Due to the absence of any penalty function method, the normal objectives $(f_n, \forall n < N_f + 1)$ and the net constraint violation objective (f_{N_f+1}), get similar quantitative importance. This, together with the constraint-dominance criterion, favour feasible solutions, but also helps retain genetic traits of infeasible solutions with substantially better objective values as well. This speeds up convergence to the Pareto front especially when it is located at the boundary of the feasible region. Nevertheless, it should be noted that unless the whole prey population lies in the infeasible region (in the objective space) the progressing Pareto front will always constitute of feasible solutions, because the Pareto front is formed by the non-dominated elite solutions.

2.4 Diversity preservation

A multi-objective problem prefers a reasonably uniform distribution of solutions along the whole span of the Pareto front. This calls for preservation of diversity in the objective space. In other words, an efficient multi-objective optimisation algorithm is expected to promote generation of new solutions (evolution) that do not closely resemble their parents or other nearby solutions (in the objective space). Here, 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 and could be determined by the following equation (Chowdhury et al., 2008; Chowdhury and Dulikravich, 2009).

$$\begin{aligned} \omega &= 10^{-\left(2+ t/t_{\max}\right)} \\ \eta_n &= \omega \times \min\left(f_n^{new\ prey}, f_n^{old\ prey}\right) \end{aligned} \tag{9}$$

Here, ω is the window size of the hypercube and η_n is the half side length of the hypercube corresponding to the nth objective.

2.5 Sectional convergence (biased weighing of objectives)

A prominent drawback of the original predator-prey algorithm is its tendency to converge to a small section of the Pareto front due to absence of local selection pressure chiefly based on non-dominance. A new and innovative concept of sectional convergence has been introduced by Chowdhury et al. (2008) to deal with this possible lack of effective variation in the prey population. Instead of running the algorithm throughout for the same initial specified distribution of weights (for predator-objective association), a different distribution of weights within a smaller biased range (< 1.0) is applied after certain number of function evaluations. This redistribution is governed by the following equations in case of two-objective optimisation problems.

$$w_{1,m} = \frac{(iterp-1)M_P + m}{iterp_{\max} M_P + 1}$$
$$w_{2,m} = 1 - w_{1,m} \tag{10}$$

Here, m is the mth predator, $iterp_{\max}$ is the maximum allowed number of primary iterations, that is, maximum number of times redistribution is allowed, and $iterp$ is the present primary iteration. In case of multi-objective optimisation with more than two objectives, a different formula could be used (Chowdhury et al., 2008) as shown below.

$$w_{k\max,m} = \frac{\left(iterp - \frac{iterp_{\max}}{N_f}(K-1) - 1\right)M_p + m}{\frac{iterp_{\max}}{N_f} M_P + 1} \times 0.75 \tag{11a}$$

here,

$$K = \frac{iterp - 1}{iterp_{\max} / N_f} + 1 \tag{11b}$$

then

$$w_{n,m} = 1 - w_{k\max,m} \tag{11c}$$

Here, $w_{n,m}$ is the weight associated with the nth objective function for the mth predator and $w_{k\max,m}$ is the maximum allowable weight associated with any kth objective function $(k \neq n)$ for the mth predator. The weights $(w_{k,m})$ associated with the objective functions other than the nth objective are distributed using Sobol's (1976) algorithm within the range 0 to $w_{k\max,m}$. However, in this case (that is, problems where $N_f > 2$), $iterp_{\max} / N_f \in N_S$ is an essential condition.

This added feature involving biased distribution of weights does away with the often observable drawback of PP which is its tendency to converge to a small section of the Pareto front due to absence of selection pressure chiefly based on non-dominance. Nevertheless, such sectional convergence comes at the cost of an increased number of function evaluations which might be necessary only in case of complex problems such as sharp discontinuities or mixed convex-concave Pareto fronts or orders of magnitude difference between the objective functions.

2.6 Elitism

In order to retain the genetic traits of the best solutions it is necessary to introduce some form of elite preservation mechanism into the algorithm. This, when judiciously applied, accelerates the rate of convergence to the Pareto front. In MPP, the secondary set (elite set or archive) consisting of the non dominated solutions from each generation is maintained at a fixed strength Ne using the clustering technique designed by Deb et al. (2002). After each generation, certain randomly selected solutions/prey (from the main population), if found to be dominated, are replaced from the 2D lattice by randomly selected elite solutions. This new additional attribute boosts the speed of convergence of this algorithm. However, the allowed number of such replacements should be carefully chosen to avoid introducing excessive elitism. Here, the total number of allowed replacements is always kept below $N_S/2$, which was empirically found to be an approximate threshold beyond which diversity is likely to be compromised.

2.7 Additional features

During the course of development of MPP a few other alterations/additional features were also implemented, but not included in the final version of the algorithm. This was due to certain drawbacks associated with each one of them. A couple of them are being presented here, keeping in mind that a more judicious application of any of these features, in the future, might help to improve the dependability or performance of MPP or other similar evolutionary optimisation algorithms. They have been discussed below.

Controlled killing in active localities: Instead of killing exactly one prey (the weakest) at each active locality during a generation, the predator was allowed to kill 'ℓ' number of the weakest local prey depending on the 'non-domination' quality of the locality. The value of ℓ for each locality was computed according to the following formula.

$$\ell = \begin{cases} 0 & \text{if } ne \geq 3 \\ 1 & \text{if } 3 > ne > 0 \\ 2 & \text{if } ne = 0 \end{cases} \tag{12}$$

Here, ne is the number of prey from that locality that qualified for the elite set when last updated ($ne \in \{1, 2, 3, 4\}$). This reduced the required number of function evaluations, but severely hindered further progress when solutions converged to a local Pareto front.

Relocating prey: Like predators, prey were also relocated randomly within the same 2D lattice after every '*nm*' iterations, where $nm = N_S/M_P$. A favourable genetic mixing was observed, leading to greater diversity, but at the cost of noticeably increased number of function evaluations.

3 Numerical experiments

3.1 Unconstrained two-objective test case

MPP was tested to evaluate its performance by running it on some well-known unconstrained two-objective test problems, with known analytical solution. The first six test cases analysed are taken from the multi-objective optimisation comparison by Zitzler et al. (2000) namely the ZDT test cases. Two other popular test cases with known analytical solutions for the Pareto front which are the Fonseca and Fleming multi-objective problem no. 2 (Fonseca and Fleming, 1995) and the Coello multi-objective problem (Coello et al., 2002) have also been used. All the eight test cases involve two-objective optimisations where both objectives are to be minimised (Table 1).

Table 1 Details of unconstrained two-objective optimisation test cases

Problem	*m*	*Variable limits*	*Objective functions*	*Analytical solution*
ZDT1	30	$x_i \in [0,1]$	$f_1 = x_1$ $g = 1 + 9\sum_{i=2}^{m} \frac{x_i}{m-1}, \quad h = 1 - \sqrt{\frac{f_1}{g}}$ $f_2 = h.g$	Set $g = 1$
ZDT2	30	$x_i \in [0,1]$	$f_1 = x_1$ $g = 1 + 9\sum_{i=2}^{m} \frac{x_i}{m-1}, \quad h = 1 - \left(\frac{f_1}{g}\right)^2$ $f_2 = h.g$	Set $g = 1$
ZDT3	30	$x_i \in [0,1]$	$f_1 = x_1$ $g = 1 + 9\sum_{i=2}^{m} \frac{x_i}{m-1}$ $h = 1 - \sqrt{\frac{f_1}{g}} - \left(\frac{f_1}{g}\right)\sin(10\pi f_1)$ $f_2 = h.g$	Set $g = 1$
ZDT4	10	$x_1 \in [0,1]$ $x_i \in [-5,5]$	$f_1 = x_1$ $g = 1 + 10(m-1)$ $+ \sum_{i=2}^{m} \left(x_i^2 - 10\cos(4\pi x_i)\right)$ $h = 1 - \sqrt{\frac{f_1}{g}}$ $f_2 = h.g$	Set $g = 1$
ZDT5	11	$x_i \in [0,1]$, 30 bit resolution $x_i \in [0,1]$, 5 bit resolution	$f_1 = 1 + u(x_1)$ $u(x_i)$ = the number of ones in the bit vector form of x_i $g = \sum_{1=2}^{m} v(u(x_i)), \quad h = \frac{1}{f_1}$ $v(u(x_i)) = \begin{cases} 2 + u(x_i) & \text{if } u(x_i) < 5 \\ 1 & \text{if } u(x_i) = 5 \end{cases}$ $f_2 = h.g$	Set $g = 10$

Note: m = number of variables

Table 1 Details of unconstrained two-objective optimisation test cases (continued)

Problem	*m*	*Variable limits*	*Objective functions*	*Analytical solution*
ZDT6	10	$x_i \in [0,1]$	$f_1 = 1 - e^{-4x_1}\sin^6(6\pi x_1)$ $g = 1 + 9\left(\frac{\sum_{i=2}^{m} x_i}{m-1}\right)^{0.25}$, $h = 1 - \left(\frac{f_1}{g}\right)^2$ $f_2 = h.g$	Set $g = 1$
Fonseca-Fleming	3	$x_i \in [-4,4]$	$f_1 = 1 - e^{\left(-\sum_{i=1}^{m}\left(x_i - \frac{1}{\sqrt{m}}\right)^2\right)}$ $f_2 = 1 - \exp\left(-\sum_{i=1}^{m}\left(x_i + \frac{1}{\sqrt{m}}\right)^2\right)$	$f_2 = 1 - e^{-\left(2-\sqrt{-\ln(1-f_1)}\right)^2}$
Coello	2	$x_i \in [0,1]$	$f_1 = x_1$ $f_2 = (1+10x_2)\left(1 - \left(\frac{x_1}{1+10x_2}\right)^2 - \frac{x_1}{1+10x_2}\sin(8\pi x_1)\right)$	$f_2 = 1 - f_1^2 - f_1 \sin(8\pi f_1)$

Note: m = number of variables

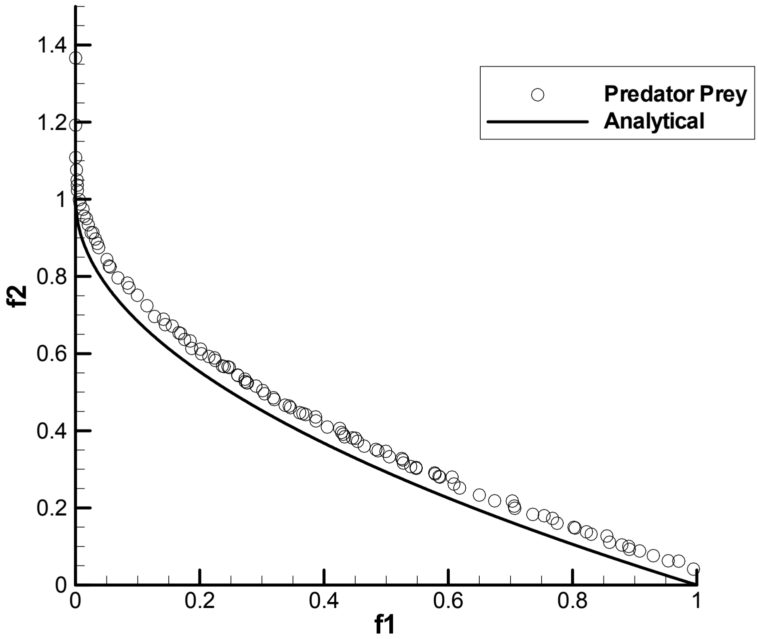
To compensate for performance fluctuations induced by random generators driving the initial population and other genetic operators, the algorithm was run five times for 25,000 function evaluations each in case of the six ZDT test cases and 2,000 function evaluations each in case of the Fonseca-Fleming and Coello test problems. The concept of sectional convergence was not implemented during these runs.

The non-dominated plots are generated by making a union of the elite set (non-dominated set) of the first five runs for each test case. The non-dominated set of the unions is then extracted and plotted as shown in Figures 2 to 9. The user defined MPP parameters used for these test cases are as follows.

Table 2 User defined MPP parameters for unconstrained two-objective test cases

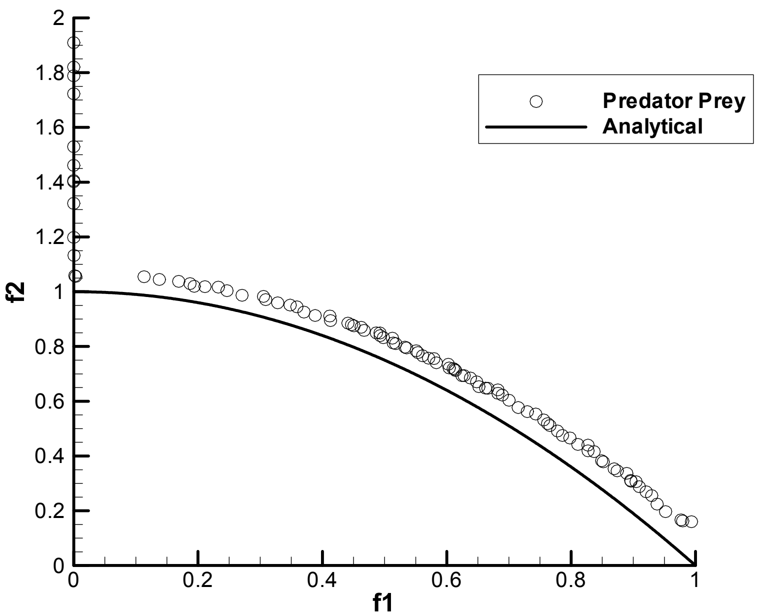
Parameter	*Value*
Population size (# prey)	100
# Predators	10
Elite set size	40
Crossover probability	1.0
Mutation probability	0.05

Figure 2 Two-objective test case ZDT 1



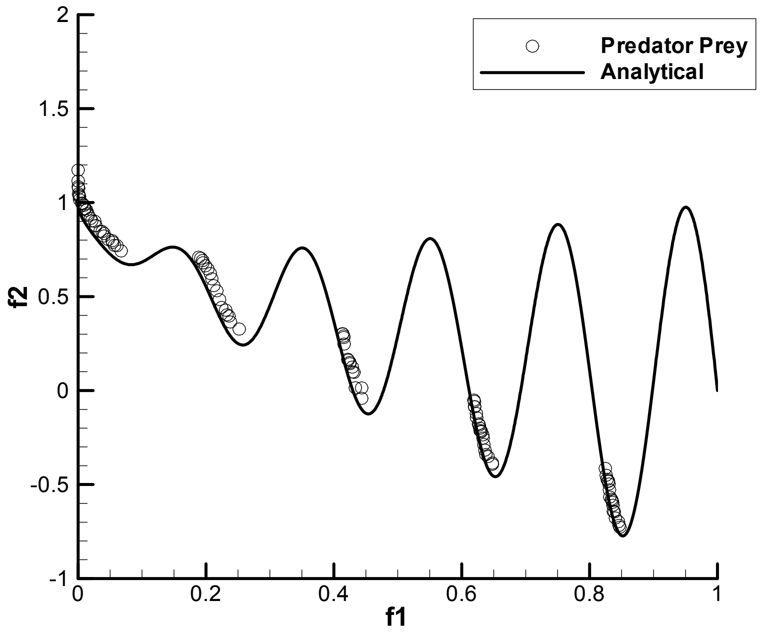
Note: Results with MPP.

Figure 3 Two-objective test case ZDT 2



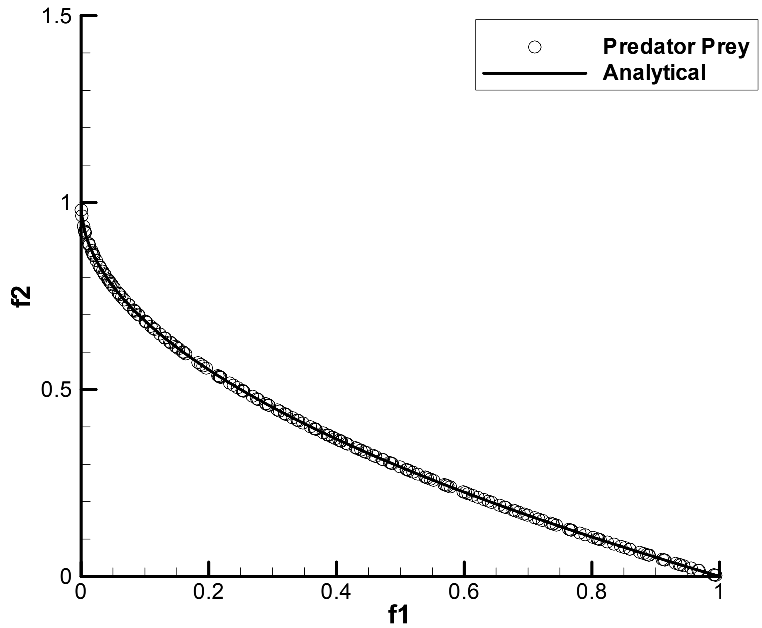
Note: Results with MPP.

Figure 4 Two-objective test case ZDT 3



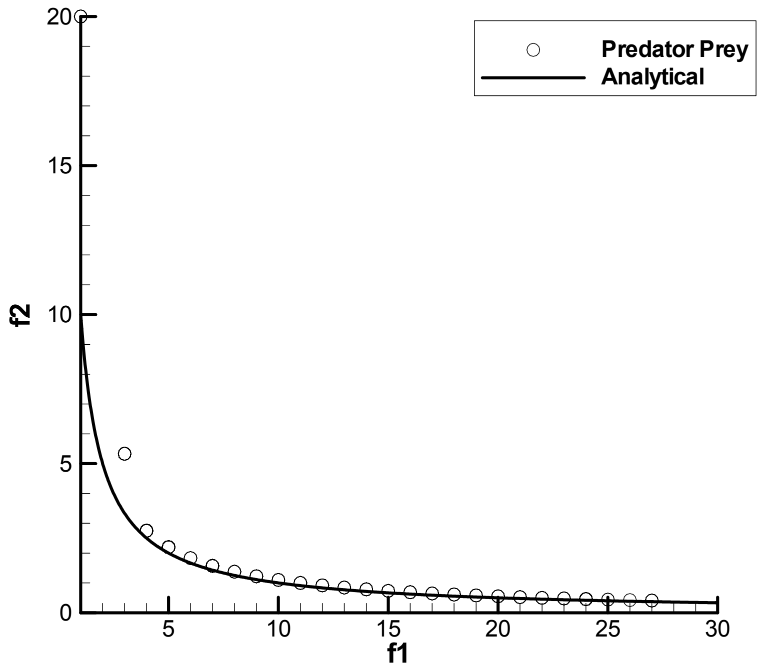
Note: Results with MPP.

Figure 5 Two-objective test case ZDT 4



Note: Results with MPP.

Figure 6 Two-objective test case ZDT 5



Note: Results with MPP.

Figure 7 MPP results for two-objective test case ZDT 6

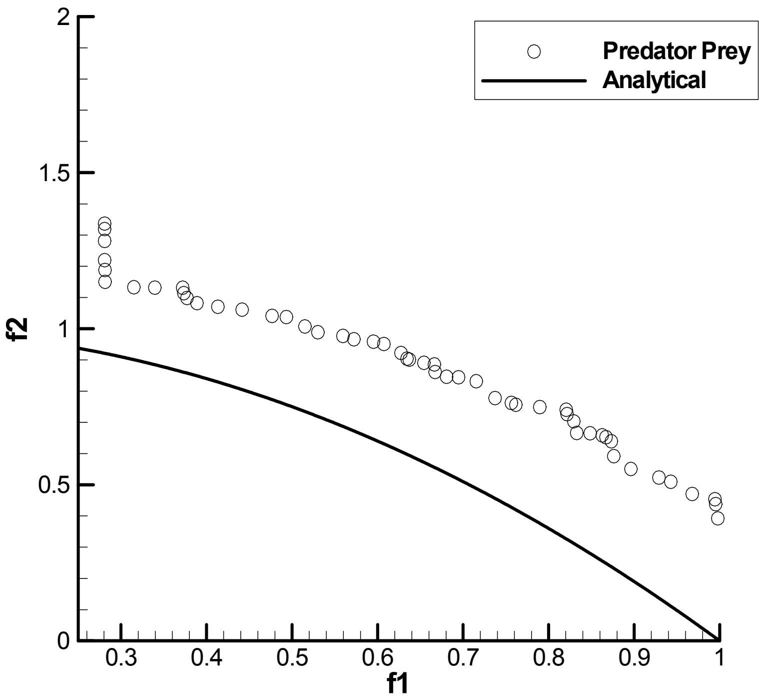


Figure 8 MPP results for two-objective test case of Fonseca and Fleming problem 2

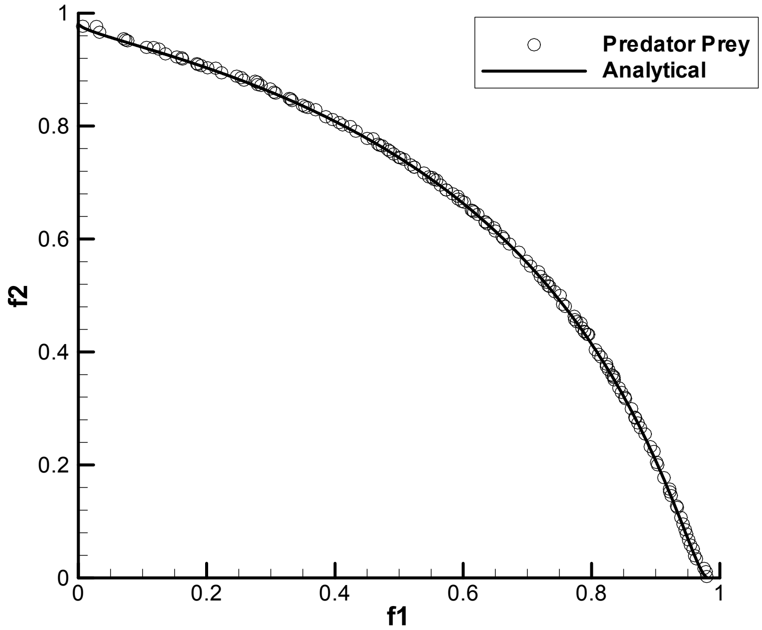
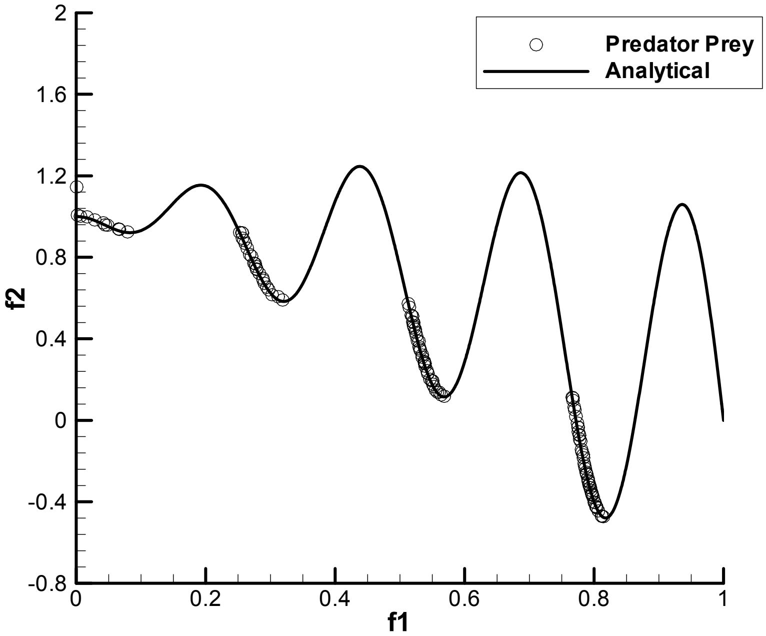


Figure 9 Two-objective test case of Coello



Note: Results with MPP.

It is observed from Figures 2 to 9, that MPP performs very well on the ZDT test cases when compared with the performances of some other well-known algorithms as shown by Moral and Dulikravich (2008) (given in Appendix A). The same is exhibited in case of the Fonseca-Fleming and Coello test problems as shown in Figures 8 and 9 respectively. In certain cases as in ZDT 1, 2 and 6 the solutions do not completely converge to the Pareto front. This is due to significant slowing down of the rate of convergence as the solutions approach the fully converged computed version of the Pareto front.

Nevertheless, it is evident from the above figures that the algorithm consistently produces a desirable spread of non-dominated solutions irrespective of the nature of the Pareto front and without using the concept of sectional convergence.

Two performance measures for evaluating the performance of multi-objective optimisation algorithms have been developed by Deb et al. (2002). The first performance metric, the gamma (γ) parameter, is a measure of the extent of convergence.

The minimum of the Euclidean distances of each computed non-dominated solution from H uniformly distributed points on the ideal Pareto front (H = 500) is calculated, the average of which gives the value of the gamma parameter. The other performance metric, namely the delta (Δ) parameter, gives a measure of the spread of solutions along the computed Pareto front. It is calculated as follows.

$$\Delta = \frac{d_f + d_l + \sum_{ns=1}^{N_S-1} \left| d_{ns} - \bar{d} \right|}{d_f + d_l + \left(N_S - 1 \right) \bar{d}} \tag{13}$$

Here, d_f and d_l are the respective Euclidean distances between the two extreme solutions and the corresponding extremities of the analytical Pareto front, d_{ns} is the Euclidean distance between consecutive solutions and $\bar{d}$ is the mean of all d_{ns} (ns = 1, 2, 3,.., N_S). A perfectly uniform distribution of solutions along the computed Pareto front with existence of exact extreme solutions will give a delta value of zero. However, in spite of accurate convergence, the gamma parameter need not be zero, due to possible lack of coincidence of computed solutions and uniformly distributed analytical Pareto points.

Table 3 shows the values of these two parameters calculated for the eight cases studied here, and also the comparison of some of them with that calculated by Deb et al. (2002) for NSGA-II. The same conditions have been used, i.e., a population of 100 solutions, subjected to 25000 function evaluations, for the six ZDT test cases. However, the Fonseca-Fleming and the Coello test cases involve 2,000 function evaluations and hence the former has not been compared with the corresponding data of Deb et al. (2002), all of which are with respect to 25,000 function evaluations.

As seen from Table 3, the performance of MPP compares well with that of real coded NSGA-II, except in the case of ZDT 2. The latter may be attributed to the vertical congregation of points near the left boundary of the Pareto front where an abrupt change in the value of f_2 corresponding to very small values of f_1 poses difficulty in properly distributing ideal Pareto points in this region. However, in the case of ZDT 3, the MPP seems to outperform both the real coded and the binary coded NSGA, in accuracy. As seen from Figure 5, a fairly accurate and well distributed non-dominated solution set is computed by MPP in the case of ZDT 4. Due to the high density of solutions along the computed Pareto front, the deviation in d_{ns} exceeds the average, $\overline{d}$, which accounts for the relatively high value of Δ (>1), calculated in case of ZDT 4.

Table 3 Performance indicators

Algorithm	*NSGA-II (real)*		*NSGA-II (binary)*		*MPP*	
Problems	*γ*	*Δ*	*γ*	*Δ*	*γ*	*Δ*
ZDT 1	0.0335	0.39	0.0009	0.46	0.0447	0.59
ZDT 2	0.0724	0.43	0.0009	0.44	0.1181	0.78
ZDT 3	0.1145	0.73	0.0434	0.58	0.0198	0.73
ZDT 4	0.5130	0.70	3.2276	0.48	0.6537	1.48
ZDT 5	N/A	N/A	N/A	N/A	0.4282	1.49
ZDT 6	0.2966	0.67	7.8068	0.64	0.2334	0.71
Fonseca-Fleming	N/A	N/A	N/A	N/A	0.0082	0.42
Coello	N/A	N/A	N/A	N/A	0.0498	1.17

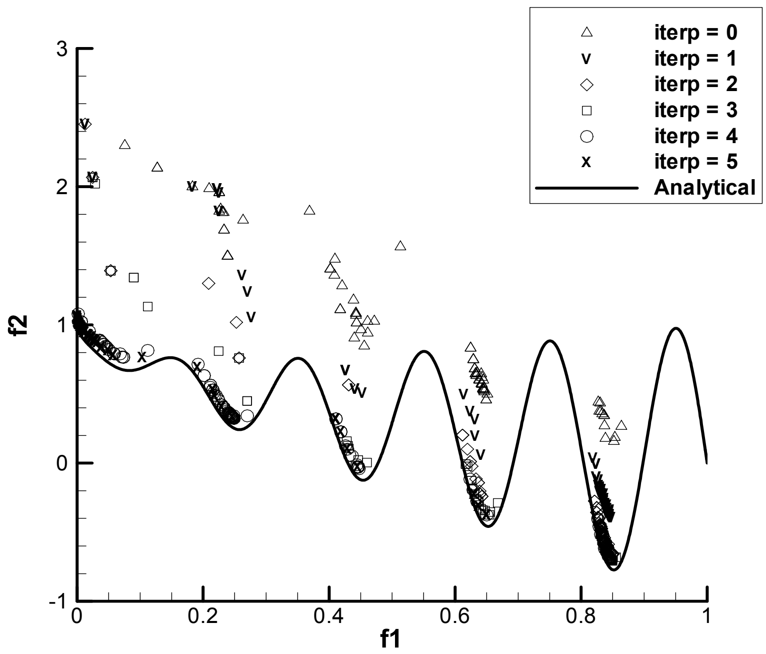
Difficulties encountered in converging to the ideal Pareto front in the case of ZDT 5, by other standard optimisation algorithms have been claimed to be not trivial, as also confirmed by Deb et al. (2002). However during the course of this study, it has been found that achieving acceptable accuracy in the case of ZDT 5 to be relatively manageable as evident from Figure 6 and Table 3. But the above is true only when the correct order of precision is used in representing the decision variables and computing the objective functions. Failure to do so might be the very reason behind the relatively low accuracy of solutions computed by other optimisation algorithms while dealing with ZDT 5.

Test case results presented in this work are generated without considering the concept/module of sectional convergence. However, sectional convergence was experimented on during the study of MPP and a visual representation is exhibited in Figure 10.

Figure 10 shows the location of solutions in the objective space at the end of each primary iteration (*iterp*), where $iterp = 0$ represents the initial global progression of solutions, and $iterp > 0$ represents the sequential sectional convergence of solutions to parts of the Pareto front. Though the progress of solutions is biased towards sections of the objective space going from right to left, the solution set as a whole always keeps moving towards the ideal Pareto front. This is desirable and eventually leads to a well distributed set of non-dominated solutions along the final computed Pareto front.

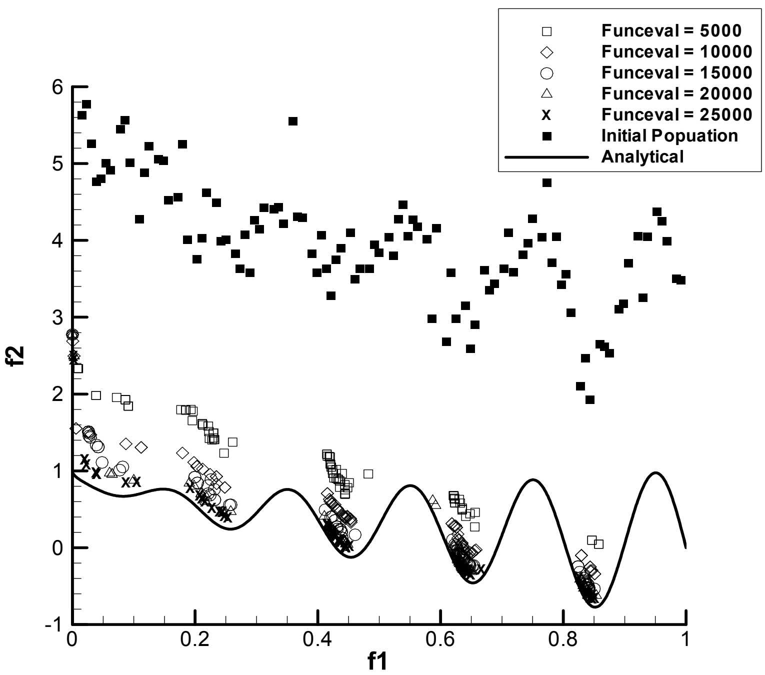
Figure 11 shows the progress of solutions towards the ideal Pareto front, in absence of the sectional convergence module. The solutions are plotted after intervals of 5,000 (approx.) function evaluations. It is observed that the solutions converge noticeably faster during the initial stages of MPP to form an intermediate computed Pareto front. The subsequent progress of this intermediate front becomes more and more exhaustive in terms of function evaluations as it nears the global Pareto front.

Figure 10 Sectional convergence for ZDT 3



Note: $iterp_{\max} = 5$

Figure 11 General convergence for ZDT 3



Note: $iterp_{\max} = 0$

3.2 Unconstrained three-objective test cases

Multi-objective optimisation algorithms often demonstrate different behaviour when working on problems with more than two objectives. The Pareto front is just a planar curve in two-objective problems which proliferates into a surface in three-objective problems, and then to a hyper surface of increasing dimensionality with every additional problem objective. This intensifies the necessity for careful preservation of diversity. Selection procedure based on either weighted sum of objectives or weak domination criterion work very differently. For example, say in the case of a problem with N_f objectives $(N_f > 2)$, solution A has one objective better than solution B, while in all other objectives solution B ranks higher. Weighted sum would most likely recognise solution B as the better solution whereas according to the principles of weak dominance both solutions are non-dominated w.r.t. each other. Predator-Prey is unique in utilising the principles of both selection procedures. However, the performance gain of such a characteristic can be appreciated only when the algorithm is tested on optimisation problems with more than two objectives. Therefore, MPP is tested on two standard scalable three-objective minimisation problems developed by Deb et al. (2006). They are summarised in Table 4.

Table 4 Details of the unconstrained three-objective optimisation test cases

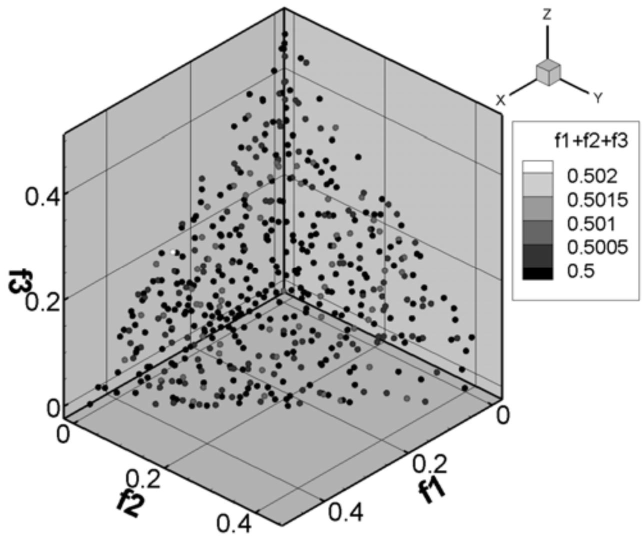
Problem	*m*	*Variable limits*	*Objective functions*	*Analytical solution*
DTLZ1	7	$x_i \in [0,1]$	$g = 100\left(5 + \sum_{i=3}^{7}\left(\begin{array}{l}(x_i - 0.5)^2 \\ -\cos(20\pi(x_i - 0.5))\end{array}\right)\right)$ $f_1 = \frac{1}{2}x_1x_2(1+g)$ $f_2 = \frac{1}{2}x_1(1-x_2)(1+g)$ $f_3 = \frac{1}{2}(1-x_1)(1+g)$	$x_j = 0$ $j = 3,4,..,7$ $\sum_{k=1}^{3} f_k = 0.5$
DTLZ2	12	$x_i \in [0,1]$	$g = \sum_{3}^{12}(x_i - 0.5)^2$ $f_1 = (1+g)\cos(x_1\pi/2)\cos(x_2\pi/2)$ $f_2 = (1+g)\cos(x_1\pi/2)\sin(x_2\pi/2)$ $f_3 = (1+g)\sin(x_1\pi/2)$	$x_j = 0.5$ $j = 3,4,..,12$ $\sum_{k=1}^{3} f_k{}^2 = 1$

Due to similar reasons as in case of the ZDT test cases, both DTLZ test cases were run 30 times, 30,000 function evaluations each, and the final version of the computed Pareto front is formed by extracting the non dominated set from the union of the final elite sets of the first five runs. The user defined parameters specified in the algorithm for these test cases are presented in Table 5.

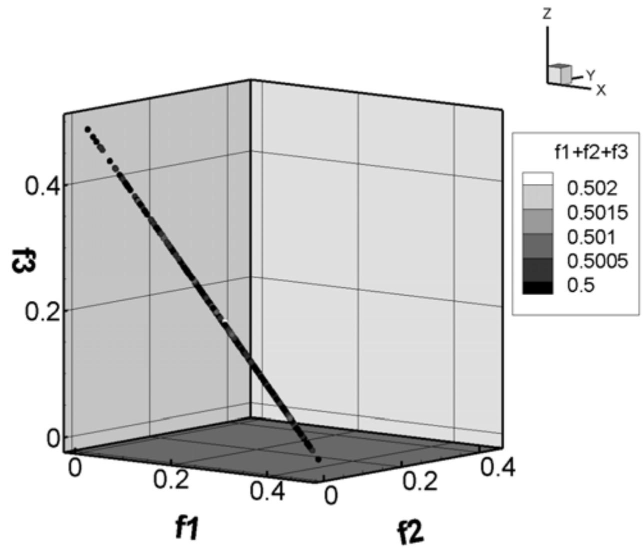
Table 5 General MPP parameters for unconstrained three-objective test cases

Parameter	*Value*
Population size (# prey)	100
# Predators	10
Elite set size	40
Crossover probability	1.0
Mutation probability	0.05

Figure 12 Three-objective test case DTLZ1 with $iterp_{\max} = 0$: (a) view 1 (b) view 2



(a)

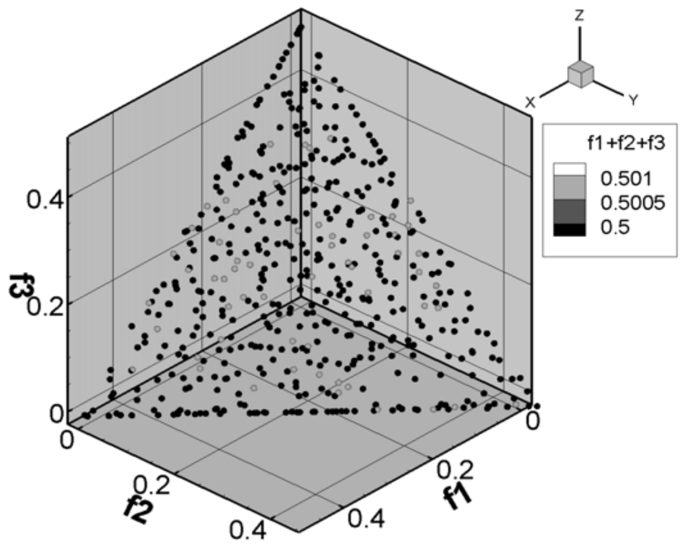


(b)

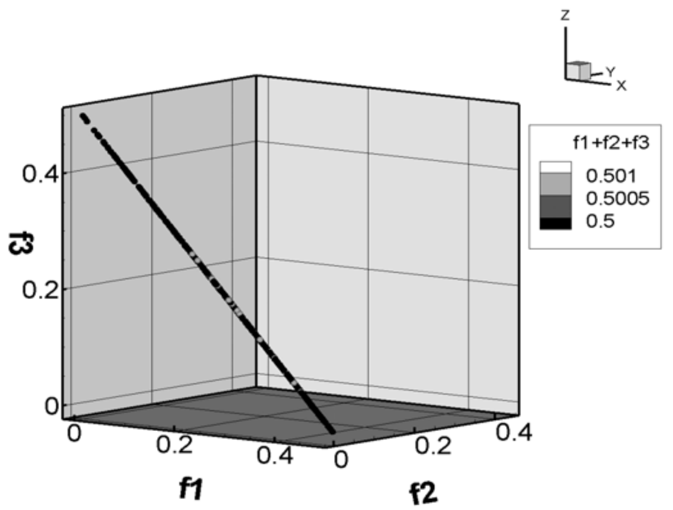
Different views of the final version of the computed Pareto front for the three-objective problems DTLZ1 and DTLZ2 have been illustrated in Figures 12 to 15. It is observed that MPP performs very well in computing Pareto solutions that are both reasonably accurate and well distributed along the Pareto front. Sectional convergence scheme as seen from Figures 13 and 15 helps in covering the whole global Pareto front more effectively.

Hence, the boundaries of the global Pareto front computed by MPP are crisply defined when using sectional convergence.

Figure 13 Three-objective test case DTLZ1 with $iterp_{\max} = 0$: (a) view 1 (b) view 2



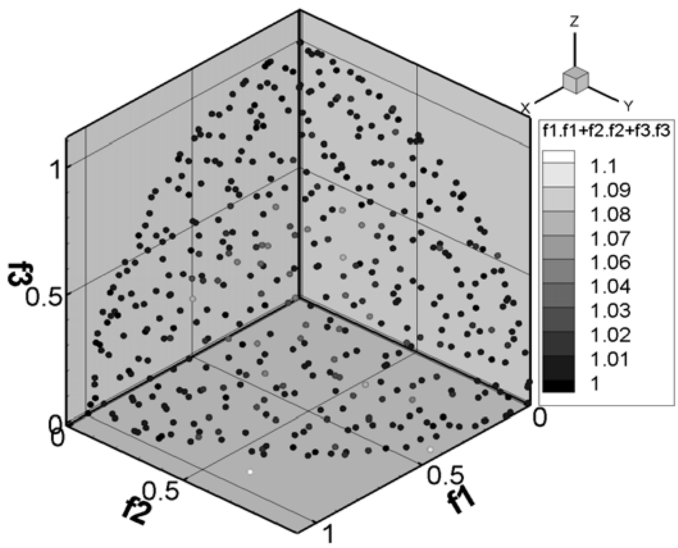
(a)



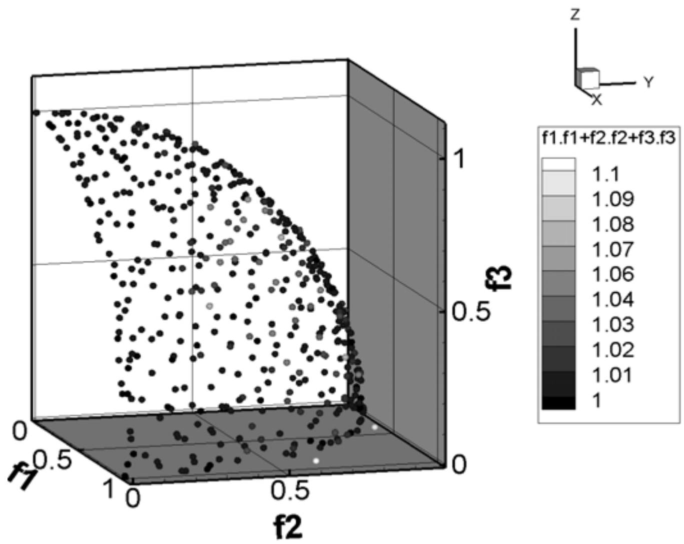
(b)

It would be too optimistic to extrapolate the performance appreciation of MPP from three objectives to N objectives $(N_f > 3)$. Pareto fronts computed by MPP in case of the DTLZ test cases indicate that MPP has the potential to achieve reasonably accurate well distributed Pareto fronts in case of optimisation problems with more than two objectives; a quality not so common among the standard multi-objective optimisation algorithms available in literature and practice. However, scalability of the performance of MPP with significantly higher number of objectives needs further investigation.

Figure 14 Three-objective test case DTLZ2 with $iterp_{\max} = 0$: (a) view 1 (b) view 2

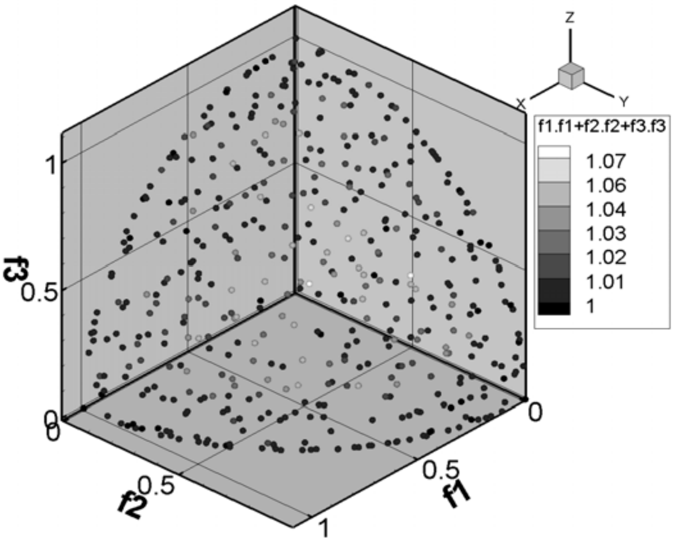


(a)

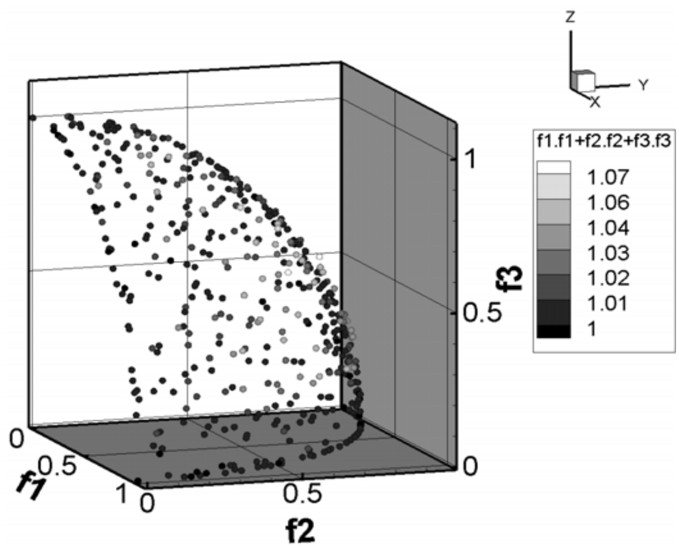


(b)

Figure 15 Three-objective test case DTLZ2 with $iterp_{\max} = 3$: (a) view 1 (b) view 2



(a)



(b)

It is seen from Figures 2 to 9 and Figures 12 to 15 that MPP performs very well (often better) both in case of two and three objective problems, when compared with one of the more popular previous versions of the predator-prey algorithm (see Appendix A) despite of working under relatively stricter conditions (fewer number of function evaluations).

3.3 Constrained multi-objective test cases

To examine the constraint handling capability of MPP, it was tested on three well-known constrained two-objective test cases studied by Deb et al. (2002). Two standard test cases with known analytical solutions namely Binh multi-objective optimisation problem no. 2 (Binh and Korn, 1997) and the Osyczka multi-objective optimisation problem no. 2 (Osyczka and Kundu, 1995) have also been used to test MPP. All these test cases are two-objective minimisation problems and are summarised in Table 6.

Table 6 Details of the constrained two-objective optimisation test cases

Problem	*m*	*Variable limits*	*Objective functions*	*Constraints*
CONSTR	2	$x_1 \in [0.1,1]$ $x_1 \in [0,5]$	$f_1 = x_1$ $f_2 = (1 + x_2)/x_1$	$x_2 + 9x_1 \geq 6$ $-x_2 + 9x_1 \geq 1$
SRN	2	$x_1 \in [-20,20]$	$f_1 = (x_1 - 2)^2 + (x_2 - 1)^2 + 2$ $f_2 = 9x_1 - (x_2 - 1)^2$	$x_1^2 + x_2^2 \leq 225$ $x_1 - 3x_2 \leq -10$
TNK	2	$x_i \in [0,\pi]$	$f_1 = x_1$ $f_2 = x_2$	$-x_1^2 - x_2^2 + 1$ $+0.1\cos\left(16\tan^{-1}\left(\frac{x_1}{x_2}\right)\right) \leq 0$ $(x_1 - 0.5)^2 + (x_2 - 0.5)^2 \leq 0.5$
Binh and Korn (1997)	2	$x_i \in [0,5]$ $x_i \in [0,3]$	$f_1 = 4x_1^2 + 4x_2^2$ $f_2 = (x_1 - 5)^2 + (x_2 - 5)^2$	$(x_1 - 5)^2 + x_2^2 - 25 \geq 0$ $-(x_1 - 8)^2 - (x_2 + 3)^2$ $+7.7 \geq 0$
Osyczka and Kundu (1995)	6	$x_1 \in [0,10]$ $x_1 \in [0,10]$ $x_1 \in [1,5]$ $x_1 \in [0,6]$ $x_1 \in [1,5]$ $x_1 \in [0,10]$	$f_1 = -\left(\begin{array}{l}25(x_1 - 2)^2 \\ +(x_2 - 2)^2 \\ +(x_3 - 1)^2 \\ +(x_4 - 4)^2 \\ +(x_5 - 1)^2\end{array}\right)$ $f_2 = x_1^2 + x_2^2 + x_3^2 + x_4^2 + x_5^2 + x_6^2$	$x_1 + x_2 - 2 \geq 0$ $6 - x_1 - x_2 \geq 0$ $2 + x_1 - x_2 \geq 0$ $2 - x_1 + 3x_2 \geq 0$ $4 - (x_3 - 3)^2 - x_4 \geq 0$ $(x_5 - 3)^2 + x_6 - 4 \geq 0$

Each constrained test case given in Table 6 was run 30 times and the final computed version of the Pareto front is formed in the same way as in the ZDT and the DTLZ test cases. However, the final computed versions of the Pareto fronts in case of all the five constrained test cases (Table 6) were created of only those elite set solutions that do not violate any of the problem constraints. For the final Pareto solutions,

$$f_{N_f+1} = \sum_{ic=1}^{p} \max(g_{ic}, 0) + \sum_{ic=p+1}^{p+q} \max((h_{ic} - \varepsilon), 0) = 0 \tag{14}$$

It is worth mentioning that MPP achieved the elite set, constituted of such feasible global Pareto solutions, in each of these test cases. The user defined values in the MPP pertinent to the constrained test cases are presented in Table 7.

Table 7 General MPP parameters for two-objective test cases

Parameter	*First three constrained two-objective test cases*	*Last two constrained two-objective test cases*
	Value	*Value*
Population size (# prey)	100	100
# Predators	10	10
Elite set size	40	100
Crossover probability	1.0	1.0
Mutation probability	0.05	0.05
# Primary iterations (sections)	0, 3	0, 6

A higher number of primary iterations and greater elite set size were used in case of the Binh and Korn (1997) and the Osyczka and Kundu (1995) problems (Table 6). This is to counteract the relatively greater difficulty in covering the whole Pareto front in these two test problems. The converged versions of the Pareto fronts computed by MPP in each of these test cases are shown in Figures 16 to 20.

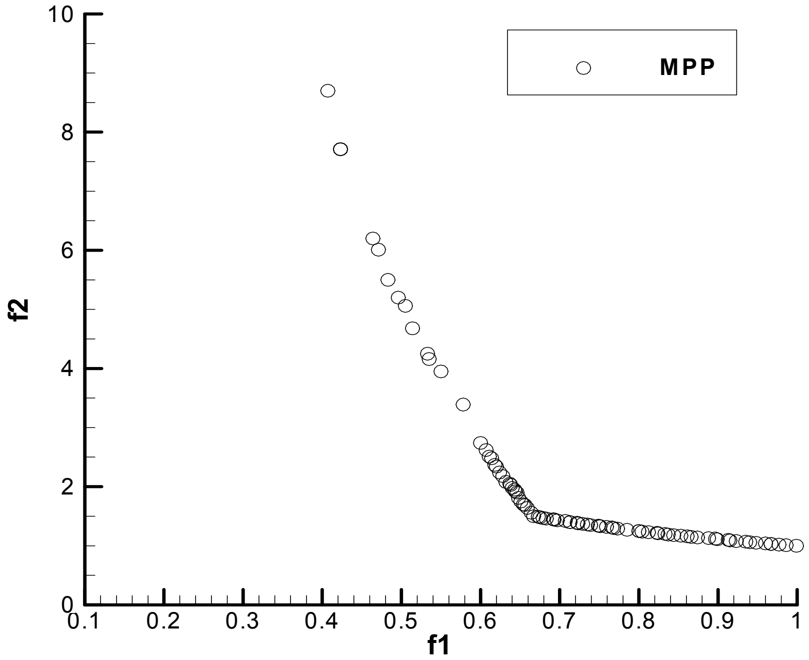
The final computed versions of the Pareto fronts for SRN, TNK and Binh constrained multi-objective problems as shown in Figures 16, 17 and 18 respectively are fairly accurate and well distributed. However, in the Binh problem there is significant improvement in performance when using the sectional convergence scheme (Figures 18a and 18b). In case of CONSTR and Osyczka constrained multi-objective problems (Figures 16 and 20), though solutions converge to the global Pareto front, their distribution on the final computed versions of the Pareto front is not uniform, even with the sectional convergence scheme.

Overall, MPP compares well in performance, with other popular algorithms such as NSGA II (Deb et al., 2002) and IOSO algorithms (IOSO, 2003) in solving similar constrained multi-objective problems at the expense of limited number of function evaluations. Nevertheless, appropriate implementation of the sectional convergence scheme is necessary for certain problems (Chowdhury et al., 2009) in order to attain a reasonable spread of solutions along the final computed version of the Pareto front.

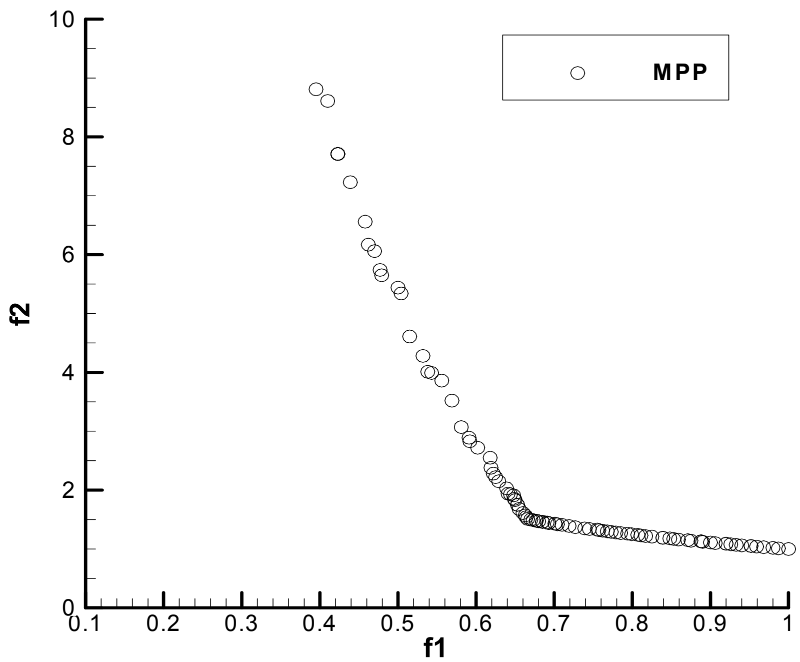
The remarkable feature of MPP is its ability to consistently produce *feasible* Pareto solutions, irrespective of the number or nature (i.e., linear or non-linear) of problem constraints involved. This is accomplished without normalisation of any objective functions or constraint functions, or application of computationally costly penalty function methods.

Figure 21 demonstrates an immediate migration of solutions into the feasible region and concomitant advancement towards the global Pareto front during the initial stages of the algorithm in case of Osyczka and Kundu (1995) test case. Hence, the pace at which MPP drives the population into the feasible domain and subsequently converges to the global Pareto front is appreciable – a quality which may be attributed to the simultaneous application of the added constraint objective (to be minimised) and constraint dominance criterion introduced by Deb et al. (2002).

Figure 16 Constrained two-objective test case CONSTR with (a) $iterp_{\max} = 0$ (b) $iterp_{\max} = 3$

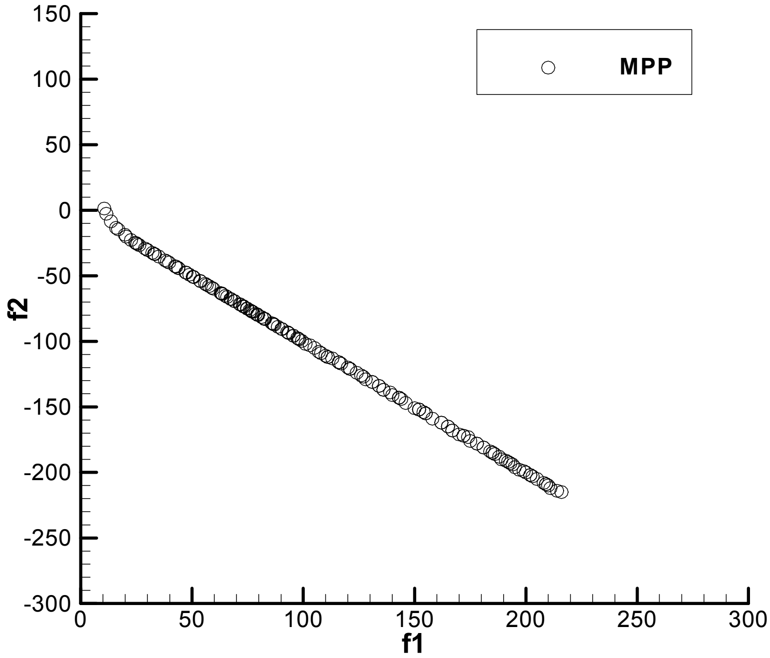


(a)

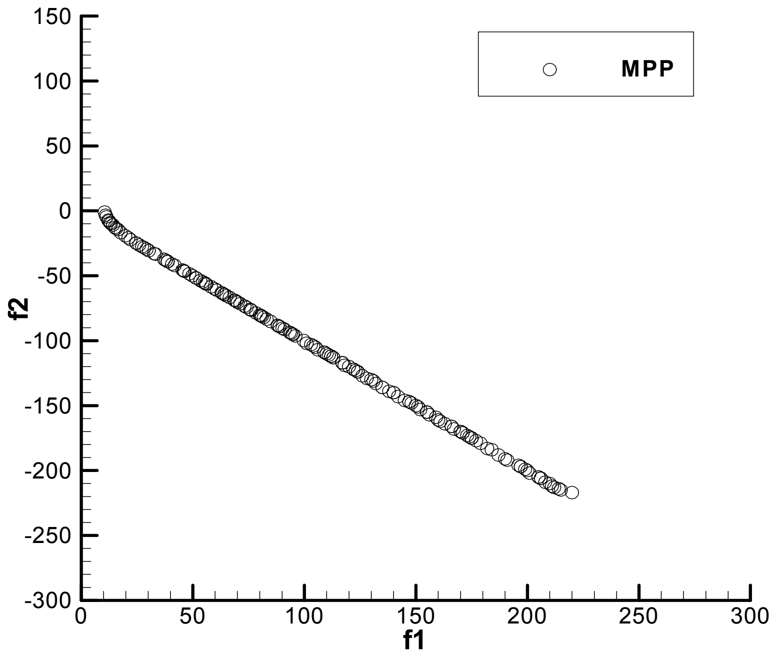


(b)

Figure 17 Constrained two-objective test case SRN with (a) $iterp_{\max} = 0$ (b) $iterp_{\max} = 3$

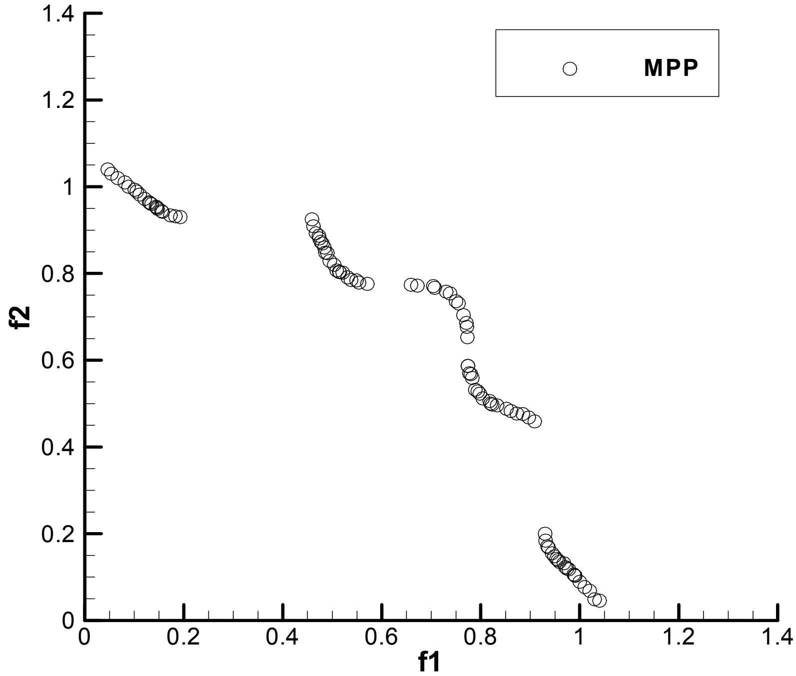


(a)

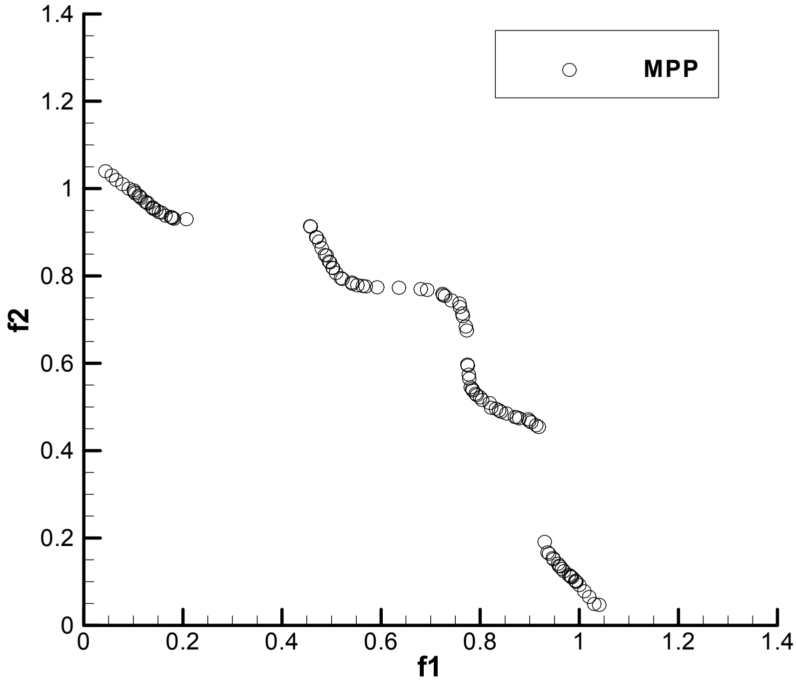


(b)

Figure 18 Constrained two-objective test case TNK with (a) $iterp_{\max} = 0$ (b) $iterp_{\max} = 3$

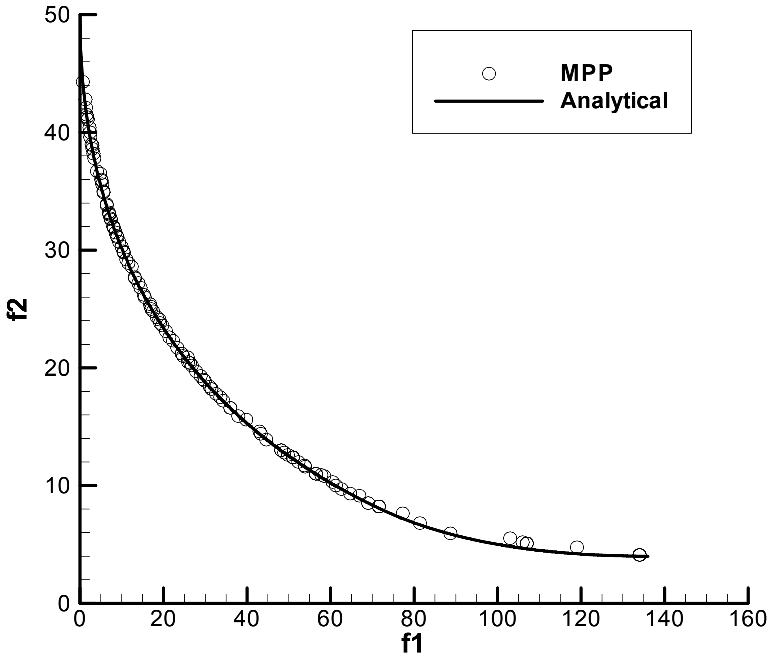


(a)

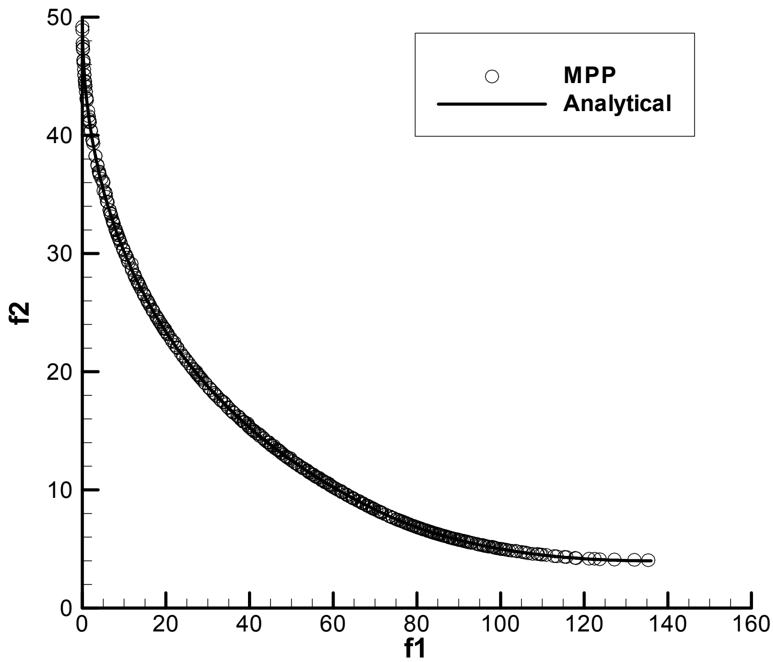


(b)

Figure 19 Constrained two-objective test case of Binh and Korn (1997) with (a) $iterp_{\max} = 0$ (b) $iterp_{\max} = 6$

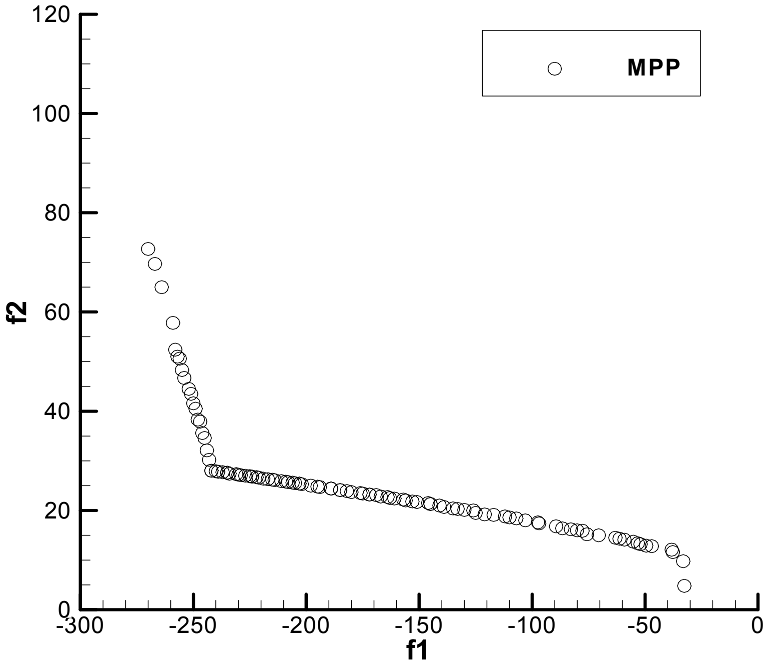


(a)

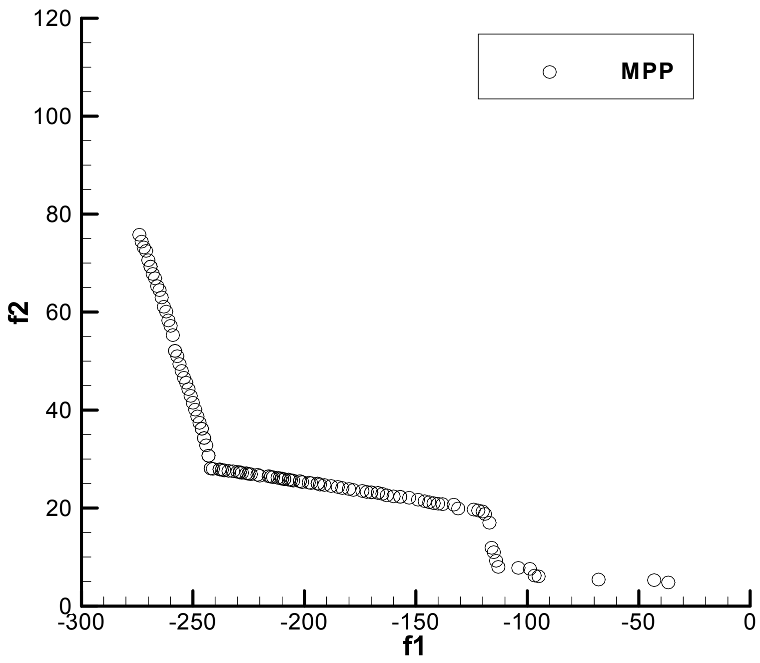


(b)

Figure 20 Constrained two-objective test case of Osyczka and Kundu (1995) with (a) $iterp_{\max} = 0$ (b) $iterp_{\max} = 6$

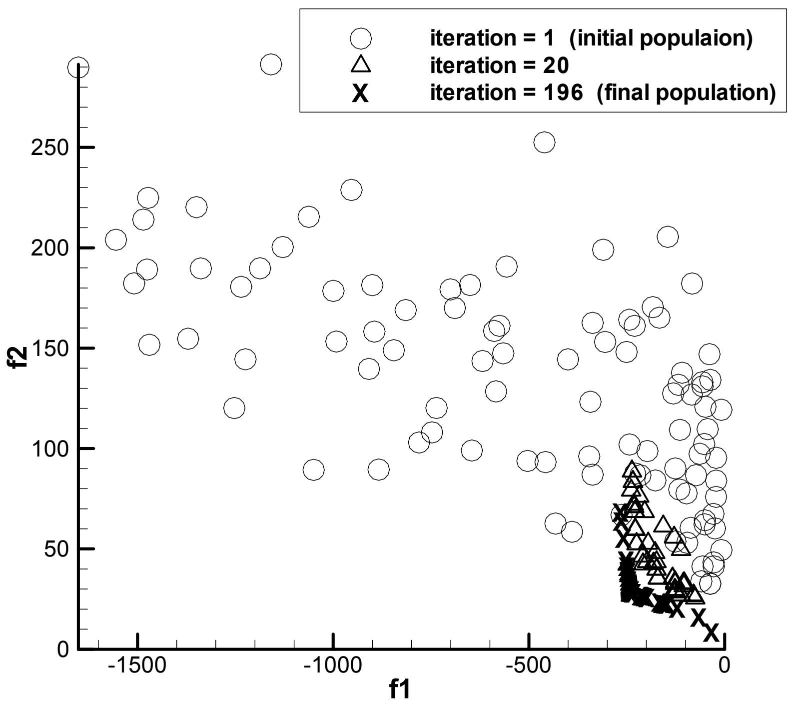


(a)



(b)

Figure 21 Progress of solutions towards the final computed Pareto front for Osyczka and Kundu (1995) problem



4 Conclusions

Application of optimisation models to real life systems whether engineering/scientific systems or financial systems, demands efficient optimisation algorithms that are simplistic in execution, provide reliable solutions and are computationally inexpensive. The MPP algorithm provides one such means of searching for optimal solutions. This multi-objective algorithm, with added constraint handling capacity, has been tried and thoroughly validated against test problems of different types. The pertinent analysis results show that MPP is competent in producing reliable solutions, and for certain cases even does better than well known algorithms presently available in literature. Performance of the constraint handling technique in driving solutions into the feasible domain at the expense of a reasonable number of function evaluations is also appreciable. MPP needs to be tested on multiple and varied practical problems to assess its capability in dealing with complex real world problems.

MPP employs the concept of weighted sum of objectives without any normalisation of the objectives, which leads to relatively poor distribution of Pareto solutions in certain complex multi-objective cases. Nevertheless, the inclusion of the concept of sectional convergence using biased weighing of objectives and careful hypercube sizing ensures a desirable distribution of the Pareto solutions even for these poorly behaved cases.

The MPP algorithm presents a concordant application of the basic traits of evolutionary algorithms, classical weighed sum approach and certain ingenious techniques such as sectional convergence, hypercube operator, epidemic operator, etc. to constrained and unconstrained multi-objective problems. A combination of such distinct features is rare in optimisation literature and provides a foundation to construct robust composite optimisation algorithms with features adaptive to both the problem and the progress of the algorithm through the function space towards the Pareto front.

References

Binh, T.T. and Korn, U. (1997) 'MOBES: a multiobjective evolution strategy for constrained optimization problems', *Proc. of The Third International Conference on Genetic Algorithms (Mendel 97)*, Brno, Czech Republic, pp.176–182.

Chowdhury, S. and Dulikravich, G.S. (2009) 'Improvements to single-objective constrained predator-prey evolutionary optimization algorithm', *Structural and Multidisciplinary Optimization*, (to appear).

Chowdhury, S., Dulikravich, G.S. and Moral, R.J. (2009) 'Modified predator-prey (MPP) algorithm for constrained multi-objective optimization', in Burczynski, T. (Ed.): *Proceedings of EUROGEN 2009*, Cracow, Poland, 15–17 June 2009.

Chowdhury, S., Moral, R.J. and Dulikravich, G.S. (2008) 'Predator-prey evolutionary multi-objective optimization algorithm: performance and improvements', in Toropov, V. (Ed.): *Proceedings of 7th ASMO-UK/ISSMO International Conference on Engineering Design Optimization*, Bath, UK, 7–8 July 2008.

Coello, C.A.C., Veldhuizen, D.A.V. and Lamont, G.B. (2002) *Evolutionary Algorithms for Solving Multi-objective Problems*, Kluwer Academic Publishers.

Deb, K. (2002) *Multi-objective Optimization Using Evolutionary Algorithms*, Wiley & Sons, Chichester, UK.

Deb, K. and Rao, U.N. (2005) 'Investigating predator-prey algorithms for multi-objective optimization', *KanGAL Report Number 2005010*, IIT Kanpur, India.

Deb, K., Pratap, A., Agarwal, S. and Meyarivan, T. (2002) 'A fast and elitist multiobjective genetic algorithm: NSGA-II', *IEEE Transactions on Evolutionary Computation*, Vol. 6, No. 2, pp.182–197.

Deb, K., Thiele, L., Laumanns, M. and Zitzler, E. (2006) 'Evolutionary multiobjective optimization', *Advanced Information and Knowledge Processing*, pp.105–145, Springer, Berlin Heidelberg.

Eshelman, L.J. and Schaffer, J.D. (1993) 'Real coded genetic algorithms and interval schemata', *Foundations of Genetic Algorithms 2 (FOGA 2)*, pp.187–202.

Fonseca, C.M. and Fleming, P.J. (1995) 'An overview of evolutionary algorithms in multiobjective optimization', *Evolutionary Computation*, Vol. 3, pp.1–16.

Grimme, C. and Schmitt, K. (2006) 'Inside a predator-prey model for multiobjective optimization – a second study', *Proc. Genetic and Evolutionary Computation Conf. (GECCO 2006)*, Seattle WA, ACM Press, New York, pp.707–714.

IOSO NM Version 1.0 – User Guide (2003) IOSO Technology Center, Moscow, Russia.

Knowles, J. and Corne, D. (2007) 'Quantifying the effects of objective space dimension in evolutionary multiobjective optimization', *Evolutionary Multi-Criterion Optimization*, LNCS 4403, pp.757–771.

Laumanns, M., Rudolph, G. and Schwefel, H.P. (1998) 'A spatial predator-prey approach to multi-objective optimization: a preliminary study', *Proceedings of the Parallel Problem Solving from Nature*, Vol. 5, pp.241–249.

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

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

Moral, R.J. and Dulikravich, G.S. (2008) 'Multi-objective hybrid evolutionary optimization with automatic switching among constituent algorithms', *AIAA Journal*, Vol. 46, No. 3, pp.673–700.

Osyczka, A. and Kundu, S. (1995) 'A new method to solve generalized multicriteria optimization problems using the simple genetic algorithm', *Structural Optimization*, Vol. 10, pp.94–99.

Silva, A., Neves, A. and Costa, E. (2002) 'An empirical comparison of particle swarm and predator-prey optimization', *Lecture Notes in Computer Science*, Vol. 2464, pp.103–110.

Sobol, M. (1976) 'Uniformly distributed sequences with an additional uniform property', *USSR Computational Mathematics and Mathematical Physics*, Vol. 16, pp.236–242.

Zitzler, E., Deb, K. and Thiele, L. (2000) 'Comparison of multiobjective evolutionary algorithms: empirical results', *Evolutionary Computation*, Vol. 8, No. 2, pp.173–195.

Appendix A Performance of PP algorithm by Deb and Rao (2005) on standard test cases

Figure 22 Test problem ZDT 1 using PP

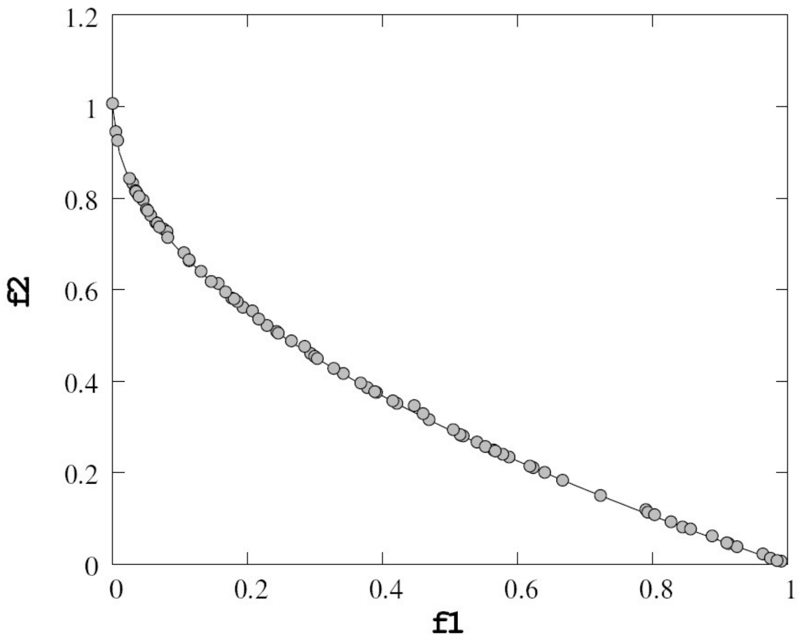


Figure 23 Test problem ZDT 2 using PP

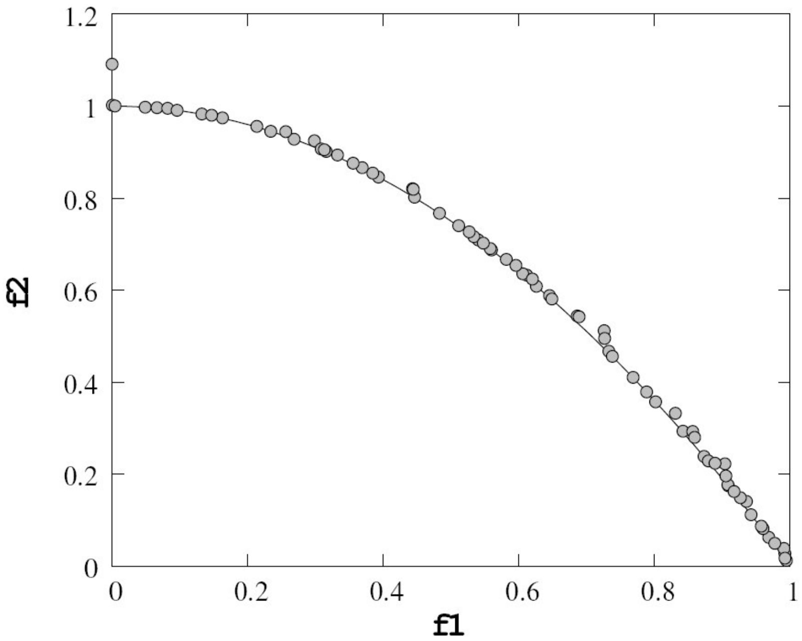


Figure 24 Test problem ZDT 3 using PP

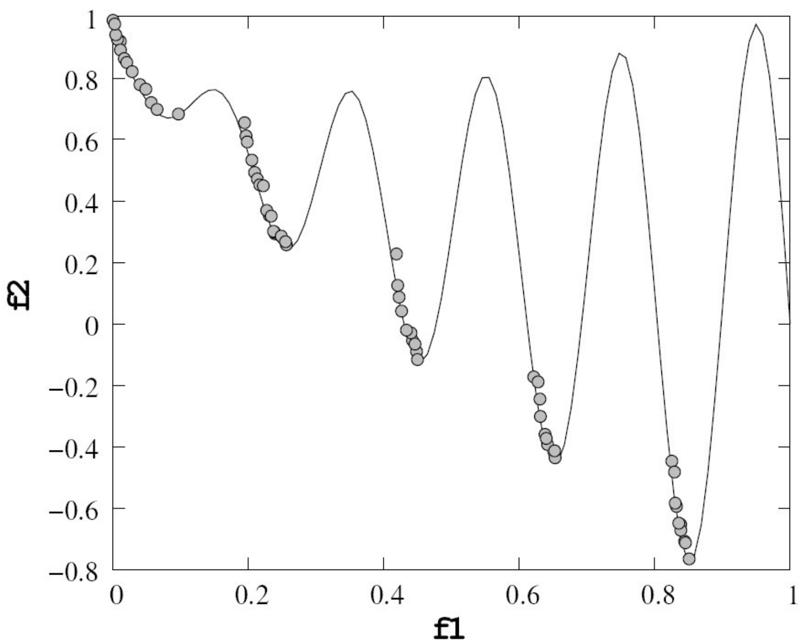


Figure 25 Test problem ZDT 4 using PP

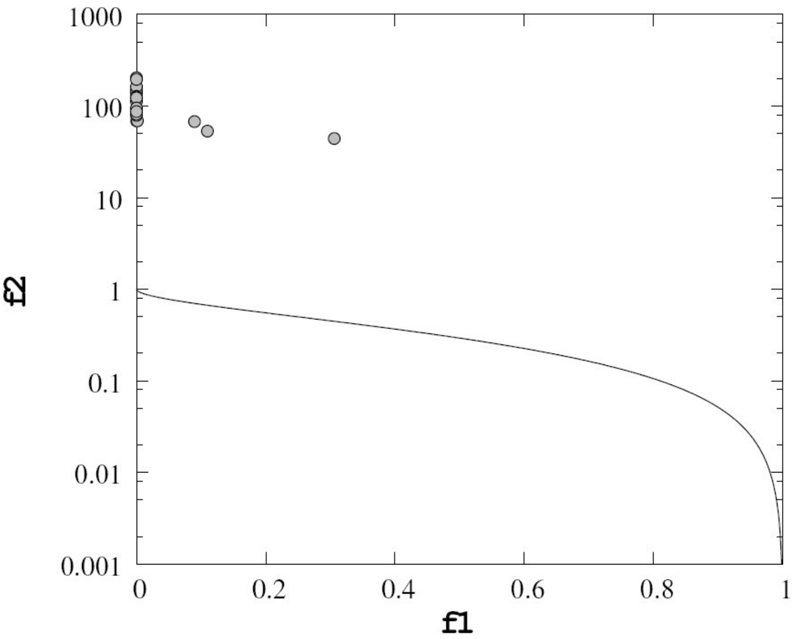


Figure 26 Test problem ZDT 6 using PP

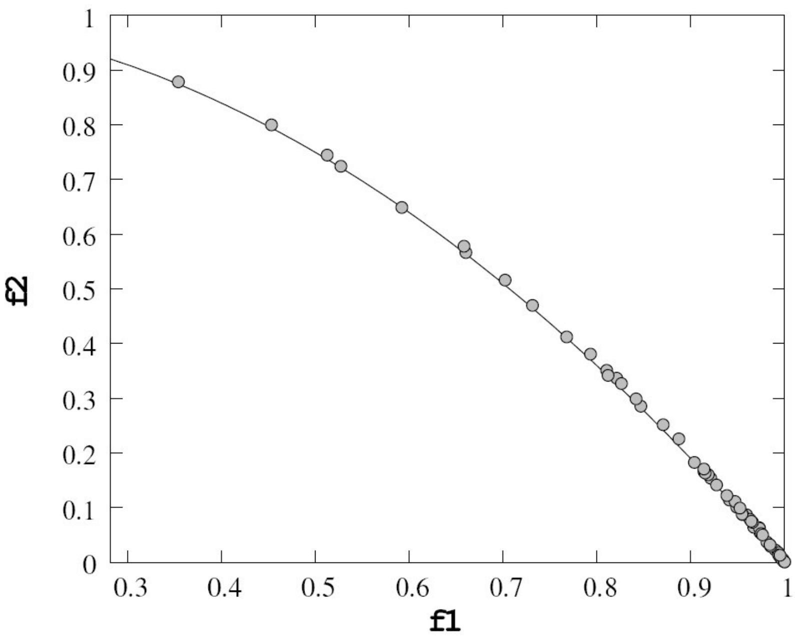


Figure 27 Test problem DTLZ 2 using PP