A Bi-objective Constrained Optimization Methodology Using a Hybrid Multi-Objective and Penalty Function Approach

Rituparna Datta
Department of Mechanical Engineering,
Indian Institute of Technology Kanpur
Kanpur, PIN: 208018, INDIA
Email: rdatta@iitk.ac.in

Kalyanmoy Deb
Department of Electrical & Computer Engineering,
Michigan State University,
East Lansing, MI 48824, USA
Email: kdeb@egr.msu.edu

COIN Report Number 2015007

Abstract

Single objective evolutionary constrained optimization has been widely searched and researched by plethora of researchers in last two decades. On the other hand, multi-objective constraint handling using evolutionary algorithms has not been actively proposed. However, real-world multi-objective optimization problems consist of one or many non-linear and non-convex constraints. In the present work, we develop an evolutionary algorithm based constraint handling methodology, to deal with constraints in multi-objective optimization problems. The method is a combination of an evolutionary multi-objective optimization coupled with classical weighted sum approach based local search method and is an extended version of our previously developed constraint handling method for single objective optimization [4]. A constrained bi-objective problem is converted into a tri-objective problem where the additional objective is formed using summation of constrained violation. The proposed method is applied to four constrained multi-objective problem. The non-dominated solutions are compared with a standard evolutionary multi-objective optimization algorithm (NSGA-II) with respect to hypervolume and attainment surface. The simulation results illustrates the effectiveness of the proposed approach.

1 Introduction

Single objective optimization algorithms with constraints have been efficiently solved using Evolutionary Algorithms (EAs) during last two decades. However, real-world optimization problems consist of several objectives. Multi-objective optimization considers the objectives as a vector of multiple objectives rather than a single objective function and the solution is not unique but a set of solutions which are also known as Pareto-optimal solutions. To be specific, such a set is obtained in cases where the objectives are in conflict with each other. In most of the situations, conventional optimization techniques cannot handle such complexities well due to the involvement of non-linear and non-convex constraints and objective functions as a result poorly performs to produce a satisfactory solution. Evolutionary algorithms (EAs) are flexible, derivative free and an alternative way to tackle these non-linear and non-convex optimization efficiently. Some state-of-the-art evolutionary algorithm based multi-objective optimization methods are Pareto Archived Evolution Strategy (PAES) [7], Non-dominated sorting genetic algorithm-II (NSGA-II) [3], Strength Pareto Evolutionary Algorithm (SPEA-II) [15] and Multi Objective Evolutionary Algorithm based on Decomposition (MOEA/D) [14]. The foundation of these methods are deeply routed in population based methods that are borrowed from statistics and are combined with nature inspired methods that lend them robustness properties. Not relying on gradient information and continuity of the describing functions, these methods often provide an attractive alternative to solve real world problems. With an increase in available computational power and a move towards parallel and high performance computing we expect increased use of these methods in all sorts of problems in science and engineering.

The main focus by most researchers in constraint handling was the development of efficient constraint handling methods for single objectives. We have developed a bi-objective way to handle constraints in single objective optimization [4]. The method is a combination of bi-objective approach integrated with classical penalty function method. The bi-objective method was used to estimate the penalty parameter and to supply initial seed to the start the penalty function based local search. The classical optimization was used to ensure convergence. Constrained multi-objective optimization is not as popular as single objective constraint handling [9, 12, 8]. Wright and Loosemore [11] proposed a multi-objective constraint handling method where constraint violations is taken as "infeasibility objective". This method in turn give rise to a new objective which help in handling the constraints efficiently. This concept is extended in [5, 9, 12] to solve the multi-objective constrained optimization problems. Young [13] proposed a constrained multiobjective EA by extending the idea of blended space [1]. Wef and Wang [10] proposed an infeasible elitist based particle swarm optimization to handle constraints in multi-objective optimization. Constraint handling is carried out in an evolutionary way in which diversity is ensured by partitioning of search space [6].

Handling constraints in multi-objective optimization is much more challenging than that of single objective one. In multi-objective optimization, in addition to assure convergence and diversity, all Pareto-optimal solutions must satisfy all inequality and/or equality constraints in order to call a feasible solution. The proposed multi-objective constraint handling methodology is an extended version of our single objective one [4], in higher dimensions. In the present work, we restrict ourselves in solving bi-objective constraint handling problems. In next section, we briefly explain the details of the proposed approach for handling constraints in multi-objective optimization. The proposed approach is first applied to a problem constructed by us in this paper. We also test the efficiency of the methodology in a number of test problems [2]. Results shows that the proposed multi-objective constraint handling method is efficient compared to state-of the art multi objective evolutionary algorithm.

2 Multi-Objective Approach

We are interested in solving the following M-objective constrained optimization problem:

$$\begin{array}{rll} \text{Minimize} & \{f_1(\mathbf{x}), f_2(\mathbf{x}), \ldots, f_M(\mathbf{x})\}, & \\ \text{Subject to} & g_j(\mathbf{x}) \geq 0, & j = 1, 2, \ldots, J, \\ & h_k(\mathbf{x}) = 0, & k = 1, 2, \ldots, K. \\ & x_i^{(L)} \leq x_i \leq x_i^{(U)}, & i = 1, 2, \ldots, n. \end{array} \tag{1}$$

We now define a constraint violation function $CV(\mathbf{x})$, as follows:

$$CV(\mathbf{x}) = \sum_{j=1}^{J} \langle \bar{g}_j(\mathbf{x}) \rangle + \sum_{k=1}^{K} |\bar{h}_k(\mathbf{x})|. \tag{2}$$

Here, $\bar{g}_j$ and $\bar{h}_k$ are normalized constraint functions.

We shall now consider a conversion of the above problem (Equation 1) into a $(M+1)$-dimensional minimization problem, as follows:

$$\begin{array}{rl} \text{Minimize} & \{f_1(\mathbf{x}), f_2(\mathbf{x}), \ldots, f_M(\mathbf{x}), CV(\mathbf{x})\}, \\ \text{Subject to} & x_i^{(L)} \leq x_i \leq x_i^{(U)}, \quad i = 1, 2, \ldots, n. \end{array} \tag{3}$$

Since evolutionary optimization (EO) algorithms handles variables bounds naturally in its initialization procedure and also in subsequent genetic operations so that no solution violates the specified variable bounds, the solution to the above problem using an EO reduces the above problem to a unconstrained multi-objective optimization problem. An evolutionary multi-objective or many-objective optimization (EMO) procedure can be applied to solve the above $(M+1)$-objective problem.

Let us say that a set of K non-dominated points are obtained using an EMO procedure. We suggest the following step-by-step procedure to solve the original multi-objective constrained minimization problem given in Equation 1.

3 Multi-objective Constraint Handling Algorithm:

Step 1: Find a set of (say K) non-dominated points of the $(M+1)$-objective problems with variable bounds given in Equation 3 after τ generations of a suitable EMO procedure. Since we are interested in solutions close to the Pareto-optimal front of the original M-objective problem, a constraint on $CV(\mathbf{x}) \leq \epsilon$ can be added in the NSGA-II application. we have used $\epsilon = 0.1J$ in all our studies (where J is the number of constraints).

Step 2: A M-dimensional surface is fitted through these points to obtain CV-objective as a function of M original objectives: $CV(f_1, f_2, \ldots, f_M)$. The fitted surface and the obtained NSGA-II solutions for a hypothetical two-objective scenario are shown schematically in Figure 1(a).

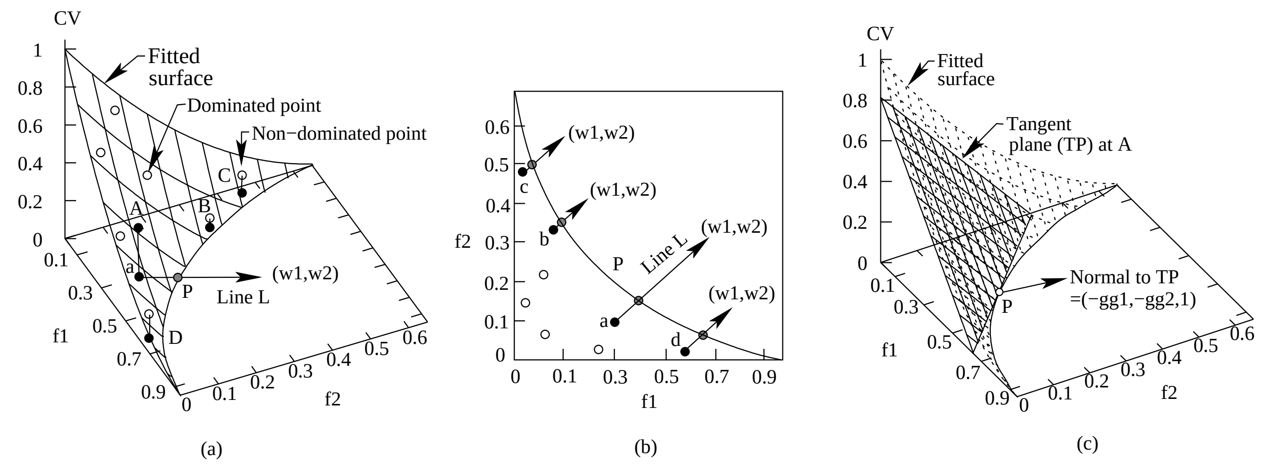

Figure 1: (a) Fitted surface and intersection with a line of improvement is shown. (b) The non-dominated points are identified on a projected f_1-f_2 plane. (c) For a particular point A, the corresponding penalty surface is shown.

Step 3: All K points are considered for their original M objective values and non-dominated points are found with an opposite sense of their optimization types. For example, if all M objectives were to be minimized, then here non-dominated solutions corresponding to *maximization* of all objectives should be found. This process is illustrated in Figure 1(b). These process should identify points that are closer to the Pareto-optimal front of the original M-objective problem. Let us say that there are H such points that are non-dominated.

Step 4: For each of the H non-dominated point (say $\mathbf{z}$), Steps 4 and 5 are performed. First, the intersecting point of a line from the point $\mathbf{z}$ towards a specified direction $\mathbf{w} = (w_1, w_2, \ldots, w_M)^T$ with the fitted surface is obtained. This operation is illustrated in Figure 1(a). Let us say that the intersecting point is P_z. We suggest using $w_i = 1/(f_i^{\max} - f_i^{\min})$ here, where $f_i^{\min}$ and $f_i^{\max}$ are minimum and maximum objective value of i-th objective of all H non-dominated solutions.

Step 5: The gradient vector of the fitted surface is obtained at the point P_z and the corresponding penalty function is obtained as follows:

$$P(\mathbf{x}) = \sum_{i=1}^{M} \left(-\left.\frac{\partial CV}{\partial f_i}\right|_{P_z} \right) f_i(\mathbf{x}) + k \times CV(\mathbf{x}), \tag{4}$$

where k is a parameter greater than or equal to one. We choose $k = 2$ for our study here. Figure 1(c) shows the above penalty function hyper-plane passing through P_z. Since the hyper-plane is the tangent to the fitted surface at P_z, at this point the $P(\mathbf{x})$ value will be minimum. Thus, we minimize the above penalty function starting from solution $\mathbf{z}$ using a classical optimization algorithm (say, using `fmincon()` routine of Matlab) to obtain a solution $\mathbf{y}$. The solution $\mathbf{y}$ is introduced in the EMO population for further processing.

The parameter $k = 1$ makes the right combination of weights for the penalty function given in Equation 4 to be ideally used to find the corresponding Pareto-optimal solution. However, to ensure getting close to the

Pareto-optimal front of the M-objective problem, we suggest using a value larger than one. We have used $\tau = 5$ and value of k is taken as 2 (to have the scale factor).

This method is similar in principle to the bi-objective constrained handling method described elsewhere [4], which we discuss next.

3.1 Similarity with Previous Single-objective Method

When $M = 1$ (that is, solving a single-objective constraint handling problem), Step 1 solves a bi-objective minimization problem (f and CV). In Step 2, a fitted curve of CV as a function of f is obtained. In the earlier constraint handling study, f versus CV was obtained. Step 3 will choose a single solution having the highest f value for a minimization problem (which will correspond to the smallest CV value). Thus, there is only one solution of interest ($H = 1$) for single-objective optimization and Steps 4 and 5 will be executed only once. Step 4 is trivial in this case, as it means identifying the intersecting point P of the fitted curve with line $CV = 0$. In Step 5, the gradient of the fitted curve $CV(f)$ is obtained at the intersecting point P as $gg = \frac{dCV}{df}|_P$ and using Equation 4, the following penalty function is formed:

$$P(\mathbf{x}) = (-gg)f(\mathbf{x}) + k \times CV(\mathbf{x}).$$

Optimizing above function is equivalent to optimizing the following function:

$$\begin{aligned} P(\mathbf{x}) &= f(\mathbf{x}) + k\frac{1}{-gg} \times CV(\mathbf{x}), \\ &= f(\mathbf{x}) + k\left(-\left.\frac{df}{dCV}\right|_P\right) \times CV(\mathbf{x}). \end{aligned} \quad (5)$$

In the earlier single-objective constraint handling algorithm, the penalty parameter $R = -\frac{df}{dCV}$ was computed at the $CV = 0$ solution of the fitted curve $f(CV)$. Thus, we observe that the multi-objective constraint handling approach proposed above degenerates to the single-objective constraint handling algorithm proposed in the earlier study [4].

4 Procedure

The idea for two-objective problems are as follows. First, from NSGA-II with clustering approach, non-dominated solutions are obtained. Then, the following quadratic surface is fitted:

$$CV(z_1, z_2) = a_0 + a_1 z_1 + a_2 z_2 + z_3 z_1^2 + a_4 z_2^2 + a_5 z_1 z_2.$$

It is likely that a non-dominated point (say A: $(\bar{z}_1, \bar{z}_2)$) is infeasible to the original problem and we would like to move in a direction $d = (w_1, w_2)$ in order to reach the Pareto-optimal front of two-objective problem. Any point from A towards d can be written as follows:

$$\begin{aligned} z_1 &= \bar{z}_1 + \alpha w_1, \\ z_2 &= \bar{z}_2 + \alpha w_2. \end{aligned}$$

Now, intersecting the line from A towards d with the fitted surface will take us to the non-dominated front dictated by the fitted surface. The intersection point can be found as follows. In the pseudo-code, `z1`=$\bar{z}_1$ and `z2`=$\bar{z}_2$ and `root` is α.

```
A = a3*w1*w1+a4*w2*w2+a5*w1*w2;
B = a1*w1+a2*w2+2.0*a3*z1*w1+2.0*a4*z2*w2
    +a5*(z1*w2+z2*w1);
C = a0+a1*z1+a2*z2+a3*z1*z1+a4*z2*z2
    +a5*z1*z2;
discr = B*B-4.0*A*C;
if (discr>=0.0)
{
  root1 = (-B+sqrt(discr))/(2.0*A);
  root2 = (-B-sqrt(discr))/(2.0*A);
```

```
  if (root1>0.0 && root2<0.0)
    root = root1;
  else if (root2>0.0 && root1<0.0)
    root = root2;
  else if (root1>0.0 && root2>0.0)
    root = (root1<root2) ? root1 : root2;
  else if (root1<0.0 && root2<0.0)
    {printf("No root possible.\n");
     exit(-1);}
  // compute the intersection point
  z1 = z1 + root * w1;
  z2 = z2 + root * w2;
}
  CV = a0+a1*z1+a2*z2+a3*z1*z1+a4*z2*z2
        +a5*z1*z2;
  grad1 = (a1+2.0*a3*z1+a5*z2);
  grad2 = (a2+2.0*a4*z2+a5*z1);
```

If there is an intersection point (`disrc<0.0`), the gradients are computed at the intersection point, else the original point $z_1 = \bar{z}_1$ and $z_2 = \bar{z}_2$ are used to compute gradients. In the above pseudo-code, z_1 and z_2 are not updated, if `discr<0.0`.

Now, the penalized function is as follows:

$$P(x) = \texttt{(-grad1)} * f_1(x) + \texttt{(-grad2)} * f_2(x) + k * CV(x).$$

5 An Example

Let us say that after fitting eight points obtained by NSGA-II, we have obtained the fitted surface as:

$$CV(z_1, z_2) = 1 - 2z_1 - 2.5z_2 + z_1^2 + 1.5z_2^2.$$

The surface is shown in Figure 1(a).

Projection of eight points on the f_1-f_2 plane are shown in Figure 1(b). At this plane, we identify the non-dominated points by considering maximization of f_1 and f_2. There are four non-dominated points shown in filled circles and four dominated points shown in open circles.

Let us consider one of the non-dominated points (say A =(0.3, 0.1, 0.255)). Its projection on the f_1-f_2 is shown as the point a (=$(z_1, z_2) = (0.3, 0.1)$). Our goal is to now move along a direction (w_1, w_2) in order to reach the fitted surface. This is because if the fitted surface correspond to the true PO surface of the three-objective problem, the intersecting point would have been the corresponding PO point of the two-objective (f_1 and f_2) problem.

Here, we use $w_1 = w_2 = 1$ (equal weight for both objectives). Using the supplied code, we obtain the intersecting point $P = (0.374709, 0.174709)$. The gradients of the fitted surface at this point are given below: $\frac{\partial CV}{\partial f_1} = -1.250581$, $\frac{\partial CV}{\partial f_2} = -1.975872$. Thus, the penalty function that we should minimize from the initial point a ($= (0.3, 0.1)$) is as follows:

$$P(x) = 1.250581 f_1(x) + 1.975872 f_2(x) + kCV(x).$$

We can choose $k = 1$ or 2. The penalty surface with $k = 1$ is shown in Figure 1(c). The penalty surface (linear hyper-plane) is tangent at P. This is similar in principle to the single-objective constraint handling strategy we have used before.

Some important issues are:

1. The above procedure can also be extended for more than two-objective optimization problems.
2. The choice of w-vector can arbitrary. We have used equal weights for the present study.
3. For fitted surfaces where there is no intersecting point P, we use the original point A to compute the gradient vector and form the respective penalty function.

5.1 Local Search

To select points for local search we neglect the modified objective ($CV(\mathbf{x})$) and maximize the other two objectives (original objective objectives), from the three objective non-dominated solutions. Non-dominated solutions are identified from the maximization of original two objectives. Figure 2 shows the non-dominated solutions in maximization sense. Here modified objective function ($CV(\mathbf{x})$) is neglected. We use each non-dominated solution from two-objective maximization, as the initial solution to start the local search.

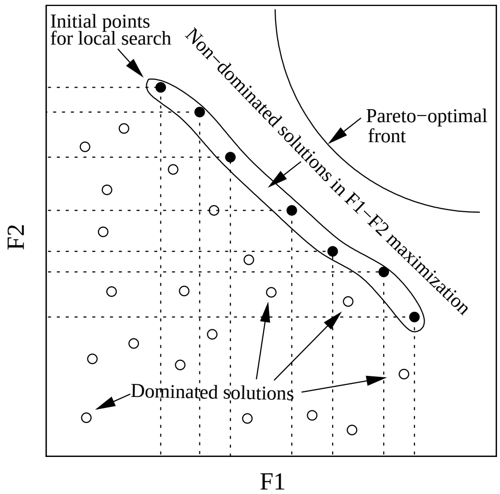

Figure 2: Two objective maximization

6 Diversity Improvement

We maintain an external population of local searched points, which are the non-dominated solutions with respect to original problem given in Equation 1. Figure 3 shows the non-dominated solutions in the external population. After the execution of local search, there is a chance of loosing diversity in the Pareto-optimal solutions. To ensure diversity, we divide X-axis in a number of equal divisions from ideal point to nadir point. Then each division is checked, if there exist any non-dominated solutions. Thus if any division is empty, we perform local search from the non-dominated points with respect to the modified problem given in Equation 3.

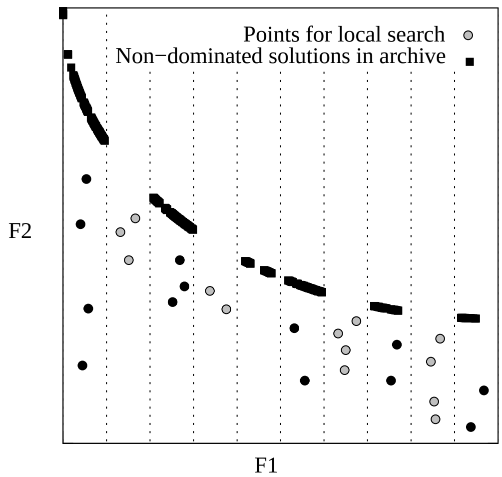

Figure 3: Diversity Improvement in original objective space.

7 Simulation Results on Standard Test Problems

We first test our proposed methodology in a test problem constructed by us. Then, we apply of our approach on some standard multi-objective constrained optimization test problems taken from literature [2].

For every test problem, the algorithm is tested for 11 times with different initial populations. Population size is taken as 100 (unless stated otherwise) and other parameters are as follows:

SBX probability= 0.9,
SBX index = 10,
polynomial mutation probability = $1/n$,
and mutation index = 100.

The algorithm is terminated when the function evaluations exceed $50,000$. Now the proposed strategy is applied in some of the constrained multi-objective test problems.

7.1 Problem MP1

First we solve a bi-objective problem constructed by us:

$$\left.\begin{array}{rl} \text{minimize} & f(\mathbf{x}) = x_1, \\ \text{minimize} & f(\mathbf{x}) = x_2, \\ \text{subject to} & g_1(\mathbf{x}) \equiv x_1^2 + x_2^2 \leq 0, \\ & -2 \leq x_1 \leq 2, \quad -2 \leq x_2 \leq 2. \end{array}\right\} \tag{6}$$

The problem is a two variable multi-objective optimization with a single inequality constraint which is a active. Figure 4 shows the decision variable space for this problem. The active constraint ($g_1(\mathbf{x})$) divide the variable space in two parts. Out of these two parts, one is feasible decision space and the other is infeasible decision space. Figure 4 also indicate that, only the region inside the circle is feasible for this problem. The problem is solvable analytically. We solve the problem analytically and show the Pareto-optimal solutions in Figure 5. From the figure we can see the he Pareto-optimal front is convex.

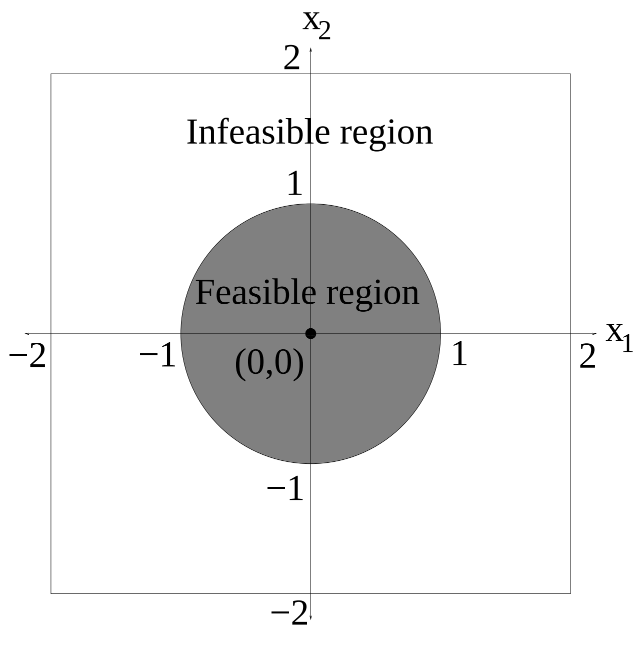

Figure 4: Decision variable space for problem MP1.

We solve the problem using NSGA-II and our proposed method. Comparative study is made with respect to hypervolume and attainment surface measure. Following parameters are used for NSGA-II and proposed hybrid method:

Population size = 40, 100 (for NSGA-II).

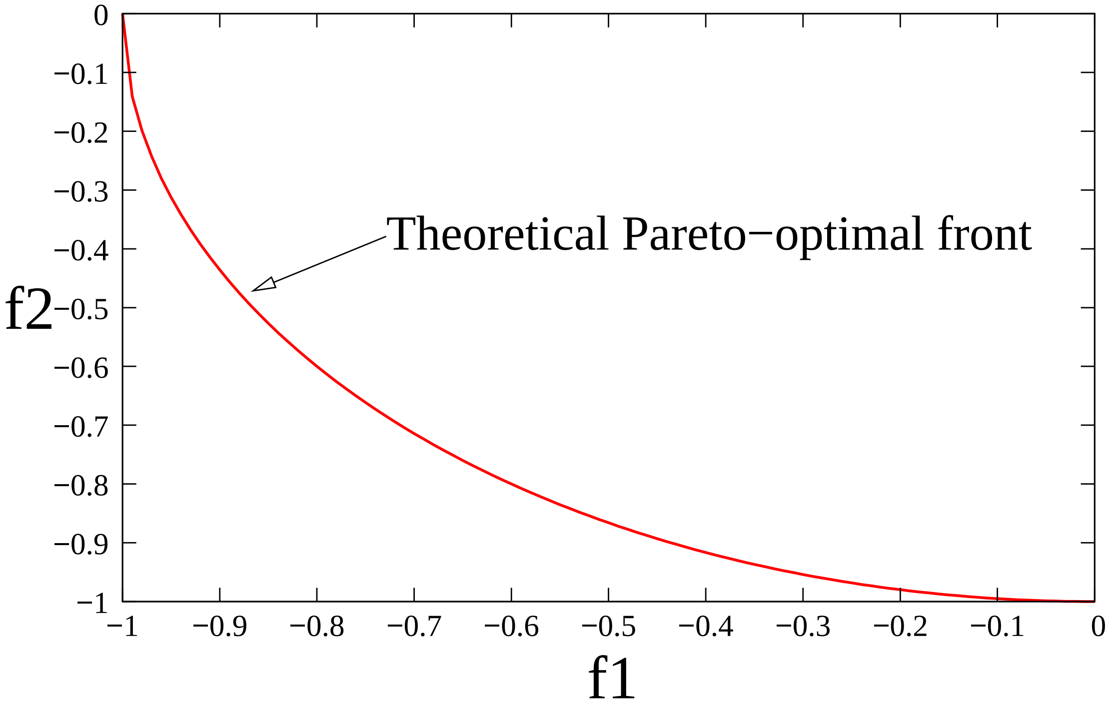

Figure 5: Theoretical Pareto-optimal solutions for problem MP1.

Population size = 40 (for proposed method).

The comparative study is done first with hypervolume, in terms of fixed population of proposed method with different population size of NSGA-II. Figure 6 shows the theoretical hypervolume, hypervolume using NSGA-II and proposed method. In first case population size is taken is 40 in case of of NSGA-II and proposed method. Figure 6 clearly indicate that proposed method outperforms NSGA-II in terms of hypervolume. To get a better hypervolume in case of NSGA-II, we change the population size from 40 to 100 for NSGA-II and keep population size of fixed for proposed method as 40.

From Figure 7, it is evident that, in this case also performance of proposed method is better than NSGA-II. From the hypervolume based comparison of our method and NSGA-II, it can be concluded that out method is better in compare to NSGA-II, even though we take a higher population in NSGA-II.

After comparing with respect to hypervolume, we also compare both the algorithm in term of attainment surface measure. Attainment surface signifies a combination of both convergence and diversity of obtained solutions. Figures 8 and 9 show 0%, 50% and 100% attainment surfaces after $50,000$ function evaluations along with Pareto-optimal front. It can be observed from Figures 8 and 9 that for the proposed procedure shows excellent convergence and diversity.

In both hypervolume and attainment surface measures, our proposed method showed its efficacy by generating a well spread non-dominated solutions which is desirable for ideal multi-objective optimization algorithm.

7.2 Problem BNH

$$\left.\begin{array}{ll}\text{Minimize} & f(\mathbf{x}) = 4x_1^2 + 4x_2^2,\\ \text{Minimize} & f(\mathbf{x}) = (x_1-5)^2 + (x_2-5)^2,\\ \text{subject to} & g_1(\mathbf{x}) \equiv (x_1-5)^2 + x_2^2 \leq 25,\\ & g_2(\mathbf{x}) \equiv (x_1-8)^2 + (x_2+3)^2 \geq 77,\\ & 0 \leq x_1 \leq 5, \quad 0 \leq x_2 \leq 3.\end{array}\right\}$$

This problem is a bi-objective problem with two inequality constraints. We solve the problem using proposed constraint handling method and original NSGA-II. Figure 10 shows the comparison of both NSGA-II and proposed method with respect to hypervolume. Figure 10 indicate that for this problem, both NSGA-II and proposed method are comparable.

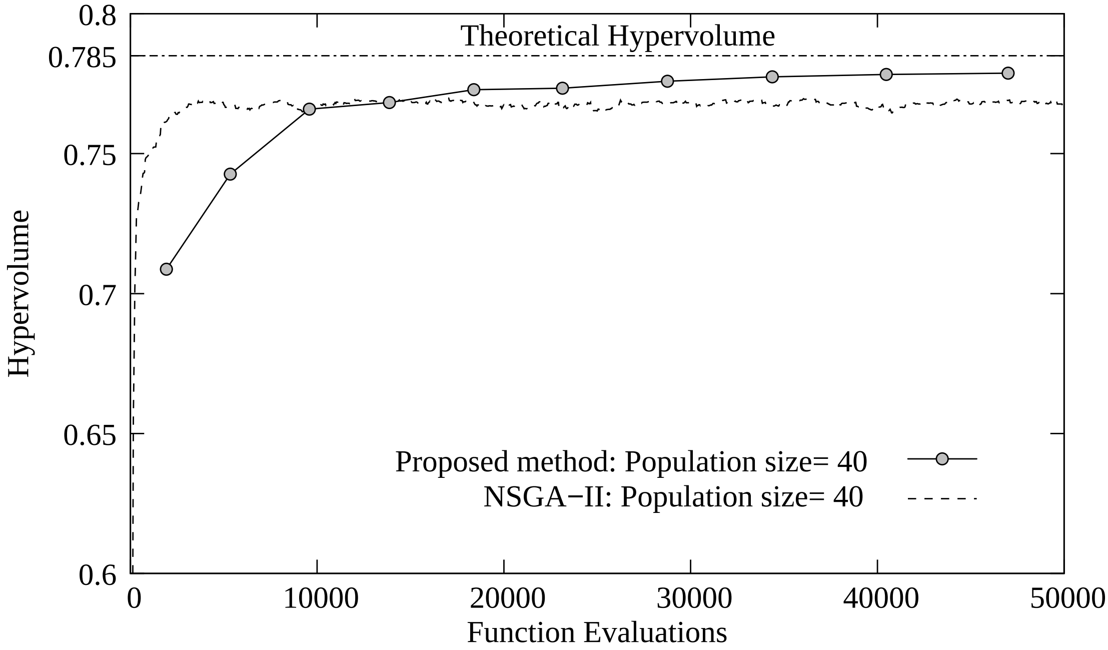

Figure 6: Comparison of proposed method with NSGA-II based on hypervolume for problem MP1.

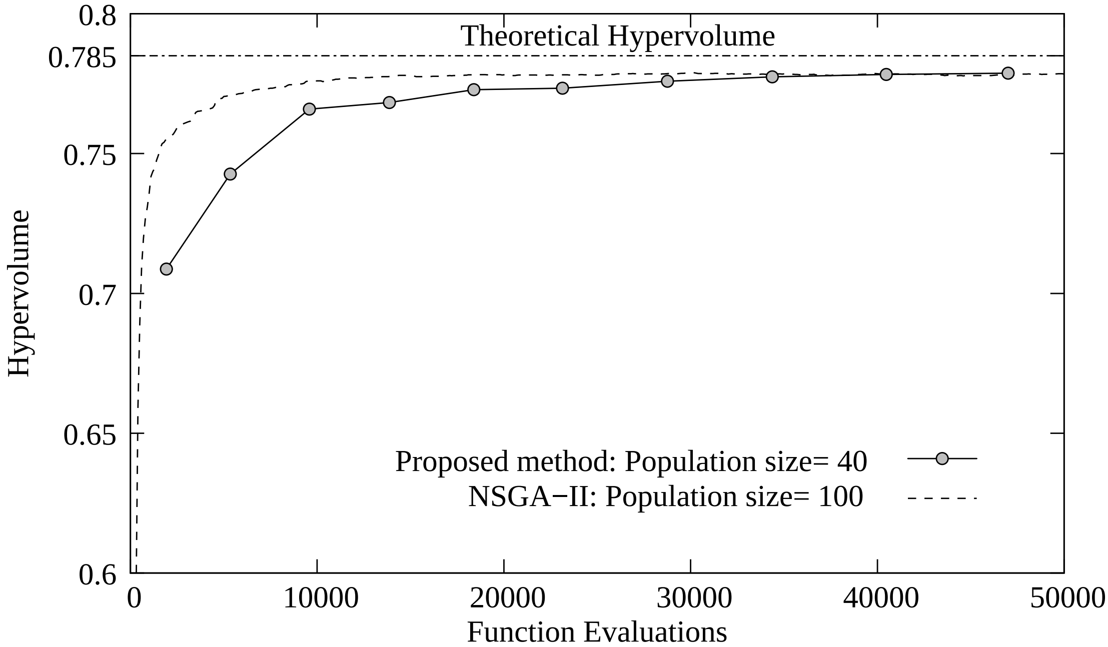

Figure 7: Comparison of proposed method with NSGA-II based on hypervolume for problem MP1

Figures 11 and Figure 12 show 0%, 50% and 100% attainment surfaces after $50,000$ function evaluations along with Pareto-optimal front. In attainment surface measure also performance of the proposed algorithm is similar to NSGA-II.

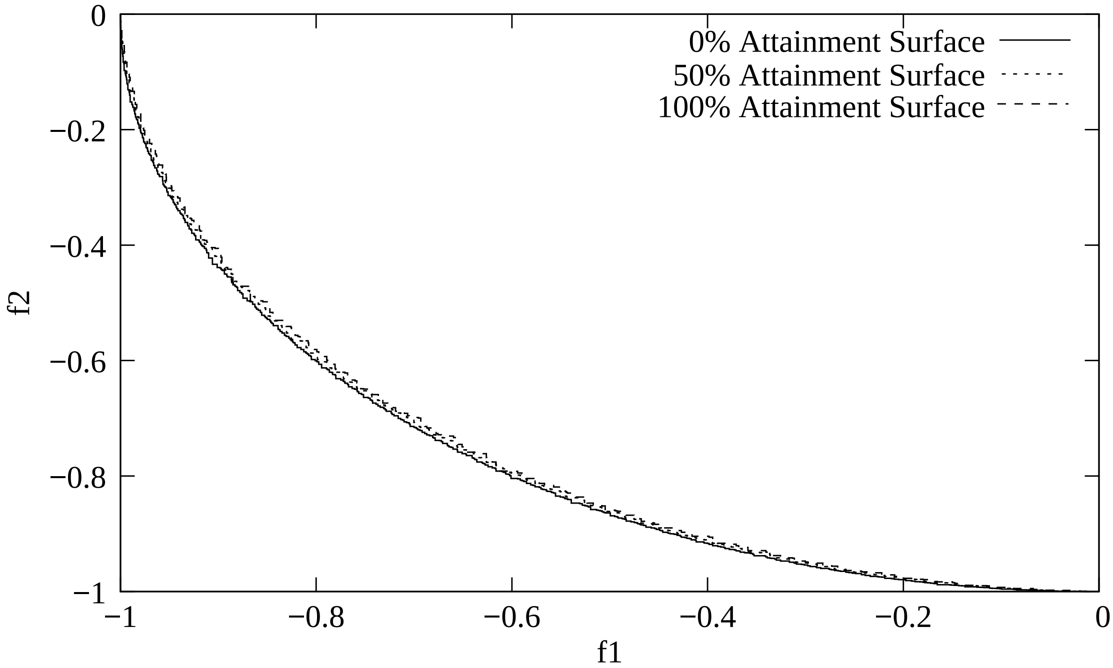

Figure 8: Attainment Surface using NSGA-II for problem MP1.

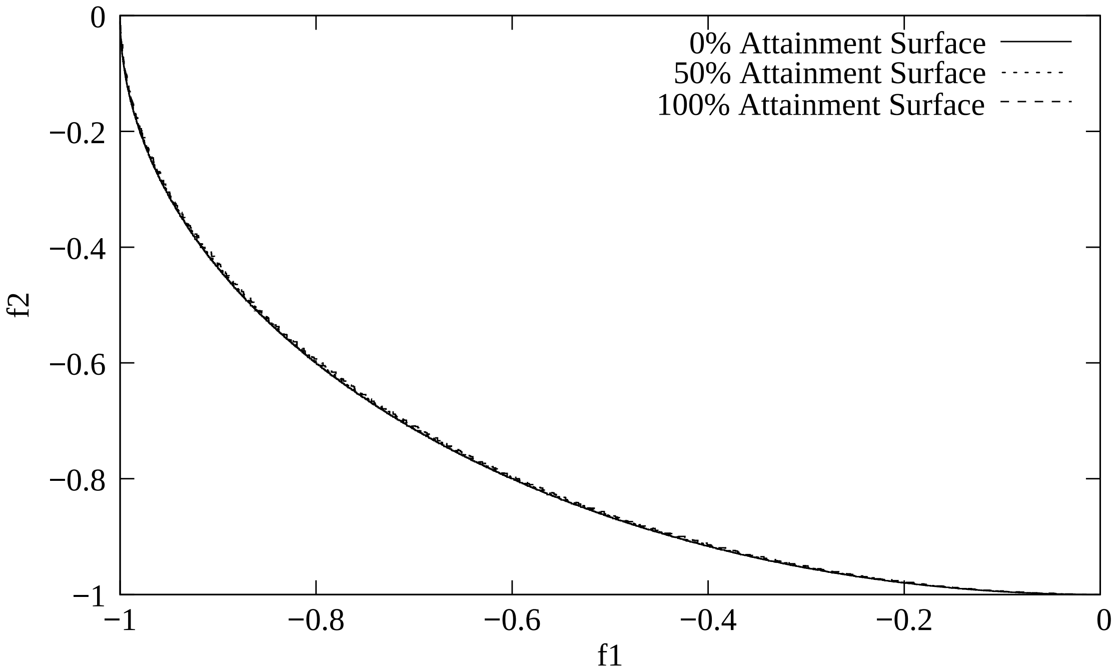

Figure 9: Attainment Surface using proposed approach for problem MP1.

7.3 Problem SRN and Problem CONSTR1

The description of SRN is as follows:

$$\left.\begin{array}{rl}\text{Minimize} & f(\mathbf{x}) = 2 + (x_1 - 2)^2 + (x_2 - 1)^2, \\ \text{Minimize} & f(\mathbf{x}) = 9x_1 - (x_2 - 1)^2, \\ \text{subject to} & g_1(\mathbf{x}) \equiv x_1^2 + x_2^2 \leq 225, \\ & g_2(\mathbf{x}) \equiv x_1 - 3x_2 + 10 \leq 0, \\ & -20 \leq x_1 \leq 20, \quad -20 \leq x_2 \leq 20. \end{array}\right\}$$

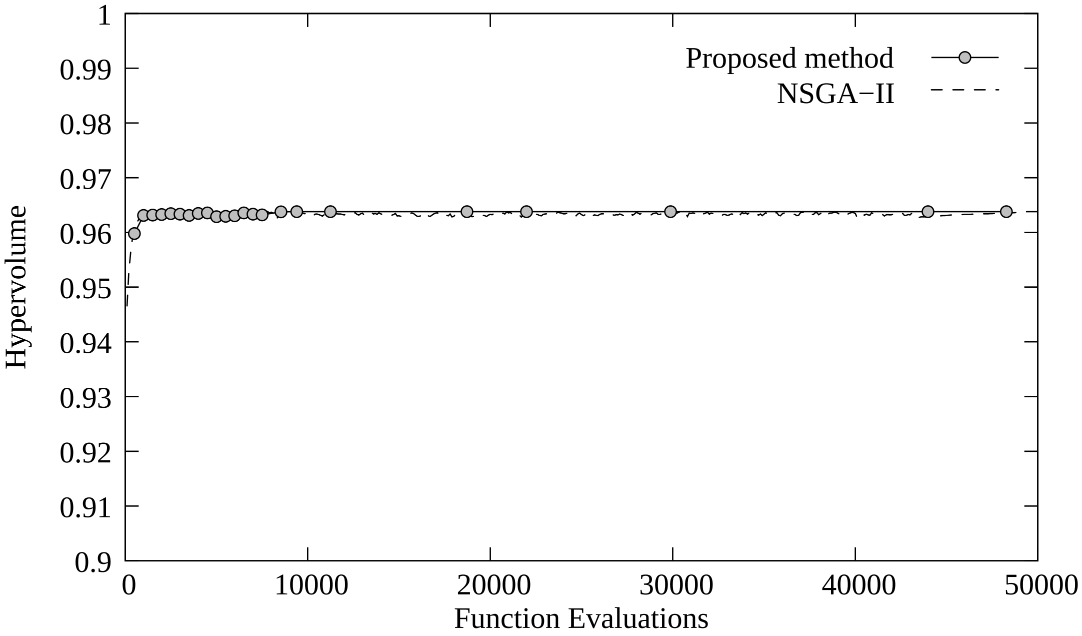

Figure 10: Comparison of proposed method with NSGA-II based on hypervolume for problem BNH.

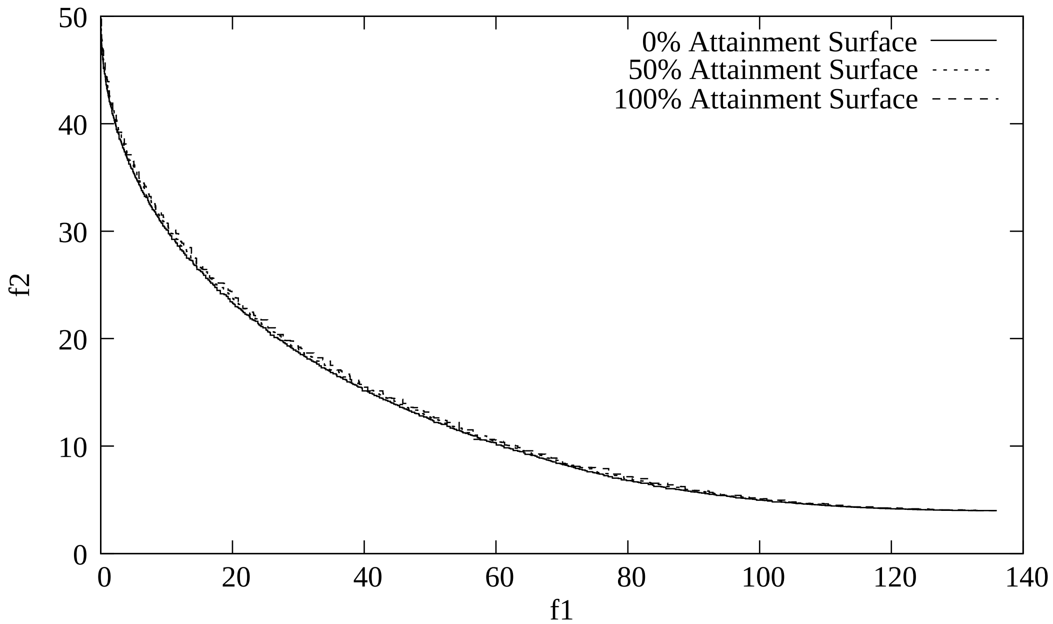

Figure 11: Attainment Surface using NSGA-II for problem BNH.

The problem formulation of CONSTR1 is as follows:

$$\left.\begin{array}{rl}\text{Minimize} & f(\mathbf{x}) = x_1,\\ \text{Minimize} & f(\mathbf{x}) = \dfrac{1+x_2}{x_1},\\ \text{subject to} & g_1(\mathbf{x}) \equiv x_2 + 9x_1 \geq 6,\\ & g_2(\mathbf{x}) \equiv -x_2 + 9x_1 \geq 1,\\ & 0.1 \leq x_1 \leq 1, \quad 0 \leq x_2 \leq 5.\end{array}\right\}$$

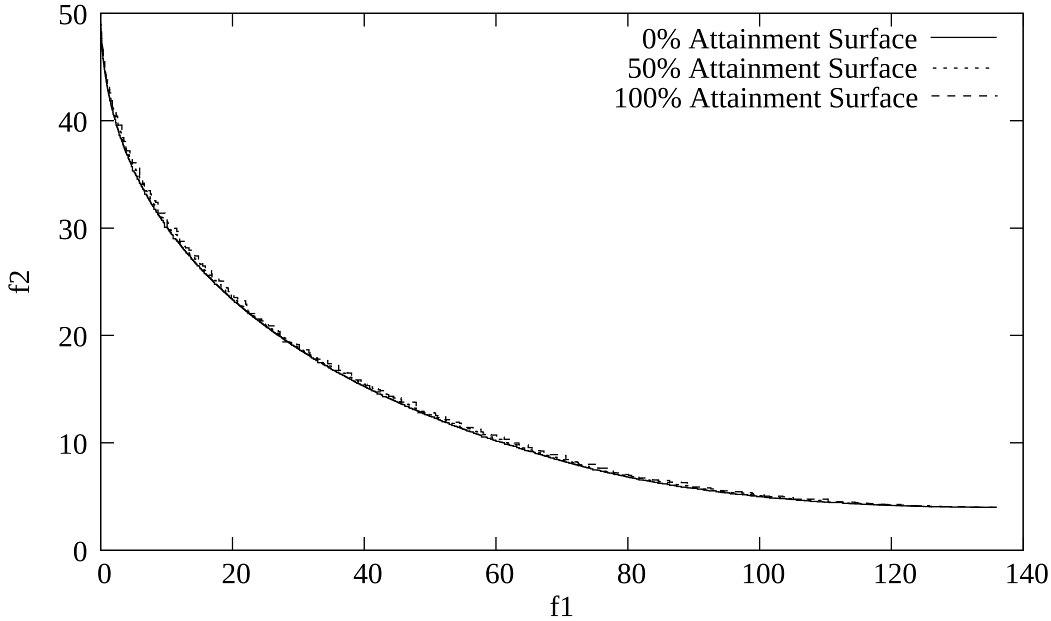

Figure 12: Attainment Surface using proposed approach for problem BNH.

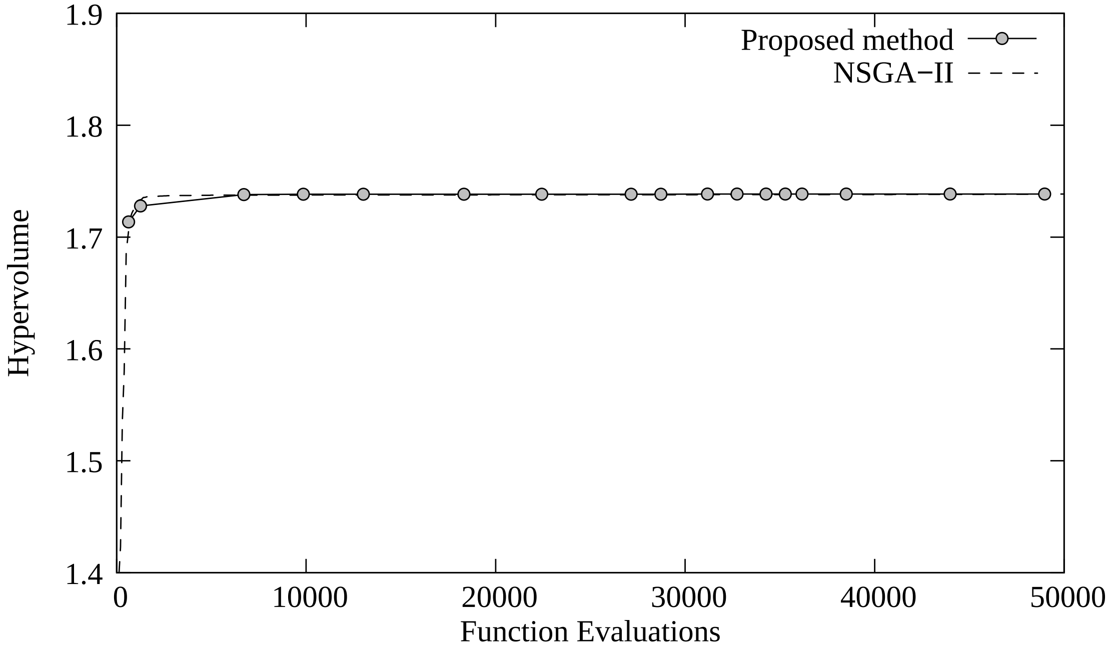

Figure 13: Comparison of proposed method with NSGA-II based on hypervolume for problem SRN.

Both problems SRN and Problem CONSTR1 has two inequality constraints. Figure 13 and Figure 14 indicate that for this problem both NSGA-II and proposed method are comparable in terms of normalized hypervolume measure.

Figures 15 and 16 show 0%, 50% and 100% attainment surfaces after 50,000 function evaluations along with Pareto front. In attainment surface also performance of proposed constraint handling method is alike with NSGA-II.

Figures 17 and 18 depicts 0%, 50% and 100% attainment surfaces after 50,000 function evaluations along with Pareto front for CONSTR1 problem. In terms of attainment surface measure both proposed method is

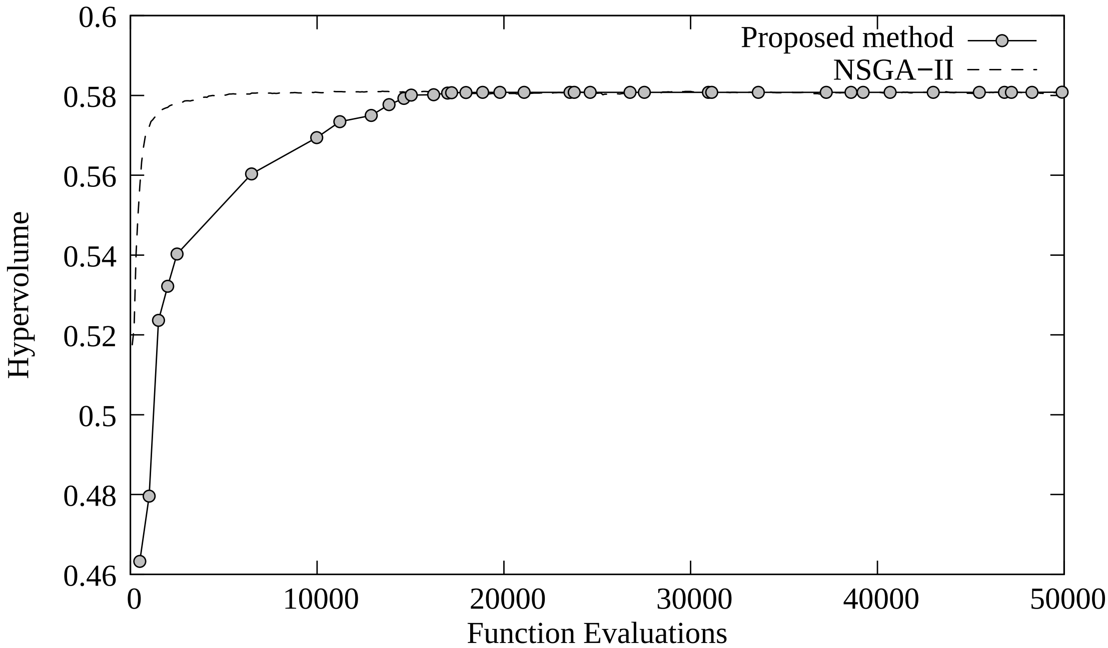

Figure 14: Comparison of proposed method with NSGA-II based on hypervolume for problem CONSTR1.

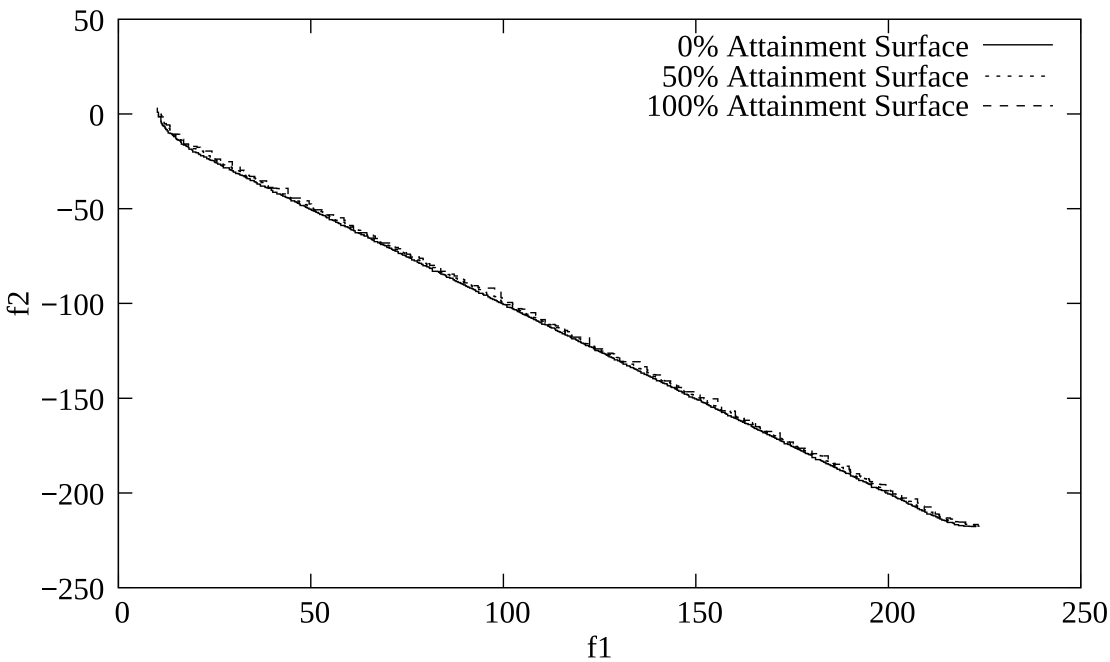

Figure 15: Attainment Surface using NSGA-II for problem SRN.

comparable with NSGA-II.

8 Conclusions

A multi-objective constraint handling approach is proposed in the present study. The proposed approach is an extended version of the hybrid bi-objective evolutionary and penalty function based classical optimization

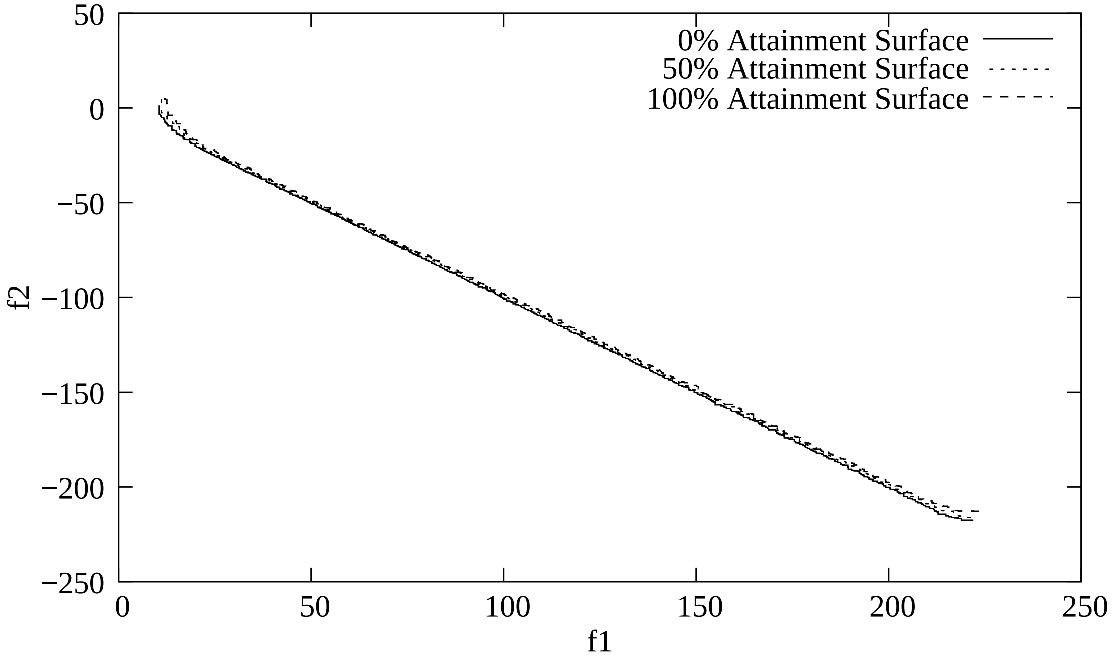

Figure 16: Attainment Surface using proposed approach for problem SRN.

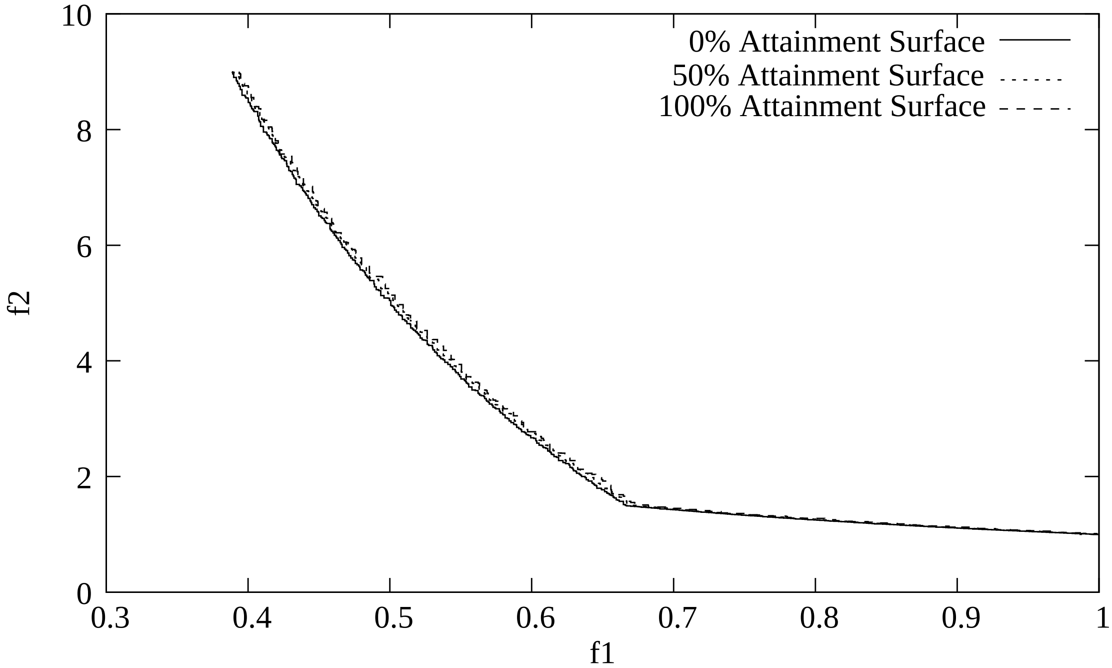

Figure 17: Attainment Surface using NSGA-II for problem CONSTR1.

approach. We restrict ourselves in solving bi-objective constraint handling problems in present study. However, scope of this constraint handling approach is that, it can be extended to any number of objectives.

The proposed approach is tested in four test problems, out of which one is proposed in the present study (problem MP1). Results are compared with NSGA-II due to its popularity in solving multi-objective optimization problems efficiently. In both the cases, 11 independent runs have been performed with different initial populations and identical parameter setting. The performance of both the algorithms are compared

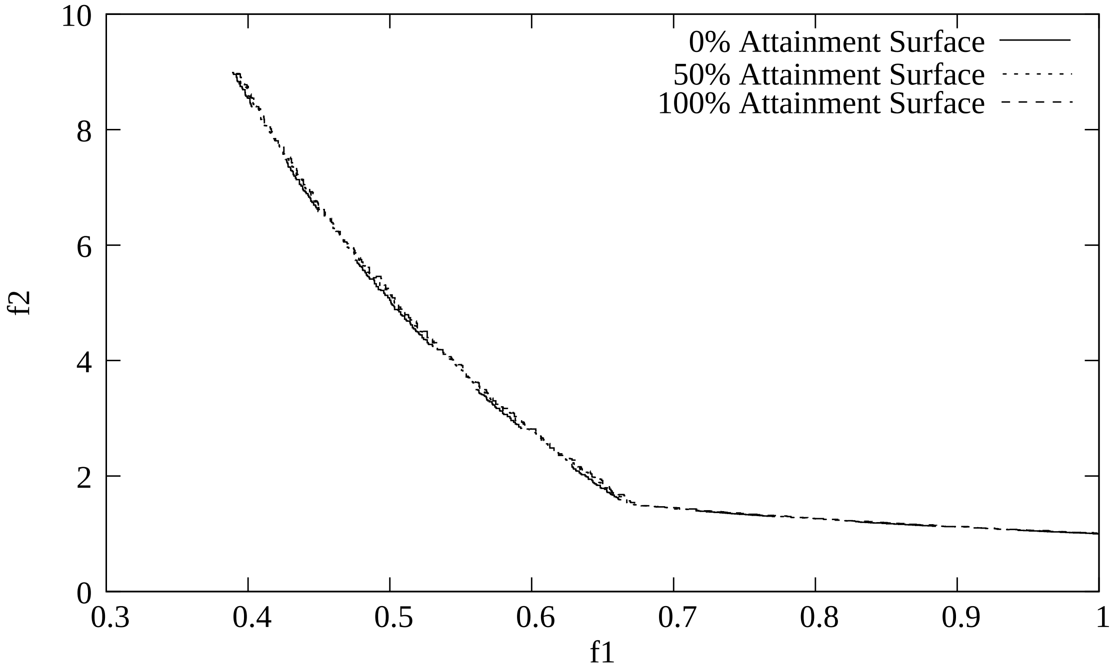

Figure 18: Attainment Surface using proposed approach for CONSTR1.

using hypervolume metric and attainment surface measure.

For the first problem (MP1), the proposed constraint method outperforms NSGA-II with very less number of population members. However, in case of other three test problems performance of both proposed method and NSGA-II are similar using hypervolume and attainment surface measure. Results from hypervolume metric and 0%, 50% and 100% attainment surfaces measures show that proposed method is robust and is comparable with the performance of NSGA-II. The comparative studies also indicate that the proposed multi-objective constraint handling method is successfully able to generate a well spread non-dominated solutions along with convergence, which is desirable in multi-objective evolutionary algorithms.

References

[1] A. Angantyr, J. Andersson, and J.-O. Aidanpaa. Constrained optimization based on a multiobjective evolutionary algorithm. In *Proceedings of Congress on Evolutionary Computation*, pages 1560–1567, 2003.

[2] K. Deb. *Multi-objective optimization using evolutionary algorithms*. Chichester, UK: Wiley, 2001.

[3] K. Deb, S. Agrawal, A. Pratap, and T. Meyarivan. A fast and elitist multi-objective genetic algorithm: NSGA-II. *IEEE Transactions on Evolutionary Computation*, 6(2):182–197, 2002.

[4] K. Deb and R. Datta. A fast and accurate solution of constrained optimization problems using a hybrid bi-objective and penalty function approach. In *Proceedings of the Congress on Evolutionary Computation (CEC-2010)*, pages 1–8, 2010.

[5] A. Isaacs, T. Ray, and W. Smith. Blessings of maintaining infeasible solutions for constrained multi-objective optimization problems. In *Evolutionary Computation, 2008. CEC 2008.(IEEE World Congress on Computational Intelligence). IEEE Congress on*, pages 2780–2787. IEEE, 2008.

[6] F. Jimenez, A.F. Gomez-Skarmeta, G. Sanchez, and K. Deb. An evolutionary algorithm for constrained multi-objective optimization. In *Evolutionary Computation, 2002. CEC'02. Proceedings of the 2002 Congress on*, volume 2, pages 1133–1138. IEEE, 2002.

[7] J.D. Knowles and D.W. Corne. Approximating the nondominated front using the pareto archived evolution strategy. *Evolutionary computation*, 8(2):149–172, 2000.

[8] Efrn Mezura-Montes and Carlos A. Coello Coello. Constraint-handling in nature-inspired numerical optimization: Past, present and future. *Swarm and Evolutionary Computation*, 1(4):173 − 194, 2011.

[9] T. Ray, H. Singh, A. Isaacs, and W. Smith. Infeasibility driven evolutionary algorithm for constrained optimization. In Efrn Mezura-Montes, editor, *Constraint-Handling in Evolutionary Computation*, pages 145–165. Springer, Berlin, 2009.

[10] J. Wef and Y. Wang. An infeasible elitist based particle swarm optimization for constrained multiobjective optimization and its convergence. *International Journal of Pattern Recognition and Artificial Intelligence*, 24(3):381–400, 2010.

[11] J. Wright and H. Loosemore. An infeasibility objective for use in constrained pareto optimization. In *Evolutionary Multi-Criterion Optimization*, pages 256–268. Springer, 2001.

[12] G. Yen. An adaptive penalty function for handling constraint in multi-objective evolutionary optimization. In Efrn Mezura-Montes, editor, *Constraint-Handling in Evolutionary Computation*, pages 121–143. Springer, Berlin, 2009.

[13] N. Young. Blended ranking to cross infeasible regions in constrainedmultiobjective problems. In *Computational Intelligence for Modelling, Control and Automation, 2005 and International Conference on Intelligent Agents, Web Technologies and Internet Commerce, International Conference on*, volume 2, pages 191–196. IEEE, 2005.

[14] Q. Zhang and H. Li. Moea/d: A multiobjective evolutionary algorithm based on decomposition. *Evolutionary Computation, IEEE Transactions on*, 11(6):712–731, 2007.

[15] E. Zitzler, M. Laumanns, and L. Thiele. SPEA2: Improving the Strength Pareto Evolutionary Algorithm for Multiobjective Optimization. In K.C. Giannakoglou et al., editors, *Evolutionary Methods for Design, Optimisation and Control with Application to Industrial Problems (EUROGEN 2001)*, pages 95–100. International Center for Numerical Methods in Engineering (CIMNE), 2002.