A Hybrid Integrated Multi-Objective Optimization Procedure for Estimating Nadir Point

Kalyanmoy Deb[1,⋆], Kaisa Miettinen[2], and Deepak Sharma[1]

[1] Department of Mechanical Engineering
Indian Institute of Technology Kanpur, PIN 208016, India
{deb,dsharma}@iitk.ac.in
[2] Department of Mathematical Information Technology
P.O. Box 35 (Agora), FI-40014
University of Jyväskylä, Finland
kaisa.miettinen@jyu.fi

Abstract. A nadir point is constructed by the worst objective values of the solutions of the entire Pareto-optimal set. Along with the ideal point, the nadir point provides the range of objective values within which all Pareto-optimal solutions must lie. Thus, a nadir point is an important point to researchers and practitioners interested in multi-objective optimization. Besides, if the nadir point can be computed relatively quickly, it can be used to normalize objectives in many multi-criterion decision making tasks. Importantly, estimating the nadir point is a challenging and unsolved computing problem in case of more than two objectives. In this paper, we revise a previously proposed serial application of an EMO and a local search method and suggest an integrated approach for finding the nadir point. A local search procedure based on the solution of a bi-level achievement scalarizing function is employed to extreme solutions in stabilized populations in an EMO procedure. Simulation results on a number of problems demonstrate the viability and working of the proposed procedure.

1 Introduction

A *nadir* point signifies, in principle, opposite to that meant by an *ideal* point, in the context of multi-objective optimization. An ideal point is an M-dimensional objective vector (where M is the number of objectives) constructed with best feasible objective values and is a comparatively easy to compute. For minimization problems, in principle, this calls for solving M single-objective minimization problems and collecting each optimal objective values to form the ideal point. On the other hand, a nadir point is constructed with the worst objective values of Pareto-optimal solutions. In minimization problems, this task is different from

⋆ Also Finland Distinguished Professor, Helsinki School of Economics, PO Box 1210, FIN-00101, Helsinki, Finland, (Kalyanmoy.Deb@hse.fi).

M. Ehrgott et al. (Eds.): EMO 2009, LNCS 5467, pp. 569–583, 2009.

simply maximizing M objective functions one at a time. This is because the search of the worst value of an objective must be restricted within the Pareto-optimal solutions. This is the reason why the estimation of nadir point has been found to be a complex task [13,11] and there does not exist any provable algorithm for the task, even for linear multi-objective optimization problems having three or more objectives.

With the advent of efficient evolutionary optimization procedures for multi-objective optimization, some attention has been made in the recent past in developing procedures for estimating the nadir point. Simplistic ideas, such as finding a set of Pareto-optimal solutions by an EMO procedure and then choosing the extreme solutions for estimating the nadir point, to more sophisticated ideas, such as replacing the focus of EMO to find a wide-spreaded set of solutions on the entire Pareto-optimal front to find only the critical extreme Pareto-optimal points [4,17], are suggested. Most of these EMO methodologies have shown to find an approximation of the nadir point, rather than to estimate the exact nadir point. Recent studies [6,5] suggested a two-step serial procedure of employing a modified NSGA-II procedure to identify extreme near Pareto-optimal solutions and then a local search procedure to converge to the true extreme Pareto-optimal points.

In this study, we suggest and simulate a hybrid integrated approach in which a local search procedure is used within the modified NSGA-II algorithm sparingly to achieve the nadir point estimation task. We restrict our discussions for real-parameter optimization problems, but the concept can very well be used for other types of optimization problems. The suggested local search procedure is based on utilizing a reference point based approach, a so-called achievement scalarizing function [18] which is widely used in the MCDM field. Using this scalarized function, any point in the objective space can be projected on the Pareto optimal front and the scalarizing function does not need any artificial information like weights [15]. In the procedure proposed, the achievement scalarizing function is used in a bi-level manner to guarantee getting reliable enough information about extreme values in the Pareto optimal front for estimating the nadir point. Based on a statistical analysis of the performance of the NSGA-II procedure, the execution of the local search event is decided dynamically at every generation. Both NSGA-II and local search procedures are terminated using statistical performance criteria. Simulation results on a number of test problems and three engineering problems are presented to demonstrate the efficacy of the proposed procedure.

2 Nadir Objective Vector

We consider multi-objective optimization problems involving M conflicting objectives ($f_i : \mathcal{S} \rightarrow \mathbf{R}$) as functions of decision variables $\mathbf{x}$:

$$\begin{array}{c} \text{minimize } \{f_1(\mathbf{x}), f_2(\mathbf{x}), \ldots, f_M(\mathbf{x})\}, \\ \text{subject to } \mathbf{x} \in \mathcal{S}, \end{array} \tag{1}$$

where $\mathcal{S} \subset \mathbf{R}^n$ denotes the set of feasible solutions. Problem (1) gives rise to a set of *Pareto-optimal* solutions or a Pareto-optimal front (P^*), providing a trade-off among the objectives. In the sense of minimization of objectives, Pareto-optimal solutions can be defined as follows [15]:

Definition 1. *A decision vector* $\mathbf{x}^* \in \mathcal{S}$ *and the corresponding objective vector* $\mathbf{f}(\mathbf{x}^*)$ *are Pareto-optimal if there does not exist another decision vector* $\mathbf{x} \in S$ *such that* $f_i(\mathbf{x}) \leq f_i(\mathbf{x}^*)$ *for all* $i = 1, 2, \ldots, M$ *and* $f_j(\mathbf{x}) < f_j(\mathbf{x}^*)$ *for at least one index* j.

In what follows, we assume that the Pareto-optimal front is bounded. We now define a nadir objective vector, that is, a nadir point, as follows.

Definition 2. *An objective vector* $\mathbf{z}^{\mathrm{nad}} = (z_1^{\mathrm{nad}}, \ldots, z_M^{\mathrm{nad}})^T$ *constructed using the worst values of objective functions in the complete Pareto-optimal front* P^* *is called a nadir objective vector.*

Hence, for minimization problems we have $z_j^{\mathrm{nad}} = \max_{\mathbf{x} \in P^*} f_j(\mathbf{x})$. Estimation of the nadir objective vector is, in general, a difficult task. Unlike the *ideal objective vector* $\mathbf{z}^* = (z_1^*, \ldots, z_M^*)^T$, which can be found by minimizing each objective individually over the feasible set S (or, $z_j^* = \min_{\mathbf{x} \in \mathcal{S}} f_j(\mathbf{x})$), the nadir point cannot be formed by maximizing objectives individually over $\mathcal{S}$. To find the nadir point, Pareto-optimality of solutions used for constructing the nadir point must be first established. This makes the task of finding the nadir point a difficult one. To illustrate this aspect, let us consider a bi-objective minimization problem shown in Figure 1. If we maximize f_1 and f_2 individually, we obtain points A and B, respectively. These two points can be used to construct the so-called *worst objective vector*, $\mathbf{z}^w$. In many problems (even in bi-objective optimization problems), the nadir objective vector and the worst objective vector are not the same point, which can also be seen in Figure 1.

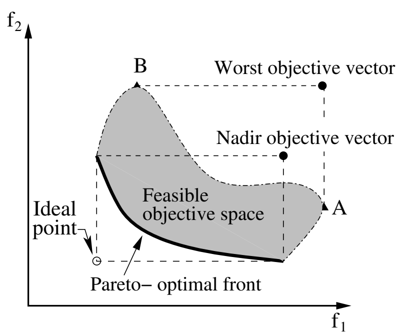

Fig. 1. The nadir and worst objective vectors

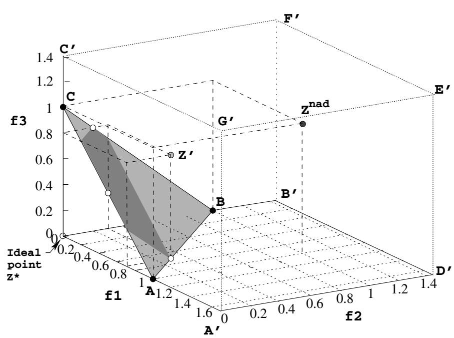

Fig. 2. Payoff table may not produce the true nadir point

3 Existing Methods

3.1 Payoff Table Method

Benayoun et al. [1] introduced the first interactive multi-objective optimization method for estimating the nadir point by using a *payoff table*. To be more specific, each objective function is first minimized individually and then a table is constructed where the i-th row of the table represents values of all objective functions calculated at the point where the i-th objective obtained its minimum value. Thereafter, the maximum value of the j-th column can be considered as an estimate of the upper bound of the j-th objective in the Pareto-optimal front and these maximum values together may be used to construct an approximation of the nadir objective vector. The main difficulty of such an approach is that solutions are not necessarily unique and thus corresponding to the minimum solution of an objective there may exist more than one solutions having different values of other objectives, in problems having more than two objectives. In these problems, the payoff table method may not result in an accurate estimation of the nadir objective vector. To illustrate, consider a three-objective problem shown in Figure 2. Minimization of the first objective will result in any solution on the trapezium CBB′F′C′C. If the point marked in a small circle on line CB is obtained by an optimization algorithm and similarly other two circles on lines CA and AB are obtained for minimizations of f_2 and f_3, respectively, a wrong estimate (z') of the nadir point (z^{nad}) will be made.

3.2 Evolutionary Approaches

The nadir point is associated with Pareto-optimal solutions and, thus, determining a set of Pareto-optimal solutions will facilitate the estimation of the nadir point. Since an EMO algorithm is aimed at finding a set of Pareto-optimal solutions, it may be an ideal way to find the nadir objective vector. Several approaches are proposed recently.

In the naive approach, first a well-distributed set of Pareto-optimal solutions can be attempted to find by an EMO [4]. Thereafter, an estimate of the nadir objective vector can be made by picking the worst values of each objective [17]. In the context of the problem depicted in Figure 2, this means first finding a well-represented set of solutions on the plane ABC and then estimating the nadir point from them. Since EMO algorithms are not found to converge well and maintain a well-diverse set of solutions for more than three objectives [7], the accuracy of the estimated nadir point using the naive approach is questionable.

Szczepanski and Wierzbicki [17] have simulated the idea of solving multiple bi-objective optimization problems suggested in [8] using an EMO approach and construct the nadir point by accumulating all bi-objective Pareto-optimal fronts together. As discussed in our earlier study [5], such a technique is not generic and requires additional objective and variable-space niching techniques to correctly estimate the nadir point. Moreover, the procedure requires $\binom{M}{2}$ bi-objective optimizations, making it a daunting task particularly for problems having more than three objectives.

However, the idea of concentrating on a preferred region on the Pareto-optimal front, instead of finding the entire Pareto-optimal front, can be pushed further. An emphasis can be placed in an EMO approach to find only the critical extreme points of the Pareto-optimal front. Our earlier study [4] suggested two approaches in the crowding distance operator of the NSGA-II procedure and concluded in favor of the extremized crowding distance approach. In the extremized-crowded NSGA-II approach [4], we emphasized in concentrating on the best and worst solutions of each objective. In this approach, solutions on a particular non-dominated front are first sorted from minimum (with rank $R_i^{(m)} = 1$) to maximum (with rank $= N_f$) based on each objective. The rank of solution i for the m-th objective $R_i^{(m)}$ is assigned as $\max\{R_i^{(m)}, N_f - R_i^{(m)} + 1\}$. Two extreme solutions for every objective get a rank equal to N_f (number of solutions in the non-dominated front), the solutions next to these extreme solutions get a rank $(N_f - 1)$, and so on. After a rank is assigned to a solution by each objective, the maximum value of the assigned ranks is declared as the crowding distance.

Like other evolutionary optimization studies, the proposed extremized crowded NSGA-II approach did not ensure converging to the true extreme solutions exactly, as evolutionary algorithms are expected to find a near-optimal solution, rather than a true optimal solution in a finite number of solution evaluations. However, in the pursuit of estimating the nadir point for the purpose of normalizing objectives for executing different multi-objective optimization algorithms or for knowing the true range of Pareto-optimal solutions for decision-making, it is important to find the true extreme Pareto-optimal points, so that the nadir point can be estimated accurately.

In a recent study [6], the extremized crowded NSGA-II approach is ended with a bi-level local search operation on all extreme solutions to take them arbitrary closer to the true extreme solutions, so that the nadir point can be estimated more accurately. In this paper, we re-address the issue of the serial application of NSGA-II and the local search procedure and suggest a hybrid integrated approach for an accurate estimation of the nadir point.

4 Proposed Integrated Approach

Instead of applying the local search on the extreme solutions obtained by the extremized crowded hybrid NSGA-II procedure, we propose an integrated NSGA-II approach in which at certain generations the extreme solutions of the best non-dominated front are modified by the local search procedure to push them towards their true values. With such an integrated procedure, the attained accuracy is likely to be better and it would have a smaller chance of getting stuck to intermediate solutions, thereby leading to an accurate estimation of the nadir point. In the following, we outline an iteration of the proposed integrated NSGA-II procedure in which the population P_t is the current parent population of size N. Every member (i) of P_t is already ranked based on its non-domination level (ND_i) and its crowding distance (CD_i) within the population members of its own non-domination level.

Step 1: Population P_t is used to create an offspring population Q_t by using binary tournament selection, recombination and mutation operators. Two solutions are chosen at random from P_t and a hierarchical selection based on ND followed by CD is used to complete the tournament selection operation. Thereafter, two such selected solutions are recombined using the simulated binary crossover operator [3,2] to create two offspring solutions, each of which is then mutated by using the polynomial mutation operator [2]. These operators involve the following parameters: recombination probability p_c, SBX index η_c, mutation probability p_m, and mutation index η_m.

Step 2: Populations P_t and Q_t are combined together and ranked into different levels of non-domination: $P_t \cup Q_t = \{\mathcal{F}_1, \mathcal{F}_2, \ldots\}$. The set $\mathcal{F}_1$ contains non-dominated solutions of level one, and so on. Thereafter, the best N population members are chosen from the combined $2N$ population based on ranking and crowding distance criteria.

Step 3: Depending on a check on whether to perform the local search or not (which we describe a little later), in the set $\mathcal{F}_1$, we identify the worst solution ($\mathbf{x}^{(j)}$) with respect to each objective j, and modify it by using a local search procedure. The modified solution ($\mathbf{y}^{(j)}$) replaces the worst population member. For M objectives, there are M such local search operations performed in each iteration of the proposed procedure. The estimated nadir point ($\mathbf{z}^{\text{est}}$) at generation t is then formed from the extreme solutions obtained by the local searches. Non-domination ranking and crowding distance computations are redone on the modified population, which we refer to as P_{t+1}.

This procedure is similar to the original NSGA-II procedure, except that the crowding distance computation is different suiting the need for emphasizing extreme solutions for the task of estimating the nadir point and that a local search procedure is used to update the extreme objective-wise solutions to make sure that the nadir point can be estimated with a desired accuracy.

We now describe the local search procedure here. The best ($f_j^{\min}$) and worst ($f_j^{\max}$) values of each objective j of the set $\mathcal{F}_1$ are first noted. We apply a bi-level local search procedure from each worst solution (solution $\mathbf{x}^{(j)}$ for which the j-th objective has the worst value in $\mathcal{F}_1$) to find the corresponding optimal solution $\mathbf{y}^{(j)}$ using the following bi-level optimization procedure. The upper-level optimization (described in (2)) uses an objective vector ($\mathbf{z}$, referred here as a reference point) as a variable vector and maximizes the j-th objective value of the optimal solution obtained by solving the corresponding augmented achievement scalarizing problem [15] (we refer to this task as the lower-level optimization task, described in (3)):

$$\begin{aligned} &\text{maximize}_{(\mathbf{z})}\ f_j^*(\mathbf{z}),\\ &\quad\text{subject to}\ z_i \geq {f_i^{(j)}}_{\text{EA}} - 0.5(f_i^{\max} - f_i^{\min}), \quad i = 1,2,\ldots,M,\\ &\qquad\qquad\quad\ z_i \leq {f_i^{(j)}}_{\text{EA}} + 1.5(f_i^{\max} - f_i^{\min}), \quad i = 1,2,\ldots,M. \end{aligned} \tag{2}$$

The term $f_j^*(\mathbf{z})$ is the optimal value of the j-th objective function of the optimal solution to the following lower-level optimization problem for which $\mathbf{z}$ is kept fixed [18]:

$$\begin{array}{l} \text{minimize}_{(\mathbf{y})} \max_{i=1}^{M} \left(\frac{f_i(\mathbf{y})-z_i}{f_i^{\max}-f_i^{\min}} \right) + \rho \sum_{k=1}^{M} \left(\frac{f_k(\mathbf{y})-z_k}{f_k^{\max}-f_k^{\min}} \right), \\ \text{subject to} \quad \mathbf{y} \in \mathcal{S}, \end{array} \tag{3}$$

Figure 3 illustrates this local search procedure. In the lower-level optimization problem, the search is performed on the original decision variable space. The solution $\mathbf{y}{*}^{(j)}(\mathbf{z})$ to this lower-level optimization problem determines the optimal objective vector $\mathbf{f}^*$ from which we extract the j-th component and use it in the upper-level optimization problem. Thus, for every reference point $\mathbf{z}$ (a solution for the upper-level problem), the corresponding optimal augmented achievement scalarizing function is found in the lower-level loop. The upper-level optimization is initialized with the NSGA-II solution $\mathbf{z}^{(0)} = \mathbf{f}(\mathbf{x}^{(j)})$ and the lower-level optimization is initialized with the NSGA-II solution $\mathbf{y}^{(0)} = \mathbf{x}^{(j)}$.

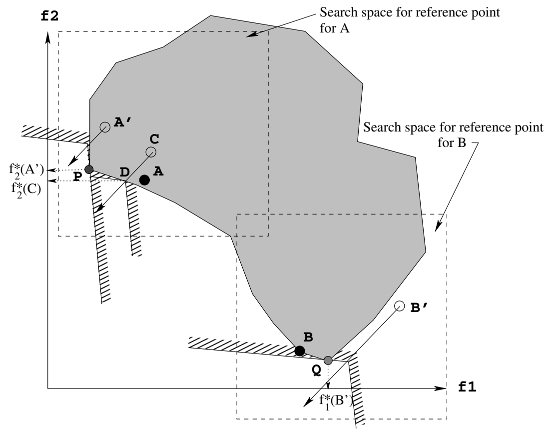

Fig. 3. Each arrow corresponds to a lower-level search for a specified reference point (C, A' or B'). The upper-level search finds a reference point having optimal worst objective (such as A' or B').

We now discuss the termination criterion of each optimization procedure. For terminating the overall NSGA-II procedure, we compute a *normalized distance* (ND) metric as follows:

$$\mathcal{D} = \sqrt{\frac{1}{M} \sum_{i=1}^{M} \left(\frac{z_i^{\text{est}} - z_i^*}{z_i^w - z_i^*} \right)^2}. \tag{4}$$

Here, the vectors $\mathbf{z}^*$ and $\mathbf{z}^w$ are the ideal and worst objective vectors of the optimization problem, respectively. These quantities can be computed once before the NSGA-II procedure by solving $2M$ different single-objective optimizations of minimizing and maximizing each objective at a time.

Since the exact final value of the $\mathcal{D}$ metric is not known a priori on an arbitrary problem, we record the change in $\mathcal{D}$ for the past τ ($= 50$ used here) generations. Say, $\mathcal{D}_{\max}$, $\mathcal{D}_{\min}$, and $\mathcal{D}_{\text{avg}}$, are the maximum, minimum, and average $\mathcal{D}$ values for the past consecutive τ generations. If the change $\Delta D = (\mathcal{D}_{\max} - \mathcal{D}_{\min})/\mathcal{D}_{\text{avg}}$ is smaller than a threshold Δ ($= 1(10^{-4})$ is used here), the NSGA-II procedure is terminated.

We use the same normalized distance metric to decide whether the local search needs to be performed in a particular generation of NSGA-II. At a generation, the change $\Delta_l D$ in normalized distance over the past τ_l ($= 20$ used here) generations is recorded. If $\Delta_l D \leq \delta$ (= 0.005 used here), the local search is performed. This reduces the number of local searches performed from not so good solutions. When the best non-dominated front has stabilized somewhat, the extreme solutions of the set are modified using the local search procedure.

Both upper and lower-level optimization tasks in the local search operation uses a point-by-point search approach which is terminated based on the chosen optimization algorithm and code used for the purpose. In all our simulations, we have used KNITRO (auto option) for the lower-level optimization task in which we have set a termination condition on the KKT error value ($\leq 10^{-6}$) or a maximum of 100 iterations whichever happens first. For the upper-level optimization task, we have used CFSQP solver [14]. The upper-level task is terminated if the norm of the Newton's direction is less than of equal to 10^{-8} or a maximum iteration of 100 is elapsed.

After the NSGA-II run is terminated, we construct the nadir point from the worst objective values of the final non-dominated set $\mathcal{F}_1$.

5 Simulation Results

In this section, we present simulation results on eight problems having three or more objectives. In most of these problems, the nadir point was difficult to obtain using the pay-off table. In all problems, we use a population of size $\max(60, 20n)$ (n is the number of variables), crossover and mutation probabilities of 0.9 and $1/n$, crossover and mutation indices of 10 and 50, respectively, and $\rho = 10^{-4}$. In each case, we make 10 different runs from different initial populations, but every time the procedure is found to converge near a particular set of extreme points, thereby leading to finding a similar nadir point every time.

5.1 Problem KM

The first problem KM, adapted from [12], is the following:

$$\begin{array}{rl} \text{minimize} & \left\{ \begin{array}{l} -x_1 - x_2 + 5 \\ \frac{1}{5}(x_1^2 - 10x_1 + x_2^2 - 4x_2 + 11) \\ (5 - x_1)(x_2 - 11) \end{array} \right\}, \\ \text{subject to} & 3x_1 + x_2 - 12 \leq 0, \quad 2x_1 + x_2 - 9 \leq 0, \quad x_1 + 2x_2 - 12 \leq 0, \\ & 0 \leq x_1 \leq 4, \quad 0 \leq x_2 \leq 6. \end{array} \tag{5}$$

The true nadir point of this problem is reported to be $\mathbf{z}^{\text{nad}} = (5, 4.6, -14.25)^T$ [9]. Table 1 shows the three extreme solutions ($\mathbf{x}^*$) found by our proposed approach. It is clear that when the worst objective values are collected together, we obtain an identical point (up to two decimal points) as that in the true nadir point. Figure 4 shows that the normalized distance value gets stabilized at around 40 generation and since $\Delta D = 50$ is used, it took another 50 generations to terminate the hybrid procedure. Interestingly, the $\mathcal{D}$ value reaches the final stabilized value very quickly, thereby indicating the efficiency of the proposed procedure.

Table 1. Extreme points found by the proposed approach on problem KM

$\mathbf{x}^*$		Estimated $\mathbf{z}^{\text{nad}}$		
0.000	0.000	**5.000**	2.200	-55.000
0.000	6.000	-1.000	**4.600**	-25.001
3.500	1.501	0.000	-3.100	**-14.251**

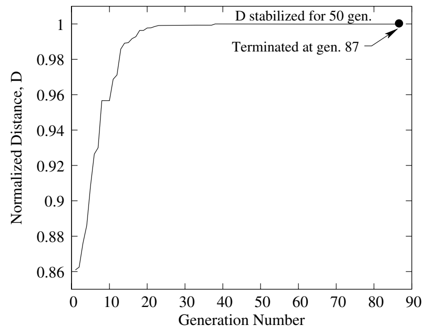

Fig. 4. Variation of $\mathcal{D}$ with generation on KM

5.2 Problem SW1

The second problem SW1 is as follows [17]:

$$\begin{aligned} &\text{minimize} \left\{ \begin{array}{l} f_1(\mathbf{x}) = -(100 - 7x_1 - 20x_2 - 9x_3) \\ f_2(\mathbf{x}) = -(4x_1 + 5x_2 + 3x_3) \\ f_3(\mathbf{x}) = -x_3 \end{array} \right\}, \\ &\text{subject to } 1\tfrac{1}{2}x_1 + x_2 + 1\tfrac{3}{5}x_3 \le 9, \quad x_1 + 2x_2 + x_3 \le 10, \\ &\qquad x_i \ge 0, \quad i = 1, 2, 3. \end{aligned} \tag{6}$$

The previous study [17] reported the true nadir point to be $\mathbf{z}^{\text{nad}} = (-3.6364, 0, 0)^T$. Table 2 shows two extreme solutions ($\mathbf{x}^*$) (hence, the true nadir point) found by our proposed approach. Figure 5 shows the progress of the proposed approach.

5.3 Problem SW2

The third problem SW2 originates from [17]:

$$\begin{aligned} &\text{minimize} \left\{ \begin{array}{l} 9x_1 + 19.5x_2 + 7.5x_3 \\ 7x_1 + 20x_2 + 9x_3 \\ -(4x_1 + 5x_2 + 3x_3) \\ -(x_3) \end{array} \right\}, \\ &\text{subject to } 1.5x_1 - x_2 + 1.6x_3 \le 9, \quad x_1 + 2x_2 + x_3 \le 10, \\ &\qquad x_i \ge 0, \quad i = 1, 2, 3. \end{aligned} \tag{7}$$

The true nadir point for this problem is reported to be $\mathbf{z}^{\text{nad}} = (94.5, 96.3636, 0, 0)^T$ [17]. The original study [17] found a close point $(94.4998, 95.8747, 0, 0)^T$ using multiple, bi-objective optimization simulation using an EMO procedure. The outcome is not identical to the true nadir point. Table 3 shows the three extreme solutions found by our proposed approach. We obtain the true nadir point. Due to an identical behavior of $\mathcal{D}$ variation with generation number on this and subsequent problems, we do not show the figures here.

Table 2. Extreme points found by the proposed approach on problem SW1

$\mathbf{x}^*$			Estimated $\mathbf{z}^{\text{nad}}$		
0.0000	3.1818	3.6364	**-3.6364**	-26.8182	-3.6364
0.0000	0.0000	0.0000	-100.0000	**0.0000**	**0.0000**

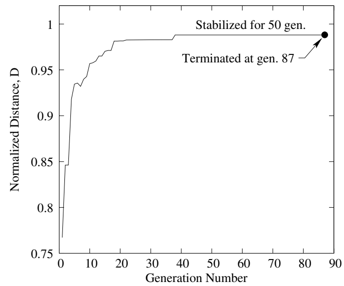

Fig. 5. Variation of $\mathcal{D}$ with generation on SW1

Table 3. Extreme points found by the proposed approach on problem SW2

$\mathbf{x}^*$			Estimated $\mathbf{z}^{\text{nad}}$			
4.0000	3.0000	0.0000	**94.5000**	88.0000	-31.0000	0.0000
0.0000	3.1818	3.6363	89.3182	**96.3636**	-26.8182	-3.6363
0.0000	0.0000	0.0000	0.0000	0.0000	**0.0000**	**0.0000**

5.4 Problem KSS1

The linear KSS1 problem [13] was found to be difficult for estimating the nadir point:

$$\begin{aligned} &\text{maximize} \left\{ \begin{array}{l} 11x_2 + 11x_3 + 12x_4 + 9x_5 + 9x_6 - 9x_7 \\ 11x_1 + 11x_3 + 9x_4 + 12x_5 + 9x_6 - 9x_7 \\ 11x_1 + 11x_2 + 9x_4 + 9x_5 + 12x_6 + 12x_7 \end{array} \right\}, \\ &\text{subject to } \textstyle\sum_{i=1}^{7} x_i = 1, \\ &\qquad\qquad x_i \geq 0, \quad i = 1, 2, \ldots, 7. \end{aligned} \tag{8}$$

The true nadir point is reported to be $\mathbf{z}^{\text{nadir}} = (0, 0, 0)^T$ [13]. Table 4 shows the three extreme solutions found by our proposed approach. Our approach finds a near nadir point with a slight error in the second objective value (as shown in Figure 6 the error is not visually detectable). This problem is a difficult one to solve for estimating the exact nadir point, because of the slow slope leading to each of the three extreme points, as shown by a set of representative solutions obtained through a *clustered* NSGA-II, in which NSGA-II's crowding distance method is replaced by the k-mean clustering method [2]. In this problem, it is easy to get stuck to a non-dominated point close to one or more extreme points. Our approach seems to have found the exact extreme values for first and third

Table 4. Extreme points found by the proposed approach on problem KSS1

$\mathbf{x}^*$							Estimated $\mathbf{z}^{\text{nad}}$		
1.000	0.000	0.000	0.000	0.000	0.000	0.000	**0.000**	11.000	11.000
0.000	0.994	0.000	0.000	0.000	0.001	0.004	10.910	**-0.026**	11.006
0.000	0.000	1.000	0.000	0.000	0.000	0.000	11.000	11.000	**0.000**

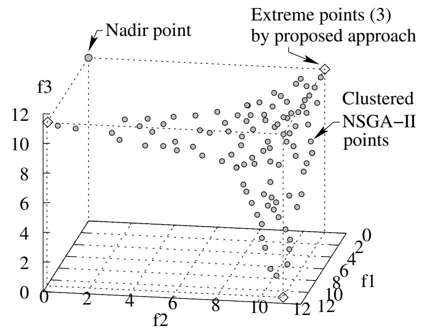

Fig. 6. Pareto-optimal front shows long narrow regions near extreme points in problem KSS1

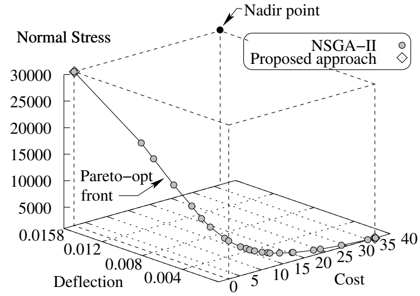

Fig. 7. Pareto-optimal front and two obtained points for problem WELD

objectives and managed to get to a near-by point around the extreme of the second objective.

5.5 Problem KSS2

Next, we consider another linear problem KSS2 [13]:

$$
\begin{array}{rl}
\text{maximize} & (x_1, x_2, x_3), \\
\text{subject to} & x_1 + 2x_2 + 2x_3 \leq 8, \quad 2x_1 + 2x_2 + x_3 \leq 8, \quad 3x_1 - 2x_2 + 4x_3 \leq 12, \\
& x_i \geq 0, \quad i = 1, 2, 3.
\end{array}
\tag{9}
$$

The nadir point is reported to be $\mathbf{z}^{\text{nad}} = (0, 0, 0)^T$. Table 5 presents the extreme solutions obtained by our approach. The true nadir point is found by our approach in this problem.

Table 5. Extreme points found by the proposed approach on problem KSS2

$\mathbf{x}^*$			Estimated $\mathbf{z}^{\text{nad}}$		
0.000	3.818	0.166	**0.000**	3.818	0.166
3.344	0.000	0.432	3.344	**0.000**	0.433
3.253	0.628	0.000	3.253	0.628	**0.000**

Now we consider three more problems, borrowed from engineering fields. On each of these problems, the exact nadir point is not known, but wherever possible we explain the accuracy of the nadir point obtained by our approach.

5.6 Problem WELD

The WELD problem has four variables and three objectives, and is formulated in [6]. Our previous study [6] introduced the WELD problem which has four

Table 6. Extreme points found by the proposed approach on problem WELD

$\mathbf{x}^*$				Estimated $\mathbf{z}^{\text{nad}}$		
1.7356	0.4788	10.0000	5.0000	**36.4221**	0.000439	1008.0000
0.2444	6.2175	8.2915	0.2444	2.3810	**0.015759**	**30000.1284**

variables and three objectives. The nadir point was estimated to be $\mathbf{z}^{\text{nad}} = (36.4209, 0.0158, 30000)^T$. Table 6 presents two extreme points found by our proposed approach of this paper. The extreme points for the second and third objectives are found to be identical in this problem, indicating that although the problem has three objective functions, the Pareto-optimal front is two-dimensional, as is also confirmed by the original NSGA-II points in Figure 7. The nadir point estimated by our approach is $(36.4221, 0.0158, 30000.1284)^T$, which is close to that obtained by the earlier study [6].

5.7 Problem CAR

The seven-variable, three-objective CAR problem is described in [10]. No previous study exists on this problem for finding the nadir point. In Table 7, we present two extreme points obtained by our procedure. Thus, the nadir point estimated by our approach for this problem is $\mathbf{z}^{\text{nad}} = (42.767, 4.000, 12.521)^T$. Figure 8 shows the complete Pareto-optimal front with a set of representative clustered NSGA-II solutions. It is clear from the plot that the above two extreme points are adequate to cover the extreme objective values of the Pareto-optimal front and is able to locate the nadir point of the problem.

5.8 Problem WATER

Finally, we consider the WATER problem [16], which is also described in [2]. For this problem, the exact nadir point is not known. However, since there are

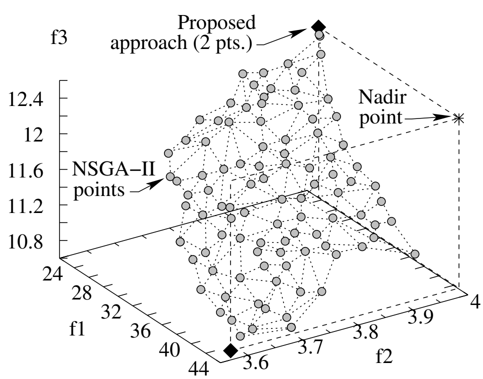

Fig. 8. Extreme objective vectors covers the entire Pareto-optimal front for problem CAR

Table 7. Extreme points found by the proposed approach on problem CAR

$\mathbf{x}^*$							Estimated $\mathbf{z}^{\mathrm{nad}}$		
1.500	1.350	1.500	1.500	2.625	1.200	1.200	**42.767**	3.585	10.611
0.500	1.226	0.500	1.208	0.875	0.884	0.400	23.589	**4.000**	**12.521**

Table 8. Extreme points found by the proposed approach on problem WATER

$\mathbf{x}^*$			Estimated $\mathbf{z}^{\mathrm{nad}}$				
0.010	0.100	0.100	**1.038**	0.020	0.949	0.075	5.649
0.450	0.098	0.010	0.916	**0.900**	0.936	0.033	0.002
0.114	0.100	0.010	0.918	0.228	**0.951**	0.031	0.285
0.098	0.010	0.100	0.918	0.197	0.095	**2.671**	**5.713**

three variables and five objectives, some redundancy in the objectives is expected for the Pareto-optimal solutions. An application of NSGA-II to this problem [2] (page 388) was found to indicate some correlations among the obtained representative solutions. Table 8 presents the extreme points obtained for this problem by our approach. We observe that the extreme points for objectives f_4 and f_5 come from an identical solution. The estimated nadir point using our procedure is $\mathbf{z}^{\mathrm{nad}} = (1.038, 0.900, 0.951, 2.671, 5.713)^T$.

6 Conclusions

In this paper, we have extended our previous study on a serial implementation of an EMO procedure followed by an MCDM based local search approach to find extreme points accurately for estimating the nadir point of a multi-objective optimization problem. The nadir point in multi-objective optimization is used in normalizing objectives which is necessary for different multi-criterion optimization algorithms. Besides, the task of estimating the nadir point for three or more objectives is a open research task in multi-criterion optimization literature. Nadir points can only be estimated accurately if (i) objective-wise extremes and (ii) Pareto-optimal solutions are found. Due to this two-pronged requirements, we have suggested a bi-level local search task. The local search is employed with extreme non-dominated solutions only when the best non-dominated front has stabilized somewhat, thereby making the overall method computationally tractable. On a set of five test problems and three engineering design problems, the proposed integrated procedure has able to find the exact nadir point quite accurately.

This work is also important from another point of view. This work demonstrates how a local search approach can be integrated with an evolutionary population-based approach adaptively and used sparingly for a complex optimization to ensure accurate convergence.

Acknowledgments

Authors acknowledge the support from the Academy of Finland, Foundation of Helsinki School of Economics (grant 118319), and the Jenny and Antti Wihuri Foundation during the course of this study.

References

1. Benayoun, R., de Montgolfier, J., Tergny, J., Laritchev, P.: Linear programming with multiple objective functions: Step method (STEM). Mathematical Programming 1(3), 366–375 (1971)
2. Deb, K.: Multi-objective optimization using evolutionary algorithms. Wiley, Chichester (2001)
3. Deb, K., Agrawal, R.B.: Simulated binary crossover for continuous search space. Complex Systems 9(2), 115–148 (1995)
4. Deb, K., Chaudhuri, S., Miettinen, K.: Towards estimating nadir objective vector using evolutionary approaches. In: Proceedings of the Genetic and Evolutionary Computation Conference (GECCO 2006), pp. 643–650. The Association of Computing Machinery (ACM), New York (2006)
5. Deb, K., Miettinen, K.: A review of nadir point estimation procedures using evolutionary approaches: A tale of dimensionality reduction. Technical Report KanGAL Report Number 2008004, Kanpur Genetic Algorithms Laboratory (KanGAL), Indian Institute of Technology Kanpur, India (2008)
6. Deb, K., Miettinen, K., Chaudhuri, S.: Estimating nadir objective vector: Hybrid of evolutionary and local search. Technical Report Electronic Working Paper W-440, Helsinki School of Economics, Finland (2008)
7. Deb, K., Saxena, D.: Searching for Pareto-optimal solutions through dimensionality reduction for certain large-dimensional multi-objective optimization problems. In: Proceedings of the World Congress on Computational Intelligence (WCCI 2006), pp. 3352–3360 (2006)
8. Ehrgott, M., Tenfelde-Podehl, D.: Computation of ideal and nadir values and implications for their use in MCDM methods. European Journal of Operational Research 151, 119–139 (2003)
9. Eskelinen, P., Miettinen, K., Klamroth, K., Hakanen, J.: Pareto navigator for interactive nonlinear multiobjective optimization. OR Spectrum (in press)
10. Gu, L., Yang, R.J., Tho, C.H., Makowski, L., Faruque, O., Li, Y.: Optimization and robustness for crashworthiness of side impact. International Journal of Vehicle Design 26(4) (2001)
11. Isermann, H., Steuer, R.E.: Computational experience concerning payoff tables and minimum criterion values over the efficient set. European Journal of Operational Research 33(1), 91–97 (1988)
12. Klamroth, K., Miettinen, K.: Integrating approximation and interactibe decision making in multicrietria optimization. Operations Research 56, 222–234 (2008)
13. Korhonen, P., Salo, S., Steuer, R.: A heuristic for estimating nadir criterion values in multiple objective linear programming. Operations Research 45(5), 751–757 (1997)

14. Lawrence, C., Zhau, J.L., Tits, A.L.: User's guide for cfsqp version 2.5: A C code for solving (large scale) constrained nonlinear (minimax) optimization problems, generating iteration satisfying all inequality constraints. Technical Report Institute for Systems Research TR-94-16r1, Electrical Engineering Department and Institute for Systems Research, University of Maryland, College Park, MD 20742 (1997)
15. Miettinen, K.: Nonlinear Multiobjective Optimization. Kluwer, Boston (1999)
16. Ray, T., Tai, K., Seow, K.C.: An evolutionary algorithm for multiobjective optimization. Engineering Optimization 33(3), 399–424 (2001)
17. Szczepanski, M., Wierzbicki, A.P.: Application of multiple crieterion evolutionary algorithm to vector optimization, decision support and reference point approaches. Journal of Telecommunications and Information Technology 3, 16–33 (2003)
18. Wierzbicki, A.P.: The use of reference objectives in multiobjective optimization. In: Fandel, G., Gal, T. (eds.) Multiple Criteria Decision Making Theory and Applications, pp. 468–486. Springer, Berlin (1980)