A Hybrid Framework for Evolutionary Multi-objective Optimization

Karthik Sindhya, Kaisa Miettinen, and Kalyanmoy Deb, *Fellow, IEEE*

***Abstract*—Evolutionary multi-objective optimization algorithms are widely used for solving optimization problems with multiple conflicting objectives. However, basic evolutionary multi-objective optimization algorithms have shortcomings, such as slow convergence to the Pareto optimal front, no efficient termination criterion, and a lack of a theoretical convergence proof. A hybrid evolutionary multi-objective optimization algorithm involving a local search module is often used to overcome these shortcomings. But there are many issues that affect the performance of hybrid evolutionary multi-objective optimization algorithms, such as the type of scalarization function used in a local search and frequency of a local search. In this paper, we address some of these issues and propose a hybrid evolutionary multi-objective optimization framework. The proposed hybrid evolutionary multi-objective optimization framework has a modular structure, which can be used for implementing a hybrid evolutionary multi-objective optimization algorithm. A sample implementation of this framework considering NSGA-II, MOEA/D, and MOEA/D-DRA as evolutionary multi-objective optimization algorithms is presented. A gradient-based sequential quadratic programming method as a single objective optimization method for solving a scalarizing function used in a local search is implemented. Hence, only continuously differentiable functions were considered for numerical experiments. The numerical experiments demonstrate the usefulness of our proposed framework.**

***Index Terms*—Memetic optimization, MOEA/D, MOEA/D-DRA, multicriteria optimization, multiple criteria decision making (MCDM), NSGA-II, Pareto optimality.**

I. Introduction

MULTIPLE conflicting objectives for optimization typically arise in all fields of science and engineering. Researchers and practitioners formulate such problems as multi-objective optimization problems (also called multicriteria, multiple objective, and vector optimization) in which the goal is to minimize or maximize several conflicting objective functions simultaneously. In multi-objective optimization, there are usually several optimal solutions with different tradeoffs, called Pareto optimal solutions. The set of Pareto optimal solutions in the objective space is called the Pareto optimal front.

There exist at least two different research fields which focus on solving multi-objective optimization problems: multiple criteria decision making (MCDM) [1]–[3] and evolutionary multi-objective optimization (EMO) [4] and [5]. Although they address similar problems as emphasized, they have different research goals [6]. On the one hand, in MCDM, typically the aim is to support a human decision maker in identifying the most preferred solution. Here, a multi-objective optimization problem is often scalarized into a single objective optimization problem accounting for the preference information of a decision maker and later solved using an appropriate mathematical programming technique. On the other hand, an EMO based on an evolutionary algorithm works with a population of individuals and attempts to find a set of nondominated solutions near the Pareto optimal front. Typically, EMO algorithms do not involve any preference information or scalarization of objectives.

Presently, EMO algorithms are widely used for solving multi-objective optimization problems. Some of the advantages that have fueled increased use of EMO algorithms include the following:

1) obtaining a set of nondominated solutions opposed to a single solution;
2) ease in handling problems with multiple local, discrete, and nonconvex Pareto optimal fronts;
3) flexibility in handling a wide range of types of variables, objective functions, and constraints, such as nonlinear, discontinuous.

Despite their wide success and applicability to many practical problems [7]–[11], several EMO algorithms (e.g., NSGA-II [12] and SPEA2 [13]) have shortcomings, which include the lack of theoretical convergence proof to the Pareto optimal front, slow convergence speed to the Pareto optimal front, and the lack of an efficient stopping criterion. Recently, efforts have been made to address these shortcomings with the aid of hybrid EMO algorithms [14]–[17].

In this paper, we refer to an algorithm that uses a local search in an EMO algorithm as a hybrid EMO algorithm. It is also referred to in the literature as a memetic multi-objective evolutionary algorithm [18]–[20]. In a hybrid EMO algorithm, an EMO algorithm plays the role of a global optimizer by searching the entire search space to find the most promising region(s) with a population of individuals, while a local search module locally improves the individuals of a population. Hybrid EMO algorithms have been shown in the

Manuscript received March 9, 2011; revised December 12, 2011, April 8, 2012, and May 14, 2012; accepted May 19, 2012. Date of publication June 12, 2012; date of current version July 25, 2013.

K. Sindhya and K. Miettinen are with the Department of Mathematical Information Technology, P.O. Box 35 (Agora), FI-40014, University of Jyväskylä, Finland (e-mail: karthik.sindhya@jyu.fi; kaisa.miettinen@jyu.fi).

K. Deb is with the Department of Mechanical Engineering, Indian Institute of Technology Kanpur, Kanpur 208016, India (e-mail: deb@iitk.ac.in).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TEVC.2012.2204403

literature to be more effective than EMO algorithms alone [14]–[17], [21].

In the literature, different procedures for a local search have been used in a hybrid EMO algorithm. Among them, the Pareto dominance scheme and the optimization of a weighted sum of objectives are common, which are briefly described as follows.

1) In the Pareto dominance scheme, a neighborhood $\Phi(A)$ is considered around an individual A chosen from the EMO population. Next, B individuals are considered in $\Phi(A)$ and the individual that is better in all objectives as compared to the other $B-1$ individuals replaces the individual A in a population. If more than one individual exists in $\Phi(A)$ which is better in all objectives, then one of them is chosen to replace the individual A in a population. The Pareto dominance scheme has been shown to be ineffective on problems with more than two objectives [22].
2) A multi-objective optimization problem can be transformed into a single objective optimization problem using a scalarizing function. A weighted sum of objective functions is one type of a scalarizing function. Here, a weighted sum of objective functions is formulated with predetermined weights and solved using an appropriate local search algorithm. The resulting locally optimal solution will replace an individual A chosen from a population, if the weighted sum of objective function values of the locally optimal solution is better than that of the individual A. A weighted sum of objective functions is known to be inappropriate in handling nonconvex problems [2].

A comprehensive literature review on hybrid EMO algorithms with different scalarizing functions considered is presented in [23].

In practice, there are many issues that may affect the performance of a hybrid EMO algorithm [24]. Some of them include the following:

1) the type of scalarization function used in a local search;
2) the frequency of a local search (probability of local search);
3) choice of individuals for a local search;
4) lack of an effective termination criterion for a hybrid EMO algorithm;
5) maintaining a balance between the modified exploration and exploitation introduced by a local search.

To tackle the above issues, in [17], we proposed a hybrid EMO approach with a probability of local search based on a saw-tooth-type probability function. A periodic increase and decrease of probability of local search was found to be useful in a hybrid EMO approach. In addition, our hybrid EMO approach used an achievement scalarizing function (ASF) [25] as a scalarization function in a local search, which can handle both convex and nonconvex problems and generate any Pareto optimal solution [2], [25]. In the successive study [23], we formulated our hybrid EMO approach into an algorithm and, in addition, proposed an efficient termination criterion based on the optimal values of ASFs. A significant reduction in the number of function evaluations as compared to the EMO algorithm alone was achieved. The EMO algorithm used was NSGA-II, and sequential quadratic programming (SQP) was used as a local search algorithm to solve ASFs. In [26], we proposed a hybrid EMO algorithm, where we used different weight factors (described in Section II) in an ASF for every local search. The study showed significant reduction in the number of function evaluations and diverse Pareto optimal solutions. Recently, in [27], the issue of choice of individuals for local search has been studied in detail on combinatorial problems. The authors reported a good performance when a local search is done on the individuals of a population before recombination operation.

In this paper, we continue considering hybrid approaches and propose a new modular hybrid EMO framework to tackle the issues affecting the performance of hybrid EMO algorithms previously mentioned. A hybrid EMO framework is an abstraction consisting of different functional modules synchronized to tackle the issues affecting the performance of a hybrid EMO algorithm. This framework lays a foundation upon which hybrid EMO algorithms can be effectively implemented. Our hybrid EMO framework includes six modules (steps) having distinct roles.

1) An EMO algorithm is used for solving a multi-objective optimization problem.
2) A project and cluster module is used for selecting individuals for a local search.
3) A local search module incorporating a local search is used to locally improve selected individuals of a population.
4) A diversity enhancement module is suggested for increasing the diversity of a population, when the need arises.
5) A termination module is used to stop the hybrid EMO algorithm.
6) A local search module incorporating a local search algorithm is used to locally improve all individuals of the final population. This is to guarantee at least the local convergence of the final population of a hybrid EMO algorithm.

A different local search algorithm can be used in Steps 3 and 6. For example, a derivative free local search algorithm, i.e., Nelder and Mead algorithm [28] in Step 3 can be used to reduce the number of function evaluations and a derivative-based algorithm (assuming differentiability), i.e., SQP in Step 6, can be used to guarantee at least the local convergence of the final population. In this paper, the same local search algorithm is used in Steps 3 and 6. The choice for a suitable local search algorithm is based on the characteristics (i.e., linear, nonlinear, differentiable, nondifferentiable, and so on) of the problem.

The framework is general and is independent of the type of problem (continuous or combinatorial). Here, we consider only multi-objective optimization problems with differentiable functions and continuous decision variables. An ASF is used for converting a multi-objective optimization problem into a single objective optimization problem and subsequently solved using SQP as a local search algorithm (because the functions involved are differentiable). One should note that an ASF can be formulated for any type of multi-objective optimization problem. However, a local search algorithm used to solve the single objective problem has to be correspondingly chosen depending on the characteristics of the problem.

The proposed hybrid EMO framework is modular and flexible because any clustering algorithm, diversity enhancement procedure, and a local search algorithm can be used within it. For instance, to enhance diversity, we propose a diversity enhancement module applicable to NSGA-II [5] and another diversity enhancement module applicable to MOEA/D [29] and MOEA/D-DRA [30]. Here, a population is projected on a hyperplane and clustered. Subsequently, individuals from different clusters are chosen to rebuild the parent population. A local search is used to locally improve an individual. In this paper, we show the efficacy of the hybrid EMO framework by considering three widely used EMO algorithms: NSGA-II, MOEA/D, and MOEA/D-DRA. The MOEA/D-DRA algorithm is based on an MOEA/D algorithm with dynamic resource allocation. The dynamic resource allocation provides different computational efforts for different subproblems in MOEA/D-DRA. To summarize, in this paper, we address the main issues that may be encountered during the implementation of a hybrid EMO algorithm using a modular hybrid EMO framework.

The remainder of this paper is organized as follows. In Section II, we introduce some concepts related to this paper on multi-objective optimization. Our hybrid EMO framework is presented in Section III with a detailed description of the steps involved in each of the modules. Then, in Section IV, we present a hybrid NSGA-II algorithm as an instance of our hybrid EMO framework and describe the test and parameter settings, parametric study, and results obtained by the hybrid NSGA-II algorithm in comparison with the original NSGA-II on several test problems. In this section, we also briefly discuss the effect of various modules of the hybrid EMO framework on the performance of the hybrid NSGA-II algorithm. In Section V, we present a hybrid MOEA/D algorithm as an instance of our hybrid EMO framework and show the efficacy of the algorithm using numerical experiments on several test problems. In Section VI, we present a hybrid MOEA/D-DRA algorithm as an instance of our hybrid EMO framework and show the efficacy of the algorithm on test problems with complicated Pareto sets. Finally, conclusions are drawn in Section VII with some future research directions.

II. Concepts

In this section, concepts on multi-objective optimization relevant to this paper are described. Here, we deal with multi-objective optimization problems of the form

$$\begin{array}{ll} \text{minimize} & \{f_1(\mathbf{x}), f_2(\mathbf{x}), \ldots, f_k(\mathbf{x})\} \\ \text{subject to} & \mathbf{x} \in S \subset \mathbb{R}^n \end{array} \tag{1}$$

with $k \geq 2$ conflicting objective functions $f_i : S \to \mathbb{R}$. We denote the vector of objective function values by $\mathbf{f}(\mathbf{x}) = (f_1(\mathbf{x}), f_2(\mathbf{x}), \ldots, f_k(\mathbf{x}))^T$ to be called an objective vector. The decision vectors $\mathbf{x} = (x_1, x_2, \ldots, x_n)^T$ belong to the search space S defined possibly with constraint functions.

Definition 1: A decision vector $\mathbf{x}^* \in S$ for problem (1) is a Pareto (optimal) solution, if there does not exist another $\mathbf{x} \in S$ such that $f_i(\mathbf{x}) \leq f_i(\mathbf{x}^*)$ for all $i = 1, 2, \ldots, k$ and $f_j(\mathbf{x}) < f_j(\mathbf{x}^*)$ for at least one index j. This definition also dictates global Pareto optimality.

Definition 2: Let $B(\mathbf{x}^*, \delta)$ denote an open ball with a center $\mathbf{x}^*$ and a radius $\delta > 0$, $B(\mathbf{x}^*, \delta) = \{ \mathbf{x} \in \mathbb{R}^\mathbf{n} \mid \|\mathbf{x}^* - \mathbf{x}\| < \delta\}$. A decision vector $\mathbf{x}^* \in S$ is said to be locally Pareto optimal if there exists $\delta > 0$ such that $\mathbf{x}^*$ is Pareto optimal in $S \cap B(\mathbf{x}^*, \delta)$. An objective vector is locally Pareto optimal if the decision vector corresponding to it is locally Pareto optimal.

In general, (1) typically has many Pareto optimal solutions. Additionally, (1) involves more than one objective; hence, scalarization techniques are commonly used in the MCDM field to obtain Pareto optimal solutions. The literature has a plethora of scalarization functions [2] and among them an achievement scalarizing function [2] and [25] is commonly used. It was proposed in [25] and uses a reference point $\bar{\mathbf{z}} \in \mathbb{R}^k$. In the context of this paper, reference points are objective vectors of individuals chosen for a local search.

An example of an achievement scalarizing function [2] and [25] is given by

$$\begin{array}{ll} \text{minimize} & \max_{i=1}^{k} w_i(f_i(\mathbf{x}) - \bar{z}_i) \\ & +\rho \sum_{i=1}^{k} w_i(f_i(\mathbf{x}) - \bar{z}_i) \\ \text{subject to} & \mathbf{x} \in S \end{array} \tag{2}$$

where $w_i = (\frac{1}{z_i^{\max} - z_i^{\min}})$ is a weight factor (usually used for normalizing) assigned to each objective function f_i. The maximum and minimum values of each objective function in a population are represented as $z_i^{\max}$ and $z_i^{\min}$, respectively. An augmentation coefficient ρ takes a small positive value (i.e., $\rho = 10^{-6}$). Problem (2) produces (properly) Pareto optimal solutions with bounded tradeoffs [2]. In other words, when (2) is solved using a suitable local search algorithm, the reference point $\bar{\mathbf{z}}$ is projected onto the Pareto front in the direction specified by the weight factor. In this paper, the direction of projection is the same for any local search in each generation.

Problem (2) has a nondifferentiable objective function, and hence, gradient-based solvers cannot be used. When all the functions in (1) are differentiable, we can utilize an equivalent differentiable formulation for (2) by introducing an extra real-valued variable (α) and k new constraints [2]

$$\begin{array}{ll} \text{minimize} & \alpha + \rho \sum_{i=1}^{k} w_i(f_i(\mathbf{x}) - \bar{z}_i) \\ \text{subject to} & w_i(f_i(\mathbf{x}) - \bar{z}_i) \leq \alpha \text{ for all } i=1, \ldots, k \\ \mathbf{x} \in S & \alpha \in \mathbb{R}. \end{array} \tag{3}$$

The advantages of using an achievement scalarizing function are the following [2].

1) The optimal solution of an achievement scalarizing function is always Pareto optimal.
2) Any (properly) Pareto optimal solution can be obtained by changing the reference point.

3) The optimal value of an achievement scalarizing function is zero, when the reference point is Pareto optimal.

These properties of achievement scalarizing functions are valid independently of the convexity of the problem. Because of these advantages, we propose to utilize achievement scalarizing functions in our hybrid EMO framework.

In this paper, we use NSGA-II and MOEA/D as EMO algorithms to test the efficacy of our hybrid EMO framework. The MOEA/D algorithm uses a weighted Chebyshev metric method [2] to formulate the following weighted Chebyshev problem:

$$\begin{aligned} \text{minimize} \quad & G(\mathbf{x}|\lambda, \mathbf{z^{min}}) = \\ & \max_{i=1}^{k} [\lambda_i (f_i(\mathbf{x}) - z_i^{min})] \\ \text{subject to} \quad & \mathbf{x} \in \mathcal{S} \end{aligned} \tag{4}$$

where $\lambda = (\lambda_i, \ldots, \lambda_k)$ is a weight vector assigned to each individual of a population. This weight of an individual is used to evaluate its fitness value by computing the corresponding weighted Chebyshev function (G).

III. Hybrid EMO Framework

In this section, we present the hybrid EMO framework for implementing hybrid EMO algorithms. Henceforth, we refer to the algorithms based on the hybrid EMO framework as hybrid EMO algorithms. The skeleton for the hybrid EMO framework is shown as a flowchart in Fig. 1.

The framework consists of six modules.

1) *EMO algorithm*: The main module contains the mating selection and the variation (crossover and mutation) steps, which are essential and integral components of many EMO algorithms.
2) *Project and cluster*: In this module, the individuals of a population are projected on a hyperplane and clustered in the objective space. Subsequently, a measure for cluster quality is calculated as an approximate quality measure of the diversity of a population. It is necessary to maintain a diverse population of individuals to explore the search space and ultimately obtain a diverse set of Pareto optimal solutions. The steps involved in the project and cluster module are enumerated in Algorithm 1.

   Any fast clustering algorithm can be used in Step 2 of this module. In Fig. 2, the circles 1, 2, 3, 4, and 5 represent individuals of population P_t and squares a, b, and c are their orthogonal projections on a hyperplane and belong to the population $P_t^{'}$. It can be seen that although 2, 3, and 4 are different individuals in P_t, when projected on a hyperplane, they are represented by the same individual b in $P_t^{'}$. Additionally, all the individuals of $P_t^{'}$ form a single cluster in Fig. 2. A local search on any individual in $P_t^{'}$ will produce the same Pareto optimal solution PO. It is important to note that if every local search produces an identical Pareto optimal solution, the diversity of a population can decrease, which is discussed later in the diversity enhancement module. If the cluster quality index Q has a small value, the individuals in $P_t^{'}$ are very close and form one cluster. The population $P_t^{'}$ can be considered diverse, when Q has a large value. In Fig. 2, since the individuals of $P_t^{'}$ belong to a single cluster, Q will have a very low value. It is essential to maintain a well spread population $P_t^{'}$ to obtain different Pareto optimal solutions representing different regions of the Pareto optimal front.

   The index Q at generation t, Q-current is used to calculate a lower bound, $Q_{\text{bound}} = 0.25 \cdot Q$-current. The

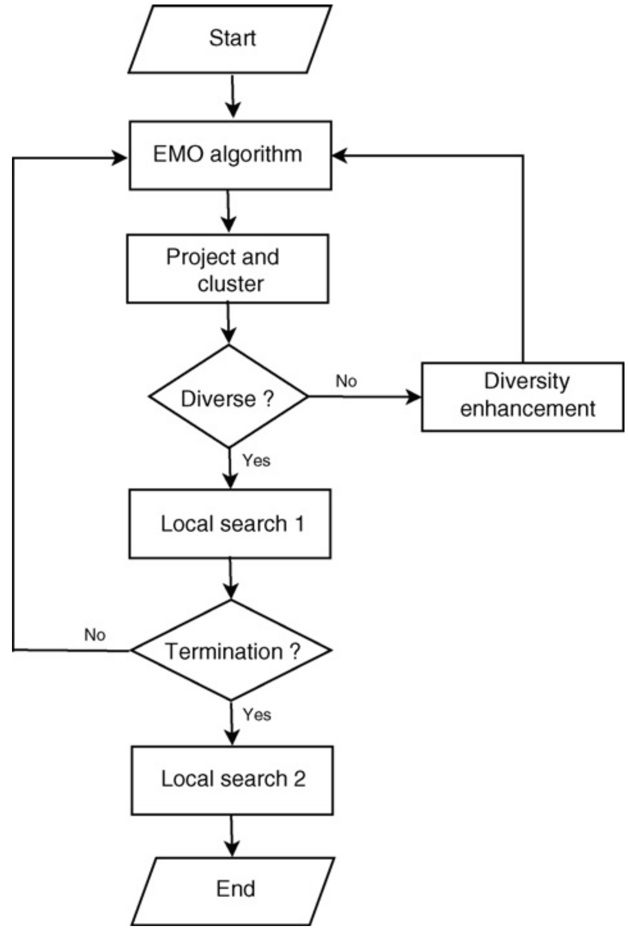

Fig. 1. Hybrid EMO framework.

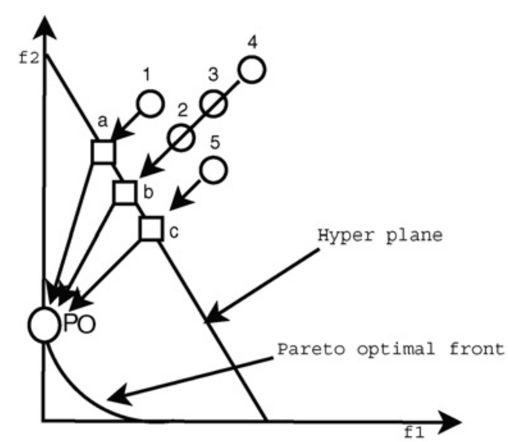

Fig. 2. Projection of individuals on a hyperplane.

Algorithm 1 Project and cluster module

Input to the module: Size of population, N; number of clusters, K; and population at generation t, P_t

Output from the module: Cluster quality index at generation t, Q-current; and clusters, RP_i^t, $i = 1, 2, \ldots, K$ at generation t

Step 1. Consider a set Ψ consisting of objective vectors, $F^i = (0, \ldots, 0, f_i^{max}, 0, \ldots, 0)^T \in \mathbb{R}^k$, where f_i^{max} is the maximum function value of objective function f_i in a population P_t, $f_i^{max} \in \mathbb{R} \setminus 0$ and $i = 1, \ldots, k$. Next, we define a hyperplane, $H = \{\mathbf{f} \in \mathbb{R}^k : \langle \mathbf{w}^S, \mathbf{f} \rangle + b^S = 0\}$, where, H is a linear subspace of dimension $k-1$ in a k-dimensional Euclidean space, $\langle \cdot, \cdot \rangle$ is Euclidean dot product in $\mathbb{R}^k \times \mathbb{R}^k$, $\Psi \subset H$, $\mathbf{w}^S = (\frac{1}{f_1^{max}}, \ldots, \frac{1}{f_k^{max}})$ and $b^S = -1$. An orthogonal projection, PI of an objective vector $\mathbf{f}$ of an individual onto H is given by

$$PI = \frac{1 - \langle \mathbf{w}^S, \mathbf{f} \rangle}{\|\mathbf{w}^S\|^2} \mathbf{w}^S + \mathbf{f}. \quad (5)$$

Using (5), project all individuals of population P_t orthogonally onto H to get population $P_t^{'}$. In (5), if $f_i^{max} = 0$, set $f_i^{max} = 10^{-06}$.

Step 2. Cluster $P_t^{'}$ into K clusters. The subpopulation in each cluster is denoted by RP_i^t, $i = 1, 2, \ldots, K$.

Step 3. Evaluate the diversity of $P_t^{'}$ using the following cluster quality index [31]:

$$Q = \sum_{i=1}^{K} \frac{1}{|RP_i^t|} \sum_{C_j \in RP_i^t} (D(C_j, \sigma_i)),$$

where K is the number of clusters, σ_i represents the cluster centroid of the cluster i and C_j represents a point in the cluster i. Furthermore, $D(C_j, \sigma_i)$ is the Euclidean distance of the point in the cluster i to its centroid and $|RP_i^t|$ is the number of individuals in the cluster i.

Step 4. Replace the subpopulation in each cluster with the corresponding individuals in P_t.

Q_{bound} calculated at generation t is used as a lower bound for the next t_c generations. Starting from the $(t+1)$th generation, at every generation Q-current is compared to Q_{bound} until the $(t+t_c)$th generation is reached. During these t_c generations, if Q-current is lower than Q_{bound}, a population is declared to have a bad diversity, and otherwise it is declared to have a good diversity. When a population is declared to have a bad diversity, the diversity enhancement module (described later in this section) is subsequently activated. In this module, we suggest using Q_{bound} instead of Q-current at generation t as a lower bound for the next t_c generations because when a population of a hybrid EMO algorithm converges close to the Pareto optimal front, the individuals of a population are only redistributed along the nondominated front. Hence, Q-current may oscillate around the lower bound during the t_c generations leading, in turn, to an oscillation between a good and a bad diversity. Hence, by using Q_{bound} we can prevent undue oscillations and subsequently prevent activation of the diversity enhancement module when a population is in the proximity of the Pareto optimal front. The value of 25% is used as a heuristic based on our empirical studies.

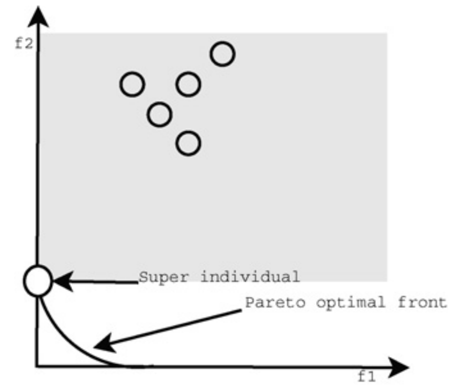

Fig. 3. Creation of superindividuals.

Algorithm 2 Local search 1 module

Input to the module: Population at generation t, P_t; clusters, RP_i^t, $i = 1, 2, \ldots, K$ at generation t; and probability of local search, p_{local}.

Output from the module: Improved population at generation t, $P_t^{'}$.

Step 1. Select a random cluster i and choose an individual A from RP_i^t with a probability p_{local}. If no individual is chosen from RP_i^t, terminate the module.

Step 2. Set $\bar{\mathbf{z}} = \mathbf{f}$ for (3), where $\mathbf{f}$ is the chosen objective vector of A. Next perform a local search by formulating (3) and solving it using any appropriate local search algorithm.

Step 3. Replace the individual in P_t corresponding to A in RP_i^t with a solution obtained from the local search. The new population with the new solution is referred to as $P_t^{'}$.

Step 4. When e.g. NSGA-II is used as an EMO algorithm, reassign ranks and crowding distances to the individuals of the population $P_t^{'}$.

3) *Local search 1*: The steps involved in the local search 1 module are shown in Algorithm 2. This is the first module involving a local search in the hybrid EMO framework. A local search is used to speed up the convergence rate of a hybrid EMO algorithm with a probability p_{local}. The local search 1 module is activated only when the population has been declared to have a good diversity in the project and cluster module. It can be seen that in Algorithm 2, only one individual is subjected to a local search in any generation, as a local search on multiple individuals can consume function evaluations.

4) *Diversity enhancement*: In a hybrid EMO algorithm, during a local search it may happen that an individual is generated far from the current population, as shown in Fig. 3. We call this a superindividual. A superindividual may dominate the entire population, as shown by the grey area in Fig. 3. Because of its superiority, the

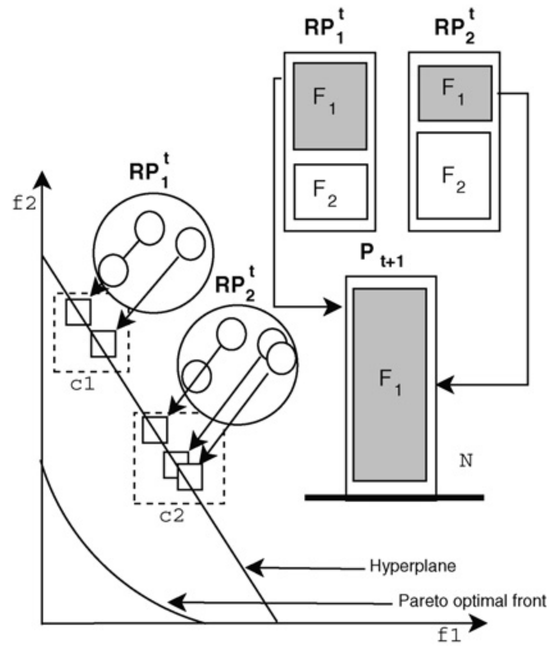

Fig. 4. Diversity enhancement in NSGA-II.

emphasis given to this individual will be higher than the remaining individuals in a population. The increased selection pressure of such superindividuals will cause an expeditious deletion of dominated solutions. Because the diversity enhancement of EMO algorithms only works on nondominated solutions, as a result, the lateral diversity can be lost. Hence, appropriate diversity enhancement is necessary to restore lateral diversity in the population. In addition, the diversity in the objective space is necessary for the efficient working of the hybrid EMO framework so that the local search 1 module has distinct reference points to obtain solutions in different regions of the Pareto optimal front. In this module, the user may use either the diversity enhancement corresponding to the EMO algorithm used in the EMO algorithm module or anything that can enhance the lateral diversity.

Next, we present a diversity enhancement algorithm that is applicable to the NSGA-II algorithm in Algorithm 3. Fig. 4 shows the steps involved in the diversity enhancement module for the NSGA-II algorithm. The same diversity enhancement procedure can be used for other generational EMO algorithms provided they maintain an explicit parent and offspring population. In Fig. 4, RP_1^t and RP_2^t are the two subpopulations identified in the population R_t by the project and cluster module corresponding to clusters $c1$ and $c2$. A nondominated sorting was performed to identify fronts $\mathcal{F}_1$ and $\mathcal{F}_2$ in RP_1^t and RP_2^t. Subsequently, all the individuals in $\mathcal{F}_1$ from RP_1^t and RP_2^t were used to obtain P_{t+1}. Here, the total number of individuals in $\mathcal{F}_1$ from RP_1^t and RP_2^t was equal to N; hence, no individuals in $\mathcal{F}_2$ from RP_1^t and RP_2^t were needed.

In Algorithm 4, we present a diversity enhancement algorithm that is applicable to the MOEA/D algorithm.

Algorithm 3 Diversity enhancement module

Input to the module: Combined population $R_t = P_t \cup Q_t^{'}$, at iteration t; and population size, N.

Output from the module: Parent population at iteration $t+1$, P_{t+1}.

Step 1. Send R_t of size 2·N to the project and cluster module.

Step 2. Perform nondominated sorting to each subpopulation RP_j^t and identify different fronts $\mathcal{F}_i^j$, $i = 1, 2, ...,$ etc. and $j = 1, 2, \ldots, K$.

Step 3. Initialize $P_{t+1} = \phi$, $i = 1$ and $j = 1$. As long as $|P_{t+1}| + |\mathcal{F}_i^j| \leq N$ perform $P_{t+1} = P_{t+1} \cup \mathcal{F}_i^j$, $j = j + 1$, and if $j > K$, set $j = 1$ and $i = i + 1$.

Step 4. Reassign ranks and crowding distances to P_{t+1} as in the NSGA-II algorithm. For a detailed procedure for assigning ranks and crowding distances to a population, see [5].

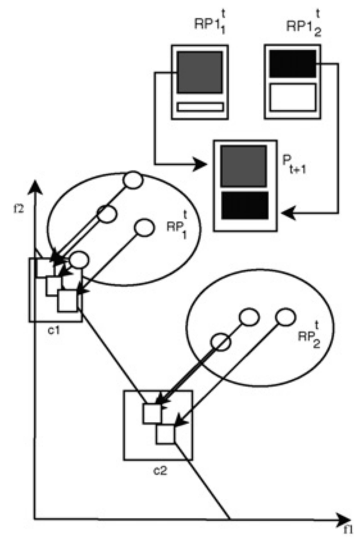

Fig. 5. Diversity enhancement in MOEA/D.

Fig. 5 shows the steps involved in this algorithm. After the two subpopulations RP_1^t and RP_2^t are identified by the project and cluster module, the subpopulations are classified as best and worst in Step 2 of Algorithm 4. These classified subpopulations are referred to as $RP1_1^t$ and $RP1_2^t$. The individuals in each subpopulation are sorted in ascending order with respect to their respective value of G in Step 4. Based on the number of individuals that can be chosen from subpopulations $RP1_1^t$ and $RP1_2^t$, as calculated in Step 3 of Algorithm 4, a new population P_{t+1} of size N is obtained. Here, we prefer to have more individuals from a subpopulation which has an individual with the best value of G in P_{t+1}.

5) *Termination criterion*: Many hybrid EMO algorithms are terminated after a prefixed number of generations or when no new individuals have entered the nondominated set after a prefixed number of generations. These stopping criteria involve large computational costs and do not gauge the convergence of a hybrid EMO algorithm to the Pareto optimal front. Hence, there is

Algorithm 4 Diversity enhancement module - MOEA/D

Input to the module: Combined population $R_t = P_t \cup Q_t$, at iteration t; and population size, N.

Output from the module: Parent population at iteration $t+1$, P_{t+1}.

Step 1. Send R_t of size $|Q_t| + N$ to the project and cluster module, where $|Q_t|$ is the number of individuals in Q_t.

Step 2. Find the best solution in each subpopulation RP^t_j based on the value of G. Using the best solution from each subpopulation classify the subpopulation from best cluster to worst cluster, i.e., $RP1^t_1$ is the subpopulation which contains an individual in R_t with the best value of G and $RP1^t_K$ is the subpopulation which contains an individual in R_t with the worst value of G.

Step 3. The number of individuals from each subpopulation $RP1^t_j$ to rebuild the parent population at iteration $t+1$, P_{t+1} is calculated as, $R_j = N\frac{1-r}{1-r^{K+1}}r^{j-1}$. If for any j, $R_j > \left|RP1^t_j\right|$, then redistribute the number of solutions to other subpopulations such that $\sum_{j=1}^{K} R_j = N$.

Step 4. For every $j = 1, \ldots, K$, sort the individuals of subpopulation $RP1^t_j$ in an ascending order based on the value of G and choose R_j best number of individuals. The R_j best individuals in $RP1^t_j$ is shown by grey and black rectangles in Fig. 5.

Step 4. Initialize $P_{t+1} = \phi$, $i = 1$ and $j = 1$. Perform $P_{t+1} = P_{t+1} \cup \eta_{i,j}$, where $\eta_{i,j} \in RP1^t_j$, increment $i = i+1$, until $i \leq R_j$. If $j > K$, go to Step 5, else increment $j = j+1$.

Step 5. Stop.

a need for a stopping criterion within the hybrid EMO framework that is automatic and ensures an adequate convergence close enough to the Pareto optimal front. A local search using an ASF as a scalarizing function provides a termination criterion for a hybrid EMO algorithm. The optimal value of an ASF Ω_t at every generation t is stored in an archive. Then, a moving average $\Gamma_t = \frac{1}{\Lambda}\sum_{j=t-\Lambda+1}^{t} \Omega_t$ is calculated at a generation t (after Λ generations), where Λ is a pre-fixed number of generations. If $\Gamma_t \leq \epsilon$, where ϵ is a pre-fixed small positive scalar, at any generation t, the hybrid EMO algorithm is terminated. The efficacy of this termination criterion has been previously demonstrated in [23]. The resulting final population is sent to the local search 2 module.

If only a strict budget of function evaluations is allowed, due to the restrictions in computational time available, a termination criterion based on the maximum number of function evaluations can be used in this module. In such a case, the local search 2 module may have a sufficient number of function evaluations to guarantee convergence of the final population (at least locally).

6) *Local search 2*: This is the last module in the hybrid EMO framework. The steps involved in the local search 2 module are enumerated in Algorithm 5. Here, a local search is applied on all individuals of the population obtained so far by considering each individual as a reference point of (3) at a time. The local search 2 module guarantees at least the local Pareto optimality of the final population.

Algorithm 5 Local search 2 module

Input to the module: Population of size N.

Output from the module: Final population of size N which is at least locally Pareto optimal.

Step 1. Set $i = 1$. Let $\mathbf{f}^i$ be the objective vector of the i^{th} individual. As long as $i \leq N$, perform a local search by formulating (3) using a reference point $\bar{\mathbf{z}} = \mathbf{f}^i$. Set $i = i+1$.

Step 2. Replace the input population with the new population obtained from Step 1 to get a final population.

In this section, we presented a hybrid EMO framework that can be used to implement a hybrid EMO algorithm. The local search 1 module enhances the convergence speed and the diversity enhancement module enhances the diversity of the population when there is a lapse in diversity. A bad diversity is detected using a cluster quality index calculated in the project and cluster module. Finally, the local search 2 module ensures at least the local Pareto optimality of the final population of a hybrid EMO algorithm.

IV. Numerical Experiments Using NSGA-II as an EMO algorithm

In this section, we first present an instance of our hybrid EMO framework using NSGA-II as an EMO algorithm and call it the hybrid NSGA-II (HN) algorithm. Subsequently, we present the results and analysis with the parameter settings used for the numerical experiments to show the advantage of using the HN algorithm against the NSGA-II algorithm alone. In addition, we also show the effectiveness of different modules of the hybrid EMO framework, i.e., cluster quality index, project and cluster, and enhanced diversity preservation modules on the HN algorithm.

In Algorithm 6, we present the HN algorithm. In the HN algorithm, Steps 1–7 are the same as in the NSGA-II algorithm. Next, the population of the HN algorithm is sent to the project and cluster module in Step 8, where the population is clustered and diagnosed for diversity, i.e., good or bad. If the diversity of the population is good, a local search is performed in Step 10 of the HN algorithm by activating the local search 1 module. Otherwise, in Step 11, the diversity of the population is enhanced by activating the diversity enhancement module. Finally, when the termination criterion is satisfied in Step 12, the HN algorithm is terminated. In the presented HN algorithm, we have not incorporated the local search 2 module so that the quality of the final population can be compared against the NSGA-II algorithm. Furthermore, in the termination criterion module, we use the maximum number of function evaluations as a termination criterion to be able to compare the performance of the HN algorithm to the NSGA-II algorithm. Normally, the local search 2 module must be included in the HN algorithm.

Algorithm 6 HN algorithm

Step 1: Generate a random initial population P_0 of size N and set generation count $t = 0$.
Step 2: Sort population to different nondomination levels (fronts) and assign each solution a fitness equal to its nondomination level (1 is the best level).
Step 3: Create offspring population $Q_t^{'}$ of size N using binary tournament selection, recombination and mutation operations.
Step 4: Combine the parent and the offspring populations and create $R_t = P_t \cup Q_t^{'}$.
Step 5: Perform nondominated sorting to R_t and identify different fronts F_i, $i = 1, 2, ...$ etc.
Step 6: Set a new population $P1_{t+1} = \phi$. Set a count $i = 1$ and as long as $|P1_{t+1}| + |F_i^j| \leq N$, perform $P1_{t+1} = P1_{t+1} \cup F_i$ and $i = i+1$. Here, $|P1_{t+1}|$ and $|F_i|$ represent the cardinality of $P1_{t+1}$ and F_i, respectively.
Step 7: Perform the crowding-sort procedure and include the most widely spread (N-$|P1_{t+1}|$) members of F_i by using the crowding distance values in the sorted F_i to $P1_{t+1}$.
Step 8: Send population $P1_{t+1}$ to the project and cluster module and get $P1_{t+1}^{'}$.
Step 9: If the population $P1_{t+1}^{'}$ has a good diversity, go to Step 10, else go to Step 11
Step 10: Send $P1_{t+1}^{'}$ to the local search 1 module and go to Step 12.
Step 11: Send $P1_{t+1}^{'}$ to the diversity enhancement module.
Step 12: Check if the termination criterion is satisfied. If yes, go to Step 13, else set $t = t + 1$ and return to Step 2.
Step 13: Stop.

A. Experiments

Numerical experiments were carried out to test the performance of the HN algorithm against the NSGA-II algorithm. Here, the SQP in KNITRO [32] was used in the HN algorithm as a local search algorithm to solve ASFs in the local search 1 module. A set of eight test problems of different types was considered from the ZDT [5] and DTLZ test suites [33]. The problems were: 1) bi-objective, ZDT4; 2) three objectives, DTLZ $(1, \ldots, 6)$; and 3) four objectives, DTLZ2 (DTLZ24). In this paper, we chose ZDT4 as the only bi-objective problem since we are interested in demonstrating the efficacy of the HN algorithm on problems with more than two objectives and many EMO algorithms are efficient for bi-objective problems and do not, thus, need hybridization. On the other hand, the ZDT4 test problem involves a large number of local Pareto optimal fronts and can be challenging to any hybrid EMO algorithm.

Next, we present the parameter settings used for all subsequent tests.

1) Performance metric: Inverted generational distance (IGD) [34] is used as a performance metric as it provides a measure of both proximity and diversity of the nondominated solutions in the objective space with respect to the Pareto optimal front. If P^* is the set of uniformly distributed Pareto optimal solutions in the objective space and P the obtained approximation set of nondominated solutions in the objective space from an EMO algorithm, the IGD value for the approximation set is defined by
$$IGD(P, P^*) = \frac{\sum_{\mathbf{v} \in P^*} d(\mathbf{v}, P)}{|P^*|} \tag{6}$$
where $d(\mathbf{v}, P)$ is the minimum Euclidean distance between $\mathbf{v}$ and points in P and $|P^*|$ is the number of points in P^*.
2) Stopping criterion is the maximal number of function evaluations (FE). Since the test problems used in our study are of varying computational complexity, we need different FE values for different problems. Based on an empirical study, we have pre-evaluated the minimum number of function evaluations that may be necessary for the nondominated solutions from an EMO algorithm to be in the proximity of the Pareto optimal front for different test problems to be: a) 5000 for DTLZ2, DTLZ4, and DTLZ5; b) 15 000 for DTLZ6; c) 20 000 for ZDT4, DTLZ1, and DTLZ24; and d) 50 000 for DTLZ3.
3) The number of independent runs = 15.

The HN and NSGA-II algorithms involve the following parameters to be set, the values of which are commonly used in the literature.

1) Population size: 100 for bi-objective problems and 200 for three and four objectives for both algorithms.
2) Crossover and mutation probability: 0.9 for crossover and 1/number of variables for mutation for both the algorithms.
3) SBX and mutation distribution indices: For the HN algorithm, SBX and mutation distribution indexes are set as 10 and 20, respectively, for the bi-objective problem and as 15 and 20, respectively, for $k > 2$. For the NSGA-II algorithm, SBX and mutation distribution indices are set as 15 and 20, respectively, for all test problems.
4) Probability of local search (p_{local}): A heuristic for setting p_{local} is suggested in the next subsection with the aid of numerical experiments on test problems.
5) Local search termination criterion: The local search terminates with the KNITRO's termination criterion (KKT optimality error).

In the next subsection, we present the results of study of the parameter p_{local}, i.e., the frequency of local search and present some heuristics for setting its value.

B. Parametric Study

Using the aforementioned parameter settings for the HN algorithm, a parametric study was carried out using different values of $p_{\text{local}} = 1/(\eta \cdot N)$, where $\eta = 1, \ldots, 5$ and N is the population size. In other words, we varied p_{local} so that the HN algorithm can have approximately one local search for every ηth generation.

The results with the eight test problems are tabulated in Table IV. Table IV contains the median IGD values for different p_{local} values at different FE function evaluations

TABLE I
SUMMARY OF THE STUDY TO FIND THE BEST p_{local} FOR THE HN ALGORITHM

Test Problem	*FE*	$p_{\text{local}} = 1/(\eta \cdot N)$ IGD values (smaller value is better)				
		$\eta = 1$	$\eta = 2$	$\eta = 3$	$\eta = 4$	$\eta = 5$
ZDT4(2 objectives)	20 000	0.0055	0.0079	0.0053	0.0051	0.0050
DTLZ1(3 objectives)	20 000	0.4405	*0.3990*	**0.3741**	0.4644	0.5741
DTLZ2(3 objectives)	5000	0.0569	0.0574	0.0554	0.0552	0.0557
DTLZ3(3 objectives)	50 000	**0.2000**	1.1730	*0.6793*	1.0520	1.0477
DTLZ4(3 objectives)	5000	0.0572	0.0562	0.0540	0.0531	0.0536
DTLZ5(3 objectives)	5000	0.0049	0.0061	0.0057	0.0049	0.0050
DTLZ6(3 objectives)	15 000	**0.1604**	0.7873	0.3968	*0.2384*	0.4841
DTLZ24(4 objectives)	20 000	0.1201	0.1184	0.1195	0.1205	0.1195

across all the chosen test problems. Here, only the median values are shown as they depict a general trend of the results. In addition, variance of the median IGD values for every test problem across all the p_{local} values was calculated. The best and the second best IGD values are marked in bold and italics, respectively, only for those test problems which have a variance higher than 10^{-03}. It can be seen that IGD values are marked in bold face only in three test problems [DTLZ (1, 3 and 6)], which implies that the HN algorithm is not very sensitive to the parameter p_{local}. Problems DTLZ1 and DTLZ3 have multiple local Pareto optimal fronts, and a lower value of p_{local} was useful for the HN algorithm to move faster toward the Pareto optimal front. When $p_{\text{local}} = 1/(\eta \cdot N)$ ($\eta = 1$ and 3), the first and the second best IGD values arise in three test problems altogether. On one hand, a higher p_{local} means a larger number of local search and hence there is a higher probability to create more superindividuals as described in Section III. On the other hand, a lower p_{local} means a lower number of local search, which in turn will not significantly enhance the convergence speed of the HN algorithm as compared to the NSGA-II algorithm. Therefore, we propose to consider $p_{\text{local}} = 1/(3 \cdot N)$, which is also a median value of all the η values considered.

C. Study of the Efficacy of the HN Algorithm Against the NSGA-II Algorithm

In this section, we present the results obtained from the comparative studies between the NSGA-II and HN algorithms. The nondominated solutions from both algorithms were recorded after a number of *FE1* and *FE2* function evaluations and their IGD values were calculated.

In Table II, we summarize the results from the study. The number of function evaluations used for termination for different test problems is summarized as follows.

1) ZDT4(2 objectives): *FE1* = 20 000 and *FE2* = 21 000.
2) DTLZ1(3 objectives): *FE1* = 20 000 and *FE2* = 30 000.
3) DTLZ2(3 objectives): *FE1* = 5000 and *FE2* = 10 000.
4) DTLZ3(3 objectives): *FE1* = 30 000 and *FE2* = 40 000.
5) DTLZ4(3 objectives): *FE1* = 5000 and *FE2* = 10 000.
6) DTLZ5(3 objectives): *FE1* = 5000 and *FE2* = 10 000.
7) DTLZ6(3 objectives): *FE1* = 15 000 and *FE2* = 20 000.
8) DTLZ24(4 objectives): *FE1* = 20 000 and *FE2* = 30 000.

We can decipher the following information from Table II.

1) In all the test problems, the HN algorithm has a lower median IGD value, which is an indication of both closeness to the Pareto optimal front and spread of the nondominated solutions.
2) In the test problems ZDT4, DTLZ1, and DTLZ3, multiple local fronts exist. The nondominated solutions of the HN algorithm have significantly lower IGD values as compared to the NSGA-II algorithm after both *FE1* and *FE2* function evaluations. The IGD values using the HN algorithm for ZDT4, DTLZ1, and DTLZ3 are around 26, 71, and 80 percentage lower, respectively, than the IGD values for NSGA-II with *FE1* function evaluations. This clearly indicates the merit of the hybrid EMO framework based HN algorithm to handle problems with multiple local fronts. The local search on solutions from different clusters and diversity enhancement helped the HN algorithm achieve a significantly faster convergence without compromising diversity.
3) The test problems DTLZ2, DTLZ4, DTLZ5, and DTLZ24 appear to be comparatively easier when compared to the other test problems, as the nondominated solutions from NSGA-II are able to converge near the Pareto optimal front in *FE1* function evaluations. In these test problems, both algorithms have nearly the same IGD values. The reason for the comparable performance could be the diversity enhancement module that tries to enhance the diversity of the population by adding dominated solutions to the population, simultaneously maintaining a balance between convergence and diversity. However, it can be observed from Table II that the IGD values for even such problems are better for the HN algorithm after both *FE1* and *FE2* function evaluations.
4) The test problem DTLZ6 has no local Pareto optimal fronts, but is designed to be a harder problem as compared to DTLZ5 by changing the g function [33]. From Table II, it can be observed that NSGA-II has a higher IGD value even after *FE2* function evaluations. However, the nondominated solutions from the HN algorithm could easily get closer to the Pareto optimal front as compared to nondominated solutions from the NSGA-II algorithm already in *FE1* function evaluations. This faster convergence can be attributed to the presence of the local search 1 module.

D. Study of the Effect of the Cluster Quality Index on the Population

As mentioned in the hybrid EMO framework, the HN algorithm uses the cluster quality index to predict whether the diversity is good or bad. In Figs. 6 and 7, we plot the evolution of the cluster quality index with generations for ZDT4 and DTLZ2. These problems were chosen to show the behavior of the cluster quality index for problems with and without multiple local fronts, respectively. The cluster quality index in any generation (Q-current) is represented by a circle and the lower bound (Q-bound), which is used to decide about the diversity of a population, is represented by the bold black

TABLE II
COMPARISON OF THE IGD VALUES FOR THE NSGA-II AND HN ALGORITHM (SMALLER VALUE IS BETTER)

Test		*FE1*		*FE2*	
Problem	Type	Original NSGA-II	HN NSGA-II	Original NSGA-II	HN NSGA-II
ZDT4(2-obj)	Best	0.0064	0.0049	0.0056	0.0047
	Median	0.0143	**0.0053**	0.0112	**0.0052**
	Worst	0.1001	0.0148	0.0683	0.0088
DTLZ1(3-obj)	Best	0.4627	0.0823	0.0486	0.0185
	Median	1.3167	**0.3741**	0.2768	**0.0546**
	Worst	3.0039	1.3280	0.8126	0.7833
DTLZ2(3-obj)	Best	0.0537	0.0482	0.0484	0.0475
	Median	0.0616	**0.0554**	0.0505	**0.0499**
	Worst	0.0684	0.0649	0.0530	0.0565
DTLZ3(3-obj)	Best	1.1158	0.0595	0.1766	0.0511
	Median	3.0437	**1.0520**	0.8353	**0.0596**
	Worst	5.9207	3.7220	2.7572	1.0122
DTLZ4(3-obj)	Best	0.0569	0.0499	0.0466	0.0461
	Median	0.0653	**0.0540**	0.0503	**0.04907**
	Worst	0.0829	0.0680	0.0524	0.0510
DTLZ5(3-obj)	Best	0.0102	0.0032	0.0035	0.0029
	Median	0.0133	**0.0057**	0.0040	**0.0030**
	Worst	0.0168	0.0090	0.0046	0.0035
DTLZ6(3-obj)	Best	1.8807	0.0035	1.2435	0.0029
	Median	2.0057	**0.3937**	1.3365	**0.0564**
	Worst	2.2175	2.1015	1.5137	0.4681
DTLZ24(4-obj)	Best	0.1144	0.1155	0.1150	0.1151
	Median	0.1204	**0.1195**	0.1191	**0.1182**
	Worst	0.1321	0.1256	0.1247	0.1242

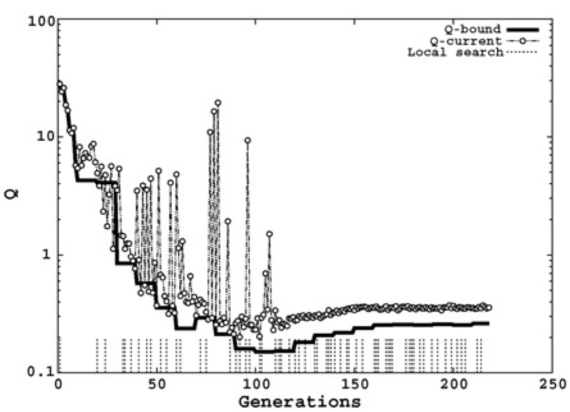

Fig. 6. Q evolution over generations for ZDT4.

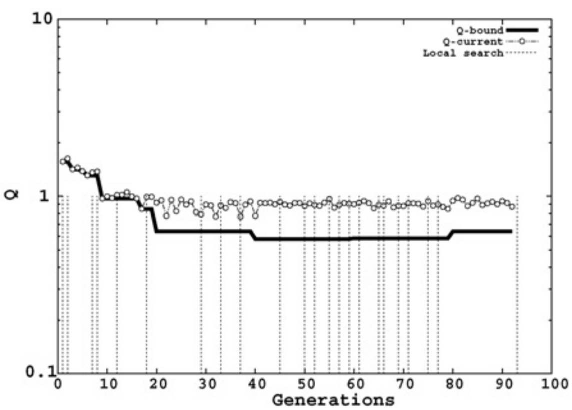

Fig. 7. Q evolution over generations for DTLZ2.

line. The vertical dotted lines represent the local search in a particular generation.

In Figs. 6 and 7, it can be seen that in the first generation, Q-current has a high value, implying that the initial population is diverse and with successive generations Q-current decreases and stabilizes when the Pareto optimal front is reached. The evolution of Q-current follows a zig–zag path in ZDT4, as shown in Fig. 6. The intermediate spikes in Q-current are due to the diversity enhancement by the diversity enhancement module. The loss in diversity for the population may occur because of the local search, and the solutions may get stuck in the locally Pareto optimal fronts. It can be seen in Fig. 6 that the diversity enhancement module has worked with the crossover and mutation operators on the population to increase the diversity. The above study clearly indicates the strength of the HN algorithm to prevent the loss in diversity of the population. In Fig. 7, since in DTLZ2 there are no multiple local fronts, we do not see a similar trend as in ZDT4. In addition, the HN algorithm, for most of the generations, has maintained a good diversity and hence, the enhanced diversity preservation module, after the 20th generation, is not activated. In the figure, the diversity enhancement module is not activated if Q-bound is lower than Q-current.

E. Study of the Effect of the Local Search 1 Module

For this paper, we make a small change in the HN algorithm. Instead of the local search 1 module, a local search is performed on randomly chosen individuals from the population with a probability p_{local}. This change should help us under-

TABLE III
COMPARISON OF THE IGD VALUES FOR THE HN-V1 AND HN ALGORITHMS (SMALLER VALUE IS BETTER)

Test Problem	*FE*	HN-v1	HN
ZDT4	20 000	0.0071	**0.0053**
DTLZ1	20 000	0.4757	**0.3741**
DTLZ2	5000	0.0570	**0.0554**
DTLZ3	60 000	0.2953	**0.0596**
DTLZ4	5000	0.0567	**0.0540**
DTLZ5	5000	0.0087	**0.0057**
DTLZ6	20 000	0.1041	**0.0564**
DTLZ24(4-obj)	20 000	0.1199	**0.1195**

stand the importance of choosing individuals from different clusters. The HN algorithm, with this change, is referred to here as the HN-v1 algorithm. The median IGD values from the test runs with the same previously mentioned parameter settings are tabulated in Table III.

It can be observed that in every case, the IGD values for the HN algorithm are better than for the HN-v1 algorithm. Choosing an individual from different clusters for local search makes a bigger difference for the DTLZ1, DTLZ3, and DTLZ6 test problems. As seen before in this section, these test problems are also difficult for the NSGA-II algorithm. For these test problems, the convergence of the population closer to the Pareto optimal front can be due to the presence of the project and cluster and the local search 1 modules in the HN algorithm.

F. Study of the Dynamics of the Diversity Enhancement Module in the HN Algorithm

In this section, we consider a sample run of the HN algorithm and show the evolution of the population with and without the diversity enhancement module (discussed in Section III) for the ZDT4 test problem. The population for the HN algorithm with and without the diversity enhanced operator is plotted for generations 1, 4, 5, and 8 in Figs. 8–11, respectively.

The initial population is shown in Fig. 8, where it can be seen that most of the population is biased toward low values of f_1 in the objective space, i.e., left side of the figure. Hence, the population may develop a tendency to drift toward the left part of the objective space by generating more solutions in that region. In generations 4 (Fig. 9) and 5 (Fig. 10), we can see that without the diversity enhancement module the population has drifted toward the left and thereby lost diversity in the objective space, as it can be easy for the operators to generate solutions there and also because of the locally improved solutions arising from the local search 1 module. With the diversity enhancement module, the population has still retained a better diversity. It can also be seen in Fig. 10 that the objective space diversity is maintained in the HN algorithm with the diversity enhancement module, but the convergence speed to the Pareto optimal front has decreased. Subsequently, in generation 8 (Fig. 11), the HN algorithm with the diversity enhancement module has improved the convergence speed,

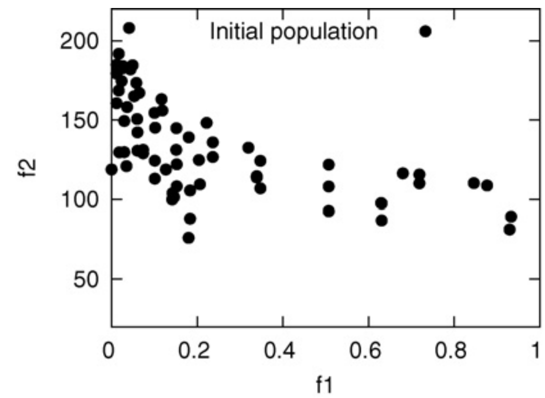

Fig. 8. Population in generation 1 for ZDT4.

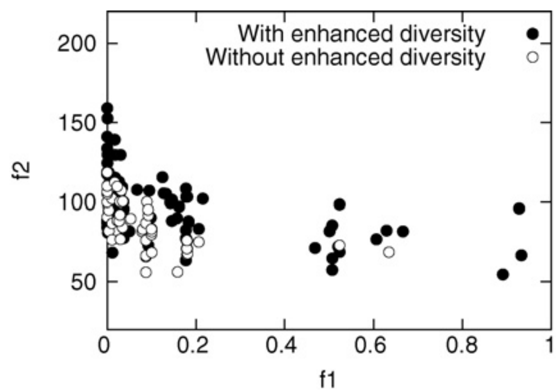

Fig. 9. Population in generation 4 for ZDT4.

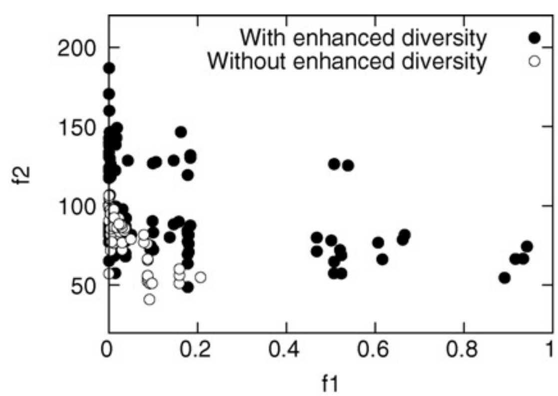

Fig. 10. Population in generation 5 for ZDT4.

and the population is in the same range as the population without the diversity enhancement module besides having better diversity. Thus, this paper clearly demonstrates the importance of the diversity enhancement module in a hybrid EMO algorithm.

V. NUMERICAL EXPERIMENTS USING MOEA/D AS AN EMO ALGORITHM

In this section, we first present an instance of our hybrid EMO framework using MOEA/D as an EMO algorithm and call it the hybrid MOEA/D (HMOEA/D) algorithm. Subsequently, we present the results and analysis with the parameter

Algorithm 7 Hybrid MOEA/D algorithm

Step 1. Generate a uniform spread of N weight vectors: $\{\lambda^1, \ldots, \lambda^N\}$. Here $\lambda^N = (\lambda_1^N, \ldots, \lambda_k^N)$.
Step 2. For every weight vector λ^i, set $B(i) = J_1, \ldots, J_T$, where $J_1, \ldots, J_T$, are the indexes of the T closest weight vectors.
Step 3. Generate a random initial population P_0 of size N.
Step 4. Initialize a vector $\mathbf{z^{min}}$, where z_i is the best function value for objective f_i of P_0. Set generation count $t = 1$.
Step 5. Set counter $k = 1$.
Step 6. Select J_k and randomly two other indices, $J_c, J_d \in B(k)$, and generate a new solution y from the individuals $I_{J_k}, I_{J_c}, I_{J_d} \in P_t$ by using differential evolution [35] genetic operators.
Step 7. Apply a repair operator on y to produce y^1.
Step 8. For $j = 1, \ldots, m$, if z_j is greater than the j^{th} function value of y^1, update z_j with the j^{th} function value of y^1.
Step 9. For each index $k1 \in B(k)$, if $G(y^1, \lambda^{k1}) \leq G(I_{k1}, \lambda^{k1})$, then, if $I_{k1} \cap Q_t = \emptyset$, $Q_t = Q_t \cup I_{k1}$ and $I_{k1} = y^1$ else $Q_t = Q_t \cup I_{k1}$ and $I_{k1} = y^1$.
Step 10. If $k < N$, $k = k + 1$ and go to Step 6.
Step 11. Send P_t to the project and cluster module.
Step 12. If P_t has good diversity, send P_t to the local search 1 module, else go to Step 13.
Step 13. Create a combined population $R_t = P_t \cup Q_t$ and send R_t to the diversity enhancement module.
Step 14. Check if the termination criterion is satisfied. If yes, go to Step 15, else set $t = t + 1$ and return to Step 5.
Step 15. Stop

settings used for the numerical experiments to show the advantage of using the HMOEA/D algorithm against the MOEA/D algorithm alone. In addition, we also show the evolution of cluster quality index and IGD value with generations.

A complete description of the HMOEA/D algorithm is provided in Algorithm 7. In the HMOEA/D algorithm, Steps 1–10 are almost the same as in the MOEA/D algorithm. We collect the deleted solutions in Step 9 of HMOEA/D algorithm in Q_t. In Step 11 of the HMOEA/D algorithm, the population P_t is sent to the project and cluster module. Here, P_t is clustered and checked for diversity. If the diversity of P_t is good, a local search is performed in Step 12 of the algorithm, otherwise in Step 9 of the HMOEAD algorithm, we combine the populations P_t and Q_t to obtain a new population R_t and send to the diversity enhancement module. Here, we retain the termination condition of the HN algorithm, i.e., the maximum number of function evaluations for the same reason as before, i.e., to enable comparison. Additionally, as in Algorithm 6, we omitted the local search 2 module in Algorithm 7. Normally, the local search 2 module must be included in the HMOEA/D algorithm.

A. Experiments

The HMOEA/D algorithm was coded in MATLAB [36] and numerical experiments were carried out to demonstrate the efficacy of the HMOEA/D algorithm against the MOEA/D algorithm alone. Here, SQP in MATLAB was used as a local search algorithm to solve ASFs in the local search 1 module. The same set of eight test problems and the performance metric used in Section IV were considered.

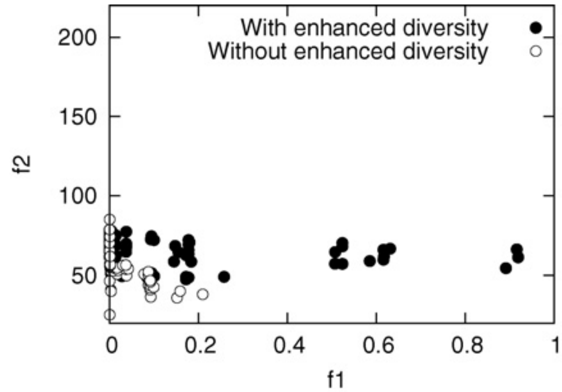

Fig. 11. Population in generation 8 for ZDT4.

The MOEA/D and HMOEA/D algorithms involve the following parameters to be set.

1) Population size: 101 for bi-objective problems, 210 for three objective, and 220 for four objective problems.
2) Parameters of DE genetic operators: F=0.5 and CR=0.5.
3) Local search termination criterion: The maximum number of SQP solver iterations is limited to 2 to prevent excessive consumption of function evaluations during local search in the local search 1 module.
4) The number of independent runs = 15.

Next, we present the results of the parametric study to set the parameter p_{local}, with the efficacy studies of the HMOEA/D algorithm as against the MOEA/D algorithm. Different values of $p_{\text{local}} = 1/(\eta \cdot N)$, where $\eta = 10, 15$, and 20 and N is the population size, were used to test for bi-objective and three-objective test problems, whereas for four-objective test problems $\eta = 5, 10$ and 15 were used. In other words, we varied p_{local} so that the HMOEA/D algorithm can have approximately one local search for every ηth generation.

The results with the number of function evaluations allowed are tabulated in Table IV. They contain the best, median and worst IGD values for different η values and function evaluations. The HMOEA/D algorithm has performed better than MOEA/D algorithm in six (ZDT4, DTLZ1, DTLZ2, DTLZ4, DTLZ5, DTLZ6, and DTLZ24) out of eight test problems considered. The best median IGD value is marked in the bold face in Table IV. We can decipher the following information from Table IV.

1) In ZDT4, a bi-objective test problem, $\eta = 15$, i.e., approximately one local search for every 15 generations has a lower median IGD value. In addition, in 10 000 function evaluations, the best performing alternative using HMOEA/D has almost 50% lower IGD value as compared to the MOEA/D algorithm. This indicates that the HMOEA/D algorithm is significantly faster in reaching the Pareto optimal front as compared to the MOEA/D algorithm.
2) In test problems with three objective functions, $\eta = 20$, i.e., approximately one local search for every 20 generations has a lower median IGD value. It can also be seen

TABLE IV

COMPARISON STUDY OF HMOEA/D ALGORITHM AS AGAINST MOEA/D ALGORITHM

Test Problem	Algorithm Type	η	Function Evaluations	Best IGD Value	Median IGD Value	Worst IGD Value
ZDT4	HMOEA/D	10	10 000	0.007029	0.038531	0.167672
		15	10 000	0.006767	**0.017964**	0.196249
		20	10 000	0.006750	0.028978	0.199995
		10	15 000	0.004218	0.005286	0.032980
		15	15 000	0.004322	**0.005017**	0.012908
		20	15 000	0.004358	0.005725	0.014977
	MOEA/D		10 000	0.008942	0.039809	0.209749
			15 000	0.004568	0.012266	0.066401
DTLZ1	HMOEA/D	10	15 000	0.018076	0.019863	1.688604
		15	15 000	0.018213	0.019682	0.314513
		20	15 000	0.018213	**0.019673**	0.302368
		10	20 000	0.017877	0.018138	1.185334
		15	20 000	0.017838	**0.018265**	0.258085
		20	20 000	0.017838	**0.018265**	0.258058
	MOEA/D		15 000	0.018073	0.047081	0.606179
			20 000	0.017904	0.018925	0.322391
DTLZ2	HMOEA/D	10	2500	0.059420	0.063697	0.089782
		15	2500	0.059420	0.063697	0.112639
		20	2500	0.059420	**0.063620**	0.112639
		10	5000	0.051056	0.053009	0.054204
		15	5000	0.051129	0.053009	0.056725
		20	5000	0.050939	**0.052276**	0.05672
	MOEA/D		2500	0.057491	0.065458	0.070515
			5000	0.052191	0.054032	0.059670
DTLZ3	HMOEA/D	10	40 000	0.092140	0.103249	1.113672
		15	40 000	0.092017	0.105007	1.120615
		20	40 000	0.092017	0.109029	4.179735
		10	45 000	0.091545	0.096442	0.143814
		15	45 000	0.091160	0.097805	1.111167
		20	45 000	0.091160	0.100073	4.167857
	MOEA/D		40 000	0.092197	**0.095910**	1.124986
			45 000	0.091469	**0.093488**	0.299136
DTLZ4	HMOEA/D	10	10 000	0.052352	0.066016	0.939872
		15	10 000	0.051193	0.065710	0.939872
		20	10 000	0.051193	**0.061100**	0.939872
		10	15 000	0.049606	0.058414	0.939872
		15	15 000	0.049606	0.056942	0.939872
		20	15 000	0.049606	**0.056702**	0.939872
	MOEA/D		10 000	0.052822	0.070444	0.939872
			15 000	0.050191	0.055246	0.939872
DTLZ5	HMOEA/D	10	5000	0.006520	0.007145	0.008985
		15	5000	0.006520	0.007253	0.008831
		20	5000	0.006520	**0.007252**	0.008831
		10	10 000	0.005786	0.006259	0.006587
		15	10 000	0.005917	**0.006264**	0.006552
		20	10 000	0.005836	**0.006264**	0.006558
	MOEA/D		5000	0.007258	0.008560	0.009402
			10 000	0.006255	0.006443	0.006748
DTLZ6	HMOEA/D	10	2500	0.005195	0.011811	0.973373
		15	2500	0.005195	0.011811	0.974845
		20	2500	0.005195	0.011811	0.974845
		10	5000	0.005021	0.005619	0.139825
		15	5000	0.005021	0.005671	0.139825
		20	5000	0.005021	0.005671	0.145838
	MOEA/D		2500	0.004926	**0.006698**	0.991955
			5000	0.005210	**0.005551**	0.005934
DTLZ24	HMOEA/D	5	10 000	0.162310	0.168081	0.182709
		10	10 000	0.160819	0.167084	0.178488
		15	10 000	0.160819	**0.167022**	0.178434
		5	15 000	0.154628	**0.160423**	0.172503
		10	15 000	0.155786	0.160591	0.167712
		15	15 000	0.154207	0.158917	0.170875
	MOEA/D		10 000	0.160008	0.167442	0.179930
			15 000	0.154086	0.160612	0.1810131

in Table IV that the IGD values for η = 20 and η = 15 are close, and hence either of them can be considered as a safe choice.

3) In DTLZ3, the median IGD values for the MOEA/D algorithm are slightly better than HMOEA/D algorithm. The reason for slightly higher median IGD values can be due to the presence of a number of local Pareto optimal fronts in DTLZ3. On local search, individuals can be stuck in these local basins of attraction. But, the best IGD values for η = 15 and η = 20 for HMOEA/D algorithm are slightly better than those of the MOEA/D algorithm.

4) In DTLZ6, the median IGD values for the MOEA/D algorithm are better than those of the HMOEA/D algorithm. Here, the Pareto optimal front is a curve and we suspect the unnecessary activation of the diversity enhancement module to be the cause for higher median IGD values for the HMOEA/D algorithm. However, the worst IGD values for the HMOEA/D algorithm are lower than those of the MOEA/D algorithm.

5) In DTLZ24, the median IGD values for HMOEA/D are slightly better than those of MOEA/D. It can also be seen in Table IV that either η = 10 or η = 15 can be considered as a safe choice.

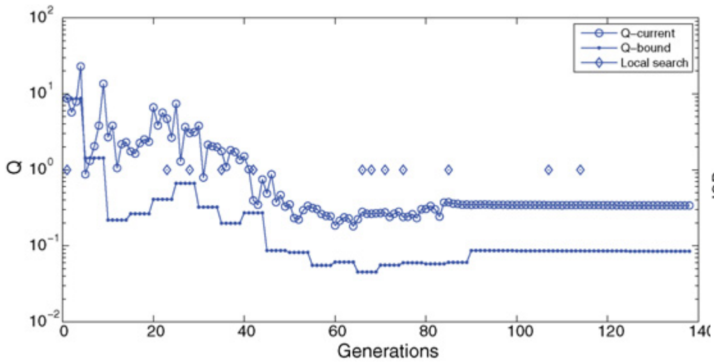

Fig. 12. Q evolution over generations for ZDT4 using the HMOEA/D algorithm.

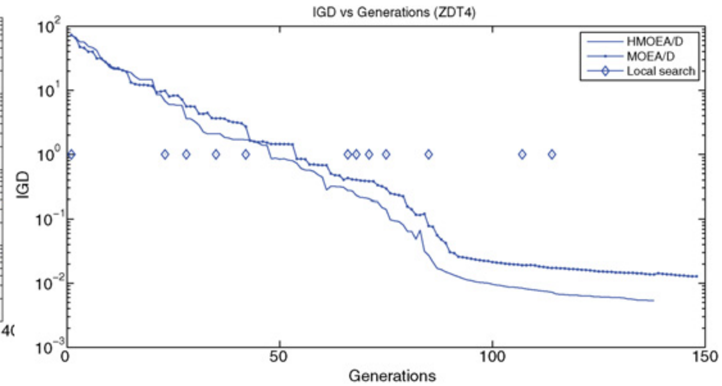

Fig. 13. Convergence rate of HMOEA/D as against MOEA/D algorithm (ZDT4).

6) In Table IV, it can be seen that $\eta = 10$ results with higher IGD values for most values. Hence, too frequent activation of the local search 1 module can be considered a deterrent to the best performance of the HMOEA/D algorithm.

As mentioned in the hybrid EMO framework, the HMOEA/D algorithm uses a cluster quality index to predict the diversity of a population to be good or bad.

In Fig. 12, we plot the evolution of the cluster quality index with generations for ZDT4. Here, the cluster quality index (Q-current) in any generation is represented by a circle. During the initial generations, Q-current has a high value indicating that the initial population is diverse. When the first local search 1 module was activated, the population lost diversity (Q-current fell below Q-bound). Subsequently, the diversity enhancement module is activated and the diversity in population is restored. After generation 90, Q-current stabilizes indicating that the population has reached the Pareto optimal front.

Next, we consider a sample run of the HMOEA/D and MOEA/D algorithms and show the evolution of IGD value with generations in Fig. 13. It can be seen in Fig. 13 that during the initial generations the MOEA/D algorithm is faster than the HMOEA/D algorithm. This behavior is due to the activation of the diversity enhancement module (see Fig. 12). After generation 20, local search 1 module is activated multiple times; hence, the IGD values of HMOEA/D decrease rapidly. The same behavior can also be seen after generation 50. Fig. 13 clearly shows the benefit of employing the HMOEA/D algorithm as against the MOEA/D algorithm.

VI. Numerical Experiments Using MOEA/D-DRA as an EMO Algorithm

In this section, we further extend our hybrid EMO framework using MOEA/D-DRA [30] as an EMO algorithm in Algorithm 8. The hybrid MOEA/D-DRA is referred to as the HMOEA/D-DRA algorithm. Here, we show the advantage of using HMOEA/D-DRA algorithm versus MOEA/D-DRA algorithm alone with numerical experiments on test problems from the CEC 2009 test suite [37]. The CEC 2009 test suite consists of test problems with complicated Pareto sets, i.e., nonlinear dependencies between decision variables. The MOEA/D-DRA algorithm [30] was shown as a high-performing algorithm in the CEC 2009 multi-objective optimization algorithm competition [37].

A complete algorithmic description of the MOEA/D-DRA algorithm is provided in [30]. In the MOEA/D algorithm, every subproblem is given equal importance, i.e., they receive the same amount of computational effort. In the MOEA/D-DRA [30] algorithm proposed by Zhang *et al.*, different computational efforts were allotted to different subproblems. Here, a utility is computed for each subproblem and the computational effort is distributed based on the utility.

In Algorithm 8, we describe the HMOEA/D-DRA algorithm. The prominent changes in Algorithm 8 as compared to the MOEA/D-DRA algorithm are as follows.

1) In Step 7, we send the population P_t to the project and cluster module to estimate the diversity of the population.
2) In Step 9, we collect the deleted individuals in Q_t and subsequently combine with population P_t, if there is a lapse in diversity. The combined population R_t is sent to the diversity enhancement module to replenish the diversity.
3) In Step 7, if the diversity is found to be good, the population P_t is sent to the local search 1 module in Step 12.

Additionally, as in Algorithm 6, we do not incorporate the local search 2 module. Normally, the local search 2 module must be included in the HMOEA/D-DRA algorithm. Next, we present the results of the numerical experiments to show the efficacy of the HMOEA/D-DRA algorithm.

A. Experiments

The MOEA/D-DRA algorithm programmed in MATLAB provided by Zhang *et al.* was used to integrate the hybrid framework. The resulting program was used to demonstrate the efficacy of the HMOEA/D-DRA algorithm versus the MOEA/D-DRA algorithm. As in Section V, SQP in MATLAB was used in the local search 1 module 1.

The test problems and the parameter settings used are the following.

1) Test problems:

   a) CEC 2009 test suite [37]: UF1 (bi-objective), UF2 (bi-objective), UF3 (bi-objective), UF7 (bi-objective), UF8 (three objectives), UF10 (three objectives).
2) Population size: $N = 600$ for two and three objectives.
3) Number of weight vectors in the neighborhood $(T) = 0.1N$ and $n_r = 0.01N$.
4) Probability for selecting the mating/update range $(\delta) = 0.9$.
5) Genetic operators: DE crossover ratio (CR) = 1.0 and DE scaling factor (F) = 0.5, mutation distribution index $(\eta) = 20$ and mutation probability $(p_m) = 1/n$.
6) Stopping criteria: Maximum number of function evaluations:
   a) UF1 = 50 000, UF2 = 50 000, UF3 = 50 000, UF7 = 50 000, UF8 = 100 000, and UF10 = 100 000.
7) The first local search is activated at $l = 10$th generation. Subsequently, local search is activated every $2l$th generation provided the diversity is good. This heuristic was fixed based on an empirical study.
8) Total number of independent runs = 15.

In the numerical experiments, we limited the number of function evaluations to be lower than that used in the CEC 2009 competition [37] to gauge the speed of convergence of the HMOEA/D-DRA algorithm versus the MOEA/D-DRA algorithm. The CEC 2009 test suite consists of both differentiable and nondifferentiable multi-objective optimization problems. In our numerical experiments, we have chosen only the differentiable multi-objective optimization problems, due to the use of a gradient based solver (SQP) in the local search 1 module. In addition, the three five-objective problems in the CEC test suite could not be used due to the lack of available implementation in MATLAB and problems in software-related issues. For calculating the IGD values, we use only 100 individuals for bi-objective problems and 150 individuals for three objective problems from the final population as suggested in [30]. The algorithm choosing the individuals from the final population is described in [30].

The results are tabulated in Algorithm 8 with the best median IGD values marked in bold. In test problems UF1, UF2, and UF3, the HMOEA/D-DRA algorithm outperformed the MOEA/D-DRA algorithm and is at par with MOEA/D-DRA algorithm on a three-objective UF10 test problem. In test problems UF7 and UF8, the MOEA/D-DRA algorithm has attained a better median IGD value as compared to the HMOEA/D-DRA algorithm. We suspect the degradation in performance in some test problems is due to the lack of an explicit diversity preservation operator to maintain the diversity of solutions along the Pareto optimal front in the MOEA/D-DRA algorithm. However, it can be seen from Algorithm 8 that the HMOEA/D-DRA algorithm could yield the best IGD value for both UF7 and UF8 in 15 test runs.

Algorithm 8 Hybrid MOEA/D-DRA algorithm

Step 1. Generate a uniform spread of N weight vectors: $\{\lambda^1, \dots, \lambda^N\}$. Here $\lambda^N = (\lambda_1^N, \dots, \lambda_k^N)$.
Step 2. For every weight vector λ^i, set $B(i) = J_1, \dots, J_T$, where $J_1, \dots, J_T$, are the indexes of the T closest weight vectors.
Step 3. Generate a random initial population P_0 of size N.
Step 4. Initialize a vector $\mathbf{z^{min}}$, where z_i is the best function value for objective f_i of P_0.
Step 5. Initialize utility $\pi^i = 1$, $i = 1, \dots, N$. Set generation count $t = 1$.
Step 6. Select subproblems for search: An initial I is formed by selecting the indexes of the subproblems whose objectives are the individual objectives f_i. Subsequently, select other $[N/5] - k$ indexes and add them to I using a 10-tournament selection.
Step 7. Send P_t to the project and cluster module.
Step 8. Set counter $k = 1$.
Step 9. Generate a random number, *rand* (uniformly distributed on the open interval (0, 1)). Set $\xi_t = B(i)$, if $rand < \delta$, else, set $\xi_t = \{1, \dots, N\}$.
Step 10. Select J_k and randomly two other indices, $J_c, J_d \in \xi_t$, and generate a new solution y from the individuals $I_{J_k}, I_{J_c}, I_{J_d} \in \xi_t$ by using genetic operators as suggested in [30].
Step 11. Apply a repair operator on y to produce y^1 as suggested in [30] to be within the boundary.
Step 12. For $j = 1, \dots, m$, if z_j is greater than the j^{th} function value of y^1, update z_j with the j^{th} function value of y^1.
Step 13. Set $c = 0$ and then perform the following: (a) If $c = n_r$ or $\xi_t = \emptyset$, go to Step 14, else, pick an index $k1$ from ξ_t randomly. (b) If $G(y^1, \lambda^{k1}) \leq G(I_{k1}, \lambda^{k1})$, then, if $I_{k1} \cap Q_t = \emptyset$, $Q_t = Q_t \cup I_{k1}$ and $I_{k1} = y^1$ else $Q_t = Q_t \cup I_{k1}$ and $I_{k1} = y^1$. $c = c + 1$. (c) Delete $k1$ from ξ_t and go to (a).
Step 14. If $k < |I|$, $k = k + 1$ and go to Step 9.
Step 15. If P_t has good diversity, send P_t to the local search 1 module, else go to Step 16.
Step 16. Create a combined population $R_t = P_t \cup Q_t$ and send R_t to the diversity enhancement module.
Step 17. Check if the termination criterion is satisfied. If yes, go to Step 18, else set $t = t+1$, if t is a multiplication of 50, update the utility as suggested in [30] and return to Step 6.
Step 18. Stop

VII. Conclusion and Future Research Directions

In this paper, we presented a hybrid EMO framework to enhance the performance of hybrid EMO algorithms. The hybrid EMO framework addressed several issues that may affect the performance of a hybrid EMO algorithm. The salient features of our hybrid EMO framework can be briefly summarized as follows.

1) The type of scalarization function: A hybrid EMO algorithm usually involves a local search module to locally improve individuals in a population. To achieve a fast, directed, and accurate convergence, we used an ASF to be solved by an appropriate local search algorithm. An ASF has many advantages, such as any Pareto optimal solution can be obtained by changing the reference point unlike the weighted sum of the objectives function and the optimal solution of an ASF is always (properly)

TABLE V
COMPARISON OF THE IGD VALUES FOR THE MOEA/D-DRA AND HMOEA/D-DRA ALGORITHM (SMALLER VALUE IS BETTER)

Test Problem	Type	MOEA/D-DRA	HMOEA/D-DRA
UF1(2-obj)	Best	0.01750	0.01502
	Median	0.0320	**0.0220**
	Worst	0.0650	0.0349
UF2(3-obj)	Best	0.0136	0.0133
	Median	0.0211	**0.0198**
	Worst	0.1227	0.0590
UF3(3-obj)	Best	0.0170	0.0326
	Median	0.0899	**0.0577**
	Worst	0.1601	0.1176
UF7(3-obj)	Best	0.0121	0.00917
	Median	**0.0152**	0.02114
	Worst	0.2958	0.6930
UF8(3-obj)	Best	0.0678	0.0655
	Median	**0.0839**	0.0983
	Worst	0.0972	0.1924
UF10(3-obj)	Best	0.4930	0.4738
	Median	0.5923	**0.5920**
	Worst	0.6507	0.7381

Pareto optimal; hence, the convergence of the solutions can be guaranteed. In addition, from a theoretical point of view, ASF based local search can handle any number of objectives, which gives this procedure a distinct advantage over a local search procedure consisting of a Pareto dominance scheme.

2) Choice of individuals for local search: The projection of individuals of a population on a hyperplane and then clustering them enabled us to pick individuals in an effective manner for local search to generate Pareto optimal solutions in different regions of the Pareto optimal front.
3) Frequency of local search: The frequency with which a local search was done greatly affected the balance between exploration and exploitation. In this paper, we used a probability of local search to control the number of individuals of a population improved by a local search in a generation of a hybrid EMO algorithm.
4) Termination criterion: The optimal solution of an ASF was always zero, when a reference point was Pareto optimal. This fact can be used to have an efficient termination criterion for hybrid EMO algorithms.
5) Maintaining a balance between exploration and exploitation: The above issues were mainly connected to exploitation as they provided convergence properties to our hybrid EMO framework. The exploitation was shown to be efficiently balanced by exploration using an enhanced diversity preservation module in our hybrid EMO framework. The framework has an automatic switching mechanism based on the cluster quality index to switch off the local search module and enhance diversity of the individuals of a population when there is a lapse in diversity.

Clustering of population provided three prominent advantages to our hybrid EMO framework: 1) the cluster quality index derived from the clusters helped the hybrid algorithm to assess the diversity of a population; 2) Pareto optimal solutions in different regions of the Pareto optimal front can be obtained by choosing individuals from different clusters; and 3) diversity of the population can be restored if there is a lapse in diversity.

We showed the efficacy of the proposed hybrid EMO framework with numerical experiments by considering the widely used EMO algorithm, NSGA-II, and two recently proposed EMO algorithms, MOEA/D and MOEA/D-DRA as examples. The hybridization proposed involved applying a single objective optimization method, which was appropriate to the problem characteristics in question. In this paper, we used a sequential quadratic programming method that was applicable to continuously differentiable functions. Hence, only continuously differentiable functions were considered for numerical experiments. However, gradient-free methods could be used as well. The MOEA/D-DRA algorithm was chosen to show the efficacy of a hybrid algorithm based on our hybrid framework on test problems with nonlinear interdependences between decision variables. We showed that an algorithm based on our hybrid EMO framework can achieve faster convergence toward the Pareto optimal front without loss in diversity, which was also our goal in this research. Further research topics include development of the diversity enhancement module, a better diversity loss indicator, and a self-adaptive probability of local search for hybrid EMO algorithms.

ACKNOWLEDGMENT

The authors would like to thank Dr. S. Äyrämö for his valuable comments and suggestions on clustering. Also, the discussions with T. Haanpää on projections in higher dimensions were immensely helpful.

REFERENCES

[1] V. Chankong and Y. Y. Haimes, *Multiobjective Decision Making Theory and Methodology*. New York: Elsevier Science, 1983.
[2] K. Miettinen, *Nonlinear Multiobjective Optimization*. Boston, MA: Kluwer, 1999.
[3] Y. Sawaragi, H. Nakayama, and T. Tanino, *Theory of Multiobjective Optimization*. Orlando, FL: Academic Press, 1985.
[4] C. A. C. Coello, G. B. Lamont, and D. A. V. Veldhuizen, *Evolutionary Algorithms for Solving Multi-Objective Problems*, 2nd ed. New York: Springer, 2007.
[5] K. Deb, *Multi-Objective Optimization Using Evolutionary Algorithms*. Chichester, U.K.: Wiley, 2001.
[6] J. Branke, K. Deb, K. Miettinen, and R. Slowinski, Eds., *Multiobjective Optimization: Interactive and Evolutionary Approaches*. Berlin/Heidelberg, Germany: Springer, 2008.
[7] A. Chipperfield and P. Fleming, "Multi-objective gas turbine controller design using genetic algorithms," *IEEE Trans. Ind. Electron.*, vol. 43, no. 5, pp. 583–587, Oct. 1996.
[8] C. A. Coello Coello and G. B. Lamont, *Applications of Multi-Objective Evolutionary Algorithms*. Singapore: World Scientific, 2004.
[9] A. L. Jaimes and C. A. Coello Coello, "Multi-objective evolutionary algorithms: A review of the state-of-the-art and some of their applications in chemical engineering," in *Multi-Objective Optimization: Techniques and Applications in Chemical Engineering*, G. P. Rangaiah, Ed. Singapore: World Scientific, 2009, pp. 61–90.
[10] A. L. Medaglia, J. G. Villegas, and D. M. Rodriguez-Coca, "Hybrid bi-objective evolutionary algorithms for the design of a hospital waste management network," *J. Heuristics*, vol. 15, no. 2, pp. 153–176, 2009.
[11] C. Tapia and C. A. Coello Coello, "Applications of multi-objective evolutionary algorithms in economics and finance: A survey," in *Proc. CEC*, 2007, pp. 532–539.
[12] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan, "A fast and elitist multi-objective genetic algorithm: NSGA-II," *IEEE Trans. Evol. Comput.*, vol. 6, no. 2, pp. 182–197, Apr. 2002.

[13] E. Zitzler, M. Laumanns, and L. Thiele, "SPEA2: Improving the strength Pareto evolutionary algorithm for multi-objective optimization," in *Proc. Evol. Methods Des. Optimization Control Applicat. Ind. Problems*, 2002, pp. 95–100.
[14] M. P. Hansen, "Use of substitute scalarizing functions to guide a local search based heuristic: The case of moTSP," *J. Heuristics*, vol. 6, no. 3, pp. 419–431, 2000.
[15] H. Ishibuchi and T. Murata, "A multi-objective genetic local search algorithm and its applications to flowshop scheduling," *IEEE Trans. Syst. Man Cybern. C: Applicat. Rev.*, vol. 28, no. 3, pp. 392–403, Aug. 1998.
[16] A. Jaszkiewicz, "Genetic local search for multi-objective combinatorial optimization," *Eur. J. Oper. Res.*, vol. 137, no. 1, pp. 50–71, 2002.
[17] K. Sindhya, K. Deb, and K. Miettinen, "A local search based evolutionary multi-objective approach for fast and accurate convergence," in *Proc. 10th Parallel Problem Solving From Nature (PPSN X)*, 2008, pp. 815–824.
[18] H. Ishibuchi, T. Yoshida, and T. Murata, "Balance between genetic search and local search in memetic algorithms for multi-objective permutation flowshop scheduling," *IEEE Trans. Evol. Comput.*, vol. 7, no. 2, pp. 204–223, Apr. 2003.
[19] J. D. Knowles and D. W. Corne, "M-PAES: A memetic algorithm for multi-objective optimization," in *Proc. CEC*, 2000, pp. 325–332.
[20] A. Lara, G. Sanchez, C. A. C. Coello, and O. Schütze, "HCS: A new local search strategy for memetic multi-objective evolutionary algorithms," *IEEE Trans. Evol. Comput.*, vol. 14, no. 1, pp. 112–132, Feb. 2010.
[21] E.-G. Talbi, M. Rahoual, M. H. Mabed, and C. Dhaenens, "A hybrid evolutionary approach for multicriteria optimization problems: Application to the flow shop," in *Proc. Evol. Multi-Criterion Optimization*, 2001, pp. 416–428.
[22] K. Ikeda, H. Kita, and S. Kobayashi, "Failure of Pareto-based MOEAs: Does nondominated really mean near to optimal?" in *Proc. CEC*, 2001, pp. 957–962.
[23] K. Sindhya, K. Deb, and K. Miettinen, "Improving convergence of evolutionary multi-objective optimization with local search: A concurrent-hybrid algorithm," *Natural Comput.*, vol. 10, no. 4, pp. 1407–1430, 2011.
[24] H. Ishibuchi and K. Narukawa, "Some issues on the implementation of local search in evolutionary multi-objective optimization," in *Proc. Genet. Evol. Comput.*, 2004, pp. 1246–1258.
[25] A. P. Wierzbicki, "The use of reference objectives in multi-objective optimization," in *Multiple Criteria Decision Making: Theory and Application*, G. Fandel and T. Gal, Eds. Berlin/Heidelberg, Germany: Springer, 1980, pp. 468–486.
[26] K. Sindhya, K. Miettinen, and K. Deb, "An improved concurrent-hybrid algorithm for enhanced diversity and accuracy in evolutionary multi-objective optimization," in *Evolutionary and Deterministic Methods for Design, Optimization and Control, Applications to Industrial and Societal Problems*, T. Burczynski and J. Periaux, Eds. Barcelona, Spain: CIMNE, 2011, pp. 182–187.
[27] H. Ishibuchi, Y. Hitotsuyanagi, Y. Wakamatsu, and Y. Nojima, "How to choose solutions for local search in multi-objective combinatorial memetic algorithms," in *Proc. Parallel Problem Solving From Nature (PPSN)*, 2010, pp. 516–525.
[28] J. A. Nelder and R. Mead, "A simplex method for function minimization," *Comput. J.*, vol. 7, no. 4, pp. 308–313, 1965.
[29] Q. Zhang and H. Li, "MOEA/D: A multi-objective evolutionary algorithm based on decomposition," *IEEE Trans. Evol. Comput.*, vol. 11, no. 6, pp. 712–731, Dec. 2007.
[30] Q. Zhang, W. Liu, and H. Li, "The performance of a new version of MOEA/D on CEC09 unconstrained MOP test instances," in *Proc. CEC*, 2009, pp. 203–208.
[31] A. Jagota, *Microarray Data Analysis and Visualization*. Santa Cruz, CA: Bioinformatics by the Bay Press, 2001.
[32] R. H. Byrd, J. Nocedal, and R. A. Waltz, "Knitro: An integrated package for nonlinear optimization," in *Large-Scale Nonlinear Optimization*, G. D. Pillo and M. Roma, Eds. Berlin/Heidelberg, Germany: Springer, 2006, pp. 35–39.
[33] K. Deb, L. Thiele, M. Laumanns, and E. Zitzler, "Scalable multi-objective optimization test problems," in *Proc. CEC*, 2002, pp. 825–830.
[34] C. A. Coello Coello and N. C. Cortés, "Solving multi-objective optimization problems using an artificial immune system," *Genet. Programming Evolvable Mach.*, vol. 6, no. 2, pp. 163–190, 2005.
[35] R. Storn and K. Price, "Differential evolution: A simple and efficient heuristic for global optimization over continuous spaces," *J. Global Optimization*, vol. 11, no. 4, pp. 341–359, Dec. 1997.
[36] *MATLAB*, Version 7.11.0.584 (R2010b), MathWorks, Inc., Natick, MA, 2010.
[37] Q. Zhang, A. Zhau, S. Zhao, P. N. Suganthan, W. Liu, and S. Tiwari, "Multi-objective optimization test instances for the CEC09 special session and competition," Univ. Essex/Nanyang Technol. Univ., Essex, U.K./Singapore, Tech. Rep. CES-487, 2008.

Karthik Sindhya received the B.E. degree in chemical engineering from the R. V. College of Engineering, Bangalore, India, the M.E. degree in chemical engineering from the Birla Institute of Technology and Science (BITS), Pilani, India, and the Ph.D. degree in mathematical information technology from the University of Jyväskylä, Finland, in 2011.

He is currently a Post-Doctoral Researcher with the Department of Mathematical Information Technology, University of Jyväskylä. He was previously a Researcher with the Kanpur Genetic Algorithms Laboratory, Indian Institute of Technology Kanpur, India, and with the Helsinki School of Economics, Helsinki, Finland. His current research interests include single and multi-objective evolutionary algorithms, learning in multi-objective optimization, multiple criteria decision making, and applications to engineering design problems.

Kaisa Miettinen received the M.Sc. degree in mathematics and the Ph.D. degree in mathematical information technology from the University of Jyväskylä, Finland.

She is currently a Professor of industrial optimization with the Department of Mathematical Information Technology, University of Jyväskylä, where she heads the Research Group on Industrial Optimization. She is a part-time Visiting Professor with the KTH Royal Institute of Technology, Stockholm, Sweden. She is the Vice-Rector with the University of Jyväskylä. She wrote the monograph *Nonlinear Multiobjective Optimization* (Boston, MA: Kluwer/Springer, 1999) and about 130 peer-reviewed articles. Her current research interests include multi-objective optimization (theory, methods, and software), multiple criteria decision making, nonlinear programming, evolutionary algorithms, hybrid approaches, and various applications of optimization.

Dr. Miettinen is the President of the International Society on Multiple Criteria Decision Making and the Chair of the Continuous Optimization Working Group of EURO (EUROPT).

Kalyanmoy Deb (F'12) received the Bachelors degree from the Indian Institute of Technology Kharagpur, India, in 1985, and the Masters and Doctoral degrees in engineering mechanics from the University of Alabama, Tuscaloosa, in 1989 and 1991, respectively.

He is currently the Gurmukh and Veena Mehta Endowed Chair Professor of mechanical engineering with the Department of Mechanical Engineering, Indian Institute of Technology Kanpur, India. He is an Adjunct Professor with the Department of Business Technology, Aalto University, Aalto, Finland, and is a Visiting Professor with the University of Skovde, Skovde, Sweden. He has written two textbooks on optimization and more than 250 international journal and conference papers. His current research interests include computational optimization, multi-objective optimization, and evolutionary algorithms. More information about his research can be found at http://www.iitk.ac.in/kangal/deb.htm.

Dr. Deb is the recipient of the Prestigious Shanti Swarup Bhatnagar Prize in Engineering Sciences in India in 2005 and the Infosys Prize in Engineering and Computer Science from the Infosys Science Foundation, Bangalore, India, for his contributions to the emerging field of evolutionary multi-objective optimization in 2012. He received the Thomson Citation Laureate Award from Thomson Scientific for having the highest number of citations in computer science from 1996 to 2005 in India. He is the recipient of the Edgeworth Pareto Award from the International Society on Multiple Criteria Decision Making. He is a fellow of the Indian National Academy of Engineering, the Indian National Academy of Sciences, and the International Society of Genetic and Evolutionary Computation. He received the Fredrick Wilhelm Bessel Research Award from the Alexander von Humboldt Foundation in 2003. He is an Associate Editor and is on the editorial boards of more than 15 major international journals.