Improving convergence of evolutionary multi-objective optimization with local search: a concurrent-hybrid algorithm

Karthik Sindhya · Kalyanmoy Deb · Kaisa Miettinen

Published online: 18 February 2011
© Springer Science+Business Media B.V. 2011

Abstract A local search method is often introduced in an evolutionary optimization algorithm, to enhance its speed and accuracy of convergence to optimal solutions. In multi-objective optimization problems, the implementation of local search is a non-trivial task, as determining a goal for local search in presence of multiple conflicting objectives becomes a difficult task. In this paper, we borrow a multiple criteria decision making concept of employing a reference point based approach of minimizing an achievement scalarizing function and integrate it as a search operator with a concurrent approach in an evolutionary multi-objective algorithm. Simulation results of the new concurrent-hybrid algorithm on several two to four-objective problems compared to a serial approach, clearly show the importance of local search in aiding a computationally faster and accurate convergence to the Pareto optimal front.

Keywords Multicriteria optimization · Multiple criteria decision making · Pareto optimality · Evolutionary algorithms · Hybrid algorithms · Achievement scalarizing functions · NSGA-II

K. Sindhya (✉) · K. Miettinen
Department of Mathematical Information Technology, University of Jyväskylä,
P.O. Box 35, 40014 Agora, Finland
e-mail: karthik.sindhya@jyu.fi

K. Miettinen
e-mail: kaisa.miettinen@jyu.fi

K. Deb
Department of Business Technology, Aalto University School of Economics,
P.O. Box 21220, 00076 Aalto, Finland
e-mail: deb@iitk.ac.in

Present Address:

K. Deb
Department of Mechanical Engineering, Indian Institute of Technology Kanpur, Kanpur 208016, India

1 Introduction

Evolutionary algorithms have been widely used to solve multi-objective optimization problems since more than a decade. Evolutionary multi-objective optimization (EMO) deals with a population of points and yields a set of solutions which are non-dominated and near to the Pareto optimal front. Some of the main advantages of using EMO algorithms include: obtaining a set of non-dominated solutions in a single run, overcoming multiple local Pareto optimal fronts easily and flexibility in handling: discrete, nonlinear, multi-modal and large-scale problems. These algorithms have been commonly used in many practical problems including engineering design (Deb 2001; Coello and Lamont 2004), financial applications (Tapia and Coello 2007) and scheduling applications (Ishibuchi and Murata 1998; Jaskiewicz 2002; Shaw et al. 1999).

Commonly used EMO algorithms such as non-dominated sorting genetic algorithm (NSGA-II) (Deb et al. 2002a), strength Pareto evolutionary algorithm (SPEA) (Zitzler and Thiele 1998a), etc. particularly focus on the distribution quality. Despite their success and wide-spread applications, these algorithms are often criticized for their lack of theoretical convergence proof to the Pareto optimal front. A past study (Laumanns et al. 2002) has demonstrated and argued that, EMO algorithms such as NSGA-II, SPEA, etc. with a finite size archive for storing non-dominated solutions, can allow an evolving population to fluctuate near the Pareto optimal front (convergence to the Pareto optimal front followed by a departure of some solutions out of the front). This phenomenon can result from a constant emphasis on the diversity maintenance pursued in EMO on a finite size population. In other words, to make the diversity among non-dominated solutions better, a Pareto optimal solution may be replaced by a non-Pareto optimal solution, as the algorithm has no way to determine the Pareto optimality of solutions without the use of any optimality conditions.

Multiple criteria decision making (MCDM) techniques (Sawaragi et al. 1985; Chankong and Haimes 1983; Miettinen 1999) are also commonly used to deal with multi-objective optimization problems. These techniques constitute a collection of approaches (Miettinen 1999) and have theoretical convergence proofs. In MCDM a multi-objective problem is often scalarized into a single-objective problem, accounting for the preference information of a human decision maker (DM) (Miettinen 1999). Subsequently, the scalarized problem is solved by using an appropriate mathematical programming technique to find a single Pareto optimal solution. The MCDM field has many methods to solve multi-objective optimization problems. Among them, the reference point method is commonly used (Miettinen 1999). Reference points which are formed by the aspiration values of the DM, can be utilized to derive an achievement scalarization function (Wierzbicki 1980). An achievement scalarization function has optimal solutions on the Pareto optimal front only (Miettinen 1999; Wierzbicki 1980). In simple words, with this function a reference point is projected on to the Pareto optimal front.

The criticism of EMO algorithms can be bridged somewhat by incorporating MCDM approaches into EMO algorithms as a local search. The EMO algorithm as a global optimizer deploys a global search to find promising regions of importance, while a MCDM approach as a local optimizer starts from a particular solution in an EMO algorithm population and converges quickly to a locally optimal solution. The roles of global and local optimizers can be used to negotiate different multi-modal function landscapes and reach the Pareto optimal front fast and accurately. Such a hybridization shall give rise to a global search algorithm with guaranteed convergence provided the chosen local search approach guarantees converging to a local optimum. Convergence in this paper is an indicator of the proximity of non-dominated solutions generated by using an EMO algorithm to the Pareto optimal front.

Incorporating local search in an EMO approach is not easy, as the local search usually involves a single objective to be minimized. Multi-objective optimization problems involve multiple conflicting objectives and choosing one among them is unfair, as all are equally important at least during the optimization process. The principal goal of our research is to have an algorithm with two prominent features, i.e., convergence and speed to reach the Pareto optimal front and use an appropriate scalarizing function in local search. The Karush–Kuhn–Tucker (KKT) (Kuhn and Tucker 1951) termination conditions of mathematical programming techniques to be utilized in local search can be used to guarantee convergence for our concurrent-hybrid EMO algorithm. On these lines, we have suggested an approach (Sindhya et al. 2008) to combine an achievement scalarizing function (to be optimized in the local search phase) with the NSGA-II. In our concurrent-hybrid algorithm, any appropriate mathematical programming technique can be used in the local search phase and we fix the mathematical programming technique only during the numerical tests. In this paper, we extend our previous approach into a structured algorithm, perform extensive numerical tests including a parametric study of our concurrent-hybrid EMO algorithm and explain the interactions between an EMO algorithm and a local search to make the overall algorithm efficient. Henceforth, we refer our concurrent-hybrid EMO algorithm as the concurrent-hybrid NSGA-II algorithm.

The rest of this paper is organized as follows. In Sect. 2, we introduce some concepts and specifically describe achievement scalarizing functions. Then in Sect. 3, we summarize a broad survey of available literature on hybridization ideas suggested for EMO algorithms. Our concurrent-hybrid NSGA-II algorithm is presented in Sect. 4 with an overview on frequency of local search and termination criteria. Then Sect. 5 describes the test and parameter settings, parametric study and results obtained from the comparison studies between the concurrent-hybrid NSGA-II and serial-hybrid NSGA-II algorithms on a number of test and practical problems. This section also briefly discusses possible approaches for ensuring diversity in a concurrent-hybrid algorithm. In Sect. 6, we discuss the dynamics of our concurrent-hybrid NSGA-II algorithm. Finally, conclusions are drawn in Sect. 7.

2 Concepts and achievement scalarizing functions

We consider a multi-objective optimization problem of the form

$$\begin{aligned} &\text{minimize} \quad \{f_1(\mathbf{x}), f_2(\mathbf{x}), \dots, f_k(\mathbf{x})\} \\ &\text{subject to} \quad \mathbf{x} \in \mathcal{S}, \end{aligned} \quad (1)$$

with $k \geq 2$ conflicting objective functions $f_i : \mathbb{R}^n \rightarrow \mathbb{R}$. We denote the vector of objective function values by $\mathbf{f}(\mathbf{x}) = (f_1(\mathbf{x}), f_2(\mathbf{x}), \dots, f_k(\mathbf{x}))^T$ to be called an objective vector. The decision vectors $\mathbf{x} = (x_1, x_2, \dots, x_n)^T$ belong to the feasible region $\mathcal{S}$, which is a subset of the decision variable space $\mathbb{R}^n$.

For problem (1), a decision vector $\mathbf{x}^* \in \mathcal{S}$ is Pareto optimal if there does not exist another $\mathbf{x} \in \mathcal{S}$ such that $f_i(\mathbf{x}) \leq f_i(\mathbf{x}^*)$ for all $i = 1, 2, \dots, k$ and $f_j(\mathbf{x}) < f_j(\mathbf{x}^*)$ for at least one index j . To be more specific, a decision vector $\mathbf{x}^* \in \mathcal{S}$ is Pareto optimal if $(\mathbf{f}(\mathbf{x}^*) - \mathbb{R}_+^k \setminus \{\mathbf{0}\}) \cap Z = \emptyset$, where $Z = \mathbf{f}(\mathcal{S})$ is the feasible objective region and $\mathbb{R}_+^k = \{\mathbf{x} \in \mathbb{R}^n | x_i \geq 0 \text{ for } i = 1, \dots, n\}$. An objective vector is Pareto optimal if the corresponding decision vector is Pareto optimal. The above definition also dictates global Pareto optimality. A decision vector $\mathbf{x}^* \in \mathcal{S}$ and the corresponding objective vector $\mathbf{z}^* = \mathbf{f}(\mathbf{x}^*) \in Z$ are properly Pareto optimal if $(\mathbf{z}^* - \mathbb{R}_\rho^k \setminus \{\mathbf{0}\}) \cap Z = \emptyset$, where, $\mathbb{R}_\rho^k = \{\mathbf{z} \in \mathbb{R}^k | \max_{i=1, \dots, k} z_i$

$+\rho \sum_{i=1}^k z_i \geq 0\}$ and $\rho > 0$ is a predetermined scalar. Parameter ρ produces properly Pareto optimal solutions with trade-offs bounded by ρ and $1/\rho$ (Wierzbicki 1980; Miettinen 1999). Pareto optimal and properly Pareto optimal solutions are illustrated in Fig. 1. The dotted section corresponds to properly Pareto optimal solutions and the bold and dotted section together are Pareto optimal solutions. The point $\mathbf{z}^*$ is an example of properly Pareto optimal solutions. The set of properly Pareto optimal solutions is a subset of Pareto optimal solutions, which can also be seen in Fig 1.

Let $B(\mathbf{x}^*, \delta)$ denote an open ball with a center $\mathbf{x}^*$ and a radius $\delta > 0$, $B(\mathbf{x}^*, \delta) = \{\mathbf{x} \in \mathbb{R}^n \mid \|\mathbf{x}^* - \mathbf{x}\| < \delta\}$. A decision vector $\mathbf{x}^* \in \mathcal{S}$ is said to be locally Pareto optimal if there exists $\delta > 0$ such that $\mathbf{x}^*$ is Pareto optimal in $\mathcal{S} \cap B(\mathbf{x}^*, \delta)$. An objective vector is locally Pareto optimal if the decision vector corresponding to it is locally Pareto optimal. A vector is *weakly Pareto optimal* if there does not exist any other feasible vector for which all objective values are better.

In multi-objective optimization, several conflicting objective functions are optimized simultaneously. Hence, it is impossible to find a single optimal solution where all the objectives have reached their individual optima. Typically problem (1) has many Pareto optimal solutions and hence, we need a DM who is an expert in the domain of the problem under consideration, to identify the most preferred solution among them.

In the MCDM field, multi-objective methods can be classified in the following four classes (Miettinen 1999):

1. No-preference methods: No preference information is used and a neutral compromise solution is found.
2. A posteriori methods: A representative set of Pareto optimal solutions is presented to the DM and s(he) selects one among them based on her/his preferences.
3. A priori methods: The DM is first asked to present her/his preferences and the best possible solution arising from her/his preferences is found.
4. Interactive methods: The DM articulates preference information iteratively and thus directs the solution process progressively.

Interactive methods are commonly used (Miettinen et al. 2008) and many interactive procedures have been developed, among them the reference point method (Wierzbicki 1980) is commonly used. A reference point is a vector formed by the desirable values for each objective function specified by the DM. Using the reference point, a scalarizing function like an achievement scalarizing function (Wierzbicki 1980) is formed and optimized to find a solution that best satisfies the aspirations of the DM. An achievement scalarizing function has many advantages (Miettinen 1999; Wierzbicki 1980) such as

Fig. 1 Pareto optimality

1. The optimal solution of an achievement scalarizing function is always (weakly) Pareto optimal.
2. Any 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.

Assuming we have a reference point $\bar{\mathbf{z}} \in \mathbb{R}^k$, an example of an achievement scalarizing function is given by

$$\begin{aligned} & \text{minimize} && \max_{i=1}^k [w_i(f_i(\mathbf{x}) - \bar{z}_i)] \\ & \text{subject to} && \mathbf{x} \in \mathcal{S}, \end{aligned} \quad (2)$$

where $w_i = \frac{1}{z_i^{\max} - z_i^{\min}}$ is a weight factor assigned to each objective function f_i . The maximum and minimum values of each objective in the non-dominated set are represented as $z_i^{\max}$ and $z_i^{\min}$, respectively. In the above setting, the weight factors are used to normalize the values of each objective function f_i . In the context of EMO algorithms and successive studies, we use currently available worst and best function values in the non-dominated set as $z_i^{\max}$ and $z_i^{\min}$, respectively. For other ways to utilize the weight factors, see, e.g. Luque et al. (2009).

Assuming all the functions involved are differentiable, we obtain an equivalent differentiable formulation by introducing an extra real-valued variable (α) and k new constraints (Miettinen 1999)

$$\begin{aligned} & \text{minimize} && \alpha \\ & \text{subject to} && [w_i(f_i(\mathbf{x}) - \bar{z}_i)] \leq \alpha \text{ for all } i = 1, \dots, k, \\ & && \mathbf{x} \in \mathcal{S}, \quad \alpha \in \mathbb{R}. \end{aligned} \quad (3)$$

The extra real-valued variable (α) is henceforth referred as a slack variable.

The formulations (2) and (3) may produce a weakly Pareto optimal solution. This can be avoided by adding an augmentation term (Wierzbicki 1986; Miettinen 1999). An augmented achievement scalarizing function can be written as

$$\begin{aligned} & \text{minimize} && \max_{i=1}^k \frac{f_i(\mathbf{x}) - \bar{z}_i}{z_i^{\max} - z_i^{\min}} + \rho \sum_{i=1}^k \frac{f_i(\mathbf{x}) - \bar{z}_i}{z_i^{\max} - z_i^{\min}} \\ & \text{subject to} && \mathbf{x} \in \mathcal{S}, \end{aligned} \quad (4)$$

where ρ is a sufficiently small positive scalar called an augmentation coefficient, e.g. $\rho = 10^{-3}$, is same as used in the definition of properly Pareto optimal solution. Augmentation terms may also improve the computational efficiency (shorter computation time) for the above problem (Miettinen et al. 2006).

Now, having introduced achievement scalarization functions, we shall present a brief overview of the literature in the field of hybrid multi-objective optimization involving EMO algorithms as a background for the new hybrid algorithm to be proposed.

3 Previous studies on hybrid EMO approaches

The use of local search in EMO has received some attention in the recent past. One of the important goals has been to reach the Pareto optimal front using less function evaluations than with EMO algorithms alone. There are at least two different ways EMO algorithms can be hybridized: a serial and a concurrent approach (Goel and Deb 2002), and we classify the literature based on this distinction.

1. In a *serial* approach, global and local searches are applied serially one after the other with appropriate termination conditions, i.e. a global search algorithm is run for a pre-fixed number of generations and the resulting solutions are later improved using a local search. This procedure guarantees convergence to the local Pareto optimal front (provided the associated local search is guaranteed to find a local optimum), but the switch over to a local search from a global optimizer is not easy to fix a priori on any practical problem. A delay in terminating a global solver can consume excess function evaluations and an early termination may yield a locally Pareto optimal solution. For terminating a local search, standard procedures such as determining whether the error in violations of the KKT conditions of optimality is within a limit or if there is no improvement in a locally optimal solution over past few iterations may be implemented. Based on the multisexual genetic algorithm (MSGA), Levia et al. (2000) proposed three hybrid algorithms: MSGA-LS1, MSGA-LS2 and MSGA-LS3. The Pareto dominance-based scheme (Coello et al. 2007) is used in all the three algorithms without objective function scalarization. MSGA-LS1 and MSGA-LS3 deploy a serial approach and use simulated annealing (Kirkpatrick et al. 1983) and neighborhood search respectively, as local search algorithms, although simulated annealing is regarded as a global algorithm. A local search is applied to all points in the non-dominated set obtained by the MSGA. The MSGA-LS3 algorithm was reported to perform best when tested on simple test functions (linear and quadratic functions with no local Pareto optimal fronts). Goel and Deb (2002) proposed hybrid algorithms based on an a posteriori (serial) and an online (concurrent) approach, using a weighted sum of objectives as an objective function for a local search. The a posteriori approach deploys the neighborhood search as a local search algorithm on all the non-dominated points obtained by the NSGA-II and was reported as performing best with respect to diversity and convergence to the Pareto optimal front, when tested on mechanical shape optimization problems. Talbi et al. (2001) used local search as a means for acceleration and refinement of genetic search. Once the maximum number of generations in a genetic algorithm (GA) is used, a local search is applied to all points in the previously obtained non-dominated set. A weighted sum of objectives was used to transform the multi-objective problem to a single objective problem and the neighborhood search was used as a local search. A new hybrid algorithm which uses the Pareto descent method (PDM) as a local search method was proposed by Harada et al. (2006). The PDM finds feasible Pareto descent directions which improve all objective functions simultaneously by solving computationally inexpensive linear programming problems. The authors implemented both a serial and a concurrent approach and suggested the serial approach to perform better than the concurrent one. In Emmerich et al. (2007), a new hybrid algorithm using the S-metric (Zitzler and Thiele 1998b) was proposed. Here, an evolutionary multi-objective optimizer was initially used to generate a set of non-dominated solutions close to the Pareto optimal front and subsequently a gradient based method was used to compute a local approximation set with respect to the S-metric.
2. In a *concurrent* approach, a local search is embedded within a global optimizer so that some or all intermediate solutions are modified by the local search. In an EMO framework, a local search may be considered as an additional EMO operator which attempts to bring an intermediate solution to a locally Pareto optimal solution. The problem of the switch over between a global and a local optimizer is avoided in this approach. The termination criterion of a local search can be a standard one (as described previously in a serial approach), but the termination of a global optimizer need not be ad-hoc, as in the serial approach: it can be based on whether there is an improvement in

the locally Pareto optimal solutions over past few iterations. Ishibuchi and Murata (1998) proposed the first hybrid algorithm of a multi-objective genetic algorithm (MOGA). A linear utility function with random weights was used as a fitness function and each selection was based on a different weight vector. A local search in the form of a restricted neighborhood search was applied to all population members after a crossover and mutation operation in every generation. The proposed algorithm was applied on a multi-objective flowshop scheduling problem and good performance was reported. The multi-objective genetic local search (MOGLS) method was proposed by Jaskiewicz (2002). It applied the neighborhood search as a local search in every generation. Jaskiewicz used either a weighted linear utility function or a weighted Chebyshev utility function, with weights drawn at random as an objective function for local search. However, weighted linear functions were reported to perform better than a weighted Chebyshev utility function on bi-objective and three objective instances of the traveling salesman problem (TSP). The proposed MOGLS algorithm was compared with the Ishibuchi and Murata's hybrid algorithm and multi-objective simulated annealing (MOSA). The MOGLS method significantly outperformed other hybrid algorithms. A memetic algorithm for multi-objective optimization, the memetic Pareto archived evolution strategy (M-PAES) was proposed by Knowles and Corne (2000b). The algorithm combines the $(1 + 1)$ -PAES (Knowles and Corne 2000a) with a crossover operation and the $(1 + 1)$ -evolutionary strategy (ES) (Beyer and Schwefel 2002) is used as a local search strategy. A local search was applied continually on all individuals. Verification of the new algorithm was carried out on a set of multi-objective 0/1 knapsack problems and compared with the $(1 + 1)$ -PAES and the SPEA. The M-PAES outperformed the $(1 + 1)$ -PAES but was similar in performance to SPEA. The MSGA-LS2 proposed by Levia et al. (2000) used a concurrent approach applying simulated annealing on every individual in every generation. Goel and Deb (2002) also proposed an online or a concurrent approach, in which every solution created by the genetic operators in MOEA is modified by the local search procedure before being accepted. However, both Levia et al. and Goel and Deb report a serial approach to perform better than a concurrent approach. Hansen (2000) considered the use of a Chebyshev program, which means optimizing a particular scalarizing function for a local search in the tabu search (Glover 1989) and which has been tested to be superior in handling a multi-objective TSP. Ishibuchi et al. (2006) proposed an idea of probabilistically using a scalarizing fitness function (weighted sum) and the NSGA-II fitness evaluation mechanism during parent selection in EMO. Computational experiments on 0/1 knapsack problems with two, three and four objectives showed improvement in the performance of EMO algorithms. Sindhya et al. (2008) proposed a concurrent hybrid approach using NSGA-II as an EMO algorithm. An achievement scalarizing function (Miettinen 1999) was used as a scalarizing function and optimized using e.g., the sequential quadratic programming (SQP) method. Preliminary results reported were promising on some problems from the ZDT (Deb 2001) and DTLZ (Deb et al. 2002b) test suites.

In addition to the above studies on different algorithms, there exist some studies on hybridization and different implementation schemes of local search with EMO algorithms. Ishibuchi and Narukawa (2004) compared two implementation schemes of a local search in hybrid EMO algorithms, a weighted sum of multiple objectives and the Pareto dominance. The two implementations were tested on a multi-objective 0/1 knapsack and multi-objective flowshop scheduling problems. A weighted sum based scheme showed better results especially in terms of the diversity of obtained non-dominated solutions. Ishibuchi

et al. (2003) examined the positive and negative effects of hybridization with a local search on the MOGLS, using a weighted sum of multiple objectives. The positive effect consisted of an improvement in convergence speed to the Pareto optimal front and as a negative effect there was an increase in computational time per generation. Such positive and negative effects were examined by computational experiments on multi-objective permutation flowshop scheduling problems. The MOGLS was modified by choosing only good individuals for local search and assigning local search directions to each solution. This modified MOGLS was shown to produce better results than SPEA and NSGA-II. The performance of SPEA and that of NSGA-II were also significantly improved for test problems when hybridized with a local search of MOGLS. Ishibuchi et al. (2003) also tried to strike a balance between genetic and local searches using fixed parameters and motivating researchers to design adaptive multi-objective memetic algorithms that can control the balance between genetic and local searches.

From the above survey one can see that optimizing a weighted sum of objectives for the Pareto optimal set generation seems to have been the most common scalarizing procedure in multi-objective optimization solved with a naive neighborhood search based procedure. However, it is known that with a weighted sum of objectives, getting all the points on the Pareto optimal front is impossible unless the Pareto optimal front is convex (Osyczka 1984; Miettinen (1999). But, this serious shortcoming has not received any attention in algorithms belonging to the serial and concurrent approaches discussed above. Thus, there is a need to rethink the type of scalarizing function which could be used in a local search.

No superiority over the concurrent and serial hybrid EMO approaches could be seen, as references employ different algorithms and different test functions for testing. However, we prefer concurrent hybrid EMO algorithms to avoid the switch over problem between global and local searches. Realizing this inherent shortcoming in the serial-hybrid approaches and noting that often less effective scalarization functions have been used in the literature, we proposed a concurrent approach of hybridizing global and local search in Sindhya et al. (2008), borrowing a more effective idea of an achievement scalarizing function, based on reference point methods from the MCDM literature for a local search. As mentioned in Sect. 2, an achievement scalarizing function can yield any Pareto optimal solution by just altering a reference point, which scores this scalarizing function higher than a weighted sum of objectives. Here, EMO solutions can be considered as reference points. In every generation of the hybrid approach, reference points were chosen with a probability and an achievement scalarizing function was optimized to get a locally Pareto optimal solution. The preliminary results were shown to be promising for some problems derived from ZDT and DTLZ test suites. However, it must be noted that the paper represented just the initial studies proposing an approach to integrate an achievement scalarizing function based local search into the EMO algorithm and needed a formulation of an algorithm as well as further testing on problems and parametric studies. In this paper we evolve the approach into an algorithm with further tests on two practical problems (discussed later in this paper) besides test problems, present an exhaustive parametric study and present a termination criterion for the proposed hybrid algorithm.

4 Concurrent-hybrid algorithm

The desire to improve the speed and accuracy of existing EMO algorithms has enticed us to suggest a hybrid algorithm. We use NSGA-II as our EMO algorithm, mainly because it is modular and functionally decomposable (Deb 2008). This enables us to introduce a local

search easily as an additional operator. A local search algorithm is not pre-fixed for solving the achievement scalarizing function, and the user may use either a gradient based or a non-gradient based algorithm, whichever is found suitable to the problem in question. Our concurrent-hybrid NSGA-II is an extended version of the original NSGA-II, with the following two changes:

1. Local search is introduced as an additional operator sparingly, using a predefined probability function.
2. A convergence based termination criterion is introduced for the NSGA-II algorithm.

4.1 Concurrent-hybrid NSGA-II algorithm

Next, we briefly describe our concurrent-hybrid NSGA-II algorithm.

1. Generate a random initial population P_0 of size N and set generation count $t = 0$.
2. Sort population to different non-domination levels (fronts) and assign each solution a fitness equal to its non-domination level (1 is the best level). Minimization of the fitness is assumed.
3. Create offspring population Q_t of size N using binary tournament selection, recombination and mutation operations.
4. Perform local search on some individuals picked with a probability of P^{local}_t from the offspring population Q_t . Replace the offspring population with the improved solution from the local search.
5. Combine the parent and the offspring populations and create $R_t = P_t \cup Q'_t$.
6. Perform non-dominated sorting to R_t and identify different fronts $F_i, i = 1, 2, \dots$, etc.
7. Set new population $P_{t+1} = \emptyset$. Set a count $i = 1$ and as long as $|P_{t+1}| + |F_i| \leq N$, perform $P_{t+1} = P_{t+1} \cup F_i$ and $i = i + 1$.
8. Perform the crowding-sort procedure (Deb 2001) and include the most widely spread ($N - |P_{t+1}|$) members of F_i by using the crowding distance values in the sorted F_i to P_{t+1} .
9. Check if the termination criterion is satisfied. If yes, go to step 10, else set $t = t + 1$ and return to Step 2.
10. Perform local search on every individual in the population P_{t+1} .

In the concurrent-hybrid NSGA-II algorithm all the offspring produced in Step 4 are subjected to local search with a probability in every generation. The locally optimized solutions then replace the corresponding original offspring created by the NSGA-II. The optimal slack variable value generated during the local search (or the optimal solution of the achievement scalarizing function) is then utilized to terminate (which we describe later) our algorithm. Once the concurrent-hybrid NSGA-II algorithm is terminated in Step 9, we then apply a local search on all non-dominated points to guarantee convergence of all non-dominated solutions to the Pareto optimal front.

Since the above algorithm is based on the NSGA-II algorithm, the framework will change for other EMO algorithms, but the two changes can be incorporated to any other hybridized EMO algorithm to boost their performance with respect to a lower function evaluation consumption.

In the next subsection, we describe the two changes, namely sparing use of local search and termination criteria in detail.

4.2 Frequency of local search

Before we discuss the frequency with which a local search operator is to be used, we need to understand the purpose of employing it with an EMO algorithm. The NSGA-II procedure emphasizes non-dominated and less-crowded solutions in a population, thereby maintaining a *push* towards the Pareto optimal front and simultaneously keeping an adequate spread among non-dominated solutions. Our proposed local search operator (achievement scalarizing function optimization task) is capable of taking a solution all the way to a nearest *locally* Pareto optimal front, due to the properties of achievement scalarizing functions. However, the proposed achievement scalarizing function optimization algorithm works along a particular direction in the objective space dictated by the choice of weight factors discussed in Sect. 2. Thus, if the local search is applied to many population members early on, a number of difficulties which we discuss next may crop up:

1. The resulting solutions may all lie on a specific region on the locally Pareto optimal front, thereby reducing the much-needed diversity in the EMO population.
2. The resulting solutions will most likely dominate the entire population, thereby reducing the importance of dominated solutions. An overemphasis of locally optimized solutions will quickly lead to a loss of diversity in the EMO population.
3. A recent study on *innovization* (Deb and Srinivasan 2008; Deb and Sindhya 2008) revealed the fact that locally and globally Pareto optimal solutions possessed some common characteristics (a type of a ‘signature’ of the solutions on that front). Many locally optimal solutions on a particular sub-optimal front may possess similar sub-optimal *traits*, which may later be difficult to change by genetic operators applied on the population.

The above issues indicate that a local search, if employed, must be performed sparingly and judiciously, at the beginning of the optimization process. However, when the population is close to the Pareto optimal front, the frequency of local search can be increased in order to ensure Pareto optimality of more and more solutions in a computationally fast manner. Thus, we can think of a local search strategy which is performed modestly in the beginning and then with more intensity during further generations.

However, there is another issue which must be considered when solving complex optimization problems. A convergence of an algorithm to a front with an adequate spread does not necessarily mean a convergence to the desired globally Pareto optimal front. To achieve global optimality, the operators of an optimization algorithm must constantly look for better solutions despite an apparent convergence to a front. But, local search operators are designed to facilitate faster and accurate convergence to optimal solutions. They cannot differentiate convergence to a locally or a globally optimal solution. Moreover, because of the local nature of search operator, a further application of the local search operator to a locally optimal solution will not improve the solution in any way. Thus, when a convergence to an optimal solution is achieved, it is not wise to continue with the local search operator any more, until there is an improvement of the solution by other search operators. Therefore, to provide a global perspective to the overall search procedure, ideally we should employ the local search procedure in a periodic fashion.

Initially, the frequency of local search phases must be limited, and then increased with further generations until a particular convergence level is achieved. Thereafter, the frequency of the local search must be reduced drastically to allow global search operators to act and find hopefully better solutions. The frequency of the local search operator can then be increased again till another level of convergence is achieved. This cycle can be continued until no further improvement in the solutions is achieved.

We implement this cyclic nature of a local search operator here by invoking the operator with a saw-tooth probability function. The probability P_t^{local} starts from zero at generation one and is then linearly increased to P_t^{local} at generation t . At generation $t + 1$, P_{t+1}^{local} drops to zero and then again increases linearly, which we refer as the “fixed strategy”. In Step 4 of the algorithm, we locally optimize the offspring solutions with a probability P_t^{local} using a mathematical programming technique. Thus, the stopping criterion of the mathematical programming technique used in local search may also be used in the operator with a saw-tooth probability function. When a gradient based solver is used as a mathematical programming technique, KKT optimality error value can be used as a stopping criterion for local search. If the average KKT optimality error value for the past H local searches is smaller than the threshold η indicating that the population is close to a locally Pareto optimal front, then P_t^{local} can be reduced to zero and then a linear increase in further generations can be restarted. However, in this paper we use the “fixed strategy” described earlier.

4.3 Termination criteria

The concurrent-hybrid NSGA-II algorithm involves three termination criteria, i.e., local search termination criteria in Steps 4 and 10 and algorithm termination criterion in Step 9.

1. *Local search termination criteria:* The concurrent-hybrid NSGA-II algorithm invokes a local search in two different Steps (4 and 10). The termination criterion in Step 4 is prefixed to (a low) t_1 number of iterations, as our motivation in Step 4 is only to supply improved offspring to speed up the concurrent-hybrid NSGA-II algorithm. In Step 10 of the concurrent-hybrid NSGA-II algorithm we allow the local search to terminate by its inbuilt termination criteria of KKT optimality error (ε) and maximum number of function evaluations (t_2) (which is usually fixed to be a very large number), to guarantee optimality.
2. *Algorithm termination criterion:* EMO algorithms usually are terminated after a pre-specified number of generations or when no new solutions have entered the set after a prefixed number of generations in Step 9. These stopping criteria involve large computation costs and do not indicate the convergence to the Pareto optimal front. Hence, there is a need of a stopping criterion for the hybrid algorithm, which should be automatic, ensuring an adequate convergence to a Pareto optimal front with an appropriate diversity. To terminate our hybrid algorithm, we may either utilize the value of a running average of the slack variable α or the optimal value of an achievement scalarizing function being close to zero over a prefixed number of generations. Also, if more than one local search is done in a generation, we take the average of the slack variable of all local searches for the corresponding generation.

5 Numerical experiments and discussions

In this section, we present the test and parameter settings, parametric study and the comparative results produced on comparing the efficacy of the proposed concurrent-hybrid NSGA-II algorithm with a simple serial-hybrid NSGA-II algorithm.

5.1 Test setting

We compare our concurrent-hybrid NSGA-II algorithm with the serial-hybrid NSGA-II algorithm in this study, so that we have two sets of non-dominated solutions from two variations

of the hybrid algorithm for comparison, all solutions being globally or at least locally Pareto optimal. In our tests, we have used the KNITRO solver (Byrd et al. 2006) using the SQP technique as a local solver. We chose test problems, ranging from two to four objectives from ZDT and DTLZ test suites (Deb 2001; Deb et al. 2002b) and two practical problems: a welded beam design problem (Deb 2001) and the water resources planning problem (Galperin and Wiecek 1999). The two practical problems are described subsequently:

1. Water resources planning (WRP): The problem considered describes a simple water resource planning problem. A multipurpose dam is to be constructed, such that the cost of construction f_1 and the water loss f_2 (volume/year) are minimized, and the total storage capacity f_3 of the reservoir is maximized. The decision variables are chosen to be total man-hours devoted to building the dam, x_1 , and the mean radius of the lake impounded (in miles), x_2 . The resulting formulation is:

$$\begin{aligned} \text{minimize} \quad & f_1(\mathbf{x}) = e^{0.01x_1} x_1 0.02x_2^2 \\ & f_2(\mathbf{x}) = 0.5e^{0.01x_1} \\ & f_3(\mathbf{x}) = -e^{0.005x_1} x_1^{0.001} x_2^2 \\ \text{subject to} \quad & 0.01 \leq x_1 \leq 1.3 \\ & 0.01 \leq x_2 \leq 10. \end{aligned} \quad (5)$$

2. Welded beam design (WELD): A beam needs to be welded on another beam and must carry a certain load F . It is desired to find four design parameters (thickness of the beam b , width of the beam t , length of weld l , and weld thickness h) for which the cost of the beam is minimum and simultaneously the vertical deflection at the end of the beam is minimum. We have the following problem:

$$\begin{aligned} \text{minimize} \quad & f_1(\mathbf{x}) = 1.1047h^2l + 0.04811tb(14.0 + l) \\ & f_2(\mathbf{x}) = \frac{2.1952}{t^3b} \\ \text{subject to} \quad & g_1(\mathbf{x}) = 13600 - \tau(\mathbf{x}) \geq 0.0 \\ & g_2(\mathbf{x}) = 30000 - \sigma(\mathbf{x}) \geq 0.0 \\ & g_3(\mathbf{x}) = b - h \geq 0.0 \\ & g_4(\mathbf{x}) = P_c(\mathbf{x}) - 6000 \geq 0.0 \\ & 0.125 \leq h, b \leq 5.0 \\ & 0.1 \leq l, t \leq 10.0 \end{aligned} \quad (6)$$

$$\text{where} \quad \tau(\mathbf{x}) = \sqrt{(\tau_1)^2 + (\tau_2)^2 + (l\tau_1\tau_2)/\sqrt{0.25(l^2 + (h+t)^2)}}$$

$$\tau_1 = 6,000/(\sqrt{2}hl)$$

$$\tau_2 = \frac{6,000(14 + 0.5l)\sqrt{0.25(l^2 + (h+t)^2)}}{2\{0.707hl(l^2/12 + 0.25(h+t)^2)\}}$$

$$\sigma(\mathbf{x}) = \frac{504,000}{t^2b}$$

$$P_c(\mathbf{x}) = 64,746.022(1 - 0.0282346t)tb^3.$$

The problem has four constraints. The first constraint ensures that the shear stress developed at the support location of the beam is smaller than the allowable shear strength of the

material (13,600 psi). The second constraint ensures that normal stress developed at the support location of the beam is smaller than the allowable yield strength of the material (30,000 psi). The third constraint ensures that the thickness of the beam is not smaller than the weld thickness from a practical standpoint. The fourth constraint makes sure that the allowable buckling load (along t direction) of the beam is more than the applied load F .

Each of these problems were solved 10 times with different seeds and the best, median and worst values of performance metrics [that is, number of function evaluations and hyper-volume (HV) values (Zitzler and Thiele 1998b)] were noted to reach the Pareto optimal front. For testing purposes, we made a minor change and utilized a modified stopping criterion in Step 9 of our concurrent-hybrid NSGA-II algorithm (for problems from the test suites). This was mainly done to fix a goal and find the best algorithm for all test problems. In Step 9 a stopping criterion based on the discrepancy (we call it an error metric here) in objective f_k between the obtained solution and the corresponding f_k value obtained by substituting other objective values (f_1 to f_{k-1}) in the Pareto optimal relationship ($f_k = f_k(f_1, \dots, f_{k-1})$) was calculated for each non-dominated solution. If the sum of the square of errors generated by all non-dominated solutions were less than, or equal to 0.001 (chosen arbitrarily here), the hybrid algorithm was terminated to apply the local search on all non-dominated solutions. For the two practical problems, we do not know the Pareto optimal front a priori, hence, the termination criterion based on the maximum number of function evaluations (fixed to be 25,000) as a fixed goal had to be used in Step 9 of our algorithm. The error metric and maximum number of function evaluations were also used as termination criteria for serial-hybrid NSGA-II, to maintain uniformity for comparison. Thereafter, to check the diversity of the obtained set of Pareto optimal solutions, we calculated their HV values.

In the next subsection, we present the parameter setting utilized for the numerical experiments. This parameter setting can also be used as a guideline for any practical problem using our hybrid algorithm.

5.2 Parameter setting

An EMO algorithm like NSGA-II includes parameters such as crossover and mutation probability, SBX (Deb 2001) and mutation distribution index (Deb 2001). The values of these parameters determine, to a greater extent, whether an algorithm will find a near Pareto optimal front with speed, accuracy and diversity.

Choosing the right parameter value is a time consuming task, and an attempt has been made to suggest appropriate parameter settings for the concurrent-hybrid NSGA-II algorithm. Parameters for the hybrid algorithm can be broadly classified into NSGA-II and the local search specific parameter settings. In what follows, we list the parameters and their values used in the tests.

1. NSGA-II parameters:

- (a) A population size of 100 for bi-objective and 200 for problems with more objectives can be considered for both the concurrent and the serial hybrid NSGA-II algorithms.
- (b) Both the concurrent and the serial hybrid NSGA-II algorithms use the crossover probability of 0.9.
- (c) The mutation probability is fixed, based on a commonly used heuristic (1/number of variables), for all problems except DTLZ2. For DTLZ2, both the concurrent

and the serial hybrid NSGA-II algorithms require a smaller probability of mutation to converge to the desired accuracy. Hence, a low mutation probability of 0.001 is used for better results.

- (d) The SBX distribution index η_c and the mutation distribution index η_m have been fixed to be 15 and 20 respectively, based on a detailed study for both the concurrent and the serial hybrid NSGA-II algorithms.
2. Local search parameters:
 - (a) The frequency of applying the local search on individuals in Step 4 of the concurrent-hybrid NSGA-II algorithm, can be controlled using P_{local} , a scheme described in the previous section. The probability P_{local} reaches 0.01 at every $25(k - 1)$ generation, where k is the number of objectives. This scheme is chosen to ensure that at least one cycle of P_{local} is guaranteed before the algorithm is terminated. The local search in Step 10 of our algorithm, is applied on all individuals in the final population to guarantee convergence to the Pareto optimal front.
 - (b) The maximum number of iterations in local search in Step 4 was fixed based on a detailed study. We present the study and setting in the next subsection.
 3. The final relative stopping tolerance for the KKT (optimality) error is 10^{-4} , for both concurrent-hybrid and serial-hybrid NSGA-II algorithm.

5.3 Parametric study

The maximum number of iterations of the local search in Step 4 has a direct effect on the number of function evaluations consumed by the concurrent-hybrid NSGA-II algorithm. Hence, a detailed study without Step 10 was performed on the test functions with the proposed hybrid algorithm, to fix the maximum number of iterations of the local search. The median values of the best and second best number of iterations of the local search that consumes less number of function evaluations are marked with bold and italic-bold face respectively. The best number of iterations is chosen based on the frequency of best and second best positions attained across all test functions.

The best, median and worst number of function evaluations (with the corresponding hypervolume) for 10 separate runs are shown in Table 1, which contains the following information:

1. ZDT4 and DTLZ1 problems have respectively 100 and 161,050 distinct local Pareto optimal fronts, each of which can attract an algorithm. In such problems, higher number of iterations can lead EMO solutions to locally Pareto optimal solutions. Subsequently, EMO algorithm operators have to aid the algorithm by creating solutions outside the basin of attraction. This can be an extra computational burden on the EMO algorithm. Hence, a lower number of iterations of local search that would generate only improved solutions may suffice and optimal solutions may not be necessary.
2. ZDT1, ZDT2, ZDT3 and DTLZ2 problems have no local Pareto optimal fronts and display an opposite trend. A higher number of iterations of local search is more beneficial for faster convergence, as optimal solutions arising from every local search belong to the Pareto optimal front. In ZDT2, the number of function evaluations

Table 1 The results of the study to find the best number of iterations of the local search for concurrent-hybrid algorithm

Test problem	Type	Number of iterations of the local search—function evaluations (hypervolume)					
		1	2	3	4	5	15
ZDT1	Best	6,960 (0.9278)	4,969 (0.9283)	5,057 (0.9280)	4,804 (0.9221)	4,421 (0.9209)	3,818 (0.9130)
	Median	7,720 (0.9280)	6,831 (0.9283)	6,037 (0.9233)	5,357 (0.9284)	5,699 (0.9217)	6,461 (0.9180)
	Worst	8,772 (0.92852)	7,585 (0.92853)	6,368 (0.92819)	6,486 (0.92811)	7,685 (0.92802)	7,995 (0.92824)
ZDT2	Best	8,145 (0.64835)	2,385 (0.21707)	2,294 (0.2100)	1,761 (0.2100)	1,761 (0.2100)	1,761 (0.2100)
	Median	9,445 (0.65231)	7,900 (0.65213)	6,996 (0.65185)	6,076 (0.65245)	5,415 (0.65236)	5,740 (0.65204)
	Worst	10,675 (0.65155)	9,923 (0.65229)	8,658 (0.65092)	8,421 (0.65265)	7,324 (0.65111)	7,356 (0.65111)
ZDT3	Best	7,535 (0.74099)	7,754 (0.77403)	7,235 (0.72231)	9,192 (0.77380)	6,149 (0.71434)	7,576 (0.77404)
	Median	12,892 (0.74099)	10,555 (0.77441)	12,188 (0.77417)	11,971 (0.77440)	10,913 (0.77433)	13,290 (0.77442)
	Worst	23,277 (0.77439)	19,393 (0.77447)	18,951 (0.77440)	13,380 (0.77405)	21,613 (0.77447)	22,817 (0.77446)
ZDT4	Best	19,246 (0.89559)	19,689 (0.92754)	14,637 (0.89281)	16,137 (0.84652)	19,312 (0.77464)	13,732 (0.87181)
	Median	21,916 (0.92951)	23,384 (0.92772)	22,991 (0.92764)	22,860 (0.92761)	24,108 (0.89755)	25,320 (0.90981)
	Worst	24,552 (0.91078)	26,890 (0.92662)	29,914 (0.92773)	26,916 (0.92854)	29,253 (0.92811)	34,159 (0.92883)
3-DTLZ1	Best	42,805 (1.1994)	39,817 (1.1966)	40,844 (1.2007)	25,581 (1.1917)	17,212 (1.2006)	26,442 (1.2048)
	Median	59,241 (1.1979)	53,724 (1.1968)	56,678 (1.1955)	46,136 (1.2045)	42,816 (1.1941)	54,493 (1.1870)
	Worst	74,804 (1.1971)	107,020 (1.2008)	82,809 (1.1915)	87,608 (1.2045)	64,255 (1.1943)	77,718 (1.1696)
3-DTLZ2	Best	23,056 (0.87335)	17,507 (0.86339)	14,970 (0.87210)	15,871 (0.8809)	16,438 (0.87149)	14,887 (0.88380)
	Median	26,090 (0.86748)	23,555 (0.87200)	21,713 (0.87413)	23,639 (0.87470)	20,718 (0.88400)	18,010 (0.86471)
	Worst	29,449 (0.87441)	28,302 (0.87041)	24,730 (0.87959)	24,511 (0.88544)	25,611 (0.87754)	21,381 (0.87734)
4-DTLZ2	Best	52,366 (1.0538)	44,942 (1.0642)	51,056 (1.0939)	54,280 (1.0954)	48,765 (1.1022)	39,996 (1.1013)
	Median	69,849 (1.0869)	62,095 (1.0822)	58,071 (1.00671)	63,077 (1.0736)	63,093 (1.0863)	62,031 (1.0984)
	Worst	86,946 (1.0965)	99,120 (1.0917)	86,795 (1.0997)	185,573 (1.1112)	184,582 (1.0902)	69,114 (1.0773)

stabilized after four iterations of local search, as four iterations were sufficient for local search to terminate with the preset KKT error.

From Table 1 we can see that, five iterations of the local search is better in most test instances. However, any number of iterations of the local search between 4 and 15 can be a reasonable choice for any practical problem.

The maximum number of iterations was fixed to be a very high positive integer in Step 10 (10,000 in both the concurrent and the serial hybrid NSGA-II algorithms). We consider this to be a safe choice, as the local search terminates by its inbuilt termination criterion based on the KKT tolerance in the KNITRO and not on the maximum number of function evaluations reached.

5.4 Comparative studies

In this subsection we compare our algorithm with the serial-hybrid NSGA-II algorithm. In Table 2, we show the best, median and worst numbers of function evaluations for 10 separate runs for both algorithms with each of the previously mentioned test problems. HV values are presented in parentheses in Table 4.

The better algorithm in terms of the smallest number of required function evaluations is marked in bold face. Median values are used henceforth for all comparisons, as it is a better measure of central tendency of our data set. From the results, it is clearly seen that the concurrent-hybrid algorithm uses a lower number of function evaluations as compared to the serial-hybrid NSGA-II algorithm in every test problem.

In most problems, the corresponding HV values of the best run for both the algorithms are comparable in magnitude, suggesting that convergence has not been achieved with loss in diversity. In the concurrent-hybrid NSGA-II algorithm on local search, it is more likely that individuals far from the current population are created. We refer to these individuals as super individuals. The increased selection pressure on these super individuals can cause a rapid deletion of the dominated solutions. Subsequently, the diversity in the population decreases. On one hand, the super individuals are beneficial in the sense of increasing the convergence speed of the concurrent-hybrid NSGA-II algorithm to the Pareto optimal front, on the other hand a liability in the sense of decreasing the spread of non-dominated solutions. In ZDT2, it is interesting to see a very low HV value corresponding to the best run for the concurrent-hybrid NSGA-II algorithm, which is probably resulting from the creation of super individuals.

In the tests, where SQP was used as a local solver, the approximate gradients were computed by forward differences to show a generalized procedure when the exact gradient was not available. Although our algorithm with approximate gradients outperformed the serial-hybrid NSGA-II algorithm, it was questionable whether in some real world engineering problems such a high number is allowed. Function evaluations in Table 2 can further be reduced by supplying the exact gradients or using non-gradient based algorithms for local search. Table 3 compares the number of function evaluations taken by the exact versus approximate derivative based concurrent-hybrid NSGA-II algorithm with each of the test problems. The results show on an average around 44 percentage decrease in function evaluations, when the exact derivative was used. This further strengthens our motivation for a fast and accurately converging hybrid evolutionary multi-objective algorithm.

Table 2 Comparison of the number of function evaluations for the concurrent and serial approaches

Test problem	Serial approach		Concurrent approach			
	Best	Median	Worst	Best	Median	Worst
ZDT1	30,083 (0.9289)	31,043 (0.9283)	33,468 (0.9285)	13,328 (0.9214)	14,518 (0.9285)	16,991 (0.9286)
ZDT2	29,384 (0.6526)	31,760 (0.6530)	32,344 (0.6532)	1,861 (0.2100)	13,748 (0.6513)	15,716 (0.6510)
ZDT3	33,691 (.7738)	37,325 (0.7742)	38,545 (0.7742)	16,595 (0.7155)	20,866 (0.7744)	29,628 (0.7744)
ZDT4	35,006 (0.9274)	54,214 (0.9284)	63,584 (0.9286)	34,459 (0.9286)	37,724 (0.8982)	43,142 (0.9286)
3-DTLZ1	201,957 (1.664)	252,952 (1.1965)	351,954 (1.1964)	66,369 (1.1995)	146,506 (1.1931)	290,792 (1.2002)
3-DTLZ2	35,757 (0.8694)	43,722 (0.8813)	70,606 (0.8687)	26,665 (0.8705)	31,604 (0.8765)	36,006 (0.8803)
4-DTLZ2	69,449 (1.0861)	93,835 (1.0701)	128,794 (1.0750)	61,028 (1.0960)	74,187 (1.0834)	194,581 (1.0782)

Table 3 Comparison of the number of function evaluations using the exact and approximate gradients in the concurrent-hybrid algorithm

Test problem	Exact gradient			Approximate gradient		
	Best	Median	Worst	Best	Median	Worst
ZDT1	3,751	4,354	5,189	13,328	14,518	16,991
ZDT2	1,706	4,510	5,721	1,861	13,748	15,716
ZDT3	14,879	17,340	23,687	16,595	20,886	29,628
ZDT4	18,763	21,975	26,148	34,459	37,724	43,142
3-DTLZ1	40,031	85,763	120,964	66,369	146,506	290,792
3-DTLZ2	15,017	19,230	24,380	26,665	31,604	36,006
4-DTLZ2	26,672	48,330	56,887	61,128	74,187	194,581

Table 4 Comparison of hypervolume for the concurrent and serial approach based NSGA-II

Test problem	Serial approach			Concurrent approach		
	Best	Median	Worst	Best	Median	Worst
ZDT1	0.9291	0.9287	0.9283	0.9289	0.9276	0.9214
ZDT2	0.6534	0.6530	0.6526	0.6531	0.6518	0.2100
ZDT3	0.7743	0.7742	0.7738	0.7744	0.7737	0.7155
ZDT4	0.9287	0.9286	0.9274	0.9287	0.9280	0.7758
3-DTLZ1	1.1981	1.1947	1.1664	1.2040	1.1994	1.1931
3-DTLZ2	0.8813	0.8694	0.8615	0.8850	0.8765	0.8645
4-DTLZ2	1.0983	1.0765	1.0602	1.0993	1.0857	1.0691
WRP	0.5703	0.5647	0.5635	0.5706	0.5660	0.5644
WELD	1.4196	1.4193	1.4082	1.4198	1.4188	1.4143

Larger value is better and marked in bold face

An explicit diversity mechanism has not been introduced in our hybrid algorithm, except the NSGA-II's original crowding distance. We present the HV values reached in 25,000 function evaluations for all the test and practical problems in Table 4. In bi-objective problems, HV values in the serial approach are slightly better than in the concurrent approach. The reason can be attributed to creation of super-individuals, which may decrease the number of non-dominated solutions. However, with an additional diversity preservation mechanism, we are confident of improving this discrepancy. With more objectives, the number of solutions a given solution dominates decreases, which provides extra diversity to the algorithm. Hence, HV values of our concurrent algorithm are better than in the serial-hybrid NSGA-II algorithm. Overall, one can say that, the savings in the number of function evaluations are so dramatic that the usefulness of the concurrent-hybrid algorithm is clear.

Some of the ideas that could be considered for ensuring diversity, are: Firstly, the local search direction can be biased differently for different EMO solutions based on the location of an EMO solution and on the undiscovered regions of the Pareto optimal front. Secondly, the crowding distance operator of the NSGA-II can be replaced with a better diversity preserving procedure, such as clustering (Kukkonen and Deb 2006; Zitzler et al. 2001) and lastly, instead of using a generational evolutionary optimization approach such as NSGA-II, a steady-state procedure [such as in the epsilon-MOEA (Deb et al. 2003)] may

be adopted. In this way, every new solution created by a hybrid approach can be evaluated for its convergence and diversity enhancement properties against the rest of the non-dominated solutions of an EMO population.

6 Understanding the dynamics of the concurrent-hybrid NSGA-II algorithm

In this section we discuss two concepts used in our concurrent hybrid NSGA-II algorithm: firstly, the value of the slack variable (α), as a measure of convergence and secondly, effect of introducing locally optimal points on convergence. To explain these concepts we consider bi-objective test problems.

6.1 Slack variable as a measure of convergence

Let us first test the termination criterion in Step 9 of our concurrent-hybrid NSGA-II algorithm proposed in Sect. 4.3 on ZDT1 and ZDT4 test problems. ZDT1 has no local Pareto optimal fronts, whereas ZDT4 has 100 distinct Pareto optimal fronts and we want to test the efficacy of the new method in both cases. Figures 2 and 3 shows the variation of

Fig. 2 Variation of slack variable with generation in ZDT1

Fig. 3 Variation of slack variable with generation in ZDT4

average slack variable values (α) with generations. Here α indicates the distance of the reference point from the local Pareto optimal front. Hence, the absolute value of the average α parameter decreases as we choose reference points near to the Pareto optimal front and is zero when the reference point is Pareto optimal.

In ZDT1, average α values start with a large absolute value in initial generations, as points are far away from the Pareto optimal front initially and soon increase to zero. This is due to the fact that there is no locally Pareto optimal front in ZDT1 and there is no reason for an algorithm to get stuck to any intermediate front. However in ZDT4, the average α values follow a zigzag behavior and are zero whenever an individual chosen for the local search is on the local Pareto optimal front. When individuals are on a local Pareto optimal front, the EMO recombination operators aid the algorithm to overcome local optimality, by creating solutions outside the local basin of attraction. The average α value changes again, which can also be seen in Fig. 3. This distinction in the performance of an algorithm in ZDT1 (unimodal) and ZDT4 (multimodal) has not been so clearly observed in any previous study.

6.2 Effect of the local search on convergence

Let us next briefly discuss the working principle of the local search by examining its performance on ZDT1 and ZDT4 problems. In Figs. 4 and 5, we plot the average error metric values versus the generation counter for both the approaches. The vertical lines

Fig. 4 Average error metric for ZDT1

Fig. 5 Average error metric for ZDT4

Fig. 6 Populations approach the Pareto optimal front faster in the concurrent-hybrid NSGA-II–ZDT1. Local search at generation 8 produces three good (two weakly Pareto-optimal and one Pareto-optimal) solutions

indicate the generations at which at least one local search is executed. It is interesting to observe that soon after the first local search has taken place at generation 8 and 7 in ZDT1 and ZDT4, respectively, the error metric values drop sharply and the population gathers momentum to move towards the Pareto optimal front for both the problems. Figure 6 shows all 100 population members at the start of generations 7, 8 and 13 in ZDT1.

For example, for ZDT1, the first local search takes place at generation 8 and three good solutions are found (shown with a big triangle in Fig. 6). The presence of these solutions in the population and its subsequent recombination with other population members causes the population to come closer to the Pareto optimal front in next few generations. The population at generation 13 is shown in Fig. 6 in diamonds. To demonstrate the effect of the local search, next we run the serial-hybrid NSGA-II in which no local search is performed at generation 8. The initial population and other NSGA-II parameters are kept the same. Populations at generations 13 and 20 are shown in Fig. 7. Although they maintain a similar error metric value till generation 7, the serial-hybrid NSGA-II algorithm fails to keep pace with the concurrent-hybrid NSGA-II algorithm thereafter.

Fig. 7 Populations approach the Pareto optimal front slowly in the serial hybrid NSGA-II–ZDT1. No local search is used at generation 8

It should be noted that it is hard to plot the populations with respect to function evaluations, as the algorithms work generation-wise. The concurrent-hybrid and serial-hybrid algorithm have consumed 1,957 and 1,300 function evaluations respectively at generation 13. To demonstrate the status with comparable function evaluations for both algorithms, we also plotted the population at generation 20 for the serial-hybrid algorithm, for which 2,000 function evaluations have taken place. Figure 7 shows the position of the population in generation 13 of the concurrent-hybrid algorithm to be ahead of the serial-hybrid algorithm in generation 20 (after the same number of function evaluations).

7 Conclusion

In this paper, we have argued that an efficient implementation (balance between exploration and exploitation) of a local search procedure in an EMO algorithm is not straightforward. To take advantage of fast and accurate convergence to the Pareto optimal solutions, EMO algorithms must use a continuous, directed and provable local search procedure (a mathematical programming technique with convergence proofs). In this study, we have proposed a concurrent-hybrid NSGA-II algorithm, where we have suggested an augmented achievement scalarizing function to be optimized with a local search method. The local search procedure has been implemented as an additional operator in a concurrent approach and applied to EMO populations with a varying probability. There are two prominent advantages for an achievement scalarizing function: an optimal solution of an achievement scalarizing function is always a Pareto optimal solution and any properly Pareto optimal solution can be obtained by changing the reference point. This has definitely helped EMO algorithms to improve performance. We have successfully solved several standard test problems, involving two to four objectives and two practical problems in a computationally faster way, with good diversity and accuracy.

This study has addressed a fundamental aspect of hybridizing an EMO algorithm and a local search. As the explanations and simulation results indicate, the extent of local search must be balanced with respect to global search in an EMO algorithm. This balance of exploiting current non-dominated solutions by a local search and the limited frequency of local search for allowing a proper exploration of the search space by an evolutionary algorithm is vital for a successful working of the hybrid algorithm. Here, we have used a carefully designed probability distribution function for local search to achieve this exploitation-exploration balance. However, other techniques and variants of probability functions can also be used. It remains important to investigate which mechanism would provide more flexibility and stability in the performance of the hybrid algorithm.

Achieving convergence and diversity of solutions to the Pareto optimal front are our quest. Although in this study we have relied on the NSGA-II crowding distance operator for diversity preservation, the faster and accurate convergence achieved with the proposed local search is now ready to be coupled with a more efficient and explicit diversity ensuring operator. However, the present study has clearly shown the advantage and potential of hybridizing a local search in a fast and accurate computation of Pareto optimal solutions.

Acknowledgments The support provided by the Academy of Finland (grant 118319) and the Jenny and Antti Wihuri foundation is highly appreciated.

References

- Beyer HG, Schwefel HP (2002) Evolution strategies—a comprehensive introduction. *Nat Comput* 1(1):3–52
- Byrd RH, Nocedal J, Waltz RA (2006) KNITRO: an integrated package for nonlinear optimization. In: Pillo G, Roma M (eds) *Large-scale nonlinear optimization*. Springer, Berlin, pp 35–39
- Chankong V, Haimes YY (1983) *Multiobjective decision making theory and methodology*. Elsevier, New York
- Coello, CAC, Lamont, GB (eds) (2004) *Applications of multi-objective evolutionary algorithms*. World Scientific, Singapore
- Coello CAC, Lamont GB, Veldhuizen DAV (2007) *Evolutionary algorithms for solving multi-objective problems*. 2nd edn. Springer, New York
- Deb K (2001) *Multi-objective optimization using evolutionary algorithms*. Wiley, Chichester
- Deb K (2008) A robust evolutionary framework for multi-objective optimization. In: Keijzer M et al (eds) *Proceedings of the 10th annual conference on Genetic and evolutionary computation (GECCO 2008)*. ACM, New York, pp 633–640
- Deb K, Sindhya K (2008) Deciphering innovative principles for optimal electric brushless D.C. permanent magnet motor design. In: *Proceedings of the world congress on Evolutionary computation (CEC 2008)*. IEEE Press, Piscataway, pp 2283–2290
- Deb K, Agarwal S, Pratap A, Meyarivan T (2002a) A fast and elitist multi-objective genetic algorithm: NSGA-II. *IEEE Trans Evol Comput* 6(2):182–197
- Deb K, Thiele L, Laumanns M, Zitzler E (2002b) Scalable multi-objective optimization test problems. In: *Proceedings of the congress on Evolutionary computation (CEC 2002)*, vol 1. IEEE Press, Piscataway, pp 825–830
- Deb K, Mohan M, Mishra S (2003) Towards a quick computation of wellspread Pareto-optimal solutions. In: Fonseca CM et al (eds) *Proceedings of the evolutionary multi-criterion optimization (EMO 2003)*. Springer, Berlin, pp 222–236
- Deb K, Srinivasan A (2008) Innovization: discovery of innovative design principles through multiobjective evolutionary optimization. In: Knowles J (eds) *multiobjective problem solving from nature*. Springer, Berlin, pp 243–262
- Emmerich M, Deutz A, Beume N (2007) Gradient-based/evolutionary relay hybrid for computing Pareto front approximations maximizing the S-metric. In: Bartz-Beielstein T et al (eds) *Proceedings of the 4th international workshop on Hybrid metaheuristics (HM 2007)*. Springer, Berlin, pp 140–156
- Galperin EA, Wiecek MM (1999) Retrieval and use of the balance set in multiobjective global optimization. *Comput Math Appl* 37:111–123
- Glover F (1989) Tabu search-part I. *ORSA J Comput* 1(3):190–206
- Goel T, Deb K (2002) Hybrid methods for multi-objective evolutionary algorithms. In: Wang L, Tan KC, Furuhashi T, Kim JH, Yao X (eds) *Proceedings of the 4th Asia-Pacific conference on Simulated evolution and learning (SEAL 2002)*, vol 1. Nanyang Technical University, Singapore, pp 188–192
- Hansen MP (2000) Use of substitute scalarizing functions to guide a local search based heuristic: the case of moTSP. *J Heuristics* 6(3):419–431
- Harada K, Sakuma J, Kobayashi S (2006) Local search for multiobjective function optimization: Pareto descent method. In: Keijzer M et al (eds) *Proceedings of the 6th annual conference on Genetic and evolutionary computation (GECCO 2006)*. ACM, New York, pp 659–666
- Ishibuchi H, Murata T (1998) A multi-objective genetic local search algorithm and its applications to flowshop scheduling. *IEEE Trans Syst Man Cybernet C* 28(3):392–403
- Ishibuchi H, Narukawa K (2004) Some issues on the implementation of local search in evolutionary multiobjective optimization. In: Deb K et al (eds) *Proceedings of the genetic and evolutionary computation (GECCO 2004)*. Springer, Berlin, pp 1246–1258
- Ishibuchi H, Yoshida T, Murata T (2003) Balance between genetic search and local search in memetic algorithms for multiobjective permutation flowshop scheduling. *IEEE Trans Evol Comput* 7(2):204–223
- Ishibuchi H, Doi T, Nojima Y (2006) Incorporating of scalarization fitness functions into evolutionary multiobjective optimization algorithms. In: Runarsson TP et al (eds) *Proceedings of the parallel problem solving from nature (PPSN IX 2006)*. Springer, Berlin, pp 493–502
- Jaszkiewicz A (2002) Genetic local search for multiple objective combinatorial optimization. *Eur J Oper Res* 137(1):50–71
- Kirkpatrick S, Gelatt CD Jr, Vecchi MP (1983) Optimization by simulated annealing. *Science* 220(4598):671–680
- Knowles JD, Corne DW (2000a) Approximating the non-dominated front using the Pareto archived evolutionary strategy. *Evol Comput* 8(2):149–172

- Knowles JD, Corne DW (2000b) M-PAES: a memetic algorithm for multiobjective optimization. In: Proceedings of the congress on Evolutionary computation (CEC 2000), vol 1. IEEE Press, Piscataway, pp 325–332
- Kuhn HW, Tucker AW (1951) Nonlinear programming. In: Neyman J (ed) Proceedings of the second Berkeley symposium on Mathematical statistics and probability. University of California Press, Berkeley, pp 481–492
- Kukkonen S, Deb K (2006) Improved pruning of non-dominated solutions based on crowding distance for bi-objective optimization problems. In: Proceedings of the congress on Evolutionary computation (CEC 2006). IEEE Press, Piscataway, pp 1179–1186
- Laumanns M, Thiele L, Deb K, Zitzler E (2002) Combining convergence and diversity in evolutionary multi-objective optimization. *Evol Comput* 10(3):263–282
- Levia HA, Esquivel SC, Gallard RH (2000) Multiplicity and local search in evolutionary algorithms to build the Pareto front. In: Proceedings of the XXth international conference of the Chilean Computer Science Society, pp 7–13
- Luque M, Miettinen K, Eskelinen P, Ruiz F (2009) Incorporating preference information in interactive reference point methods for multiobjective optimization. *Omega* 37(2):450–462
- Miettinen K (1999) Nonlinear multiobjective optimization. Kluwer, Boston
- Miettinen K, Mäkelä MM, Kaario K (2006) Experiments with classification based scalarizing functions in interactive multi-objective optimization. *Eur J Oper Res* 175(2):931–947
- Miettinen K, Ruiz F, Wierzbicki AP (2008) Introduction to multiobjective optimization: interactive approaches. In: Branke J (eds) Multiobjective optimization: interactive and evolutionary approaches. Springer, Berlin
- Osyczka A (1984) Multicriterion optimization in engineering with Fortran programs. Ellis Horwood Limited, Chichester
- Sawaragi Y, Nakayama H, Tanino T (1985) Theory of multiobjective optimization. Academic Press, Orlando
- Shaw KJ, Nortcliffe AL, Thomson M, Love J, Fleming PJ, Fonseca CM (1999) Assessing the performance of multiobjective genetic algorithms for optimization of a batch process scheduling problem. In: Proceedings of the congress on Evolutionary computation (CEC 1999). IEEE Press, Piscataway, pp 37–45
- Sindhya K, Deb K, Miettinen K (2008) A local search based evolutionary multi-objective approach for fast and accurate convergence. In: Rudolph G et al (eds) Proceedings of the Parallel Problem Solving from Nature (PPSN X 2008), Springer, Berlin/Heidelberg, pp 815–824
- Talbi EG, Rahoual M, Mabed MH, Dhaenens C (2001) A hybrid evolutionary approach for multicriteria optimization problems: Application to the flow shop. In: Zitzler E et al (eds) Proceedings of the evolutionary multi-criterion optimization (EMO 2001), Springer, Berlin, pp 416–428
- Tapia C, Coello CAC (2007) Applications of multi-objective evolutionary algorithms in economics and finance: a survey. In: Proceedings of the congress on Evolutionary computation (CEC 2007), IEEE Press, pp 532–539
- Wierzbicki AP (1980) The use of reference objectives in multiobjective optimization. In: Fandel G, Gal T (eds) MCDM theory and application. Springer, Hagen, pp 468–486
- Wierzbicki AP (1986) On the completeness and constructiveness of parametric characterizations to vector optimization problems. *OR Spektrum* 8:73–87
- Zitzler E, Thiele L (1998a) An evolutionary algorithm for multiobjective optimization: the strength Pareto approach. Technical Report 43, Computer Engineering and Networks Laboratory, ETH
- Zitzler E, Thiele L (1998b) Multi-objective optimization using evolutionary algorithms—a comparative case study. In: Eiben AE et al (eds) Proceedings of the parallel problem solving from nature (PPSN V 1998). Springer, Berlin, pp 292–301
- Zitzler E, Laumanns M, Thiele L (2001) SPEA2: Improving the strength Pareto evolutionary algorithm for multiobjective optimization. In: Giannakoglou KC et al (eds) Proceedings of the evolutionary methods for design optimization and control with applications to industrial problems (EUROGEN 2001). International Center for Numerical Methods in Engineering (CIMNE), pp 95–100