Towards a Better Diversity of Evolutionary Multi-Criterion Optimization Algorithms using Local Searches

Haitham Seada[1*], Mohamed Abouhawwash[2], Kalyanmoy Deb[3]

[1]Department of Computer Science and Engineering, Michigan State University
428 South Shaw Lane, East Lansing, MI 48824, USA, seadahai@msu.edu
[2]Department of Mathematics, Faculty of Science, Mansoura University
Mansoura 35516, Egypt, saleh1284@mans.edu.eg
[3]Department of Electrical and Computer Engineering, Michigan State University
428 South Shaw Lane, East Lansing, MI 48824, USA, kdeb@egr.msu.edu

[*]To whom correspondence should be addressed

COIN Report Number 2016004

http://www.coin-laboratory.com

Abstract

EMO algorithms use a population of solutions to find a representative set of Pareto-optimal solutions. Since the diversity of obtained solutions is also an equally important factor, particularly for an eventual decision-making process, EMO researchers have been paying attention to both convergence and diversity preservation methods pro-actively in designing an EMO algorithm. NSGA-III is a recently proposed reference direction based algorithm that was shown to be successful up to as many as 15 objectives. NSGA-III uses a direction-wise population count as a niching method to preserve diversity within a population. In this study, we propose a diversity-enhanced version of NSGA-III using a biased weighted-sum based local search to attain extreme Pareto solutions. The identification of extreme points enables the algorithm to use an achievement scalarization function (ASF) based local search to cover the entire Pareto front more reliably than the original NSGA-III procedure. The two local search optimizers are carefully weaved into the fabric of NSGA-III niching procedure. The final algorithm, maintains the total number of functions evaluations to a minimum, enables using small population sizes, and achieves higher diversity without sacrificing convergence property on a number of multi and many-objective problems.

1 Introduction

Traditionally, single objective evolutionary optimization researchers focused on the convergence abilities of their algorithms. However, since its advent, Evolutionary Mutliobjective Optimization (EMO) introduced another key concept to the field, diversity. EMO switched the interest from only one solution to a set of competing solutions representing the trade-off among conflicting objectives, the Pareto front. Most early EMO algorithms relied on the concept of Pareto domination [*3*] for convergence. This practice continued until today. The major difference between these algorithms was in the techniques they use to maintain diversity among solutions. Several diversity preservation approaches were borrowed from single-objective evolutionary computation literature [*5, 7*].

SPEA2 [*15*] and NSGA-II [*4*]. are very good examples of successful algorithms that dominated the. field for years. They were however unable to maintain diversity in more than two objectives [*10, 12*]. Maintaining diversity in more than three objectives remained an obstacle until Zhang and Li proposed MOEA/D in 2007 [*14*]. Instead of the widely used crowding distance operator adopted by NSGA-II, MOEA/D used a decomposition based approach where a multiobjective optimization problem is divided into a number of different single objective optimization subproblems. Along with a population based algorithm, MOEA/D was able to solve up to four objectives. In 2014, Deb and Jain, proposed NSGA-III [*6*] [*9*]. Their algorithm used a predefined evenly distributed set of reference directions to guide the search procedure. With a carefully designed normalization mechanism, their algorithm was shown to successfully maintains diversity up to fifteen objectives.

All these efforts remained in their majority isolated form mathematical optimization techniques and even from single-objective evolutionary optimization algorithms. Ishibuchi IM-MOGLS [*8*] is considered the first attempt to use mathematical optimization in the context of EMO. His idea was simply to combine several objectives using a weighted sum approach and start a local search from each individual in his combined parents-offspring population. Subsequent researchers followed Ishibuchi's steps [*11*]. The use of Achievement Scalarization Functions (ASF) in EMO has also been explored in a few studies among which are Bosman's [*2*] and the work of Sindhya et al. [*13*]

In this study we explore the possibility of integrating a couple of local search mechanisms into a many-objective optimization algorithm.

Our candidate many-objective optimization algorithm upon which we build is NSGA-III. NSGA-III, emphasises non-domination thoroughout the whole optimization process. This approach influences the algorithm to seek convergence then diversity, in this specific order. This convergence-comes-first approach that NSGA-III follows has some drawbacks which we summarize in the following points.

1. Losing diversity in early stages of the optimization process might be incurable in later generations, causing the population either to be trapped in local Pareto fronts or lose part(s) of the true Pareto front. The latter effect is usually observed with problems having non-continuous or disconnected Pareto fronts.

2. Most of the association and normalization efforts spent by NSGA-III at early generations are useless. This is true because reference directions keep changing with the least change in any of the extreme points reached so far.

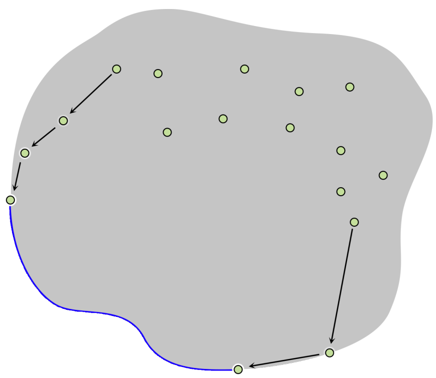

Figure 1: *Phase-1* of Div-NSGA-III

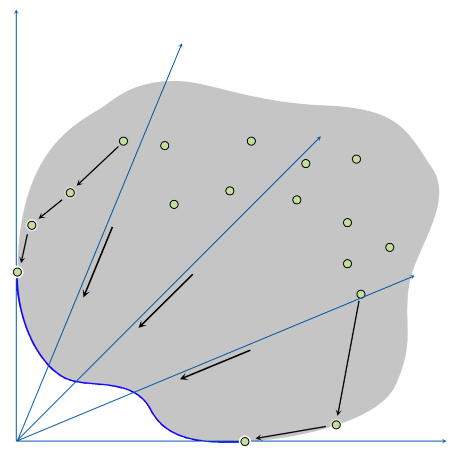

Figure 2: *Phase-2* of Div-NSGA-III

In order to solve these issues, we proposed our new diversity based NSGA-III (Div-NSGA-III). The next section describes our proposed algorithm in detail. Section 3, discusses the theoretical and practical reasons behind employing each of our local search techniques in its place. Section 4, provides extensive simulation results on a set of both unconstrained and constrained, bi-objective multi-objective and many-objective optimization problems. Finally, we conclude this study in Section 5.

2 Diversity-Based NSGA-III

Our proposed approach, has the same general structure of NSGA-III in terms of normalization and relying on reference directions for diversity preservation. However, it has a modified niching mechanism. Instead of continuously emphasising convergence, our approach breaks this pattern, every α generations. The modified is niching mechanism is outlined in Algorithm 1. Initially, the algorithm seeks the extreme points using a biased-weighted-sum local search, starting from the already existing extreme points. We call this *phase-1* (lines 1 to 4). Any problem has a number of extreme points equal to the number of objectives M. If a better extreme point – than the one already existing

– is found, the old one will no longer serve an extreme point. The new point will be used instead for later normalizations. And since The number of iterations required to find the true extreme points of the Pareto front is unknown beforehand, *phase-1* will continue for an undefined number of generations. See Figure 1.

A stagnant extreme point is a one that is not improving anymore using local search. If the algorithm finds itself trying to repeat extreme point local search from the same starting point, it implies that the last local search was unsuccessful and consequently any further local search operations from the same starting point and using the same optimizer will be useless. Once all extreme points reach stagnation, the algorithm shifts its focus to maintaining better diversity among solutions and moves to *phase-2*.

In *phase-2*, our approach re-normalizes the current population using the extreme points attained so far. Association is then performed using the new reference directions. As opposed to the traditional front-by-front accommodation scheme adopted in NSGA-II and NSGA-III, this algorithm selects the closest individual to each reference direction. And since each reference directions represents a niche in the objective space, the algorithm will be trying to fill-in all the gaps in the current front. Regardless of an individual's rank, if it is the only representative of its niche, it will be selected at the expense of – may-be – better ranked individuals which are already outperformed in their own niche (line 4).

The procedure explained above does not guarantee generating enough individuals to fill the next population. And even if it did, some of these individuals might be already dominated. Hence, the proposed algorithm makes an initial attempt to cover empty reference directions (i.e. those having no associations so far), using an ASF based local search. Local search starts from the closest first front point to the empty direction. The algorithm then moves to the final step of *phase-2* which is enhancing those individuals having bad (dominated) associations. And in order to save function evaluations, the total number of local search operations performed per cycle is bounded by the parameter β which is set to 2 in all our our simulations (lines 5 to 9). *choose()* routine in Algorithm 1 refers to choosing the suitable individual and direction according to the logic described above, to conduct a local search. Figure 2 shows *phase-2*.

Finally, the new population is completed by adding the best-ranked individuals among those overlooked by all the previous steps (lines 10 to 14).

It is important to mention that although extreme points can reach a state of stagnation by the end of *phase-1*, this does not mean that it has found the true extremes of the Pareto front. That is why the two phases of Div-NSGA-III are always alternating. So, if during evolution, a better extreme point was found using the traditional genetic operators, the algorithm switches back immediately to *phase-1* and waits for another state of stagnation. The key idea behind this setup is that although local search can be very useful, it is not sufficient alone. It is after all a classical single-objective point by point procedure, which is prone to be trapped in local optima, and unable to solve a whole multi-objective optimization problem through decomposition. On the other hand, EMO algorithms are powerful, reliable but lack the ability to focus or even identify those points/sections that need more attention. Hence, we propose this approach of seamless information exchange and alternation between the two techniques.

3 Local Search Formulations

In this study we employ Matlab's® `fmincon()` optimization routine. First, we formulate the local search operation in the form of a single-objective optimization problem, then `fmincon()` is applied to reach a solution. As mentioned in Section 2, our approach includes two types of local search, one for searching extreme points (*phase-1*)

Algorithm 1 Modified NSGA-III Niching

Input: merged population (G), population size (N), reference directions (D), ideal point (I), intercepts (T), maximum number of function evaluations ($FeMax$), maximum number of local search operations per iteration β

Output: New Population P'

$F \leftarrow getFeasible(P)$
$E \leftarrow getExtremePoints(F)$
$E(i) \leftarrow BWS_i(FeMax), i = 1, \ldots, M$ % *phase-1*
$P' \leftarrow getBestInNiche(d, G), \forall d \in D$
if $stagnant(E)$ **then**
  for i = 1 to β **do** % *phase-2*
    $P' \leftarrow ASF(choose(x), I, T, FeMax)$
  end for
end if
while $|P'| \leq N$ **do**
  $x \leftarrow highestRank(G)$
  $G \leftarrow G \setminus x$
  $P' \leftarrow x, s.t. x \notin P'$
end while

and the other for locally searching solutions along some specific reference direction (*phase-2*). Although both types of local search use the same optimizer, they differ in the way the problem is formulated.

Extreme points local search optimizers are formulated in a Biased Weighted Sum (BWS) form. Equation 1 shows how a minimization problem aiming at finding extreme point (i) is formulated. Note that $\tilde{f}_k(x)$ is the normalized value of objective k. For all our simulations we use $\varepsilon = 0.01$.

$$\begin{aligned} \underset{\mathbf{x}}{\text{Minimize}} \quad & \mathrm{BWS_i}(\mathbf{x}) = \varepsilon \tilde{f}_i(x) + \sum_{j=1,\ j \neq i}^{M} w_j \tilde{f}_j(x), \\ \text{subject to} \quad & \varepsilon << min_{j=1, j \neq i}^{M} w_j \end{aligned} \tag{1}$$

The first term in Equation 1 is included to avoid weak extreme points. Thanks to this term, weak extreme points will have larger objective values than their true extreme counterparts.

For directional local search, an Achievement Scalarization Function (ASF) is used, as shown in Equation 2. Given a specific reference direction, an ASF is a means of converting a multi-objective optimization problem to a single-objective optimization problem whose optimal solution is the intersection point between the given direction and the original Pareto front.

$$\begin{aligned} \underset{\mathbf{x}}{\text{Minimize}} \quad & \text{ASF}(\mathbf{x}, \mathbf{z}^r, \mathbf{w}) = \max_{i=1}^{M} \left(\frac{\tilde{f}_i(x) - u_i}{w_i} \right), \\ \text{subject to} \quad & g_j(\mathbf{x}) \leq 0, \quad j = 1, 2, \ldots, J. \end{aligned} \tag{2}$$

The reason behind using two different formulations instead of one is that each of them has its merits and weaknesses. A weighted sum approach is straightforward, easy to understand and implement and – as will be shown later – easy to optimize using `fmincon()`. But, it is theoretically unable to attain any solution lying on a non-convex section of the front. Consequently, it cannot be used as a general local search procedure for finding Pareto points. However, since extreme points by their very nature can never be located on non-convex sections of the front, weighted sum can theoretically reach any extreme point. Combined with its simplicity and ease of use, BWS becomes the perfect choice for searching extreme points. On the other hand, ASF can – in general – reach any Pareto point if the appropriate reference direction is provided, which makes it the perfect choice to be used as a general local search mechanism with a reference direction based algorithm like NSGA-III. But when it comes to extreme points, ASF faces some serious problems that we summarize in the next points:

1. Unless the starting point is placed in a certain position with respect to the extreme

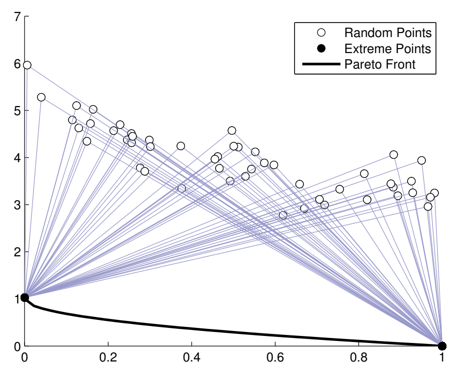

Figure 3: Extreme LS in ZDT1

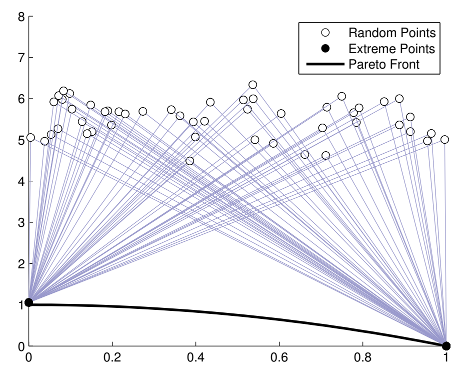

Figure 4: Extreme LS in ZDT2

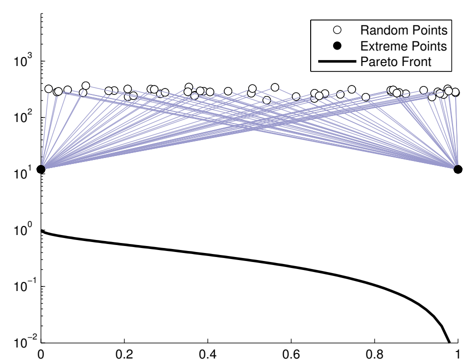

Figure 5: Extreme LS in ZDT4

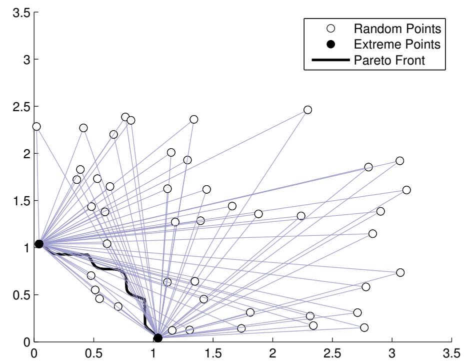

Figure 6: Extreme LS in TNK

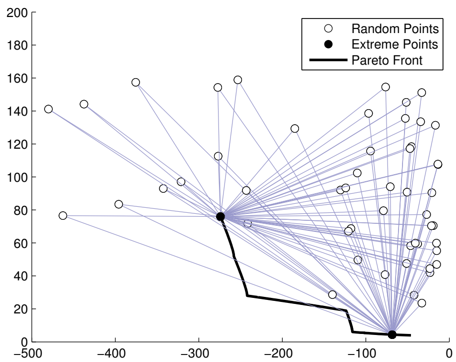

Figure 7: Extreme LS in OSY

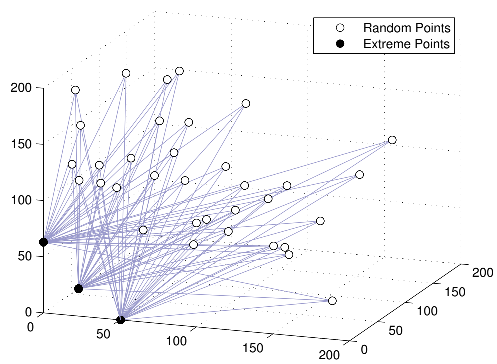

Figure 8: Extreme LS in DTLZ1

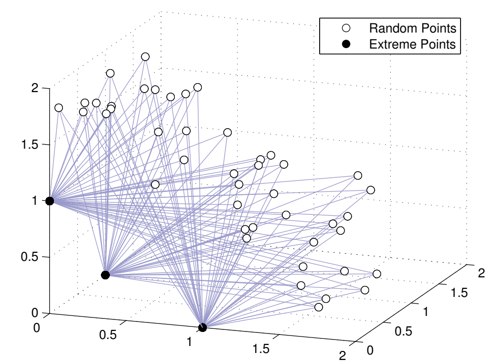

Figure 9: Extreme LS in DTLZ2

point (inferior to the extreme point in all objectives), an ASF based local search will never be able to reach it.

2. We observed that, the more flat/steep the provided direction is, the more difficult the problem is for `fmincon()` to solve.

For all these reasons, we use a BWS based local search in *phase-1* to seek extreme points, and an ASF based local search in *pahse-2* to cover gaps and enhance internal diversity. In Figures 3,4,5, we show how `fmincon()` can reach extreme points with ease and consistency. In each of these figures, a random population is created. Each individual of this population is used as a starting point for local search operations (BWS_1, BWS_2, ..., BWS_M) seeking the M extreme points of the problem in hand. To validate the efficacy of our approach, each optimization problem was given a maximum limit of 1000 function evaluations per single local search operation (the actual limit in our simulation results will be much smaller than that). The same experiment is performed for constrained bi-objective optimization problems as shown in Figures 6 and 7. It is interesting to observe how starting from an infeasible point the optimizer can reach the true extreme point or come close to it. Figures 8 and 9 show the same effect in the three objectives version of DTLZ1 and DTLZ2. It is worth noting that ASF based local searches were unable to handle even some of the easiest problems (e.g. ZDT2) due to the reasons discussed earlier. On the contrary, BWS based local search, can reach the true extreme points of easy problems in one attempt. However, for harder problems, one local search per extreme point is not sufficient. In these situations, our proposed approach with its inherent information exchange and alternation of phases should shows its aptitude.

4 Results

In this section we compare our Div-NSGA-III to NSGA-III across a wide variety of bi, multi and many-objective optimization problems. Both constrained and unconstrained. Over 31 runs, we show the merits of using our approach. For fair comparisons in each experiment, the total number of function evaluations per run is kept equal. The rest of parameters used are listed in Table 1

Table 1: Parameters - From left to right: problem name, population size, total number function evaluations per run, local search frequency, maximum limit of function evaluations per each local search operation

Problem	**N**	**FE**	α	**Max. LS FE**
ZDT1	40	9757	10	100
ZDT2	40	9783	10	100
ZDT3	40	11941	10	100
ZDT4	48	18767	10	100
ZDT6	40	9413	10	100
VarDens	40	9381	10	300
TNK	100	21380	10	300
BNH	40	12213	10	300
SRN	60	18141	10	300
OSY	48	27264	10	500
DTLZ1(3)	92	47733	10	500
DTLZ1(5)	212	106947	10	500
DTLZ1(10)	276	138276	10	500
DTLZ2(3)	92	38011	10	500
DTLZ2(5)	212	86460	10	500
DTLZ2(10)	276	113948	10	500

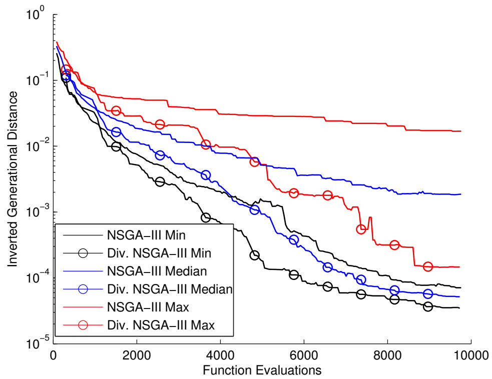

Figure 10: IGD vs. number of function evaluations (ZDT1)

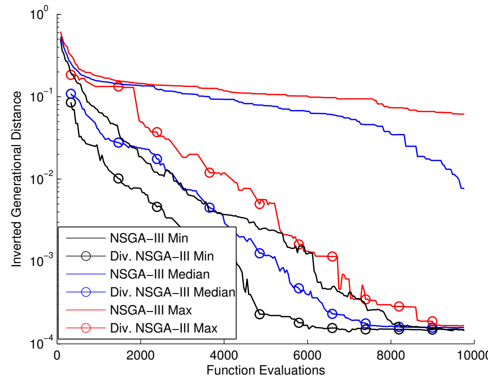

Figure 11: IGD vs. number of function evaluations (ZDT2)

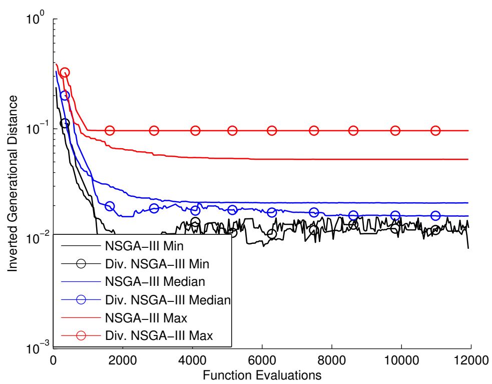

Figure 12: IGD vs. number of function evaluations (ZDT3)

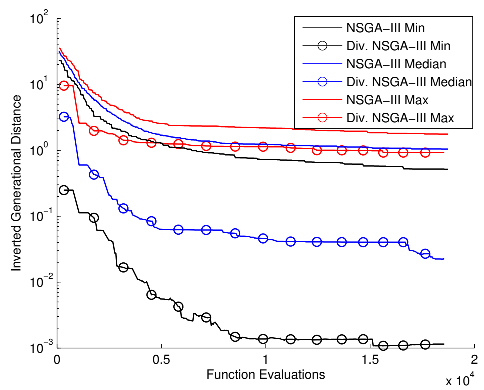

Figure 13: IGD vs. number of function evaluations (ZDT4)

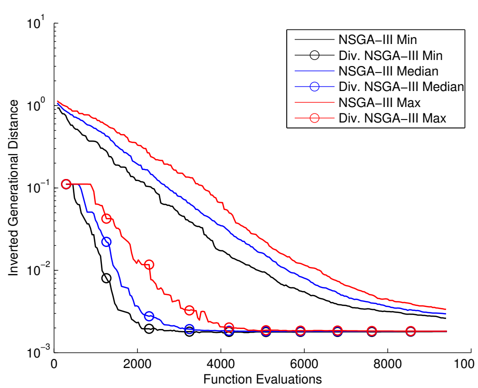

Figure 14: IGD vs. number of function evaluations (ZDT6)

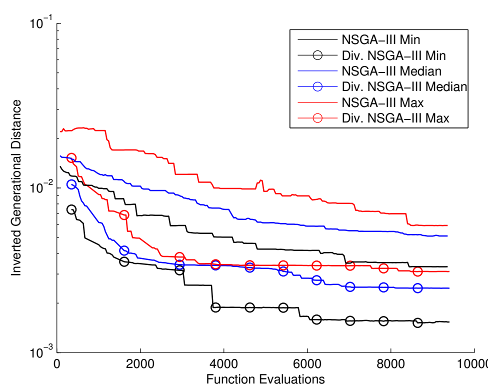

Figure 15: IGD vs. number of function evaluations (VarDens)

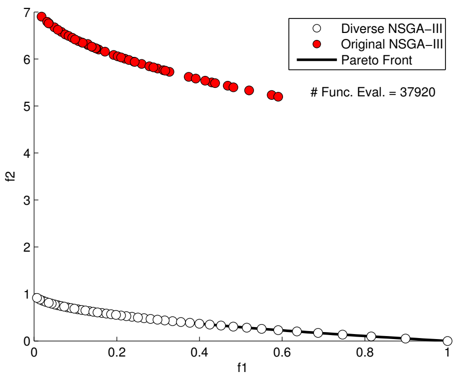

Figure 16: Median fronts of both algorithms (ZDT4)

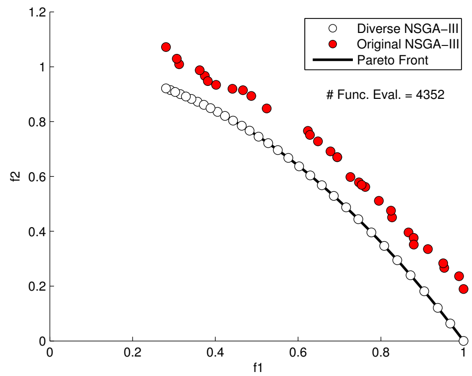

Figure 17: Median fronts of both algorithms (ZDT6)

4.1 Bi-objective results

In addition to the standard ZDT problem suite, we also include the recently proposed Variable-Density problem (*VarDens*) [*1*]. In VarDens the Pareto front in divided into two disconnected section. The first is easy to reach as it lies on the dense side of the objective space, while the other is very difficult to attain, because it lies on the sparse side. Figures 10,11,12,13,14,15 clearly shows the superiority of the proposed approach over the original NSGA-III. And although our approach is obviously much faster in reaching a close-to-Zero IGD, the advantages of using Div-NSGA-III surpass this speedup factor. Actually, in some problems, using a small population size is completely prohibitive. It can prevent the algorithm form converging no matter how long will the algorithm continue to try. A good example of this situation is ZDT4. In this problem, NSGA-III with a population of 48 individuals never converges to the true Pareto front. Our simulation results showed that getting NSGA-III (original) trapped in a local Pareto front using this relatively small population size is unavoidable. A recent study [*1*] used local search to resolve this issue with ZDT4, however their resolution came at the expense of diversity. Figure 16 shows how our approach on the other hand captures both diversity and convergence using only 48 individuals. A similar situation can be observed even with easy problems. For example, using a population of 4 individuals to solve ZDT1 is not possible. Only four individuals cannot generate sufficient offspring to maintain the desired diversity. On the other hand, using our approach automatically decomposes the problem into two BWS based optimizations and two ASF based optimizations, the four of which are easy single objective problems due to the simplicity of the original problem (ZDT1). And, in just four simple single-objective optimizations, the four individuals end evenly distributed on the true Pareto front (figure omitted for brevity). At last, the speedup factor remains an advantage that cannot be underestimated as shown in Figure 17 for ZDT6.

Constrained bi-objective results are shown in Figures 18,19,20,21. In easy problems, the difference is minimal. In difficult problems however, our approach clearly outperforms NSGA-III, most notably in OSY, which represents another example of the situation discussed above, see Figure 22

4.2 Multi/Many-objective results

Finally, we discuss the performance of our approach versus NSGA-III in both multi-objective (3 obj.) and many-objective (10 obj.) optimization problems. Figures 23 and 24 shows the superiority of our approach in both versions of the problem. We also conducted the same experiment on DTLZ1, however the two algorithms behaved almost identically (figures omitted for brevity).

5 Conclusion

In this study, we propose an enhanced version of the recently proposed NSGA-III. Our proposed approach focuses on diversity without sacrificing convergence, through

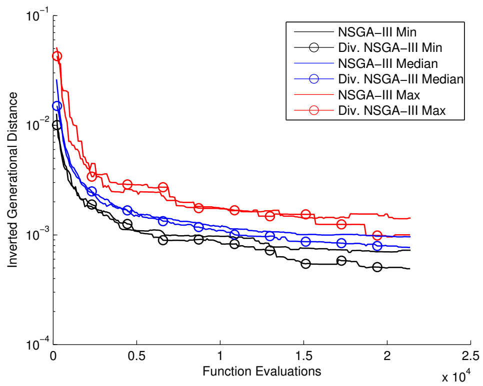

Figure 18: IGD vs. number of function evaluations (TNK)

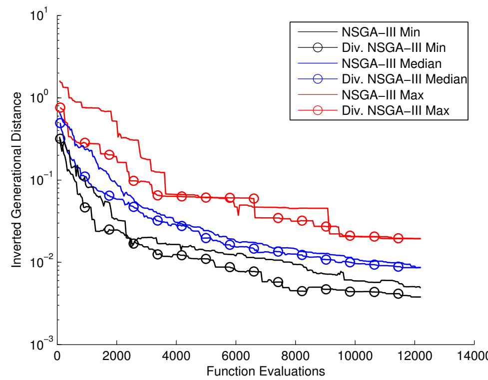

Figure 19: IGD vs. number of function evaluations (BNH)

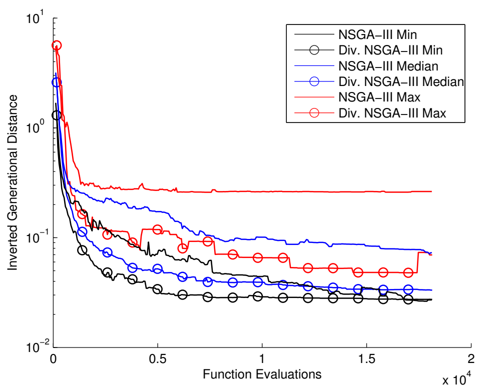

Figure 20: IGD vs. number of function evaluations (SRN)

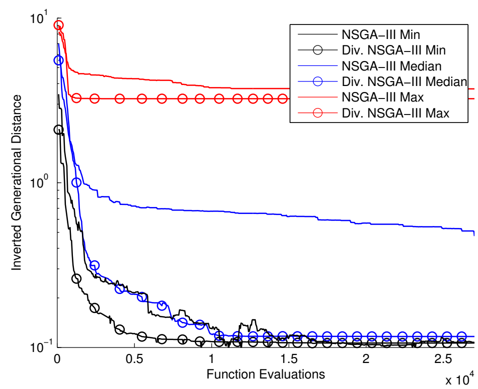

Figure 21: IGD vs. number of function evaluations (OSY)

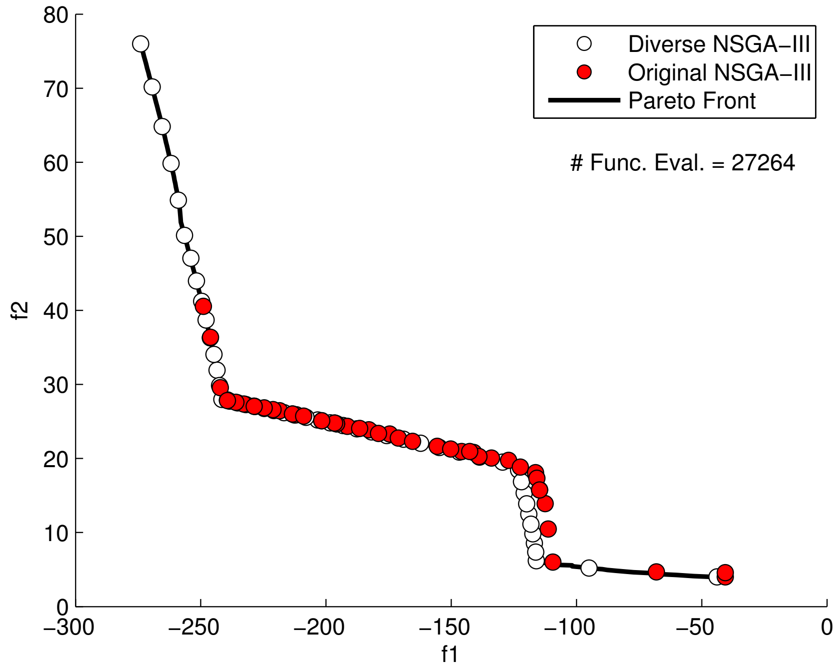

Figure 22: Median fronts of both algorithms (OSY)

a carefully designed niching operator, where NSGA-III is augmented with two local search mechanisms. The first mechanism aims at stretching the front as mich as possible, while the second works on filling internal gaps. The two mechanisms keep alternating until the desired diversity is attained. Our simulation results clearly show the superiority of our approach compared to the the original NSGA-III over a wide range of problems. This study provides a successful combination where each technique benefits from the merits of the other through a seamless exchange of information throughout the whole optimization process. Possible extensions of this work include self-adaptive reference directions and dynamic balancing between convergence and diversity.

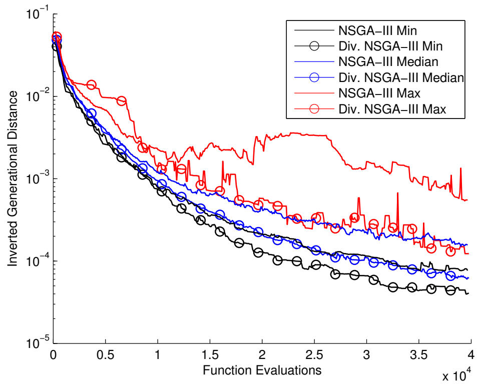

Figure 23: IGD vs. number of function evaluations (DTLZ2(3))

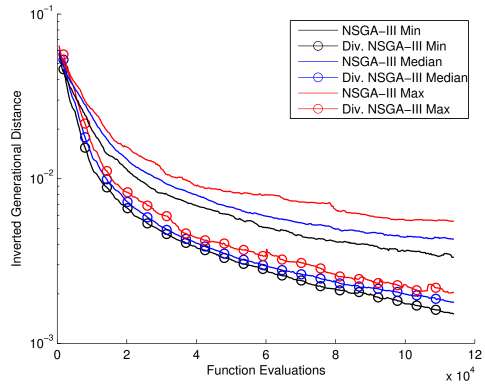

Figure 24: IGD vs. number of function evaluations (DTLZ2(10))

References

[1] M. Abouhawwash, H. Seada, and K. Deb. Karush-kuhn-tucker optimality based local search for enhanced convergence of evolutionary multi-criterion optimization methods. 2015.

[2] P. A. Bosman. On gradients and hybrid evolutionary algorithms for real-valued multiobjective optimization. *Evolutionary Computation, IEEE Transactions on*, 16(1):51–69, 2012.

[3] K. Deb. *Multi-objective optimization using evolutionary algorithms*, volume 16. John Wiley & Sons, 2001.

[4] K. Deb, S. Agrawal, A. Pratap, and T. Meyarivan. A fast elitist non-dominated sorting genetic algorithm for multi-objective optimization: Nsga-ii. *Lecture notes in computer science*, 1917:849–858, 2000.

[5] K. Deb and D. E. Goldberg. An investigation of niche and species formation in genetic function optimization. In *Proceedings of the Third International Conference*

on Genetic Algorithms, pages 42–50, 1989.

[6] K. Deb and H. Jain. An evolutionary many-objective optimization algorithm using reference-point-based nondominated sorting approach, part i: solving problems with box constraints. *Evolutionary Computation, IEEE Transactions on*, 18(4):577–601, 2014.

[7] K. A. DeJong. *An Analysis of the Behavior of a Class of Genetic Adaptive Systems*. PhD thesis, Ann Arbor, MI: University of Michigan, 1975. Dissertation Abstracts International 36(10), 5140B (University Microfilms No. 76-9381).

[8] H. Ishibuchi and T. Murata. A multi-objective genetic local search algorithm and its application to flowshop scheduling. *Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions on*, 28(3):392–403, 1998.

[9] H. Jain and K. Deb. An evolutionary many-objective optimization algorithm using reference-point based nondominated sorting approach, part ii: handling constraints and extending to an adaptive approach. *Evolutionary Computation, IEEE Transactions on*, 18(4):602–622, 2014.

[10] V. Khare, X. Yao, and K. Deb. Performance scaling of multi-objective evolutionary algorithms. In *Evolutionary Multi-Criterion Optimization*, pages 376–390. Springer, 2003.

[11] J. D. Knowles and D. W. Corne. M-paes: A memetic algorithm for multiobjective optimization. In *Evolutionary Computation, 2000. Proceedings of the 2000 Congress on*, volume 1, pages 325–332. IEEE, 2000.

[12] D. Saxena and K. Deb. Trading on infeasibility by exploiting constraint's criticality through multi-objectivization: A system design perspective. In *Proceedings of the*

Congress on Evolutionary Computation (CEC-2007), pages 919–926. Piscatway, NJ: IEEE Press, 2007.

[13] K. Sindhya, K. Deb, and K. Miettinen. A local search based evolutionary multi-objective optimization approach for fast and accurate convergence. In *Parallel Problem Solving from Nature–PPSN X*, pages 815–824. Springer, 2008.

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

[15] E. Zitzler, M. Laumanns, L. Thiele, E. Zitzler, E. Zitzler, L. Thiele, and L. Thiele. Spea2: Improving the strength pareto evolutionary algorithm, 2001.