Surrogate Many Objective Optimization: Combining Evolutionary Search, ϵ-Dominance and Connected Restarts

Taimoor Akhtar[1(✉)], Christine A. Shoemaker[2,3], and Wenyu Wang[2]

[1] Environmental Research Institute, National University of Singapore, Singapore, Singapore
erita@nus.edu.sg

[2] Department of Industrial Systems Engineering and Management, National University of Singapore, Singapore, Singapore
wenyu_wang@u.nus.edu

[3] Department of Civil and Environmental Engineering, National University of Singapore, Singapore, Singapore

Abstract. Scaling multi-objective optimization (MOO) algorithms to handle many objectives is a significant computational challenge. This challenge exacerbates when the underlying objectives are computationally expensive, and solutions are desired within a limited number of expensive objective evaluations. A surrogate model-based optimization framework can be effective in MOO. However, most prior model-based algorithms are effective for 2–3 objectives. This study investigates the combined use of ϵ-dominance, connected restarts and evolutionary search for efficient Many-objective optimization (MaOO). We built upon an existing surrogate-based evolutionary algorithm, GOMORS, and propose ϵ-GOMORS, i.e., a surrogate-based iterative evolutionary algorithm that combines Radial Basis Functions and ϵ-dominance-based evolutionary search, to propose new points for expensive evaluations in each algorithm iteration. Moreover, a novel connected restart mechanism is introduced to ensure that the optimization search does not get stuck in locally optimum fronts. ϵ-GOMORS is applied to a few benchmark multi-objective problems and a watershed calibration problem, and compared against GOMORS, ParEGO, NSGA-III, Borg, ϵ-NSGA-II and MOEA/D on a limited budget of 1000 evaluations. Results indicate that ϵ-GOMORS converges more quickly than other algorithms and the variance of its performance across multiple trials, is also less than other algorithms.

Keywords: Expensive optimization · Many objectives · Meta-models

This work was partially supported by the Singapore National Research Foundation, Prime Minister's Office, Singapore under its Campus for Research Excellence and Technological Enterprise (CREATE) programme (E2S2-CREATE project CS-B) and by Prof. Shoemaker's NUS startup grant.

H. A. Le Thi et al. (Eds.): WCGO 2019, AISC 991, pp. 681–690, 2020.
https://doi.org/10.1007/978-3-030-21803-4_68

1 Introduction

Many real-world optimization problems are multi-objective, where evaluation of objectives is computationally expensive. Multi-Objective Evolutionary Algorithms (MOEAs) are extremely popular for solving computationally expensive multi-objective problems, since their population-based structure allows MOEAs to converge to the Pareto front and simultaneously find diverse trade-off solutions [5].

Despite their inherent capability of simultaneously pursuing convergence and diversity, MOEAs may still require many expensive simulations to find suitable trade-off solutions, especially for Many-objective Optimization (MaOO) problems [1,2]. Iterative use of surrogate models in optimization can significantly reduce computational effort for expensive MO problems.

The taxonomy of iterative surrogate multi-objective optimization is discussed in [10]. Many iterative surrogate algorithms have been proposed in past literature for expensive multi-objective optimization, and are dominated by methods that either use Gaussian Processes (GP) [6,7,11] or Radial Basis Functions (RBFs) [1,14] as surrogates. However, most surrogate methods introduced in the past are only designed for and tested on problems with up to 3 objectives.

Since many real world optimization application can have many objectives (more than three), this study proposes ϵ-GOMORS, an extension of the GOMORS algorithm [1], that is designed to handle many objectives. GOMORS is an iterative surrogate MO algorithm that uses RBFs as surrogates and has performs better than the GP-based ParEGO [11] on a limited evaluation budget and especially on problems with more than 10 decision variables.

ϵ-GOMORS replaces the use of non-dominance archiving in GOMORS by ϵ-non-dominance archiving introduced in [13]. This ϵ-non-dominance archiving mechanism is a computationally efficient alternative to non-dominance archiving and has been used in some non-surrogate MOEAs to improve algorithm run-time efficiency and scale performance on many-objective problems [9,12].

An additional challenge associated with Multi-objective algorithms is that they can get stuck in locally optimum solutions and fronts, especially for problems with multi-modal objectives. Restart mechanisms have been used to alleviate this challenge in the past, especially in MOEAs [9,12] and single objective surrogate algorithms [15,16]. ϵ-GOMORS also incorporates a novel restart mechanism to ensure that the algorithm does not get stuck in locally optimum fronts.

2 The ϵ-GOMORS Algorithm

2.1 The Iterative Surrogate Optimization Framework

The general framework of iterative Multi-Objective optimization with surrogates, as defined in [10], has three core components within the iterative loop (i.e., after algorithm initialization), namely, (i) methodology for fitting surrogate model(s), (ii) generating candidate solutions using surrogates and (iii) selecting points for expensive evaluations from candidate solutions.

2.2 The GOMORS Algorithm

The ϵ-GOMORS algorithm introduced in this study is an extension of the surrogate multi-objective algorithm GOMORS [1]. GOMORS follows the iterative surrogate framework discussed in Sect. 2.1, and uses Radial Basis Functions (RBFs) as surrogates. One RBF surrogate is fitted for each expensive objective. Hence, assuming that $F(x) = [f_1(x), \ldots, f_k(k)]$ is the set of k expensive objectives being solved in our MOO, $\widehat{F}_m(x) = [\widehat{f}_{m,1}, \ldots, \widehat{f}_{m,k}]$ is the set of inexpensive surrogate functions fitted on the m points expensively evaluated so far.

During the iterative loop of GOMORS, two auxiliary problems are solved, i.e., the **Global surrogate problem** (defined in Eq. 1) and the **Gap optimization problem** (defined in Eq. 2. Equations 1 and 2 define the two auxiliary problems, where x_L and x_U are the lower and upper bounds of the original MOO problem being solved, and x^{crowd} is the least crowded (as per crowding distance [5]) evaluated solution amongst the non-dominated evaluated solutions. r is a vector that defines the neighborhood of x^{crowd}. Hence, the **Gap optimization problem** is a search in the neighborhood of x^{crowd}.

$$\begin{aligned} &\text{minimize: } \widehat{F}_m(x) = [\widehat{f}_{m,1}(x), \ldots, \widehat{f}_{m,k}(x)]^T \\ &\text{subject to } x_L \le x \le x_U \end{aligned} \tag{1}$$

$$\begin{aligned} &\text{minimize: } \widehat{F}_m(x) = [\widehat{f}_{m,1}(x), \ldots, \widehat{f}_{m,k}(x)] \\ &\text{subject to: } (x^{crowd} - r) \le x \le (x^{crowd} + r) \end{aligned} \tag{2}$$

NSGA-II is used in GOMORS, as the embedded algorithm for solving the auxiliary problems of Eqs. 1 and 2. Moreover, for solving the **Global surrogate problem** of Eq. 1, the non-dominated archive (of expensively evaluated points) is *injected* into the initial population of NSGA-II. Two candidate populations are consequently generated. Let P_A be the final population generated after solving the **Global surrogate problem** and let P_B be the final population generated after solving the **Gap optimization problem**. GOMORS then uses multiple rules [1] to select multiple new points, from P_A and P_B, for expensive evaluations in each algorithm iteration.

2.3 The ϵ-GOMORS Framework

Figure 1 provides an overview of the iterative framework of ϵ-GOMORS. As is depicted in Fig. 1 the algorithmic framework of ϵ-GOMORS is very similar to GOMORS. Each iterative loop of ϵ-GOMORS starts with fitting of RBF surrogates (on each objective). The algorithm continues by independently solving the two auxiliary problems discussed in Sect. 2.2. Multiple points are then selected for expensive evaluations from the two populations, i.e., P_A and P_B.

2.4 ϵ-Non-Dominance Archiving and ϵ-NSGA-II

In order to handle problems with many objectives, ϵ-GOMORS maintains an **ϵ-non-dominance archive** (introduced by [13]) instead of a **non-dominance**

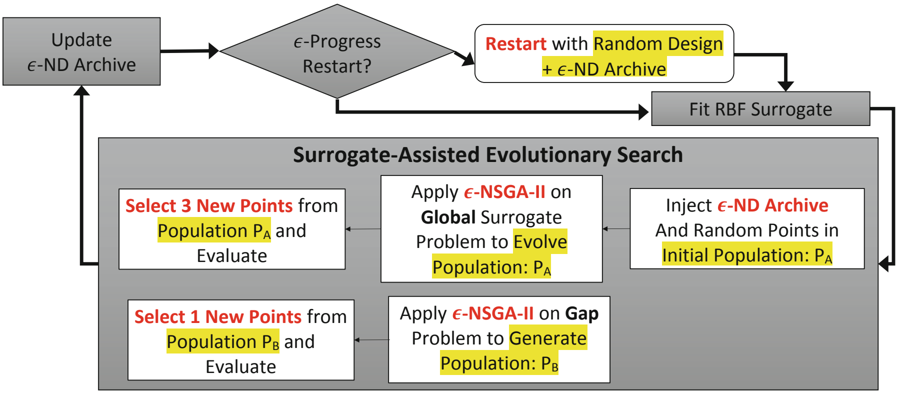

Fig. 1. General algorithm framework of ϵ-GOMORS.

archive (maintained in GOMORS). Let $F(x) = [f_1(x), \ldots, f_k(k)]$ be the k expensive objectives being solved in our MOO, where $x \in \mathcal{D}$:

Definition 1. *An objective vector* $y = [y_1, \ldots, y_k]$ ***dominates*** *(i.e.,* $y \prec z$*) another vector* $z = [z_1, \ldots, z_k]$ *if and only if* $y_i \leq z_i$ *for all* $1 \leq i \leq k$*, and* $y_j < z_j$ *for some* $1 \leq j \leq k$*.*

Definition 2. *Given a set of solutions* $S = \{x \mid x \in \mathcal{D}\}$ *, a subset (archive) of solutions* $S^\dagger \subset S$ *is* ***non-dominated*** *in* S *if there does not exist a solution* $x \in S$ *which dominates* $x^\dagger \in S^\dagger$*, i.e,* $S^\dagger = \{x^\dagger \in S \mid \nexists x \in S, F(x) \prec F(x^\dagger)\}$*.*

Definition 3. *Given an* $\epsilon > 0$*, an objective vector* $y = [y_1, \ldots, y_k]$ ***ϵ-box dominates*** *(i.e.,* $y \prec_\epsilon z$*) another vector* $z = [z_1, \ldots, z_k]$ *if and only if (i)* $\lfloor \frac{y}{\epsilon} \rfloor \prec \lfloor \frac{z}{\epsilon} \rfloor$ *OR (ii)* $\lfloor \frac{y}{\epsilon} \rfloor = \lfloor \frac{z}{\epsilon} \rfloor$ *and* $||y - \epsilon \lfloor \frac{y}{\epsilon} \rfloor|| < ||z - \epsilon \lfloor \frac{z}{\epsilon} \rfloor||$*.* $\lfloor . \rfloor$ *is the floor function.*

Definition 4. *Given a set of evaluated solutions* $S = \{x \mid x \in \mathcal{D}\}$*, a subset (archive)* $S^* \subset S$ *is* ***ϵ-non-dominated*** *in* S *if there does not exist a solution* $x \in S$ *which* ϵ*-box dominates* $x^* \in S^*$*, i.e,* $S^* = \{x^* \in S \mid \nexists x \in S, F(x) \prec_\epsilon F(x^*)\}$*.*

The ϵ-box dominance concept used in ϵ-non-dominance archiving (see Definitions 3 and 4) essentially divides the objective space into hyperboxes with box dimensions defined by the vector $\epsilon = [\epsilon_1, \ldots, \epsilon_k]$. Each objective solution, y, resides in a hyperbox, where the lower left corner of that box (also called box-value), denoted by $\lfloor \frac{y}{\epsilon} \rfloor = [\lfloor \frac{y_1}{\epsilon_1} \rfloor, \ldots, \lfloor \frac{y_k}{\epsilon_k} \rfloor]$, represents the objective value of that box. $\lfloor a \rfloor$ is the greatest integer less than or equal to a, i.e., $\lfloor . \rfloor$ is the floor function. Consequently, an objective vector y ϵ-box dominates vector z if (i) the box-value of y dominates the box-value of z or (ii) if both y and z are in the same box (i.e., have the same box-value) but y is closer to the lower left corner of the box, than z. The vector, $\epsilon = [\epsilon_1, \ldots, \epsilon_k]$, is user-defined.

A core advantage of ϵ-non-dominance archiving is that it is computationally efficient for many objectives and hence, is more feasible than non-dominance archiving [9,12]. If the expensive optimization problem has many objectives, the auxiliary problems of Eqs. 1 and 2 will also have many objectives. Hence, solving the auxiliary problems requires an algorithm that is suitable for many-objective problems. ϵ-GOMORS thus uses ϵ-NSGA-II [12] as the auxiliary solver instead of NSGA-II (see Fig. 1). Moreover, for the Gap problem of Eq. 2, a solution is randomly selected from the ϵ-non-dominance archive, as x^{crowd}.

2.5 Connected Restarts

The ϵ-GOMORS algorithm also incorporates a novel restart mechanism (see Fig. 1). Restarts have been widely used in optimization algorithms in the past, to re-initialize the optimization search if it stagnates (e.g., if the algorithm gets stuck in a local optima) [9,12,15].

The restart mechanism of ϵ-GOMORS has two levels. The first restart level is triggered if the ϵ-non-dominance archive does not "improve" for a few algorithm iterations. Improvement is assessed via the ϵ-progress metric introduced in [9]. At the first restart level, the algorithm restarts with a new Symmetric Latin Hypercube (LHS) design plus the ϵ-non-dominant solutions from the previous start. The first restart level is called 'connected restart', and the purpose of this restart is to simultaneously inject new random solutions into the search (exploration), and retain the best solutions found so far (exploitation and elitism). After a few 'connected restarts' are registered, the second restart level is invoked. At this level, the algorithm restarts with only a new random initial design (Symmetric LHS). Hence, this restart level is called 'independent restart'.

3 Experiments and Results

3.1 Experimental Setup

Test Problems ϵ-GOMORS is designed to be efficient for many-objective problems. Hence, we test its performance on two widely used test problems, namely DTLZ2 and DTLZ4, that are scalable in the number of objectives, and were proposed in [4]. An additional challenge associated with DTLZ4 is that it has a non-uniform distribution of points on the Pareto front. We compare performance of ϵ-GOMORS and other algorithms designed for many-objective optimization, on DTLZ2 and DTLZ4 with 2, 4 and 6 objectives. The number of decision variables for both test problems are set to $n_{obj} + 9$ (as per the recommendations given in [4]), where n_{obj} is the number of objectives.

Cannonsville Watershed Calibration Problem We also test ϵ-GOMORS on Multi-Objective calibration of the SWAT Cannonsville watershed model developed by Tolson and Shoemaker [18]. We calibrate 15 hydrologic parameters of the Cannonsville watershed model in this study by formulating the calibration as a bi-objective global optimization problem (unconstrained). The model is calibrated on a 10 year historical flow time-series data set (obtained from United

States Geological Survey (USGS) Station 01421618), and the two calibration objectives represent different errors between simulated and observed data. Running time of a 10-year simulation of the Cannonsville model is around 1 min. The watershed calibration problem is called 'CFLOW' in subsequent discussions.

Alternate Algorithms Performance of ϵ-GOMORS is compared against numerous surrogate-based and non-surrogate (mostly evolutionary) algorithms. For the scalable DTLZ problems [4], performance of ϵ-GOMORS is compared against two MOEAs designed for many-objective optimization, ϵ-NSGA-II [12] and NSGA-III [3]. For the watershed calibration problem we compare ϵ-GOMORS with the surrogate algorithms ParEGO [11] and GOMORS [1] and the evolutionary algorithms MOEA/D [19] and Borg [9] (these algorithms have been applied to water resources problems in the past).

Performance Assessment Methods This study focuses on multi-objective optimization of computationally expensive functions. Hence we limit all optimization experiments to a budget of 1000 function evaluations. Moreover, since all algorithms compared in this study are stochastic, multiple optimization trials (10 trials) are run for each algorithm on each test problem. Hypervolume coverage, H_c, is used as the metric for assessing multi-objective performance of an algorithm. The Hypervolume coverage is defined as follows:

$$H_c(P) = \frac{H_v(P) - H_v(P_{init})}{H_v(P^*) - H_v(P_{init})} \tag{3}$$

Let P be the set of non-dominated solutions obtained by an algorithm and let P^* be the Pareto front of the multi-objective problem being solved. Moreover, let $H_v(A)$ be the Hypervolume [6] of the objective space dominated by an arbitrary set A. Consequently, $H_c(P)$ is the proportion of total feasible objective space (after subtracting the space dominated by initial solutions, i.e., P_{init}) dominated by P. Higher values of H_c are desirable and ideal value is 1.

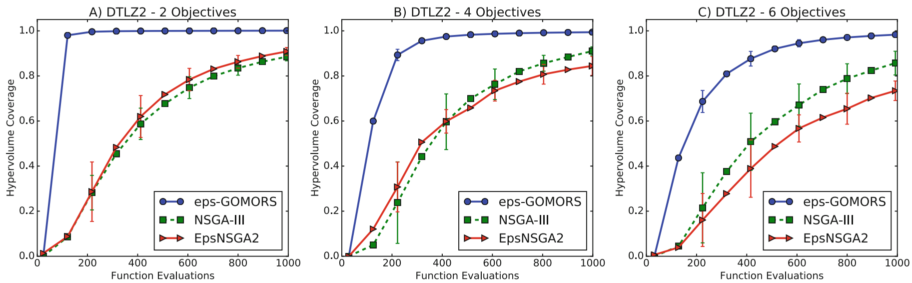

Fig. 2. ***DTLZ2 Progress Plots***: Hypervolume coverage progress curves (averaged over multiple trials) of all algorithms for DTLZ2 [4], with 2, 4 and 6 objectives. Each subplot corresponds to a Hypervolume progress plot (higher values are better) comparison for a fixed number of objectives (depicted in subplot title).

3.2 Results

Progress Curves - Many Objective Test Problems Results for the two scalable test problems, DTLZ2 and DTLZ4 [4] are compared by plotting the Hypervolume coverage (H_c) obtained by an algorithm against the number of completed function evaluations. We call these plots *progress curves* in subsequent discussions. Figures 2 and 3 illustrate the *progress curves* for DTLZ2 and DTLZ4, respectively. ϵ-GOMORS is labeled as "eps-GOMORS" and ϵ-NSGA-II is labeled as "EpsNSGA2" in Figs. 2 and 3.

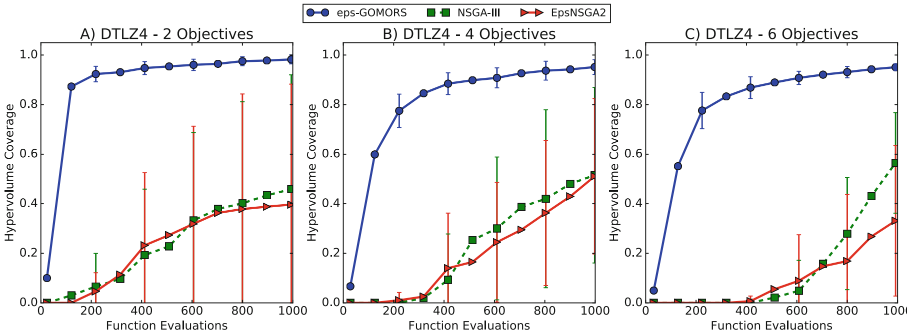

Fig. 3. ***DTLZ4 Progress Plots***: Hypervolume coverage progress curves (averaged over multiple trials) of all algorithms for DTLZ4 [4], with 2, 4 and 6 objectives. Each subplot corresponds to a Hypervolume progress plot (higher values are better) comparison for a fixed number of objectives (depicted in subplot title).

Figure 2 compares the *progress curves* of ϵ-GOMORS, ϵ-NSGA-II and NSGA-III for the DTLZ2 test problem with 2, 4 and 6 objectives (sub-figures A, B and C, respectively), and with a budget of 1000 function evaluations each. Figure 2 clearly indicates that ϵ-GOMORS is the fastest converging of all three algorithms, since the curves for ϵ-GOMORS are highest for all three DTLZ2 variants.

Results for the DTLZ4 test case (see Fig. 3) are similar, and performance of ϵ-GOMORS is significantly better than the other algorithms after 1000 function evaluations. This is true for all DTLZ4 variants, i.e., the 2-objective, the 4-objective and the 6-objective case. Overall, results of both test problems indicate that, on a limited function evaluations budget, performance of ϵ-GOMORS is better than ϵ-NSGA-II and NSGA-III, for 2, 4 and 6 objectives. Hence, our results indicate that ϵ-GOMORS is effective for multi-objective optimization (and also for problems with many objectives), when function evaluations are expensive and the evaluation budget is limited (less than 1000).

Results-Watershed Calibration Problem The Hypervolume Coverage (H_c) *progress curves* of ϵ-GOMORS, ParEGO, Borg and MOEA/D, for the watershed calibration problem, i.e., CFLOW (see Sect. 3.1 for problem definition), are illustrated in Fig. 4. Please note that since the Pareto front (P^* in Eq. 3) for

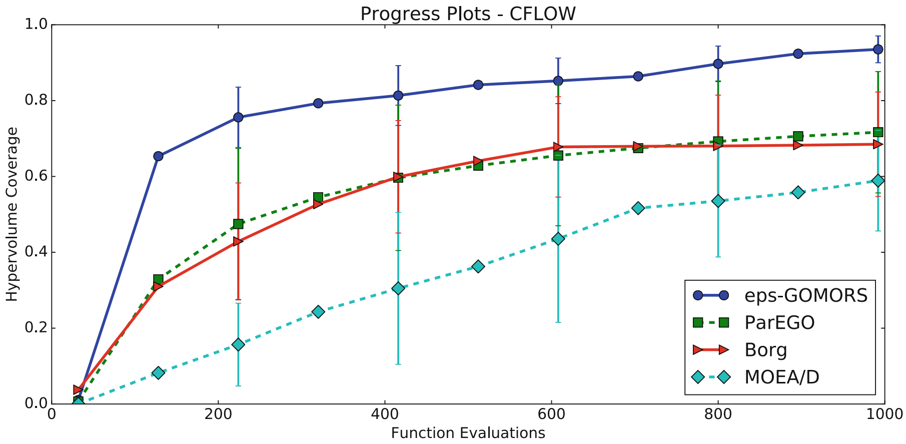

Fig. 4. ***Watershed Porblem Progress Plots***: Hypervolume coverage progress curves (high curves are better) of all algorithms (averaged over multiple trials) for the bi-objective Cannonsville Watershed calibration problem.

CFLOW is not known, it is estimated by consolidating the non-dominated solutions obtained from all optimization experiments (these include additional trials with more than 1000 evaluations). Figure 4 shows that ϵ-GOMORS is clearly the most efficient amongst all algorithms compared, for a budget of 1000 function evaluations. Moreover, efficiency of ϵ-GOMORS is such that ϵ-GOMORS, in less than 200 function evaluations, achieves the Hypervolume coverage attained by ParEGO (the next best algorithm) after 1000 evaluations. This essentially means that ϵ-GOMORS is **5 times faster** than ParEGO when evaluations are limited to 1000.

A key difference between GOMORS and ϵ-GOMORS is the connected restarts mechanism that has been introduced in ϵ-GOMORS. A core purpose of introducing connected restarts in ϵ-GOMORS is to ensure that the algorithm does not get stuck in locally optimum fronts. Figure 5 provides an illustration of the effect of ϵ-GOMORS' restart mechanism, by plotting the non-domination fronts of the best and worst solutions (according to Hypervolume coverage) obtained by ϵ-GOMORS across multiple trials (see Fig. 5-A). Figures 5-B and 5-C plot the best and worst non-dominated fronts of GOMORS and ParEGO, respectively.

Figure 5 illustrates that the difference between the best and worst non-dominated fronts (multiple trials) for ϵ-GOMORS (see Fig. 5A) is considerably less than the corresponding difference for GOMORS, for the CFLOW calibration problem. Since the objectives for the CFLOW problem are multi-modal [17], the better performance of ϵ-GOMORS (relative to GOMORS) across multiple optimization trials, may be attributed to the restart mechanism introduced in ϵ-GOMORS. Figure 5 also shows that performance of ϵ-GOMORS is significantly better than ParEGO (see Fig. 5-C) in terms of converging to the estimated Pareto front.

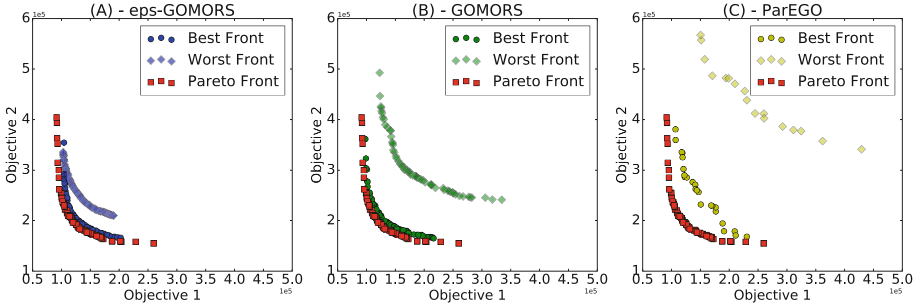

Fig. 5. ***Watershed Problem - Non-Dominated Fronts***: Visualizations of best and worst Non-Dominated (ND) fronts (lower fronts are better) obtained by (A) eps-GOMORS, (B) GOMORS and (C) ParEGO, for CFLOW watershed problem, after 1000 evaluations; compared against estimated true front.

4 Conclusion

ϵ-GOMORS is a novel extension of the surrogate MO algorithm GOMORS [1], that incorporates ϵ-dominance and connected restarts, to handle many-objective optimization problems. A restart mechanism is introduced in ϵ-GOMORS to ensure that the algorithm does not get stuck in locally optimum trade-off solutions.

Results of ϵ-GOMORS on two many-objective test problems are promising, indicating that the ϵ-dominance concept allows the algorithm to scale well for up to six objectives. Moreover, results on many-objective test problems also show that ϵ-GOMORS is more efficient than other state-of-the-art many-objective evolutionary (non-surrogate) algorithms, ϵ-NSGA-II and NSGA-III, when evaluation budget is limited to 1000.

When applied to a watershed calibration problem, ϵ-GOMORS is more reliable than GOMORS (i.e., the variance of ϵ-GOMORS' performance across multiple optimization trials is less), and considerably more efficient that other surrogate (e.g., the Gaussian Process-based ParEGO) and non-surrogate algorithms it is compared against. In future, we intend to test ϵ-GOMORS with different surrogate taxonomies [2] to further improve efficiency for surrogate many objective optimization. Python implementation of ϵ-GOMORS is available upon request, and will be made available online in future, as part of the pySOT toolbox [8].

References

1. Akhtar, T., Shoemaker, C.A.: Multi objective optimization of computationally expensive multi-modal functions with RBF surrogates and multi-rule selection. J. Global Optim. **64**(1), 17–32 (2016)
2. Deb, K., Hussein, R., Roy, P.C., Toscano, G.: A taxonomy for metamodeling frameworks for evolutionary multi-objective optimization. IEEE Trans. Evol. Comput. 1–1 (2018)

3. Deb, K., Jain, H.: An evolutionary many-objective optimization algorithm using reference-point-based nondominated sorting approach, part i: solving problems with box constraints. IEEE Trans. Evol. Comput. **18**(4), 577–601 (2014)
4. Deb, K., Thiele, L., Laumanns, M., Zitzler, E.: Scalable multi-objective optimization test problems. In: Proceedings of the 2002 Congress on Evolutionary Computation, CEC 2002 (Cat. No.02TH8600), vol. 1, pp. 825–830, May 2002
5. Deb, K., Kalyanmoy, D.: Multi-Objective Optimization Using Evolutionary Algorithms, 1 edn. Wiley (2001)
6. Emmerich, M.T.M., Deutz, A.H., Klinkenberg, J.W.: Hypervolume-based expected improvement: monotonicity properties and exact computation. In: 2011 IEEE Congress of Evolutionary Computation (CEC), pp. 2147–2154, June 2011
7. Emmerich, M., Yang, K., Deutz, A., Wang, H., Fonseca, C.M.: A Multicriteria Generalization of Bayesian Global Optimization, pp. 229–242. Springer International Publishing, Cham (2016)
8. Eriksson, D., Bindel, D., Shoemaker, C.: Surrogate optimization toolbox (pysot). https://github.com/dme65/pySOT (2015)
9. Hadka, D., Reed, P.: Borg: an auto-adaptive many-objective evolutionary computing framework. Evol. Comput. **21**(2), 231–259 (2013)
10. Horn, D., Wagner, T., Biermann, D., Weihs, C., Bischl, B.: Model-Based Multi-Objective Optimization: Taxonomy, Multi-point Proposal, Toolbox and Benchmark, pp. 64–78. Springer International Publishing, Cham (2015)
11. Knowles, J.: ParEGO: a hybrid algorithm with on-line landscape approximation for expensive multiobjective optimization problems. IEEE Trans. Evol. Comput. **8**(5), 1341–66 (2006)
12. Kollat, J.B., Reed, P.M.: A computational scaling analysis of multiobjective evolutionary algorithms in long-term groundwater monitoring applications. Adv. Water Resour. **30**(3), 335–353 (2007)
13. Laumanns, M., Thiele, L., Deb, K., Zitzler, E.: Combining convergence and diversity in evolutionary multiobjective optimization. Evol. Comput. **10**(3), 263–282 (2002)
14. Mueller, J.: Socemo: Surrogate optimization of computationally expensive multiobjective problems. INFORMS J. Comput. **29**(4), 581–596 (2017)
15. Regis, R.G., Shoemaker, C.A.: A stochastic radial basis function method for the global optimization of expensive functions. INFORMS J. Comput. **19**(4), 497–509 (2007)
16. Regis, R.G., Shoemaker, C.A.: Combining radial basis function surrogates dynamic coordinate search in high dimensional expensive black-box optimization. Eng. Optim. **45**(5), 529–555 (2013)
17. Shoemaker, C.A., Regis, R.G., Fleming, R.C.: Watershed calibration using multistart local optimization and evolutionary optimization with radial basis function approximation. Hydrol. Sci. J. **52**(3), 450–465 (2007)
18. Tolson, B., Shoemaker, C.: Cannonsville reservoir watershed SWAT2000 model development, calibration and validation. J. Hydrol. **337**, 68–86 (2007)
19. Zhang, Q., Li, H.: MOEA/D: a multiobjective evolutionary algorithm based on decomposition. IEEE Trans. Evol. Comput. **11**(6), 712–731 (2007)