Accelerating Large-scale Multiobjective Optimization via Problem Reformulation

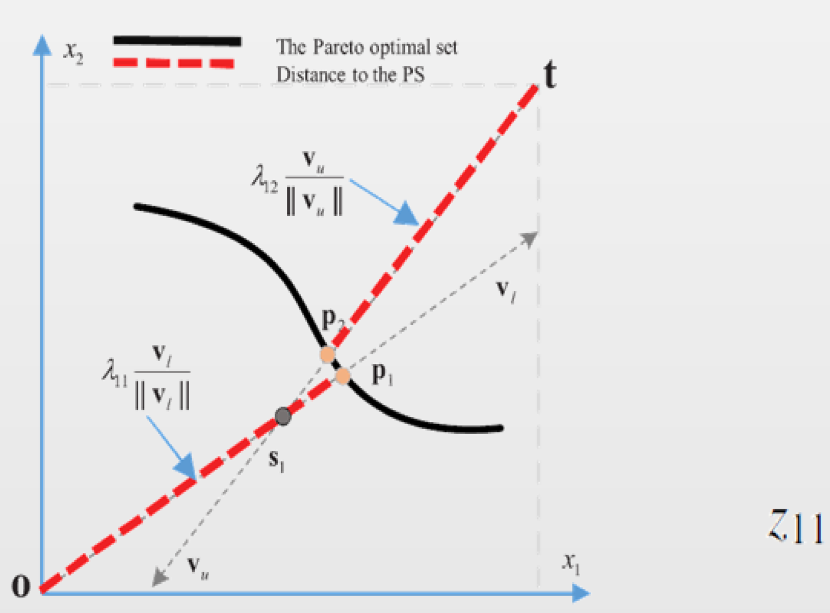
Cheng He[1], Lianghao Li[2], Ye Tian[3], Xingyi Zhang[4], Ran Cheng[1], Yaochu Jin[1] , Xin Yao[1]
Department of Computer Science and Engineering, Southern University of Science and Technology
School of Automation, Huazhong University of Science and Technology
Institute of Physical Science and Information Technology, Anhui University
School of Computer Science and Technology, Anhui University

Introduction

With the progress in industrial automation, the demands for black-box multiobjective optimization with large-scale decision variables rise rapidly. Though evolutionary algorithms (EAs) are applicable to black-box optimization, their random search principle also leads to the unbearable cost of function evaluations. This work proposes to reformulate the large-scale multiobjective optimization problem into extremely low-dimensional ones, e.g., from 1000 to 20, aiming at obtaining acceptable solutions using 1/20 costs of conventional EAs. Moreover, this idea is also applicable to all existing EAs, enhancing their capability in solving large-scale multiobjective optimization problems with fewer function evaluations and computation time. Thanks to the low dimensionality of the weight variables and reduced objective space, a set of quasi-optimal solutions can be obtained efficiently. Finally, a multiobjective evolutionary algorithm is used to spread the quasi-optimal solutions over the approximate Pareto optimal front evenly. Experiments have been conducted on a variety of large-scale multiobjective problems with up to 5000 decision variables. Four different types of representative algorithms are embedded into the proposed framework and compared with their original versions, respectively. Furthermore, the proposed framework has been compared with two state-of-the-art algorithms for large-scale multiobjective optimization. The experimental results have demonstrated the significant improvement benefited from the framework in terms of its performance and computational efficiency in large-scale multiobjective optimization.

Methodology

Bi-directional Vector Guided Search in the Decision Space



$$\mathbf{v}_l = \mathbf{s}_1 - \mathbf{o}$$

$$\mathbf{v}_u = \mathbf{t} - \mathbf{s}_1$$

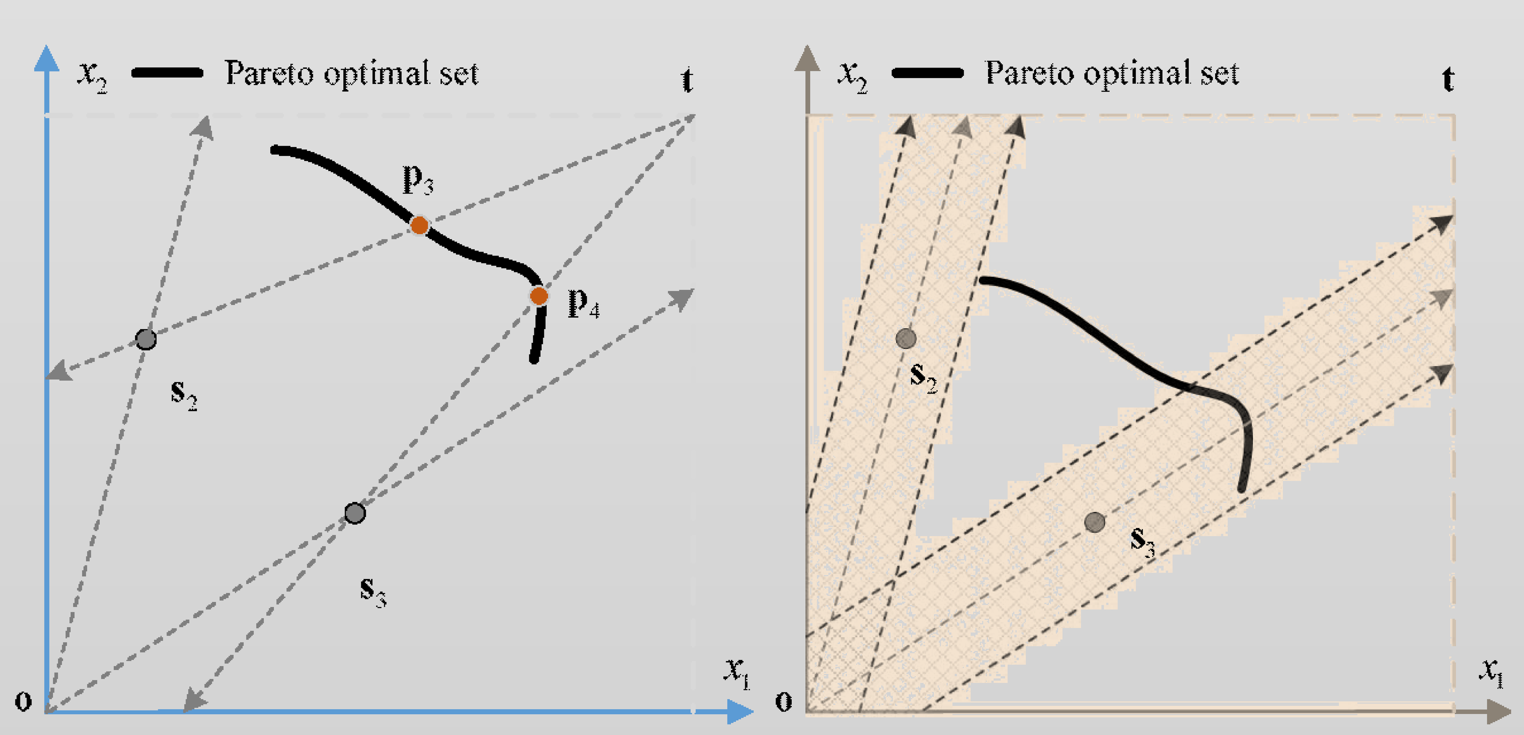
$$\mathbf{p}_1 = \mathbf{o} + \lambda_{11}\frac{\mathbf{v}_l}{||\mathbf{v}_l||} l_{\max}$$

$$\mathbf{p}_2 = \mathbf{t} - \lambda_{12}\frac{\mathbf{v}_u}{||\mathbf{v}_u||} l_{\max}$$

$$z_{11}(\lambda_{11}) = F\left(\mathbf{o} + \lambda_{11}\frac{\mathbf{v}_l}{||\mathbf{v}_l||} l_{\max}\right)$$

$$\begin{aligned}&\text{minimize} \quad F(\mathbf{x}) = (f_1(\mathbf{x}), f_2(\mathbf{x}), \ldots, f_M(\mathbf{x}))\\ &\text{subject to} \quad \mathbf{x} \in X\end{aligned} \Rightarrow \begin{aligned}&\text{maximize} \quad G(\Lambda) = H(Z'(\Lambda))\\ &\text{subject to} \quad \Lambda \in \Re^{2r}\end{aligned}$$

The core idea is to use some promising solutions to construct some search directions in the decision space, and then the large-scale multiobjective optimization problem is reformulated into a low-dimensional single-objective optimization problem. Specifically, the number of decision variables in the reformulated optimization is $2r$, where r is the number of selected reference solutions (usually set to 10). Thus, even an MOPs with 1000 decision variables can be simplified to a simple SOP (D=20).



Example illustrating the advantage of the proposed bi-directional weight variable association strategy in a 2-D decision space. s_2 and s_3 are the reference solutions, p_3 and p_4 are the intersections, and o, t are the lower and upper boundary points, respectively. Illustration of the (a) proposed bidirectional weight variable association strategy in a 2-D decision space and (b) unidirectional weight variable association strategy in a 2-D decision space.

Pseudo Code

Algorithm 1 Main Framework of the Proposed LSMOF

Input: Z (original LSMOP), FE_{max} (total FEs), Alg (embedded MOEA), N (population size for Alg), r (number of reference solutions), tr (threshold).
Output: P (final population).

$P \leftarrow$ Initialization(N, Z)
/************First Stage************/
while $t \leq tr \times FE_{max}$ **do**
  $Z' \leftarrow$ Problem_Reformulation(P, r, Z)
  $A, \Delta t \leftarrow$ Single_Objective_Optimization(Z')
  $P \leftarrow$ Environmental_Selection$(A \bigcup P, N)$
  $t \leftarrow t + \Delta t$
end while
/**********Second Stage**********/
$P \leftarrow$ Embedded_MOEA (P, N, Alg, Z)

Algorithm 2 Fitness Assignment Strategy in LSMOF

Input: Λ (weight vector), R (reference solution set), P (current population).
Output: $fit(\Lambda)$ (fitness value of Λ).

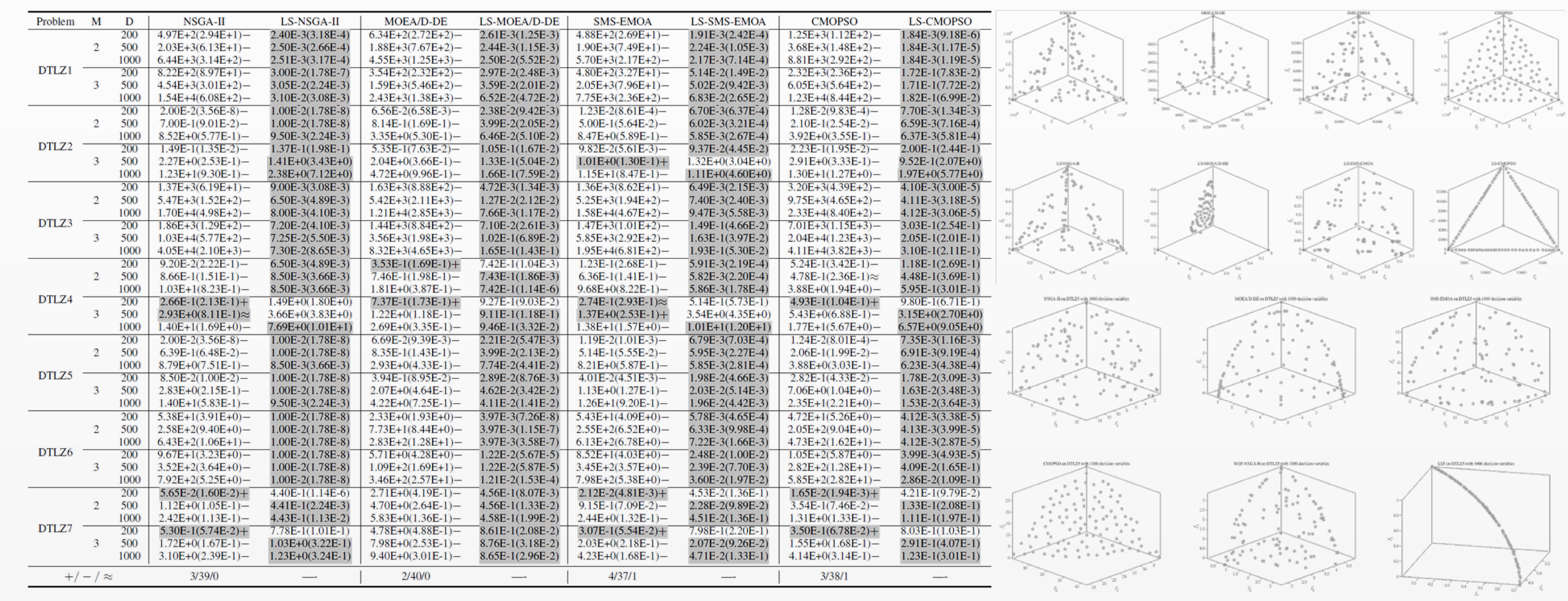
$Nad \leftarrow$ Calculate the nadir point of P in objective space
for $i \leftarrow 1 : r$ **do**
  $\mathbf{v}_l, \mathbf{v}_u \leftarrow$ Calculate direction vectors using (4)
  $\mathbf{p}_1, \mathbf{p}_2 \leftarrow$ Calculate the weight variable associated solutions using (5)
  $z_{i1}(\lambda_{i1}), z_{i2}(\lambda_{i2}) \leftarrow$ Calculate the objective vectors using (6)
end for
$Z'(\Lambda) \leftarrow \{z_{11}(\lambda_{11}), z_{12}(\lambda_{12}), \ldots, z_{r1}(\lambda_{r1}), z_{r2}(\lambda_{r2})\}$
$fit(\Lambda) \leftarrow$ Hypervolume$(Z'(\Lambda), Nad)$

Algorithm 3 Single-Objective Optimization in LSMOF

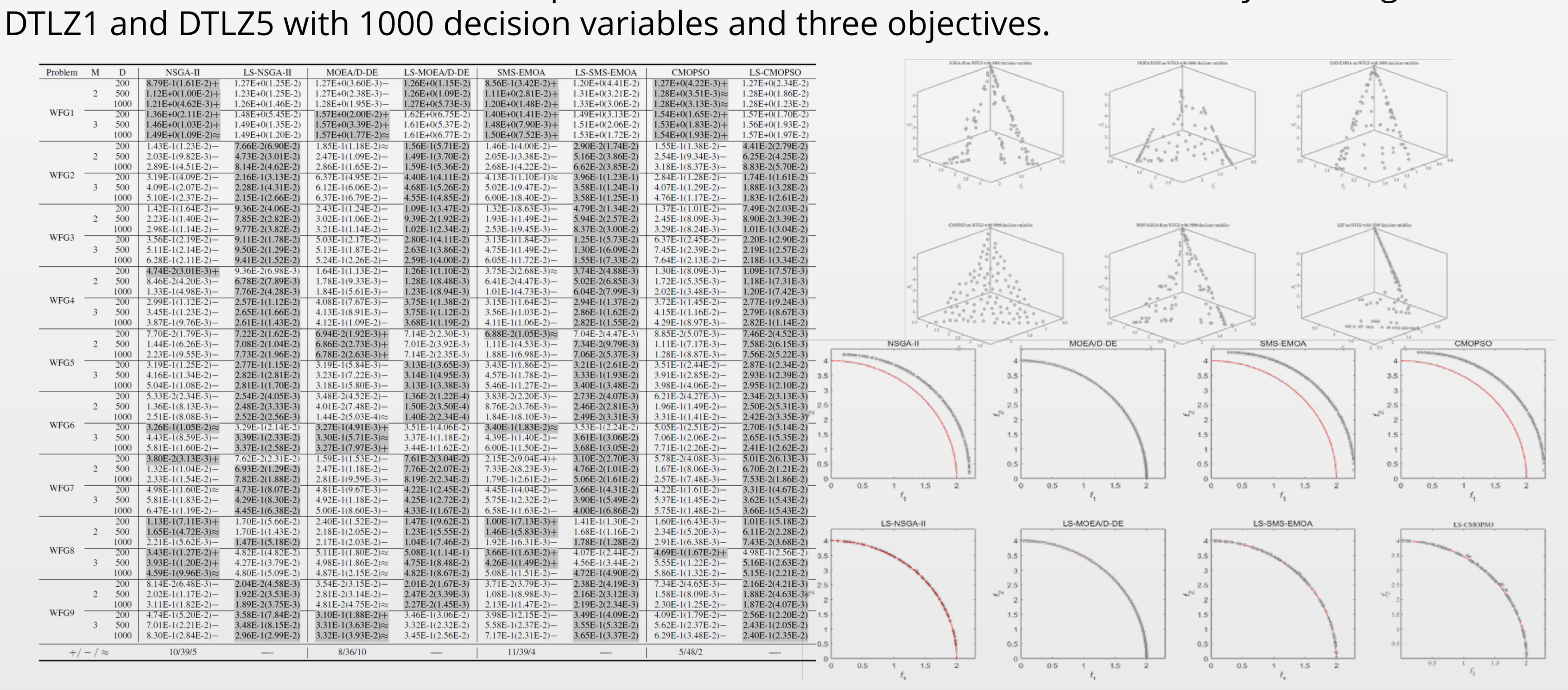
Input: NI (population size of DE), g (maximum number of iterations), CR (cross constant), F_m (scaling factor).
Output: A (population), Δt (number of FEs).

$\Delta t \leftarrow 0$
$P_\Lambda \leftarrow \{\Lambda_1, \ldots, \Lambda_{NI}\}$ /*Initialization*/
$fit(\Lambda_1), \ldots, fit(\Lambda_{NI}) \leftarrow$ Calculate the fitnesses of elements in P_Λ using Algorithm 2
$A \leftarrow$ Collet the generated candidate solutions during the fitness assignment
$\Delta t \leftarrow \Delta t + |A|$ /*$|A|$ denotes the element size of A*/
for $r \leftarrow 1 : g$ **do**
  for $i \leftarrow 1 : NI$ **do**
    $c_1, c_2, c_3 \leftarrow$ Randomly select three indices in $[1, NI]$
    $\mathbf{a} \leftarrow \Lambda_{c_1} + F_m(\Lambda_{c_2} - \Lambda_{c_3})$
    for $j \leftarrow 1 : |\Lambda_1|$ **do**
      if $rand_j[0, 1) \leq CR$ or $j = j_{rand}$ **then**
        $\mathbf{b}_i \leftarrow$ Choose the jth element of Λ_i
      else
        $\mathbf{b}_i \leftarrow$ Choose the jth element of $\mathbf{a}$
      end if
    end for
    $fit(\mathbf{b}) \leftarrow$ Calculate $\mathbf{b}$'s fitness using Algorithm 2
    $A' \leftarrow$ Collet the generated candidate solutions during the fitness assignment
    $\Delta t \leftarrow \Delta t + |A'|$
    if $fit(\mathbf{b}) \geq fit(\Lambda_i)$ **then**
      $\Lambda_i \leftarrow \mathbf{b}$
    end if
    $A \leftarrow A \cup A'$
  end for
end for

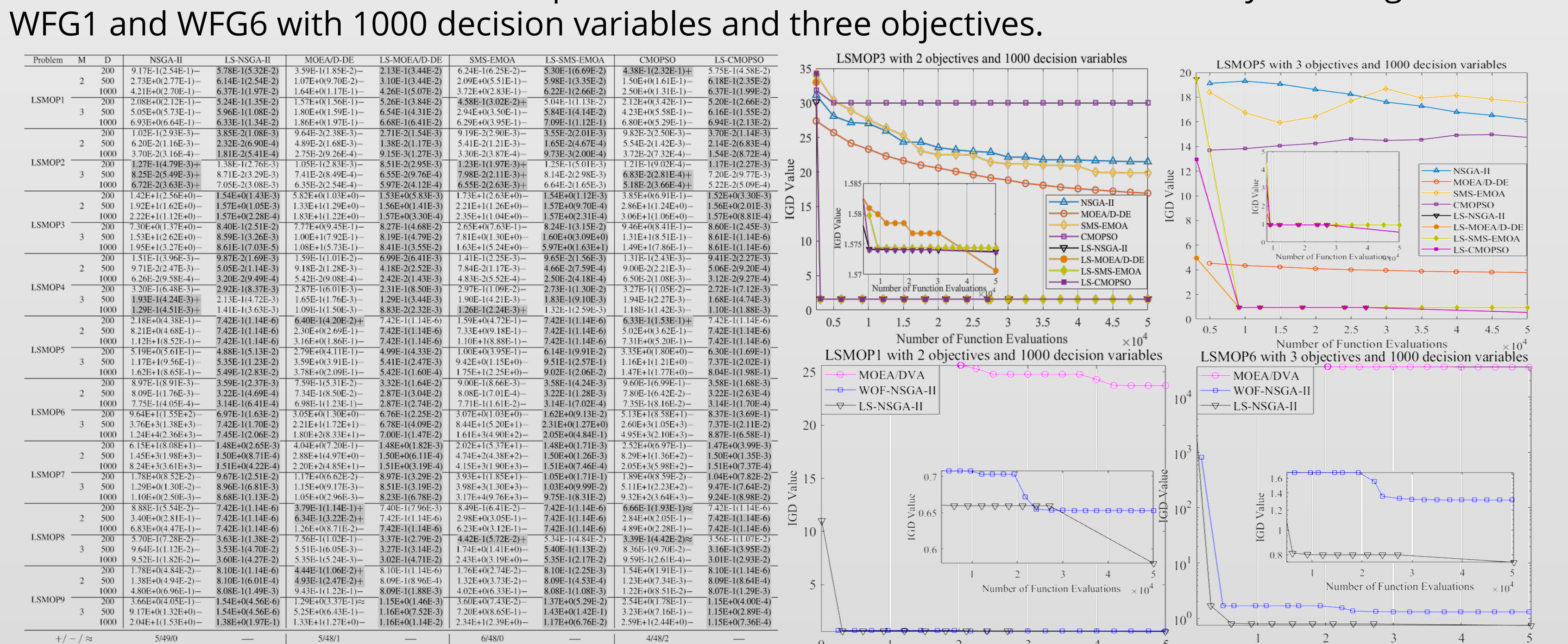
Results



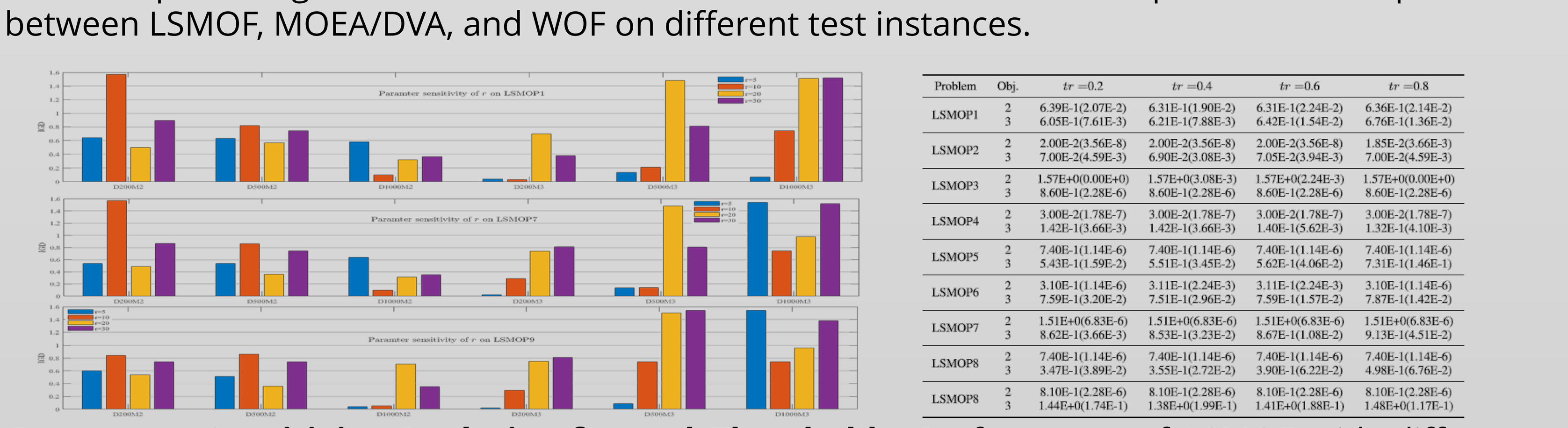
General Performance on DTLZ Problems: IGD results obtained by four compared algorithms and their modified version on 42 DTLZ problems; non-dominated fronts obtained by each algorithm on DTLZ1 and DTLZ5 with 1000 decision variables and three objectives.



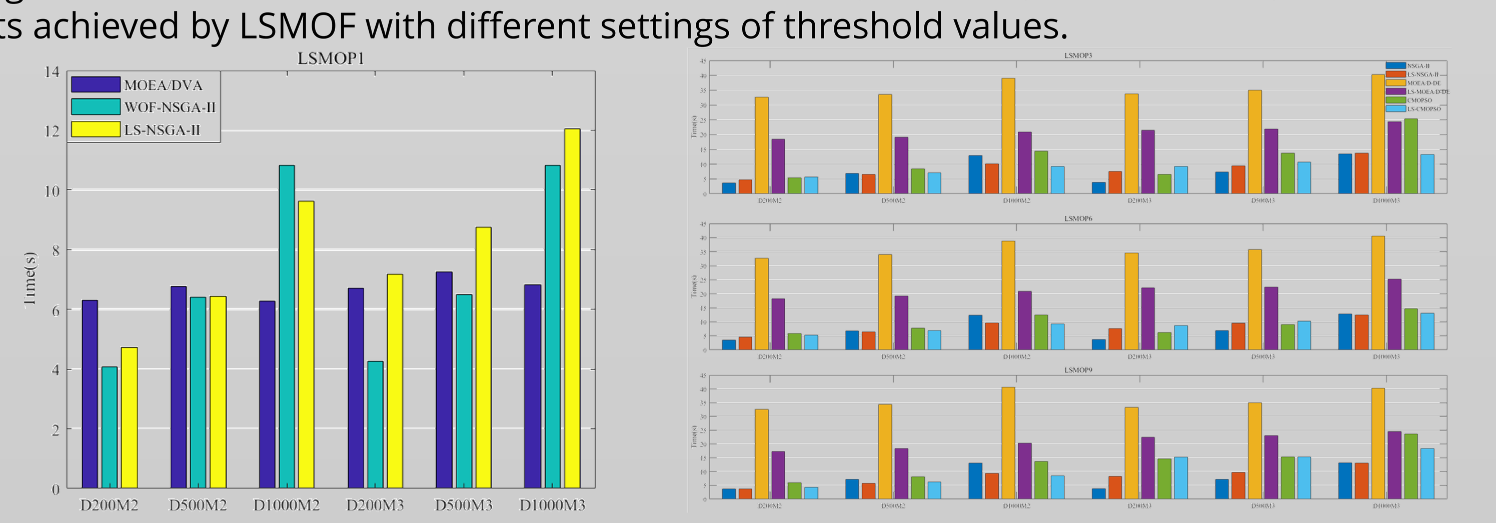
General Performance on WFG Problems: IGD results obtained by four compared algorithms and their modified version on 54 WFG problems; non-dominated fronts obtained by each algorithm on WFG1 and WFG6 with 1000 decision variables and three objectives.



General Performance on LSMOP Problems: IGD results and convergence profiles obtained by four compared algorithms and their modified version on 54 LSMOP problems; comparisons between LSMOF, MOEA/DVA, and WOF on different test instances.



Parameter Sensitivity Analysis of r and Threshold: Performance of LSMOF with different settings of number of reference solutions on LSMOP1, LSMOP7, and LSMOP9; statics of IGD results achieved by LSMOF with different settings of threshold values.



Computation Time Analysis: Average computation time of MOEA/DVA, WOF-NSGA-II, and LS-NSGA-II on LSMOP1, where M is the number of objectives and D is the number of decision variables; average computation time of NSGA-II, MOEA/D-DE, CMOPSO, and their LSMOF-based versions on LSMOP3, LSMOP6, and LSMOP9.

Conclusion

This work proposed a problem reformulation-based framework for reducing the number of decision variables to an extremely small level for saving the FEs. We try to use less than 10% of available FEs (e.g., less than 50,000) to achieve quasi-optimal results (e.g., IGD results around 10^{-1} level) , aiming to enhance the applicability and versatility of evolutionary algorithms. Source codes are available at https://www.chenghehust.com/assets/code/LSMOF_py.zip.

@article{he2019accelerating,
title={Accelerating large-scale multiobjective optimization via problem reformulation},
author={He, Cheng and Li, Lianghao and Tian, Ye and Zhang, Xingyi and Cheng, Ran and Jin, Yaochu and Yao, Xin},
journal={IEEE Transactions on Evolutionary Computation},
volume={23},
number={6},
pages={949--961},
year={2019},
publisher={IEEE}
}