

**Algorithm XXXX: VTMOP: Solver for Blackbox Multiobjective Optimization Problems**

TYLER H. CHANG, Virginia Polytechnic Institute and State University, Dept. of Computer Science, USA

and Argonne National Laboratory, Mathematics and Computer Science Division, USA   
LAYNE T. WATSON, Virginia Polytechnic Institute and State University, Depts. of Computer Science, Mathe-

matics, and Aerospace and Ocean Engineering, USA   
JEFFREY LARSON, Argonne National Laboratory, Mathematics and Computer Science Division, USA NICOLE NEVEU, SLAC National Accelerator Laboratory, USA   
WILLIAM I. THACKER, Winthrop University, USA   
SHUBHANGI DESHPANDE, Oracle Labs, USA   
THOMAS C. H. LUX, Virginia Polytechnic Institute and State University, Dept. of Computer Science, USA

VTMOP is a Fortran 2008 software package containing two Fortran modules for solving computationally expensive bound-constrained blackbox multiobjective optimization problems. VTMOP implements the algorithm of [32], which handles two or more objectives, does not require any derivatives, and produces well-distributed points over the Pareto front. The irst module contains a general framework for solving multiobjective optimization problems by combining response surface methodology, trust region methodology, and an adaptive weighting scheme. The second module features a driver subroutine that implements this framework when the objective functions can be wrapped as a Fortran subroutine. Support is provided for both serial and parallel execution paradigms, and VTMOP is demonstrated on several test problems as well as one real-world problem in the area of particle accelerator optimization.

CCS Concepts: · **Mathematics of computing** → **Mathematical optimization**; · **Applied computing** → **Multi-criterion optimization and decision-making**.

Additional Key Words and Phrases: multiobjective optimization, blackbox optimization, trust region methods, response surface methodology, adaptive weighting schemes

1 INTRODUCTION

Multiobjective optimization problems (MOPs) arise in many disciplines of science and engineering and are prevalent in the ield of multidisciplinary design optimization [60]. In such problems, a multidisciplinary team of engineers must collaborate to design a large complex system, such as an aircraft, balancing design tradeofs spanning diferent engineering subields. In this context, the solution to the MOP is a multidimensional tradeof

Authors’ addresses: [Tyler H. Chang,](https://orcid.org/0000-0001-9541-7041) tchang@anl.gov, Virginia Polytechnic Institute and State University, Dept. of Computer Science, Blacksburg, Virginia, USA, 24061 and Argonne National Laboratory, Mathematics and Computer Science Division, Lemont, Illinois, USA, 60439; [Layne T. Watson,](https://orcid.org/0000-0003-2009-107X) ltw@cs.vt.edu, Virginia Polytechnic Institute and State University, Depts. of Computer Science, Mathematics, and Aerospace and Ocean Engineering, Blacksburg, Virginia, USA, 24061; [Jefrey Larson,](https://orcid.org/0000-0001-9924-2082) jmlarson@anl.gov, Argonne National Laboratory, Mathematics and Computer Science Division, Lemont, Illinois, USA, 60439; [Nicole Neveu,](https://orcid.org/0000-0002-5012-3856) nneveu@slac.stanford.edu, SLAC National Accelerator Laboratory, Menlo Park, California, USA, 94025; [William I. Thacker,](https://orcid.org/0000-0001-7806-6593) thackerw@winthrop.edu, Winthrop University, Rock Hill, South Carolina, USA, 29733; [Shubhangi Deshpande,](https://orcid.org/0000-0001-8668-5992) shubhangid@gmail.com, Oracle Labs, Belmont, California, USA, 94002; [Thomas C. H.](https://orcid.org/0000-0002-1858-4724)

[Lux,](https://orcid.org/0000-0002-1858-4724) tchlux@vt.edu, Virginia Polytechnic Institute and State University, Dept. of Computer Science, Blacksburg, Virginia, USA, 24061.

ACM acknowledges that this contribution was authored or co-authored by an employee, contractor, or ailiate of the United States government. As such, the United States government retains a nonexclusive, royalty-free right to publish or reproduce this article, or to allow others to do so, for government purposes only.

© 2022 Association for Computing Machinery.

0098-3500/2022/5-ART $15.00   
<https://doi.org/10.1145/3529258>

ACM Trans. Math. Softw.



|  |  |  |
| --- | --- | --- |
| 2 | • | Chang et al. |

surface, called the Pareto front, describing the interactions between several conlicting design criteria. Under-standing the shape of the Pareto front allows the design engineers to make an informed decision about how to balance design tradeofs a posteriori based on the results of the design optimization. Speciic examples of MOPs in engineering are described by [19], who present a ship hull design problem, and by [67], who describe a chemical engineering problem.

The standard form for a MOP is a real vector-valued minimization problem. The multiobjective cost function F is of the form F : X → Y, where X ⊂ Rdis the feasible design space and Y ⊂ Rpis the feasible objective space. For y,z ∈ Rd, the notation y < z indicates that y is componentwise less than z, and the notation y<= z indicates that y is componentwise less than or equal to z, allowing for the possibility that y = z. This paper considers MOPs where X is a simply bounded set, meaning that X = [L,U ] = {x ∈ Rd: L<= x<= U } for some L,U ∈ Rdwith L < U . Then Y is the image of [L,U ] under F. Conceptually, F can be decomposed into p scalar cost functions fi : X → R, i = 1, . . ., p such that F (x) = (f1(x), . . ., fp(x))⊤.

The solution to a MOP is deined by the partial ordering ≤ on Y. For Y,Z ∈ Y, Y ≤ Z if Y is componentwise less than or equal to Z, with strict inequality in at least one component. An objective value F (x∗) is said to be nondominated if F (x) ̸≤ F (x∗) for all x ∈ X. If F (x∗) is nondominated, then x∗is said to be eicient, and the pair (x∗, F (x∗)) is Pareto optimal. The Pareto front is given by the set of all nondominated objective points and often is accompanied by the set of all eicient designs, called the eicient set. The Pareto front has dimension at most p − 1 and can be nonsmooth or even discontinuous. Further reading on MOPs can be found in [33]. In order to discuss the convergence of algorithms, it is convenient to use some additional terminology. In particular, a point x†is locally Pareto optimal if there is no y in a neighborhood of x†such that F (y) dominates F (x†). Additionally, when discussing the solutions returned by a multiobjective solver, the term approximately nondominated may be used to indicate an objective value F ( ˆx) that is not dominated by any other objective points evaluated by the solver. Then ˆx is approximately eicient, and ( ˆx, F ( ˆx)) is approximately Pareto optimal.

Remark 1.1. When discussing MOPs, one should distinguish the design and objective spaces. In this paper, x, y, and z denote points in the design space, and X, Y, and Z are reserved to denote points in the objective space (or a projection of the objective space, as described in Section 3.1). Each of these entities may be annotated with a superscript to index points or vectors from a set or sequence or with a subscript to denote the components of a point or vector.

This paper focuses on blackbox MOPs. A process, such as the objective function F, is said to be a blackbox when it can be evaluated at speciied inputs to obtain outputs, but the process provides no additional information that is not immediately available in its output. Often, blackbox functions are the result of physical experiments or computationally expensive numerical simulations that require substantial computing resources and time for each evaluation. In some cases, a blackbox multiobjective function F could imply that each component function fi is an individual blackbox, which must be evaluated separately at a great computational expense. In other cases, all components of F may be obtained through a single integrated simulation.

This paper presents the VTMOP software package for solving computationally expensive blackbox MOPs subject to simple bound constraints. Distinctive features of VTMOP are that it (1) handles any p>= 2, (2) does not require derivative information, and (3) devotes considerable efort to produce well-distributed points over the Pareto front. VTMOP also features a solver subroutine that can be used to solve MOPs when F is available as a Fortran callable function; a łreturn-to-callerž interface that can be used when F must be decoupled from the optimization algorithm; both parallel and serial execution paradigms; a checkpointing system for storing and recovering function evaluation and iteration data; and support for user-deined local optimization and surrogate modeling procedures.

The paper is organized as follows. Section 2 provides some background on multiobjective optimization algorithms (MOAs) and widely distributed software packages. Section 3 outlines the VTMOP algorithm and

ACM Trans. Math. Softw.



|  |  |  |
| --- | --- | --- |
| Algorithm XXXX: VTMOP: Solver for Blackbox Multiobjective Optimization Problems | • | 3 |

worker subroutines. Section 4 presents two interfaces for solving blackbox MOPs using VTMOP: a return-to-caller interface and a driver subroutine, both of which use the worker subroutines outlined in Section 3. Section 5 is a brief summary of the VTMOP users’ manual. Section 6 provides performance results for VTMOP on several analytic test problems. Finally, Section 7 demonstrates VTMOP’s performance on a real-world particle accelerator optimization problem.

2 ALGORITHMS AND SOLVERS FOR MULTIOBJECTIVE OPTIMIZATION

This section reviews MOAs and techniques to solve blackbox MOPs of the form

x ∈[L,U ]F (x), (1)

where the minimization is understood as Pareto optimality. In general, the solution to (1) can be an uncountably ininite set, described by a (p − 1)-dimensional manifold in the objective space. (When the Pareto front has dimensionality less than p − 1, this indicates that at least one objective could be eliminated.) The MOAs presented in this section produce a set of approximately nondominated objective points (and corresponding approximately eicient designs), which provide a discrete approximation to the Pareto front.

2.1 Multiobjective Optimization Techniques and Algorithms

Three fundamentally diferent approaches can be used to solve MOPs. The irst approach consists of a priori methods, where the decision makers are able to express some preference about the tradeof **before** viewing the results of the optimization. In these cases, the decision maker typically supplies a preference function that reduces the MOP to a scalar optimization problem. The second approach consists of a posteriori methods, where the decision maker is not able to express any preference until **after** viewing the results of the optimization procedure. This class is the most general class of methods and the subject of this paper. The third class consists of łhuman-in-the-loopž algorithms, where the decision maker is able to progressively provide feedback on preference while the algorithm is running. A summary of all three techniques can be found in [51]; a detailed survey of a posteriori techniques for computationally expensive blackbox MOPs is given by [39].

A standard a posteriori technique for solving blackbox MOPs is to use multiple scalarization functions. Scalar-ization reduces a MOP to a scalar optimization problem by composing a scalarization function G : Rp→ R with F. The resulting function G ◦ F can be minimized by using a scalar blackbox optimization solver to ind a single (approximately) nondominated point. Optimizing diferent scalarizations can produce a discrete set of approx-imately nondominated points, thus approximating the Pareto front. Further reading on general scalarization schemes can be found in Chapter 4 of [33]. One common scalarization scheme is the weighted sum method, which uses a vector of nonnegative weights w = (w1, . . ., wp) to produce a łweighted averagež cost function

p

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Gw | � | F (x) | � | = | � | wi fi (x). | (2) |

When w > 0, any minimizer of (2) is Pareto optimal. However, if any of the component functions fi are nonconvex, then it is possible that not every point on the Pareto front can be obtained by solving (2) for any vector of weights w. Furthermore, naïvely minimizing (2) using weights that are spaced equally tends to produce clusters of objective values and can leave gaps along the Pareto front. These give the decision maker a poor understanding of its true shape. Therefore, efective weighted sum scalarization depends on an adaptive weighting scheme. Further reading on weighted sum scalarization can be found in Chapter 3 of [33].

One issue with scalarization techniques is that each subproblem must be solved individually as a blackbox scalar optimization problem. Therefore, the cost of solving the MOP is many times that of solving each scalar subproblem. When F is computationally expensive to evaluate, this can be prohibitive. The response surface

ACM Trans. Math. Softw.



|  |  |  |
| --- | --- | --- |
| 4 | • | Chang et al. |

methodology (RSM) can be used to mitigate these expenses [53]. In the multiobjective RSM, a computationally cheap (to evaluate) surrogate function is it to each objective function by using values from evaluating the objective at experimental design points. Then multiple scalarizations of these surrogates are optimized, each producing a candidate design point. If the design of experiments is thorough and the surrogates are accurate, evaluating these candidate designs will produce numerous points near the Pareto front. Thus many approximate Pareto points can be found for little more than the cost of evaluating a single experimental design.

In modern settings, evolutionary MOAs [1] are perhaps the most widely used class of MOAs. These heuristic algorithms extend evolutionary algorithms to the multiobjective case by using a itness sorting criterion based on the partial ordering ≤. An elitist evolutionary MOA also maintains previously observed nondominated solutions, using them to ensure that the region of the objective space that is dominated by the kth generation is a superset of the dominated region after the (k − 1)th generation. Perhaps the most well-known example of an evolutionary MOA is the nondominated sorting genetic algorithm (NSGA-II) [29], an elitist evolutionary MOA with low iteration complexity. NSGA-II tends to have poor performance when the number of objectives is large, so it has an extension NSGA-III [16] that takes a more a priori approach, by attempting to minimize along user-given reference directions. Evolutionary MOAs have an advantage over most scalarization schemes in that they are capable of producing solution points over the Pareto front in each generation. However, the function evaluation budget requirement for evolutionary MOAs tends to be too large for them to be used for computationally expensive problems. For the test problems presented in [30], the recommended budget for an evolutionary algorithm to solve problems with 5ś20 design variables and 3 objectives is between 20,000 and 50,000 function evaluations. For a computationally expensive problem where each function evaluation could require minutes, hours, or days and occupy massive parallel resources, requiring 20,000 function evaluations would be prohibitive. For computationally expensive problems, evolutionary MOAs are often combined with the RSM to reduce computational expense, as described in [34] and [52].

Another class of algorithms is direct search MOAs that extend scalar direct search methods. One example of direct search MOAs consists of multiobjective extensions of the scalar global optimization algorithm DIviding RECTangles (DIRECT) [40]. DIRECT is globally convergent for nonconvex functions if the objective function is Lipschitz continuous with a Lipschitz constant that is bounded in the feasible design space X. Multiobjective DIRECT (MODIR) [19] extends DIRECT to the multiobjective case by deining and subdividing potentially Pareto optimal hyperintervals. Multiobjective DIRECT algorithms of this form ofer similar convergence guarantees as Jones’ DIRECT when every component function of the objective is Lipschitz continuous [50]. However, in practice, many function evaluations are needed to achieve high-quality solution sets, so [19] recommend combining MODIR with a faster locally convergent MOA, such as the derivative-free multiobjective line search algorithm of [49]. Another example of direct search MOAs are the extensions of mesh adaptive direct search (MADS) [7] to MOPs. These techniques combine a global search step with a local polling strategy, which drives convergence to locally Pareto optimal points. In general, these algorithms converge to points that are locally Pareto optimal even for nonsmooth component functions, and they may even converge to a true Pareto point in the limit, depending on the choice of search step. Notable examples of multiobjective MADS algorithms include the biobjective algorithm BiMADS [8] and the algorithm MultiMADS [9], both of which formulate scalarizations of the problem to attempt to produce evenly spaced points on the Pareto front. More recent MADS extensions, such as DMulti-MADS [14] have avoided explicit scalarization by iterating on subsets of the nondominated poll points. Finally, Direct MultiSearch (DMS) [28] and MultiGLODS [27] are other notable direct search techniques, which also use polling to drive local convergence, but do not speciically extend the MADS polling strategy.

VTMOP implements the MOA of [32], which combines an adaptive weighting scheme for weighted sum scalarization, RSM, and trust-region methods, the latter of which are widely used in scalar optimization [56]. In the context of the RSM, each local trust region (LTR) deines a region of the design space in which the current response surface approximation can be used as a surrogate. In the context of MOPs, these LTRs can be used

ACM Trans. Math. Softw.



|  |  |  |
| --- | --- | --- |
| Algorithm XXXX: VTMOP: Solver for Blackbox Multiobjective Optimization Problems | • | 5 |

to control the spacing of points on the Pareto front and force solutions in nonconvex portions of the Pareto front when combined with a weighted sum scalarization approach [59]. [64] show that the technique of inding Pareto minima of surrogates within a sequence of LTRs converges to a locally Pareto optimal point, given that the surrogates’ accuracies are improving. [32] rely on a global search via DIRECT followed by RSM inside LTRs to produce subsequences of converging solutions. Therefore, this MOA’s convergence is largely driven by the convergence of the global search strategy and the accuracy of the surrogates. As discussed in Section 3, VTMOP is designed to allow for various search strategies, LTR optimizers, and surrogate procedures. However, for the techniques suggested by [32], Lipschitz continuity of all fi will guarantee convergence.

2.2 Multiobjective Optimization Sotware

Relatively few of these techniques and algorithms are available in widely distributed software packages for solving computationally expensive MOPs. For such problems, many simulation packages support multiobjective design optimization by solving a single scalarized subproblem, producing a single Pareto optimal design. For example, NASA’s FUN3D package performs luid dynamics based design simulations and provides support for multiobjective analyses by using the design drivers KSOPT, PORT, or SNOPT to solve a single scalarized subproblem, as described in Section 9.9 of [13]. This approach falls under the a priori techniques mentioned in Section 2.1. While this approach is reasonable for the extremely limited evaluation budget of most computational luid dynamics based analysesÐ[13] recommend a budget of just 20 simulationsÐit results in no understanding of the complex design tradeofs and is less general than the a posteriori approaches favored here.

With a posteriori methods, the following options are available. An implementation of BiMADS is available in the widely used package NOMAD v. 3 [45]. This implementation can be used for constrained, unconstrained, mixed integer, and continuous problems with p = 2 objectives, but since BiMADS only targets biobjective problems, this does not extend to p > 2 objectives. For p > 2 objectives, MATLAB implementations of DMS and MultiGLODS are available in the BoostDFO MATLAB toolbox [61], including a parallel implementation of both. A Fortran 90 implementation is available for MODIR, which is designed to use a user-speciied portion of its function evaluation budget to improve MODIR’s solutions via a multiobjective line search. Finally, although there are many widely used implementations of both, the oicial implementations of NSGA-II and NSGA-III are available within the Python package PyMOO [15].

Several other software packages are worth mentioning, although they are not immediately relevant to this paper. The Python package PyMOSO [26] solves multiobjective simulation optimization problems on an integer lattice and provides a Python framework for implementing new multiobjective simulation optimization algorithms. Additionally, [3] provide a MATLAB toolbox for applying surrogate modeling to MOPs.

Fitting into the above mentioned landscape of available multiobjective software, VTMOP provides a robust framework and solver for bound-constrained blackbox optimization problems, with an emphasis on producing evenly distributed Pareto front approximations for p > 2 objectives. Some of the techniques used to achieve eiciency and scalability for computationally expensive problems incur a nontrivial iteration cost. Therefore, VTMOP is best suited for MOPs that are computationally expensive to evaluate.

Problems withp > 2 objectives (the so-called łmany-objectivež cases) are generally considered to be signiicantly more diicult to solve than problems where p = 2 because of the relative sparsity of any reasonably sized set of solution points on a high-dimensional Pareto front. In general, other software packages and algorithms for solving these problems either place an additional focus on inding uniformly spaced solution sets or recommend reducing the number of objectives [48]. VTMOP falls into the former of these two categories. NSGA-III [16] is one of the most commonly used examples of the irst approach, although it is not a purely a posteriori solution since it depends upon user input of precomputed uniformly spaced reference directions.

ACM Trans. Math. Softw.



|  |  |  |
| --- | --- | --- |
| 6 | • | Chang et al. |

Another challenge in many-objective optimization is that of efectively presenting visualizations of the Pareto front approximation to a decision maker. Addressing this challenge is beyond the scope of this work, but several techniques are described by [65].

3 A FRAMEWORK FOR MULTIOBJECTIVE OPTIMIZATION

The MOA of [32] that is implemented in this paper applies the RSM to a sequence of LTRs, each centered at a design point corresponding to an łisolatedž objective value from the current set of nondominated points. The algorithm begins from the zeroth iteration, and the kth iteration of the algorithm can be summarized by the following four-step process.

1. If k > 0, update the current set of nondominated points, and then identify an isolated point F ( ˜x(k)) in the current set of nondominated points. Update thekth LTR ∆(k), centering about ˜x(k), and assign thekth set W(k) of adaptive weights based on objective values near F ( ˜x(k)). If k = 0, then there is no current nondominated set; ∆(0)= [L,U ], and W(0)is assigned predetermined values.

2. Perform the kth exploration by sampling F within ∆(k); if k = 0, then a large exploration budget may be necessary.

3. Fit p surrogates,ˆf(k) i ≈ fi for i = 1, . . ., p, using the data gathered from Step 2 (plus any data available from previous iterations if k > 0). Apply each weight vector in W(k)toˆf(k) 1 , . . .,ˆf(k) p , and minimize the resulting scalarized surrogate problems by using a local optimization strategy. This produces the kth batch C(k)of candidate design points.

4. Evaluate F (z) for all z ∈ C(k). Increment k. Update the current database of function values; check for termination conditions; and if no termination conditions have been triggered, proceed to the next iteration. A lowchart for this process is illustrated in Figure 1. Detailed descriptions of Steps 1ś4 are presented in Sections 3.1, 3.2, 3.3, and 3.4, respectively. Note that evaluations of F are required only during Steps 2 and 4.

3.1 Computing the Local Trust Region and Choosing the Adaptive Weights

In order to construct the kth LTR and select the adaptive weights when k > 0, VTMOP must identify an łisolated pointž in the current nondominated point set. VTMOP centers the kth LTR at a design point, in the current approximation to the eicient set, whose image is łfar awayž from any neighbors in the objective space. This enables VTMOP to spend function evaluations where they are needed most and ill in gaps on the current approximation to the Pareto front. When p = 2, the Pareto front is a one-dimensional curve, which admits a natural (left-to-right) ordering of points in objective space. [59] identify an isolated point by considering the Euclidean distance from each point in the nondominated set to its left and right neighbors. However, this approach does not generalize to p > 2.

[32] generalize this approach to arbitrary dimension using the Delaunay graph. Let D(k)=�x(k,1), . . ., x(k,n(k))�denote the set of n(k)design points that have been evaluated at the start of iteration k, and let F(k)=�F (x(k,1)), . Then the kth approximation to the Pareto front is . . ., F (x(k,n(k)))�

P(k)=�Y ∈ F(k): X ≰ Y for all X ∈ F(k)�, (3)

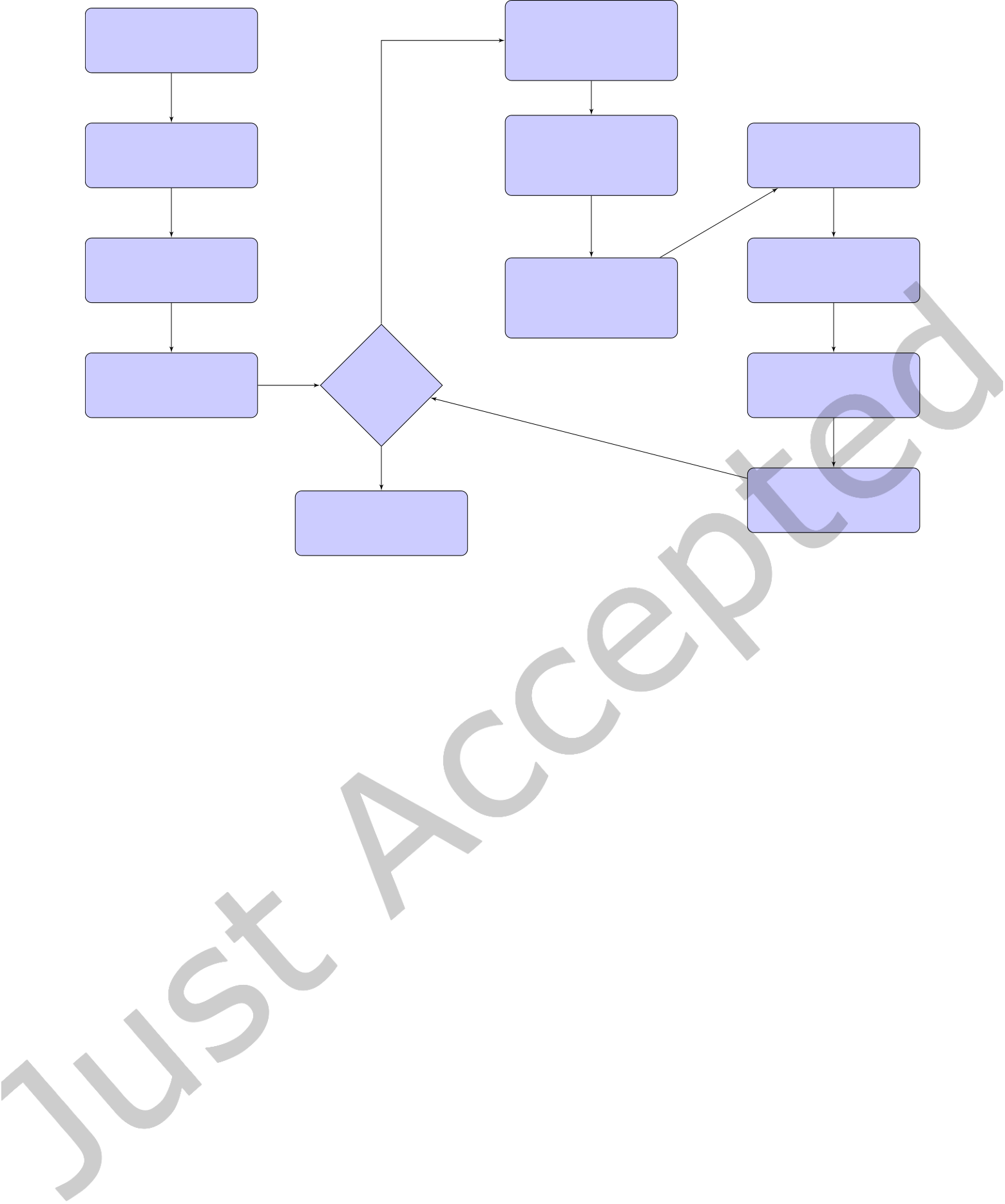
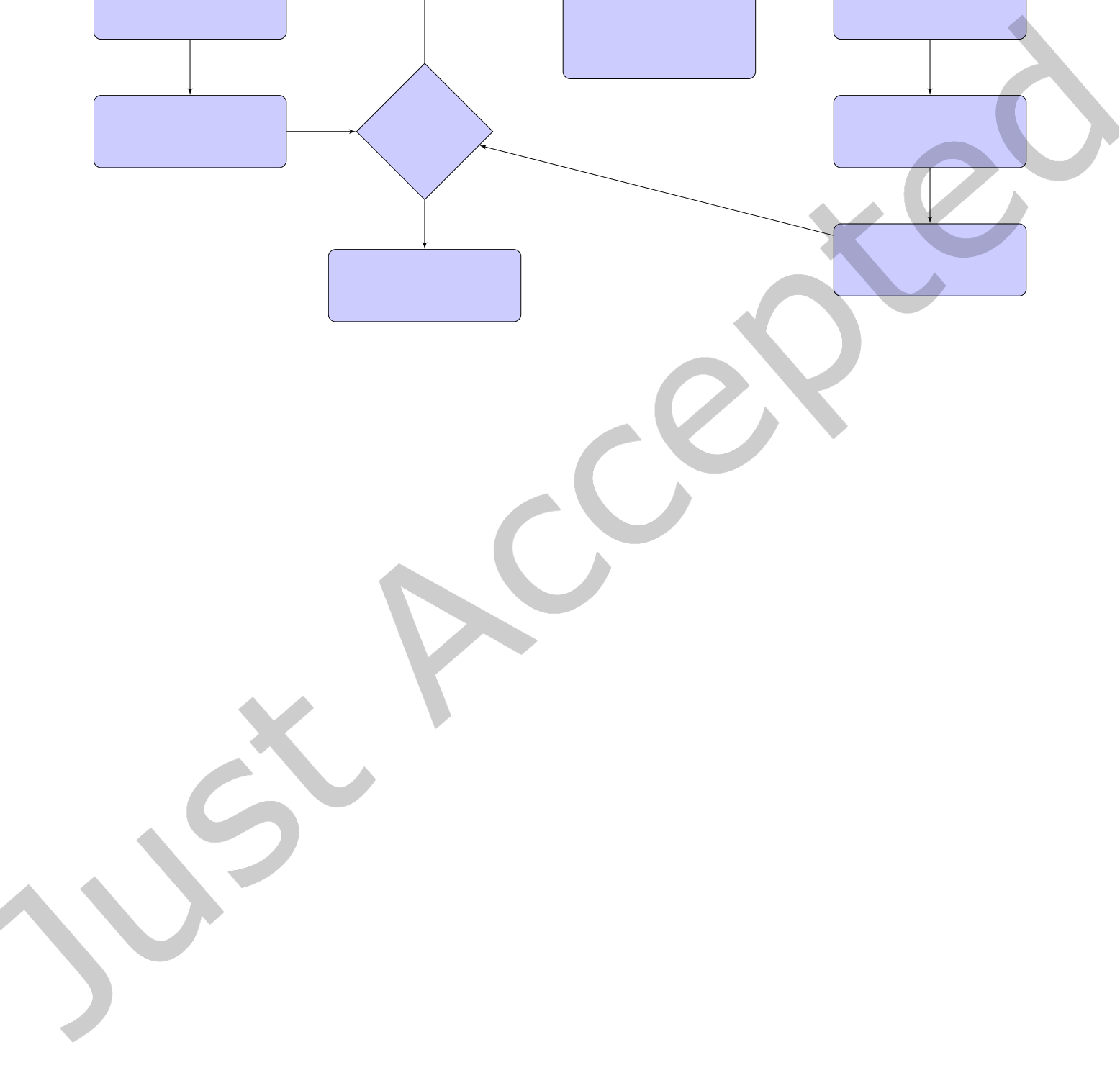
and the kth approximation to the eicient set is

E(k)=�x(k,i): F

F � � x(k,i)� x(k,i)�= F  
 ∈ P(k)and

�x(k,j)�⇒ i<= j for j = 1, . . . ,n(k)�.   
 (4)

ACM Trans. Math. Softw.



|  |  |  |
| --- | --- | --- |
| Algorithm XXXX: VTMOP: Solver for Blackbox Multiobjective Optimization Problems | • | 7 |

k ← k + 1;

|  |  |
| --- | --- |
| k = 0;  initial search in X | compute the kth  approx. non- |

dominated set

|  |  |  |  |
| --- | --- | --- | --- |
| fit a surro- | no | identify the most | search over kth LTR |
| isolated point F(˜x(k)) |
| gate to each fi | in the kth approx. |
| optimize p + 1 weight- | nondominated set | update surrogates |
| choose the kth |
| ings of the surrogates |
| adaptive weights |

and fit the kth   
LTR about ˜x(k)

|  |  |  |
| --- | --- | --- |
| evaluate candidate de- | stopping | optimize the surro- |
| criteria | gates using the kth |
| signs predicted above |
| reached? | adaptive weights |

yes

|  |  |
| --- | --- |
| compute final approx.  nondominated and | evaluate the designs  predicted above |

efficient set and return

Fig. 1. Algorithm flowchart for VTMOP. For further details on each of these steps, see Sections 3.1, 3.2, 3.3, and 3.4 .

Remark 3.1. One practicality of many real-world MOPs, especially those with p > 2 objectives, is that certain regions of the Pareto front may be fundamentally uninteresting. In particular, decision makers may be interested in studying tradeofs only after certain bounds on the objective values have been met. In these situations, the default version of VTMOP may waste precious function evaluations illing in gaps in uninteresting regions of the objective space. VTMOP can account for this by allowing users to provide lower/upper bounds on the range of łinterestingž objective values, via optional inputs Ol and Ob. When Ol and Ob are given, (3) is replaced by

P(k)=�Y ∈ F(k): X ≰ Y for all X ∈ F(k)such that Ol<= X<= Ou�.

Note that the inclusion of this option does not afect the computation of the inal Pareto front approximation, which always includes all observations that are not dominated. Also note that this option must be used with care since providing overly strict bounds may prevent VTMOP from identifying any points that satisfy them, resulting in early termination. One strategy to circumvent this issue is to run early iterations with relaxed bounds, then tighten the bounds in later iterations after several points have already been found that meet the stricter criteria.

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Because the Pareto front is (generically) a (p − 1)-dimensional manifold, the irst step is to project P(k)into its natural dimension. Let P(k)=�Y(k,1), . . ., Y(k,m(k))�, where m(k)is the number of unique points in the kth | | | | | | | | | | | | |
| approximately nondominated set; let Y(k,i) 1 | | | , . . ., Y(k,i) p | | denote the components of Y(k,i); and let s(k)be a constant | | | | | | | |
| such that s(k)< Y(k,i) p | for i = 1, . . ., m(k). Then the (p − 1)-dimensional projected set is given by | | | | | | | | | | | |
| Π(k)= | | �Z(k,1), . . . ,Z(k,m(k))� | | , Z(k,i)= | |  | Y(k,i) 1 , . . . , | Y(k,i) p−1 | |  | . | (5) |
|  | Y(k,i) p − s(k) | Y(k,i) p | − s(k) | � |  |
| Remark 3.2. In order to avoid division by zero or a small number when computing (5), the denominatorsY(k,i) p | | | | | | | | | | | | −s(k) |

must be strictly greater than 0 for all i = 1, . . ., m(k). In VTMOP, this is done by choosing s(k)= minY ∈P(k) Yp − 1.

ACM Trans. Math. Softw.



|  |  |  |
| --- | --- | --- |
| 8 | • | Chang et al. |

Remark 3.3. Duplicate values are determined by using distance in the projected space in order to prevent Π(k)

from containing duplicate values, which would present problems in the coming computations. Speciically, VTMOP computes (3), (4), and (5) at the same time, and for any Z(k,i)that satisies ∥Z(k,i)− Z(k,j)∥2 < ϵ, the corresponding values Y(k,i)and Y(k,j)are treated as duplicates. In particular, if j < i, then Z(k,i)is omitted from Π(k), Y(k,i)is omitted from P(k), and F−1(Y(k,i)) is omitted from E(k). Here ϵ is the objective space tolerance, a

small scale and machine and user preference dependent constant.

After computing the projected set Π(k), its Delaunay graph DG(Π(k)) is used to infer a neighborhood structure. Further details on the Delaunay graph and its computation are provided in Section 3.1.1. From DG(Π(k)), an isolation score is computed for eachY(k,i)∈ P(k). Let N(k,i)=�Y(k,j): Z(k,j)be a neighbor of Z(k,i)in DG(Π(k))�.

Then the isolation score for Y(k,i)is

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ((k,i)) |  | ∥Y(k,i)− Y ∥2 | . |  |
| δY = | Y ∈N(k,i)� | |N(k,i)| |  | (6) |

In general, the kth LTR is centered at a design point ˜x(k)that is chosen so that F ( ˜x(k)) ∈ arg maxY ∈P(k)δ (Y ), whose neighborhood˜N(k)is the corresponding N(k,i). If the Y from the arg max is not unique, one chooses the Y(k,i)with smallest i.

Remark 3.4. In rare cases, it is possible that m(k)= 1. Then ˜x(k)is given by the only element of E(k), and˜N(k)= ∅.

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Let ˜r(k)∈ (0, 1) denote the kth LTR radius fraction. Then the kth LTR is given by | | | | | | | | | (7) |
| ∆(k)= | � | ˜x(k)− ˜r(k)(U − L), ˜x(k)+ ˜r(k)(U − L) | � | ∩ | � | L,U | � | . |

Note that ∆(k)is a simply bounded set.

Remark 3.5. Ideally, performing the kth iteration within ∆(k)will generate design data in the neighborhood of ˜x(k)and ill in the Pareto front in the neighborhood of F ( ˜x(k)). When the Pareto front is nonsmooth or

discontinuous, however, VTMOP may fail to produce any approximately Pareto optimal points that are near F ( ˜x(k)) in the objective space. In these cases, ˜x(k)could be a re-occurrence of a previous LTR center. Every time∥ ˜x(k)− ˜x(k′)∥2 < µ for some k′< k, the fraction ˜r(k)is decayed using ˜r(k)= τ ˜r(k∗), where 0 < τ < 1, and k∗is the largest such k′. Here µ is the design space tolerance, a small scale and machine dependent constant. If ˜r(k) drops below some predetermined tolerance, then ˜x(k)is skipped, and the next most isolated point is used instead. For every **new** value of ˜x(k), ˜r(k)is initialized to a default value, regardless of whether the LTR radius has been decayed in previous iterations. The starting trust region radius fraction ρ(0), the decay factor τ, the trust region tolerance ρ(1), and the design space tolerance µ are all optional inputs to VTMOP.

The zeroth iteration is a special case. VTMOP allows users to (optionally) supply a database of previously

evaluated designs. However, VTMOP assumes that there is insuicient data available at the start of the zeroth iteration; there can be no E(0)or P(0). Instead, ∆(0)= [L,U ], resulting in an exploration of the entire design space.

When F is nonconvex, the convergence of VTMOP to the Pareto front is entirely dependent on a thorough design

space exploration during the zeroth iteration.

After computing ∆(k), the kth set of adaptive weights is selected. When k = 0 or if |˜N(k)| < 1, then the adaptive weights are given by W(k)=

in Rp. Otherwise, |W(k)| = p + |˜N(k)|, and the values of each weight vector are determined as described in�e(1), . . ., e(p),�p i=1e(i)/p�, where e(i)is the ith standard basis vector

Section 3.1.2. The entire process of identifying an isolated point, constructing the kth LTR, and choosing the

adaptive weights is summarized in Algorithm 1.

ACM Trans. Math. Softw.



|  |  |  |
| --- | --- | --- |
| Algorithm XXXX: VTMOP: Solver for Blackbox Multiobjective Optimization Problems | • | 9 |

**Algorithm 1** Identify an isolated point, construct an LTR, and set adaptive weights.

**input**

k is the current iteration of VTMOP;

[L,U ] is the feasible design space;   
D(k)is the kth design database, containing n(k)design points that were evaluated in previous iterations of VTMOP (if k = 0, then D(0)could contain designs that were evaluated during previous analyses or D(0)= ∅);   
F(k)=�F (x) : x ∈ D(k)�;

ρ(0)is the starting trust region radius (given as a scalar fraction of U − L); ρ(1)is the trust region tolerance (given as a scalar fraction of U − L);

τ is the decay factor for the trust region radius;

µ is the design space tolerance;   
If k > 1, then ˜r(1), . . ., ˜r(k−1)are the previous trust region radii fractions; If k > 1, then ˜x(1), . . ., ˜x(k−1)are the previous trust region centers;

**begin**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **if** k = 0 **then** | | = | �Y(k,i)= F | �y(k,i)� ��� 1 ≤ i ≤ m(k)� | as in (3); |
| **else** | ∆(k)← [L,U ];  W(k)←�e(1), . . . ,e(p),�p i=1e(i)/p�;  P(k)←�Y ∈ F(k): X ≰ Y for all X ∈ F(k)� |
| E(k)← subset of D(k)as in (4); | |
| Compute Π(k)using (5); | |

m(k)← |Π(k)|; **comment** The cardinality |P(k)| = |E(k)| = |Π(k)| may have changed. Compute DG(Π(k)) (Section 3.1.1);   
**for** i = 1, . . ., m(k)**do**   
 Compute N(k,i)based on DG(Π(k));   
 Compute δ (Y(k,i)) using (6);

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **enddo** | � | 1, . . . ,m(k)� | ; ˜x(k)← null; | � | ; |
| **initialize** J(k)← |
| **while** ˜x(k)= null **and** J(k)� ∅ **do** | | | |
| i∗= min{arg max i ∈J(k) δ�Y(k,i)�};  **if** ∥ ˜x(k′)− y(k,i∗)∥2 < µ for any k′< k **then**  k∗← max  **if** τ ˜r(k∗)> ρ(1)**then** �k′ ��� k′< k and��� ˜x(k′)− y(k,i∗)���2 < µ ˜r(k)← τ ˜r(k∗); | | | |
| ˜x(k)← y(k,i∗); | | | |
| ˜N(k)← N(k,i∗); | | | |
| **else** | | | |
| J(k)← J(k)\ {i∗}; | | | |

**endif**

**else**  
˜r(k)← ρ(0);  
˜x(k)← y(k,i∗);  
˜N(k)← N(k,i∗);

**endif**  ACM Trans. Math. Softw.

**enddo**   
**if** ˜x(k)= null **then**

All candidate LTR centers have been maximally reined, which is a termination condition for VTMOP;

**return** an appropriate termination message;

**endif**  
∆(k)←�˜x(k)− ˜r(k)(U − L), ˜x(k)+ ˜r(k)(U − L)�∩ [L,U ] as in (7);

**if** |˜N(k)| < 1 **then**

**else** W(k)←�e(1), . . . ,e(p),�p i=1e(i)/p�;

Compute W(k)using (8), as described in Section 3.1.2;



|  |  |  |
| --- | --- | --- |
| 10 | • | Chang et al. |

The dominant costs for Algorithm 1 are computing P(k)using (3) and computing the Delaunay graph DG(Π(k)) as described in Section 3.1.1. The O time complexity for computing (3) is O(n(k)m(k)); and as described in Section 3.1.1, the O time complexity of computing DG(Π(k)) is a cubic function of m(k)in practice. In practice, the time required by Algorithm 1 is signiicantly less than the time required to solve the surrogate optimization problems, as described in Section 3.3 and Algorithm 2. A parallel implementation of Algorithm 1 is also briely described in the beginning of Section 4.3, which further reduces the wallclock time. In the package VTMOP, the subroutine VTMOP\_LTR implements Algorithm 1.

3.1.1 Computing the Delaunay Graph. DG(Π(k)) is deined in terms of the Voronoi tessellation of Π(k). The Voronoi tessellation of Π(k)is a covering of Rp−1by closed convex polytopes (with disjoint interiors), such that (1) each polytope contains exactly one point in Π(k)(this point is called the Voronoi site of that polytope) and each point in Π(k)is in some polytope interior, and (2) every point in each polytope is as close or closer to its Voronoi site than to any other Voronoi site. The node set for DG(Π(k)) is given by the set of all Voronoi sites, and the edge list is given by connecting all sites that share a boundary in the Voronoi tessellation.

In the literature, the Delaunay graph is computed as a byproduct of the Delaunay triangulation [18]. However, because of the exponential size of Delaunay triangulations in high-dimensional spaces [41], this approach becomes prohibitively expensive when the number of objectives is of moderate size, for example, when p > 6. Another strategy for computing the Delaunay graph is to compute each Voronoi polytope individually [24], allowing for each node’s neighborhood to be constructed separately and in parallel. However, this approach ultimately requires more total computations than when constructing the complete Delaunay triangulation, and it is not robust when the input data exhibits geometrical degeneracies.

Instead, VTMOP implements a novel strategy for constructing the connectivity matrix for the Delaunay graph; this strategy is capable of scaling to values of p that are far larger than the number of objectives in any reasonably sized MOP. The proposed Delaunay graph algorithm utilizes the software package DELAUNAYSPARSE [22] to generate a quadratic number of Delaunay simplices, and the expected complexity of this approach is cubic in m(k)and sub-quartic in p [24]. This approach is also highly parallelizable, and across numerous real-world and test problems from the literature, the values of m(k)and p do not grow large enough to constitute a signiicant expense in VTMOP. Since the details of this Delaunay graph algorithm are novel but not critical to understanding VTMOP, interested readers are directed to the Appendix A.

3.1.2 Choosing the Adaptive Weights. [32] assign a set W(k)of p + |˜N(k)| weight vectors in the kth iteration, given that k > 0 and |˜N(k)| > 0. The irst p of these weight vectors are always the standard basis vectors e(1), . . ., e(p). By searching for an individual minimum of each component function in every iteration, VTMOP expands the coverage of its approximation to the Pareto front.

The remaining |˜N(k)| weight vectors are assigned based on the objective scaling in order to ill in gaps in the current approximation to the Pareto front. For each˜Y ∈˜N(k), the corresponding adaptive weight vector˜W is given by

|  |  |  |
| --- | --- | --- |
| where ˜c is a normalizing constant such that ˜W1 + . . . +˜Wp = 1. ˜Wi =  0,  ���fi�˜x(k)�  ˜c  −˜Yi���  ,   if  otherwise,���fi�˜x(k)� | −˜Yi��� < ϵ, | (8) |
| If k = 0 or |˜N(k)| = 0, then there are no łgapsž to ill in. Therefore, VTMOP focuses on expansion, optimizing each individual component function by applying pure weight vectors e(1), . . ., e(p). Additionally, an equal weighting of all objectives�p i=1e(i)/p is applied. | | |

ACM Trans. Math. Softw.



|  |  |  |
| --- | --- | --- |
| Algorithm XXXX: VTMOP: Solver for Blackbox Multiobjective Optimization Problems | • | 11 |

Remark 3.6. Certain components of˜W in (8) and the basis vectors e(1), . . ., e(p)contain zero values. Recall, however, that minimizing (2) is guaranteed to produce a Pareto optimal point only when w > 0. It is impossible to guarantee an exact global solution to (2) in inite time, but it is still desirable that the true solution to each surrogate problem be Pareto optimal. Therefore, after computing the weights in this section, all zero valued components are replaced by a small machine dependent łfudge factorž and the resulting weights are renormalized. The magnitude of this fudge factor is an optional input to VTMOP.

3.2 Searching the Design Space and Trust Regions

After Algorithm 1, VTMOP performs an exploration of ∆(k), which is the irst step in the multiobjective RSM framework. Depending on whether k = 0, ∆(k)could be either the entire feasible design space [L,U ] or the kth LTR, as computed in (7).

Remark 3.7. The number of cost function evaluations used during this exploration should be much larger when k = 0 than when k > 0. There are two reasons for this. First, the feasible design space could be very large, so a large number of function evaluations may be required for a thorough exploration. Second, when F is nonconvex, the convergence of VTMOP to the Pareto front (under suitable assumptions, such as Lipschitz continuity of each fi) is entirely dependent on an efective initial search. In future iterations, every function evaluation takes place within an LTR that is centered at a design point whose image is part of the current approximation to the Pareto front.

VTMOP provides two options for the search phase. The irst option is an adaptive search technique based on the algorithm DIRECT [40]. This technique is attractive because of the global convergence guarantees of DIRECT, and it often achieves the best performance with a ixed function evaluation budget. The second option is a static search technique that uses a Latin hypercube design to generate a batch of well distributed design points within the simply bounded search region. This technique does not use function values when generating the design points, thus allowing them to be evaluated concurrently. This can be advantageous in settings where (1) it is convenient to decouple the evaluation of F from VTMOP or (2) enough computing resources are available to support many concurrent evaluations of F [21].

3.2.1 Adaptive Search Using DIRECT. [32] propose using DIRECT to perform an adaptive global search. DIRECT is applied in ∆(k)at least p times per iteration, once to each component function fi (x) = (e(i))⊤F (x), with one additional DIRECT search when k = 0 using an equal weighting of all the fis. Speciically  
•

• if k = 0, use DIRECT to minimize p + 1 functions f1, . . ., fp and�p if k > 0, use DIRECT to minimize the p component functions f1, . . ., fp. i=1 pfi; and 1

This strategy drives global convergence for Lipschitz continuous fi (by the original argument of [40]), produces additional data in the neighborhood of the eicient set, and encourages VTMOP to continue expanding its coverage of the Pareto front in iterations with k > 0. The implementation of DIRECT in VTDIRECT95 is used for the adaptive search [37].

Remark 3.8. As in Remark 3.6, a fudge factor is applied to each zero valued component of e(1), . . ., e(p), and the weights are renormalized. This approach ensures that each instance of the driver subroutine VTdirect from VTDIRECT95 will converge to a Pareto optimal point in the limit.

One important detail of DIRECT is that it samples on an implicit mesh. Since p or p + 1 instances of the driver VTdirect are run in each iteration, this will surely result in redundant design point evaluations. To avoid wasting precious function evaluations, before evaluating F, VTMOP queries its internal database. If the requested design has already been or is currently being evaluated (up to the design tolerance µ), then that design is not re-evaluated, and the corresponding objective value from the database is returned.

ACM Trans. Math. Softw.



|  |  |  |
| --- | --- | --- |
| 12 | • | Chang et al. |

Remark 3.9. The number of function evaluations when using DIRECT for a small number of iterations N grows

|  |  |  |  |
| --- | --- | --- | --- |
| like | � | (1 − ω) + ω(2d + 1) | � N −1(2d + 1), |

where 0 < ω < 1 is problem dependent and typically ω ≪ 1. Because DIRECT samples on an implicit mesh, many of these evaluations will be reused between repeated runs in the same search space, so this cost grows sublinearly with the number of objectives.

3.2.2 Static Search Using Latin Hypercube Design. As a static experimental design, the Latin hypercube design is extremely common and is one of the recommendations of [53]. VTMOP uses the subroutine LATINDESIGN from the package QNSTOP [4]. Note that since LATINDESIGN is an independent auxiliary subroutine from QNSTOP, this subroutine is extracted from the larger QNSTOP package and included. Latin hypercube designs are stochastic by nature, whereas the search via DIRECT from Section 3.2.1 is deterministic. Redundant design points are an unlikely occurrence when using the Latin hypercube design. To ensure that no function evaluations are wasted, however, VTMOP still checks all design points against its internal database before evaluating.

3.3 Solving the Surrogate Optimization Problem

The kth batch of surrogate optimization problems takes place within ∆(k). Let D(k,∗)contain all x ∈ D(k)plus all the points that were evaluated during the kth search phase from Section 3.2, and let F(k,∗)=�F (x) : x ∈ D(k,∗)�. First, p surrogatesˆf1 ≈ f1, . . .,ˆfp ≈ fp are it using the data in D(k,∗)and F(k,∗). VTMOP supports the usage of a user deined surrogate, matching the provided interface. When no user surrogate is supplied, however, VTMOP uses the subroutine LSHEP from SHEPPACK [63], which implements a linear modiied Shepard method. LSHEP was chosen as the default surrogate in VTMOP based on LSHEP’s demonstrated performance as a surrogate in [31] and the recommendation of [32].

Remark 3.10. A user deined surrogate is desirable when F is not a true blackbox function and the user is able to exploit domain knowledge to design a better surrogate. For example, if the gradient is available for any fi, then this information could be used to design a higher order surrogate. One could integrate such information into the linear modiied Shepard approximation by replacing the local linear it at each data point with the true tangent. For another example, if any component fi is computationally cheap to evaluate, then surrogate modeling of fi can be łturned ofž by setting ˆfi = fi.

Remark 3.11. It is not uncommon for real-world blackbox MOPs to contain łmissing values,ž often referred to as hidden constraints in the literature [46]. These are discrete points in the feasible design space at which F is not deined or produces nonsensical outputs. When such designs are encountered, a missing value is signaled by an IEEE NAN value in the output of F. When VTMOP its the surrogates using the default model LSHEP, a wrapper function for the SHEPPACK subroutine is used to place previously evaluated missing values in a separate łLSHEP taboo list.ž Points in the taboo list are not used to it the surrogates; but if a point in the taboo list is requested while solving the surrogate optimization problem, the output of the surrogates is overwritten by a very large number. This strategy allows VTMOP to converge in the neighborhood of missing values, while explicitly forbidding convergence to a missing value.

After itting the surrogates, the kth batch of surrogate optimization problems is solved to obtain the kth batch

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| of candidate design points | C(k)= | � | arg min | � ˆf1(z), . . . , ˆfp(z) | � ˜W : ˜W ∈ W (k)� | . | (9) |
| z ∈∆(k) |

To solve (9), VTMOP allows the usage of a user deined łlocal optimizationž subroutine. By default, VTMOP uses the polling strategy Generalized Pattern Search (GPS) [7]. It should be noted that (9) involves analytic surrogates

ACM Trans. Math. Softw.



|  |  |  |
| --- | --- | --- |
| Algorithm XXXX: VTMOP: Solver for Blackbox Multiobjective Optimization Problems | • | 13 |

whose derivative information is available, but GPS is a polling strategy for blackbox optimization. The rationale for using GPS to solve (9) is as follows.

• GPS polling iterations are locally convergent even for nonsmooth functions, thus improving the robustness of VTMOP.

• When applied to analytic functions, the practical diference between irst order algorithms and blackbox (zeroth order) algorithms is that irst order algorithms typically converge at a faster rate. Solving (9) does not require any evaluations of F, however, so it is an inconsequential expense in the context of computationally expensive blackbox optimization.

The implementation of GPS in VTMOP is designed to be lightweight. In each iteration, GPS polls in every direction along an axis aligned mesh; it steps in the direction of steepest descent, and if no descent direction is found, then it decays the mesh size by a factor of two down to a minimum of µ. The process of itting the surrogates and solving (9) is summarized in Algorithm 2.

Remark 3.12. VTMOP checks the current database for each new design point before evaluating. In the return-to-caller interface described in Section 4.1, however, the user is required to evaluate F (z) for z ∈ C(k). Therefore, VTMOP does not return any redundant or previously evaluated candidate design points (i.e., values in C(k)that are redundant or equal to values in D(k,∗)up to the design space tolerance µ), which are eliminated from C(k)in a postprocessing step. In practice, this could mean that |C(k)| < |W(k)|. This is especially common when F is nonconvex since diferent weights in W(k)could lead to the same solutions in C(k).

**Algorithm 2** Solve surrogate problems to generate candidate designs.

**input**   
D(k,∗)is a database of evaluated designs, including both D(k)and additional designs that were evaluated during the kth search (Section 3.2);   
F(k,∗)=�F (x) : x ∈ D(k,∗)�;  
∆(k)is the kth LTR;   
W(k)is the kth set of adaptive weights;   
ˆf1, . . ., ˆfp are user supplied surrogates;   
µ is the design space tolerance;

**begin**   
**initialize** C(k)= ∅;   
**for** i = 1, . . ., p **do**   
 Fit ˆfi ≈ fi using D(k,∗)and F(k,∗);   
**enddo**   
**for**˜W ∈ W(k)**do**  
 ˜z ← arg min� ˆf1(z), . . . , ˆfp(z)  
� ˜W ; z ∈∆(k)   
 **if** ∥ ˜z − x∥2 > µ for all x ∈ D(k,∗)**and** ∥ ˜z − z∥2 > µ for all z ∈ C(k)**then**  C(k)← C(k)∪ {˜z};   
 **endif**   
**enddo**   
**return** C(k)

The time complexity of Algorithm 2 is dependent on the cost for itting the surrogates, the cost for evaluating the surrogates, and the number of surrogate evaluations. By default, VTMOP performs 2,500 iterations of GPS

ACM Trans. Math. Softw.



|  |  |  |
| --- | --- | --- |
| 14 | • | Chang et al. |

polling (each requiring 2d evaluations of the surrogate) per˜W ∈ W(k). In practice, this could require less than a second to several minutes on modern hardware, depending on the size of W(k), the value of d, and the complexity of the surrogate (LSHEP’s evaluation complexity grows sublinearly with n(k)). The surrogate problems can also be solved in parallel to reduce the wallclock time of Algorithm 2. In the package VTMOP, the subroutine VTMOP\_OPT implements Algorithm 2.

3.4 Evaluating Candidate Designs and Iterating

The inal step in the kth iteration is to evaluate all z ∈ C(k)with F. After all evaluations have inished, the (k +1)st databases D(k+1)and F(k+1)have been completed. In the next iteration, VTMOP\_LTR will compute P(k+1) and E(k+1), so there is no need to do so in this stage unless a termination condition has been met. The iteration counter k is incremented at this point.

VTMOP has three potential termination conditions. The irst condition is that the budget for evaluations of F has been spent. If this limit is reached mid-iteration, the rest of the iteration aborts. The second condition is that the iteration limit has been exceeded. This condition is checked before the beginning of each iteration. The third condition is that the lowest index point in the inverse image of every point in the current approximation to the nondominated set has been used as the center of a previous LTR, whose trust region radius fraction has been decayed below the tolerance. This condition is checked by VTMOP\_LTR, as described in Algorithm 1.

When any termination condition occurs at iteration k, the current database of evaluated design points and corresponding function values is used to compute the inal nondominated and approximately eicient point sets, which are returned with the termination indicator.

Remark 3.13. Unlike in Section 3.1, upon termination VTMOP returns every observed objective vector that is nondominated and its corresponding eicient design. In particular, these may include duplicate nondominated points. In the literature, these are referred to as the (approximately) weakly nondominated and eicient sets.

4 IMPLEMENTATION AND INTERFACES

VTMOP ofers two interfaces for solving blackbox MOPs based on the framework laid out in Section 3. The irst is a return-to-caller interface, and the second is the driver subroutine VTMOP\_SOLVE. In many situations, F can easily be wrapped in a Fortran subroutine, and the driver subroutine is preferred because of its ease of use. However, the return-to-caller interface ofers a lexible alternative for situations where the computing environment makes wrapping F in a Fortran subroutine diicult or ineicient. Both interfaces are implemented in ISO Fortran 2008 and support some forms of OpenMP parallelism.

4.1 Return-to-Caller Interface

In many real-world blackbox optimization problems, special purpose computing environments [58] and libraries for coordinating the use of parallel resource [21] require the evaluation of F to be decoupled from the optimization algorithm. In these situations, it is convenient to ofer a return-to-caller interface, where F is evaluated only from outside of any Fortran subroutine’s call stack.

Here it is necessary to preserve the entire state of the algorithm VTMOP during Steps 2 and 4 (Section 3), where F is evaluated in batches (presumably in parallel). The state is recorded in a Fortran derived data type VTMOP\_TYPE, created by the subroutine VTMOP\_INIT, and stored between invocations of components of VTMOP. A user would call VTMOP\_INIT to create a VTMOP\_TYPE, then iterate through steps 1ś4 from the beginning of Section 3, calling VTMOP\_LTR for Step 1, VTMOP\_OPT for Step 3, and performing the evaluations of F in Steps 2 and 4 using their preferred methods/technologies. Further details on this process are outlined in the USERS document, which is included with the source code. An example implementation of this process is described in [21].

ACM Trans. Math. Softw.



|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 4.2 | The Driver Subroutine | Algorithm XXXX: VTMOP: Solver for Blackbox Multiobjective Optimization Problems | • | 15 |

The driver subroutine VTMOP\_SOLVE implements the same process as in Section 4.1 but without reading and writing the state and using a Fortran implementation of F. The search phase is performed either via DIRECT as in Section 3.2.1 or via Latin hypercube design as in Section 3.2.2, with the choice being speciied as an optional input. For a search via DIRECT, the łsearch budgetž inputs are used to specify the iteration limit for VTDIRECT95. For a search via Latin hypercube design, the search budgets are used to specify the number of points in the design.

In addition, VTMOP\_SOLVE ofers a checkpointing system and a parallel option described in Section 4.3. For further details, see the USERS document, included with the source code.

4.3 Parallel Implementation

Two opportunities for parallelism are ofered in VTMOP. First, the iteration tasks in the worker subroutines VTMOP\_LTR and VTMOP\_OPT can be parallelized. This includes computing DG(Π(k)) in parallel using the parallel DELAUNAYSPARSEP driver from [22], itting the p surrogates in parallel, and solving the |W(k)| surrogate problems (9) in parallel. This iteration level parallelism is provided via OpenMP 4.5 [17] and is available through either the return-to-caller interface or the driver subroutine.

The second source of parallelism is concurrently evaluating F at points requested by VTMOP. In the context of blackbox optimization, this is the more impactful source of parallelism. Since the return-to-caller interface does not evaluate F, this form of parallelism is available only through the driver subroutine (which ofers a choice between iteration task parallelism, concurrent function evaluations, or both). Recall that VTMOP requires function evaluations only while searching ∆(k)as described in Section 3.2 and when evaluating candidate designs from C(k)as described in Section 3.4. In both cases, concurrent evaluations are achieved by spawning a new OpenMP task for each evaluation of F. Note that because the actual function evaluations are handled by the user, the user is also responsible for capping the number of actual function evaluations in order to prevent the oversubscription of resources.

Remark 4.1. Within a single evaluation of F, there may be ample opportunities for parallelism, which is problem dependent and left to the user. In particular, in many real-world applications, each evaluation of F could depend on output from a multinode, distributed simulation. VTMOP does not distribute calls to F, but it does provide the ability for concurrent evaluations of F via OpenMP task-based parallelism. This places the burden of distributing calls to F on the user, but it allows for greater lexibility.

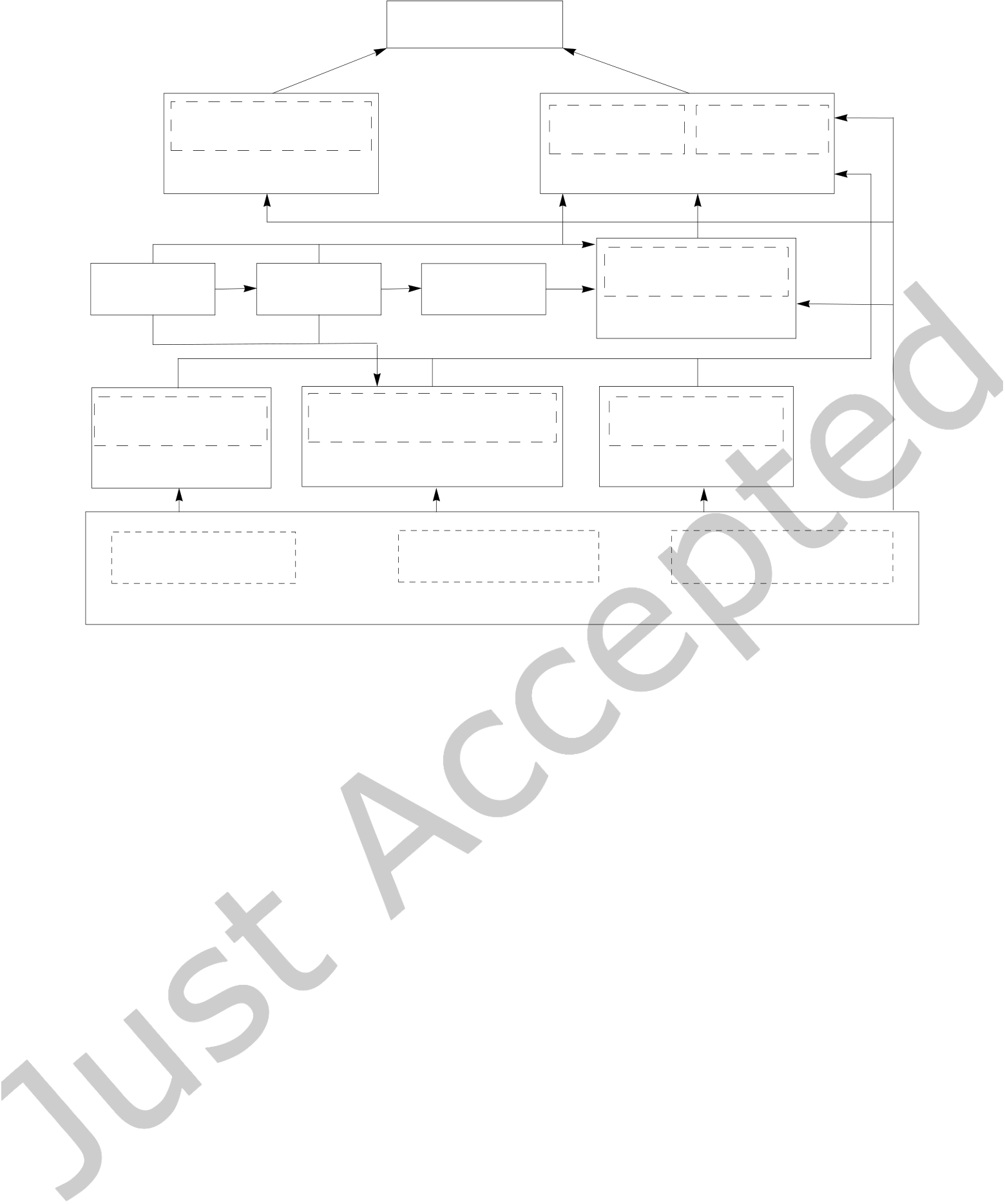
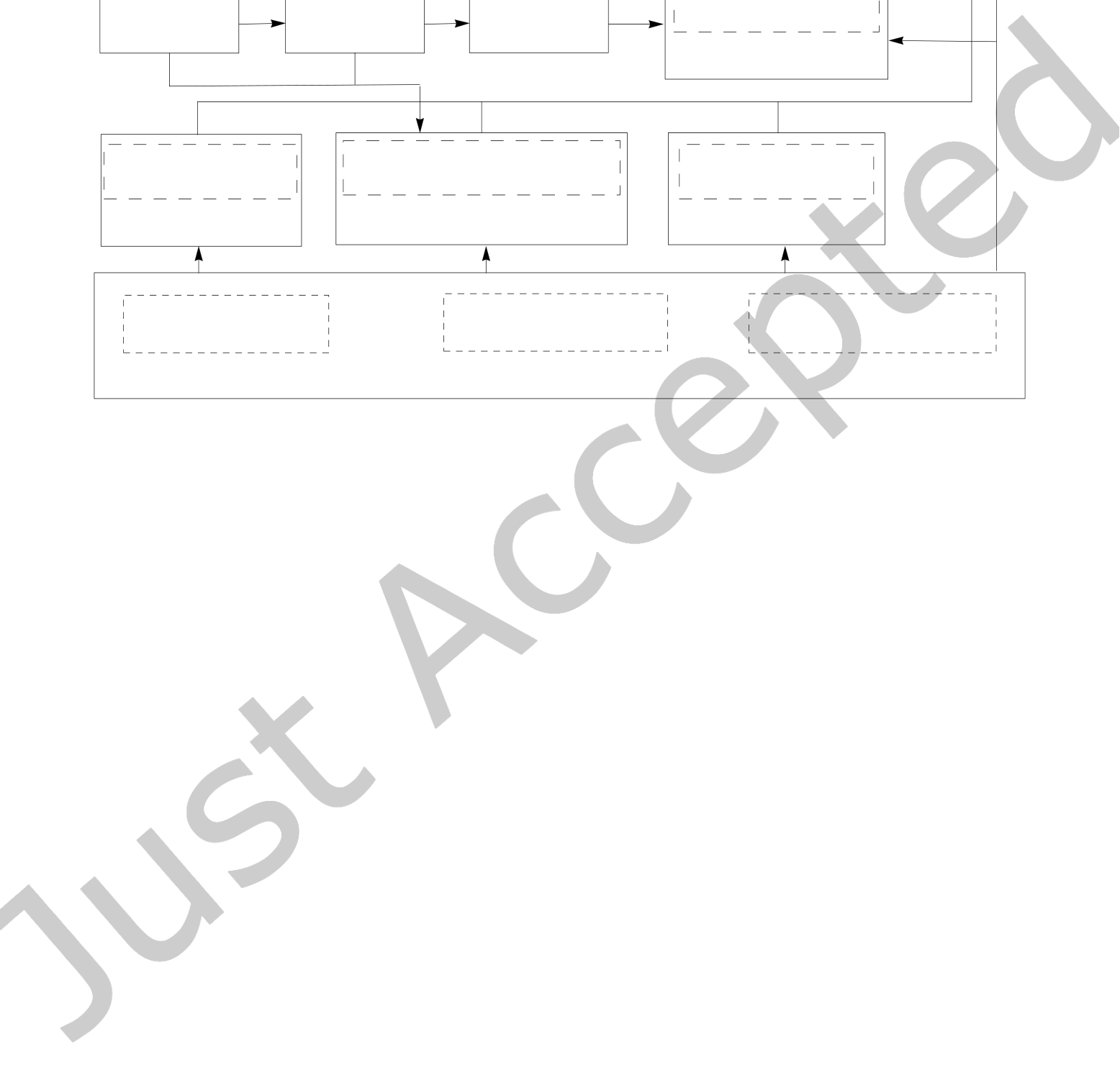
When evaluating C(k)as in Section 3.4 and when using the Latin hypercube design during the search phase, all candidates/design points can be evaluated in parallel if enough computing resources are available. For a search via DIRECT, the situation is more complicated since each batch of potentially optimal boxes in a DIRECT iteration can be divided in parallel and each instance of VTdirect can be run asynchronously. VTDIRECT95 ofers a parallel driver pVTdirect for distributing the division of boxes, but the distributed memory paradigm is not appropriate for this use case. Instead, a slightly modiied implementation of VTdirect, called bVTdirect, is provided, which uses OpenMP tasks to perform concurrent evaluations of box centers. During the kth search, VTMOP makes either p + 1 (if k = 0) or p (if k > 0) asynchronous calls to bVTdirect, resulting in nested parallelism.

Further details on the parallel implementation, such as choosing the number of threads and setting up OpenMP environment variables, are included in the USERS document. A detailed analysis of VTMOP’s parallel performance for computationally expensive functions is given in [21].

5 SUMMARY OF USAGE AND PARAMETERS

The package VTMOP is organized as follows. The ile vtmop.f90 contains implementations for the majority of algorithms described in this ile, including VTMOP\_LTR (Algorithm 1), VTMOP\_OPT (Algorithm 2), DELAUNAYGRAPH

ACM Trans. Math. Softw.



|  |  |  |
| --- | --- | --- |
| 16 | • | Chang et al. |

sample.f90

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| blas.f | VTMOP\_FUNC\_MOD | slatec.f | VTMOP\_MOD | VTMOP\_LIB |
| vtmop\_func.f90 | vtmop.f90 | |
| lapack.f | DELSPARSE\_MOD | |

delsparse.f90

|  |  |  |
| --- | --- | --- |
| {b|s}VTdirect\_MOD | LINEAR\_SHEPARD\_MOD | QNSTOPS\_MOD |
| {b|s}VTdirect.f90 | linear\_shepard.f90 | qnstop.f90 |
| REAL\_PRECISION | VTDIRECT\_GLOBAL | VTDIRECT\_COMMSUB |

shared\_modules.f90

Fig. 2. Dependency graph for VTMOP. Files are outlined in solid lines, and modules are outlined in dashed lines. An arrow from File 1 to File 2 indicates that File 2 has a direct dependency on a subroutine or module from File 1.

(Algorithm 3), and VTMOP\_SOLVE (the driver). Additional iles include vtmop\_func.f90 (implementations of test problems from Section 6); samples.f90 and samplep.f90 (examples of serial and parallel usage); sVTdirect.f90 and bVTdirect.f90 (serial and modiied parallel VTDIRECT95 drivers); shared\_modules.f90 (important shared types and modules); qnstop.f90, linear\_shepard.f90, and delsparse.f90 (minimal copies of dependencies); and blas.f, lapack.f, and slatec.f (minimal copies of common Fortran libraries). Additional usage and organization information is provided in the USAGE document.

Remark 5.1. The return-to-caller interface has numerous inputs, outputs, and parameters (many optional) when initializing the data type VTMOP\_TYPE using VTMOP\_INIT. Most of these have been discussed in previous sections and remarks, but it is important to understand all of these settings, so they are summarized here. The required inputs are the problem dimensions (d and p) and the bound constraints (L and U ). It is impossible to use VTMOP (any interface) without declaring these values. The optional inputs have been carefully assigned default values based on usability and/or trial-and-error experimentation on a wide variety of real-world problems and test problems from the literature. These optional inputs (and their default values) are design space tolerance µ and objective space tolerance ϵ (by default, the square root of the working precision, deined below); the initial trust region radius ρ(0)as a fraction of U − L, the decay rate τ, and the minimum trust region radius ρ(1)as a fraction of U − L (by default, 20% of each dimension of the feasible design space, 0.5, and 10% of the initial trust region radius fraction); the working precision of the machine (by default, the square root of the unit round-of); the fudge factor for zero weights (by default, the fourth root of the unit round-of); the procedures for performing local optimization and the local optimizer iteration budget (by default, GPS with a budget of 2, 500 iterations); the procedures for itting and evaluating the surrogates (by default, wrappers for the corresponding linear Shepard’s method routines from SHEPPACK); lower and upper bounds on the interesting objective ranges (by default,

ACM Trans. Math. Softw.



|  |  |  |
| --- | --- | --- |
| Algorithm XXXX: VTMOP: Solver for Blackbox Multiobjective Optimization Problems | • | 17 |

there are no bounds); a Boolean value specifying whether to perform parallel iteration tasks (by default, false); and the checkpointing mode (i.e., no checkpointing, use checkpointing, or recover from checkpoint, with no checkpointing by default). Every VTMOP subroutine also returns an integer valued error lag, further detailed in VTMOP’s documentation.

Remark 5.2. The driver VTMOP\_SOLVE accepts some variation of all of the parameters, inputs, and outputs described in Remark 5.1. Additional inputs and outputs include a Fortran implementation of the objective function F, several arrays for returning both the solution sets (approximately Pareto pairs) and full objective function databases, an optional list of precomputed function values, and the budget for the maximum number of blackbox function evaluations. The most notable additional parameter, which will be used in Section 6, is used to specify whether the search via DIRECT should be used (referred to as an adaptive search in the documentation), as opposed to the Latin hypercube search (by default, the search via DIRECT is used). Another optional parameter speciies the search budgets that are used when k = 0 and k > 0. When using the search via DIRECT, the default values are 10 iterations of DIRECT sampling and 5 iterations of DIRECT sampling, respectively. Otherwise, for a Latin hypercube search, the default values are 16d2and 8d, respectively.

Remark 5.3. The cost of each search phase can get out of hand if one is not mindful of the interplay between the search budgets and d when using the search via DIRECT. Based on the costs described in Remark 3.9, for large d the Latin hypercube or some incomplete orthogonal design is the better choice.

6 PERFORMANCE ON TEST PROBLEMS

The original algorithm of [32] used a search based on DIRECT. Their approach was previously compared against BiMADS from NOMAD, using a budget of only 200 function evaluations [32]. It was also compared against a private implementation of MultiMADS [9] with a budget of 30,000 evaluations of F (as recommended by [30]) for problems with up to six objectives and 36 design variables. In this section, two variations of VTMOP (one using DIRECT for the search and one using a Latin hypercube search) are compared against two widely used open source multiobjective solvers. In particular, the two VTMOP variations are compared against the NSGA-II [29] implementation in PyMOO [15] and an implementation of the direct search technique MultiGLODS [27] from the BoostDFO MATLAB toolbox [61]. Recall that NSGA-II/PyMOO is representative of evolutionary MOA performance, and MultiGLODS is a global multiobjective direct search technique.

Unless otherwise speciied, in the remainder of this section, VTMOP is run serially with its default parameter values (speciied in Remarks 5.1 and 5.2), using DIRECT (hereafter, referred to as the łDIR variantž) and Latin hypercube search (hereafter, referred to as the łLH variantž). NSGA-II is run with PyMOO’s default settings: a population size of 100, random sampling for the initial population, a binary tournament selection scheme, simulated binary crossover with probability 0.9, polynomial mutation, and eliminating duplicate values from each population. MultiGLODS is run with mostly default parameters, most notably, using a global search based on the low-discrepancy Sobol sequence where the number of samples depends on the problem dimension, and evaluating all poll directions in every iteration. One change that was made to the default settings of MultiGLODS was to enable caching of previous function evaluations with a tolerance of 10−8. Note that the DIR variant of VTMOP and the settings for MultiGLODS that use a Sobol sequence search are both deterministic, while the LH variant of VTMOP and NSGA-II are both nondeterministic. Therefore, in all future subsections, the performances of the LH variant and NSGA-II are averaged over 5 repeated runs.

Evaluation of how well points evaluated by a MOA approximate the true Pareto front is an open problem [6].

In Sections 6.1 and 6.2, three criteria to measure the quality of an approximate Pareto surface are considered:

• the number of nondominated solution points identiied,

• the distance between points on the approximate and true Pareto front, and

ACM Trans. Math. Softw.



18 • Chang et al.

• the degree to which those points are spread evenly across the entire Pareto front.

The irst of these criteria is measured by returning the cardinality of the solution set. The second criteria is measured using the root mean squared error (RMSE) between the points in the solution set and their nearest corresponding points on the true Pareto front. This metric can be easily computed only for łnicež test functions, such as those in this section, where the Pareto front can be expressed algebraically. The third criteria is measured using the discrepancy, which is a statistical technique for measuring the uniformity of a sequence of points within some underlying convex space [42, 44]. In this section, the discrepancy of the solution set on the Pareto front is approximated by using the Delaunay triangulation of its projection, as in (5). This measure for spread is fully described in [32] and hereafter referred to as the Delaunay discrepancy. As a discrepancy approximation, the Delaunay discrepancy approaches zero for asymptotically uniformly distributed sequences. Computing the Delaunay discrepancy requires the complete Delaunay triangulation of the projected set, which is obtained using the Quickhull [10] implementation in SciPy.

6.1 A Convex Problem

First consider the construction of F(c), a componentwise convex multiobjective function that is meant to capture the performance of MOAs on łnicely behavedž problems. Convex functions are expected to produce easier problems for adaptive weighting schemes since every point on the Pareto front can be achieved by minimizing a weighted sum of objectives as in (2). Let d > p, let e(i)denote the ith standard basis vector in Rd, and let e = [1, . . ., 1]⊤. Then   
 F(c)(x) =�∥x − 0.5e(1)− 0.1e∥2 2, . . . , ∥x − 0.5e(p) − 0.1e∥2� , x ∈ [0, 1]d.

The Pareto front for F(c)is a portion of a rotated parabola in Rp.

The two VTMOP variations are compared against NSGA-II and MultiGLODS on F(c)with problem sizes p = 3,d = 8 and p = 4,d = 14. Performance results for all three metrics are shown in Figures 3 and 4. Note that the number of solution points is shown on a log scale so that the results for the smaller budgets can be more easily observed.

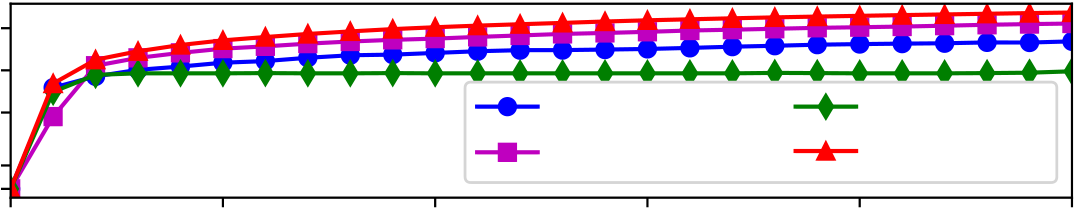
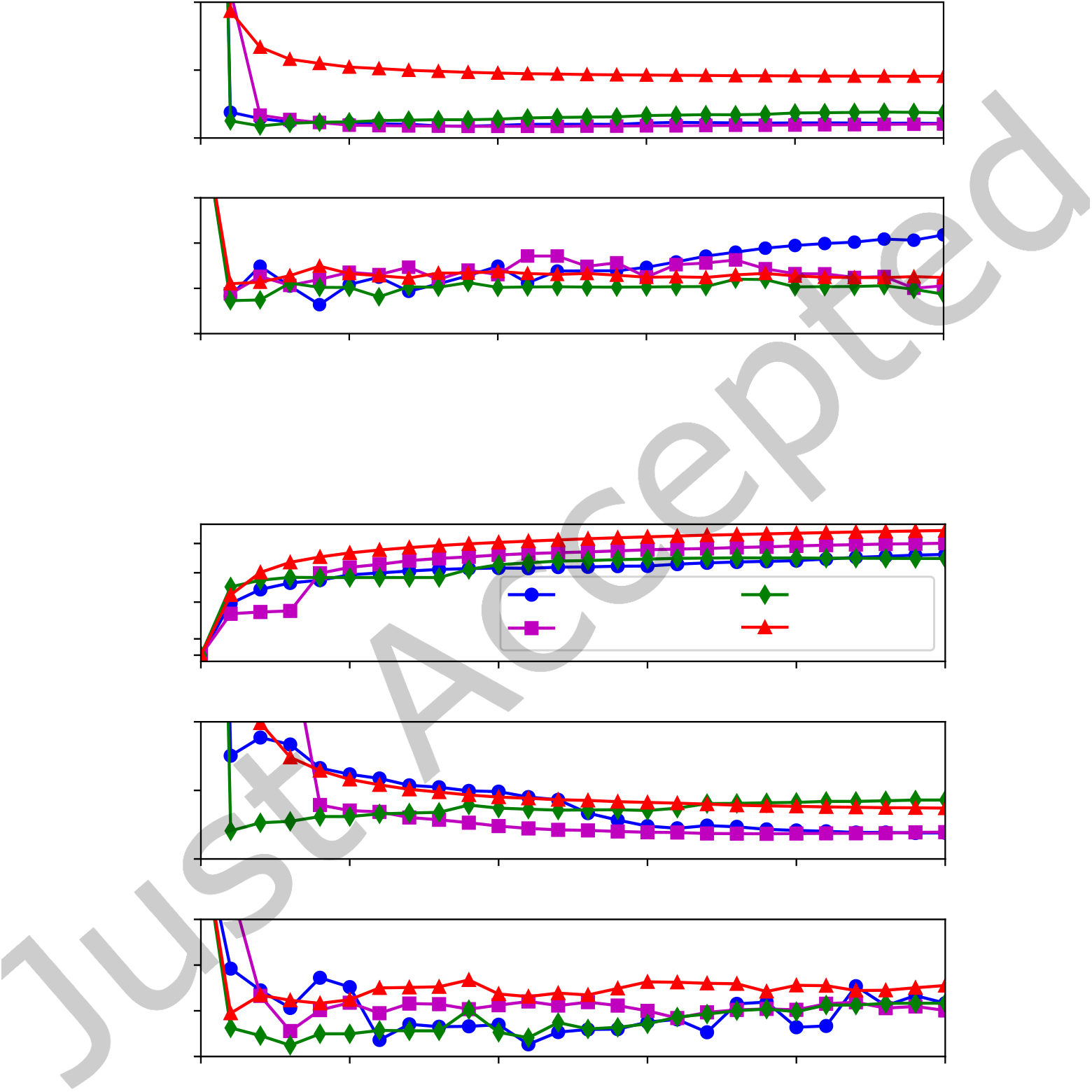
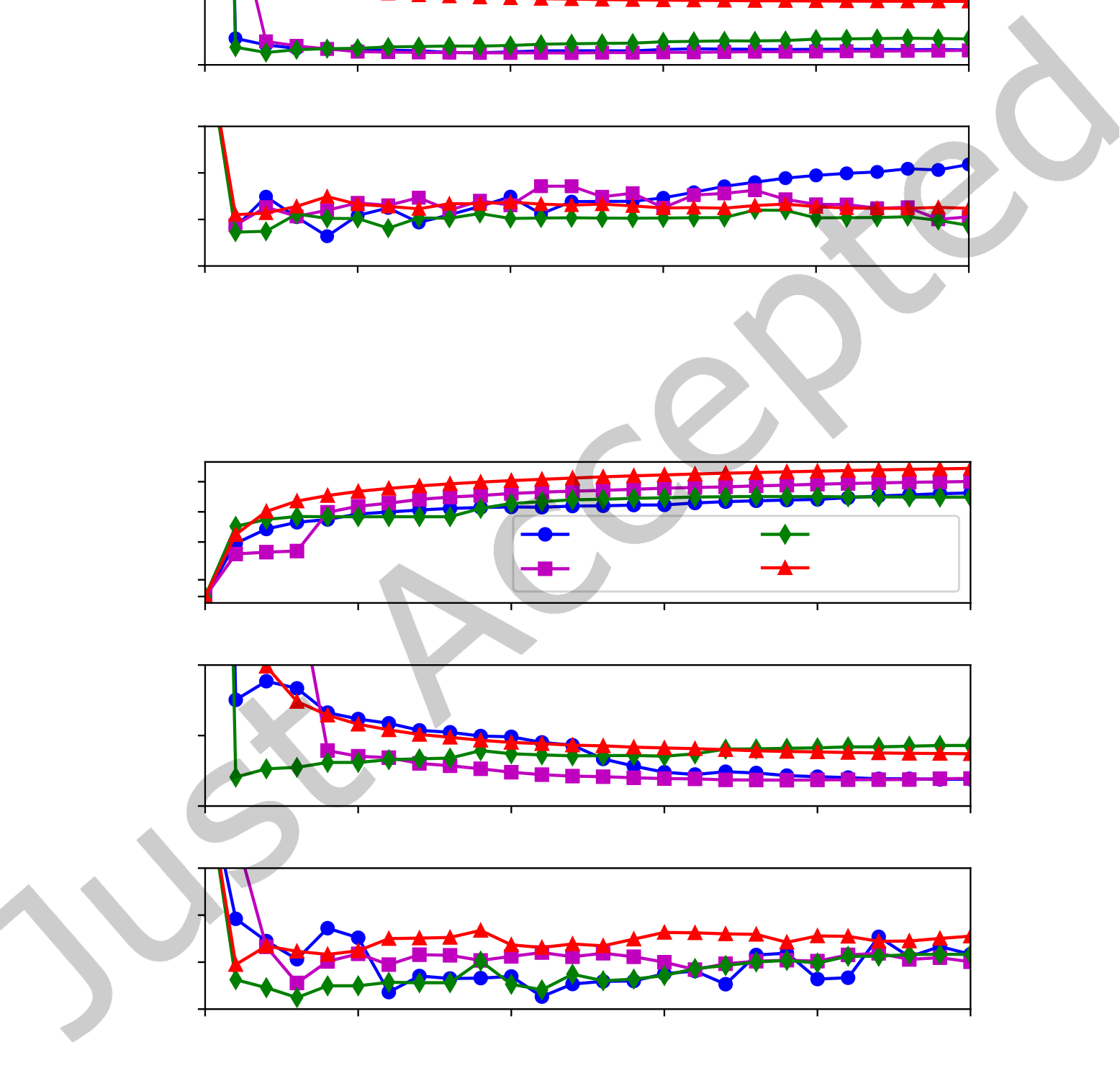
As seen in Figures 3 and 4, for the problem F(c), all MOAs considered are able to ind tens or even hundreds of approximate solutions with their RMSE less than 0.1 and Delaunay discrepancies of around 0.2 or less within the irst 5,000 function evaluations at both problem sizes. Note that it is dangerous to interpret any of the performance measures independently, as none of them improve monotonically. Rather, one might consider whether a large number of solutions with low RMSE and low discrepancy have been observed thus far in each MOA’s history, which would indicate that a reasonably dense, well-distributed set of low-error solutions is contained in every subsequent solution set. However, it is interesting to note that both the DIR variant and MultiGLODS tend to produce several low RMSE solutions particularly early on, while NSGA-II and the LH variant tend to produce more solutions in the long term. Also note that the performance statistics are somewhat smoothed for the two stochastic MOAs (LH variant and NSGA-II) due to the averaging efect.

6.2 The DTLZ Problems

The DTLZ test suite [30] is a library of scalable multiobjective test problems that is widely used in the multiobjective optimization literature. Each of the DTLZ problems has a speciic property, which makes it uniquely challenging for MOAs. Additionally, each of the DTLZ problems can be solved with any number of objectives and design variables satisfying d > p. In this section, the variations of VTMOP are compared against NSGA-II and MultiGLODS on slight variations of 4 problems from the DTLZ suite, speciically, DTLZ1, DTLZ2, DTLZ5, and DTLZ7.

One issue with applying the DIR variant directly to the DTLZ problems is that all of the DTLZ problems have their eicient set either located in best or worst case locations for DIRECT’s deterministic sampling scheme. In particular, for the DTLZ problems, the feasible design space is always the d-dimensional unit hypercube,

ACM Trans. Math. Softw.



|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Num. pts | 103 | 0 | Algorithm XXXX: VTMOP: Solver for Blackbox Multiobjective Optimization Problems | | | | | • | 19 |
| 102 | VTMOP/DIR | | | MultiGLODS | |
| 101 |
| 100 0 | VTMOP/LH | | | NSGA-II | |
| 5000 | 10000 | 15000 | 20000 | 25000 |

0.10

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| RMSE | 0.05 | | 0 | 5000 | 10000 | 15000 | 20000 | 25000 |
| 0.00 | | |
| Del. discrep. | | 0.6 0.4 0.2 0.0 | 0 | 5000 | 10000 | 15000 | 20000 | 25000 |

Number of *F* evals

Fig. 3. Performance of MOAs on F(c)for p = 3, d = 8.

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Num. pts | 103 | 0 | 5000 | 10000 | VTMOP/DIR | MultiGLODS | |
| 102 |
| 101 |
| 100 0 | VTMOP/LH | NSGA-II | |
| 15000 | 20000 | 25000 |

0.10

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| RMSE | 0.05 | | 0 | 5000 | 10000 | 15000 | 20000 | 25000 |
| 0.00 | | |
| Del. discrep. | | 0.6 0.4 0.2 0.0 | 0 | 5000 | 10000 | 15000 | 20000 | 25000 |

Number of *F* evals

Fig. 4. Performance of MOAs on F(c)for p = 4, d = 14.

ACM Trans. Math. Softw.



|  |  |  |
| --- | --- | --- |
| 20 | • | Chang et al. |

and for DTLZ1, DTLZ2, and DTLZ5, every eicient x satisies xp = 0.5, . . . ,xd = 0.5 (best case for DIRECT), while every eicient x for DTLZ7 satisies xp = 0, . . . ,xd = 0 (worst case for DIRECT). Because of the way DIRECT samples, numerous design points x satisfying xp = 0.5, . . . ,xd = 0.5 will be sampled early, while design points satisfying xp = 0, . . . ,xd = 0 will be sampled very late. Therefore, in order to maintain a fair comparison, this section uses variations of the DTLZ problems that have been modiied so that the eicient set lies in the hyperplane xp = 0.6, . . . ,xd = 0.6. In most cases, the modiication required to achieve this is somewhat obvious. The modiied DTLZ problems, along with F(c), are implemented in vtmop\_func.f90.

The recommended number of design variables for the DTLZ problems range from 5 to 10 more design variables than objectives. The problem sizes of p = 3,d = 8 and p = 4,d = 14 that were used in Section 6.1 are used again in this section, as they capture the extremes of these recommendations. Since the DTLZ problems are signiicantly more diicult than F(c), it is not uncommon for MOAs to return solutions that are hugely suboptimal. When the łoptimality gapž in a solution point (here quantiied by the distance from the solution point to the true Pareto front) is large enough, it can skew performance metrics in a way that is undesirable. Therefore, in this section, only solutions with a distance from the Pareto front of less than 0.1 are considered when computing the performance metrics.

6.2.1 DTLZ1 extremely challenging, is that it features 11d−p+1− 1 łlocal Pareto fronts,ž i.e., surfaces of points that are DTLZ1. DTLZ1 has a planar Pareto front, given by all X ∈ Rpthat satisfy�p i=1Xi = 0.5. What makes

only locally Pareto optimal.

Because of the diiculty of this problem, none of the techniques tested were able to consistently achieve any solutions with a distance from the Pareto front of less than 0.1 for the larger problem size (p = 4,d = 14), and only the DIR variant was able to achieve such solutions for the smaller problem size (p = 3,d = 8). However, even at the smaller problem size, the DIR variant wasted many iterations due to the fact that Algorithm 1 regularly identiies solutions with an optimality gap greater than 0.1 as the łmost isolated point,ž causing VTMOP to waste many iterations reining in the neighborhood of suboptimal solutions. Therefore, a second run of DIR was performed, specifying łinteresting pointsž (see Remark 3.1) as only those X ∈ Rpthat satisfy X<= 0.6e.

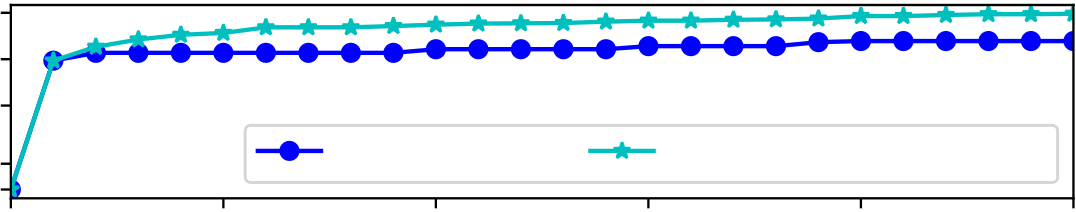
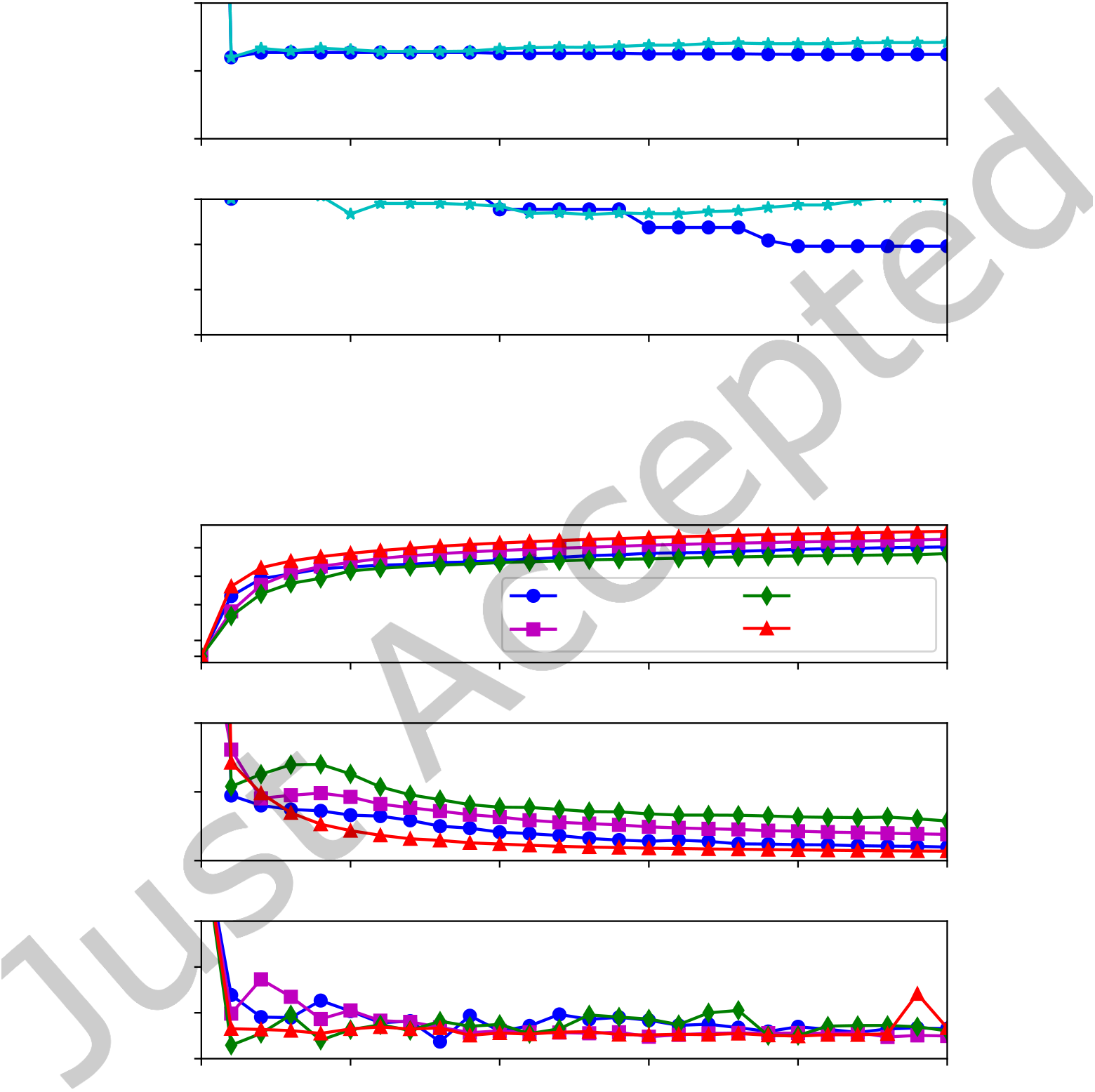
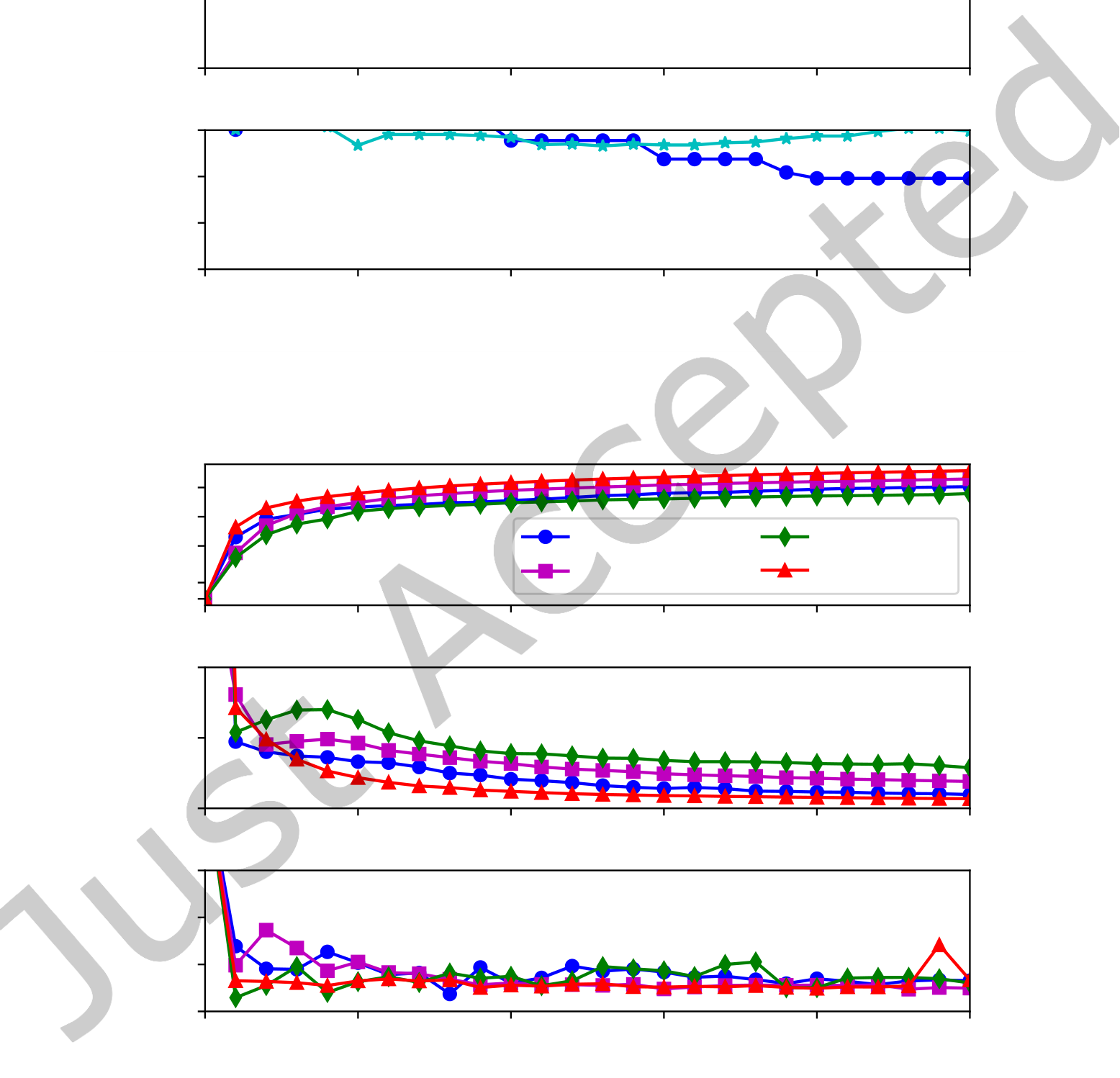
Figure 5 shows the results for both runs of the DIR variant on the smaller problem size p = 3, d = 8 (with and without constraints on the region of interest).

The results in Figure 5 appear inconclusive since VTMOP achieves a lower RMSE and discrepancy without the objective bounds, but provides more solutions with the objective bounds. Looking at the raw data, it is apparent that VTMOP actually performs signiicantly better with the objective bounds, however, many of the additional solutions found are clustered and have slightly higher errors, artiicially driving up these metrics. One notable observation is that both runs of VTMOP achieved over 100 solutions with RMSE less than 0.07 within the irst 2,000 function evaluations.

6.2.2 DTLZ2. DTLZ2 has a spherical Pareto front, given by the portion of the unit sphere that is in the positive orthant. DTLZ2 does not have any points that are locally Pareto optimal that are not globally Pareto optimal; this makes it an easier problem than DTLZ1. However, since the Pareto front for DTLZ2 is entirely concave, it is a particularly challenging problem for adaptive weighting schemes since only the łpure solutionsž (corresponding to basis vectors in Rp) can be obtained as solutions to a weighted sum scalarization, as in (2). The results for solving DTLZ2 with problem sizes p = 3,d = 8 and p = 4,d = 14 are shown in Figures 6 and 7, respectively.

Similarly as with F(c), all of the MOAs perform well on DTLZ2 at the problem size p = 3, d = 8. However, in comparison to the results for F(c)in Figure 3, Figure 6 shows fewer solution points found, but those points are generally better distributed on the Pareto front. The fact that fewer solution points are found by VTMOP is somewhat unsurprising, due to the fact that DTLZ2 has a nonconvex Pareto front, which causes diferent scalarizations to produce the same solution points. However, it may be surprising that VTMOP is not any

ACM Trans. Math. Softw.



|  |  |  |
| --- | --- | --- |
| Algorithm XXXX: VTMOP: Solver for Blackbox Multiobjective Optimization Problems | • | 21 |

103

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Num. pts | 102 | 0 | 5000 | VTMOP/DIR | VTMOP/DIR w. constraint | | |
| 101 |
| 100 0 |
| 10000 | 15000 | 20000 | 25000 |

0.10

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| RMSE | 0.05 | | 0 | 5000 | 10000 | 15000 | 20000 | 25000 |
| 0.00 | | |
| Del. discrep. | | 0.6 0.4 0.2 0.0 | 0 | 5000 | 10000 | 15000 | 20000 | 25000 |

Number of *F* evals

Fig. 5. Performance of VTMOP DIR variant (with and without constraints) on DTLZ1 for p = 3, d = 8.

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Num. pts | 103 | 0 | 5000 | 10000 | VTMOP/DIR | MultiGLODS | |
| 102 |
| 101 |
| 100 0 | VTMOP/LH | NSGA-II | |
| 15000 | 20000 | 25000 |

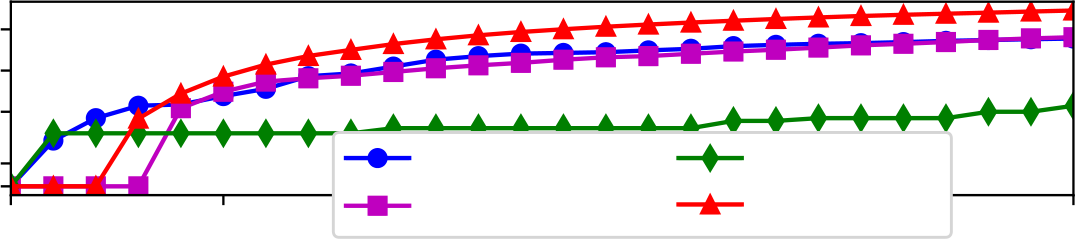
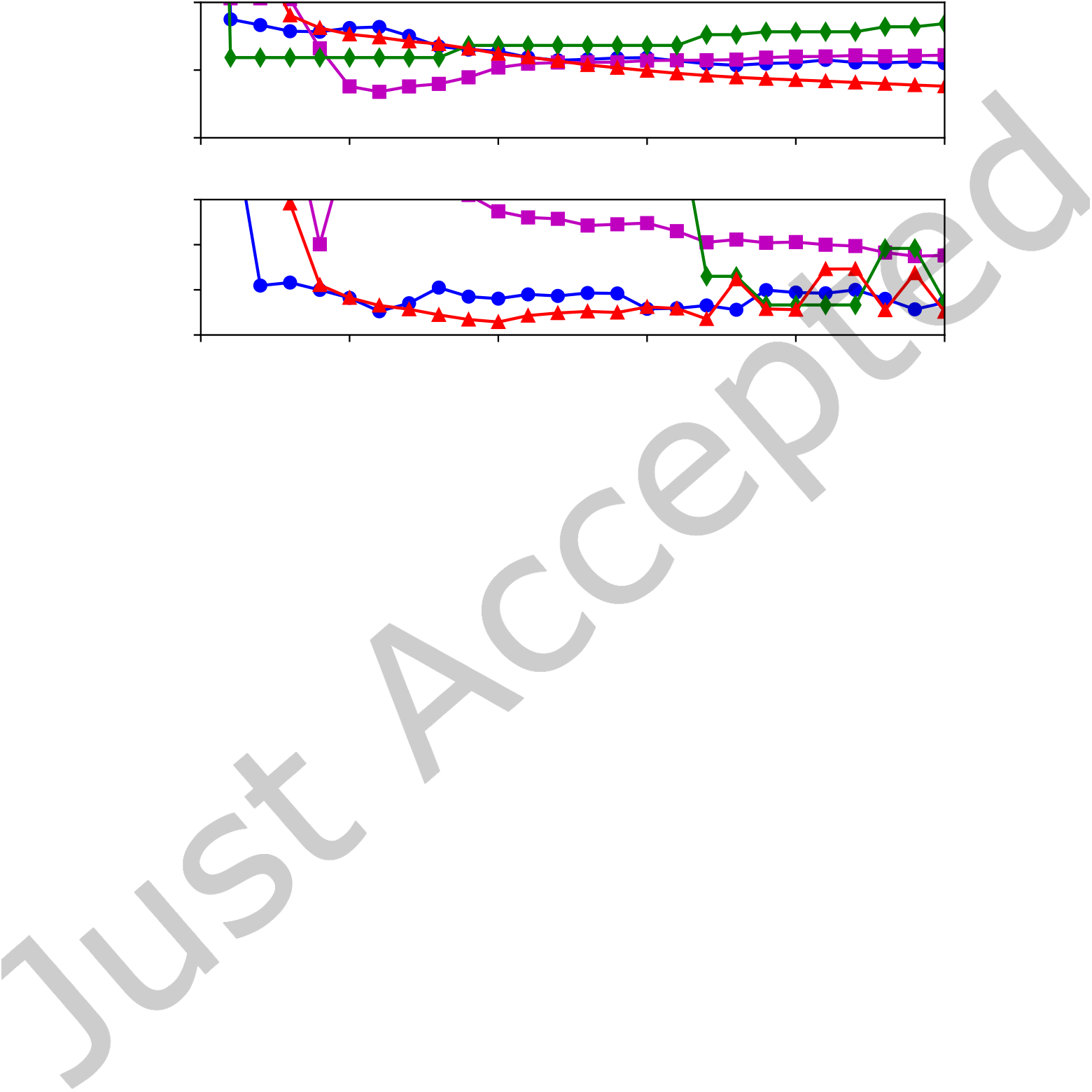
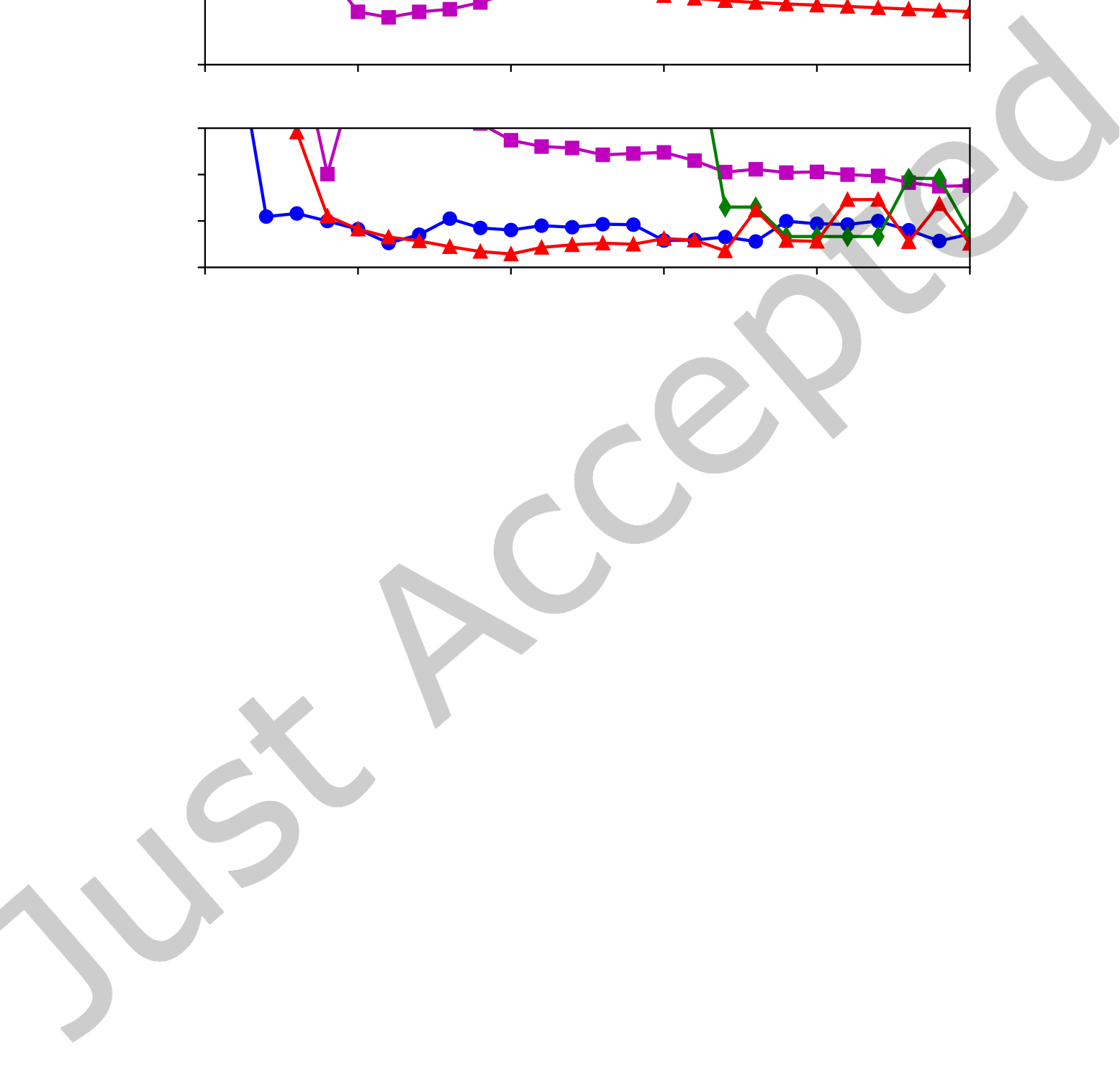
0.10

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| RMSE | 0.05 | | 0 | 5000 | 10000 | 15000 | 20000 | 25000 |
| 0.00 | | |
| Del. discrep. | | 0.6 0.4 0.2 0.0 | 0 | 5000 | 10000 | 15000 | 20000 | 25000 |

Number of *F* evals

Fig. 6. Performance of MOAs on DTLZ2 for p = 3, d = 8.

ACM Trans. Math. Softw.



|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 22 | • | Chang et al. | Num. pts | 103 | 0 | 5000 | VTMOP/DIR | MultiGLODS | 25000 |
| 102 |
| 101 |
| 100 0 |
| 10000 VTMOP/LH 15000 | NSGA-II 20000 |

0.10

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| RMSE | 0.05 | | 0 | 5000 | 10000 | 15000 | 20000 | 25000 |
| 0.00 | | |
| Del. discrep. | | 0.6 0.4 0.2 0.0 | 0 | 5000 | 10000 | 15000 | 20000 | 25000 |

Number of *F* evals

Fig. 7. Performance of MOAs on DTLZ2 for p = 4, d = 14.

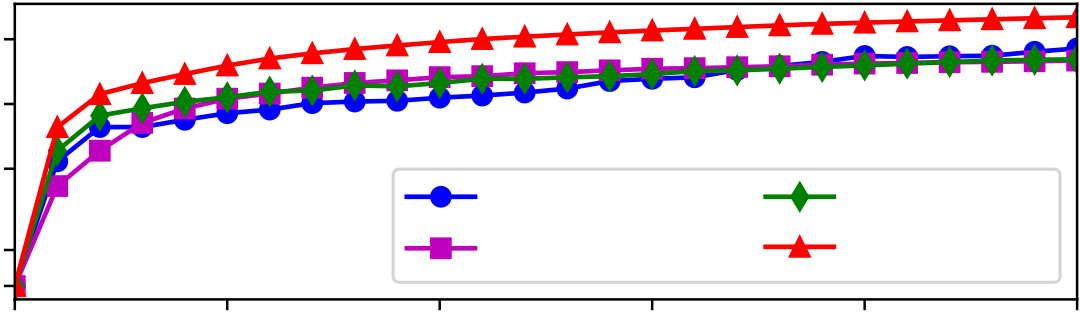
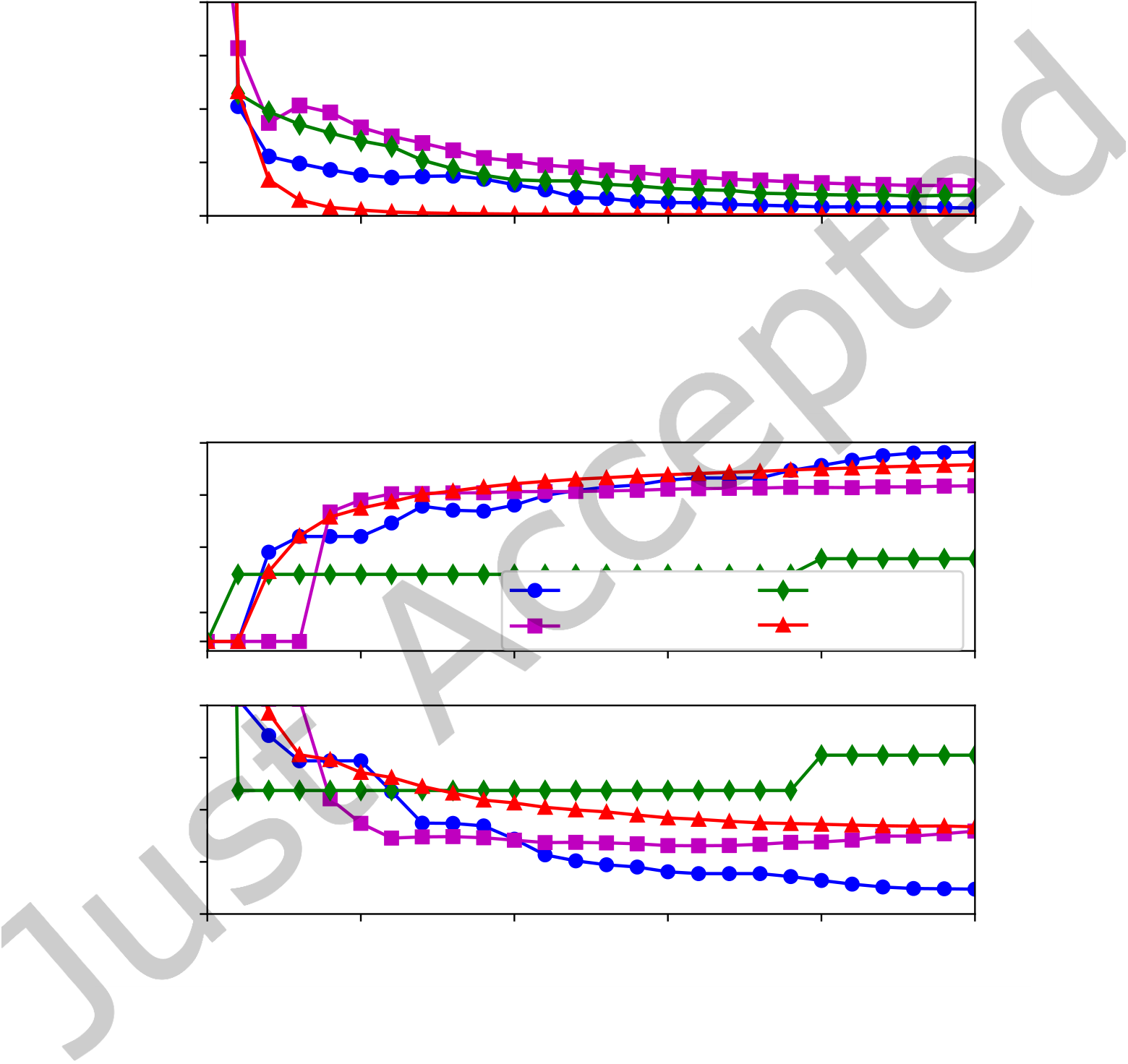
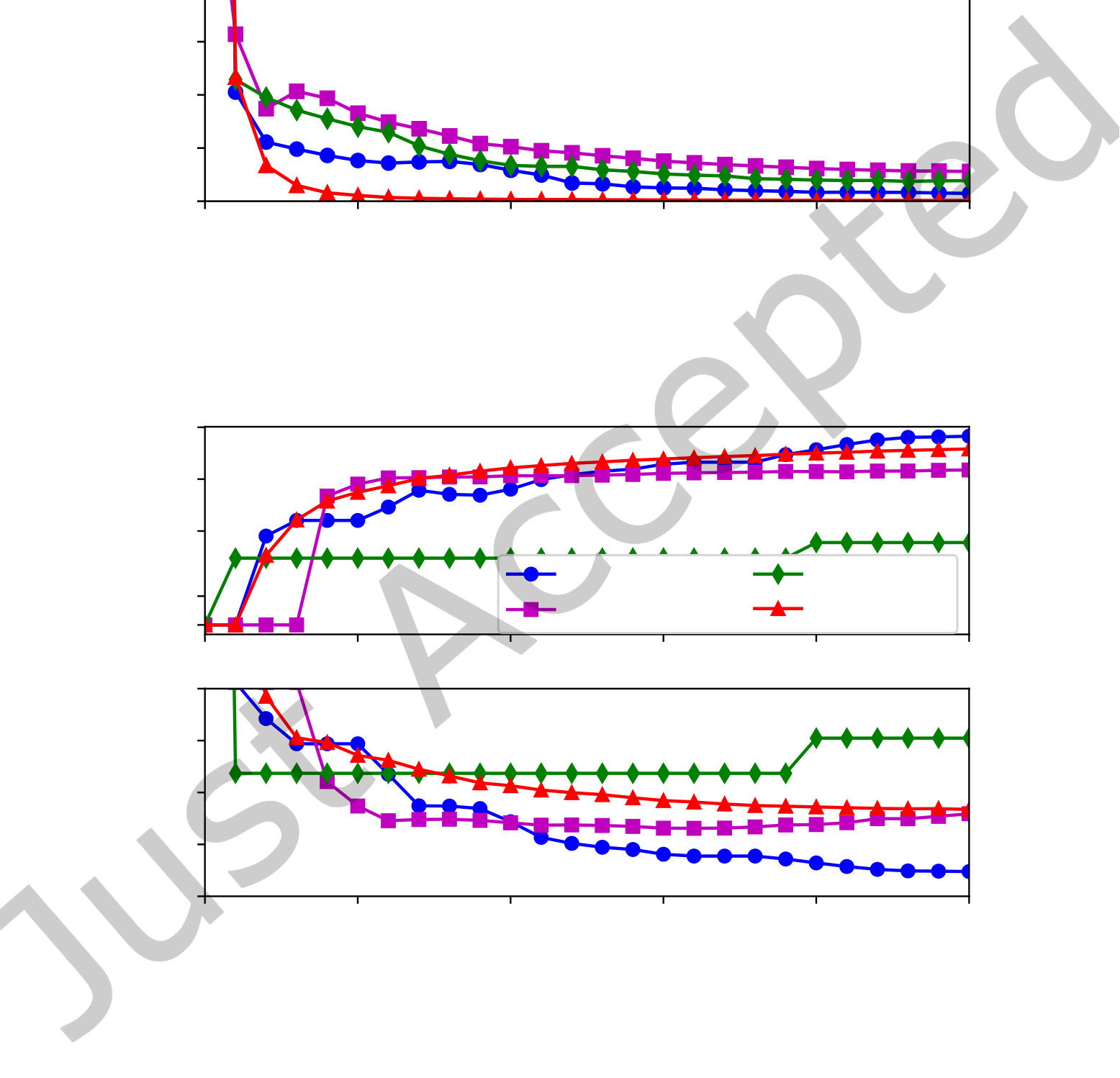
more afected than the other MOAs, which do not use scalarization schemes. This may be seen as evidence that VTMOP’s trust region approach is successfully forcing solutions in these nonconvex regions. The DIR variant again succeeds in producing several low RMSE solutions within the irst 5,000 iterations, and in the larger problem, the DIR variant inds several such solutions signiicantly sooner than all other MOAs except MultiGLODS. However, after quickly discovering these initial solutions, MultiGLODS appears to particularly struggle on the larger problem size (p = 4,d = 14) with diversity of solutions, failing to produce just 10 solutions with error less than 0.1 until after 20,000 function evaluations. Due to the lack of repeated trials for MultiGLODS, this could be attributed to a pathological sampling pattern for this problem.

6.2.3 DTLZ5. DTLZ5 has a (p − 2)-dimensional Pareto front that traces an arc of the Pareto front for DTLZ2. This is considered a degenerate case for MOPs since many methods expect the Pareto front to be a manifold with dimension p − 1. For this problem, the Delaunay discrepancy metric cannot be accurately computed since the Pareto front is not (p − 1)-dimensional. Therefore, only the number of solutions and RMSE are shown in Figures 8 and 9.

As with DTLZ2, all of the MOAs perform well on the DTLZ5 problems, producing numerous low RMSE solutions with fewer than 5,000 function evaluations. Again, MultiGLODS struggles with the diversity of solutions for the larger problem size (p = 4,d = 14), but this may be expected given that DTLZ5 is very similar to DTLZ2 in its nonconvexity. For VTMOP, these problems are a test of robustness since the (p − 2)-dimensional Pareto front can cause a degeneracy for DELAUNAYSPARSE when computing the Delaunay graph.

6.2.4 DTLZ7. DTLZ7 has a Pareto front that is discontinuous, consisting of 2p−1disconnected regions. This presents a challenge for techniques such as VTMOP that attempt to ill in gaps in the Pareto front since it is not possible to achieve a uniform spread of solutions.

ACM Trans. Math. Softw.



|  |  |  |
| --- | --- | --- |
| Algorithm XXXX: VTMOP: Solver for Blackbox Multiobjective Optimization Problems | • | 23 |

103

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Num. pts | 102 | 0 | 5000 | 10000 | VTMOP/DIR | MultiGLODS | |
| 101 |
| 100 | VTMOP/LH | NSGA-II | |
| 0 |
| 15000 | 20000 | 25000 |

0.100

0.075

|  |  |
| --- | --- |
| RMSE | 0.050 |

0.025

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 0.000 | 0 | 5000 | 10000 | 15000 | 20000 | 25000 |

Number of *F* evals

Fig. 8. Performance of MOAs on DTLZ5 for p = 3, d = 8.

103

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Num. pts | 102 | 0 | 5000 | 10000 | VTMOP/DIR | MultiGLODS | |
| 101 |
| 100 |
| 0 | VTMOP/LH | NSGA-II | |
| 15000 | 20000 | 25000 |

0.100

0.075

|  |  |
| --- | --- |
| RMSE | 0.050 |

0.025

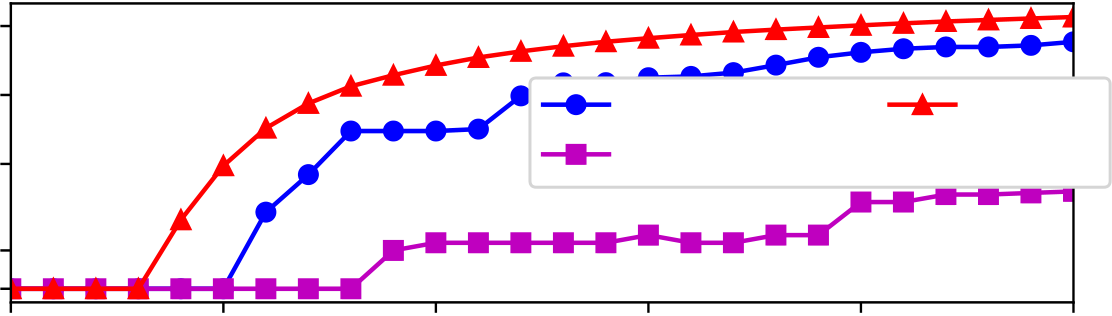
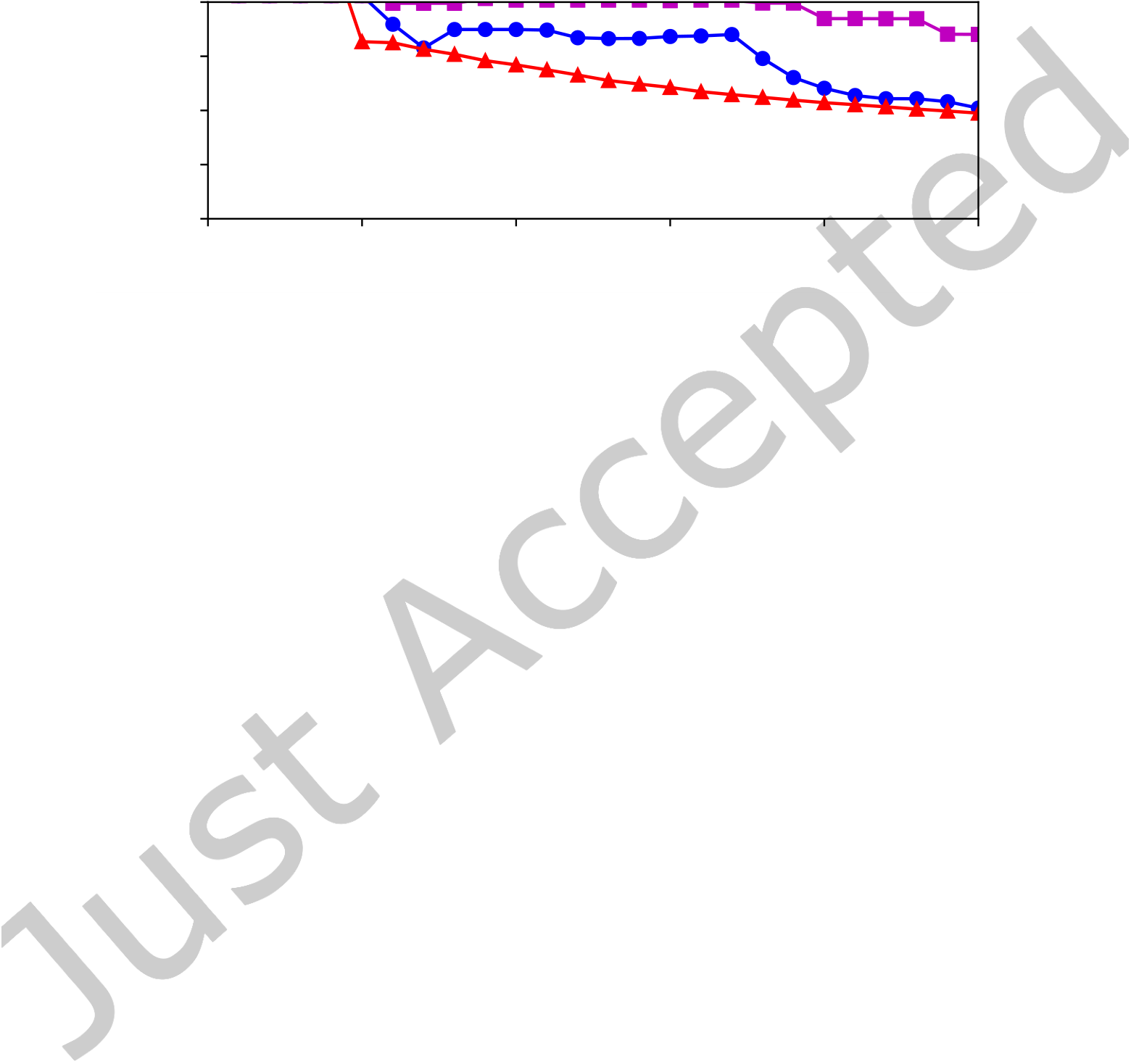
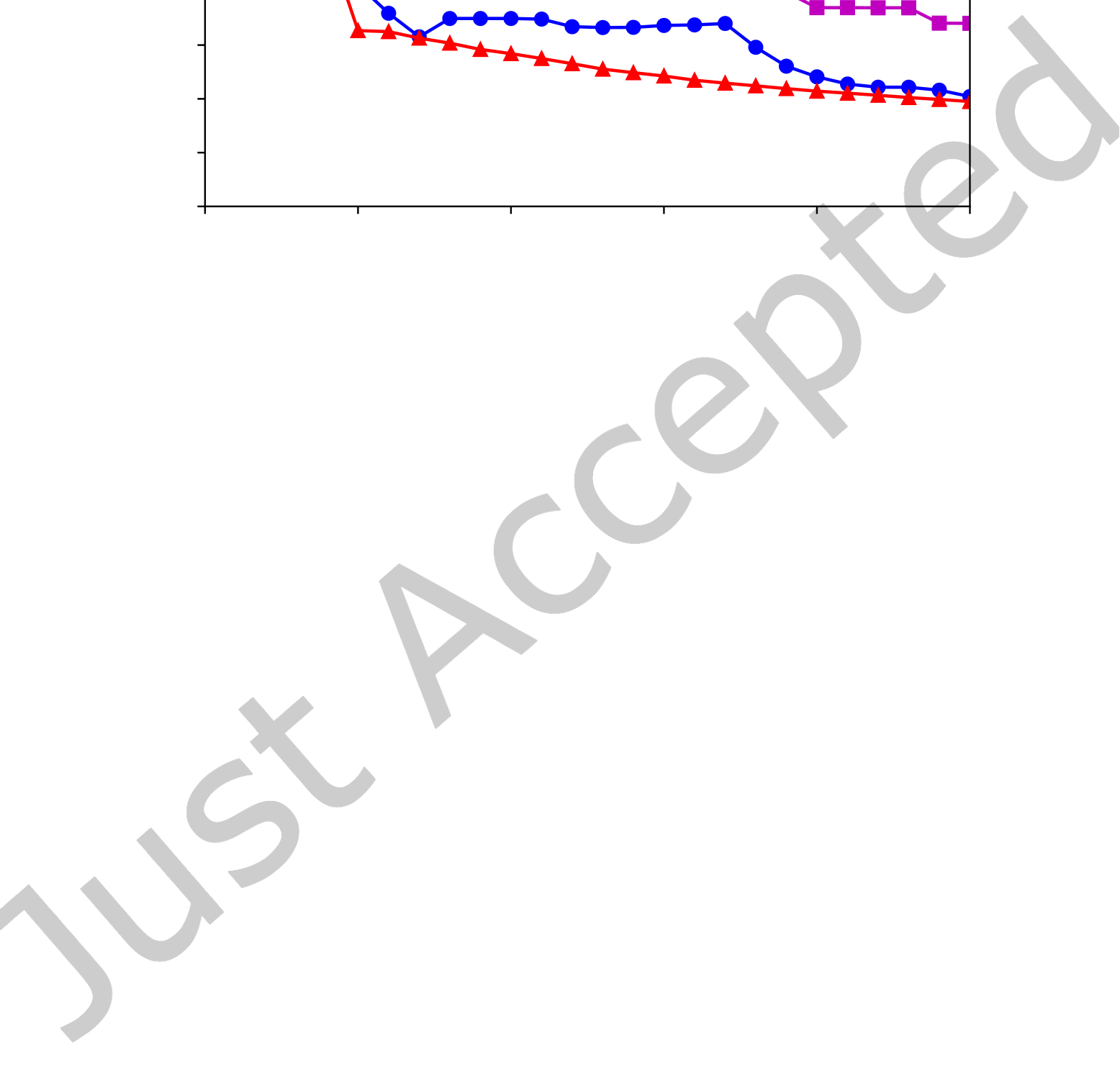
|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 0.000 | 0 | 5000 | 10000 | 15000 | 20000 | 25000 |

Number of *F* evals

Fig. 9. Performance of MOAs on DTLZ5 for p = 4, d = 14.

Similarly to DTLZ1, this problem is particularly challenging for the MOAs considered. None of the algorithms could consistently produce suitable solutions to the larger problem size (p = 4,d = 14). However, for the problem size p = 3,d = 8, both VTMOP variations and NSGA-II were able to ind several solutions. Also, similarly to DTLZ5, the discrepancy metric is not meaningful here since the Pareto front contains numerous gaps. The results for the smaller problem size are shown in Figure 10.

ACM Trans. Math. Softw.



|  |  |  |
| --- | --- | --- |
| 24 | • | Chang et al. |

103

|  |  |  |  |
| --- | --- | --- | --- |
| Num. pts | 102 | VTMOP/DIR | NSGA-II |
| 101 | VTMOP/LH |

100   
0

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 0 | 5000 | 10000 | 15000 | 20000 | 25000 |

0.100

0.075

|  |  |
| --- | --- |
| RMSE | 0.050 |

0.025

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 0.000 | 0 | 5000 | 10000 | 15000 | 20000 | 25000 |

Number of *F* evals

Fig. 10. Performance of MOAs on DTLZ7 for p = 3, d = 8.

As seen in Figure 10, NSGA-II appears to perform slightly better than VTMOP on this problem. However, the VTMOP DIR variant is able to discover several low-error solutions within the irst 10,000 function evaluations. The VTMOP LH variant is signiicantly less performant here, requiring many more evaluations to produce just a few solutions. It is worth noting that DTLZ7 is a pathological problem for VTMOP since it is not possible to łill inž the gaps in DTLZ7’s Pareto front, which is the main goal of VTMOP.

6.3 Iteration Time

As noted in Sections 6.1 and 6.2, VTMOP achieves strong performance per number of function evaluations on a variety of test problems. However, this performance comes at an expense. NSGA-II and MultiGLODS spent negligible amounts of time on iteration tasks for all problems, problem sizes, and budgets discussed. However, both variations of VTMOP required signiicant amounts of CPU time for iteration tasks, especially for larger budgets. This time was largely dominated by the time required to solve the surrogate optimization problems, as described in Section 3.3.

To collect timing results, VTMOP has been run serially on the HPC Bebop at Argonne National Laboratory. For all of the test problems and problem sizes from Sections 6.1 and 6.2, the Linux tool perf showed that both the DIR and LH variations spent between 80% and 90% of wallclock time inside the VTMOP\_OPT subroutine, either itting the LSHEP models or solving the surrogate optimization problems. The majority of remaining time was spent in VTMOP\_LTR (see Section 3.2), either extracting the Pareto front, calculating the Delaunay graph, or identifying the most isolated point. Note that the majority of these tasks could be parallelized, as described in Section 4.3.

Since the iteration costs are dependent upon the size of VTMOP’s internal database, this iteration complexity grows with the total number of function evaluations that have already been performed. In Figure 11, these iteration times are plotted against the total number of function evaluations at the time of the iteration. Note that the y-axis is at a log-scale.

As seen in Figure 11, the iteration times are similar for both, and grow very quickly with the size of the database. It appears that the LH variant may require slightly more wallclock time per iteration, but overall,

ACM Trans. Math. Softw.

|  |  |  |
| --- | --- | --- |
| Algorithm XXXX: VTMOP: Solver for Blackbox Multiobjective Optimization Problems | • | 25 |

Iteration time vs. number of F evals

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Iteration time (s) | 103 | |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | | | | | |  | |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | | |  |  | |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | | |  | |  | |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | | | | | |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | | | |  |  | |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | | | |  |  | |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | | | |  |  | |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | | | |  |  | |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | | | | | | |
|  |
|  |
|  |
|  |
|  |  |  |  |  |  |  |

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 26 | • | Chang et al. |  | 200 | Iterations completed vs. number of F evals | | | | | |
| |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | | | | | | |  | | | | | | |  | |  |  |  |  |  |  |  |  |  |  |  |  |  | |  | | | | |  |  | |  | |  |  | |  |  |  | | | | |  |  |  | T |  | M | O |  | P |  |  | R |  |  | C |  | T |  | varia nt | | | | |  | nt | | |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | |  |  | V | T | M | M | O | O | P |  | D | R | E | E | C | C | T | T |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | |  |  | V | T |  | M | O |  | P |  | H |  | v |  | a | r | i |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | |  |  |  | T |  | M | O |  | P |  |  |  | v |  | a | r | i | a |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | | | | | | |
|  |
|  |
|  |
|  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |



|  |  |  |
| --- | --- | --- |
| Algorithm XXXX: VTMOP: Solver for Blackbox Multiobjective Optimization Problems | • | 27 |

Fig. 13. Design variables and bounds for the LCLS-II superconducting injector. Two solenoid magnets are included, resulting in a total of seven design dimensions. Note that solenoid strengths were rescaled by the łScale Factorž before being passed to the optimizer.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Variable | |  |  |  |  | | --- | --- | --- | --- | | Minimum | Maximum | Unit | Scale Factor | |
| Buncher phase  Solenoid strength(s) Cavity gradient  Cavity phase | |  |  |  |  | | --- | --- | --- | --- | | -100.0 | -10.0 | Degrees | 1 | | 0.02 | 0.07 | T/m | 1,000 | | 0.0 | 32.0 | MV/m | 1 | | -40.0 | 40.0 | Degrees | 1 | |

For the following results, d = 7 and p = 3. The upper and lower bounds for the design variables are given in Table I, alongside the factors by which certain design variables were rescaled in order to maintain reasonable step sizes during optimization. The objectives of interest are the emittance (ϵx), bunch length (σz), and energy spread (dE), all of which are to be minimized.

An electron bunch in an accelerator is commonly described with six dimensions: three spatial and three momenta. Emittance is deined as the combined spatial and momentum distribution of the particles in the 6-dimensional phase space. Due to symmetries in the problem, the four transverse dimensions are monitored with ϵx, the combined spatial and momenta distribution in x. Free electron laser facilities are always in search of improved emittance values, as it directly translates to better X-ray performance downstream. Given this information, VTMOP is directed to focus on the portion of the Pareto front with emittance less than 2 µm (using the optional objective bounds input), as LCLS-II would never be operated with ϵx > 2 µm during normal operations.

The second objective, bunch length, is a metric used to gauge the longitudinal beam size (i.e., the spatial dimension in z). Emittance and bunch length are adversarial objectives that are normally considered when optimizing photoinjectors. The third objective, energy spread, is related to the spread in kinetic energy of the particle distribution. The energy spread is included as a third objective because previous p = 2 optimizations of LCLS-II have been inadequate. While solutions from initial optimization runs produced emittance and bunch length values that met performance metrics, the energy spread was too large and rendered the beam distributions unusable without modiication of cavity phases. This is a result of the energy spread’s impact on beam behavior in downstream hardware (namely bunch compressors). This third objective must be accounted for, or the need to manually modify the results after the optimization will remain. In the past, adding a constraint on the energy spread did not provide suicient improvement in the inal optimized beamline, so the choice was made to include it as an additional objective.

It is also necessary to include a penalty term, υ, in the objectives in order to account for infeasible solutions. When a set of parameters results in beam loss, particles are removed from the simulation. If a signiicant amount of particles are removed, the emittance, bunch length, and energy spread all can appear artiicially small. Given the initial and inal number of particles in the simulation (N0 and Nf ), the penalty is calculated as

|  |  |
| --- | --- |
| υ =N0 − Nf   c | , |

where c is a constant used to scale the penalty to the same order as the scaled objective values. In the case shown in this paper, the value of c = 20.0 is used. Once the penalty is calculated, it is added to all objective values for that evaluation. The addition of a strictly positive penalty term to all objectives is a natural extension of the augmented Lagrangian approach for nonlinear programming to the multiobjective case [25].

The parallel particle-in-cell code OPAL-T [2] is used to simulate the portion of the LCLS-II accelerator. One OPAL-T simulation of this beamline requires on average 4.3 minutes (with signiicant variability) on 4 cores of

ACM Trans. Math. Softw.



|  |  |  |
| --- | --- | --- |
| 28 | • | Chang et al. |

the HPC Bebop at Argonne National Laboratory. These 4-core evaluations are considered low-idelity simulations. When candidate beamline conigurations are identiied, higher idelity simulations are run to conirm the results and reine working points. Higher idelity simulations are too computationally intensive to use within the widespread genetic MOA’s that require thousands of function evaluations. Still, even for this lower idelity version of the problem, 5,000 evaluations of OPAL-T require well over 1,000 CPU hours on Bebop or over 350 hours of wallclock time when run serially.

VTMOP was applied to this problem with various budgets of OPAL-T evaluations. VTMOP was conigured to search using DIRECT and run using its default settings except for the trust region radius fractions, which were reduced to 10% and 2% of the design bounds, respectively, because a large portion of the bound-constrained design space is actually infeasible for the original problem (resulting in a nonzero penalty υ). Additionally, as previously stated, VTMOP was directed to focus on the portion of the Pareto front with emittance less than 2 µm using the optional objective bounds input. During initial tests, VTMOP was not able to identify any designs that met this strict objective bound after the 0th iteration (as discussed in Remark 3.1). Therefore, VTMOP was run for 5 iterations (1,534 function evaluations) without any bounds on the interesting range of objectives, after which numerous points meeting these bounds were identiied. Then the objective bounds were set for ϵx<= 2 µm for the remaining 3, 466 function evaluations. Finally, in order to ensure equal treatment of all objectives, in the version of OPAL-T that was run by VTMOP, the raw output values of ϵx and σz were rescaled so that their expected range would match the expected range of dE. Note that this rescaling was only for VTMOP’s internal use, and the unscaled values of ϵx and σz are reported in the following results.

The raw objective values for the solutions found by VTMOP after certain numbers of simulation evaluations are shown in Figure 14. After 4,000 evaluations, no signiicant further improvements were made. Therefore, in order to save space, only results up to a budget of 4,000 are shown in Figure 14.

Compared to a typical commissioning optimization of LCLS-II, VTMOP’s convergence within 4,000 evaluations is a marked improvement. Previous optimizations of LCLS-II were typically run to 30,000 evaluations with constraints on the objective bounds [54] or 10,000 evaluations with no constraints [47]. In both cases, only the emittance (ϵx) and bunch length (σz) were considered (p = 2), producing solutions with unit energy spreads (dE). As a result, a signiicant amount of time was required after the optimization in order to partially reconigure the solutions to improve their energy spreads. Note that adding the energy spread as a constraint to NSGA-II has been attempted in the past with poor results. The results shown in Figure 14 indicate reasonable energy spread values for a range of emittance and bunch lengths. This shows an opportunity to simultaneously reduce the number of function evaluations and avoid the need to postprocess solutions after future LCLS-II optimizations, saving a signiicant amount of time and resources.

ACKNOWLEDGEMENTS

This work was supported in part by NSF Grants DGE-154362, CNS-1565314, and CNS-1838271 and by the U.S. Department of Energy, Oice of Science through the Exascale Computing Project (Contract No. 17-SC-20-SC), the Oice of Basic Energy Sciences (Contract No. DE-AC-02-76SF00515), and the Oice of Advanced Scientiic Computing Research, Scientiic Discovery through Advanced Computing (SciDAC) Program through the FASTMath Institute under Contract No. DE-AC02-06CH11357.

This work was also supported in part by the U.S. Dept. of Energy, Oice of Science Graduate Student Research program (Contract No. DE-SC0014664). The SCGSR program is administered by the Oak Ridge Institute for Science and Education (ORISE), which is managed by ORAU under contract number DE-SC0014664. All opinions in this paper are the authors’ and do not necessarily relect the policies and views of the DOE, ORAU, or ORISE.

The authors gratefully acknowledge the computing resources provided on Bebop, a high-performance com-puting system operated by the Laboratory Computing Resource Center at Argonne National Laboratory. The

ACM Trans. Math. Softw.

|  |  |  |
| --- | --- | --- |
| Algorithm XXXX: VTMOP: Solver for Blackbox Multiobjective Optimization Problems | • | 29 |

Budget: 1000

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| σz | 0.002 | |  |  |  |  | | --- | --- | --- | --- | |  |  |  |  | | | |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |



|  |  |  |
| --- | --- | --- |
| 30 | • | Chang et al. |

[7] C. Audet and J. E. Dennis. 2006. Mesh adaptive direct search algorithms for constrainted optimization. SIAM Journal on Optimization 17, 1 (2006), 188ś217.

[8] C. Audet, G. Savard, and W. Zghal. 2008. Multiobjective optimization through a series of single-objective formulations. SIAM Journal on Optimization 19, 1 (2008), 188ś210.

[9] C. Audet, G. Savard, and W. Zghal. 2010. A mesh adaptive direct search algorithm for multiobjective optimization. European Journal of Operational Research 204, 3 (2010), 545ś556.

[10] C. B. Barber, D. P. Dobkin, and H. Huhdanpaa. 1996. The Quickhull algorithm for convex hulls. ACM Trans. Math. Software 22, 4 (1996), 469ś483.

[11] I. V. Bazarov and C. K Sinclair. 2005. Multivariate optimization of a high brightness DC gun photoinjector. Physical Review Special Topics - Accelerators and Beams 8, 3, Article 034202 (2005).

[12] W. F. Bergan, I. V. Bazarov, C. J. R. Duncan, D. B. Liarte, D. L. Rubin, and J. P. Sethna. 2019. Online storage ring optimization using dimension-reduction and genetic algorithms. Physical Review Accelerators and Beams 22, 5, Article 054601 (2019).

[13] R. T. Biedron, J. R. Carlson, J. M. Derlaga, P. A. Gnofo, D. P. Hammond, W. T. Jones, B. Kleb, E. M. Lee-Rausch, E. J. Nielson, M. A. Park, C. L. Rumsey, J. L. Thomas, K. B. Thompson, and W. A. Wood. 2019. FUN3D Manual: 13.6. Technical Report Technical Memorandum 2019-220416. NASA, Langley Research Center, Hampton, VA.

[14] J. Bigeon, S. Le Digabel, and L. Salomon. 2021. DMulti-MADS: Mesh adaptive direct multisearch for bound-constrained blackbox multiobjective optimization. Computational Optimization and Applications 79, 2 (2021), 301ś338.

[15] J. Blank and K. Deb. 2020. pymoo: Multi-objective optimization in Python. IEEE Access 8 (2020), 89497ś89509.

[16] J. Blank, K. Deb, and P. C. Roy. 2019. Investigating the normalization procedure of NSGA-III. In Proc. 10th International Conference on Evolutionary Multi-Criterion Optimization. Springer, MI, USA, 229ś240.

[17] OpenMP Architecture Review Board. 2015. OpenMP Application Programming Interface. Technical Report version 4.5.

[18] J.-D. Boissonnat, O. Devillers, and S. Hornus. 2009. Incremental construction of the Delaunay triangulation and the Delaunay graph in medium dimension. In Proc. Twenty-ifth Annual Symp. on Computational Geometry. ACM, New York, NY, 208ś216.

[19] E. F. Campana, M. Diez, G. Liuzzi, S. Lucidi, R. Pellegrini, V. Piccialli, F. Rinaldi, and A. Serani. 2018. A multi-objective DIRECT algorithm for ship hull optimization. Computational Optimization and Applications 71, 1 (2018), 53ś72.

[20] T. H. Chang, J. Larson, and L. T. Watson. 2020. Multiobjective optimization of the variability of the high-performance LINPACK solver. In Proc. Winter Simulation Conference. IEEE, Orlando, FL, 3081ś3092.

[21] T. H. Chang, J. Larson, L. T. Watson, and T. C. H. Lux. 2020. Managing computationally expensive blackbox multiobjective optimization problems using libEnsemble. In Proc. 28th High Performance Computing Symposium. SCS, Fairfax, VA, Article 31.

[22] T. H. Chang, L. T. Watson, T. C. H. Lux, A. R. Butt, K. W. Cameron, and Y. Hong. 2020. Algorithm 1012: DELAUNAYSPARSE: Interpolation via a sparse subset of the Delaunay triangulation in medium to high dimensions. ACM Trans. Math. Software 46, 4, Article 38 (2020). [23] T. H. Chang, L. T. Watson, T. C. H. Lux, B. Li, L. Xu, A. R. Butt, K. W. Cameron, and Y. Hong. 2018. A polynomial time algorithm for multivariate interpolation in arbitrary dimension via the Delaunay triangulation. In Proc. ACM Southeast Conference. ACM, Richmond, KY, Article 12.

[24] T. H. Chang, L. T. Watson, T. C. H. Lux, S. Raghvendra, B. Li, L. Xu, A. R. Butt, K. W. Cameron, and Y. Hong. 2018. Computing the umbrella neighbourhood of a vertex in the Delaunay triangulation and a single Voronoi cell in arbitrary dimension. In Proc. IEEE SoutheastCon. IEEE, St. Petersburg, FL.

[25] G. Cocchi and M. Lapucci. 2020. An augmented Lagrangian algorithm for multi-objective optimization. Computational Optimization and Applications 77, 1 (2020), 29ś56.

[26] K. Cooper and S. R. Hunter. 2020. PyMOSO: Software for multi-objective simulation optimization with R-PERLE and R-MinRLE. INFORMS Journal on Computing 32, 4 (2020), 1101ś1108.

[27] A. L. Custódio and J. F. A. Madeira. 2018. MultiGLODS: Global and local multiobjective optimization using direct search. Journal of Global Optimization 72, 2 (2018), 323ś345.

[28] A. L. Custódio, J. F. A. Madeira, A. I. F. Vaz, and L. N. Vicente. 2011. Direct multisearch for multiobjective optimization. SIAM Journal on Optimization 21, 3 (2011), 1109ś1140.

[29] K. Deb, A. Pratap, S. Agarwel, and T. Meyarivan. 2002. A fast and elitist multiobjective genetic algorithm: NSGA-II. IEEE Trans. Evolutionary Computation 6, 2 (2002), 182ś197.

[30] K. Deb, L. Thiele, M. Laumanns, and E. Zitzler. 2002. Scalable multi-objective optimization test problems. In Proc. 2002 IEEE Congress on Evolutionary Computation. IEEE, Honolulu, HI, 825ś830.

[31] L. T. Deshpande, S. Watson, J. Shu, and N. Ramakrishnan. 2011. Data driven surrogate-based optimization in the problem solving environment WBCSim. Engineering with Computers 27, 3 (2011), 211ś223.

[32] S. Deshpande, L. T. Watson, and R. A. Canield. 2016. Multiobjective optimization using an adaptive weighting scheme. Optimization Methods and Software 31, 1 (2016), 110ś133.

[33] M. Ehrgott. 2005. Multicriteria Optimization (2 ed.). Springer Science & Business Media, Heidelberg, Germany.

ACM Trans. Math. Softw.



|  |  |  |
| --- | --- | --- |
| Algorithm XXXX: VTMOP: Solver for Blackbox Multiobjective Optimization Problems | • | 31 |

[34] Y. Feng, Z. Chen, Y. Dai, F. Wang, J. Cai, and Z. Shen. 2018. Multidisciplinary optimization of an ofshore aquaculture vessel hull form based on the support vector regression surrogate model. Ocean Engineering 166 (2018), 145ś158.

[35] C. Gulliford, A. Bartnik, and I. Bazarov. 2016. Multiobjective optimizations of a novel cryocooled DC gun based ultrafast electron difraction beam line. Physical Review Accelerators and Beams 19, 9, Article 093402 (2016).

[36] C. Gulliford, A. Bartnik, I. Bazarov, and J. Maxson. 2017. Multiobjective optimization design of an rf gun based electron difraction beam line. Physical Review Accelerators and Beams 20, 3, Article 033401 (2017).

[37] J. He, L. T. Watson, and M. Sosonkina. 2009. Algorithm 897: VTDIRECT95: Serial and parallel codes for the global optimization algorithm DIRECT. ACM Trans. Math. Software 36, 3, Article 17 (2009).

[38] A. Holer, B. Terzic, M. Kramer, A. Zvezdin, V. Morozov, Y. Roblin, F. Lin, and C. Jarvis. 2013. Innovative applications of genetic algorithms to problems in accelerator physics. Physical Review Special Topics - Accelerators and Beams 16, 1, Article 010101 (2013).

[39] S. R. Hunter, E. A. Applegate, V. Arora, and B. Chong. 2019. An introduction to multiobjective simulation optimization. ACM Transactions on Modeling and Computer Simulation 29, 1 (2019), 1ś36.

[40] D. R. Jones, C. D. Perttunen, and B. E. Stuckman. 1993. Lipschitzian optimization without the Lipschitz constant. Journal of Optimization Theory and Applications 79, 1 (1993), 157ś181.

[41] V. Klee. 1980. On the complexity of d-dimensional Voronoi diagrams. Archiv der Mathematik 34, 1 (1980), 75ś80.

[42] L. Kocis and W. J Whiten. 1997. Computational investigations of low-discrepancy sequences. ACM Trans. Math. Software 23, 2 (1997), 266ś294.

[43] M. Kranjčević, A. Adelmann, P. Arbenz, A. Citterio, and L. Stingelin. 2019. Multi-objective shape optimization of radio frequency cavities using an evolutionary algorithm. Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment 920 (2019), 106ś114.

[44] S. C. Kugele, M. W. Trosset, and L. T. Watson. 2008. Numerical integration in statistical decision-theoretic methods for robust design optimization. Structural Multidisciplinary Optimization 36 (2008), 457ś475.

[45] S. Le Digabel. 2011. Algorithm 909: NOMAD: Nonlinear optimization with the MADS algorithm. ACM Trans. Math. Software 37, 4, Article 44 (2011).

[46] S. Le Digabel and S. M. Wild. 2015. A taxonomy of constraints in simulation-based optimization. Technical Report Technical Report G-2015-57. Les cashiers du GERAD, Montréal, Canada.

[47] R. Lemons, N. Neveu, J. Duris, A. Marinelli, C. Durfee, and S. Carbajo. 2022. Temporal shaping of narrow-band picosecond pulses via non-colinear sum-frequency mixing of dispersion-controlled pulses. Physical Review Special Topics - Accelerators and Beams 25, 1, Article 013401 (2022).

[48] K. Li, R. Wang, T. Zhang, and H. Ishibuchi. 2018. Evolutionary many-objective optimization: A comparative study of the state-of-the-art. IEEE Access 6 (2018), 26,194ś26,214.

[49] G. Liuzzi, S. Lucidi, and F. Rinaldi. 2016. A derivative-free approach to constrained multiobjective nonsmooth optimization. SIAM Journal on Optimization 26, 4 (2016), 2744ś2774.

[50] A. Lovison and K. Miettinen. 2021. On the extension of the DIRECT algorithm to multiple objectives. Journal of Global Optimization 79, 2 (2021), 387ś412.

[51] R. T. Marler and J. S. Arora. 2004. Survey of multi-objective optimization methods for engineering. Structural and Multidisciplinary Optimization 26, 6 (2004), 369ś395.

[52] J. Müller. 2017. SOCEMO: Surrogate optimization of computationally expensive multiobjective problems. INFORMS Journal on Computing 29, 4 (2017), 581ś596.

[53] R. H. Myers, D. C. Montgomery, and C. M. Anderson-Cook. 2016. Response Surface Methodology: Process and Design Optimization Using Designed Experiments (4 ed.). John Wiley & Sons, Inc., Hoboken, NJ.

[54] N. Neveu and Y. Ding. 2021. LCLS-II standard coniguration with a Gaussian-proile injector laser. Technical Report LCLS-II-TN-21-01. SLAC National Accelerator Laboratory, Menlo Park, CA.

[55] N. Neveu, N. Sudar, Y. Ding, G. Marcus, A. Marinelli, and C. Mayes. 2020. LCLS-II standard coniguration with a Gaussian-proile injector laser. Technical Report LCLS-II-TN-20-03. SLAC National Accelerator Laboratory, Menlo Park, CA.

[56] J. Nocedal and S. J. Wright. 2006. Numerical Optimization (2 ed.). Springer Science & Business Media, Heidelberg, Germany. [57] J. Qiang, C. Mitchell, and A. Qiang. 2016. Tuning of an adaptive uniied diferential evolution algorithm for global optimization. In Proc. 2016 IEEE Congress on Evolutionary Computation. IEEE, Vancouver, BC, Canada, 4061ś4068.

[58] C. Raghunath, T. H. Chang, L. T. Watson, M. Jrad, R. K. Kapania, and R. M. Kolonay. 2017. Global deterministic and stochastic optimization in a service oriented architecture. In Proc. 25th High Performance Computing Symposium. SCS, Virginia Beach, VA, Article 7.

[59] J. Ryu, S. Kim, and H. Wan. 2009. Pareto front approximation with adaptive weighted sum method in multiobjective simulation optimization. In Proc. Winter Simulation Conference. IEEE, Austin, TX, 623ś633.

[60] J. Sobieszczanski-Sobieski, A. Morris, and M. Van Tooren. 2015. Multidisciplinary Design Optimization Supported by Knowledge Based Engineering. John Wiley & Sons, Ltd., Chichester, UK.

[61] S. Tavares. 2020. Contributions to the development of an integrated toolbox of solvers in derivative-free optimization. Master’s thesis.

ACM Trans. Math. Softw.



|  |  |  |
| --- | --- | --- |
| 32 | • | Chang et al. |

[62] B. Terzic, A. S. Holer, C. J. Reeves, S. A. Khan, G. A. Kraft, J. Benesch, A. Freyberger, and D. Ranjan. 2014. Simultaneous optimization of the cavity heat load and trip rates in linacs using a genetic algorithm. Physical Review Special Topics - Accelerators and Beams 17, 10, Article 101003 (2014).

[63] W. I. Thacker, J. Zhang, L. T. Watson, J. B. Birch, M. A. Iyer, and M. W. Berry. 2010. Algorithm 905: SHEPPACK: Modiied Shepard algorithm for interpolation of scattered multivariate data. ACM Trans. Math. Software 37, 3, Article 34 (2010).

[64] J. Thomann and G. Eichfelder. 2019. A trust-region algorithm for heterogeneous multiobjective optimization. SIAM Journal on Optimization 29, 2 (2019), 1017ś1047.

[65] T. Tušar and B. Filipič. 2014. Visualization of Pareto front approximations in evolutionary multiobjective optimization: A critical review and the prosection method. IEEE Transactions on Evolutionary Computation 19, 2 (2014), 225ś245.

[66] L. Yang, D. Robin, F. Sannibale, C. Steier, and W. Wan. 2009. Global optimization of an accelerator lattice using multiobjective genetic algorithms. Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment 609, 1 (2009), 50ś57.

[67] A. K. Y. Yee, A. K. Ray, and G. P. Rangaiah. 2003. Multiobjective optimization of an industrial styrene reactor. Computers & Chemical Engineering 27, 1 (2003), 111ś130.

A COMPUTING THE DELAUNAY GRAPH

The pseudocode for the Delaunay graph calculation described in Section 3.1.1 is given in Algorithm 3.

**Algorithm 3** Compute the Delaunay graph.

|  |  |  |
| --- | --- | --- |
| **input** | �Z(k,1), . . . ,Z(k,m(k))� | with each Z(k,i)deined as in (5); |
| Π(k)= |

**begin**   
M(k)is a Boolean valued matrix of dimensions m(k)× m(k); **initialize**

**for** i = 1, . . ., m(k)**do**  M(k)← FALSE ... . . . FALSE ...  ; FALSE . . . FALSE

**for** j = i + 1, . . ., m(k)**do**  
 ¯Z(k,i,j)← the midpoint between Z(k,i)and Z(k,j);   
 S(k,i,j)← the vertex set for a Delaunay simplex containing ¯Z(k,i,j); **if** Z(k,i)∈ S(k,i,j)**and** Z(k,j)∈ S(k,i,j)**then**   
 M(k) i,j← TRUE;   
 M(k) j,i← TRUE;   
 **endif**   
 **enddo**   
**enddo**   
**return** M(k)

The key cost for Algorithm 3 is computing S(k,i,j). VTMOP uses the DELAUNAYSPARSE software package [22] to compute each of these simplices. The computational complexity of DELAUNAYSPARSE is O(m(k)p2cf ) per simplex constructed, where cf is the number of łlipsž required to converge. Empirically, it has been shown that cf is typically a polynomial function of p and independent of m(k)[23]. Therefore, the execution time for Algorithm 3 is a cubic function of m(k), regardless of the number of objectives p.

Algorithm 3 correctly computes the Delaunay graph when Π(k)is in general position, meaning that the Delaunay triangulation of Π(k)exists and is unique. This is established in Theorem A.1. Existence is guaranteed when Π(k)

ACM Trans. Math. Softw.



|  |  |  |
| --- | --- | --- |
| Algorithm XXXX: VTMOP: Solver for Blackbox Multiobjective Optimization Problems | • | 33 |

does not lie in a (p − 2)-dimensional linear manifold, and uniqueness is guaranteed when there is no set of p + 1 points in Π(k)that lie on the same (p − 2)-sphere.

Theorem A.1. If Π(k)is in general position, then Algorithm 3 computes the connectivity matrix for DG(Π(k)).

Proof. Suppose that Π(k)is in general position; and let S(k,i,j), Z(k,i), Z(k,j), and¯Z(k,i,j)be as deined in Algorithm 3. If S(k,i,j)contains both Z(k,i)and Z(k,j), then clearly Z(k,i)and Z(k,j)are connected in the Delaunay graph. Otherwise, since the simplex whose vertices are S(k,i,j)contains the midpoint¯Z(k,i,j), at least one facet of this simplex must separate Z(k,i)and Z(k,j). Therefore, Z(k,i)and Z(k,j)cannot be connected in the Delaunay

graph. □

Remark A.1. If DG(Π(k)) is not unique, then Algorithm 3 may fail to produce a connectivity structure that

is consistent with any Delaunay triangulation. However, the Gabriel graph is a subgraph of every Delaunay

triangulation, and therefore the Gabriel graph will always be embedded in the connectivity structure computed

by Algorithm 3, regardless of uniqueness.

Remark A.2. If the Delaunay triangulation does not exist, then Π(k)is contained in a (p − 2)-dimensional linear manifold. Typically, this occurs only when m(k)< p. Then all points in Π(k)are considered neighbors, and the

most isolated point corresponds to the projected point that is, on average, farthest away from all other points in Π(k). One extremely rare case is that m(k)≥ p but Π(k)is contained in a (p − 2)-dimensional linear manifold due to the Pareto front being a (p − 2)- or lower-dimensional manifold embedded in the objective space. In this case,

a principal component analysis is performed by using the DGESVD subroutine from LAPACK [5] to compute the singular value decomposition of the matrix whose columns are points in Π(k)after they have been shifted so that their barycenter is the origin. After computing the decomposition, Π(k)is projected onto the span of the

left singular vectors whose corresponding singular values are greater than the objective space tolerance ϵ (from Remark 3.1.3). Then DG(Π(k)) is given by computing the Delaunay graph in this reduced dimensional space.

ACM Trans. Math. Softw.