Noname manuscript No.
(will be inserted by the editor)

PAINT: Pareto Front Interpolation for Nonlinear Multiobjective Optimization

Markus Hartikainen · Kaisa Miettinen · Margaret M. Wiecek

Received: date / Accepted: date

Abstract A method called PAINT is introduced for computationally expensive multiobjective optimization problems. The method interpolates between a given set of Pareto optimal outcomes. The interpolation provided by the PAINT method implies a mixed integer linear surrogate problem for the original problem which can be optimized with any interactive method to make decisions concerning the original problem. When the scalarizations of the interactive method used do not introduce nonlinearity to the problem (which is true e.g., for the synchronous NIMBUS method), the scalarizations of the surrogate problem can be optimized with available mixed integer linear solvers. Thus, the use of the interactive method is fast with the surrogate problem even though the problem is computationally expensive. Numerical examples of applying the PAINT method for interpolation are included.

Keywords Multiobjective optimization · Interactive decision making · Computationally expensive problems · Approximation

This research was partly supported by the Academy of Finland (grant number 128495) and Jenny and Antti Wihuri foundation

M. Hartikainen
Department of Mathematical Information Technology, P.O. Box 35 (Agora), 40014 University of Jyväskylä, Finland
E-mail: markus.e.hartikainen@jyu.fi

K. Miettinen
Department of Mathematical Information Technology, P.O. Box 35 (Agora), 40014 University of Jyväskylä, Finland
E-mail: kaisa.miettinen@jyu.fi

M. M. Wiecek
Department of Mathematical Sciences, Clemson University, U.S.A.
E-mail: wmalgor@clemson.edu

1 Introduction

Multiobjective optimization means optimizing multiple conflicting objectives at the same time [see e.g., 23]. Multiobjective optimization problems may have many Pareto optimal solutions, whose objectives cannot be improved without impairing some other objectives. A vector containing the objective values for a single solution is called an outcome and the set of outcomes given by Pareto optimal solutions is often called the Pareto front. The ultimate aim of multiobjective optimization is to help a decision maker find a solution that is preferable to him/her. A decision maker is a person who has the right or is expected to make decisions concerning the real life problem that is mathematically modeled by the multiobjective optimization problem. Here we do not assume that he/she has any knowledge about different multiobjective optimization methods.

One way to classify different multiobjective optimization methods, introduced in [15] and followed in [23], is based on the relative order of the decision maker's preference articulation and optimization. In a priori methods, the preferences of the decision maker are first specified and then a single Pareto optimal solution is found with respect to these and in a posteriori methods many Pareto optimal solutions are generated and then the decision maker is expected to choose a preferred one among them. If the decision maker's preferences do not play a role or are unavailable then the method is called a no-preference method. Finally, interactive methods employ an iterative procedure and allow the decision maker to correct his/her preferences and also enable the decision maker to learn about the problem [see e.g., 28]. Interactive methods have in many instances been seen as the most prominent [8, 20, 23, 28], because they allow the decision maker to gain more insight about the problem while solving it without introducing too much cognitive load at a time. In other words, the decision maker can consider only those Pareto optimal solutions that have been generated based on his/her preferences. We have, thus, focused on interactive methods in our research.

In interactive methods, new solutions to the multiobjective optimization problem are often produced through scalarization, i.e., by converting the multiobjective optimization problem into single objective optimization problems [20]. Different interactive methods use different scalarizations and some interactive methods even use multiple scalarizations (e.g., the synchronous NIMBUS method introduced in [27]). In this paper, a scalarization of the multiobjective optimization problem refers to a single objective optimization problem whose optimal solutions are solutions to the multiobjective optimization problem. Usually, scalarizations include a way to take into account the decision maker's preferences (as e.g., in achievement scalarizing problem (8) from [33], where the reference point contains aspiration levels for the objectives). For more information about scalarizations, see e.g., [23, 26].

A drawback with iterative procedures of interactive methods is that the decision maker has to wait while new Pareto optimal solutions are generated with his/her updated preferences. This is usually done in each iteration.

Since many real-life problems are computationally expensive [see e.g., 14, 18], scalarizations of the problems may also be computationally expensive. By a computationally expensive multiobjective optimization problem we here mean a problem that requires a long time to compute the objective values for a given solution. For these problems, the time that is needed to optimize the scalarizations may be too long for the decision maker to spare. Besides, the decision maker may become unwilling to explore different solutions to the problem in the so-called learning phase (in which the most interesting region on the Pareto front is to be identified [28]).

In this paper, we develop and implement a Pareto front approximation method PAINT (PAreto front INTerpolation) that interpolates between a set of given Pareto optimal outcomes. The interpolation satisfies the property of inherent nondominance introduced in [13] and, thus, implies a surrogate problem that can replace the computationally expensive original problem and can be solved with any interactive method to yield a preferred solution for the original problem. The inherent nondominance guarantees that none of the interpolants dominate or are dominated by any of the given Pareto optimal outcomes, neither do they dominate each other. The interpolation is based on the results developed in [12], where it was suggested that a certain subcomplex of the Delaunay triangulation of the given Pareto optimal outcomes should be used. Our surrogate problem allows the decision maker to consider approximate outcomes (i.e., interpolants of the given Pareto optimal outcomes produced by the PAINT method). In this paper, we also develop a mixed integer linear formulation for the surrogate problem that is implied by the interpolation. Replacing the original computationally expensive problem with the surrogate problem will naturally offer computational time savings when an interactive method is used.

Approaches similar to ours, where a Pareto front approximation is used to search for interesting outcomes not limited to a given set, have been proposed in [8, 19, 29]. The main difference between these approaches and ours is that the others use their own tailor-made methods for finding a vector containing preferred approximate outcomes on the approximations, while our approximation aims to work (through the mixed integer linear surrogate problem) in concert with any interactive method. Furthermore, [8, 29] are only applicable to convex multiobjective optimization problems, while ours can handle nonconvex problems. As mentioned before, PAINT has been developed to produce interpolate outcomes to be used with an interactive method through the surrogate problem. However, when looking at PAINT as a plain approximation method, it can be compared to some existing methods in the literature. This kind of approximation methods can be found in a survey [30] and in papers [2, 7, 9, 16, 21]. These methods, however, do not concentrate on the question of how to choose a Pareto optimal solution on the produced approximation as we do. Also, none of the above methods guarantee the property of inherent nondominance for nonconvex optimization problems with more than two objectives.

The rest of this paper is structured as follows: In Section 2, notation and definitions are given. The PAINT method and related computational issues are described in Section 3. In Section 4, the surrogate multiobjective optimization problem implied by the approximation is discussed. Section 5 shows examples of different Pareto front approximations to demonstrate the versatility of the PAINT method. Finally, conclusions are drawn in Section 6.

2 Notation and definitions

In this paper, we study multiobjective optimization problems

$$\begin{array}{l} \min\ (f_1(x), \ldots, f_k(x)) \\ \text{s.t.}\ \ x \in S, \end{array} \tag{1}$$

where the integer k is the number of real-valued objectives. The set S is called the feasible set. A vector $z = f(x) = (f_1(x), \ldots, f_k(x))^T$ with $x \in S$ is called an outcome. The set $f(S)$ is called the outcome set. We assume that the set S is connected and that the objective functions f_i are continuous. This assumption is made in order to justify interpolating between the objective function values of Pareto optimal solutions.

The set $\mathbb{R}^k$ is called the outcome space. For a vector z in the outcome space, a component z_i is called an objective value. Because problem (1) is a minimization problem, less is preferred to more in each objective. A vector z^1 in the outcome space is said to dominate another vector z^2 in the outcome space if z^1 is at least as good as z^2 in all objectives and strictly better in at least one. If z^1 dominates z^2 then it is written $z^1 \leq z^2$.

An outcome z^1 is said to be Pareto optimal, if there does not exist another outcome z^2 so that z^2 dominates z^1. The set of all Pareto optimal outcomes is called the Pareto front. In this paper, the set P always refers to the given set of Pareto optimal outcomes on which the Pareto front approximation is based. The number of these Pareto optimal outcomes is denoted by m and the elements of this set are denoted by p^j, where $j = 1, \ldots, m$. This set is called the initial set of Pareto optimal outcomes and it is taken as given, i.e., we assume that we cannot influence the locations of these outcomes in the outcome space.

Important elements in our PAINT method are polytopes and collections of polytopes for which we mostly follow the definitions given in [10]: For a nonnegative integer a, the convex hull of $z^1, \ldots, z^{a+1} \in \mathbb{R}^k$ is denoted by $\mathcal{P}(z^1, \ldots, z^{a+1})$ and is called an a-polytope. It is said that the polytope $\mathcal{P}(z^1, \ldots, z^{a+1})$ is defined by vectors $z^1, \ldots, z^{a+1}$. A vertex of a polytope K is a point $x \in K$ for which it holds that $\lambda z + (1-\lambda)y = x$, $\lambda \in (0,1)$ and $z, y \in K$ imply $x = y = z$. The set of vertices of a polytope K is denoted by $\mathrm{vert}(K)$. A face of a polytope K is another polytope K' so that $K' = \emptyset$, $K' = K$ or there exist vectors $z^1, z^2 \in \mathbb{R}^k$ so that $K' = \{z^1 + h : h \in \mathbb{R}^k, h^T z^2 = 0\} \cap K$. A polytope $\mathcal{P}(z^1, \ldots, z^{a+1})$ is a simplex if the vectors $z^1, \ldots, z^{a+1}$ are affinely independent.

Collections of polytopes are sets whose elements are polytopes and they are denoted by calligraphic letters $\mathcal{A}, \mathcal{B}, \mathcal{K}, \ldots$. The body of a collection of polytopes $\mathcal{A}$ is $\text{body}(\mathcal{A}) = \cup_{K \in \mathcal{A}} K$. A (polyhedral) complex $\mathcal{K}$ is a special type of collection of polytopes so that (a) if it holds that a polytope $K \in \mathcal{K}$ and another polytope K' is a face of K then it must hold that $K' \in \mathcal{K}$, and (b) if it holds that polytopes $K^1, K^2 \in \mathcal{K}$ then the set $K^1 \cap K^2$ must be a possibly empty face of both polytopes K^1 and K^2. A triangulation of a finite set $P \subset \mathbb{R}^k$ is a complex $\mathcal{T}$ so that the body of $\mathcal{T}$ is the convex hull of the set P and the set of vertices in $\mathcal{T}$ is P. A polytope $\mathcal{P}(z^1, \ldots, z^k)$ with $z^1, \ldots, z^k \in P$ is called Delaunay (in P) if there exists an open ball $B \in \mathbb{R}^k$ with $\text{cl}(B) \cap P = \{z^1, \ldots, z^{a+1}\}$ (here $\text{cl}(B)$ denotes the closure of B) and $B \cap P = \emptyset$. A complex $\mathcal{D}$ is a Delaunay triangulation of P if $\mathcal{D}$ is a triangulation of P so that every polytope in $\mathcal{D}$ is Delaunay. A polyhedral complex is called a simplicial complex, if all the polytopes are simplices.

In [13], the most important concepts for the theory behind our Pareto front approximation were given. First, a set $A \subset \mathbb{R}^k$ is called *inherently nondominated*, if there does not exist $a, b \in A$ so that $a \leq b$. Given a set of outcomes, one can check whether any of them dominates the others. Inherent nondominance is a more advanced concept used to guarantee that when a new infinite set of interpolating outcomes has been generated, none of them is dominated by or dominates others. In order to connect the inherently nondominated set to a known set of Pareto optimal outcomes P it was defined that an inherently nondominated set A is called an *inherently nondominated Pareto front approximation* (based on a set of outcomes P), if it holds that $P \subset A$. Since also the set P is by definition an inherently nondominated Pareto front approximation, it was further defined that for a set $B \subset \mathbb{R}^k$, an inherently nondominated Pareto front approximation $A \subset B$ is called a B-maximal inherently nondominated Pareto front approximation, if there does not exist an inherently nondominated set A' so that $A \subsetneq A' \subset B$. A B-maximal inherently nondominated Pareto front approximation is a maximal (by inclusion) collection of points from B that could be a part of a same Pareto front as the initial Pareto optimal outcomes in P are.

Since an inherently nondominated Pareto front approximation is to be used as an interpolation of the Pareto optimal outcomes in P, it usually contains an infinite number of points. On the other hand, such an infinite set can be represented by a finite number of polytopes. Thus, it is easier to construct such interpolations from polytopes than from points. Hence, the above concepts were in [12] slightly modified to fit collections of polytopes and polyhedral complexes. A collection of polytopes $\mathcal{K}$ is called inherently nondominated and an inherently nondominated Pareto front approximation, whenever its body, $\text{body}(\mathcal{K})$, is that. Finally, if $\mathcal{B}$ is a collection of polytopes, an inherently nondominated collection of polytopes $\mathcal{K}$ is called a $\mathcal{B}$-maximal inherently nondominated Pareto front approximation if there does not exist a polytope $K \in \mathcal{B} \setminus \mathcal{K}$ so that the collection of polytopes $\mathcal{K} \cup \{K\}$ is inherently nondominated. Thus, the difference between this definition and the corresponding definition for a set is that a $\mathcal{B}$-maximal inherently nondominated Pareto front approximation is a maximal collection of polytopes (not points) in $\mathcal{B}$ that could be a part of a same Pareto front as the initial Pareto optimal outcomes in P are. A member of the body of an inherently nondominated Pareto front approximation is called an *approximate Pareto optimal outcome*.

To construct inherently nondominated collections of polytopes, a dominance between polytopes was defined in [12]. Let $K^1, K^2 \subset \mathbb{R}^k$ be polytopes. It is defined that the polytope K^1 dominates the polytope K^2 if there exist vectors $s^1 \in K^1$ and $s^2 \in K^2$ so that s^1 dominates s^2. As shown in that paper, a complex $\mathcal{K} \subset \mathcal{D}$ is inherently nondominated if and only if there does not exist polytopes $K^1, K^2 \in \mathcal{K}$ so that K^1 dominates K^2.

3 The PAINT method

In this section, we introduce the algorithm of the PAINT method, to be called Algorithm 1. We describe data structures that can be used in implementing this algorithm. Based on these data structures, we illustrate how the steps of Algorithm 1 can be carried out. We also discuss the implementation aspects of the algorithm and the complexity of the PAINT method. Finally, we develop a way of resolving dominance between two polytopes, which is needed in Algorithm 1.

The PAINT method takes as input a set of m Pareto optimal outcomes P and its output is a $\mathcal{D}$-maximal inherently nondominated Pareto front approximation based on P, where $\mathcal{D}$ is a Delaunay triangulation of the set P. The method is based on the ideas presented in [12] and it guarantees that the approximation satisfies the following rules (R1) and (R2) given in [12]: assuming that l-polytopes in the Delaunay triangulation $\mathcal{D}$ are $K^{l,j}$ with $j = 1, \ldots, t^l$, a polytope $K^{l,j} \in \mathcal{D}$ is removed, if either

(R1) there exists an outcome $p \in P$ that dominates or is dominated by the polytope $K^{l,j}$ or the polytope $K^{l,j}$ dominates itself
OR

(R2) there exists an m-polytope $K^{m,j'} \in \mathcal{D}$ with either $m > l$ or $m = l$ and $j' < j$ that is not removed, and that dominates or is dominated by the polytope $K^{l,j}$.

In [12], it is shown that a resulting polyhedral complex after having applied rules (R1) and (R2) is a $\mathcal{D}$-maximal inherently nondominated Pareto front approximation.

In the implementation of the PAINT method presented in this paper, the Delaunay triangulation and the inherently nondominated Pareto front approximation are represented as matrices D and A, respectively. In those matrices, each row represents the vertices of a polytope with each entry referring to an outcome in P. For example a row with entries 1, 2 and 3 represents a polytope defined by outcomes p^1, p^2 and p^3 in P. Polytopes that are defined with fewer outcomes than there are columns in D or A are handled by repeating the same outcome multiple times. This way of representing the complexes enables us to write e.g., the outcomes defining the polytope represented by row j in the matrix D as $p^{D_{j,1}}, p^{D_{j,2}}, \ldots, p^{D_{j,b}}$ with b being the number of columns in D.

3.1 Algorithm

The PAINT method is described in Algorithm 1. Throughout the algorithm, indices a and b stand for the number of polytopes in the Delaunay triangulation and the maximal number of outcomes defining a polytopes in the Delaunay triangulation, respectively. The algorithm begins by reading in the initial set of Pareto optimal outcomes P. Line 2 of the algorithm is concerned with constructing a Delaunay triangulation of the set P. We propose to do this with the Quickhull algorithm [1], which constructs the Delaunay triangulation by building the convex hull of a higher dimensional related set (see [3]). The Quickhull algorithm assumes that the outcomes in P are in general position (see [1]), which is not always the case for e.g., linear multiobjective optimization problems for which the outcomes may be affinely dependent. This dependency can be removed by e.g., slightly perturbing the points as proposed in [6]. Whenever necessary, we have used perturbation in the examples in Section 5.

In line 3 of Algorithm 1, the rows of matrix D representing the Delaunay triangulation are ordered in a descending order with respect to the number of different entries in the row that is defined as the number of outcomes defining the polytope that is represented by the row. The ordering ensures that the polytopes are checked in the loop starting from line 5 in a descending order with respect to the number of outcomes defining the polytope.

In lines 6, 11 and 19 of the algorithm, there are three different if-conditions. The purpose of the conditions in lines 6 and 11 is to make sure that the final approximation follows rule (R1), and the purpose of the condition in line 19 is to make sure that the final approximation follows rule (R2). Conveniently, all these conditions reduce to checking dominance between two polytopes, which is the condition in line 19: (a) the inherent nondominance of a polytope in line 6 can be reduced to checking whether a polytope dominates itself, because in [12] it was proven that a polytope is inherently nondominated if and only if it does not dominate itself, and (b) the dominance between an outcome and a polytope in line 11 is actually dominance between two polytopes because a singleton containing a vector in $\mathbb{R}^k$ is by definition a polytope.Thus, it is adequate to build mathematical tools for determining dominance between two polytopes. This is done in Subsection 3.2 by using two linear optimization problems.

Assume now that the algorithm is in the loop starting from row 5 and ending with row 26 and that the index i value is $\hat{\imath}$. If the polytope given by row $\hat{\imath}$ in D is not inherently nondominated, it dominates or is dominated by an outcome in P or it dominates or is dominated by a polytope given by any of rows $d+1, \ldots, \hat{\imath}-1$ in the current D, then the polytope is removed from D in lines 7, 12 and 20, respectively. This is done by increasing d (the number of polytopes that have been already removed) and interchanging rows d and $\hat{\imath}$ in

Algorithm 1 PAINT method: Construction of the inherently nondominated Pareto front approximation

1: Read the Pareto optimal outcomes $P = \{p^1, p^2, \ldots, p^m\} \subset \mathbb{R}^k$.
2: $D \leftarrow$ the Delaunay triangulation of P
3: Sort the rows of D in descending order w.r.t. number of different entries in the row
4: $a, b \leftarrow$ the number of rows and columns of D, respectively, $d \leftarrow 0$
5: **for** $i = 1$ to a **do**
6:   **if** the polytope given by vertices $p^{D_{i,1}}, p^{D_{i,2}}, \ldots, p^{D_{i,b}}$ is not inherently nondominated **then**
7:     $d \leftarrow d + 1$ and interchange rows i and d of the matrix D
8:   **else**
9:     *deleted* $\leftarrow$ **false**
10:     **for** $j = 1$ to m **do**
11:       **if** the polytope given by vertices $p^{D_{i,1}}, p^{D_{i,2}}, \ldots, p^{D_{i,b}}$ dominates or is dominated by outcome p^j **then**
12:         $d \leftarrow d + 1$ and interchange rows i and d of the matrix D
13:         *deleted* $\leftarrow$ **true**
14:         Break
15:       **end if**
16:     **end for**
17:     **if** not *deleted* **then**
18:       **for** $l = d + 1$ to $i - 1$ **do**
19:         **if** the polytope given by vertices $p^{D_{i,1}}, p^{D_{i,2}}, \ldots, p^{D_{i,b}}$ dominates or is dominated by the polytope given by vertices $p^{D_{l,1}}, p^{D_{l,2}}, \ldots, p^{D_{l,b}}$ **then**
20:           $d \leftarrow d + 1$ and interchange rows i and d of the matrix D
21:           Break
22:         **end if**
23:       **end for**
24:     **end if**
25:   **end if**
26: **end for**
27: $A \leftarrow \begin{bmatrix} D_{d+1,1} & \ldots & D_{d+1,b} \\ \vdots & \vdots & \vdots \\ D_{a,1} & \ldots & D_{a,b} \end{bmatrix}$
28: **return** A

the matrix D. Because in line 27 only rows in D from row $d+1$ onwards are set as matrix A, such a polytope will not be a part of the approximation. If a polytope is not removed as described above, the row representing this polytope will be inserted into matrix A (representing the inherently nondominated Pareto front approximation) in line 27.

Finally, let us consider the complexity of the proposed implementation of Algorithm 1. According to [1], the worst case complexity of the construction of the Delaunay triangulation with the Quickhull algorithm is $O((k+1)\log(m))$ for $k \leq 2$ and $O((k+1)c_m/m)$ for $k \geq 3$, where

$$c_m = O(m^{\lfloor (k+1)/2 \rfloor} / \lfloor (k+1)/2 \rfloor !).$$

According to [22], a Delaunay triangulation of the set P contains at most $O(m^{\lceil k/2 \rceil})$ polytopes, where m is the number of outcomes in P and k is the number of objectives of the multiobjective optimization problem. Thus, one may have to resolve $O(m^k)$ dominance relations between polytopes to determine which of the polytopes are to be removed. This means that resolving the dominance relations between polytopes is computationally the most expensive part of the PAINT method. Since in Subsection 3.2 it is shown that the dominance between polytopes can be resolved by solving two linear optimization problem, the worst case complexity of the PAINT method is $O(m^k)$ in linear optimization problems.

The complexity of solving the linear optimization problems of Subsection 3.2 depends on the number of objectives and the number of variables. In these problems, the numbers of objectives and variables depend linearly on the maximal number of outcomes defining a polytope in the approximation and on the number of objectives. In practical problems, the number of objectives is usually rather low, and if one perturbs the Pareto optimal outcomes before constructing the Delaunay triangulation (as suggested in [6]), the maximal number of outcomes defining a polytope is the number of objectives plus one. Since these numbers are so low, also the linear optimization problems defined in Subsection 3.2 are fairly small and quick to solve. Consequently, we may here assume that in practice the complexity of solving a single linear optimization problem within the algorithm is low and independent of the numbers of objectives and variables. With current personal computers, a feasible number of objectives is about 10 and the maximal number of Pareto optimal outcomes is about 200.

3.2 Resolving dominance between two polytopes

According to [12], a polytope $K^1 \subset \mathbb{R}^k$ dominates another polytope $K^2 \subset \mathbb{R}^k$ (which needs to be resolved in lines 6, 11 and 19 of Algorithm 1) if and only if one of the following holds:

(i) the optimal value in problem

$$\begin{array}{ll} \min & \max_{i=1,\dots,k}(s_i^1 - s_i^2) \\ \text{s.t. } s^1 \in K^1, s^2 \in K^2 & \end{array} \tag{2}$$

is less than zero
OR

(ii) the optimal value in problem (2) is exactly zero and the optimal value in problem

$$\begin{array}{ll} \min & \sum_{i=1}^{k}(s_i^1 - s_i^2) \\ \text{s.t.} & s^1 \in K^1,\ s^2 \in K^2 \\ & s_i^1 \leq s_i^2 \text{ for all } i = 1,\dots,k. \end{array} \tag{3}$$

is less than zero.

In order to efficiently solve the above problems, it is beneficial to build linear programs that are equivalent to them. Assuming that the polytope K^1 is given by row r_1 in the matrix D (representing the Delaunay triangulation) and the polytope K^2 is given by row r_2 in D, we introduce the following matrix $B \in \mathbb{R}^{k \times 2b}$ (with b being the number of columns in D and k being the number of objectives of the problem)

$$B = \begin{bmatrix} p_1^{D_{r_1,1}} & p_1^{D_{r_1,2}} & \dots & p_1^{D_{r_1,b}} & -p_1^{D_{r_2,1}} & \dots & -p_1^{D_{r_2,b}} \\ p_2^{D_{r_1,1}} & p_2^{D_{r_1,2}} & \dots & p_2^{D_{r_1,b}} & -p_2^{D_{r_2,1}} & \dots & -p_2^{D_{r_2,b}} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ p_k^{D_{r_1,1}} & p_k^{D_{r_1,2}} & \dots & p_k^{D_{r_1,b}} & -p_k^{D_{r_2,1}} & \dots & -p_k^{D_{r_2,b}} \end{bmatrix}.$$

In matrix B, the first b columns include the components of the outcomes defining the polytope K^1 and the last b columns include the opposites of the components of the outcomes defining the polytope K^2. Using the matrix B we can give the following representation for problem (2):

$$\begin{aligned} \min \quad & t \\ \text{s.t.} \quad & [B, -\mathbf{1}] \begin{bmatrix} \lambda \\ \mu \\ t \end{bmatrix} \leq \begin{bmatrix} 0 \\ \vdots \\ 0 \end{bmatrix} \\ & \textstyle\sum_{j=1}^{b} \lambda_j = 1, \ \sum_{j=1}^{b} \mu_j = 1 \\ & \lambda \in [0,1]^b, \ \mu \in [0,1]^b, \ t \in \mathbb{R}, \\ \text{where} \quad & -\mathbf{1} = \begin{bmatrix} -1 \\ \vdots \\ -1 \end{bmatrix}. \end{aligned} \tag{4}$$

In a similar way, problem (3) can be formulated as

$$\begin{aligned} \min \quad & g \begin{bmatrix} \lambda \\ \mu \end{bmatrix} \\ \text{s.t.} \quad & B \begin{bmatrix} \lambda \\ \mu \end{bmatrix} \leq \begin{bmatrix} 0 \\ \vdots \\ 0 \end{bmatrix} \\ & \textstyle\sum_{j=1}^{b} \lambda_j = 1, \ \sum_{j=1}^{b} \mu_j = 1 \\ & \lambda \in [0,1]^b, \ \mu \in [0,1]^b, \\ \text{where} \quad & g = [p_1^{D_{r_1,1}} + \dots + p_k^{D_{r_1,1}}, \dots, p_1^{D_{r_1,b}} + \dots + p_k^{D_{r_1,b}}, \\ & - p_1^{D_{r_2,1}} - \dots - p_k^{D_{r_2,1}}, \dots, -p_1^{D_{r_2,b}} - \dots - p_k^{D_{r_2,b}}]. \end{aligned} \tag{5}$$

In both problems (4) and (5), the left hand side of the inequality constraint

$$B \begin{bmatrix} \lambda \\ \mu \end{bmatrix} = \sum_{j=1}^{b} \lambda_j p^{D_{r_1,j}} - \sum_{j=1}^{b} \mu_j p^{D_{r_2,j}}$$

gives a vector that is in the set $K^1 - K^2$, because the components of both vectors $\lambda, \mu \in [0,1]^b$ sum up to one. Furthermore, by definition, each element in the set $K^1 - K^2$ can be expressed in this way. Consequently, each row in the inequality constraint of problem (4) is of the form

$$\sum_{j=1}^{b} \lambda_j p_i^{D_{r_1,j}} - \sum_{j=1}^{b} \mu_j p_i^{D_{r_2,j}} \leq t$$

for some $i = 1, \dots, k$. Thus, minimizing t as the objective function is equivalent to minimizing the maximum of $\sum_{j=1}^{b} \lambda_j p_i^{D_{r_1,j}} - \sum_{j=1}^{b} \mu_j p_i^{D_{r_2,j}}$ over all i. This implies that problem (4) is equivalent to problem (2).

In problem (5), each row in the inequality constraint is of the form

$$\sum_{j=1}^{b} \lambda_j p_i^{D_{r_1,j}} \leq \sum_{j=1}^{b} \mu_j p_i^{D_{r_2,j}}$$

for some $i = 1, \dots, k$. The objective function becomes

$$g \begin{bmatrix} \lambda \\ \mu \end{bmatrix} = \sum_{l=1}^{k} \left(\sum_{j=1}^{b} \lambda_j p_l^{D_{r_1,j}} - \sum_{j=1}^{b} \mu_j p_l^{D_{r_2,j}} \right).$$

The inequality constraints and the objective function are thus the same as in problem (3).

Problems (4) and (5) can be solved with any linear programming solvers e.g., ILOG CPLEX (see `http://www.cplex.com/`) or GLPK (see `http://www.gnu.org/software/glpk/`) or with the MATLAB linprog function (see `http://www.mathworks.com/access/helpdesk/help/toolbox/optim/ug/linprog.html`).

4 Decision making with the approximation – a surrogate problem

In this section, we discuss how one can use interactive methods with the approximation created in the previous section. The main tool is the surrogate multiobjective optimization problem implied by the inherently nondominated Pareto front approximation. The surrogate problem was introduced in [12, 13]. In order to accomplish this, there are two issues that need to be addressed:

1. The scalarizations of the surrogate problem given by the interactive method at use need to be efficiently solvable in order to present the new computed solutions to the decision maker in real time.
2. Once a preferred approximate outcome on the Pareto front approximation has been found with the help of the surrogate problem and an interactive method, one must find the closest Pareto optimal outcome on the Pareto front of the original problem.

Our solution to the first issue is a multiobjective mixed integer linear formulation for the surrogate problem. Powerful mixed integer linear solvers, e.g., CPLEX and GLPK, allow then to efficiently solve scalarizations of the surrogate problem provided that the scalarization of the interactive method maintains linearity of the problem. This enables us to efficiently use many interactive methods e.g., the synchronous NIMBUS [23, 25, 27], because all its scalarizations can be formulated in a way that they maintain linearity.

In [12], the surrogate problem given by an inherently nondominated Pareto front approximation $\mathcal{A}$ was formulated as

$$\begin{aligned} &\min\ (z_1,\dots,z_k) \\ &\text{s.t.}\ \ z\in\cup_{K\in\mathcal{A}}K. \end{aligned} \tag{6}$$

The idea of the surrogate problem is that its outcomes are in the same space as the outcomes in P and, thus, the decision maker should be able to articulate his/her preferences about them when using an interactive method to solve problem (6), because the objectives have corresponding meanings in (6) and in the original problem..

Problem (6) cannot be inputted into standard multiobjective optimization solvers because it contains a non-algebraic constraint $z\in\cup_{K\in\mathcal{A}}K$. Problem (6), however, has an equivalent mixed integer linear formulation

$$\begin{array}{ll} \min & (z_1,\dots,z_k) \\ \text{s.t.} & \sum_{j=1}^{a}\sum_{l=1}^{b}\lambda_{j,l}=1 \\ & \sum_{l=1}^{b}\lambda_{j,l}\le y_j,\ \text{ for all } j=1,\dots,a \\ & \sum_{j=1}y_j=1 \\ & \lambda\in[0,1]^{a\times b} \\ & y\in\{0,1\}^a, \\ \text{where} & z_i=\sum_{j=1}^{a}\sum_{l=1}^{b}\lambda_{j,l}p_i^{A_{l,j}}\ \text{ for all } i=1,\dots,k. \end{array} \tag{7}$$

In multiobjective optimization problem (7), there are two variables $\lambda\in[0,1]^{a\times b}$ and $y\in\{0,1\}^a$. The component $\lambda_{j,l}$ of variable λ is for all $j=1,\dots,a$ and $l=1,\dots,b$ the coefficient of the vertex l of the polytope given by row j in the matrix A. The variable y determines which of the rows of matrix λ are nonzero. Note, that the third equality states that all but one component of the variable y must be zero. Matrix A refers to the representation of the inherently nondominated Pareto front approximation given by the PAINT method. According to the first equality constraint, all components $\lambda_{j,l}$ must sum up to one. According to the second inequality constraint, the entries in all but one row in the matrix representing the variable λ must some up to zero, because only one component of y nonzero. Thus, problem (7) is equivalent to problem (6).

As an example of scalarizations that maintain linearity we consider the achievement scalarizing problem [33], which can be formulated for multiobjective optimization problem (1) as

$$\min_{x\in S}\max_{i=1,\dots,k} w_i(f_i(x)-\bar{z}_i) \tag{8}$$

with the normalizing weight w in the positive orthant of $\mathbb{R}^k$ and the reference point $\bar{z} \in \mathbb{R}^k$ representing desirable objective values. For example, with the CPLEX solver and using an Acer laptop with Core 2 Duo P8700 processor and 4 GB of RAM, we solved the achievement scalarizing problem formulated for a surrogate problem implied by an inherently nondominated Pareto front approximation with the number of objectives $k = 5$, number of initial points $m = 330$, number of polytopes $a = 1824$ and the maximal number of outcomes defining a polytope $b = 5$ in approximately 3 seconds. We consider this Pareto front approximation to be as large as we need for decision making. We also consider the computational time of 3 seconds to be small enough that the formulations and tools of this section can be used with e.g., the NIMBUS method to compute new approximate Pareto optimal outcomes for the decision maker. These results show promise for further applicability of the PAINT method.

The surrogate problem, while useful in solving computationally expensive problems, also has some limitations. First, the problem does not provide information about the preimage of the Pareto front in the decision space. Second, the outcomes of the surrogate problem include all the inherently nondominated interpolations between the Pareto optimal outcomes and, thus, the method does not detect any possible areas where the Pareto front is disconnected. The first limitation is an issue if the decision variables are very meaningful to the decision maker in making decisions. This may sometimes be the case in engineering problems, even though the opposite is argued in [17]. The second limitation is an issue if the Pareto front has big areas of discontinuity and the decision maker is especially interested in those areas in the outcome space. These are topics of further research on this subject.

Finally, when a preferred approximate Pareto optimal outcome has been found, we must find the decision vector in the decision space S that gives the outcome on the actual Pareto front of the original problem that is the best match to it. As described above, we propose to solve the achievement scalarizing problem formulated for the original problem with the preferred approximate Pareto optimal outcome as the reference point. If the resulting outcome on the actual Pareto front is satisfactory to the decision maker, we stop. If we find out that the approximate Pareto optimal outcome was satisfactory to the decision maker but the actual Pareto optimal outcome was not, then we must update the inherently nondominated Pareto front approximation. This is done by adding the new Pareto optimal outcome to the set P and running the PAINT method again. This yields a Pareto front approximation that is more accurate close to the new Pareto optimal outcome. Note, however, that both of these operations (finding the closest element on the actual Pareto front and rerunning the PAINT method) take time. Hence, repeating them multiple times should be avoided, if possible.

5 Examples

In this section, we present examples of inherently nondominated Pareto front approximations constructed with the PAINT method described in Algorithm 1. The implementation of Algorithm 1 was written under GNU Octave [5]. Delaunay triangulations of the finite sets of Pareto optimal outcomes were constructed with a Qhull implementation (see `http://www.qhull.org/`) of the Quickhull algorithm [1]. Optimization problems (4) and (5) were solved with GLPK (GNU Linear Programming Kit, see `http://www.gnu.org/software/glpk/`). This all was done on an Acer laptop with Core 2 Duo P8700 processor and 4 GB of RAM running Fedora 12.

The different example problems with multiple objectives have been selected to demonstrate different types of Pareto fronts and to show that the PAINT method is suitable for approximating all of them. The problems in Subsections 5.1, 5.2 and 5.3 are test problems that are used for testing evolutionary multiobjective optimization algorithms. These problems are not computationally expensive but have been selected because their Pareto front has an interesting geometry to depict in three dimensions. Let us point out that even though the PAINT method has been developed for enabling faster solution processes for computationally expensive multiobjective optimization problems, it can naturally be applied to computationally inexpensive problems. This is because the computational expense of the problem does not play any role within the PAINT method after the initial set of Pareto optimal outcomes has been computed. The problem in Subsection 5.4 is a real application of multiobjective optimization to wastewater treatment planning. With it, we demonstrate how the PAINT method can be used with real-life applications.

We emphasize that the purpose of the examples is not to illustrate the decision making aspect (including the surrogate problem), which has to be studied with real-life problems and with real decision makers. As mentioned, here we want to demonstrate how PAINT works as an approximation method. Naturally, the benefits of the surrogate problem will better show up with problems with more than three objectives, for which one cannot graphically present the Pareto front nor its approximation.

5.1 The three-objective Viennet's test problem

The Pareto front of the three-objective Viennet's test problem [32] consists of a one-dimensional curve in $\mathbb{R}^3$. Figure 1a illustrates 240 Pareto optimal outcomes for this problem generated with a local search assisted evolutionary multiobjective optimization algorithm [31]. This figure is drawn to give an understanding of what the Pareto front looks like. If we dealt with a computationally expensive multiobjective optimization problem, the generation of this large set would be too time-consuming.

Table 1 lists a subset of 20 Pareto optimal outcomes $\{p^1, p^2, \dots, p^{20}\}$ which are randomly selected from the Pareto optimal outcomes in Figure 1a. This

Table 1: The initial set of Pareto optimal outcomes P for the Viennet's test problem

Outcome	z_1	z_2	z_3
p^1	0.030032	0.880533	0.991796
p^2	0.000000	1.000000	-0.628862
p^3	1.000000	0.894206	0.058335
p^4	0.929845	0.893301	0.062197
p^5	0.029038	0.880566	1.000000
p^6	0.001497	0.955310	-0.588038
p^7	0.111630	0.881325	0.471040
p^8	0.451435	0.888326	0.124438
p^9	0.000529	0.972149	-0.616055
p^{10}	0.602037	0.887156	0.097756
p^{11}	0.013991	0.895472	0.014610
p^{12}	0.128648	0.881001	0.450349
p^{13}	0.000014	0.995955	-0.628542
p^{14}	0.002388	0.945419	-0.557775
p^{15}	0.262487	0.882276	0.212985
p^{16}	0.845441	0.889814	0.070032
p^{17}	0.386630	0.883861	0.143005
p^{18}	0.004519	0.928747	-0.471886
p^{19}	0.000457	0.974112	-0.617907
p^{20}	0.005243	0.924974	-0.439333

subset is taken as the initial set of Pareto optimal outcomes P for the problem. These Pareto optimal outcomes are also illustrated in Figure 1b. Notice that the vectors are distributed rather non-uniformly. If we dealt with a computationally expensive multiobjective optimization problem, generating this set would obviously be a lot less time-consuming than generating the outcomes in Figure 1a.

Figure 1c shows the body of a Delaunay triangulation of the set P, which contains 395 polytopes. As it can be seen, the body of the Delaunay triangulation covers the whole convex hull of P. By comparing Figures 1a and 1c one can see that the Delaunay triangulation as such does not give a good approximation of the Pareto optimal outcomes because its body contains many vectors that are very far from the Pareto optimal outcomes.

After the Delaunay triangulation has been constructed the inappropriate polytopes are removed from it. In this case, the inherently nondominated Pareto front approximation is represented (as described in Section 3) by the matrix

$$
\begin{bmatrix}
1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 & 10 & 11 & 12 & 13 & 14 & 15 & 16 \\
1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 & 10 & 11 & 12 & 13 & 14 & 15 & 16 \\
1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 & 10 & 11 & 12 & 13 & 14 & 15 & 16 \\
1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 & 10 & 11 & 12 & 13 & 14 & 15 & 16
\end{matrix}
$$

$$
\begin{matrix}
17 & 18 & 19 & 20 & 1 & 1 & 1 & 1 & 2 & 3 & 3 & 4 & 5 & 6 & 6 & 7 \\
17 & 18 & 19 & 20 & 5 & 7 & 11 & 12 & 13 & 4 & 16 & 16 & 11 & 9 & 14 & 12 \\
17 & 18 & 19 & 20 & 1 & 1 & 1 & 1 & 2 & 3 & 3 & 4 & 5 & 6 & 6 & 7 \\
17 & 18 & 19 & 20 & 1 & 1 & 1 & 1 & 2 & 3 & 3 & 4 & 5 & 6 & 6 & 7
\end{matrix}
$$

$$
\begin{matrix}
8 & 8 & 9 & 10 & 10 & 11 & 12 & 13 & 14 & 15 & 18 & 1 & 1 & 3 & 8 \\
10 & 17 & 19 & 16 & 17 & 20 & 15 & 19 & 18 & 17 & 20 & 5 & 7 & 4 & 10 \\
8 & 8 & 9 & 10 & 10 & 11 & 12 & 13 & 14 & 15 & 18 & 11 & 12 & 16 & 17 \\
8 & 8 & 9 & 10 & 10 & 11 & 12 & 13 & 14 & 15 & 18 & 1 & 1 & 3 & 8
\end{bmatrix}^T
$$

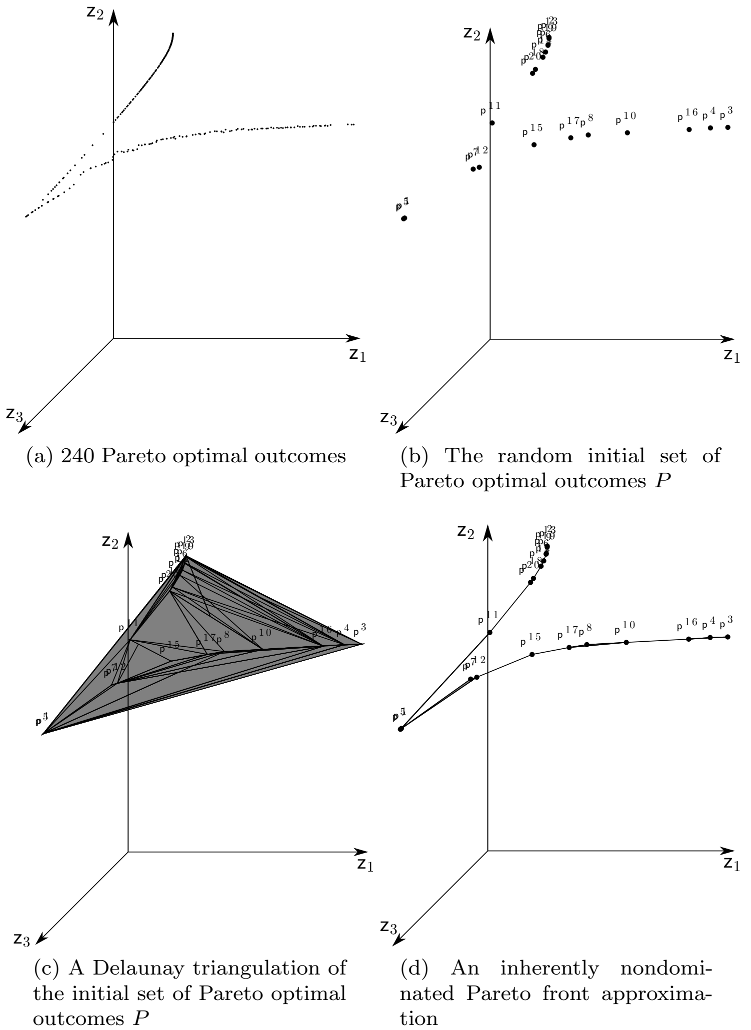
Notice that the matrix has been transposed and cut into three parts to save space. The body of the this approximation is shown in Figure 1d. Comparing this figure with Figure 1c one may notice that this approximation is a subset of the Delaunay triangulation. By comparing Figures 1a and 1d, one can see that the approximation is able to describe also many of the Pareto optimal outcomes in Figure 1a but not in the set P.

All in all, 88.1% of the polytopes in the Delaunay triangulation were removed in the PAINT method. All of these were removed due to rule (R1), because 58.7% of the polytopes were removed as they were not inherently nondominated and, after this, 29.4% of the polytopes were removed as they dominated or were dominated by an outcome in P. The PAINT method took 2.3 seconds on the configuration described above. The $\mathcal{D}$-maximal Pareto front approximation contained 47 polytopes.

5.2 The three-objective DTLZ2 test problem

The Pareto front of the three-objective DTLZ2 test problem [4] consists of the subset of the unit sphere that is in the positive orthant of $\mathbb{R}^k$. A set of 200 Pareto optimal outcomes can be seen in Figure 2a. Again, this set would be time-consuming to produce for a computationally expensive multiobjective optimization problem. As in the case of Viennet's test problem, the Figure 2b illustrates a random subset of this consisting of 20 outcomes.

Figure 2c depicts an inherently nondominated Pareto front approximation which is the output of the PAINT method. Since the Pareto front of this problem is a two-dimensional part of the surface of the ball, the body of the inherently nondominated Pareto front approximation is two-dimensional even though no information about the dimensionality was given to the algorithm. As it should be, the approximation can again describe also the Pareto optimal outcomes that are not in the initial set of Pareto optimal outcomes P.



(a) 240 Pareto optimal outcomes

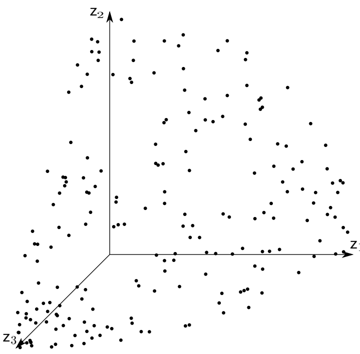
(b) The random initial set of Pareto optimal outcomes P

(c) A Delaunay triangulation of the initial set of Pareto optimal outcomes P

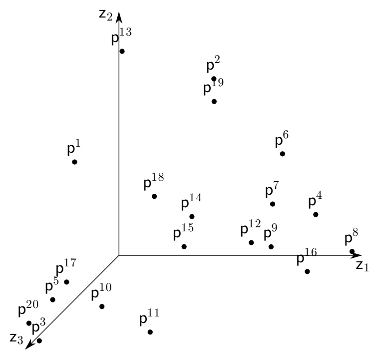
(d) An inherently nondominated Pareto front approximation

Fig. 1: Approximating the Pareto front of the three-objective Viennet's test problem

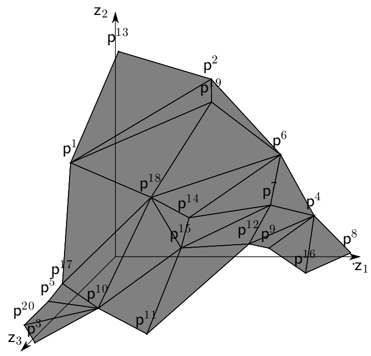
In this case, 56.8% of the polytopes in the Delaunay triangulation were removed due to (R1) as 33.6% of the polytopes were not inherently nondominated and 23.2% were inherently nondominated, but dominated or were dominated by an outcome in P. Of the polytopes that were not removed due to rule (R1), 31.8% were removed due to rule (R2). In total, the PAINT method took 8.3 seconds. The final $\mathcal{D}$-maximal inherently nondominated Pareto front approximation given by PAINT contained 85 polytopes.



(a) 200 Pareto optimal outcomes



(b) The random initial set of Pareto optimal outcomes P



(c) An inherently nondominated Pareto front approximation

Fig. 2: Approximating the Pareto front of the three-objective DTLZ2 test problem

5.3 The four-objective DTLZ2 test problem

The four-objective DTLZ2 test problem is similar to the three-objective one but with one more objective. That is, the Pareto front of this problem consists of a subset of the surface of the unit ball that is in the positive orthant of $\mathbb{R}^4$.

Table 2 lists the initial Pareto optimal outcomes in P. These outcomes are randomly generated vectors on the Pareto front of this problem.

Of the polytopes in the Delaunay triangulation 38.8% were removed due to rule (R1) as 19.8% were not inherently nondominated and 19.0% were inherently nondominated, but dominated or were dominated by an outcome in P. Furthermore, 34.2% of the polytopes that were not removed due to rule (R1) were removed due to rule (R2). Here the PAINT method took 111.2 seconds. The $\mathcal{D}$-maximal inherently nondominated Pareto front approximation given by PAINT contained 148 polytopes. The body of the inherently

Table 2: The given set of Pareto optimal outcomes P for the four-objective DTLZ2 test problem

Outcome	z_1	z_2	z_3	z_4
p^1	0.1712	0.7377	0.6467	0.0909
p^2	0.4399	0.2015	0.4218	0.7668
p^3	0.2457	0.1934	0.6819	0.6613
p^4	0.7375	0.3556	0.2103	0.5343
p^5	0.2430	0.6618	0.6582	0.2641
p^6	0.4781	0.6984	0.0493	0.5304
p^7	0.0952	0.4074	0.5114	0.7506
p^8	0.5689	0.3861	0.7032	0.1810
p^9	0.1823	0.0621	0.7022	0.6854
p^{10}	0.3304	0.5462	0.5204	0.5672
p^{11}	0.3074	0.7199	0.2407	0.5738
p^{12}	0.5578	0.3268	0.3376	0.6842
p^{13}	0.3685	0.5472	0.4703	0.5861
p^{14}	0.3598	0.4594	0.6085	0.5378
p^{15}	0.5401	0.1739	0.6530	0.5016
p^{16}	0.0395	0.2779	0.7468	0.6030
p^{17}	0.7778	0.4705	0.4110	0.0682
p^{18}	0.5801	0.6938	0.1218	0.4089
p^{19}	0.6200	0.2287	0.4903	0.5682
p^{20}	0.5681	0.3916	0.6646	0.2868

nondominated Pareto front approximation can be seen in Figure 3. In the figure the approximation has been projected to $\mathbb{R}^3$ by the projection function $p : \mathbb{R}^4 \to \mathbb{R}^3, (x_1, x_2, x_3, x_4) \mapsto (x_1, x_2, x_3)$ and the fourth objective is marked with colour.

5.4 Wastewater treatment planning

Finally, we have a real life multiobjective optimization problem of wastewater treatment planning from [11]. The problem considers the so-called activated sludge process, which is globally the most common method of wastewater treatment. In this process, biomass (which is called activated sludge) suspended in the wastewater to be treated is cultivated and maintained in an aerated bioreactor. The wastewater is purified, i.e., organic carbon, nitrogen and phosphorus are removed during its retention in the bioreactor. The bioreactor is followed by a clarifier basin, in which the biomass is separated by gravitational settling and returned to the bioreactor, and the treated wastewater is directed as overflow to further treatment or to discharge. Excess activated sludge is removed from the process and treated separately. The process performs nitrification, i.e., oxidation of ammonium nitrogen to nitrate nitrogen by autotrophic, slow-growing micro-organisms. The biochemical reactions involved consume a lot of oxygen and alkalinity. Oxygen is supplied by aeration compressors and alkalinity partly by influent wastewater, partly by adding chemicals, e.g., Na_2CO_3.

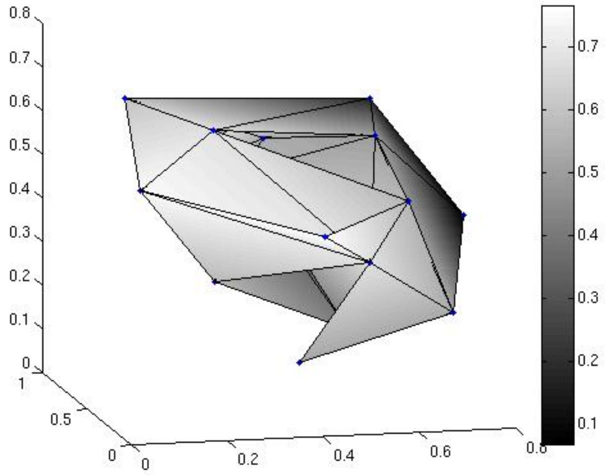


Fig. 3: A projection of an inherently nondominated Pareto front approximation for the 4-objective DTLZ2 test problem. The value of the fourth objective is marked with colour.

Aeration consumes energy and chemicals cost money, so minimizing the need for aeration and alkalinity addition is important for the operational economy of the plant.

The considered multiobjective optimization problem is computationally expensive and no closed form equations are known for the objectives. In [11], the process was simulated and a global solver was used to find Pareto optimal solutions with the interactive method NIMBUS. Because the problem is computationally expensive, it took a long time to find new Pareto optimal solutions. Hence, this problem has room for improvement with our methods.

We considered the 11 solutions generated with NIMBUS in [11]. One of the 11 solutions was dominated by another solution in the set and, thus, it was dropped. We assumed that the ten remaining solutions were Pareto optimal, which was in fact uncertain because a global solver was used to optimize the scalarizations of the multiobjective optimization problem that were given by the interactive method NIMBUS. The outcomes implied by these solutions (shown in Table 3) were given to the PAINT method as input P. The PAINT method took 2.5 seconds and the final inherently nondominated Pareto front approximation contained 37 polytopes. The body of the Pareto front approximation is shown in Figure 4.

In computational tests, the achievement scalarizing problem formulated for problem (7) takes 0.01 seconds to be solved with CPLEX for various values of reference points and weights. After a preferred solution to the surrogate problem has been found, one can project the outcome on the Pareto front

Table 3: The given set of Pareto optimal outcomes P for the multiobjective wastewater problem

Outcome	residual ammonium nitrogen concentration $[gN/m^3]$	alkalinity chemical dosing rate $[m^3/d]$	aeration energy consumption $[kW]$
p^1	8.05	218	460
p^2	3.52	286	490
p^3	1.69	326	506
p^4	4.9	298	477
p^5	1.11	336	515
p^6	0.55	347	528
p^7	9.36	246	448
p^8	30.2	7.23	308
p^9	0.9	333	519
p^{10}	0.72	332	524

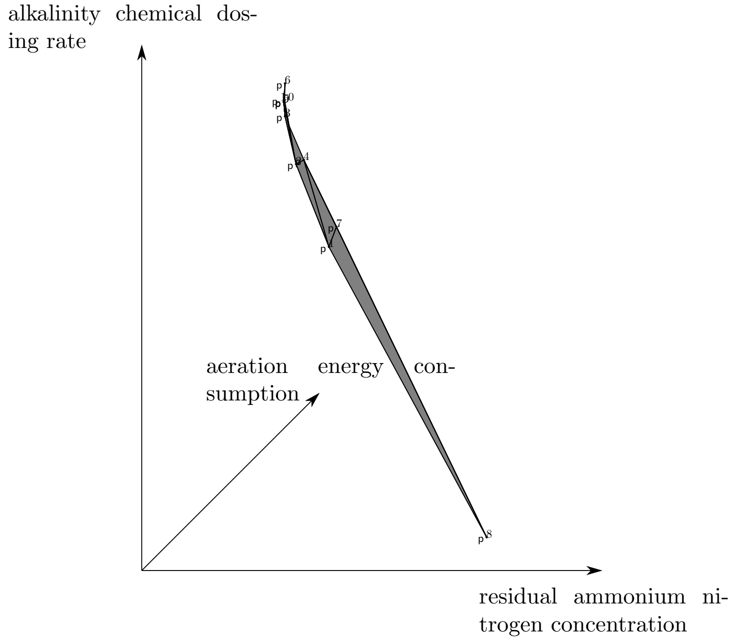


Fig. 4: An inherently nondominated Pareto front approximation for the multiobjective wastewater treatment planning problem

by solving globally the achievement scalarizing problem of the reference point method with the preferred solution to the surrogate problem as the preferred solution.

In this case, the initial set of Pareto optimal outcomes was not the best possible for our approximation because, as can be seen from Figure 4, the outcome p^8 is very far from the other outcomes. This may lead to inaccuracy of the approximation in interpolating between outcome p^8 and the others.

If one had used the PAINT method in [11], the decision making process would have been much faster with the interactive method. Because with the help of the surrogate problem one could have provided the decision maker quickly with new approximate Pareto optimal outcomes that correspond to his/her preferences, the decision maker should have been more inclined to further explore the Pareto front. This could have led the decision maker to get more insight about the problem.

6 Conclusions

We have demonstrated how one can in practice generate a Pareto front approximation by interpolating between a small finite set of given Pareto optimal outcomes. The interpolation is produced by the PAINT method proposed in this paper. The theory behind this method is based on a property called inherent nondominance previously introduced by the authors. In this paper, we have described a way of implementing the method and presented computational results on interpolating on Pareto fronts with the implementation. We have also illustrated how the Pareto front approximation implies a mixed integer linear surrogate problem for the original one that can be used in decision making concerning the original problem. This approach is especially useful with computationally expensive multiobjective optimization problems.

The combination of the following two benefits of our approach make it novel: (1) the proposed Pareto front approximation method can approximate non-convex Pareto fronts, and (2) any interactive method can be used in decision making with the proposed Pareto front approximation because of the mixed integer linear surrogate problem that it implies. These benefits make our approach applicable to different computationally expensive multiobjective optimization problems and enable it to support various decision makers.

As an example of interactive methods that can be applied with the Pareto front approximation produced, we have considered achievement scalarizing function based approaches. The achievement scalarizing problem formulated for the presented surrogate problem can be solved e.g., with the CPLEX linear solver. This enables us to use an interactive method based on decision maker's preferences formulated as a reference point, like the NIMBUS method. While the surrogate problem is a multiobjective optimization problem that can in theory be used with any interactive method, the implementations of the methods may restrict its use because the surrogate problem contains integer variables.

Further research on this subject includes applying the PAINT method with real-life problems and real decision makers. We can use the interface of IND-NIMBUS [24] (see `http://ind-nimbus.it.jyu.fi/`) and solve the scalarized problems of the mixed integer surrogate problem with the CPLEX solver. In this way, we can study the performance of the PAINT method in decision making.

Acknowledgements

We thank Mr. Karthik Sindhya for providing the Pareto optimal outcomes for the Viennet's test problem. We also thank Dr. Jussi Hakanen for his help with the wastewater treatment planning problem.

References

1. C. B. Barber, D. P. Dobkin, and H. Huhdanpää. The Quickhull Algorithm for Convex Hulls. *ACM Transactions on Mathematical Software*, 22:469–483, 1996.
2. V. E. Bezerkin, G. K. Kamenev, and A. V. Lotov. Hybrid Adaptive Methods for Approximating a Nonconvex Multidimensional Pareto Frontier. *Computational Mathematics and Mathematical Physics*, 46:1918–1931, 2006.
3. K. Q. Brown. Voronoi Diagrams from Convex Hulls. *Information Processing Letters*, 9:223–228, 1979.
4. K. Deb, L. Thiele, M. Laumanns, and E. Zitzler. Scalable Multi-Objective Optimization Test Problems. *IEEE International Conference on E-Commerce Technology*, 1:825–830, 2002.
5. J. W. Eaton. *GNU Octave Manual.* Network Theory Limited, 2002.
6. H. Edelsbrunner and E. P. Mücke. Simulation of Simplicity: a Technique to Cope with Degenerate Cases in Geometric Algorithms. *ACM Transactions on Graphics*, 9:66–104, 1990.
7. R. V. Efremov and G. K. Kamenev. Properties of a Method for Polyhedral Approximation of the Feasible Criterion Set in Convex Multiobjective Problems. *Annals of Operations Research*, 166:271–279, 2009.
8. P. Eskelinen, K. Miettinen, K. Klamroth, and J. Hakanen. Pareto Navigator for Interactive Nonlinear Multiobjective Optimization. *OR Spectrum*, 32:211–227, 2010.
9. T. Goel, R. Vaidyanathan, R. T. Haftka, W. Shyy, N. V. Queipo, and K. Tucker. Response Surface Approximation of Pareto Optimal Front in Multi-Objective Optimization. *Computer Methods in Applied Mechanics and Engineering*, 196:879–893, 2007.
10. B. Grünbaum. *Convex Polytopes.* Interscience Publishers, London, 1967.
11. J. Hakanen, K. Miettinen, and K. Sahlstedt. Wastewater Treatment: New Insight Provided by Interactive Multiobjective Optimization. *Decision Support Systems*, 51:328–337, 2011.
12. M. Hartikainen, K. Miettinen, and M. M. Wiecek. Constructing a Pareto Front Approximation for Decision Making. *Mathematical Methods of Operations Research*, 73:209–234, 2011.

13. M. Hartikainen, K. Miettinen, and M. M. Wiecek. Pareto Front Approximations for Decision Making with Inherent Non-dominance. In Y. Shi, S. Wang, G. Kou, and J. Wallenius, editors, *New State of MCDM in the 21st Century, Selected Papers of the 20th International Conference on Multiple Criteria Decision Making 2009*, pages 35–46. Springer-Verlag Berlin, Heidelberg, 2011.
14. M. Hasenjäger and B. Sendhoff. Crawling Along the Pareto Front: Tales From the Practice. In *The 2005 IEEE Congress on Evolutionary Computation (IEEE CEC 2005)*, pages 174–181, Piscataway, NJ, 2005. IEEE Press.
15. C. Hwang and A. S. M. Masud. *Multiple Objective Decision Making - Methods and Applications: a State-of- the-Art Survey.* Springer-Verlag Berlin, Heidelberg, 1979.
16. G. K. Kamenev. Study of an Adaptive Single-Phase Method for Approximating the Multidimensional Pareto Frontier in Nonlinear Systems. *Computational Mathematics and Mathematical Physics*, 49:2103–2113, 2009.
17. R. L. Keeney. *Value-Focused Thinking: a Path to Creative Decisionmaking.* Harward University Press, Harward, 1996.
18. T. Laukkanen, T.-M. Tveit, V. Ojalehto, K. Miettinen, and C.-J. Fogelholm. An Interactive Multi-Objective Approach to Heat Exchanger Network Synthesis. *Computers & Chemical Engineering*, 34:943–952, 2010.
19. A. V. Lotov, V. A. Bushenkov, and G. A. Kamenev. *Interactive Decision Maps.* Kluwer Academic Publishers, Boston, 2004.
20. M. Luque, F. Ruiz, and K. Miettinen. Global Formulation for Interactive Multiobjective Optimization. *OR Spectrum*, 33:27–48, 2011.
21. J. Martin, C. Bielza, and D. R. Insua. Approximating Nondominated Sets in Continuous Multiobjective Optimization Problems. *Naval Research Logistics*, 52:469–480, 2005.
22. P. McMullen. The Maximum Number of Faces of a Convex Polytope. *Mathematika*, 17:179–184, 1970.
23. K. Miettinen. *Nonlinear Multiobjective Optimization.* Kluwer Academic Publishers, Boston, 1999.
24. K. Miettinen. IND-NIMBUS for Demanding Interactive Multiobjective Optimization. In T. Trzaskalik, editor, *Multiple Criteria Decision Making'05*, pages 137–150. The Karol Adamiecki University of Economics in Katowice, Katowice, 2006.
25. K. Miettinen and M. Mäkelä. Interactive Bundle-based Method for Nondifferentiable Multiobjective Optimization: NIMBUS. *Optimization*, 34:231–246, 1995.
26. K. Miettinen and M. M. Mäkelä. On Scalarizing Functions in Multiobjective Optimization. *OR Spectrum*, 24:193–213, 2002.
27. K. Miettinen and M. M. Mäkelä. Synchronous Approach in Interactive Multiobjective Optimization. *European Journal of Operational Research*, 170:909–922, 2006.
28. K. Miettinen, F. Ruiz, and A. P. Wierzbicki. Introduction to Multiobjective Optimization: Interactive Approaches. In J. Branke, K. Deb, K. Miet-

tinen, and R. Slowinski, editors, *Multiobjective Optimization: Interactive and Evolutionary Approaches*, pages 27–57. Springer-Verlag Berlin, Heidelberg, 2008.
29. M. Monz. *Pareto Navigation - Algorithmic Foundation of Interactive Multi-criteria IMRT Planning*. PhD thesis, University of Kaiserslautern, 2006.
30. S. Ruzika and M. M. Wiecek. Approximation Methods in Multiobjective Programming. *Journal of Optimization Theory and Applications*, 126:473–501, 2005.
31. K. Sindhya, K. Deb, and K. Miettinen. Improving Convergence of Evolutionary Multi-Objective Optimization with Local Search: a Concurrent-Hybrid Algorithm. *Natural Computing*, To Appear. DOI:10.1007/s11047-011-9250-4.
32. R. Viennet, C. Fonteix, and I. Marc. Multicriteria Optimization Using a Genetic Algorithm for Determining a Pareto Set. *International Journal of Systems Science*, 27:255–260, 1996.
33. A. P. Wierzbicki. On the Completeness and Constructiveness of Parametric Characterizations to Vector Optimization Problems. *OR Spectrum*, 8:73–87, 1986.