Int. J. Appl. Math. Comput. Sci., 2009, Vol. 17, No. , –
DOI:

amcs

VERIFIED METHODS FOR COMPUTING PARETO SETS: GENERAL ALGORITHMIC ANALYSIS

BOGLÁRKA G.-TÓTH *, VLADIK KREINOVICH **

* Department of Differential Equations, Institute of Mathematics
Budapest University of Technology and Economics (BME)
Egry József u. 1, 1111 Budapest, Hungary
bog@math.bme.hu

** Department of Computer Science
University of Texas at El Paso, 500 W. University
El Paso, Texas 79968, USA, vladik@utep.edu
http://www.cs.utep.edu/vladik

In many engineering problems, we face multi-objective optimization, with several objective functions $f_1, \ldots, f_n$. We want to provide the user with the Pareto set – set of all possible solutions x which cannot be improved in all categories (i.e., for which $f_j(x') \geq f_j(x)$ for all j and $f_j(x') > f_j(x)$ for some j is impossible). The user should be able to select an appropriate trade-off between, say, cost and durability. We extend the general results about the (verified) algorithmic computability of maxima locations to show that Pareto sets can also computed.

Keywords: Multi-objective optimization, Pareto set, verified computing.

1. Introduction

In engineering problems, we are usually interested in finding the solution which is the best under given constraints. In many practical problems, an objective function $f(x)$ is explicitly given. In this case, "the best" means that we want to find a solution which maximizes the value of this objective function, i.e., a solution x for which the value $f(x)$ cannot be improved – i.e., for which the inequality $f(x') > f(x)$ is impossible.

Usually, if there are several such optimizing solutions, the user will be able to select a one with the largest possible value of some other important objective function. For example, if we have several plant designs x with the same expected profit $f(x)$, it may be reasonable to select the most environmentally friendly of these design. In view of the possibility (and importance) of the additional user choice, it is desirable not just to present the user with a single optimizing solution x, but rather present the user with the entire *set* of all possible optimizing solutions.

In many practical situations, there are efficient algorithms for computing this optimizing set. However, it is known that in general, the problem of computing the optimizing set is not algorithmically decidable (see, e.g., (Kreinovich, Lakeyev, Rohn, and Kahl, 1998)). This undecidability result is caused not so much by the complexity of the problem, but rather by the idealization that we made when we assumed that we know the *exact* expression $f(x)$ for the objective function. Of course, in practice, we rarely know such an expression. Usually, the known expression $\widetilde{f}(x)$ describes the actual (unknown) objective function $f(x)$ only *approximately*, with some accuracy $\varepsilon > 0$: $|f(x) - \widetilde{f}(x)| \leq \varepsilon$. In this case, the only information that we have about the actual objective function $f(x)$ is that for every x, its value belongs to the interval $\boldsymbol{f}(x) \stackrel{\text{def}}{=} [\widetilde{f}(x) - \varepsilon, \widetilde{f}(x) + \varepsilon]$. Different objective functions $f(x)$ from this "function interval" attain their maxima, in general, at different points x. It is therefore reasonable to provide the user with the set of all possible optimizing solutions corresponding to all possible functions $f(x) \in \boldsymbol{f}(x)$. As we will show in this paper, this set *can* be algorithmically computed – if we take into account that the accuracy ε is also not exactly known.

The above description is still somewhat idealized, because it assumes that we have a *single* objective function

that we are trying to maximize – albeit an imprecisely known one. In other words, we assume that we have already agreed how to combine different characteristics describing different aspects of the problem into a single numerical quantity. In practice, we usually have several objective functions $f_1(x), \ldots, f_n(x)$ describing different aspects of the possible solution x, such as profit, environmental friendliness, safety, etc. Ideally, we should maximize the values of *all* these characteristics, but in reality, there is often a trade-off: e.g., to achieve more environmental friendliness, it is often necessary to slightly decrease the profit; there is a similar trade-off between cost and durability.

In many situations, the user does not have a clear a priori idea which trade-offs are beneficial and which are not; in other words, the user does not have a single combined objective function $f(x)$ that would enable him or her to make an ultimate decision. In such situations, it is reasonable to present the user with the *set* of all possible solutions – and let the user decided between different possible solutions from this set. The only possible solutions x that we do not want to present to the user are solutions x which can be improved in all the senses, i.e., solutions for which, for some other solution x', we have $f_j(x) \le f_j(x')$ for all j and $f_j(x) < f_j(x')$ for some j. The set of all such "non-improvable" solution is known as the *Pareto set*. The problem is how to compute the Pareto set.

This problem is known to be computationally difficult; see, e.g., (Ruzika and Wiecek, 2005). Efficient algorithms are only known for specific classes of problems: e.g., for special location problems (Nickel and Puerto, 2005) and for problems with linear objective functions (Figueira, Greco, and Ehrgott, 2004). This difficulty has an explanation: in the above idealized formulation, when we know the *exact* expressions for all the objective functions $f_1(x), \ldots, f_n(x)$, this problem becomes, in general, algorithmically unsolvable.

In practice, as we have mentioned, we know each of these functions $f_j(x)$ only with some accuracy ε_j. It turns out that if we appropriately take this uncertainty into account, then (verified) algorithms for computing the resulting Pareto set become possible. Such algorithms were described, for the case of $n = 2$ objective functions f_j defined on bounded subsets of $\mathbb{R}^m$, in (Fernández, Tóth, Plastria, and Pelegrín, 2006; Fernández and Tóth, 2006; Tóth and Fernández, 2006; Fernández and Tóth, 2007; Fernández and Tóth, to appear). In this paper, we extend these algorithms to the general case of arbitrary computable objective functions defined on a general computable set X.

2. Towards the algorithmic formulation of the problem: what is a computable set, what is a computable function

In a multi-criterion optimization problem, we have the set of alternatives X, we have several objective functions

$$f_j : X \to \mathbb{R},$$

and we are interested in describing the Pareto set – or some other similar notion of a solution. In order to analyze this problem from the algorithmic viewpoint, we need to know how this information is represented in a computer, i.e., from the computational viewpoint. In other words, we must start with a "computable" set X and "computable" functions f_j, and we must generate the computable Pareto set S.

The notions of computable numbers, computable sets, and computable functions are known; they form the so-called *computable mathematics* (also known as *constructive mathematics*); see, e.g., (Beeson, 1985; Bishop and Bridges, 1985; Kushner, 1985; Beeson, 1987; Aberth, 2007). However, these notions are not unique: depending on the practical application, we may end up with different notions of constructive sets, constructive numbers, etc. Let us therefore analyze our problem from the computational viewpoint and see which definitions naturally appear.

Let us start with the representation of a set. The easiest set to represent in a computer is a finite set $X = \{x_1, \ldots, x_m\}$: the finite set can be (and usually is) simply represented by listing all its elements $x_1, \ldots, x_m$.

In real life, however, the set of alternatives is usually infinite, with one or more parameters which can take any values from certain intervals. In this case, it is not possible to *exactly* list all possible alternatives. It is also not possible to *exactly* produce the optimal solution to the optimization problem – e.g., to produce the exact real number, we need to describe infinitely many digits, and a computer can only produce finitely many digits in any given time interval. In such cases, we can only generate an *approximation* to the optimal solution. For the notion of the approximation to be meaningful, we must be able, for every two given alternatives $x, x' \in X$, to describe how close these alternatives are. In other words, we need to be able to describe the *distance* $d(x, x')$ between every two elements, i.e., the set X must be a *metric space*.

For given two elements x and x', the distance $d(x, x')$ is a real number. We cannot always compute this number exactly – this would require infinitely many bits, but we need to be able to compute the value of this metric with an arbitrary accuracy. In other words, the values of the distance must be a computable number in the following precise sense.

This real number can also only be computed with some accuracy. It is reasonable to say that a real number is

computable if we can compute it with any given accuracy.

Definition 1. By a *computable real number*, we mean a pair $\langle x, \mathcal{U}\rangle$, where x is a real number, and the algorithm $\mathcal{U}$, given a natural number k, produces a rational number r_k for which $|x - r_k| \le 2^{-k}$.

Remark 1. For example, $\sqrt{2}$ is a computable real number because we can compute it with any given accuracy. Inside the computer, a computable number is represented by the algorithm $\mathcal{U}$. So, when we say that we can compute something (e.g., x^2) based on the computable real number input x, we mean that, based on the algorithm $\mathcal{U}$ approximating the real number x, we can generate an algorithm approximating x^2.

It is known that standard arithmetic operations can be performed on computable real numbers: the sum, the difference, the product, etc., of two computable real numbers are computable as well. Similarly, for every computable real number x, the values $\sin(x)$, $\exp(x)$, $\ln(x)$, etc., are also computable; see, e.g., (Beeson, 1985; Bishop and Bridges, 1985; Kushner, 1985; Beeson, 1987; Aberth, 2007).

Similarly, we can describe the notion of a computable set: we cannot list *exactly* all the elements of this set, but we should be able, for any given accuracy $\varepsilon = 2^{-k}$, to list all the elements with this accuracy, i.e., to produce a finite list $\{x_1, \ldots, x_k\}$ that represents all the elements from the set X with the accuracy ε. In other words, for every element $x \in X$, there is an ε-close element from this finite list, i.e., an element x_i for which $d(x, x_i) \le \varepsilon$. Such a finite list is called an *ε-net*.

We must also be able to effectively compute the distance between any two listed elements – whether they are listed for the same accuracy 2^{-k} or for two different accuracies $2^{-k} \ne 2^{-k'}$. Thus, we arrive at the following definitions.

Definition 2. Let $\langle X, d\rangle$ be a metric space, and let $\varepsilon > 0$ be a real number. A finite set $\{x_1, \ldots, x_k\} \subseteq X$ is called an *ε-net* for X if for every $x \in X$, there exists an i for which $d(x, x_i) \le \varepsilon$.

Definition 3. By a *computable set*, we mean a metric space $\langle X, d\rangle$ equipped with two algorithms:

- an algorithm that, given a natural number k, produces a (finite) 2^{-k}-net X_k for X; and
- an algorithm that for every two elements $x \in X_k$ and $x' \in X_{k'}$, computes the distance $d(x, x')$ (i.e., for any integer $m > 0$, computes a rational number which is 2^{-m}-close to $d(x, x')$).

Remark 2. For complete metric spaces, the existence of a finite ε-net for every $\varepsilon > 0$ is equivalent to compactness. Because of this, what we call computable sets are sometimes called *computable compact sets*.

Remark 3. No additional information is required about the elements of each finite set

$$X_k = \{x_{k,1}, x_{k,2}, \ldots, x_{k,m_k}\}.$$

Each element $x_{k,l}$ can be represented, e.g., by its indices k and l.

Example 1. The simplest examples of computable sets are:

- A non-degenerate interval $[\underline{a}, \overline{a}]$, with $\underline{a} < \overline{a}$. For such an interval, we can take, as X_k, the set of all rational numbers of the type $p/2^k$ (with integer p) from this interval.
- A non-degenerate multi-interval (box) $[\underline{a}_1, \overline{a}_1] \times \ldots \times [\underline{a}_m, \overline{a}_m]$ with $\underline{a}_i < \overline{a}_i$ and the sup metric

$$d((a_1, \ldots, a_m), (a'_1, \ldots, a'_m)) = \max_{i=1,\ldots,m} |a_i - a'_i|.$$

  We can take, as X_k, the set of all rational-valued points $(p_1/2^k, \ldots, p_m/2^k)$ from this box.

For the Euclidean distance, we can choose a similar set but with coordinates of the type $p_i/2^{k+k_0}$, where $2^{k_0} > \sqrt{m}$.

A computable element can be now naturally defined as an element which can be approximated with any given accuracy.

Definition 4. Let $\langle X, d\rangle$ be a computable metric space, with 2^{-k}-net X_k. By a *computable element* of X, we mean a pair $\langle x, \mathcal{U}\rangle$, where $x \in x$ and $\mathcal{U}$ is an algorithm that, given an integer $k > 0$, produces an element $r_k \in X_k$ for which $d(x, x_k) \le 2^{-k}$.

Remark 4. One can easily see that for the interval $[\underline{a}, \overline{a}]$, computable elements are simply computable real numbers from this interval. Similarly, for the m-dimensional box, computable elements are simply tuples of computable numbers $(a_1, \ldots, a_m)$ from this box.

To complete the description of a (multi-criteria) optimization problem, we also need to define the notion of a computable function f from a computable set to real numbers. Intuitively, we must be able, given an arbitrary computable element $x \in X$, to compute the value $f(x)$. In the computer, a computable element is given by its 2^{-l}-approximations r_l. Thus, the only way to compute $f(x)$

with a given accuracy 2^{-k} is to compute the value $f(r_l)$ for an appropriate approximation r_l to x.

For example, since in the computer, the value $\sqrt{2}$ is represented only approximately, to compute $\sin(\sqrt{2})$ with a given accuracy, we must know with what accuracy we must determine $\sqrt{2}$ to get the desired accuracy in $\sin(\sqrt{2})$.

So, we arrive at the following definition.

Definition 5. By a *computable function* from a computable set $\langle X, d\rangle$ (with 2^{-k}-nets X_k) to real numbers, we mean a function $f : X \to \mathbb{R}$ which is equipped with two algorithms:

- an algorithm that, given a natural number k and an element $x \in X_k$, computes the value $f(x)$ (i.e., for any integer $m > 0$, computes a rational number which is 2^{-m}-close to $f(x)$);
- an algorithm that, given a natural number k, produces a natural number l for which $d(x, x') \le 2^{-l}$ implies $|f(x) - f(x')| \le 2^{-k}$.

Remark 5. As we have mentioned earlier, all standard computer-implemented functions such as $\sqrt{}$, exp, sin, ln, etc., are computable in this sense. In particular, the possibility to find l from k is based on the fact that most of these functions have a Lipschitz property $|f(x) - f(x')| \le L \cdot d(x, x')$ for a known L.

It is also known that a composition of computable functions is also computable. Thus, all practical objective functions are computable in this sense.

Now, we have all the desired definitions, so we are ready to start the analysis of our problem – of computing the Pareto set.

3. Computing the optimum set

Before we analyze the general problem of computing the Pareto set, let us analyze the simplest case when we have only one objective function $f_1 = f$. In this case, the problem of computing the Pareto set turns into a problem of computing the *optimum* set in the following sense:

Definition 6. Let $f : X \to \mathbb{R}$ be a function. We say that an element $x_0 \in X$ is *optimal* if there exists no $x \in X$ for which $f(x) > f(x_0)$. The set $M(f)$ of all optimal elements is called the *optimum set*.

Remark 6. It is usually assumed that the objective function f is continuous. Continuous objective functions describe the usual consequences of different actions, since usually a small change in the solution only leads to a small change in the consequences.

In principle, there are some cases when the objective function is *not* continuous. For example, for some undesired side products of an industrial process, there is usually a threshold beyond which heavy fines start. In such situations, however, the desire is to avoid exceeding this threshold. Thus, the environmentally proper way of handling these situations is *not* to incorporate these fines into the profit estimates, but rather to avoid such undesirable situations altogether, and to view these restrictions as constraints that limit the set X of possible solutions. On thus restricted set, the objective function is continuous.

So, in the following text, we assume that all functions f are continuous, i.e., that $f \in C(X)$, where $C(X)$ denotes the set of all continuous functions $f : X \to \mathbb{R}$.

The problem of finding the optimum set $M(f)$ is, in general, algorithmically impossible to solve. For example, in (Kreinovich, 1975; Kreinovich, 1979; Kreinovich *et al*., 1998), it has been proven that no algorithm is possible that, given a computable polynomial of one variable which attains its optimum at exactly two points, will return these two optimizing points.

There are economic-relevant versions of this algorithmic impossibility result. For example, in (Nachbar and Zame, 1996), it is proven that even in the idealized conflict situations in which we know the opponent's strategy – and in which, thus, our gain $f(x)$ is uniquely determine by our response x – it is, in general, algorithmically impossible to compute the optimal response to this strategy, i.e., a response that maximizes the expected gain $f(x)$.

Good news is that, in practice, we only know the objective function $f(x)$ with some uncertainty $\varepsilon > 0$; in other words, we know a function $\widetilde{f}(x)$, and we know that the actual (unknown) objective function differs from $\widetilde{f}(x)$ by no more than ε.

Definition 7. Let $\langle X, d\rangle$ be a metric space.

- By a *function interval* over X, we mean a pair $\boldsymbol{f} = \langle \widetilde{f}, \varepsilon\rangle$, where $\widetilde{f} : X \to \mathbb{R}$ is a continuous function and $\varepsilon > 0$ is a real number.
- We say that a function $f : X \to \mathbb{R}$ *belongs* to the interval $\boldsymbol{f} = \langle \widetilde{f}, \varepsilon\rangle$ if $|f(x) - \widetilde{f}(x)| \le \varepsilon$ for all x.

Definition 8. Let $\boldsymbol{f} = \langle \widetilde{f}, \varepsilon\rangle$ be a function interval. By its *optimum set*, we mean the set of all the points where at least one continuous function $f \in \boldsymbol{f}$ attains it maximum, i.e., the set

$$M_\varepsilon(\widetilde{f}) = M(\boldsymbol{f}) \stackrel{\text{def}}{=} \bigcup_{f \in \boldsymbol{f} \cap C(X)} M(f).$$

From the purely *mathematical viewpoint*, this definition correctly describes our intuitive ideas. However, as we will show, from the *computational* viewpoint, this definition is much more complex than the definition of the optimum set $M(f)$ and, thus, needs to be simplified.

Indeed, we defined an optimal element $x_0 \in M(f)$ as an element for which $f(x) > f(x_0)$ is impossible, i.e., for which $f(x_0) \geq f(x)$ for all $x \in X$. Thus, to check that a given solution x is optimal, we can simply check that $f(x_0) \geq f(x)$ for all $x \in X$. So, we need to search over all elements of X.

If we literally apply our new definition, them, to check that x_0 is an optimal element, we must first find an appropriate function $f \in \boldsymbol{f}$ and then check that for this selected function f, we have $f(x_0) \geq f(x)$ for all $x \in X$. So, we need to search over not only all elements of X, but also over all possible functions $f \in \boldsymbol{f}$.

It turns out that the above definition can indeed be simplified.

Definition 9. Let $\widetilde{f} : X \to \mathbb{R}$ be a function, and $\Delta > 0$ be a real number. We say that an element $x_0 \in X$ is Δ-*optimal* if $\widetilde{f}(x_0) \geq \widetilde{f}(x) - \Delta$ for all $x \in X$.

Proposition 1. *For every continuous function $\widetilde{f} : X \to \mathbb{R}$ and for every $\varepsilon > 0$, an element $x_0 \in X$ belongs to the optimum set $M_\varepsilon(f)$ if and only if it is $(2 \cdot \varepsilon)$-optimal for $\widetilde{f}$.*

Thus, the set $M_\varepsilon(\widetilde{f})$ can be described as the set of all the elements $x_0 \in X$ which are $(2 \cdot \varepsilon)$-optimal for the nominal objective function $\widetilde{f}$.

With this reformulation, checking whether a given element x_0 belongs to the optimal set $M_\varepsilon(\widetilde{f})$ indeed becomes not more difficult than checking whether $x_0 \in M(f)$: it is sufficient to search over all elements $x \in X$, and to check that $\widetilde{f}(x_0) \geq \widetilde{f}(x) - 2 \cdot \varepsilon$ for all these elements.

Proof. Let us first show that if $x_0 \in M_\varepsilon(\widetilde{f})$, i.e., if x_0 is optimal for some function f for which

$$|f(x) - \widetilde{f}(x)| \leq \varepsilon,$$

then $\widetilde{f}(x_0) \geq \widetilde{f}(x) - 2 \cdot \varepsilon$ for all $x \in X$. Indeed, since x_0 is optimal for f, we have $f(x_0) \geq f(x)$. From

$$|f(x_0) - \widetilde{f}(x_0)| \leq \varepsilon$$

and

$$|f(x) - \widetilde{f}(x)| \leq \varepsilon,$$

we conclude that $\widetilde{f}(x_0) \geq f(x_0) - \varepsilon$ and that

$$f(x) \geq \widetilde{f}(x) - \varepsilon.$$

Thus,

$$\widetilde{f}(x_0) \geq f(x_0) - \varepsilon \geq f(x) - \varepsilon \geq$$
$$(\widetilde{f}(x) - \varepsilon) - \varepsilon = \widetilde{f}(x) - 2 \cdot \varepsilon.$$

Vice versa, let $x_0 \in X$ be an element for which $\widetilde{f}(x_0) \geq \widetilde{f}(x) - 2 \cdot \varepsilon$ for all $x \in X$. Let us prove that there exists a function $f \in \boldsymbol{f}$ for which x_0 is optimal. Indeed, as such a function f, we can take

$$f(x) \stackrel{\text{def}}{=} \min(g(x), h(x)),$$

where

$$g(x) \stackrel{\text{def}}{=} \widetilde{f}(x) + \varepsilon \cdot \max(1 - d(x_0, x), 0);$$

$$h(x) \stackrel{\text{def}}{=} \widetilde{f}(x_0) + \varepsilon.$$

For $x = x_0$, we have

$$f(x_0) = g(x_0) = h(x_0) = \widetilde{f}(x_0) + \varepsilon;$$

for all other elements $x \in X$, we have

$$f(x) \leq h(x) = \widetilde{f}(x_0) + \varepsilon.$$

Thus, we indeed have $f(x_0) \geq f(x)$ for all $x \in X$ – i.e., x_0 is indeed optimal for f.

To complete our proof, we must prove that $f \in \boldsymbol{f}$, i.e., that $|f(x) - \widetilde{f}(x)| \leq \varepsilon$ for all $x \in X$. Indeed, $f(x)$ is defined as the minimum of two expressions $g(x)$ and $h(x)$. The first expression $g(x)$ adds, to $\widetilde{f}(x)$, a value ε multiplied by a coefficient $\max(1 - d(x_0, x), 0)$. We always have

$$0 \leq \max(1 - d(x_0, x), 0) \leq 1 - d(x_0, x) \leq 1$$

and thus,

$$0 \leq \max(1 - d(x_0, x), 0) \leq 1.$$

So, if the minimum $f(x)$ is equal to the first expression $g(x)$, we do get $|f(x) - \widetilde{f}(x)| \leq \varepsilon$.

What if the minimum $f(x)$ is equal to the second expression $h(x) = \widetilde{f}(x_0) + \varepsilon$? Since $h(x) \leq g(x)$, i.e.,

$$\widetilde{f}(x_0) + \varepsilon \leq \widetilde{f}(x) + \varepsilon \cdot \max(1 - d(x_0, x), 0),$$

we have

$$f(x) = h(x) = \widetilde{f}(x_0) + \varepsilon \leq$$
$$\widetilde{f}(x) + \varepsilon \cdot \max(1 - d(x_0, x), 0) \leq \widetilde{f}(x) + \varepsilon,$$

so

$$f(x) \leq \widetilde{f}(x) + \varepsilon.$$

From our assumption $\widetilde{f}(x_0) \geq \widetilde{f}(x) - 2 \cdot \varepsilon$, we conclude that

$$f(x) = \widetilde{f}(x_0) + \varepsilon \geq (\widetilde{f}(x) - 2 \cdot \varepsilon) + \varepsilon = \widetilde{f}(x) - \varepsilon,$$

so

$$f(x) \geq \widetilde{f}(x) - \varepsilon.$$

Thus, when $f(x) = h(x)$, we also have

$$|f(x) - \widetilde{f}(x)| \leq \varepsilon.$$

The proposition is proven. ■

The upper bound ε of the approximation error is also only know with uncertainty. At best, we know an interval $[\underline{\varepsilon}, \overline{\varepsilon}]$ for this bound. The larger ε, the largest the corresponding function interval $\boldsymbol{f} = \langle \widetilde{f}, \varepsilon \rangle$ and thus, the larger the optimum set $M_\varepsilon(\widetilde{f})$; so, if $\underline{\varepsilon} \le \varepsilon \le \overline{\varepsilon}$, we have

$$M_{\underline{\varepsilon}}(\widetilde{f}) \subseteq M_\varepsilon(\widetilde{f}) \subseteq M_{\overline{\varepsilon}}(\widetilde{f}).$$

Because of this relation, the following theorem provides the desired algorithm for computing the optimum set:

Theorem 1. *There exists an algorithm that, given a computable function $\widetilde{f}$ from a computable set X to real numbers and two rational numbers $0 < \underline{\varepsilon} < \overline{\varepsilon}$, produces a finite list of elements $L \subseteq X$ and a rational number $\delta > 0$ with the following two properties:*

- *If $x_0 \in M_{\underline{\varepsilon}}(\widetilde{f})$, then $d(x_0, x) \le \delta$ for some $x \in L$.*
- *If $d(x_0, x) \le \delta$ for some $x \in L$, then $x_0 \in M_{\overline{\varepsilon}}(\widetilde{f})$.*

The list L and the accuracy δ provide a description of the desired optimum set. Specifically, the desired optimum set is approximated by the set of all the elements which are δ-close to one of the elements from the given list, i.e., by the union of the corresponding balls $B_\delta(x) \stackrel{\text{def}}{=} \{x' : d(x, x') \le \delta\}$:

$$M_{\underline{\varepsilon}}(\widetilde{f}) \subseteq \bigcup_{x \in L} B_\delta(x) \subseteq M_{\overline{\varepsilon}}(\widetilde{f}).$$

Proof. The main idea of the proof is that we take a finite approximation to X, an approximation to $\widetilde{f}$, find the "optimum set" for the corresponding approximate problem, and then show that this solution to the approximate problem is indeed the desired approximation to the actual optimum set $M(\boldsymbol{f})$.

Let us find the appropriate approximation to the set X. The difference $\Delta\varepsilon \stackrel{\text{def}}{=} \overline{\varepsilon} - \underline{\varepsilon}$ is a positive rational number. Since comparing two rational numbers is straightforward, we can thus find the smallest natural number k for which $2^{-k} \le \dfrac{\Delta\varepsilon}{4}$.

By using an algorithm from the definition of a computable function $\widetilde{f}$, we can find a natural number l for which $d(x, x') \le 2^{-l}$ implies $|\widetilde{f}(x) - \widetilde{f}(x')| \le 2^{-k}$. By using an algorithm from the definition of a computable set, we can algorithmically find a 2^{-l}-net $X_l = \{x_{l,1}, \ldots, x_{l,m_l}\}$ for the metric space X. This finite set X_l will be our approximation to the actual set X. (As we will see later, the value l is selected so as to provide the desired approximation accuracy for the resulting optimum set.)

The next step is to approximate the given real-valued function $\widetilde{f} : X \to \mathbb{R}$ by a rational-valued function defined on the finite set X_l. By using an algorithm from the definition of a computable function, for each element $x_{l,i} \in X_l$, we can compute a rational number y_i which is 2^{-k}-close to $\widetilde{f}(x_{l,i})$, i.e., for which $|y_i - \widetilde{f}(x_{l,i})| \le 2^{-k}$. As the desired approximation, we can now take a function that assigns, to each element $x_{l,i} \in X_l$, the corresponding rational number y_i.

Let us now find the optimum set for the resulting approximate problem. In the original problem, we had an interval $[\underline{\varepsilon}, \overline{\varepsilon}]$ of possible values of ε. To define our approximate set, let us take the midpoint $\widetilde{\varepsilon} \stackrel{\text{def}}{=} \dfrac{\underline{\varepsilon} + \overline{\varepsilon}}{2}$ of this interval.

In view of Proposition 1, for the approximate problem, the optimum set can be described as follows. First, we find the set I of all the indices i for which

$$y_i \ge y_{i'} - 2 \cdot \widetilde{\varepsilon}$$

for all $i' = 1, \ldots, m_l$. Then, we take the set

$$L = \{x_{l,i} : i \in I\}$$

of the corresponding elements $x_{l,i} \in X_l$.

Let us show that this finite list satisfies the desired two properties for $\delta = 2^{-l}$.

Let us start our proof with the second property. We want to prove that if for some $x_0 \in X$ and for some $i \in I$, we have $d(x_0, x_{l,i}) \le 2^{-l}$, then $x_0 \in M_{\overline{\varepsilon}}(\widetilde{f})$, i.e., $\widetilde{f}(x_0) \ge \widetilde{f}(x) - 2 \cdot \overline{\varepsilon}$ for all $x \in X$. Indeed, let $x \in X$. Since $X_l = \{x_{l,1}, \ldots, x_{l,m_l}\}$ is a 2^{-l}-net, there exists an i' for which $d(x, x_{l,i'}) \le 2^{-l}$. Due to our choice of l, we can conclude that $|\widetilde{f}(x) - \widetilde{f}(x_{l,i'})| \le 2^{-k}$. Due to our choice of $y_{i'}$, we have $|y_{i'} - \widetilde{f}(x_{l,i'})| \le 2^{-k}$ and thus,

$$|\widetilde{f}(x) - y_{i'}| \le |\widetilde{f}(x) - \widetilde{f}(x_{l,i'})| + |y_{i'} - \widetilde{f}(x_{l,i'})| \le 2 \cdot 2^{-k},$$

and

$$y_{i'} \ge \widetilde{f}(x) - 2 \cdot 2^{-k}.$$

Similarly, from $d(x_0, x_{l,i}) \le 2^{-l}$, we conclude that $|\widetilde{f}(x_0) - \widetilde{f}(x_{l,i})| \le 2^{-k}$. Due to our choice of y_i, we have $|y_i - \widetilde{f}(x_{l,i})| \le 2^{-k}$ and thus,

$$|\widetilde{f}(x_0) - y_i| \le |\widetilde{f}(x_0) - \widetilde{f}(x_{l,i})| + |y_i - \widetilde{f}(x_{l,i})| \le 2 \cdot 2^{-k},$$

and

$$\widetilde{f}(x_0) \ge y_i - 2 \cdot 2^{-k}.$$

From $y_i \ge y_{i'} - 2 \cdot \widetilde{\varepsilon}$ and $y_{i'} \ge \widetilde{f}(x) - 2 \cdot 2^{-k}$, we can now conclude that

$$\widetilde{f}(x_0) \ge y_i - 2 \cdot 2^{-k} \ge y_{i'} - 2 \cdot 2^{-k} - 2 \cdot \widetilde{\varepsilon} \ge$$

$$\widetilde{f}(x) - 4 \cdot 2^{-k} - 2 \cdot \widetilde{\varepsilon}.$$

By our choice of k, we have $4 \cdot 2^{-k} \le \Delta\varepsilon$, hence

$$\widetilde{f}(x_0) \ge \widetilde{f}(x) - \Delta\varepsilon - 2 \cdot \widetilde{\varepsilon}.$$

By definition, $\Delta\varepsilon = \overline{\varepsilon} - \underline{\varepsilon}$ and $2 \cdot \widetilde{\varepsilon} = \overline{\varepsilon} + \underline{\varepsilon}$, so we have

$$\widetilde{f}(x_0) \ge \widetilde{f}(x) - (\overline{\varepsilon} - \underline{\varepsilon}) - (\overline{\varepsilon} + \underline{\varepsilon}) = \widetilde{f}(x) - 2 \cdot \overline{\varepsilon}.$$

The second property is proven.

Let us now prove the first property. We want to prove that if $x_0 \in M_{\underline{\varepsilon}}(\widetilde{f})$, i.e., if $\widetilde{f}(x_0) \ge \widetilde{f}(x) - 2 \cdot \underline{\varepsilon}$ for all $x \in X$, then $d(x_0, x_{l,i}) \le \delta = 2^{-l}$ for some $i \in I$.

Indeed, since X_l is an 2^{-l}-net, there exists an element $x_{l,i} \in X_l$ for which $d(x_0, x_{l,i}) \le \delta = 2^{-l}$. We need to prove that $i \in I$, i.e., that $y_i \ge y_{i'} - 2 \cdot \widetilde{\varepsilon}$ for all i'.

By definition of the value y_i, we have

$$|y_i - \widetilde{f}(x_{l,i})| \le 2^{-k},$$

so $y_i \ge \widetilde{f}(x_{l,i}) - 2^{-k}$. By the choice of l, from $d(x_0, x_{l,i}) \le 2^{-l}$, we conclude that

$$|\widetilde{f}(x_0) - \widetilde{f}(x_{l,i})| \le 2^{-k},$$

hence $\widetilde{f}(x_{l,i}) \ge \widetilde{f}(x_0) - 2^{-k}$. Combining this inequality with $y_i \ge \widetilde{f}(x_{l,i}) - 2^{-k}$, we conclude that

$$y_i \ge (\widetilde{f}(x_0) - 2^{-k}) - 2^{-k} = \widetilde{f}(x_0) - 2 \cdot 2^{-k}.$$

We assumed that $\widetilde{f}(x_0) \ge \widetilde{f}(x) - 2 \cdot \underline{\varepsilon}$ for all $x \in X$; in particular, this is true for $x = x_{l,i'}$. Thus, we have $\widetilde{f}(x_0) \ge \widetilde{f}(x_{l,i'}) - 2 \cdot \underline{\varepsilon}$. Combining this inequality with $y_i \ge \widetilde{f}(x_0) - 2 \cdot 2^{-k}$, we conclude that

$$y_i \ge \widetilde{f}(x_0) - 2 \cdot 2^{-k} \ge (\widetilde{f}(x_{l,i'}) - 2 \cdot \underline{\varepsilon}) - 2 \cdot 2^{-k}.$$

By definition of the value $y_{i'}$, we have

$$|y_{i'} - \widetilde{f}(x_{l,i'})| \le 2^{-k},$$

so $\widetilde{f}(x_{l,i'}) \ge y_{i'} - 2^{-k}$. Thus, we have

$$y_i \ge \widetilde{f}(x_{l,i'}) - 2 \cdot \underline{\varepsilon} - 2 \cdot 2^{-k} \ge (y_{i'} - 2^{-k}) - 2 \cdot \underline{\varepsilon} - 2 \cdot 2^{-k} =$$

$$y_{i'} - 2 \cdot \underline{\varepsilon} - 3 \cdot 2^{-k}.$$

We have selected k so that $4 \cdot 2^{-k} \le \Delta\varepsilon$, hence

$$3 \cdot 2^{-k} < 4 \cdot 2^{-k} \le \Delta\varepsilon,$$

and

$$y_i \ge y_{i'} - 2 \cdot \underline{\varepsilon} - \Delta\varepsilon.$$

Substituting $\Delta\varepsilon = \overline{\varepsilon} - \underline{\varepsilon}$ into this inequality, we conclude that

$$y_i \ge y_{i'} - 2 \cdot \underline{\varepsilon} - (\overline{\varepsilon} - \underline{\varepsilon}) = y_{i'} - (\underline{\varepsilon} + \overline{\varepsilon}).$$

By definition of $\widetilde{\varepsilon}$, we have $\underline{\varepsilon} + \overline{\varepsilon} = 2 \cdot \widetilde{\varepsilon}$, so we get the desired inequality

$$y_i \ge y_{i'} - 2 \cdot \widetilde{\varepsilon}.$$

■

Remark 7. We want to emphasize that while, to the best of our knowledge, Theorem 1 is new, it is fully in line with the general understanding of specialists in computable mathematics. Its proof, while somewhat technically cumbersome, naturally follows from the known results of computable mathematics. The reason why we have presented this result and its proof in all the detail is that Theorem 1 provides a pattern following which we prove the *main* result of this paper – Theorem 2 on computability of general Pareto sets. It would have been much more difficult to understand the general proof of Theorem 2 without first going through the particular case $n = 1$ – case in which the notion of the Pareto set turns into a simpler notion of the optimum set.

Remark 8. Once we established that the algorithm exists, the natural next question is: how efficient is this algorithm? Since the above algorithm requires that we consider all the elements of the corresponding ε-net, its number of steps grows as the number of these elements. For an m-dimensional box this number is $\approx V/\varepsilon^m$, so it grows exponentially with the dimension m of the box.

This is, however, acceptable, since in general, the optimization problems are NP-hard (Kreinovich *et al.*, 1998), and therefore, the worst-case exponential time is inevitable (unless, of course, it turns out that, contrary to the expectations of most computer scientists, P = NP and thus, all such problems can be solved in feasible (polynomial) times).

It is worth mentioning that, as mentioned in (Nachbar and Zame, 1996), in the conflict situations in which the *exact* optimal strategy is not algorithmically computable, it *is* possible to compute an "approximate" ε-optimal strategy. However, for small ε, the computation of this ε-optimal strategy requires the analysis of all possible combinations of m moves for some large integer m – hence, requires the computation time that exponentially grows with m.

Remark 9. In the above text, we assume that we know the objective function $f(x)$ with given *absolute* accuracy, i.e., that we know that the actual (unknown) objective function $f(x)$ satisfies the inequality $|f(x) - \widetilde{f}(x)| \le \varepsilon$ for a given function $\widetilde{f}(x)$.

In some practical situations, we know the non-negative function $f(x)$ with *relative* uncertainty, i.e., we know that the actual (unknown) objective function $f(x)$

satisfies the inequality

$$\left|\frac{f(x)-\widetilde{f}(x)}{\widetilde{f}(x)}\right| \le \varepsilon$$

for a given function $\widetilde{f}(x)$. For example, we may know $f(x)$ with an accuracy 10% ($\varepsilon = 0.1$) or 5% ($\varepsilon = 0.05$).

These situations can be reduced to the case of absolute uncertainty if we switch to a logarithmic space, i.e., if we consider a new objective function $F(x) \stackrel{\text{def}}{=} \ln(f(x))$. This change does not affect the optimum set – since the logarithm is a strictly increasing function, the functions $f(x)$ and $F(x)$ attain their maxima at exactly the same points: $M(f) = M(F)$. The above relative-accuracy restriction on $f(x)$ has the form

$$1-\varepsilon \le \frac{f(x)}{\widetilde{f}(x)} \le 1+\varepsilon.$$

By taking the logarithms of all three parts of this inequality, we get an equivalent inequality

$$\ln(1-\varepsilon) \le F(x) - F_\sim(x) \le \ln(1+\varepsilon),$$

where we denoted $F_\sim(x) \stackrel{\text{def}}{=} \ln(\widetilde{f}(x))$. This inequality, in its turn, can be reformulated as

$$F_\sim(x) + \ln(1-\varepsilon) \le F(x) \le F_\sim(x) + \ln(1+\varepsilon),$$

i.e., as the condition that for every x, the (unknown) value $F(x)$ belongs to the interval

$$[F_\sim(x) + \ln(1-\varepsilon), F_\sim(x) + \ln(1+\varepsilon)].$$

The width $w \stackrel{\text{def}}{=} \ln(1+\varepsilon) - \ln(1-\varepsilon)$ of this interval is the same for all x, so we can take the midpoint

$$\widetilde{F}(x) \stackrel{\text{def}}{=} F_\sim(x) + \frac{\ln(1+\varepsilon)+\ln(1-\varepsilon)}{2}$$

of this interval and describe the above inequality in the equivalent form

$$|F(x) - \widetilde{F}(x)| \le \varepsilon',$$

where $\varepsilon' \stackrel{\text{def}}{=} w/2$ is the interval's radius (half-width).

This is exactly the inequality with which we started our absolute-accuracy case analysis. So, we can indeed reduce the solution of a relative-accuracy problem to the absolute-accuracy case.

4. Computing Pareto sets: general case

Now we are ready to deal with the general problem of computing the Pareto set.

Definition 10. Let X be a set and let $f_j : X \to \mathbb{R}$, $j = 1, 2, \ldots, n$, be functions from X to real numbers. We say that an element $x_0 \in X$ is *Pareto-optimal* if there exists no $x \in X$ for which $f_j(x) \ge f_j(x_0)$ for all j and $f_j(x) > f_j(x_0)$ for some j. The set $P(f_1, \ldots, f_n)$ of all Pareto-optimal elements is called the *Pareto set*.

In practice, we only know each of the objective functions f_j with some accuracy $\varepsilon_j > 0$.

Definition 11. Let $\boldsymbol{f}_j = \langle \widetilde{f}_j, \varepsilon_j \rangle$, $j = 1, 2, \ldots, n$, be function intervals. By the *Pareto set* corresponding to these function intervals, we mean the set of all the points which are Pareto-optimal for at least one combination $f_j \in \boldsymbol{f}_j$, i.e., the set

$$P_{\varepsilon_1,\ldots,\varepsilon_n}(\widetilde{f}_1, \ldots, \widetilde{f}_n) = P(\boldsymbol{f}_1, \ldots, \boldsymbol{f}_n) \stackrel{\text{def}}{=} \bigcup_{f_1 \in \boldsymbol{f}_1 \cap C(X), \ldots, f_n \in \boldsymbol{f}_n \cap C(X)} P(f_1, \ldots, f_n).$$

Similarly to the case of the optimum set, we can simplify this definition. In original definition of a Pareto-optimal element x_0, for every $x \in X$, we cannot have $f_j(x) \ge f_j(x)$ for all j and $f_j(x) > f_j(x)$ for some j. Thus, for every $x \in X$, either there exists an j for which $f_j(x) < f_j(x_0)$, or we have $f_j(x) \le f_j(x_0)$ for all j. Thus, a natural "Δ"-version of this definition takes the following form:

Definition 12. Let X be a set, let $\widetilde{f}_1, \ldots, \widetilde{f}_n$ be functions from the set X to real numbers, and let $\Delta_1, \ldots, \Delta_n$ be positive real numbers. We say that an element $x_0 \in X$ is $(\Delta_1, \ldots, \Delta_n)$-*Pareto optimal* if for every $x \in X$, there exists an index j for which $\widetilde{f}_j(x_0) \ge \widetilde{f}_j(x) - \Delta_j$.

For the general Pareto sets, we no longer have exact equivalence between this "Δ"-definition and the definition of the Pareto set for the sequence of function intervals, but we have an "almost" equivalence in the following precise sense:

Proposition 2. *Let X be a metric space, let $\widetilde{f}_1, \ldots, \widetilde{f}_n$ be continuous functions from X to real numbers, and let $\varepsilon_1, \ldots, \varepsilon_n$ be positive real numbers. Then the following two properties hold:*

- *If an element x_0 belongs to the Pareto set $P_{\varepsilon_1,\ldots,\varepsilon_n}(\widetilde{f}_1, \ldots, \widetilde{f}_n)$, then it is $(2\cdot\varepsilon_1, \ldots, 2\cdot\varepsilon_n)$-Pareto optimal for the functions $\widetilde{f}_1, \ldots, \widetilde{f}_n$.*
- *If for some values $\varepsilon'_1 < \varepsilon_1, \ldots, \varepsilon'_n < \varepsilon_n$, an element $x_0 \in X$ is $(2\cdot\varepsilon'_1, \ldots, 2\cdot\varepsilon'_n)$-Pareto optimal for the functions $\widetilde{f}_1, \ldots, \widetilde{f}_n$, then x_0 belongs to the Pareto set $P_{\varepsilon_1,\ldots,\varepsilon_n}(\widetilde{f}_1, \ldots, \widetilde{f}_n)$.*

We say that this is "almost" equivalence since we can take the values ε'_j arbitrarily close to ε_j.

Proof. Let us first show that if

$$x_0 \in P_{\varepsilon_1,\ldots,\varepsilon_n}(\widetilde{f}_1,\ldots,\widetilde{f}_n),$$

i.e., if x_0 is optimal for some functions $f_1,\ldots,f_n$ for which

$$|f_j(x)-\widetilde{f}_j(x)|\le\varepsilon_j,$$

then x_0 is $(2\cdot\varepsilon_1,\ldots,2\cdot\varepsilon_n)$-Pareto optimal for the functions $\widetilde{f}_1,\ldots,\widetilde{f}_n$, i.e., for every $x\in X$, there exists an index j for which $\widetilde{f}_j(x_0)\ge\widetilde{f}_j(x)-2\cdot\varepsilon_j$.

Indeed, let us pick an arbitrary element $x\in X$. Since x_0 is Pareto-optimal for the functions $f_1,\ldots,f_n$, either there exists an j for which $f_j(x_0)>f_j(x)$ or for every j, we have $f_j(x_0)\ge f_j(x)$. In both cases, we have $f_j(x_0)\ge f_j(x)$ for some j. From

$$|f_j(x_0)-\widetilde{f}_j(x_0)|\le\varepsilon_j$$

and

$$|f_j(x)-\widetilde{f}_j(x)|\le\varepsilon_j,$$

we conclude that $\widetilde{f}_j(x_0)\ge f_j(x_0)-\varepsilon_j$ and that

$$f_j(x)\ge\widetilde{f}_j(x)-\varepsilon_j.$$

Thus,

$$\widetilde{f}_j(x_0)\ge f_j(x_0)-\varepsilon_j\ge f_j(x)-\varepsilon_j\ge$$

$$(\widetilde{f}_j(x)-\varepsilon_j)-\varepsilon_j=\widetilde{f}_j(x)-2\cdot\varepsilon_j.$$

Vice versa, let $\varepsilon'_j<\varepsilon_j$, and let $x_0\in X$ be an element for which, for every $x\in X$, there exists an index j for which $\widetilde{f}_j(x_0)\ge\widetilde{f}_j(x)-2\cdot\varepsilon'_j$. Let us prove that there exist functions $f_j\in\boldsymbol{f}_j$ for which x_0 is Pareto-optimal, i.e., for which for every $x\in X$, either there exists a j for which $f_j(x_0)>f_j(x)$, or for all j, we have $f_j(x_0)\ge f_j(x)$.

Indeed, we can take

$$f_j(x)\stackrel{\text{def}}{=}\min(g_j(x),h_j(x))+$$

$$(\varepsilon_j-\varepsilon'_j)\cdot\max(1-d(x_0,x),0),$$

where

$$g_j(x)\stackrel{\text{def}}{=}\widetilde{f}_j(x)+\varepsilon'_j\cdot\max(1-d(x_0,x),0);$$

$$h_j(x)\stackrel{\text{def}}{=}\widetilde{f}_j(x_0)+\varepsilon'_j.$$

For every $x\in X$, there exists an index j for which $\widetilde{f}_j(x_0)\ge\widetilde{f}_j(x)-2\cdot\varepsilon'_j$. Let us prove that for this same index j, we have $f_j(x_0)>f_j(x)$. Indeed, for $x=x_0$, we have $g_j(x_0)=h_j(x_0)=\widetilde{f}_j(x_0)+\varepsilon'_j$ and

$$\max(1-d(x_0,x),0)=1,$$

thus,

$$f_j(x_0)=\widetilde{f}_j(x_0)+\varepsilon'_j+(\varepsilon_j-\varepsilon'_j)=\widetilde{f}_j(x_0)+\varepsilon_j.$$

For all other elements $x\in X$, we have $d(x_0,x)>0$ hence $1-d(x,x_0)<1$ and $\max(1-d(x,x_0),0)<1$. Thus,

$$f_j(x)\le h_j(x)+(\varepsilon_j-\varepsilon'_j)\cdot\max(1-d(x_0,x),0)<$$

$$h_j(x)+(\varepsilon_j-\varepsilon'_j)=(\widetilde{f}_j(x_0)+\varepsilon'_j)+(\varepsilon_j-\varepsilon'_j)=$$

$$\widetilde{f}_j(x_0)+\varepsilon_j=f_j(x_0).$$

So, for this j, we indeed have $f_j(x_0)>f_j(x)$ – i.e., x_0 is indeed Pareto-optimal for $(f_1,\ldots,f_n)$.

To complete our proof, we must prove that for every j, we have $f_j\in\boldsymbol{f}_j$, i.e., that $|f_j(x)-\widetilde{f}_j(x)|\le\varepsilon_j$ for all $x\in X$.

Indeed, we have already proved, in our proof of Proposition 1, that

$$|\min(g_j(x),h_j(x))-\widetilde{f}_j(x)|\le\varepsilon'_j.$$

The difference $(\varepsilon_j-\varepsilon'_j)\cdot\max(1-d(x_0,x),0)$ between $f_j(x)$ and $\min(g_j(x),h_j(x))$ is bounded by $\varepsilon_j-\varepsilon'_j$:

$$|f_j(x)-\min(g_j(x),h_j(x))|\le\varepsilon_j-\varepsilon'_j.$$

Thus, we have

$$|f_j(x)-\widetilde{f}_j(x)|\le|f_j(x)-\min(g_j(x),h_j(x))|+$$

$$|\min(g_j(x),h_j(x))-\widetilde{f}_j(x)|\le(\varepsilon_j-\varepsilon'_j)+\varepsilon'_j=\varepsilon_j.$$

The proposition is proven. ■

Theorem 2. *There exists an algorithm that, given n computable functions $\widetilde{f}_1,\ldots,\widetilde{f}_n$ from a computable set X to real numbers and $2n$ rational numbers $0<\underline{\varepsilon}_j<\overline{\varepsilon}_j$, $j=1,\ldots,n$, produces a finite list of elements $L\subseteq X$ and a rational number $\delta>0$ with the following two properties:*

- *If $x_0\in P_{\underline{\varepsilon}_1,\ldots,\underline{\varepsilon}_n}(\widetilde{f}_1,\ldots,\widetilde{f}_n)$, then $d(x_0,x)\le\delta$ for some $x\in L$.*
- *If $d(x_0,x)\le\delta$ for some $x\in L$, then*

$$x_0\in P_{\overline{\varepsilon}_1,\ldots,\overline{\varepsilon}_n}(\widetilde{f}_1,\ldots,\widetilde{f}_n).$$

The list L and the accuracy δ provide a description of the desired Pareto set. Specifically, the desired optimum set is approximated by the set of all the elements which are δ-close to one of the elements from the given list, i.e., by the union of the corresponding balls

$$P_{\underline{\varepsilon}_1,\ldots,\underline{\varepsilon}_n}(\widetilde{f}_1,\ldots,\widetilde{f}_n)\subseteq$$

$$\bigcup_{x\in L}B_\delta(x)\subseteq P_{\overline{\varepsilon}_1,\ldots,\overline{\varepsilon}_n}(\widetilde{f}_1,\ldots,\widetilde{f}_n).$$

Proof. The main idea of the proof is the same as for the optimum set: we take a finite approximation to X, an approximation to $\widetilde{f}$, find the "Pareto set" for the corresponding approximate problem, and then show that this solution to the approximate problem is indeed the desired approximation to the actual Pareto set $P(\boldsymbol{f}_1, \ldots, \boldsymbol{f}_n)$.

Let us find the appropriate approximation to the set X. For every j, the difference $\Delta\varepsilon_j \stackrel{\text{def}}{=} \overline{\varepsilon}_j - \underline{\varepsilon}_j$ is a positive rational number. Since comparing two rational numbers is straightforward, we can thus find the smallest natural number k_j for which $2^{-k_j} \le \dfrac{\Delta\varepsilon_j}{8}$.

By using an algorithm from the definition of a computable function $\widetilde{f}_j$, we can find a natural number l_j for which $d(x, x') \le 2^{-l_j}$ implies $|\widetilde{f}_j(x) - \widetilde{f}_j(x')| \le 2^{-k_j}$. Thus, for the largest $l \stackrel{\text{def}}{=} \max(l_1, \ldots, l_n)$ of this natural numbers, we have the following property: $d(x, x') \le 2^{-l}$ implies $|\widetilde{f}_j(x) - \widetilde{f}_j(x')| \le 2^{-k_j}$ for all $j = 1, \ldots, n$.

By using an algorithm from the definition of a computable set, we can algorithmically find a 2^{-l}-net $X_l = \{x_{l,1}, \ldots, x_{l,m_l}\}$ for the metric space X. This finite set X_l will be our approximation to the actual set X.

The next step is to approximate each given real-valued function $\widetilde{f}_j : X \to \mathbb{R}$ by a rational-valued function defined on the finite set X_l. By using an algorithm from the definition of a computable function, for each element $x_{l,i} \in X_l$ and for each $j = 1, \ldots, n$, we can compute a rational number $y_{i,j}$ which is 2^{-k_j}-close to $\widetilde{f}_j(x_{l,i})$, i.e., for which $|y_{i,j} - \widetilde{f}_j(x_{l,i})| \le 2^{-k_j}$. As the desired approximation to the function $\widetilde{f}_j$, we can now take a function that assigns, to each element $x_{l,i} \in X_l$, the corresponding rational number $y_{i,j}$.

Let us now find the Pareto set for the resulting approximate problem. In the original problem, for every $j = 1, \ldots, n$, we have an interval $[\underline{\varepsilon}_j, \overline{\varepsilon}_j]$ of possible values of ε_j. To define our approximate set, let us take the midpoint $\widetilde{\varepsilon}_j \stackrel{\text{def}}{=} \dfrac{\underline{\varepsilon}_j + \overline{\varepsilon}_j}{2}$ of this interval.

In view of Proposition 2, for the approximate problem, the optimum set can be described as follows. First, we find the set I of all the indices i for which, for every $i' = 1, \ldots, m_l$, there exists a j for which

$$y_{i,j} \ge y_{i',j} - 2 \cdot \widetilde{\varepsilon}_j.$$

Then, we take the set

$$L = \{x_{l,i} : i \in I\}$$

of the corresponding elements $x_{l,i} \in X_l$.

Let us show that this finite list satisfies the desired two properties for $\delta = 2^{-l}$.

Let us start our proof with the second property. We want to prove that if for some $x_0 \in X$ and for some $i \in I$, we have $d(x_0, x_{l,i}) \le 2^{-l}$, then

$$x_0 \in P_{\overline{\varepsilon}_1, \ldots, \overline{\varepsilon}_n}(\widetilde{f}_1, \ldots, \widetilde{f}_n).$$

For that, in view of Proposition 2, it is sufficient to prove that for some values $\varepsilon'_j < \varepsilon_j$ $(j = 1, \ldots, n)$, for every $x \in X$, there exists a j for which $\widetilde{f}_j(x_0) \ge \widetilde{f}_j(x) - 2 \cdot \varepsilon'_j$.

Indeed, let $x \in X$. Since $X_l = \{x_{l,1}, \ldots, x_{l,m_l}\}$ is a 2^{-l}-net, there exists an i' for which $d(x, x_{l,i'}) \le 2^{-l}$. Since $i \in I$, there exists a j for which $y_{i,j} \ge y_{i',j} - 2 \cdot \widetilde{\varepsilon}_j$. Due to our choice of l, we can conclude that

$$|\widetilde{f}_j(x) - \widetilde{f}_j(x_{l,i'})| \le 2^{-k_j}.$$

Due to our choice of $y_{i',j}$, we have

$$|y_{i',j} - \widetilde{f}_j(x_{l,i'})| \le 2^{-k_j}$$

and thus,

$$|\widetilde{f}_j(x) - y_{i',j}| \le |\widetilde{f}_j(x) - \widetilde{f}_j(x_{l,i'})| + |y_{i',j} - \widetilde{f}_j(x_{l,i'})| \le 2 \cdot 2^{-k_j},$$

and

$$y_{i',j} \ge \widetilde{f}_j(x) - 2 \cdot 2^{-k_j}.$$

Similarly, from $d(x_0, x_{l,i}) \le 2^{-l}$, we conclude that $|\widetilde{f}_j(x_0) - \widetilde{f}_j(x_{l,i})| \le 2^{-k_j}$. Due to our choice of $y_{i,j}$, we have $|y_{i,j} - \widetilde{f}_j(x_{l,i})| \le 2^{-k_j}$ and thus,

$$|\widetilde{f}_j(x_0) - y_{i,j}| \le |\widetilde{f}_j(x_0) - \widetilde{f}_j(x_{l,i})| + |y_{i,j} - \widetilde{f}_j(x_{l,i})| \le 2 \cdot 2^{-k_j},$$

and

$$\widetilde{f}_j(x_0) \ge y_{i,j} - 2 \cdot 2^{-k_j}.$$

From $y_{i,j} \ge y_{i',j} - 2 \cdot \widetilde{\varepsilon}$ and $y_{i',j} \ge \widetilde{f}_j(x) - 2 \cdot 2^{-k_j}$, we can now conclude that

$$\widetilde{f}_j(x_0) \ge y_{i,j} - 2 \cdot 2^{-k_j} \ge y_{i',j} - 2 \cdot 2^{-k_j} - 2 \cdot \widetilde{\varepsilon}_j \ge \widetilde{f}_j(x) - 4 \cdot 2^{-k_j} - 2 \cdot \widetilde{\varepsilon}_j.$$

By our choice of k_j, we have $4 \cdot 2^{-k_j} \le \dfrac{1}{2} \cdot \Delta\varepsilon_j$, hence

$$\widetilde{f}_j(x_0) \ge \widetilde{f}_j(x) - \frac{1}{2} \cdot \Delta\varepsilon_j - 2 \cdot \widetilde{\varepsilon}_j.$$

By definition, $\Delta\varepsilon_j = \overline{\varepsilon}_j - \underline{\varepsilon}_j$ and $2 \cdot \widetilde{\varepsilon}_j = \overline{\varepsilon}_j + \underline{\varepsilon}_j$, so we have

$$\widetilde{f}_j(x_0) \ge \widetilde{f}_j(x) - \frac{1}{2} \cdot (\overline{\varepsilon}_j - \underline{\varepsilon}_j) - (\overline{\varepsilon}_j + \underline{\varepsilon}_j) = \widetilde{f}_j(x) - 2 \cdot \varepsilon'_j,$$

where

$$\varepsilon'_j = \frac{3}{4}\overline{\varepsilon}_j + \frac{1}{4}\underline{\varepsilon}_j < \varepsilon_j.$$

The second property is proven.

Let us now prove the first property. We want to prove that if $x_0 \in P_{\underline{\varepsilon}_1,\ldots,\underline{\varepsilon}_n}(\widetilde{f}_1,\ldots,\widetilde{f}_n)$, then

$$d(x_0, x_{l,i}) \le \delta = 2^{-l}$$

for some $i \in I$.

Indeed, let $x_0 \in P_{\underline{\varepsilon}_1,\ldots,\underline{\varepsilon}_n}(\widetilde{f}_1,\ldots,\widetilde{f}_n)$. Due to Proposition 2, this implies that x_0 is $(2\cdot\underline{\varepsilon}_1,\ldots,2\cdot\underline{\varepsilon}_n)$-Pareto optimal for the functions $\widetilde{f}_1,\ldots,\widetilde{f}_n$, i.e., that for every $x \in X$, there exists a j for which

$$\widetilde{f}_j(x_0) \ge \widetilde{f}_j(x) - 2\cdot\underline{\varepsilon}_j.$$

In particular, such a j exist for every $x = x_{l,i'} \in X_l$, i.e., that for every i', there exists a j for which

$$\widetilde{f}_j(x_0) \ge \widetilde{f}_j(x_{l,i'}) - 2\cdot\underline{\varepsilon}_j.$$

Since X_l is an 2^{-l}-net, there exists an element

$$x_{l,i} \in X_l$$

for which $d(x_0, x_{l,i}) \le \delta = 2^{-l}$. To prove the first property, it is sufficient to prove that $i \in I$, i.e., that for every i', there exists a j for which $y_{i,j} \ge y_{i',j} - 2\cdot\widetilde{\varepsilon}_j$. We will show that this inequality indeed holds for the above j, for which $\widetilde{f}_j(x_0) \ge \widetilde{f}_j(x_{l,i'}) - 2\cdot\underline{\varepsilon}_j$.

By definition of the value $y_{i,j}$, we have

$$|y_{i,j} - \widetilde{f}_j(x_{l,i})| \le 2^{-k_j},$$

so $y_{i,j} \ge \widetilde{f}_j(x_{l,i}) - 2^{-k_j}$. By the choice of l, from $d(x_0, x_{l,i}) \le 2^{-l}$, we conclude that

$$|\widetilde{f}_j(x_0) - \widetilde{f}_j(x_{l,i})| \le 2^{-k_j},$$

hence $\widetilde{f}_j(x_{l,i}) \ge \widetilde{f}_j(x_0) - 2^{-k_j}$. Combining this inequality with $y_{i,j} \ge \widetilde{f}_j(x_{l,i}) - 2^{-k_j}$, we conclude that

$$y_{i,j} \ge (\widetilde{f}_j(x_0) - 2^{-k_j}) - 2^{-k_j} = \widetilde{f}_j(x_0) - 2\cdot 2^{-k_j}.$$

Combining this inequality with $\widetilde{f}_j(x_0) \ge \widetilde{f}_j(x_{l,i'}) - 2\cdot\underline{\varepsilon}_j$, we conclude that

$$y_{i,j} \ge \widetilde{f}_j(x_0) - 2\cdot 2^{-k_j} \ge (\widetilde{f}_j(x_{l,i'}) - 2\cdot\underline{\varepsilon}_j) - 2\cdot 2^{-k_j}.$$

By definition of the value $y_{i',j}$, we have

$$|y_{i',j} - \widetilde{f}_j(x_{l,i'})| \le 2^{-k_j},$$

so $\widetilde{f}_j(x_{l,i'}) \ge y_{i',j} - 2^{-k_j}$. Thus, we have

$$y_{i,j} \ge \widetilde{f}_j(x_{l,i'}) - 2\cdot\underline{\varepsilon}_j - 2\cdot 2^{-k_j} \ge$$

$$(y_{i',j} - 2^{-k_j}) - 2\cdot\underline{\varepsilon}_j - 2\cdot 2^{-k_j} =$$

$$y_{i',j} - 2\cdot\underline{\varepsilon}_j - 3\cdot 2^{-k_j}.$$

We have selected k_j so that $8\cdot 2^{-k_j} \le \Delta\varepsilon_j$, hence

$$3\cdot 2^{-k_j} < 8\cdot 2^{-k_j} \le \Delta\varepsilon_j,$$

and

$$y_{i,j} \ge y_{i',j} - 2\cdot\underline{\varepsilon}_j - \Delta\varepsilon_j.$$

Substituting $\Delta\varepsilon_j = \overline{\varepsilon}_j - \underline{\varepsilon}_j$ into this inequality, we conclude that

$$y_{i,j} \ge y_{i',j} - 2\cdot\underline{\varepsilon}_j - (\overline{\varepsilon}_j - \underline{\varepsilon}_j) = y_{i',j} - (\underline{\varepsilon}_j + \overline{\varepsilon}_j).$$

By definition of $\widetilde{\varepsilon}_j$, we have $\underline{\varepsilon}_j + \overline{\varepsilon}_j = 2\cdot\widetilde{\varepsilon}_j$, so we get the desired inequality

$$y_{i,j} \ge y_{i',j} - 2\cdot\widetilde{\varepsilon}_j.$$

■

Remark 10. Similarly to the computation of maximum sets, the above algorithm requires that we consider all the elements of the corresponding ε-net, and thus, its number of steps grows exponentially with the dimension m of the box. While (as we have mentioned) we cannot decrease the computation time in *all* the cases (unless P=NP), it is possible to make this algorithm more efficient in *some* cases.

Some of the ideas of how to speed up this algorithm are described in (Fernández and Tóth, 2007).

Acknowledgments

This work was supported in part by NSF grant HRD-0734825, by Grant 1 T36 GM078000-01 from the National Institutes of Health, by the Japan Advanced Institute of Science and Technology (JAIST) International Joint Research Grant 2006-08, and by the Max Planck Institut für Mathematik.

The authors are thankful to the anonymous referees for valuable suggestions.

References

Aberth, O. (2007). *Introduction to Precise Numerical Methods*, Academic Press, San Diego, California.

Beeson, M. (1985). *Foundations of Constructive Mathematics: Metamathematical Studies*, Springer, Berlin/Heidelberg/New York.

Beeson, M. (1987). Some relations between classical and constructive mathematics. *Journal of Symbolic Logic* **43**: 228–246.

Bishop, E. and Bridges, D.S. (1985). *Constructive analysis*, Springer-Verlag, Berlin-Heidelberg-New York.

Fernández, J. and Tóth, B. (2006). Obtaining the efficient set of biobjective competitive facility location and design problems. In: *Proceedings of EURO XXI*, Reykjavík, Iceland, July 2–5, 2006.

Fernández, J. and Tóth, B. (2007). Obtaining an outer approximation of the efficient set of nonlinear biobjective problems, *Journal of Global Optimization* **38**(2): 315–331.

Fernández, J. and Tóth, B. (to appear). Obtaining the efficient set of nonlinear biobjective optimization problems via interval branch-and-bound methods, *Computational Optimization and Applications*, to appear.

Fernández, J., Tóth, B., Plastria, F., and Pelegrín, B. (2006). Reconciling franchisor and franchisee: a planar multiobjective competitive location and design model.In: Recent Advances in Optimization, Springer Lecture Notes in Economics and Mathematical Systems **563**: 375–398.

Figueira, J., Greco, S., and Ehrgott, M., editors (2004). *Multiple Criteria Decision Analysis: State of the Art Surveys*, Kluwer, Dordrecht.

Kreinovich, V. (1975). Uniqueness implies algorithmic computability. In: *Proceedings of the 4th Student Mathematical Conference*, Leningrad University, Leningrad, 1975, pp. 19–21 (in Russian).

Kreinovich, V. (1979). *Categories of Space-Time Models*, Ph.D. dissertation, Novosibirsk, Soviet Academy of Sciences, Siberian Branch, Institute of Mathematics (in Russian).

Kreinovich, V., Lakeyev, A., Rohn, J., and Kahl, P. (1998). *Computational Complexity and Feasibility of Data Processing and Interval Computations*, Kluwer, Dordrecht.

Kushner, B.A. (1985). *Lectures on Constructive Mathematical Analysis*, American Mathematical Society, Providence, Rhode Island.

Nachbar, J.H. andZame, W.R. (1996). Non-computable strategies and discounted repeated games. *Economic theory* **8**: 103–122

Nickel, S. and Puerto, J. (2005). *Location Theory: A Unified Approach*, Springer-Verlag, Berlin.

Ruzika, S. and Wiecek, M.M. (2005). Approximation methods in multiopbjective programming. *Journal of Optimization Theory and Applications* **126**: 473–501.

Tóth, B. and Fernández, J. (2006). Obtaining the efficient set of nonlinear biobjective optimization problems via interval branch-and-bound methods. In: *Proceedings of the 12th GAMM - IMACS International Symposium on Scientific Computing, Computer Arithmetic, and Validated Numerics SCAN'06*, Duisburg, Germany, September 26–29, 2006.

Received: 22 September 2008
Revised: 6 December 2008