Dedicated to
Professor Jakub Gutenbaum
on his 70th birthday

Control and Cybernetics

vol. **29** (2000) No. 1

Parallel approaches to parametric optimization and the convergence of interactive decision support

by

Andrzej P. Wierzbicki

National Institute of Telecommunications,
Szachowa 1, 04-894 Warsaw, Poland

Abstract: In the perspective of parallel processing, a new sense of parametric optimization might be promoted. The paper shows that it is possible to propose new parallel versions of basic optimization algorithms, as well as an advanced method of securing convergence in interactive multiobjective optimization and decision support, all based on a modified concept of parametric embedding. This general idea is exemplified for the case of the simplex algorithm of linear programming by a parameterized and coarse-grain parallel *augmented simplex algorithm,* where a linear optimization problem can be embedded into a multiple-objective family which introduces diversified directions of search cutting through the interior of the original admissible set. For the case of nonlinear programming, a parameterized and coarse-grain parallel variable metric *pulsar algorithm* is shortly presented, where parallel directional searches are combined with a parameterized variable metric to produce a pulsating, robust nonlinear programming algorithm. These two examples concern very basic optimization tools; at the other end of the spectrum of optimization-related methods, a general method called *outranking trials* of securing convergence of interactive multiobjective optimization and decision support is obtained through parameterizing an outranking relation and using basic properties of order-consistent achievement functions in reference point methodology for testing the existence of outranking points by parallel optimization runs. Thus, the paper presents the use of parallel processing to solve a wide range of modified parametric embedding problems related to optimization and decision support.

Keywords: parallel computations, optimization, interactive decision support.

1. Introduction; parametric embedding for parallelization

Various paradigms of parametric optimization can be distinguished by specifying various goals of using parametric approaches. These goals might be:

- To make possible solving of the ill-posed or ill-conditioned problems;
- To shorten computations for difficult or large-scale problems;
- To increase the robustness of solutions;
- To broaden the scope of approaches to the analyzed models.

While the first two goals of parameterization are well known, less known are possible uses of parametric approaches to the remaining two goals; this paper concentrates on demonstrating such uses. Let us comment only on the last goal; broadening of model analysis – see, e.g., Wierzbicki (1984) – can refer to many aspects, such as sensitivity analysis, but always includes optimization, seen, however, not as the goal in itself but as a tool of various tasks of analysis. Another specific feature of this paper is concentration on parallel computations in parametric optimization. There are several classical approaches to parallelization, such as *fine-grain parallelization* (single-loop parallelization), massive parallelization, etc. However, the development of very powerful processors used practically in every personal computer or workstation makes these approaches obsolete; *coarse-grain paralellization,* based on large computing tasks performed parallely, is necessary instead. This paper shows that parametric embedding and optimization might be good areas for using coarse-grain parallel approaches.

The basic idea of parametric embedding might be specified as follows. Suppose solving a specific problem – e.g. inverting a function f – is difficult (computationally intensive, or ill-conditioned, or even ill-posed). Note that we speak here of approximate solutions. If, say, a precise inversion is needed, as in cryptography, parametric embedding might be difficult to apply. We construct a parameterized family of problems which for specific parameters are easy to solve but include the original difficult problem and we solve these problems consecutively. This idea is illustrated by the following example: we have to invert, instead of a "difficult" function f, a parameterized function $F_\alpha = (1-\alpha)I + \alpha f$, where I is identity operator, for a sequence of parameter values α starting with 0 and terminating with 1. This basic idea is, however, *consecutive* in its essence, hence difficult for parallelization. If the goal of parameterization is different, e.g. to increase robustness or to broaden analysis, then we can use the same idea of parametric embedding for coarse-grain parallelization. We proceed as follows. Given a problem $\mathcal{F}$, we construct a family of parameterized problems $\mathcal{F}_\alpha$, embedding the original problem (we can use a similar specific embedding as in the example above). Then we use parallel computing for solving various instances of parameterized problems. Finally, we combine the results of parallel computations for increasing robustness or broadening analysis.

This modified idea of parametric embedding will be shown in this paper by two parallel modifications of basic optimization algorithms and an application to securing convergence of interactive multiobjective decision support.

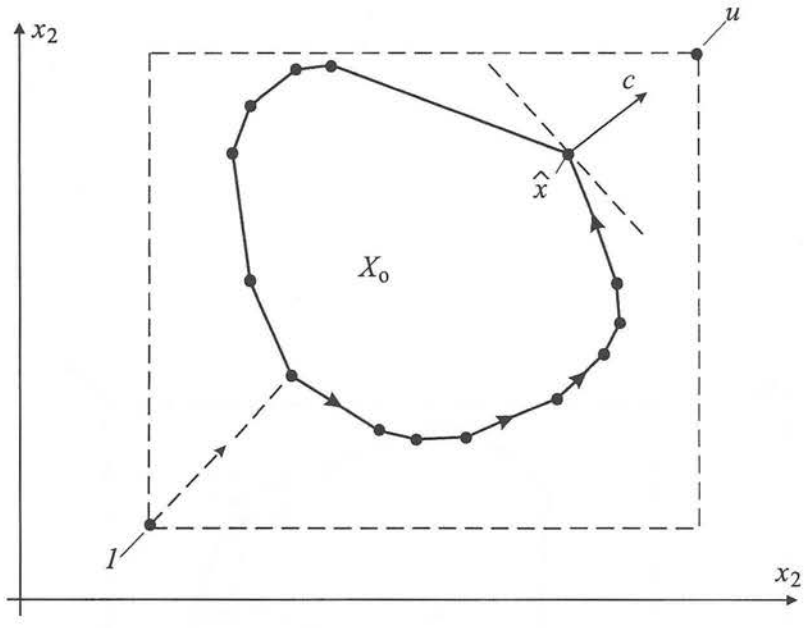

Figure 1. An illustrative example of solving a linear programming problem by a simplex method.

2. Augmented simplex algorithm

A typical linear programming problem:

$$\underset{\boldsymbol{x}\in X_o}{\text{maximize}}(q_0 = \boldsymbol{c}^T\boldsymbol{x}) \tag{1}$$

$$X_o = \{\boldsymbol{x} \in \boldsymbol{R}^n : \boldsymbol{l} \leq \boldsymbol{x} \leq \boldsymbol{u},\ \boldsymbol{b} \leq \boldsymbol{y} = \boldsymbol{A}\boldsymbol{x} \leq \boldsymbol{b} + \boldsymbol{r} \in \boldsymbol{R}^m\},$$

can be solved by two classical methods:

1) Simplex methods – many variants (two-phase, dual, big M, special for scarce matrices, etc.) – all moving through simplex vertices, as in the illustrative example of Fig. 1.
2) Karmarkar and interior point methods (elipsoid; logarithmic barrier function; primal-dual interior point methods) all moving through simplex interior, see an illustrative example in Fig. 2.

A multiobjective linear programming problem:

$$\text{“}\underset{\boldsymbol{x}\in X_o}{\text{maximize}}\text{”}(\boldsymbol{q} = \boldsymbol{C}\boldsymbol{x} \in \boldsymbol{R}^p) \tag{2}$$

$$X_o = \{\boldsymbol{x} \in \boldsymbol{R}^n : \boldsymbol{l} \leq \boldsymbol{x} \leq \boldsymbol{u}, \boldsymbol{b} \leq \boldsymbol{y} = \boldsymbol{A}\boldsymbol{x} \leq \boldsymbol{b} + \boldsymbol{r} \in \boldsymbol{R}^m\},$$

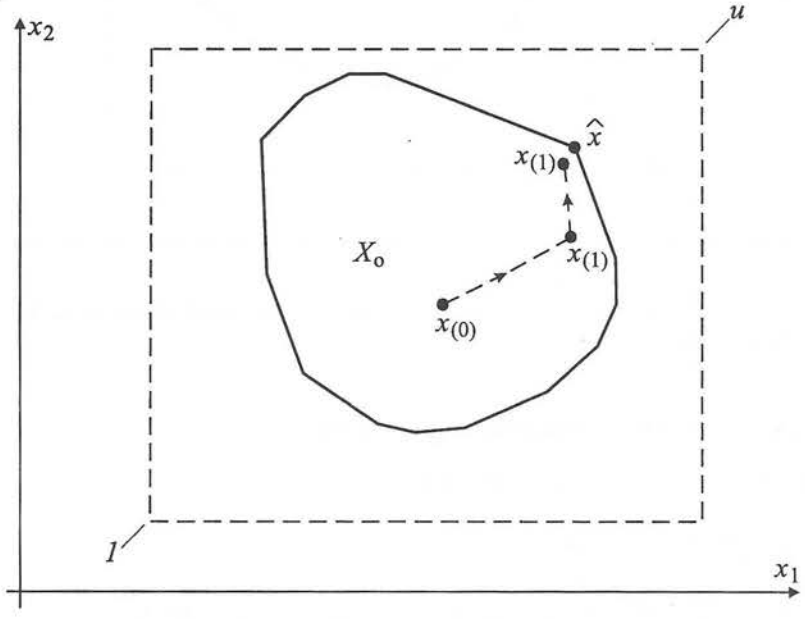

Figure 2. An illustrative example of solving a linear programming problem by an interior point method.

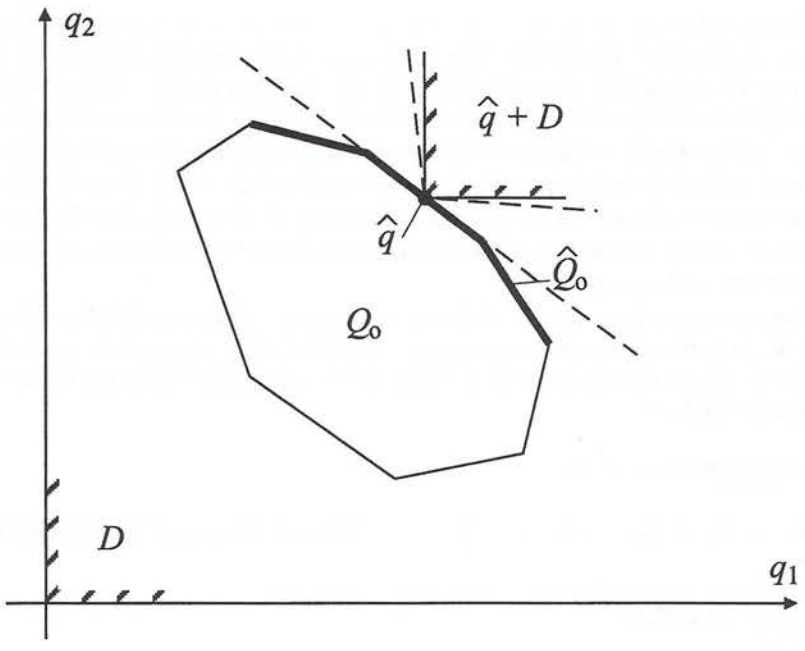

Figure 3. An illustrative example of an outcome set in objective space for a multiobjective linear programming problem and of the Pareto (efficient) frontier of this set.

where "maximize" is defined by a cone $D \subset \mathbb{R}^p$, eg.: $D = \mathbb{R}^p_+$ (the case of Pareto optimization), has typically an entire set of efficient solutions (so called Pareto frontier), which can be illustrated as in Fig. 3.

There are many methods of solving (2) – in the sense of finding at least one element of the entire set of efficient solutions. We can distinguish the following classes of methods:

1) classical methods which are based on maximizing a weighted sum of objectives, with *weighting coefficients* α_i:

$$s_1(\boldsymbol{q}, \boldsymbol{\alpha}) = \sum_{i=1}^{p} \alpha_i q_i$$

2) contemporary methods which admit that a linear approximation to a value function of a decision maker is too simplistic and results in many paradoxes, thus are based on maximizing a nonlinear scalarizing function. One of the most useful is the following *order-consisitent achievement scalarizing funcion*, see e.g. Wierzbicki (1980, 1986, 1998):

$$s_2(\boldsymbol{q}, \bar{\boldsymbol{q}}, \boldsymbol{\alpha}) = \min_{i=1,\ldots,p} \alpha_i(q_i - \bar{q}_i) + \varepsilon \sum_{i=1}^{p} \alpha_i(q_i - \bar{q}_i) \tag{3}$$

using *reference or aspiration levels* $\bar{q}_i$ in addition to weighting coefficients α_i (which play a secondary role then or are used only implicitly) as a basic way of expressing the preferences of the decision maker. The coefficient $\varepsilon \geq 0$ is usually chosen as a small positive number, in order to eliminate the so-called weakly efficient solutions and the above function is related then to a slightly broader cone D_ε that approximates $D = \mathbb{R}^p_+$ from outside and is equal to the zero-level set of this function if all $\bar{q}_i = 0$. We shall return later to the properties of such functions; here we shall use its simplest variant with $\varepsilon = 0$.

When using such a variant of the achievement scalarizing function, we can define an augmented linear programming problem as a parametric embedding of a linear programming problem. Suppose we have the following linear programming problem:

$$\underset{\boldsymbol{x} \in X_o}{\text{maximize}} (q_0 = \boldsymbol{c}^T \boldsymbol{x})$$

and let $\operatorname{Int} X_o \neq \emptyset, c_j > 0 \ \forall j = 1, \ldots, n$. This problem can be embedded as follows:

$$\max_{\boldsymbol{x} \in X_o} s_\rho(\boldsymbol{x}, \boldsymbol{d}, \bar{\boldsymbol{x}}) \tag{4}$$

$$s_\rho(\boldsymbol{x}, \boldsymbol{d}, \bar{\boldsymbol{x}}) = \rho \boldsymbol{c}^T \boldsymbol{x} + (1 - \rho) \min_{j=1,\ldots,n} ((x_j - \bar{x}_j)/d_j)$$

where:

- $\rho = 0$ in multi-criteria phase;
- $\rho = 1$ in single-criterion phase;
- $\boldsymbol{d}$ is a direction in the solution space;
- $d_i = 1/\alpha_i$ are inverse weighting coefficients for *the problem of multiobjective maximization of all components* x_j;
- $\bar{\boldsymbol{x}} = \bar{\boldsymbol{q}}$ is a reference point in the solution space.

The augmented simplex algorithm uses first a multi-criteria phase in order to shorten computations and increase the robustness of the method; then it switches (either directly or gradually) to the single-criteria phase. The solution of the multi-criteria phase problem is illustrated graphically in Fig. 4. Even if we use a simplex method for solving the problem, the algorithm cuts through the interior of the original admissible solution set along the direction $\boldsymbol{d}$ (but not through the interior of the resulting admissible solution set, since the multiobjective formulation changes this set).

This property suggests that the augmented simplex algorithm can be faster than the more classical variants of the simplex algorithms when applied to large scale problems with much more constraints than original variables. However, the augmented simplex algorithm can be also used for parallel or distributed computations, where various directions $\boldsymbol{d}$ are used on various processors in the multiobjective phase. The results of testing of such algorithm (including not only

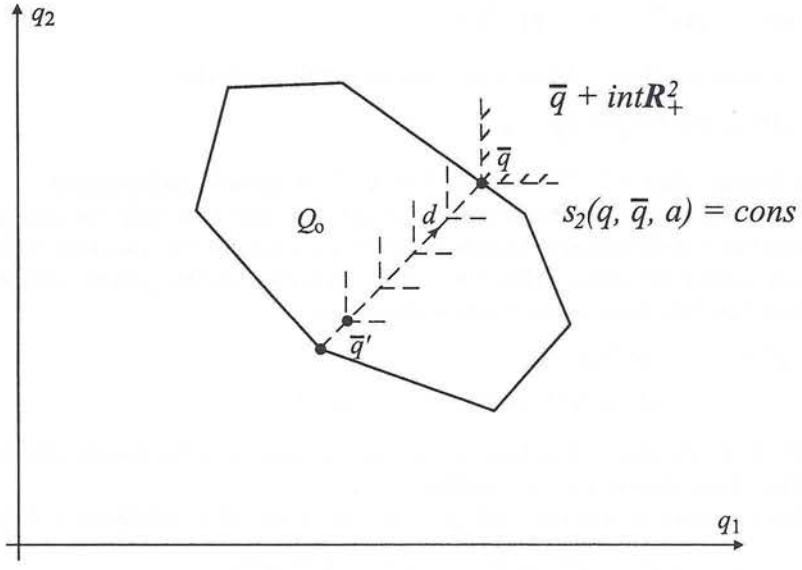

Figure 4. An illustrative example of the path of solutions in the multi-criteria phase of the augmented simplex method.

simplex-like algorithms, but also subdifferential approaches, see e.g. Wierzbicki, 1994) indicate that such parameterization and parallelization:

- sometimes shortens computations, but not necessarily in proportion to the number of processors used;
- always increases robustness, that is, gives more reliable solutions to ill-defined linear programming problems with solutions determined through almost linearly dependent constraints.

3. Pulsar variable metric

Consider the elementary problem of nonlinear programming: minimize a twice differentiable function $f : \mathbb{R}^n \to \mathbb{R}^1$ over an open set. One of the basic classes of algorithms are quasi-Newton or variable metric methods. A variable metric $\boldsymbol{V}^{(k)}$ successively approximates the Hessian $\boldsymbol{H}(\boldsymbol{x}) = \nabla^2 f(\boldsymbol{x})$ or its inverse $\boldsymbol{H}^{-1}(\boldsymbol{x})$.

If we assume initially that the function is quadratic:

$$f(\boldsymbol{x}) = 0.5 < \boldsymbol{x}, \quad \boldsymbol{A}\boldsymbol{x} > + < \boldsymbol{b}, \quad \boldsymbol{x} > + c \tag{5}$$

$$\nabla f(\boldsymbol{x}) = \boldsymbol{g}(\boldsymbol{x}) = \boldsymbol{A}\boldsymbol{x} + \boldsymbol{g} \in \mathbb{R}^n, \nabla^2 f(\boldsymbol{x}) = \boldsymbol{H}(\boldsymbol{x}) = \boldsymbol{A} \in L(\mathbb{R}^n, \mathbb{R}^n)$$

and consider following increments and relations:

$$\boldsymbol{s}^{(k)} = \boldsymbol{x}^{(k+1)} - \boldsymbol{x}^{(k)}; \boldsymbol{y}^{(k)} = \nabla f(\boldsymbol{x}^{(k+1)}) - \nabla f(\boldsymbol{x}^{(k)}); \tag{6}$$

$$y^{(k)} = Hs^{(k)}, s^{(k)} = H^{-1}y^{(k)},$$

then we can use these relations to construct $V^{(k)}$ such that:

$$s^{(j)} = V^{(k+1)}y^{(j)} \text{ for } j = 1, \ldots, k \tag{7}$$

which implies that $V^{(n+1)} = H^{-1}$, if all $y^{(k)}$ are linearly independent.

There are many known formulae for $\Delta V^{(k)}$, used not only for quadratic, but also twice differentiable functions. Most popular are the so-called rank-two variable metric formulae. However, these formulae require *precise directional search* in the following *quasi-Newton directions*:

$$\begin{aligned} d^{(k)} &= -V^{(k)}g^{(k)}; \\ &\text{where } V^{(k)} = I; g^{(k)} = \nabla f(x^{(k)})) \end{aligned} \tag{8}$$

and produce the approximations of the Hessian matrix or its inverse due to the fact that these directions are *conjugate*.

There is also a symmetric *rank-one variable metric* that is defined by formula:

$$\Delta V^{(k)} = \frac{(s^{(k)} - V^{(k)}y^{(k)}) >< (s^{(k)} - V^{(k)}y^{(k)})}{< (s^{(k)} - V^{(k)}y^{(k)}), y^{(k)} >} \tag{9}$$

where . >< . denotes the outer product. This variable metric is unpopular, because it becomes ill-defined when the matrix $V^{(k)}$ approximates well H^{-1}:

$$< (s^{(k)} - V^{(k)}y^{(k)}), y^{(k)} >=< (H^{-1} - V^{(k)})y^{(k)}, y^{(k)} > \tag{10}$$

However, this variable metric does not require that the quasi-Newton directions (8) are used, neither that precise directional search is applied, nor it is related to the conjugacy of directions (8). When modified with special safeguards, this variable metric can produce very efficient algorithms. Because of these advantages, this variable metric can be applied in parallel computations. We shall show how it can be used in a *parallel pulsar algorithm*.

Assume that $n \geq P - 2$, where P is the number of processors used (these processors are indexed by $\psi = 1, \ldots, P$). Typical jobs assigned to processors with $\psi > 1$ are directional searches. $\psi = 1$ is reserved for variable metric approximation and other coordinating tasks.

Let K denote pulsar algorithm iterations (this number is actually increased by two during each double pulsar iteration). *In odd-numbered (divergent) iterations,* processors $\psi = 2, \ldots, P$ perform directional searches (not necessarily accurate) along the following directions:

$$\begin{aligned} d^{(1,0)} &= -g^{(1)} \text{ or } d^{(K,0)} = -V^{(K-1)}g^{(K)} \\ d^{(K,j)} &= e_j = (0, \ldots, 1_{(j)}, \ldots, 0)^T, j = 1, \ldots, n \end{aligned} \tag{11}$$

to produce $n + 1$ diverse points:

$$x^{(K,j)} = x^{(K)} + \hat{\tau}^{(K,j)}d^{(j)},$$

where $\hat{\tau}^{(K,j)}$ are the corresponding step-size coefficients. We have then:

$$\begin{aligned} \boldsymbol{s}^{(K,j)} &= \hat{\tau}^{(K,j)}\boldsymbol{d}^{(j)}; \\ f^{(K,j)} &= f(\boldsymbol{x}^{(K)}+\boldsymbol{s}^{(K,j)}); \boldsymbol{g}^{(K,j)} = \nabla f(\boldsymbol{x}^{(K)}+\boldsymbol{s}^{(K,j)}); \\ \boldsymbol{y}^{(K,j)} &= \nabla f(\boldsymbol{x}^{(K)}+\boldsymbol{s}^{(K,j)}) - \nabla f(\boldsymbol{x}^{(K)}) \end{aligned} \tag{12}$$

The data $f^{(K,j)}$, $\boldsymbol{g}^{(K,j)}$, $\boldsymbol{y}^{(K,j)}$ and $\boldsymbol{s}^{(K,j)}$ are successively transmitted to the first processor that uses these data to update the approximation of the inverse Hessian by a safe-guarded rank-one formula.

In even-numbered (convergent) iteration of the pulsar algorithm, the directional searches are performed from $n+1$ various points $\boldsymbol{x}^{(K,j)}$ in corresponding quasi-Newton directions:

$$\boldsymbol{d}^{(K+1,j)} = -\boldsymbol{V}^{(K)}\boldsymbol{g}^{(K,j)} \tag{13}$$

Other operations in the even-numbered iterations (computing the increments of solutions and gradients, transmitting data to the first processor for Hessian approximation, etc.) are similar as in the odd-numbered. The even-numbered iteration of the pulsar algorithm ends when the first processor determined a new inverse Hessian approximation and a new common starting point $\boldsymbol{x}^{(K+2)}$ for the next odd-numbered iteration, chosen among $\boldsymbol{x}^{(K+1,j)}$ as a point with the lowest $f^{(K+1,j)}$; appropriate stopping tests are repeated.

All points $\boldsymbol{x}^{(K+1,j)}$ obtained in an even-numbered iteration should rather precisely approximate the minimum of the goal function f *from various sides.* This is shown by the illustrative example in Fig. 5.

This spread of points, this *pulsating* justifies the name of the pulsar algorithm, but at the same time increases the robustness of the algorithm with respect to:

- numerical inaccuracies,
- ill-conditioning of optimization problem,
- local minima
- inaccuracies of the stopping tests.

The tests of the pulsar algorithm – see Sobczyk et al. (1994) – show that the algorithm sometimes increases the speed of computations, but not proportionally to the number of processors used. On the other hand, the algorithm displays a substantially increased robustness for ill-conditioned problems; for example, it solves with high accuracy problems with singular Hessian at the optimal solution, while sequential variable metric algorithms can solve such problems only with very low accuracy.

4. Outranking trials

4.1. Interactive methods of multicriteria decision support

Multicriteria optimization and decision theory is very broad and contains many approaches and problems; see e.g. Gutenbaum (1977), Sawaragi et al. (1985),

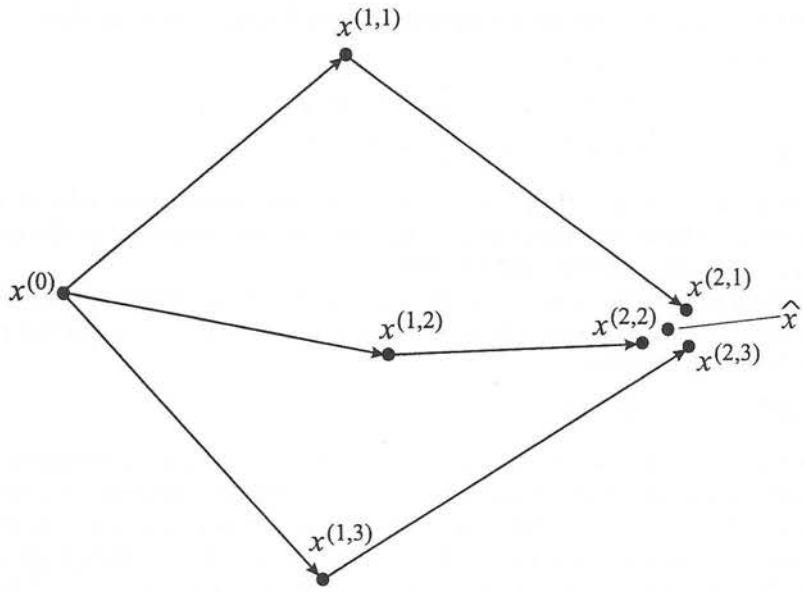

Figure 5. An illustrative example of the odd-numbered (divergent) and even-numbered (convergent) iterations of the pulsar algorithm.

Steuer (1986), Rios (1994), Gal et al. (1999). However, we limit the presentation here to interactive methods in this field. Generally, interactive methods support the interaction with *the user* of a decision support system (DSS) or a *decision maker*; in multicriteria approaches, this interaction has often the goal of selecting a point on a Pareto frontier. This point might optimize the value function of the decision maker, or be the best according to other forms of her/his preferences. We can distinguish – see Wierzbicki (1997):

- Interactive methods with proven convergence, such as:
  - the Geoffrion-Dyer-Feinberg procedure, see Geoffrion et al. (1972);
  - the Zionts-Wallenius procedure (more effective, but with more restricted class of multi-objective linear programming models – see Zionts and Wallenius, 1976, 1983);
  - the Korhonen-Laakso Procedure and Pareto Race, related to the concept of a *reference direction*, see Korhonen et al. (1985).
  - Stochastic Quasi-Gradient Procedures, see e.g. Ermolev et al. (1988).
- Interactive methods with accelerated convergence, such as:
  - Reference Ball, see Wierzbicki (1997);
  - Contracted Cone – by Steuer and Choo, see Steuer et al. (1983), later versions by Jaszkiewicz and Słowiński (1994);

- Satisficing Trade-Off – by Nakayama and Sawaragi, see Nakayama et al. (1983), Nakayama (1994);
- Light Beam Search – by Jaszkiewicz and Słowiński, see Jaszkiewicz et al. (1994).

Most approaches with proven convergence imitate mathematical programming techniques – see Bogetoft et al. (1988), Wierzbicki (1997). Other approaches concentrate on accelerating practical convergence instead, and are quite effective. Reference point approaches, mentioned earlier in this paper and exemplified here by the Reference Ball, Satisficing Trade-Off and Pareto Race procedures, are also quite effective, but their convergence was proven only for special cases. We shall present below the Outranking Trials procedure, introduced in Wierzbicki (1997). This procedure not only combines an outranking relation with a reference point approach, but also adds an important element: the application of parametric embedding and parallel computations.

4.2. Properties of reference point approach

As mentioned earlier in this paper, reference point approaches are based on the maximization of a nonlinear *order-consistent achievement scalarizing function*, which can have diverse forms – see Lewandowski et al. (1989), Wierzbicki (1986, 1999). The most fundamental of these forms is given in Eq. (3); we repeat it here with slight changes of notation:

$$\sigma(q, \bar{q}) = \min_{i=1,\ldots,p} (q_i - \bar{q}_i) + \varepsilon \sum_{i=1}^{p} (q_i - \bar{q}_i) \tag{14}$$

As already indicated, the zero-level set of this function corresponds to a slightly broader positive cone D_ε:

$$D_\varepsilon = \{q \in \boldsymbol{R}^p : \min_{i=1,\ldots,p} (q_i) + \varepsilon \sum_{i=1}^{p} (q_i) \geq 0\} \tag{15}$$

This function shifts the above cone to the *reference or aspiration point* $\bar{q}$. Due to this property, *the achievement function nonlinearly separates* the attainable criteria set Q_0, and the cone $\bar{q} + D$ is efficient if $\bar{q}$ lies on Pareto frontier. Because of this separation property and the monotonicity of $\sigma(q, \bar{q})$, we can state the most important properties of the reference point approach:

- each maximum of $\sigma(q, \bar{q})$ is efficient;
- for each $\hat{q}$ that is efficient with respect to D_ε, there exists such $\bar{q}$ that the maximum of $\sigma(q, \bar{q})$ over $q \in Q_0$ is attained at $\hat{q}$;
- if the maximum of $\sigma(q, \bar{q})$ over $q \in Q_0$ is negative, then $\bar{q} \notin Q_0$.

All these properties are independent of the convexity of Q_0; this is shown in the illustrative example in Fig. 6.

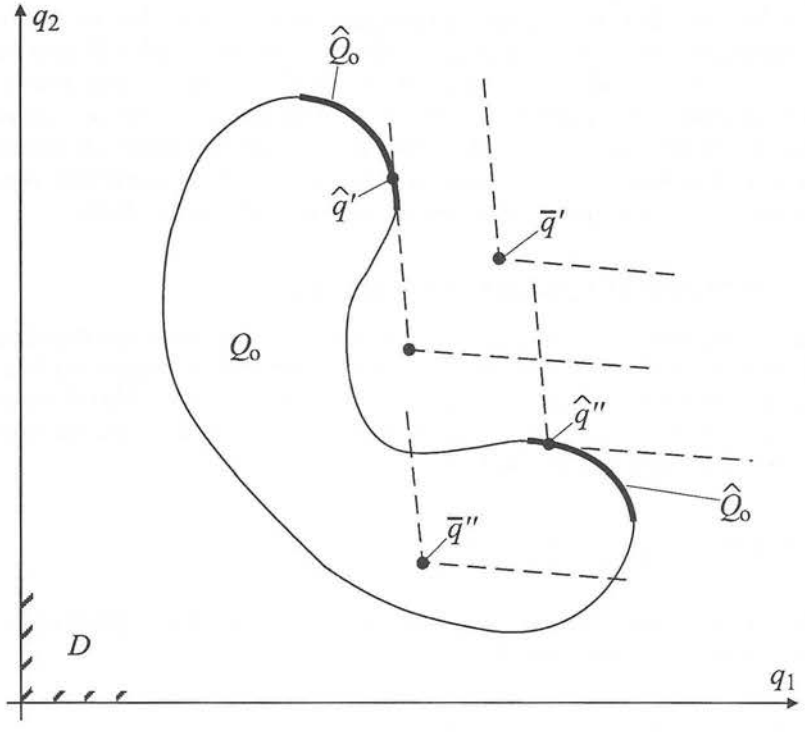

Figure 6. An illustrative example of the properties of reference point approach: $\hat{q}'$ is the maximum of the function (14) with the reference point $\bar{q}'$, and $\hat{q}''$ is the maximum of the function (14) with the reference point $\bar{q}''$; $\hat{Q}_0$ denotes the Pareto frontier of the set of attainable objectives Q_0 which is nonconvex in this example.

4.3. The decision process including outranking trials

We suppose that a multiobjective optimization model is investigated with the help of reference point methodology. We assume the following form of the decision process:

- **Stage I: Learning** – consists either of:
  1. Unrestricted change of reference points and learning about efficient outcomes;
  2. Application of any method with accelerated practical convergence.

  Stage I ends when the decision maker (DM) asks for support in checking whether a more preferred solution is available. We assume that DM preferences are stabilized at this point.
- **Stage II: Outranking Trials** – consists of a special convergent procedure using outranking relations and reference points. The use of outranking relations is justified because the DM is usually interested only in significant, not infinitesimal improvements, see Roy et al. (1981), Vincke (1998).

4.4. A family of outranking relations

In particular, we use the following family of outranking relations. The decision maker should define four threshold values for each criterion (objective outcome):

1. *veto threshold* $\Delta q_{av,i} > 0$,
2. *negative indifference threshold* $\Delta q_{ni,i} > 0$, $\Delta q_{ni,i} < \Delta q_{av,i}$
3. *component outranking threshold* $\Delta q_{co,i} > 0$,
4. *positive indifference threshold* $\Delta q_{pi,i} > 0$, $\Delta q_{pi,i} < \Delta q_{co,i}$

Given two points in the objective space: $\hat{q}$ (a current best point, usually already efficient or Pareto-optimal) and q (any point suspected of outranking the point $\hat{q}$), we can construct with these thresholds the following five index sets:

$$\begin{aligned} I_+ &= \{i \in \{1,\dots,p\} : q_i \geq \hat{q}_i + \Delta q_{pi,i}\} \\ I_- &= \{i \in \{1,\dots,p\} : q_i \leq \hat{q}_i - \Delta q_{ni,i}\} \\ I_0 &= \{1,\dots,p\} \setminus (I_+ \cup I_-) \\ I_c &= \{i \in \{1,\dots,p\} : q_i \geq \hat{q}_i + \Delta q_{co,i}\} \\ I_v &= \{i \in \{1,\dots,p\} : q_i \leq \hat{q}_i - \Delta q_{av,i}\} \end{aligned} \tag{16}$$

These index sets indicate the possibility of q outranking $\hat{q}$. If $I_v \neq \emptyset$, then there is no outranking. If $I_- = \emptyset$, then $I_v = \emptyset$; if, additionally, $I_c \neq \emptyset$, then there is *component outranking* of $\hat{q}$ by q. If $I_v = \emptyset$ and there are much more elements in I_+ than in I_-, then we can define parametrically an outranking by the difference of the cardinalities of these sets, $|I_+| - |I_-|$.

Thus, we define two outranking relations, where the second one is actually a parametric family of such relations, dependent on a number $k < p$, where p is

the number of objectives:

$$\begin{aligned} q\,\mathcal{C}\,\hat{q} &\Leftrightarrow (I_c \neq \emptyset) \wedge (I_- = \emptyset) \\ q\,\mathcal{P}_k\,\hat{q} &\Leftrightarrow (|I_+| - |I_-| \geq k) \wedge (I_v = \emptyset) \end{aligned} \tag{17}$$

It is easy to show that $\mathcal{C} \subseteq \mathcal{P}_k$, if $k = 1$ and that $\mathcal{C}$ is usually stronger than $\mathcal{P}_k$. Thus, outranking can be tested *with increasing sensitivity*:

1. start by checking if $q\,\mathcal{C}\,\hat{q}$ for some $q \in Q_0$,
2. then, set consecutively $k = p-1, p-2, \ldots, 1$ and check if $q\,\mathcal{P}_k\,\hat{q}$ for some $q \in Q_0$.

4.5. An outranking test with reference point methods

This test is based on the general property of order-consistent achievement scalarizing functions mentioned earlier:

$$\bar{q} \in Q_0 \Rightarrow \left\{ \begin{array}{ll} \max_{q \in Q_0} \sigma(q, \bar{q}) & \geq 0; \\ \hat{q} = \operatorname{argmax}_{q \in Q_0} \sigma(q, \bar{q}) & \geq \bar{q} \end{array} \right\} \tag{18}$$

Because of this property, in order to test $q\,\mathcal{C}\,\hat{q}$, it is sufficient to test the attainability of the reference point $\bar{q}^{(j)}$ defined as follows:

$$\bar{q}_i = \left\{ \begin{array}{ll} \hat{q}_i^{(j)} + \Delta q_{co,i}, & i = j \\ \hat{q}_i^{(j)} - \Delta q_{ni,i} & i \neq j \end{array} \right\}, \; j \in \{1, \ldots, p\} \tag{19}$$

If the maximum of $\sigma(q, \bar{q}^{(j)})$ is nonnegative, we can present the maximizing point to DM for acceptance. If the maximum is negative, no component outranking point exists for the objective j; we can repeat this (or compute in paralel and present to the DM) for all $j \in \{1, \ldots, p\}$.

In order to construct a test for $\mathcal{P}_k$ outranking, we define sets $I_+^{(j)}$, $I_-^{(j)}$, $I_0^{(j)}$ such that $|I_+^{(j)}| - |I_-^{(j)}| = k$, where the upper index (j) denotes one of possible subdivisions of the set $\{1, \ldots, p\}$ into such sets. The number $num(p, k)$ of such subdivisions can be generated in a combinatorial way; the sum of such numbers has an upper bound 2^p.

Using again the property (18) we observe that, in order to test for $q\,\mathcal{P}_k\,\hat{q}$, it is sufficient to test the attainability of the reference point $\bar{q}$ defined this time as follows:

$$\bar{q}_i^{(j)} = \left\{ \begin{array}{ll} \hat{q}_i + \Delta q_{pi,i}, & i \in I_+^{(j)} \\ \hat{q}_i - \Delta q_{av,i} & i \in I_-^{(j)} \\ \hat{q}_i - \Delta q_{ni,i} & i \in I_0^{(j)} \end{array} \right\}, \quad \begin{array}{l} i = 1, \ldots, p, \\ j = 1, \ldots, num(p, k) \end{array} \tag{20}$$

When testing the attainability of such reference points, we again maximize, for each $j = 1, \ldots, num(p, k)$, the order-consistent achievement function $\sigma(q, \bar{q}^{(j)})$. If the maximum value of this function is negative, no outranking

exists; if the maximum is nonnegative, the maximal point should be presented for acceptance to the decision maker. Again, we can use parallel computations and present to the decision maker possibly several points resulting from them. If no maximum with positive value of the achievement function can be found, we can decrease k and repeat the procedure; if no maximum with positive value of the achievement function can be found even for $k = 1$, no outranking points exist even in the weakest sense.

4.6. Outranking-value consistence axiom and convergence proof

It is now easy to show the convergence of Outranking Trials as described above. We need only to define the meaning of convergence; we are using outranking relations and convergence is typically understood in value function terms. Thus we assume that the decision maker has a value function (which has stabilized after the learning stage of the decision process) and that this function is consistent with the outranking relation. We need the following:

Outranking-value consistence axiom: Outranking relations defined above are consistent with a continuous, monotone value function $v(\boldsymbol{q})$ if there is a finite, positive value indifference threshold such that, if $v(\boldsymbol{q}) - v(\hat{\boldsymbol{q}}) \geq \Delta v$, then $\boldsymbol{q}$ outranks $\hat{\boldsymbol{q}}$ at least in some weak sense – and conversely.

Assuming outranking-value consistence, the proof of convergence for compact Q_0 and continuous $v(\boldsymbol{q})$ is immediate (we can use also a weaker assumption – that $sup_{\boldsymbol{q}} \in Q_0 < \infty$). We give here only an outline of the proof:

1. If the set Q_0 of attainable outcomes is compact, the continuous value function can only finitely increase .... Since Δv is finite, and every accepted step of the procedure produces a finite Δv, there might be only finitely many steps of the procedure in which an outranking point is found.
2. If no point with positive achievement can be found, then no attainable outranking point exists, due to the basic property (18). Because of outranking-value consistence axiom, this means that the value function is maximized at the endpoint with the accuracy of Δv.

Note that the proof is existential, not constructive: we cannot say in advance how many steps of the procedure are needed, because the relation between the four threshold values used in outranking relation and the increment Δv might be quite complicated.

4.7. Parallel computation aspects and concluding properties

The number of subsequent optimizations for one iteration of Outranking Trials can be large (up to 2^p, say 128 with $p = 7$). However, these computations can be made in parallel (or be distributed) and do not bother DM: she/he is only asked to approve of the decisions and outcomes suspected of outranking or to select between such decisions. Moreover, observe that the amount of computations in an iteration of Outranking Trials might be smaller in the initial iterations: as

soon as an outranking point is found and accepted by the decision maker, a new iteration starts. The amount of computations increases in final iterations, when it should be proven that no outranking points exist.

Preliminary (distributed) tests of Outranking Trials are being performed; the results until now show the applicability of the idea. An idea of combining Outranking Trials with a genetic algorithm is also being tested.

Outranking Trails offer also an alternative theoretical approach to the issue of convergence of other interactive procedures of multi-objective and decision support, such as Satisficing Trade-Off, Contracted Cone or Light Beam Search procedures.

5. Conclusions

We present here only some short conclusions stressing the main points presented here:

- Outranking Trials show an application of parametric embedding and parallel or distributed computations for the purpose of broadening analysis of multi-criteria models, in this case – proving convergence of computer-men interaction.
- Parametric embedding and parallel or distributed computations can be also used for increasing robustness of optimization procedures, as illustrated by Augmented Simplex and Variable Metric Pulsar algorithms.
- The basic objective of parallel computations – shortening of computations for difficult problems – will probably remain the main motivation (we always try to solve more difficult problems – and we always can saturate with them the most advanced computers!)
- However, it is good to know that there are also other objectives for parallelization and to use such objectives in practice.

References

BOGETOFT, P., HALLEFJORD, A. and KOK, M. (1988) On the convergence of reference point methods in multiobjective programming. *European Journal of Operations Research*, **34**, 56-68.

ERMOLEV, YU. and WETS, R.J-B. (1988) *Numerical Techniques for Stochastic Optimization.* Springer Verlag, Berlin-Heidelberg.

GEOFFRION, A.M., DYER, J.S. and FEINBERG, A. (1972) An Interactive Approach for Multicriterion Optimization, with an Application to the Operation of an Academic Department. *Management Science*, **19**, 357-368.

GUTENBAUM, J. (1977) Polyoptimization of systems with separate actions of performance indices, *Systems Science*, **3**, 283-293.

JASZKIEWICZ, A. and SŁOWIŃSKI, R. (1994) *The LBS package - a microcomputer implementation of the Light Beam Search method for multi- objective nonlinear mathematical programming.* Working Paper of the International

Institute for Applied Systems Analysis (IIASA), WP-94-07, Laxenburg, Austria.

KORHONEN, P. and LAAKSO, J. (1985) A Visual Interactive Method for Solving the Multiple Criteria Problem. *European Journal of Operational Research*, **24**, 277-287.

LEWANDOWSKI, A. and WIERZBICKI, A.P., eds. (1989) *Aspiration Based Decision Support Systems.* Lecture Notes in Economics and Mathematical Systems, **331**, Springer-Verlag, Berlin-Heidelberg.

NAKAYAMA, H. and SAWARAGI, Y. (1983) Satisficing trade-off method for multiobjective programming. In: M. Grauer and A.P. Wierzbicki, eds., *Interactive Decision Analysis.* Lecture Notes in Economics and Mathematical Systems, **229**, Springer-Verlag, Berlin-Heidelberg.

NAKAYAMA, H. (1994) Aspiration Level Approach to Interactive Multi-Objective Programming and its Applications. IIASA, WP-94-112, Laxenburg.

RIOS, S. (1994) *Decision Theory and Decision Analysis: Trends and Challenges.* Kluwer Academic Publishers, Boston-Dordrecht.

ROY, B. and VINCKE, PH. (1981) Multicriteria Analysis: Survey and New Directions. *European Journal of Operational Research*, **8**, 207-218.

SAWARAGI, Y., NAKAYAMA, H. and TANINO, T. (1985) *Theory of Multiobjective Optimization.* Academic Press, New York.

SOBCZYK, J. and WIERZBICKI, A.P. (1994) *Pulsar Algorithms - A Class of Coarse-Grained Parallel Optimization Algorithms.* IIASA Workshop on Parallel Methods in Large-Scale Optimization; IIASA, WP-94-53, Laxenburg, Austria.

STEUER, R.E. and CHOO, E.V. (1983) An interactive weighted Tchebycheff procedure for multiple objective programming. *Mathematical Programming*, **26**, 326-344.

STEUER, R.E. (1986) *Multiple Criteria Optimization: Theory, Computation, and Application.* J.Wiley & Sons, New York.

VINCKE, PH. (1992) *Multicriteria Decision Aid.* J.Wiley, Chichester-New York.

WIERZBICKI, A.P. (1980) The use of reference objectives in multiobjective optimization. In: G. Fandel, T. Gal, eds., *Multiple Criteria Decision Making; Theory and Applications,* Lecture Notes in Economic and Mathematical Systems, **177**, 468-486, Springer-Verlag, Berlin-Heidelberg.

WIERZBICKI, A.P. (1984) *Models and Sensitivity of Control Systems.* Elsevier-WNT, Amsterdam.

WIERZBICKI, A.P. (1986) On the completeness and constructiveness of parametric characterizations to vector optimization problems. *OR-Spektrum*, **8**, 73-87.

WIERZBICKI, A.P. (1993) *Augmented Simplex - a Modified and Parallel Version of Simplex Method Based on Multiple Objective and Subdifferential Optimization Approach.* IIASA Workshop on Advances in Methodology and Software for Decision Support Systems; IIASA, WP-93-059, Laxen-

burg, Austria.

WIERZBICKI, A.P. (1997) Convergence of Interactive Procedures of Multiobjective Optimization and Decision Support. In: M. Karwank, J. Spronk, J. Wallenius, eds., *Essays in Decision Making.* Springer Verlag, Berlin.

WIERZBICKI, A.P. (1999) Reference Point Approaches. In: T. Gal, T.J. Stewart, T.Hanne, eds., *Multicriteria Decision Making: Advances in MCDM Models, Algorithms, Theory, and Applications.* Kluwer Academic Publishers, Boston/Dordrecht/New York.

ZIONTS, S. AND J. WALLENIUS (1976) An Interactive Programming Method for Solving the Multicriteria Problem. *Management Science*, **22**, 653-663.

ZIONTS, S. AND J. WALLENIUS (1983) An Interactive Multiple Objective Linear Programming Method for a Class of Underlying Nonlinear Utility Functions. *Management Science*, **29**, 519-529.