Control and Cybernetics

vol. **36** (2007) No. 2

A method of approximating Pareto sets for assessments of implicit Pareto set elements

by

Ignacy Kaliszewski

Systems Research Institute, Polish Academy of Sciences
ul. Newelska 6, 01-447 Warszawa, Poland
e-mail: ignacy.kaliszewski@ibspan.waw.pl

Abstract: Deriving efficient variants in complex multiple criteria decision making problems requires optimization. This hampers greatly broad use of any multiple criteria decision making method.

In multiple criteria decision making Pareto sets, i.e. sets of efficient vectors of criteria values corresponding to feasible decision alternatives, are of primal interest. Recently, methods have been proposed to calculate assessments for any implicit element of a Pareto set (i.e. element which has not been derived explicitly but has been designated in a form which allows its explicit derivation, if required) when a finite representation of the Pareto set is known. In that case calculating respective bounds involves only elementary operations on numbers and does not require optimization.

In this paper the problem of approximating Pareto sets by finite representations which assure required tightness of bounds is considered for bicriteria decision making problems. Properties of a procedure to derive such representations and its numerical behavior are investigated.

Keywords: multiple criteria decision making, bi-criteria problems, Pareto set, efficiency, convex function approximation, sandwich algorithms.

1. Introduction

In multiple criteria decision making, Pareto sets, i.e. sets of efficient vectors of criteria values corresponding to feasible decision alternatives, are of primal interest.

In this paper we are concerned with the problem of providing finite representations of Pareto sets with the required accuracy. We focus on a class of problems in which Pareto sets possess the property of convexity.

This research is motivated by the earlier works of the author in which methods to calculate assessments for any implicit efficient element of a Pareto set (i.e.

element which has not been derived explicitly but has been designated in a form which allows its explicit derivation, if required) were proposed (Kaliszewski, 2004, 2006). These methods rely on finite representations (subsets) of Pareto sets, and bound tightness depends on accuracy of those representations. Assessments have forms of bounds, lower and upper, on element components. Calculating bounds involves only elementary operations on numbers and does not require optimization. Bounds of sufficient tightness are an attractive alternative to deriving Pareto sets elements explicitly, which, as a rule, involves optimization (Kaliszewski, 2004, 2006).

We confine ourselves here to bi-criteria decision making problems in which Pareto sets have forms of convex curves. This class of problems includes various models resulting from portfolio investment, which have gained much popularity in finance and optimization literature (Markowitz, 1959; Elton, Gruber, 1995; Ogryczak, 2002).

To ensure required accuracy of finite representations of Pareto sets we follow in broad lines a stream of papers devoted to approximations of convex curves by a finite number of points (Cohon, 1978; Burkard et al., 1987; Fruhwirth et al., 1988; Ruhe, Fruhwirth, 1989; Yang, Goh, 1997), where algorithms with guaranteed quadratic convergence were proposed. However, the specific application of Pareto set approximations to calculate assessments of Pareto set elements renders arguments presented in those works not directly applicable for our purpose.

The plan of the paper is as follows. In Section 2 we recall the formulation of the underlying model for multiple criteria decision making (MCDM) problems. In Section 3 we recall how assessments for implicit elements of Pareto sets can be calculated for an instance of the underlying model considered in this paper. Section 4 is devoted to the problem of controlling bound tightness by a method proposed for that purpose. Section 5 contains results of investigations of numerical behavior of this method, whereas Section 6 concludes.

2. Preliminaries

We consider the following underlying model for MCDM problems:

$$vmax\ f(x),\ \ x \in X_0 \subseteq \mathcal{X}, \tag{1}$$

where *vmax* stands for the identification of all efficient alternatives (we assume that all criteria are of "better if more" type), $\mathcal{X}$ is the set (space) of potential alternatives, X_0 is the set of feasible alternatives, $f : \mathcal{X} \rightarrow \mathcal{R}^k$ is the criteria map in which $f = (f_1, ..., f_k)$, and $f_i : \mathcal{X} \rightarrow \mathcal{R}$ are *criteria functions*, $i = 1, ..., k$, $k \geq 2$.

By y and Z we denote

$$y = f(x),\ \ Z = f(X_0).$$

The set of all y efficient (in this paper we refer to standard definitions of proper efficiency, efficiency, and weak efficiency, see, e.g., Miettinen, 1999; Ehrgott, 2006; Kaliszewski, 2006) in $f(X_0)$ we call *Pareto set.* Z is said to be R^k_+-convex if $Z - R^k_+$ is convex, where $R^k_+ = \{y \mid y_i \geq 0,\ i = 1, ..., k\}$.

Below we make use of selected element y^* defined as

$$y^*_i = \hat{y}_i + \epsilon,\ i = 1, ..., k,$$

where $\hat{y}$ is calculated as

$$\hat{y}_i = max_{y \in Z}\ y_i,\ i = 1, ..., k,$$

(we assume that all these maxima exist), ϵ is any positive number.

3. Bounds on Pareto set elements

In Kaliszewski (2004, 2006) methods to calculate lower and upper bounds on components of elements of Pareto sets were proposed. The methods require that a finite subset S, called *shell*, of a Pareto set is explicitly given. It is assumed that elements of S are derived by solving the optimization problem

$$\min_{y \in Z} \sum_i \lambda_i y_i, \tag{2}$$

or

$$\min_{y \in Z} \max_i \lambda_i (y^*_i - y_i) + \rho e^k (y^* - y), \tag{3}$$

or

$$\min_{y \in Z} \max_i \lambda_i ((y^*_i - y_i) + \rho e^k (y^* - y)),$$

where $\lambda_i > 0,\ i = 1, ..., k,\ \rho \geq 0,\ e^k = (1, 1, ..., 1)$.

Any of these optimization problems could also be used to derive explicitly an implicit element $y(\lambda)$ of the Pareto set (i.e. an element which has not been derived but has been designated by selecting the optimization problem and vector of weights λ). An alternative to this is to calculate bounds such that

$$L_i(S, \lambda) \leq y_i(\lambda) \leq U_i(S, \lambda),\ i = 1, ..., k,$$

and those calculations involve only elementary operations on numbers. For bound formulas see Kaliszewski (2006).

In the same work another and more straightforward method to calculate parametric bounds was proposed for $k = 2$ under the assumption that Z is R^2_+-convex and optimization problem (3) with $\rho = 0$ would be used to derive explicitly the elements of the Pareto set. Below we briefly present that method.

If Z is R^2_+-convex, then deriving elements of the Pareto set by solving optimization problem (3) with $\rho = 0$ is equivalent to finding the intercept of the half line $y = y^* - \tau t,\ t \geq 0$, with Z, where

$$\tau_i = \lambda_i^{-1},\ i = 1, 2, \tag{4}$$

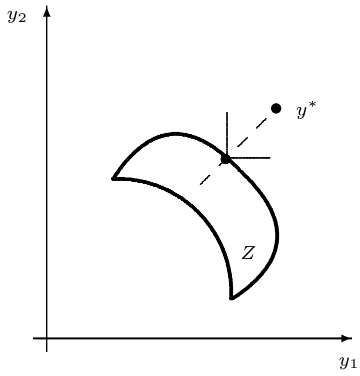

Figure 1. An interpretation of solving the optimization problem (3) with $\rho = 0$ when Z is R^2_+-convex.

(see Fig. 1, this is in fact true for any k).

Suppose that shell S has been derived by solving optimization problem (2) for n vectors $\lambda^s,\ \lambda_i^s > 0,\ i = 1,2,\ s = 1,...,n$. In other words, $S = \{y(\lambda^s)\},\ s = 1,...,n$, where each $y(\lambda^s)$ is a solution (an *explicit* element of the Pareto set, as opposed to implicit elements defined above) of (2).

Without loss of generality we can assume that elements of S are ordered with respect to decreasing values of the first component. Pairs of elements of S which correspond to two successive values of this component are called *neighbor elements*. Neighbor elements $y',\ y''$ define a *cutting line*

$$y = \alpha y' + (1-\alpha)y'',\ \alpha \in \mathcal{R}.$$

There are $n-1$ neighbor elements and $n-1$ cutting lines.

Cutting lines form a lower approximation of the Pareto set (see Fig. 2).

A lower bound $L(S,\lambda)$ for $y(\lambda)$ is given by the intercept of the half line $y = y^* - \tau t,\ t \geq 0$, with that cutting line for which t has the largest value (see Fig. 2, a lower bound marked by diamond).

Indeed, since Z is R^k_+-convex, no Pareto set element can be found "below" (in the sense that y^* lies "above") any line segment connecting neighbor elements of the shell.

Elements of S define n *supporting lines*

$$\lambda^s y = \lambda^s y(\lambda^s),\ y(\lambda^s) \in S.$$

Supporting lines form an upper approximation of the Pareto set (see Fig. 2).

An upper bound $U(S,\lambda)$ for $y(\lambda)$ is given by the intercept of the half line $y = y^* - \tau t,\ t \geq 0$, with that supporting line for which t has the largest value (see Fig. 2, an upper bound marked by an asterisk).

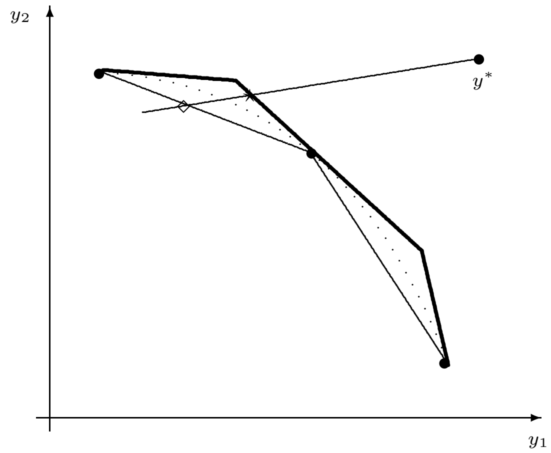

Figure 2. Lower (*thin*) and upper (*thick*) approximations of the Pareto set (*dotted*).

Indeed, no Pareto set element can be found "above" (in the sense that y^* lies "above") any line $\lambda^s y(\lambda^s) = \lambda^s y,\ y(\lambda^s) \in S$, because $y(\lambda^s)$ solves $\max_{y \in Z} \lambda^s y$.

Some precautions are to be taken for half line $y = y^* - \tau t,\ t \geq 0$, not to miss the Pareto set, and this amounts to putting some bounds on values λ_1 and λ_2, depending on the value of parameter ϵ used to define y^* (see Kaliszewski, 2006).

4. Controlling bound tightness

Given $\lambda > 0$, bound tightness can be defined by

$$err(\lambda) = ||U(\lambda, S) - L(\lambda, S)||_2, \tag{5}$$

where $|| \cdot ||_2$ is the Euclidean norm.

Clearly, bound tightness increases ($err(\lambda)$ decreases), not necessarily monotonously, with the growing number of elements in S. Here we propose a metod to control bound tightness, which ensures that

$$err = \max_{\lambda} err(\lambda)$$

is kept within the required limit. The method consists of shell building by a "check and expand" procedure.

Observe that in each triangle formed by the lower and upper approximation of the Pareto set $err(\lambda)$ takes its maximal value for that λ, for which half line $y = y^* - \tau t,\ t \geq 0$, where $\tau_i = \lambda_i^{-1},\ i = 1, ..., k$, passes through the outer vertex of the triangle (see Fig. 3, length of segment AB is equal to maximal value of $err(\lambda)$ in the triangle).

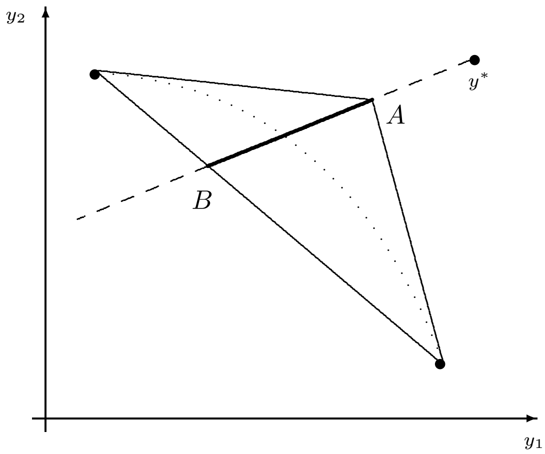

Figure 3.

For a given shell S (initially S should be composed of at least the "leftmost" and the "rightmost" element of the Pareto set), err is, presumedly, reduced if a new element of the Pareto set is derived (and S is expanded by that element) in the triangle in which $err(\lambda)$ takes the maximal value.

In order to derive a new element of the Pareto set, following Yang and Goh (1997) we propose to maximize over Z the function

$$y_1 + ry_2\,, \tag{6}$$

where $r = \frac{y_1''-y_1'}{y_2'-y_2''}$.

Graphically, this is equivalent to shifting a hyperplane parallel to the cutting line

$$y = \alpha y' + (1-\alpha)y'',\ \alpha \in \mathcal{R},$$

where y', y'' are neighbor elements in the triangle, till it becomes a supporting line of Z at an element y''' of the Pareto set (see Fig. 4). We assume that the newly derived element of the Pareto set does not belong to line segment $y'y''$, for otherwise err in this triangle would be equal to 0. The new supporting line, together with the new cutting lines formed by neighbor elements y', y''' and y''', y'' give rise to a more tight approximation of the Pareto set and, presumedly, to lower value of err (see Fig. 5).

To establish convergence of this bound tightness enforcement method, we should establish a relation between the maximal $err(\lambda)$ in a triangle (e.g. the length of line segment AB in Fig. 4) and the maximum $err(\lambda)$ in two new triangles resulting from shell expansion by an element (e.g. the length of line segment CD in Fig. 5).

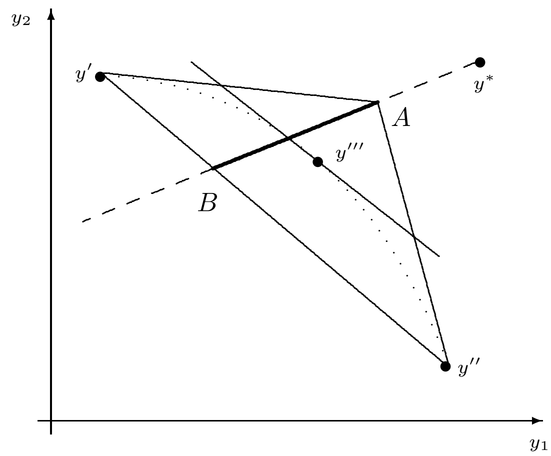

Figure 4.

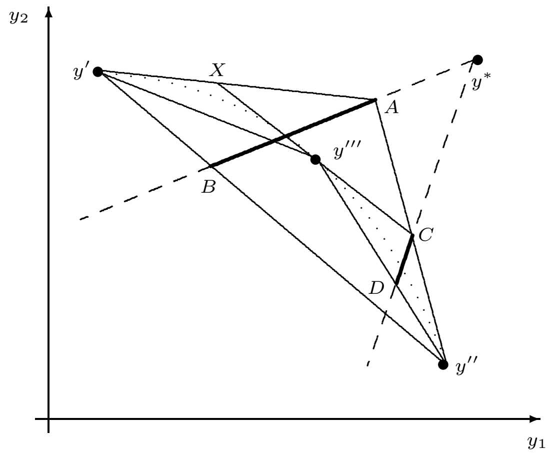

Figure 5.

The maximal $err(\lambda)$ in the new lower triangle occurs when the new element is located at point X (Fig. 6) (in this case the upper triangle reduces to a line segment). Symmetrically, the maximal $err(\lambda)$ in the new upper triangle occurs when the new element is located at the point C (in this case the lower triangle reduces to a line segment). Let us assume that the former case holds (Fig. 6).

Below, to find length of line segment AB we derive points A and B. Then, we look for a formula giving length of line segment CD as a function of element y''', which can lie anywhere on the line segment $A\,y'$.

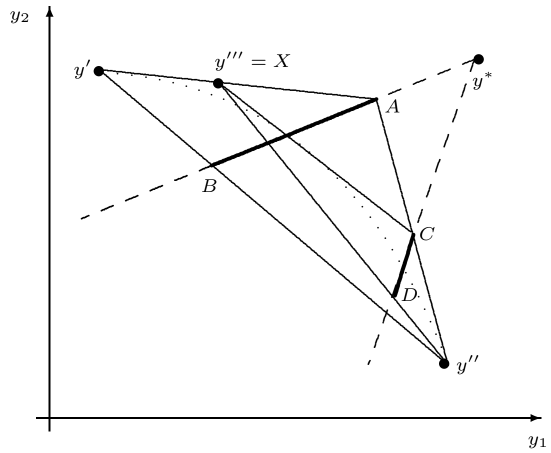

Figure 6.

Derivation of point A

Point A is found as the intersection of the supporting line at y' and the supporting line at y''.

The supporting line at y' has the form $\lambda' y = a'$ where $a' = \lambda' y'$ and the supporting line at y'' has the form $\lambda'' y = a''$ where $a'' = \lambda'' y''$. Vectors λ' and λ'' are known since they are the vectors used to derive shell elements y' and y'', respectively, by solving the optimization problem (2).

Point A is then given by the formula

$$A_1 = \frac{1}{\lambda'_1}(a' - \lambda'_2 \frac{\lambda'_1 a'' - \lambda''_1 a'}{\lambda'_1 \lambda''_2 - \lambda''_1 \lambda'_2}),$$
$$A_2 = \frac{\lambda'_1 a'' - \lambda''_1 a'}{\lambda'_1 \lambda''_2 - \lambda''_1 \lambda'_2}. \qquad \blacksquare$$

Derivation of point B

Point B is found as the intersection of the line passing through y' and y'', which is given by formula (6), and the line passing through y^* and A. The latter has the form

$$y_1 + s y_2 = y_1^* + s y_2^*,$$

where $s = \frac{A_1 - y_1^*}{y_2^* - A_2}$. Clearly (see Fig. 6), $r \neq s$.

Thus, point B is given by the formula

$$B_1 = y_1^* + s y_2^* - s \frac{y'_1 + r y'_2 - y_1^* - s y_2^*}{r - s},$$
$$B_2 = \frac{y'_1 + r y'_2 - y_1^* - s y_2^*}{r - s}. \qquad \blacksquare$$

Derivation of point C

Point C is found as the intersection of the new supporting line passing through y''' and the line passing through y'' and A.

The new supporting line has the form

$$y_1 + ry_2 = y_1''' + ry_2''',$$

where r is given by formula (6) (the new supporting line is parallel to the line segment $y'y''$).

The line passing through y'' and A has the form

$$y_1 + uy_2 = A_1 + uA_2\,,$$

where $u = \frac{A_1 - y_1''}{y_2'' - A_2}$. Clearly (see Fig. 6), $r \neq u$.

Thus, point C is given by the formula

$$C_1 = y_1''' + ry_2''' - r\frac{A_1 + uA_2 - y_1''' - ry_2'''}{u - r}\,,$$
$$C_2 = \frac{A_1 + uA_2 - y_1''' - ry_2'''}{u - r}\,. \qquad \blacksquare$$

Derivation of point D

Point D is found as the intersection of the line passing through y^* and point C, and the line passing through y'' and y'''.

The line passing through y^* and C has the form

$$y_1 + wy_2 = C_1 + wC_2\,,$$

where $w = \frac{C_1 - y_1^*}{y_2^* - C_2}$.

The line passing through y'' and y''' has the form

$$y_1 + zy_2 = y_1''' + zy_2'''\,,$$

where $z = \frac{y_1''' - y_1''}{y_2'' - y_2'''}$. Clearly (see Fig. 6), $w \neq z$.

Thus, point D is given by the formula

$$D_1 = y_1''' + zy_2''' - z\frac{C_1 + wC_2 - y_1''' - zy_2'''}{w - z}\,,$$
$$D_2 = \frac{C_1 + wC_2 - y_1''' - zy_2'''}{w - z}\,. \qquad \blacksquare$$

To establish convergence of the method one needs to know the maximal errors at successive iterations. It is easy to observe that at any iteration the maximal value of the error is attained for y''' somewhere in the interior of the line segment $A\,y'$ since the error approaches zero as y''' approaches either A or y'.

The square of length of line segment CD (square of err) is given by

$$
\begin{aligned}
L &= (||C-D||_2)^2 \\
&= \left((r-s)y_2''' - r\frac{A_1+uA_2-y_1'''}{u-r} + z\frac{C_1+wC_2-y_1'''-zy_2'''}{w-z}\right)^2 \\
&+ \left(\frac{A_1+uA_2-y_1'''-ry_2'''}{u-r} - \frac{C_1+wC_2-y_1'''-zy_2'''}{w-z}\right)^2
\end{aligned}
\tag{7}
$$

and is a function of y'''. With y''' constrained to the line segment $A\,y'$, i.e. $y''' = \alpha A + (1-\alpha)y',\ \alpha > 0$, L is to be maximized with respect to α. However, this is a highly nonlinear problem and does not offer an analytical solution. Therefore, in the next section we investigate the convergence behavior of the method numerically.

5. Numerical experiments

To investigate the numerical behavior of the method, we tested it on a set of problems.

Test problems

The set of test problems consisted of two groups.

The first group consisted of shells of five mean-variance portfolio problems. As known, Pareto sets of mean-variance efficient portfolios are R^2_+-convex if represented with $y_1 = -variance$ and $y_2 = mean$. Data were taken from Chang T.-J. et al. (2000) (they are accessible via Internet from the so-called Beasley's OR-Library, see references).

The second group consisted of shells of four quadratic curves $y_1 = -a * y_2^2 + 10$ with a equal to $1, 5, 50, 500$, selected to represent Pareto sets of R^2_+-convex problems.

Each test problem was a collection of 2000 elements $y^i = \{(y_1^i, y_2^i)\}$ ordered with respect to decreasing values of the first component. To simplify calculations vector λ maximizing λy at given y was approximated by the normal vector of the line passing through y', y'', where $(y', y),\ (y, y'')$ are neighbor elements. To make this procedure applicable for elements 1 and 2000 two additional elements y^0 and y^{2001} were constructed in the following way: $y_i^0 = y_i^1 - (y_i^2 - y_i^1)$ and $y_i^{2001} = y_i^{2000} + (y_i^{1999} - y_i^{2000}),\ i = 1, 2$. Thus, (y^0, y^1) and (y^{2000}, y^{2001}) are neighbor elements.

The element maximizing the function (6) was found by enumeration over 2000 points. Because of the large number of points (dense discrete representations of the Pareto sets) this was an insignificant simplification to maximizing this function by optimization.

Test problems are graphically represented in Fig. 7 (the first group), in Fig. 8 (the second group, problems with $a = 1$ and $a = 5$), and in Fig. 9 (the second group, problems with $a = 50$ and $a = 500$). Because of large number (2000) of elements in each test problem all those figures look like graphs of continuous curves, but in fact they represent finite sets.

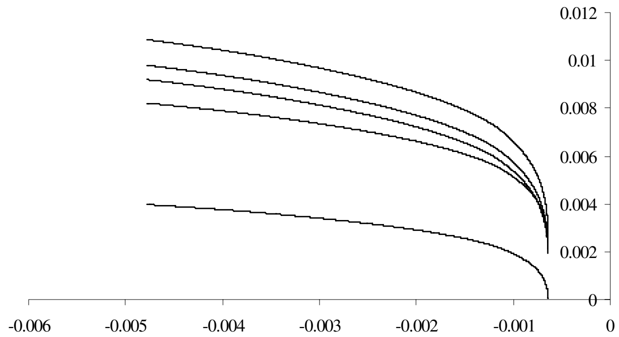

Figure 7.

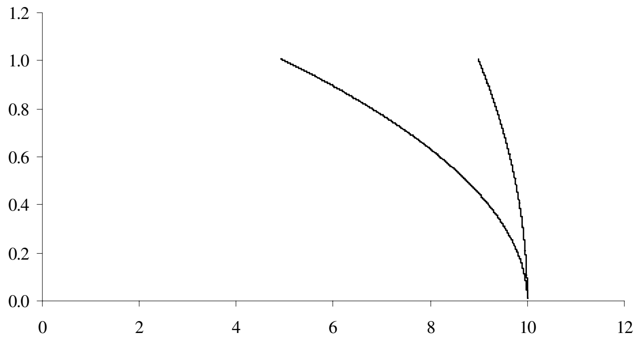

Figure 8.

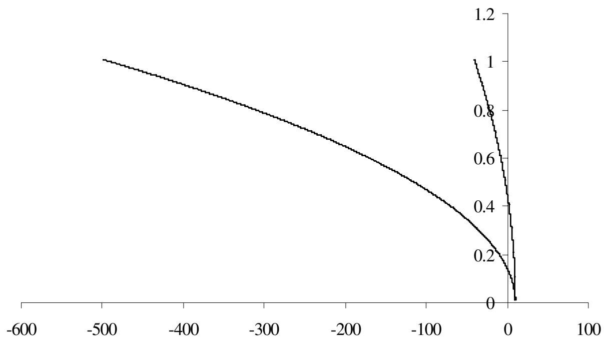

Figure 9.

Computations
For each test problem 100 iterations were performed and on each iteration error $||A - B||_2$ was calculated.

For all nine test problems in 100 iterations the initial error was reduced by four orders of magnitude. To approximate the error reduction rate quadratic curves were fitted to logarithm of errors (Fig. 10 represents errors and the fitted curve for the first test problem from the first group). The parameters of best fit curves for all nine test problems are given in Table 1.

As seen from the parameters of fitting curves the method to control bound tightness behaved in a very stable manner over a set of varied test problems. The error reduction speed, represented by parameters a and b of the fitting curves, differed over test problems insignificantly (standard deviation of 0.000053 and 0.006137, respectively), despite the different ranges of component values in the test problems.

It is not so that the error is always nonincreasing, as shown by the test problem Portef-gen4, for which the error of the first iteration 4.82 increased in the second iteration to 35.89 to drop to 12.110 in the third iteration and to decrease monotonously on succeeding iterations. This phenomenon is attributed to very different ranges of component values in this test problem and can be explained as in Figs. 11 and 12. In the first iteration (Fig. 11) line segment AB (between shaded triangle and shaded diamond) is almost parallel to the vertical axis, so that its length is small. In the second iteration (Fig. 12) line segment AB (between triangle and diamond) is extended along the horizontal axis, which results in its length being greater than it was in the first iteration. In general, because of error dependence on the element y^* (shaded disc in Figs. 11 and 12), two identical triangles formed by the lower and upper approximations of a Pareto set in two different locations can yield two different errors (see Fig. 2).

Table 1. Parameters of best fit curves.

Group	Problem	a	b	c
1	Portef1	0.000787	-0.135603	-7.746731
1	Portef2	0.000805	-0.136914	-7.946023
1	Portef3	0.000796	-0.136594	-8.546667
1	Portef4	0.000806	-0.137219	-7.780915
1	Portef5	0.000829	-0.139572	-8.737797
2	Portef-gen1	0.000840	-0.141076	-3.111671
2	Portef-gen2	0.000884	-0.144575	-1.694186
2	Portef-gen3	0.000885	-0.144247	0.090437
2	Portef-gen4	0.000697	-0.122615	1.586368
	mean	0.000814	-0.137601	-4.876353
	standard dev.	0.000053	0.006137	3.860863

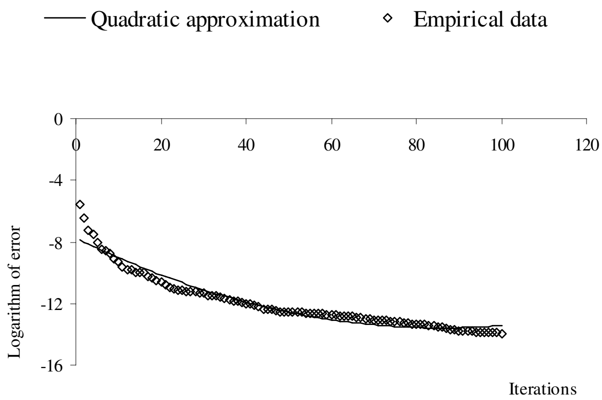

Figure 10.

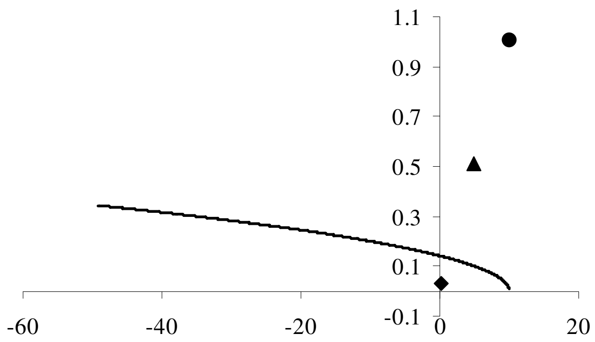

Figure 11.

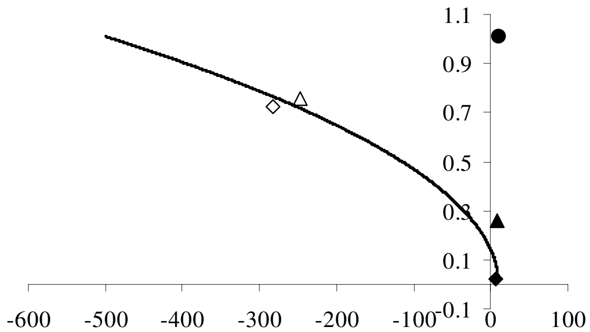

Figure 12.

6. Concluding remarks

Assessments of Pareto set elements allow to carry out decision making processes in a unified way, regardless of model (1) complexity. Model complexity plays a role only when the shell is derived, but shell derivation can be made prior to the start of the decision making process by a person (or a team) independent of the decision maker. Thus, computational complexity of decision making processes can be reduced to elementary operation on numbers easily realisable e.g. with a spreadsheet.

The numerical results presented in the previous section constitute additional argument to those given in earlier works (Kaliszewski, 2004, 2006), that assessments of Pareto set elements are a viable and versatile tool to ease decision making processes from the burden of optimization computations. This time we showed that for the bicriteria case the process of shell building to achieve the required bound tightness can be reasonably fast. This, we believe, opens a way to a broader use in MCDM methods of Pareto set elements assessments, instead of deriving such elements explicitly by optimization.

Unfortunately, the procedure for shell derivation discussed in this paper does not offers a straightforward generalization to $k > 2$. In that case other procedures (see Kaliszewski, 2006) have to be employed and their effectiveness investigated.

References

Beasley's OR-Library at http://people.brunel.ac.uk/~mastjjb/jeb/orlib/portinfo.html.

Burkard, R.E., Hamacher, H.W. and Rote, G. (1987) Approximation of convex functions and applications in mathematical programming. *Report 89, Institut für Mathematik*, Technische Universität Graz.

Cohon, J.L. (1978) *Multiobjective Programming and Planning.* Academic Press, New York.

Chang, T.-J., Meade, N., Beasley, J.E. and Sharaiha, Y.M. (2000) Heuristics for cardinality constrained portfolio optimisation. *Computers & Operations Reasearch* **27**, 1271–1302.

Elton, E.J. and Gruber, M.J. (1995) *Modern Portfolio Theory and Investment Analysis.* John Wiley & Sons, New York.

Fruhwirth, B., Burkard, R.E. and Rothe, G. (1989) Approximation of convex curves with applications to the bicriterial minimum cost flow problem. *European Journal of Operational Research* **42**, 326–338.

Kaliszewski, I. (2004) Out of the mist – towards decision-maker-friendly multiple criteria decision making support. *European Journal of Operational Research* **158**, 293–307.

Kaliszewski, I. (2006) *Soft Computing for Complex Multiple Criteria Decision Making.* Springer.

MARKOWITZ, H.M. (1959) *Portfolio Selection, Efficient Diversification of Investments.* John Wiley & Sons, New York.

MIETTINEN, K.M. (1999) *Nonlinear Multiobjective Optimization.* Kluwer Academic Publishers, Dordrecht.

OGRYCZAK, W. (2002) Multiple criteria optimization and decision under risk. *Control and Cybernetics* **31**, 975–1003.

RUHE, G. and FRUHWIRTH B. (1990) ϵ-optimality for bicriteria programs and its application to minimum cost flows. *Computing* **44**, 21–34.

YANG, X.Q. and GOH, C.J. (1997) A method for convex curve approximation. *European Journal of Operations Research* **97**, 205–212.