Math. Meth. Oper. Res. (2005) 62: 167–185
DOI 10.1007/s00186-005-0018-4

ORIGINAL ARTICLE

Nikolai D. Botkin · Josef Stoer

Minimization of convex functions on the convex hull of a point set

Received: September 2004 / Revised: April 2005 / Published online: 6 October 2005

Abstract A basic algorithm for the minimization of a differentiable convex function (in particular, a strictly convex quadratic function) defined on the convex hull of m points in R^n is outlined. Each iteration of the algorithm is implemented in barycentric coordinates, the number of which is equal to m. The method is based on a new procedure for finding the projection of the gradient of the objective function onto a simplicial cone in R^m, which is the tangent cone at the current point to the simplex defined by the usual constraints on barycentric coordinates. It is shown that this projection can be computed in $O(m \log m)$ operations. For strictly convex quadratic functions, the basic method can be refined to a noniterative method terminating with the optimal solution.

Keywords Convex functions on simplexes · Barycentric coordinates · Projection of directions on simplexes

1 Formulation of the problem

Let $\mathcal{Z} = \{z^1, z^2, \dots, z^m\}$ be a finite set of points in the Euclidean space R^n and $Z := (z^1, \dots, z^m)$ the $n \times m$-matrix with columns z^i. Let $f : R^n \to R$ be a differentiable convex function. We consider the following problem

$$\text{minimize} \quad f(x) \quad \text{subject to} \quad x \in \operatorname{co} \mathcal{Z}. \tag{1}$$

N. D. Botkin (✉)
Center of advanced european studies and research (caesar), Ludwig-Erhard-Allee 2, D-53175 Bonn, Germany
E-mail: botkin@caesar.de

J. Stoer
Institut für Angewandte Mathematik und Statistik der Universität Würzburg
Am Hubland, D-97074 Würzburg, Germany
E-mail: jstoer@mathematik.uni-wuerzburg.de

Obviously, problem (1) is equivalent to

$$\text{minimize } g(\alpha) \quad \text{subject to } \alpha \in \Lambda, \tag{2}$$

where

$$g(\alpha) := f\left(\sum_{i=1}^{m} \alpha_i z^i\right) = f(Z\alpha),$$

$$\Lambda := \left\{\alpha \in R^m : \alpha_i \geq 0, \sum_{i=1}^{m} \alpha_i = 1\right\}.$$

There are typical difficulties with these problems: the solution of (1) by a direct minimization of f on the polyhedron $P := \text{co}\,\mathcal{Z}$ by efficient standard methods like the SQP-method is not feasible, because these methods require the knowledge of a representation of $P = \{x \mid a_i^T x \leq b_i,\ i = 1, \ldots, N\}$ as intersection of finitely many halfspaces, which is hard to compute from $\mathcal{Z}$ and is unknown in general.

On the other hand, the solution of (2) leads to other difficulties. If m is larger than $d := \dim(\text{co}\,\mathcal{Z})$ (being at most n), then the affine map $A\colon \Lambda \to \text{co}\,\mathcal{Z}$ given by $\alpha \mapsto Z\alpha$ is not one to one, and in general any $x \in \text{co}\,\mathcal{Z}$ has more than one representation of the form $x = Z\alpha$, $\alpha \in \Lambda$ (recall in this context a classical result of Carathéodory that each $x \in \text{co}\,\mathcal{Z}$ has a representation $x = Z\alpha$ with an $\alpha \in \Lambda$ having at most $d + 1$ nonzero components). So, in this situation, problem (2) may have more than one optimal solution even for strictly convex functions f, where the minimizer of (1) is unique. In particular, if f is strictly convex quadratic, then g is only convex quadratic (though bounded below on R^m). Moreover, the map A will then not preserve extreme points or faces. So for every subset $I \subseteq \{1, \ldots, m\}$,

$$\Lambda(I) := \{\alpha \in \Lambda \mid \alpha_i = 0 \quad \text{for } i \in I\}$$

is a face of Λ, but in general $A(\Lambda(I))$ is not a face of $\text{co}\,\mathcal{Z}$, even though all faces of $\text{co}\,\mathcal{Z}$ have the form $A(\Lambda(I))$ for certain I. If a representation of the polyhedron $\text{co}\,\mathcal{Z}$ as intersection of finitely many halfspaces is not known, it will be even very difficult to decide, which of the z^i are relative interior points, relative boundary points, or even extreme points of $\text{co}\,\mathcal{Z}$.

With line search type algorithms for solving (2) the following may then occur: a line segment L in Λ connecting two boundary points of Λ may correspond to a line segment $A(L)$ connecting two points of $\text{co}\,\mathcal{Z}$ that are not necessarily (relative) boundary points of $\text{co}\,\mathcal{Z}$.

Problems of type (1) with the special quadratic function

$$f(x) := (x - x^c)^T (x - x^c),$$

where $x^c \in R^n$, occur in several applications, for example in computational geometry when one wishes to find the orthogonal projection of a point x^c to $\text{co}\,\mathcal{Z}$. Also certain problems in mathematical finance (portfolio theory) lead to such problems: then, for example, $\alpha \in \Lambda$ represents the composition of a portfolio.

We note at this point that the minimization of *any* strictly convex quadratic function is equivalent to an orthogonal projection problem: in fact, if $f(x) :=$

$x^T Bx - b^T x$ and B is positive definite with Cholesky decomposition $B = L^T L$ then

$$f(x) = (Lx - x^c)^T (Lx - x^c) - (x^c)^T x^c, \quad x^c := \frac{1}{2} L^{-T} b.$$

Therefore minimizing f on co $\mathcal{Z}$ is equivalent to computing the orthogonal projection of x^c to co $\tilde{\mathcal{Z}}$, $\tilde{\mathcal{Z}} := \{Lz^1, \dots, Lz^m\}$. This prior reduction requires $O(mn^2)$ operations to compute $\tilde{\mathcal{Z}}$.

Michelot (1986) has described a very elegant algorithm for computing the orthogonal projection of a point $x^c \in R^m$ to the *canonical* simplex Λ of R^m. However, his algorithm does not allow to compute the orthogonal projection of x^c to a *nonstandard* simplex co $\mathcal{Z}$ spanned by *arbitrary m* affinely independent vectors z^i, $i = 1, 2, \dots, m$.

Denote by Λ^* the set of minimizers of (2), and for arbitrary $\alpha \in \Lambda$ the set of admissible directions $\mathcal{P}(\alpha)$ by

$$\mathcal{P}(\alpha) = \left\{ p \in R^m : \sum_{i=1}^{m} p_i^2 \leq 1, \quad \exists \lambda > 0 \text{ such that } \alpha + \lambda p \in \Lambda \right\}.$$

It is easily seen that $\mathcal{P}(\alpha)$ is a convex set given by the formula

$$\mathcal{P}(\alpha) = \left\{ p \in R^m : \sum_{i=1}^{m} p_i^2 \leq 1, \ \sum_{i=1}^{m} p_i = 0, \quad p_j \geq 0, \ j \in I^0(\alpha) \right\},$$

where

$$I^0(\alpha) = \{i \in \overline{1, m} : \alpha_i = 0\}.$$

For $\alpha \in \Lambda$ and $h \in R^m$, we define

$$\omega(\alpha, h) := \min_{p \in \mathcal{P}(\alpha)} \langle h, p \rangle, \quad p^0(\alpha, h) := \arg \min_{p \in \mathcal{P}(\alpha)} \langle h, p \rangle, \tag{3}$$

where $\langle h, p \rangle := h^T p$ is the scalar product in R^m. By $|h| := \langle h, h \rangle^{1/2}$ we denote the euclidean norm. The vector $p^0(\alpha, h)$ is called the projection of the direction h onto $\mathcal{P}(\alpha)$.

The following propositions can be easily obtained using properties of convex functions.

Proposition 1 *Let* $\alpha \in \Lambda$ *and* $p \in \mathcal{P}(\alpha)$. *If* $\langle \nabla g(\alpha), p \rangle < 0$, *then there exists* $\lambda > 0$ *such that* $\alpha + \lambda p \in \Lambda$ *and* $g(\alpha + \lambda p) < g(\alpha)$.

Proposition 2 *The following is true*:

(a) $\alpha \in \Lambda \setminus \Lambda^*$ *iff* $\omega(\alpha, \nabla g(\alpha)) < 0$,
(b) $\alpha \in \Lambda^*$ *iff* $\omega(\alpha, \nabla g(\alpha)) = 0$.

Thus, we arrive at the following gradient algorithm for finding a minimizing point $\alpha^* \in \Lambda^*$.

Algorithm 1

Step 0: Choose an initial point $\alpha^0 \in \Lambda$ and set $k := 0$.
Step 1: Find $p^k = p^0(\alpha^k, \nabla g(\alpha^k)),\ \omega^k = \omega(\alpha^k, \nabla g(\alpha^k))$.
If $\omega^k = 0$ then $\alpha^* := \alpha^k$ and stop.
Step 2: Find

$$\lambda_{\max} = \max\{\lambda \geq 0 : \alpha^k + \lambda p^k \in \Lambda\} = \min_{p_i^k < 0} \left\{ -\frac{\alpha_i^k}{p_i^k} \right\}$$

($\lambda_{\max} > 0$ because $p^k \in \mathcal{P}(\alpha^k)$).
Step 3: Perform a line search along p^k in order to find

$$\lambda^k = \arg \min_{\lambda \in [0, \lambda_{\max}]} g(\alpha^k + \lambda p^k)$$

($\lambda^k > 0$ and $g(\alpha^k + \lambda^k p^k) < g(\alpha^k)$ due to Propositions 1, 2).
Step 4: Let

$$\alpha^{k+1} := \alpha^k + \lambda^k p^k, \quad k := k + 1$$

and go to Step 1.

Algorithm 1 is a gradient projection type method in its most basic form (compare, for example, with Rosen 1960; Bertsekas 1982; Michelot 1986; Calamai and Moré 1987). However, in the form given, even its convergence is questionable, because Algorithm 1 does not incorporate "anti-zig-zagging-devices". Also, if it would converge, the convergence would be rather slow (as with steepest-descent type methods).

Also, algorithmically important aspects are omitted deliberately: for example, the stopping criterion of Step 2 usually prevents termination after finitely many steps. So in practice one would stop the method in Step 1 as soon as $|\omega^k|$ is "small enough", say if $|\omega^k| \leq \text{eps}\, |\nabla g(\alpha^k)|$ for some small eps > 0. Also, Step 3 requires an exact line search, which can be done for a general (differentiable) convex function g only approximately by means of another iterative method. Exact line search poses no problem for functions g belonging to strictly convex quadratic functions f, and, also for this reason, we will limit the further refinements of Algorithm 1 in section 3 to this class of functions.

One of the purposes of this paper is to give in section 2 a fast procedure for computing $p^0(\alpha, h)$ [see (3)] for all $\alpha \in \Lambda$ and $h \in R^m$ with only $O(m \log m)$ operations. This will be used in section 3 to gain efficient refinements of Algorithm 1 for the special case of strictly convex quadratic functions f. In fact, we will even derive a finite algorithm for this case. Numerical examples are given in the final section 4.

2 A fast procedure for projecting a direction

A recursive polynomial method for finding even the orthogonal projection of a point to the canonical simplex was proposed by Michelot (1986). His method also solves the problem of finding the projection $p^0(\alpha, h)$ defined by (3) for any fixed h.

Nevertheless, we propose a different procedure, which requires less arithmetic operations than the procedure of Michelot (1986), namely only $O(m \log m)$ instead of $O(m^2)$ operations.

The following proposition establishes the uniqueness of the vector $p^0(\alpha, h)$.

Proposition 3 *If $\omega(\alpha, h) < 0$, then $p^0(\alpha, h)$ is unique.*

Proof Assume that there are two minimizing vectors $p^0,\ q^0$ (the arguments α, h are omitted). Then

$$\langle h, p^0 \rangle = \langle h, q^0 \rangle = \omega(\alpha, h) < 0,$$

which implies in particular $|p^0| = |q^0| = 1$. Then $\langle p^0, q^0 \rangle < 1$ since by Schwarz' inequality $|\langle p^0, q^0 \rangle| = 1$ is only possible for $p^0 = \pm q^0$. Obviously,

$$\hat{p} = \frac{\frac{1}{2}p^0 + \frac{1}{2}q^0}{|\frac{1}{2}p^0 + \frac{1}{2}q^0|} \in \mathcal{P}(\alpha)$$

and

$$\langle h, \hat{p} \rangle = \frac{\omega(\alpha, h)}{|\frac{1}{2}p^0 + \frac{1}{2}q^0|} = \frac{\omega(\alpha, h)}{(\frac{1}{2} + \frac{1}{2}\langle p^0, q^0 \rangle)^{\frac{1}{2}}} < \omega(\alpha, h).$$

This contradiction proves Proposition 3. □

Without any loss of generality we may assume that $I^0(\alpha) = \overline{1, r}$, i.e.

$$\alpha = (0, \ldots, 0, \alpha_{r+1}, \ldots, \alpha_m)^T,$$

where $\alpha_i > 0,\ i = r+1, \ldots, m$. For any $p \in \mathcal{P}(\alpha)$, we have $\sum_{i=1}^{m} p_i = 0$ and, therefore, $p_m = -\sum_{i=1}^{m-1} p_i$ (any index $l = r+1, \ldots, m$ could be chosen for such an elimination). Thus, we arrive at the following problem

$$\text{minimize} \ \sum_{i=1}^{m-1} -d_i p_i \qquad (4)$$

$$\text{subject to} \ \ p \in \mathcal{P}',$$

where

$$d_i = h_m - h_i,\ \ i = 1, \ldots, m-1,$$

$$\mathcal{P}' = \left\{ p \in R^{m-1} : \left(\sum_{i=1}^{m-1} p_i \right)^2 + \sum_{i=1}^{m-1} p_i^2 \le 1,\ \ p_i \ge 0,\ \ i = 1, \ldots, r \right\}.$$

Apply the Kuhn–Tucker theorem by writing the Lagrangian of (4) as

$$\mathcal{L}(p, \mu, \mu_0) = \sum_{i=1}^{m-1} -d_i p_i + \mu_0 \left(\left(\sum_{i=1}^{m-1} p_i \right)^2 + \sum_{i=1}^{m-1} p_i^2 - 1 \right) - \sum_{i=1}^{r} \mu_i p_i. \qquad (5)$$

Assume that $\mu_0 = 1/2$. The arguments which justify this assumption will be given later.

From the equation

$$\frac{\partial \mathcal{L}(p, \mu, \mu_0)}{\partial p} = 0,$$

one can easily obtain explicit formulas for p : indeed, one gets

$$p_i = \mu_i + d_i - \sum_{i=1}^{m-1} p_i, \quad i = 1, \dots, r,$$
$$p_i = d_i - \sum_{i=1}^{m-1} p_i, \quad i = r+1, \dots, m-1.$$

By adding these equations we obtain an equation for $-\sum_{i=1}^{m-1} p_i$. Inserting this expression into the last equations we obtain the formulas

$$p_i = d_i - \frac{1}{m}\sum_{j=1}^{m-1} d_j + \mu_i - \frac{1}{m}\sum_{j=1}^{r} \mu_j, \quad i = 1, \dots, r,$$
$$p_i = d_i - \frac{1}{m}\sum_{j=1}^{m-1} d_j - \frac{1}{m}\sum_{j=1}^{r} \mu_j, \quad i = r+1, \dots, m-1.$$

Denote

$$a_i = d_i - \frac{1}{m}\sum_{j=1}^{m-1} d_j$$

and assume without any loss of generality that $a_1 \leq a_2 \leq \cdots \leq a_r$. Then, for finding nonnegative $\mu_1, \mu_2, \dots, \mu_r$ satisfying the Kuhn–Tucker conditions, we have the following system

$$\begin{aligned}
p_1 &= a_1 + \mu_1 - \frac{1}{m}\sum_{j=1}^{r} \mu_j \geq 0, \quad \mu_1 \geq 0,\\
p_2 &= a_2 + \mu_2 - \frac{1}{m}\sum_{j=1}^{r} \mu_j \geq 0, \quad \mu_2 \geq 0,\\
&\dots\\
p_r &= a_r + \mu_r - \frac{1}{m}\sum_{j=1}^{r} \mu_j \geq 0, \quad \mu_r \geq 0,\\
p_{r+1} &= a_{r+1} - \frac{1}{m}\sum_{j=1}^{r} \mu_j,\\
&\dots\\
p_{m-1} &= a_{m-1} - \frac{1}{m}\sum_{j=1}^{r} \mu_j
\end{aligned} \tag{6}$$

with the complementarity conditions

$$\mu_i p_i = 0, \quad i = 1, \ldots, r. \tag{7}$$

It will be shown that solving this system requires $O(m)$ operations. Taking into account that the optimal procedure for ordering a_i requires $O(m \ \log m)$ operations implies that the number of operations for finding the projection is $O(m \ \log m)$.

Denote

$$\begin{aligned} S_0 &:= 0, \\ S_i &:= -\frac{m}{m-i} \sum_{j=1}^{i} a_j, \quad i = 1, \ldots, r. \end{aligned} \tag{8}$$

Let the index ℓ be defined as follows

$$\ell = \begin{cases} 0 & \text{if } a_1 \geq 0, \\ \max\{k \in \overline{1,r} : \forall i \in \overline{1,k} \ \ (a_i - \frac{1}{m} S_{i-1} < 0)\} & \text{if } a_1 < 0. \end{cases} \tag{9}$$

Observe that ℓ can be found in $O(m)$ operations with the use of the following recurrence relation

$$S_i = \frac{m-i+1}{m-i} S_{i-1} - \frac{m}{m-i} a_i, \quad i = 1, \ldots, r.$$

Proposition 4 *System (6) with the complementarity conditions (7) has a unique solution given by the following formulas*

$$\begin{aligned} \mu_i &= -a_i + \tfrac{1}{m} S_\ell, \quad i = 1, \ldots, \ell, \\ \mu_i &= 0, \quad i = \ell+1, \ldots, r, \end{aligned} \tag{10}$$

and

$$\begin{aligned} p_i &= 0, \quad i = 1, \ldots, \ell, \\ p_i &= a_i - \tfrac{1}{m} S_\ell, \quad i = \ell+1, \ldots, m-1. \end{aligned} \tag{11}$$

Proof By substituting (10) in (6) one can prove that (11) holds. From the definition of ℓ, it follows that $p_i \geq 0, \ \ i = \ell+1, \ldots, r$. Obviously, (7) holds for p's and μ's given by (10) and (11). Thus, it remains to prove that $\mu_i > 0, \ \ i = 1, \ldots, \ell$. In fact, from the definition of ℓ, we have for $\ell \geq 1$

$$a_\ell - \frac{1}{m} S_{\ell-1} < 0. \tag{12}$$

Relations (8) imply

$$a_\ell = -\frac{m-\ell}{m} S_\ell + \frac{m-\ell+1}{m} S_{\ell-1},$$

which with (12) yields

$$-\frac{m-\ell}{m} S_\ell + \frac{m-\ell+1}{m} S_{\ell-1} - \frac{1}{m} S_{\ell-1} < 0,$$

This gives

$$S_\ell > S_{\ell-1}.$$

The last inequality and (12) imply

$$a_\ell - \frac{1}{m} S_\ell < 0. \tag{13}$$

Since $a_1 \le a_2 \le \cdots \le a_\ell$, we have from (13)

$$a_i - \frac{1}{m} S_\ell < 0, \quad i = 1, \ldots, \ell.$$

Taking into account (10) implies that $\mu_i > 0, \ \ i = 1, \ldots, \ell$.

Proof that the combined system (6) and (7) has only one solution [which is given by (10)]. Assume there is another solution $\bar{\mu}_1, \bar{\mu}_2, \ldots, \bar{\mu}_r$. If $\bar{\mu}_i = 0, i = 1, \ldots, r$, then deduce from (6) that $a_1 \ge 0$ and, therefore, $\ell = 0$ by (9). Hence by (10), $\mu_i = 0, i = \overline{1, \ldots, r}$, i.e. $\mu_i = \bar{\mu}_i, i = 1, \ldots, r$. Suppose now that $\bar{\mu}_j > 0$ for some $j \in \overline{1, r}$. Set

$$\ell' = \max\{i \in \overline{1, r} : \bar{\mu}_i > 0\}.$$

Obviously, $\bar{\mu}_{\ell'} > 0$ and we have

$$
\begin{aligned}
&\bar{p}_1 = a_1 + \bar{\mu}_1 - \frac{1}{m} \sum_{j=1}^{\ell'} \bar{\mu}_j \ge 0,\\
&\bar{p}_2 = a_2 + \bar{\mu}_2 - \frac{1}{m} \sum_{j=1}^{\ell'} \bar{\mu}_j \ge 0,\\
&\cdots\\
&\bar{p}_{\ell'-1} = a_{\ell'-1} + \bar{\mu}_{\ell'-1} - \frac{1}{m} \sum_{j=1}^{\ell'} \bar{\mu}_j \ge 0,\\
&\bar{p}_{\ell'} = a_{\ell'} + \bar{\mu}_{\ell'} - \frac{1}{m} \sum_{j=1}^{\ell'} \bar{\mu}_j = 0,\\
&\bar{p}_{\ell'+1} = a_{\ell'+1} - \frac{1}{m} \sum_{j=1}^{\ell'} \bar{\mu}_j \ge 0,\\
&\cdots\\
&\bar{p}_r = a_r - \frac{1}{m} \sum_{j=1}^{\ell'} \bar{\mu}_j \ge 0
\end{aligned}
\tag{14}
$$

due to (6) and (7). Since $a_1 \le a_2 \le \cdots \le a_{\ell'}$ the first $\ell' - 1$ inequalities of (14) imply together with (7)

$$\bar{\mu}_i \ge \bar{\mu}_{\ell'} > 0, \quad i = 1, \ldots, \ell' - 1. \tag{15}$$

Therefore, from (7) we have

$$\begin{aligned}
\bar{p}_1 &= a_1 + \bar{\mu}_1 - \frac{1}{m}\sum_{j=1}^{\ell'} \bar{\mu}_j = 0,\\
\bar{p}_2 &= a_2 + \bar{\mu}_2 - \frac{1}{m}\sum_{j=1}^{\ell'} \bar{\mu}_j = 0,\\
&\dots\\
\bar{p}_{\ell'-1} &= a_{\ell'-1} + \bar{\mu}_{\ell'-1} - \frac{1}{m}\sum_{j=1}^{\ell'} \bar{\mu}_j = 0,\\
\bar{p}_{\ell'} &= a_{\ell'} + \bar{\mu}_{\ell'} - \frac{1}{m}\sum_{j=1}^{\ell'} \bar{\mu}_j = 0,\\
\bar{p}_{\ell'+1} &= a_{\ell'+1} - \frac{1}{m}\sum_{j=1}^{\ell'} \bar{\mu}_j \geq 0,\\
&\dots\\
\bar{p}_r &= a_r - \frac{1}{m}\sum_{j=1}^{\ell'} \bar{\mu}_j \geq 0.
\end{aligned} \tag{16}$$

Taking the sum of the first ℓ' equalities of (16), one can easily obtain that

$$\sum_{j=1}^{\ell'} \bar{\mu}_j = S_{\ell'}$$

and

$$\bar{\mu}_i = -a_i + \frac{1}{m}S_{\ell'}, \quad i = 1, \dots, \ell' \tag{17}$$

Thus, if $\ell' = \ell$ then $\bar{\mu}_i = \mu_i, \; i = 1, \dots, r$. If $\ell' < \ell$, then (16) and the definition of ℓ yield

$$\bar{p}_{\ell'+1} = a_{\ell'+1} - \frac{1}{m}S_{\ell'} < 0,$$

which is a contradiction. If $\ell' > \ell$, then the definition of ℓ and monotonicity of a_i in i imply

$$\begin{aligned}
a_{\ell+1} - \frac{1}{m}S_\ell &\geq 0\\
a_{\ell+2} - \frac{1}{m}S_\ell &\geq 0\\
&\dots\\
a_{\ell'} - \frac{1}{m}S_\ell &\geq 0.
\end{aligned} \tag{18}$$

Taking the sum of these inequalities yields

$$\sum_{i=\ell+1}^{\ell'} a_i - \frac{\ell' - \ell}{m} S_\ell \geq 0,$$

which can be represented using the definition of S_ℓ as

$$\sum_{i=\ell+1}^{\ell'} a_i + \frac{\ell' - \ell}{m - \ell} \sum_{i=1}^{\ell} a_i \geq 0. \tag{19}$$

On the other hand, we have from (17)

$$\bar{\mu}_{\ell+1} = -a_{\ell+1} + \frac{1}{m} S_{\ell'} = -\left(a_{\ell+1} - \frac{1}{m} S_\ell\right) - \left(\frac{1}{m} S_\ell - \frac{1}{m} S_{\ell'}\right). \tag{20}$$

By the definition of ℓ,

$$a_{\ell+1} - \frac{1}{m} S_\ell \geq 0. \tag{21}$$

From (8) and (19),

$$\begin{aligned}
\frac{1}{m} S_\ell - \frac{1}{m} S_{\ell'} &= -\frac{1}{m-\ell} \sum_{i=1}^{\ell} a_i + \frac{1}{m-\ell'} \sum_{i=1}^{\ell'} a_i \\
&= -\frac{1}{m-\ell} \sum_{i=1}^{\ell} a_i + \frac{1}{m-\ell'} \sum_{i=1}^{\ell} a_i + \frac{1}{m-\ell'} \sum_{i=\ell+1}^{\ell'} a_i \\
&= \frac{\ell' - \ell}{(m-\ell')(m-\ell)} \sum_{i=1}^{\ell} a_i + \frac{1}{m-\ell'} \sum_{i=\ell+1}^{\ell'} a_i \\
&= \frac{1}{m-\ell'} \left(\sum_{i=\ell+1}^{\ell'} a_i + \frac{\ell' - \ell}{m-\ell} \sum_{i=1}^{\ell} a_i \right) \geq 0.
\end{aligned} \tag{22}$$

Using (20)–(22), we obtain that $\bar{\mu}_{\ell+1} \leq 0$, which is a contradiction with (15). Thus, Proposition 4 is proved. □

Let us come back to the assumption $\mu_0 = 1/2$. If (11) gives a nonzero vector $p \in R^{m-1}$, then the vector $\hat{p} = p/\xi$, where $\xi > 0$ and $\xi^2 = \left(\sum_{i=1}^{m-1} p_i\right)^2 + \sum_{i=1}^{m-1} p_i^2$, satisfies all conditions of the Kuhn–Tucker theorem with $\mu_0 = \xi/2$ and μ_i given by (10). Hence $\hat{p}$ is a minimizing vector of (4).

Consider the case where (11) yields $p = 0$. If $\hat{p}$ is a minimizing vector of (4), which satisfies the conditions of the Kuhn–Tucker theorem with $\mu_0 > 0$, then the vector $2\mu_0 \hat{p}$ satisfies the system (6) and the complementarity conditions (7). Because of the uniqueness of the solution of (6),(7) we conclude that $\hat{p} = 0$, which is a contradiction with $\mu_0 > 0$. Therefore, $\hat{p}$ satisfies the conditions of the Kuhn - Tucker theorem with $\mu_0 = 0$. In this case one can easily obtain from (5) that $d_i = 0, i = r+1, \ldots, m-1$ (because $\mathcal{L}(\cdot, \mu, \mu_0)$ attains its minimum at $\hat{p}$), $d_i \leq 0, i = 1, \ldots, r$ (because there should be $d_i + \mu_i = 0, \mu_i \geq 0$) and $d_i = 0$ whenever $\hat{p}_i > 0, i = 1, \ldots, r$ (complementarity conditions). This implies $\sum_{i=1}^{m-1} -d_i \hat{p}_i = 0$. On the other hand, the vector $p = 0$ obtained using (11), i.e. under the assumption $\mu_0 = 1/2$, gives the same value. Note that we do not claim the uniqueness of μ_0 but prove only that the value $1/2$ is appropriate.

Remark 1 It is easily seen that $a_1 \leq a_2 \leq \cdots \leq a_r$ iff $h_1 \geq h_2 \geq \cdots \geq h_r$. By definition, the a_i and S_{i-1} are particular linear functions of the $d_j = h_m - h_j$. By rearranging terms one verifies for $i \leq r$

$$a_i - \frac{1}{m} S_{i-1} = \frac{m-i}{m-i+1} d_i - \frac{1}{m-i+1} \sum_{j=i+1}^{m-1} d_j$$
$$= \frac{1}{m-i+1} \sum_{j=i}^{m} h_j - h_i.$$

Hence,

$$\ell = \begin{cases} 0 \quad \text{if } a_1 = \frac{1}{m} \sum_{j=1}^{m} h_j - h_1 \geq 0, \\ \max\{k \in \overline{1, r} : \forall i \in \overline{1, k} \ \left(\frac{1}{m-i+1} \sum_{j=i}^{m} h_j - h_i < 0\right)\} \quad \text{if } a_1 < 0. \end{cases}$$

For $i \geq \ell + 1$ one finds in the same way

$$a_i - \frac{1}{m} S_\ell = \frac{m-\ell-1}{m-\ell} d_i - \frac{1}{m-\ell} \sum_{j=\ell+1,\, j \neq i}^{m-1} d_j$$
$$= \frac{1}{m-\ell} \sum_{j=\ell+1}^{m} h_j - h_i.$$

Thus by (11),

$$p_i = \frac{1}{m-\ell} \sum_{j=\ell+1}^{m} h_j - h_i, \quad i = \ell+1, \ldots, m-1,$$
$$p_m \stackrel{def}{=} -\sum_{j=1}^{m-1} p_j = \frac{1}{m-\ell} \sum_{j=\ell+1}^{m} h_j - h_m.$$

One can also check that

$$\omega(\alpha, h) = -\left(\sum_{i=\ell+1}^{m} h_i^2 - \frac{1}{m-\ell} \left(\sum_{i=\ell+1}^{m} h_i \right)^2 \right)^{\frac{1}{2}}.$$

Therefore, the procedure of finding $p^0(\alpha, h)$ and $\omega(\alpha, h)$ is formulated in terms of the components of h (=$\nabla g(\alpha)$), which is important when implementing the procedure on a computer.

3 Modifications of Algorithm 1

The following algorithm is again given in idealized form. It uses the idea of Bertsekas Bertsekas (1982) of computing for a given $\alpha^k \in \Lambda$ the next iterate $\alpha^{k+1} \in \Lambda$ by minimizing $g(\alpha^k(\lambda))$ along a curvilinear path $\alpha^k(\lambda)$, $\lambda \geq 0$ defined by

$$\alpha^k(\lambda) := P_\Lambda(\alpha^k - \lambda \nabla g(\alpha^k)),$$

where $P_\Lambda : R^m \to \Lambda$ is the orthogonal projection of R^m to Λ.

Algorithm 2

Step 0: Choose an initial point $\alpha^0 \in \Lambda$ and set $k := 0$.
Step 1: Update the gradient $h^k = \nabla g(\alpha^k)$, set $\alpha^{k,0} := \alpha^k$ and $l := 0$.
Step 2: Find $p^{k,l} = p^0(\alpha^{k,l}, h^k)$, $\omega^{k,l} = \omega(\alpha^{k,l}, h^k)$.
If $\omega^{k,l} = 0$ and $l = 0$, then $\alpha^* := \alpha^k$ and stop.
Step 3: Find

$$\lambda_{\max} = \max\{\lambda \geq 0 : \alpha^{k,l} + \lambda p^{k,l} \in \Lambda\} = \min_{p_i^{k,l} < 0} \left\{ -\frac{\alpha_i^{k,l}}{p_i^{k,l}} \right\}$$

($\lambda_{\max} > 0$ because $p^{k,l} \in \mathcal{P}(\alpha^{k,l})$).
Step 4: Make a line search along $p^{k,l}$ in order to find an optimal solution λ' of $\min\{g(\alpha^{k,l} + \lambda p^{k,l}) \mid \lambda \in [0, \lambda_{\max}]\}$ and set

$$\alpha' := \alpha^{k,l+1} := \alpha^{k,l} + \lambda' p^{k,l}.$$

Step 5: If $\lambda' < \lambda_{\max}$, then set $\alpha^{k+1} := \alpha'$, $k := k + 1$ and go to Step 1.
Otherwise, set $\alpha^{k,l+1} := \alpha'$, $l := l + 1$ and go to Step 2.

For strictly convex quadratic functions f, Steps 4 and 5 can be realized as follows: note that in this case, the function $\varphi(\lambda) := g(\alpha^{k,l} + \lambda p^{k,l})$, $\lambda \in R$, which has to be minimized in Step 4 is a (not necessarily strictly) convex quadratic function of $\lambda \in R$ which is bounded below, so that $\varphi'(\lambda)$ is a linear (perhaps identically zero) function. Hence Steps 4 and 5 are equivalent to:

Step 4′: Compute

$$\alpha_{\max} := \alpha^{k,l} + \lambda_{\max} p^{k,l}$$

$$x_{\max} := Z\alpha_{\max}, \quad q := Zp^{k,l}, \quad \nabla f(x_{\max}),$$

$$\varphi'(\lambda_{\max}) = (\nabla f(x_{\max}))^T q.$$

Step 5′: If $\varphi'(\lambda_{\max}) \leq 0$ set $\alpha^{k,l+1} := \alpha_{\max}$, $l := l + 1$ and go to Step 2.
Otherwise, compute $\varphi'(0) = (\nabla f(Z\alpha^{k,l}))^T q$ and

$$\lambda' := \begin{cases} 0, & \text{if } \varphi'(0) \geq 0 \\ \lambda_{\max} \frac{\varphi'(0)}{\varphi'(0) - \varphi'(\lambda_{\max})}, & \text{if } \varphi'(0) < 0, \end{cases}$$

$$\alpha^{k+1} := \alpha^{k,l} + \lambda' p^{k,l},$$

set $k := k + 1$ and go to Step 1.

Note that $Z\alpha^{k,l}$ and $\nabla f(Z\alpha^{k,l})$ need not be computed in Step 5′: they were already computed in the previous Step $l-1 \to l$.

The convergence of this type of algorithm has been been studied by Calamay and Moré Calamai and Moré (1987). The speed of convergence is still slow.

From now on, we will assume that f is a strictly convex quadratic function. Then $g(\alpha) = f(Z\alpha)$ is a convex quadratic function which is bounded below on R^m.

For the next modification, we use the idea to observe the linear manifold in which the process is running and to minimize $g(\cdot)$ over this manifold. Thus, we need an effective procedure for the unrestricted minimization of the objective functions over manifolds of relatively small dimension.

For arbitrary $I \subset \overline{1,m}$, we consider linear manifolds $L(I)$ defined by

$$L(I) := \{\alpha \in R^m : \alpha_i = 0 \text{ for } i \in I,\ \sum_{i=1}^{m} \alpha_i = 1\}.$$

The minimization of quadratic functions $g(\alpha)$ over $L(I)$ presents no problems: it has a solution (since g is bounded below), which can be computed by solving linear equations.

Algorithm 3

Step 0: Choose an initial point $\alpha^0 \in \Lambda$ and set $k := 0$.
Step 1: Update the gradient $h^k = \nabla g(\alpha^k)$, set $\alpha^{k,0} := \alpha^k$ and $l := 0$.
Step 2: Find $p^{k,l} = p^0(\alpha^{k,l}, h^k)$, $\omega^{k,l} = \omega(\alpha^{k,l}, h^k)$.
If $\omega^{k,l} = 0$ and $l = 0$, then $\alpha^* := \alpha^k$ and stop.
Step 3: Find

$$\lambda_{\max} = \max\{\lambda \geq 0 : \alpha^{k,l} + \lambda p^{k,l} \in \Lambda\} = \min_{p_i^{k,l}<0} \left\{-\frac{\alpha_i^{k,l}}{p_i^{k,l}}\right\}$$

($\lambda_{\max} > 0$ because $p^{k,l} \in \mathcal{P}(\alpha^{k,l})$).
Step 4: Perform a line search along $p^{k,l}$ in order to find

$$\lambda' = \arg\min_{\lambda \in [0,\lambda_{\max}]} g(\alpha^{k,l} + \lambda p^{k,l}).$$

Step 5: Let

$$\alpha^{k,l+1} := \alpha^{k,l} + \lambda' p^{k,l}.$$

If $|I^0(\alpha^{k,l+1})| > |I^0(\alpha^{k,l})|$, set $l := l+1$ and go to Step 2.
Step 6: Let $I' := I^0(\alpha^{k,l+1})$, find

$$\alpha' = \arg\min_{\alpha \in L(I')} g(\alpha),$$

set

$$\begin{aligned} \rho' &:= \max\{\rho \leq 1 : \alpha^{k,l+1} + \rho(\alpha' - \alpha^{k,l+1}) \in \Lambda\}, \\ \alpha^{k+1} &:= \alpha^{k,l+1} + \rho'(\alpha' - \alpha^{k,l+1}), \\ k &:= k+1, \end{aligned}$$

and go to Step 1.

The exact line search in Step 4 can be realized as in Algorithm 2, since f is a strictly convex quadratic function.

The next variant is related to Algorithm 3 but uses a recursive call to update the minimum of $g(\cdot)$ over proper faces $\Lambda(I)$ of Λ of the form $\Lambda(I) := L(I) \cap \Lambda$ with $I \neq \emptyset$. Also, if Λ' is any face of the simplex Λ, we use the shorthand notation $P(\Lambda')$ for the problem

$$\min \{g(\alpha) \mid \alpha \in \Lambda'\}$$

of minimizing g over Λ'.

Algorithm 4 Given Λ and an initial point $\tilde{\alpha} \in \Lambda$.

Step 0: Set $k := 0$ and $\alpha^0 := \tilde{\alpha}$.
Step 1: Update the gradient $h^k = \nabla g(\alpha^k)$, set $\alpha^{k,0} := \alpha^k$ and $l := 0$.
Step 2: Find $p^{k,l} = p^0(\alpha^{k,l}, h^k)$, $\omega^{k,l} = \omega(\alpha^{k,l}, h^k)$.
If $|\omega^{k,0}| = 0$ then return α^k: α^k is an optimal solution of $P(\Lambda)$.
Step 3: Find

$$\lambda_{\max} = \max\{\lambda \geq 0 : \alpha^{k,l} + \lambda p^{k,l} \in \Lambda\} = \min_{p_i^{k,l}<0} \left\{ -\frac{\alpha_i^{k,l}}{p_i^{k,l}} \right\}$$

($\lambda_{\max} > 0$ because $p^{k,l} \in \mathcal{P}(\alpha^{k,l})$).
Step 4: Perform a line search along $p^{k,l}$ in order to find

$$\lambda' = \arg \min_{\lambda \in [0, \lambda_{\max}]} g(\alpha^{k,l} + \lambda p^{k,l}).$$

Step 5: Let

$$\alpha^{k,l+1} := \alpha^{k,l} + \lambda' p^{k,l}.$$

If $|I^0(\alpha^{k,l+1})| > |I^0(\alpha^{k,l})|$, set $l := l + 1$ and go to Step 2.
Step 6: Let $I' := I^0(\alpha^{k,l+1})$.
If $I' \neq \emptyset$ then set $\hat{\alpha} := \alpha^{k,l+1}$ and go to Step 7.
Otherwise find $\alpha' = \arg \min_{\alpha \in L(I')} g(\alpha)$, set

$$\begin{aligned}
\rho' &:= \max\{\rho \leq 1 \mid \alpha^{k,l+1} + \rho(\alpha' - \alpha^{k,l+1}) \in \Lambda\}, \\
\hat{\alpha} &:= \alpha^{k,l+1} + \rho'(\alpha' - \alpha^{k,l+1}), \\
I' &:= I^0(\hat{\alpha}).
\end{aligned}$$

If $I' = \emptyset$ or $\rho' = 1$ then return $\hat{\alpha}$: $\hat{\alpha}$ is an optimal solution of problem $P(\Lambda)$ at hand.
Step 7: Find an optimal solution α^{k+1} of $P(\Lambda')$ by calling Algorithm 4 recursively with the initial data $\Lambda' := L(I') \cap \Lambda$ and $\hat{\alpha} \in \Lambda'$.
Set $k := k + 1$ and goto Step 1.

Note that Λ' is a proper subset of Λ in Step 7, since $I' \neq \emptyset$. Also in this algorithm, the exact line search in Step 4 can be realized as in Algorithm 2.

We show that Algorithm 4 is finite:

Theorem *Algorithm 4 terminates after finitely steps with an optimal solution α^* of* $\min\{g(\alpha) \mid \alpha \in \Lambda\}$ *and the optimal solution of $x^* = Z\alpha^*$ of* $\min\{f(x) \mid x \in co\, \mathcal{Z}\}$.

Proof In this proof, which is by induction on $m := 1 + \dim \Lambda$, we call $k \to k+1$ a major iteration, which consists of several several minor iterations $l \to l+1$.

Clearly, the algorithm terminates for $m = 1$ within Step 2 for $k = 0$, since then $\Lambda = \{1\}$ is a singleton and $g \mid \Lambda$ is a constant function, so that $\omega^{0,0} = 0$.

Suppose now that Algorithm 4 terminates for all simplizes Λ' of dimension $k \leq m-2$ and starting values $\tilde{\alpha} \in \Lambda'$, and assume $1 + \dim \Lambda = m$ and $\tilde{\alpha} \in \Lambda$. Then the first major iteration $k \to k+1$ with $k = 0$ of the algorithm applied to Λ and the starting point $\alpha^0 := \tilde{\alpha}$ either stops already at Step 2 with an optimal solution $\alpha^* \in \Lambda$ of $P(\Lambda)$, or the vector $p^{0,0} = \nabla g(\alpha^{0,0}) \neq 0$ is nonzero, so that by Steps 3–5 of the algorithm

$$g(\alpha^{0,l+1}) \leq g(\alpha^{0,l}) \leq \cdots \leq g(\alpha^{0,1}) < g(\alpha^{0,0}) = g(\alpha^0),$$

$$|I^0(\alpha^{0,l})| > |I^0(\alpha^{0,l-1})| > \cdots > |I^0(\alpha^{0,0})|.$$

Since $|I^0(\alpha)| \leq m$ for all $\alpha \in \Lambda$, the algorithm enters after finitely many minor iterations $l \to l+1$ Step 6 with a point $\hat{\alpha} = \alpha^{0,l+1} \in \Lambda$ and an index set $I' = I^0(\hat{\alpha})$ with $\hat{\alpha} \in \Lambda(I')$.

If $I' \neq \emptyset$ then $\dim \Lambda(I') < \dim \Lambda$ so that by induction hypothesis a minimizer α^1 of $P(\Lambda(I'))$ satisfying $g(\alpha^1) < g(\tilde{\alpha})$ is found in Step 7.

If $I' = I^0(\hat{\alpha}) = \emptyset$ then it is easily seen that Step 6 of the algorithm either finds an optimal solution $\hat{\alpha}$ of $P(\Lambda)$ or a point $\hat{\alpha} \in \Lambda$ with $I' = I^0(\hat{\alpha}) \neq \emptyset$ and $g(\hat{\alpha}) < g(\tilde{\alpha})$ so that again $\hat{\alpha} \in \Lambda' := \Lambda(I')$, $\dim \Lambda' < \dim \Lambda$. So again by induction, Step 7 will find a point $\alpha^1 \in \Lambda$ with $g(\alpha^1) < g(\tilde{\alpha})$, where α^1 is an optimal solution of $P(\Lambda'_0)$ in the proper face $\Lambda'_0 := \Lambda'$ of Λ.

Now consider the next major iteration $k \to k+1$ of Algorithm 4, where $k = 1$. If the algorithm does not terminate in Step 2, because α^1 is an optimal solution of $P(\Lambda)$, then $g(\alpha^{1,1}) < g(\alpha^1)$. By what we have shown for the previous major iteration with $k = 0$, the algorithm will enter Step 7 with a point $\hat{\alpha} \in \Lambda'_1$, where $\Lambda'_1 := \Lambda(I^0(\hat{\alpha}))$, $\dim \Lambda'_1 < \dim \Lambda = m - 1$, and $g(\hat{\alpha}) < g(\alpha^1)$. Since α^1 was an optimal solution of $P(\Lambda'_0)$ in the proper face Λ'_0 of Λ, the face Λ'_1 must be different from the face Λ'_0 which was used in the previous major iteration with $k = 0$.

So each major iteration $k \to k+1$ of Algorithm 4 for solving $P(\Lambda)$, if it does not stop in Step 2 with an optimal solution α^* of $P(\Lambda)$, will generate an iterate $\alpha^{k+1} \in \Lambda'_{k+1} := \Lambda(I^0(\alpha^{k+1}))$, where α^{k+1} is an optimal solution of $P(\Lambda'_{k+1})$ satisfying

$$\dim \Lambda'_{k+1} < \dim \Lambda, \quad g(\alpha^{k+1}) < g(\alpha^k).$$

Therefore the proper faces Λ'_k, $k = 0, 1, \ldots$, of Λ are all different, so they cannot repeat. Since the number of proper faces of Λ is finite, Algorithm 4 must stop after finitely many major iterations with an optimal solution of $P(\Lambda)$. □

4 Numerical examples

We begin with some general remarks on the complexity of the algorithms. Given $\alpha \in R^m$, computing $f(Z\alpha)$ and the gradient

$$\nabla g(\alpha) = Z^T \nabla f(Z\alpha)$$

needs one evaluation of $Z\alpha$, which requires at most $O(mn)$ operations, one evaluation of f and of ∇f (for convex quadratic f this requires at most $O(n^2)$ operations), and further $O(mn)$ operations to compute a matrix-vector product $Z^T y$. For given $h = \nabla g(\alpha)$, the computation of $p^0(\alpha, h)$ requires $O(m \log m)$ operations.

Concerning the numerical realization of the Algorithms we note that, for given $\alpha := \alpha^k \in \Lambda$, the evaluation of $Z\alpha$ required in the computation of $g(\alpha) = f(Z\alpha)$ and of the gradient

$$\nabla g(\alpha) = Z^T \nabla f(Z\alpha)$$

should be done judiciously: except for the initial iterations of the algorithms, the set $I^0(\alpha) := \{i \mid \alpha_i = 0\}$ of zero components of α is so large, that its complementary set $\{i \mid \alpha_i > 0\}$ has only about $O(n)$ components. So, it pays to compute $Z\alpha$ as a reduced sum

$$Z\alpha = \sum_{i:\alpha_i>0} \alpha_i z^i,$$

requiring about $O(n^2)$ operations. Similarly, any line search for solving

$$\min\{g(\alpha + \lambda p) \mid \lambda \in [0, \lambda_{\max}]\}$$

in Steps 3 resp. 4 of the algorithms normally requires the computation of

$$\nabla g(\alpha)^T p = \nabla f(Z\alpha)^T Zp.$$

Since also the vectors p occuring in the algorithms have many zero components (usually also p has only $O(n)$ nonzero components) it pays to compute first Zp also by a reduced sum

$$Zp = \sum_{i:\, p_i \neq 0} p_i z^i$$

and then

$$\nabla g(\alpha)^T p = \big(\nabla f(Z\alpha)\big)^T \big(Zp\big).$$

In Algorithms 3 and 4, one has to compute (occasionally: only once in each major iteration $k \to k+1$)

$$\min\{g(\alpha) \mid \alpha \in L(I')\},$$

that is, for quadratic f, the solution of linear equations with $m - |I'|$ unknowns. Since again, except for the initial iterations, $m - |I'| = O(n)$, these linear equations can be (asymptotically) solved by $O(n^3)$ operations.

For the recursive Algorithm 4, most iterations deal with the solution of small problems

$$\min\,\{g(\alpha) \mid \alpha \in \Lambda'\},$$

where $\dim \Lambda' = O(n)$. So most iterations of that algorithm require only $O(n^3)$ operations for quadratic f.

So all in all, the basic Algorithms 1 and 2 need (for quadratic f) about $O(nm) + O(m \log m) + O(n^2)$ operations per iteration. So, the efficiency of the projection method of section 2 will matter for these algorithms (and similarly for Algorithms 3, 4) only for problems with $n = O(\log m)$. Problems of this kind occur in computational geometry when dealing with polytopes in a low-dimensional space which are given as convex hull of many points.

Let us consider some numerical examples characterizing the comparative speed of the algorithms proposed for simple strictly convex quadratic f. These examples were implemented with a QuickBASIC program on a PC (Intel P-3 , 1 GHz). The effective speed is approximately equal to 5.3 MFLOPS when using QuickBASIC 4.5.

In all examples the algorithms were stopped, as soon as $|\omega^k| \leq 10^{-16}$ in Step 1 of Algorithm 1, resp. $|\omega^{k,l}| \leq 10^{-16}$ in Step 2 of Algorithms 2, 3, and 4.

Example 1 $\mathcal{Z} := \{z = (\delta_1, \delta_2, \dots, \delta_n)^T : \delta_j = \pm 1 \text{ for all } j\}$ (vertices of the unit cube in R^n). The number of vertices is $m = 2^n$. The objective function is $f(x) = |x - z^c|^2$, where $z^c = (10, 0.7, 0, \dots, 0)^T$. Thus $g(\alpha) = |\sum_{j=1}^m \alpha_j (z^j - z^c)|^2$. The programs realizing Algorithms 1–4 were run with the initial point $\alpha^0 = (1, 0, \dots, 0)^T$. For Algorithm 4, the number of iterations includes all recursive calls. In all tests, Algorithm 4 wins, because most of its iterations are running in a space of low dimension.

The results obtained for dimensions $n = 7$ and $n = 10$ are presented in the following table.

Algorithm	Iterations		Time [s]	
	n=7	n=10	n=7	n=10
1	28	126	1.38	425.9
2	53	72	1.10	113.5
3	16	53	0.38	39.83
4	13	19	0.06	2.25

Example 2 $\mathcal{Z} := \{z^i : z^i = (\xi_{i,1}, \dots, \xi_{i,n})^T,\ i = 1, \dots, m\}$, where $\xi_{i,j}$ are random values uniformly distributed on the interval $[-1, 1]$. The objective function and initial point α^0 were the same as in Example 1 except for $z^c = (10, 0, \dots, 0)^T$. The results obtained for Algorithms 1–4 are presented in the following tables.

For a first series of tests, the initial point $\alpha^0 = (1, 0, \dots, 0)^T$ was used.

n=20	Iterations					
Alg.	m=100	m=200	m=300	m=500	m=1,000	m=2,000
1	74	83	72	190	272	424
2	79	90	91	205	298	566
3	13	23	34	60	88	147
4	25	39	39	82	94	143

n=20			Time [s]			
Alg.	m=100	m=200	m=300	m=500	m=1,000	m=2,000
1	1.59	5.26	9.23	57.73	240.60	1382.01
2	1.54	4.35	6.91	44.26	189.66	1228.18
3	0.16	0.77	1.58	7.03	22.58	84.09
4	0.05	0.28	0.44	1.81	3.73	17.57

For the improved initial point $\alpha^0 := (0, 0, \ldots, 1_k \ldots, 0)^T$, where

$$k := \text{argmin}\{f(z^j) : j = 1, \ldots, m\},$$

the following results were obtained:

n=20			Iterations			
Alg.	m=100	m=200	m=300	m=500	m=1000	m=2000
1	66	63	56	151	224	486
2	74	72	91	163	219	522
3	9	3	19	28	27	68
4	21	16	53	45	63	90

n=20			Time [s]			
Alg.	m=100	m=200	m=300	m=500	m=1000	m=2000
1	1.32	3.46	5.66	40.42	174.77	1289.75
2	1.41	3.95	7.18	36.46	149.55	1221.54
3	0.11	0.22	1.48	4.07	13.01	68.60
4	0.10	0.22	0.54	1.53	4.56	13.78

Large problems. We now list some results for large scale problems, but only for Algorithm 4. We used again the function

$$f(x) = (x - x^c)^T(x - x^c),$$

where now x^c is chosen as the barycenter of co $\mathcal{Z}$, so that the optimal solution of (2) is not unique. As starting point we used $\alpha^0 = e^k \in R^m$, the k-th unit vector in R^m, where $f(z^k) = \min\{f(z^i) \mid i = 1, \ldots, m\}$. The results were obtained on a faster computer with an effective speed of about 70 MFLOPS/s.

For Example 1, the unit cube with $n \geq 10$ and $m = 2^n$, the following results were obtained:

n	$m = 2^n$	Iterations	Time [s]
10	1,024	19	0.07
11	2,048	21	0.26
12	4,096	23	1.21
13	8,192	25	4.77
14	16,384	27	21.43
15	32,786	29	170.25

For randomly generated problems of the type of Example 2, we illustrate the behaviour for large values of n resp. m. The following table gives the results for a fixed $n = 20$ and a growing number m of points in $\mathcal{Z}$:

m	Iterations	Time [s]	m	Iterations	Time [s]
1,000	134	0.2	10,000	238	4.09
2,000	152	0.43	15,000	228	8.48
3,000	147	0.54	20,000	264	14.67
4,000	150	0.84	30,000	260	45.07
5,000	142	1.20	35,000	431	98.24
6,000	131	1.67	40,000	267	177.40
7,000	161	2.14	41,000	305	188
8,000	165	2.87	60,000	311	398.5
9,000	192	3.97	80,000	246	733.87

One notices that the number of iterations is fairly constant, whereas the computing time grows faster than polynomially but not exponentially with m. This is remarkable, since the number of faces of the simplex Λ grows exponentially with its dimension $m - 1$.

Results for a fixed number m of points, $m = 5,000$, and a growing number of n are given in the follwing table:

n	Iterations	Time [s]	n	Iterations	Time [s]
5	143	0.95	90	332	14.21
10	110	1.0	95	443	19.87
15	107	1.07	100	786	46.31
20	142	1.20	101	712	38.11
30	304	1.93	110	784	52.12
50	335	4.0	120	813	73.31
70	363	8.4	150	834	140.48
80	398	12.42	200	1552	871.62

Here, the computing times seem to grow like n^3.

Acknowledgements The author was supported by the Alexander von Humboldt-Foundation, Germany.

References

Bertsekas DP (1982) Projected Newton methods for optimization problems with simple constraints. Siam J Control Optim 20:221–246

Calamai PH, Moré JJ (1987) Projectedgradients methodsfor linearly constrained problems. Math Programming 39:93–116

Michelot C (1986) A finite algorithm for finding the projection of a point onto the canonical simplex of R^n. JOTA 50(1):195–200

Rosen JB (1960) The gradient projection method of nonlinear programming, part 1, linear constraints. SIAM J Appl Math 8:181–217

Rosen JB (1961) The gradient projection method of nonlinear programming, part 2, nonlinear constraints. SIAM J Appl Math 9:514–532