Parametric Linear and Quadratic Optimization by Elimination

Volker Weispfenning
Universität Passau, D-94030 Passau, Germany
Tel: 0851-509317, Fax: 0851-509171
e-mail: weispfen@alice.fmi.uni-passau.de

Jan. 1994

MIP-9404

Abstract. We propose a new elimination method for linear and quadratic optimization involving parametric coefficients. In comparison to the classical Fourier-Motzkin method that is of doubly exponential worst-case complexity our method is singly exponential in the worst case. Moreover it applies also to the minimization of a quadratic objective functions without convexity hypothesis under linear constraints, and to objective functions with arbitrary parametric coefficients. For problems with additive parameters the method is worst-case optimal. Examples computed in a REDUCE–implementation confirm the superiority of the method over Fourier-Motzkin and its applicability to problems of interesting size.

1 Introduction

In 1826 Fourier found a method for solving systems of linear inequalities by successive elimination of variables (see [Fourier]). The method may be regarded as an extension of Gauss elimination for systems of linear equations. It was forgotten until rediscovered in this century by Dines and Motzkin (see [Dantzig, Dines, Motzkin]). One feature that stimulated the interest beyond the realm of pure mathematics was the fact that the method can be used to solve linear optimization problems. Moreover – by the very nature of an elimination method – it admits additive parameters corresponding geometrically to arbitrary translations of the linear constraints.

The major drawback of the method for practical purposes is the rapid increase of the number of constraints that happens during elimination. Despite some optimization strategies due e.g. to Kohler (see [Williams]), the method is usually unable to solve linear optimization problems of interesting size. So from the practical point of view it is non-competitive to the simplex method and its variants. From the theoretical viewpoint, however, the method remains important, since it provides a simple approach to basic structural theorems of the field such as the feasibility theorem, the Farkas Lemma and the Duality Theorem (compare [Dantzig & Eaves, Williams]).

Concerning the theoretical asymptotical complexity, the simplex algorithm and its variants have been analyzed extensively with the result that the method is exponential in the worst-case but polynomial in reasonable stochastic models (see [Borgwardt]). It seems that no detailed asymptotic analysis of the complexity of the Fourier–Motzkin method has been carried out so far. ([Schrijver] contains a weak lower bound; for problems of very special type, viz. at most two variables in each inequality there are good upper bounds (see [Chandru] and the references given there)). We will show below in section 3 that the fairly obvious doubly exponential worst-case upper bound (compare the hint in [Dantzig], section 4—4) on the complexity of the Fourier-Motzkin method is essentially tight.

The main purpose of the paper is to present a new elimination method for linear optimization that has only singly exponential complexity in the worst case. In fact the method is also applicable to quadratic optimization (quadratic objective function and linear constraints) and has a similar singly exponential worst-case bound in this case. Moreover, it can be applied also in situations where all coefficients of the input contain one or several parameters. Hence it can also be used to reduce a mixed integer optimization problem to several pure integer optimization problems.

The practical applicability of the method is exemplified by a series of examples computed by a REDUCE implementation ([Kappert]). They include linear programming problems with up to 50 variables and some textbook quadratic programming problems. The results indicate that the scope of our method goes significantly beyond that of the Fourier-Motzkin method and includes problems of interesting size.

The method is derived from a quantifier elimination method for linear problems in the reals developed in [Weispfenning 1], optimized in [Loos & Weispfenning] and implemented in [Sturm]. The basic idea is to test feasibility of a system of parametric linear constraints using a finite selection of parametric test points. (In principle, this idea dates back to Th. Skolem in the 1920's; in many variants and disguises it plays a major role in mathematical logic and logic programming. It should also be noted that the Fourier-Motzkin method disguised as a quantifier elimination method for the additive theory of the reals has been developed independently by the Polish school of logicians in the late 1920's (compare [Weispfenning 3]). In fact, the general problem to test parametric feasability of polynomial constraints falls within the scope of quantifier elimination methods for real algebra. The only fully implemented general purpose method for quantifier elimination in real algebra is the method by Collins and Hong (see[Collins & Hong, Hong 2]). Due to its generality, it will, however, not be able to solve parametric linear optimization problems of interesting size. Whether the new general purpose quantifier elimination method presented in [Weispfenning 2] has any bearing on the problem remains to be seen after the completion of the implementation of this method.)

In case of a linear optimization problem, the elimination of a single variable by our method corresponds geometrically to a passage from the polyhedron Π described by the linear constraints to the projections along a coordinate axis of some of its facets. During the elimination of further variables the different facets produced earlier do not interact, but are treated separately. It is only after the completion of the elimination process that the different values of the objective function at the vertices of Π produced by the elimination process are compared. This should be contrasted to the Fourier-Motzkin method, where elimination of a variable corresponds to the projection of all of Π along a coordinate axis. So, roughly speaking, our method is more 'local' in character than the 'global' Fourier-Motzkin method. This feature offers the additional advantage that our method should be well parallelizable.

In case of a quadratic objective function q we have to add more test points that correspond to possible local extrema of q with respect to the variable to be eliminated. Since these are, however, described as zeros of some partial derivatives of q, these parametric test points remain linear with respect to the remaining variables. This has the important consequence that all constraints remain linear throughout the elimination process.

The plan of the paper is as follows:

- In section 2 we introduce the basic notation and in particular weakly parametric optimization problems, i.e. problem where possible parameters enter the problem only as additive constants. The main results is the algorithm LINOPT for weakly parametric linear optimization, its verification, variants and some optimizations.

- Section 3 reviews the Fourier-Motzkin method and establishes its doubly exponential worst-case complexity. Moreover, it provides singly exponential upper complexity bounds for all the new optimization algorithms presented in the paper.
- Section 4 discusses the extension of LINOPT to the case, where all coefficients may depend on parameters. Finally this bound is shown to be tight for weakly parametric problems.
- Section 5 is devoted to the optimization of a quadratic objective function with respect to linear constraints in the weakly parametric case by the algorithm QUADOPT.
- Section 6 discusses the extension of QUADOPT to the fully parametric case.
- In the last section we apply the REDUCE-implementation in [Kappert] to some linear and quadratic optimization problems including some parametric examples. The results provide a first preliminary estimate on the range of the method in practice.

2 Linear optimization for weakly parametric data

In this section we describe algorithms that will find the minimum of a linear objective function subject to linear constraints in dependence of parameters in the absolute terms of the constraints and the target function. The solution will be obtained in the form of a finite list of triplets (C, T, val), where each C is a conjunction of polynomial inequalities in the parameters, each T is a tuple of parametric linear polynomials and of values $\pm\infty$ which – considered as a matrix – is in row echelon form, and each val is a polynomial in the parameters $\underline{u}$.

This solution has the following semantic significance: Suppose the parameters have been given fixed real values. If all systems C evaluate to false, then the system of constraints is infeasible for the given values of the parameters. Otherwise, consider the set M of all (C, T, val) such that C evaluates to the value 'true'. Then two cases may occur: Either none of the entries of T contains a symbol $\pm\infty$; then for each $(C, T, \mathrm{val}) \in M$ back-substitution via T produces a point in $\mathbf{R}^n$, where n is the number of variables. Pick that point that corresponds to the minimal value of val in the corresponding triplet. Then this point is a minimum point of the objective function in the polyhedron described by the linear constraints for the given values of the parameters, and the corresponding value of val is the minimum of the objective function.

In the remaining case the constraints will be feasible but will describe a polyhedron that is unbounded in those components of T, where $\pm\infty$ occurs. If in this case all values of 'val' are $\neq \infty$, then the minimum of these values will be the minimum of the objective function on the polyhedron. Otherwise, the objective function will have no minimum on the polyhedron.

We consider linear constraints of the form $p(\underline{u}, \underline{x}) \geq 0$, where $p(u_1, \ldots, u_m,\ x_1, \ldots, x_n)$ is of the form $a_0(\underline{u}) + a_1(\underline{u})x_1 + \ldots + a_n(\underline{u})x_n$ and $a_i(u_1, \ldots, u_m)$ are arbitrary polynomials in $u_1, \ldots, u_m$ with rational coefficients.

We refer to $u_1, \ldots, u_m$ as the **parameters** of p, to $x_1, \ldots, x_n$ as the **variables** of p, and to $p(\underline{u}, \underline{x})$ as a **linear term** w.r.t. $\underline{x}$.

We call $p(\underline{u}, \underline{x})$ **parameter-free**, if all a_i are rational constants and we call $p(\underline{u}, \underline{x})$ **weakly parametric**, if all a_i except possibly $a_0(\underline{u})$ are rational constants. The **$\underline{u}$-degree** of $p(\underline{u}, \underline{x})$ is the maximum of all total degrees of the $a_i(\underline{u})$ $(0 \leq i \leq n)$.

We say the constraint $p(\underline{u}, \underline{x}) \geq 0$ is **in normal-form** with respect to some variable $x = x_r$ $(1 \leq r \leq n)$, if it is written as

$$\alpha x \geq \beta, \quad \text{where} \quad \alpha = a_r(\underline{u}),$$

$$\beta = \beta(\underline{u}, x_1, \ldots, x_{r-1}, x_{r+1}, \ldots, x_n) = -(a_0(\underline{u}) + \sum_{i=1, i\neq r}^{n} a_i(\underline{u}) \cdot x_i).$$

Note that for weakly parametric p, $\alpha \in \mathbf{Q}$.

Similar conventions apply to the objective function $q(\underline{u}, \underline{x}) = q(u_1, \ldots, u_m,\ x_1, \ldots, x_n)$, which is a linear term w.r.t. $\underline{x}$ of the form $c_0(\underline{u}) + c_1(\underline{u})x_1 + \ldots + c_n(\underline{u})x_n$, where $c_i(u_1, \ldots, u_m)$ are arbitrary polynomials in $u_1, \ldots, u_m$ with rational coefficients. Again we call $q(\underline{u}, \underline{x})$ **parameter-free**, if all c_i are rational constants, and we call $q(\underline{u}, \underline{x})$ **weakly parametric**, if all c_i except possibly $c_0(\underline{u})$ are rational constants. The **$\underline{u}$-degree** of $q(\underline{u}, \underline{x})$ is the maximum of all total degrees of the $c_i(\underline{u})$ $(0 \leq i \leq n)$. We say the objective function $q(\underline{u}, \underline{x}) \geq 0$ is **in normal-form** with respect to some variable $x = x_r$ if it is written as

$$q = \gamma x + \delta, \quad \text{where} \quad \gamma = c_r(\underline{u}),$$

$$\delta = \delta(\underline{u}, x_1, \ldots, x_{r-1}, x_{r+1}, \ldots, x_n) = c_0(\underline{u}) + \sum_{i=1, i\neq r}^{n} c_i(\underline{u}) \cdot x_i.$$

Again for weakly parametric q, $\gamma \in \mathbf{Q}$. Geometrically speaking, the condition that a linear term is weakly parametric means that its graph is determined up to a translation only.

In the sequel, we will use the notation $p[t/x_r]$ for the polynomial obtained from p by substituting the polynomial t for x_r in p. We will also formally substitute the values $\pm\infty$ for the variable x into a weakly parametric constraint $\alpha x \geq \beta$ that is in normal form w.r.t. x. The semantic meaning of such a substitution is the obvious one. i.e.

$$\alpha\infty \geq \beta \quad \Leftrightarrow \quad \text{'true'} \;\; \text{if} \;\; \alpha > 0,$$

$$\alpha\infty \geq \beta \quad \Leftrightarrow \quad 0 \geq \beta \quad \text{if} \;\; \alpha = 0,$$

$$\alpha\infty \geq \beta \quad \Leftrightarrow \quad \text{'false'} \;\; \text{if} \;\; \alpha < 0.$$

Similar for $-\infty$ in place of ∞.

Theorem 2.1 (Weakly parametric linear optimization) *Let*

$$(*) \quad p_i(\underline{u}, x_1, \ldots, x_n) \geq 0 \quad (i \in S)$$

be a finite system of weakly parametric constraints, let $q(\underline{u}, x_1, \ldots, x_n)$ be a weakly parametric linear objective function and let $1 \leq r \leq n$. Suppose the normal form of $p_i(\underline{u}, \underline{x}) \geq 0$ with respect to $x = x_r$ is $\alpha_i x \geq \beta_i$ and the normal form of $q(\underline{u}, \underline{x})$ with respect to $x = x_r$ is $\gamma x + \delta$. Group the subscripts of constraints into pairwise disjoint sets

$$I_{-1} = \{\, i \in S \mid \alpha_i < 0 \,\}\,, \quad I_0 = \{\, i \in S \mid \alpha_i = 0 \,\} \quad I_1 = \{\, i \in S \mid \alpha_i > 0 \,\}$$

Then for arbitrary real values of $u_1, \ldots, u_m, x_1, \ldots, x_{r-1}, x_{r+1}, \ldots, x_n$ and an arbitrary real number y the following equivalences hold:

1. *Suppose $\gamma > 0$. Then*

   $$(*) \qquad \bigwedge_{i \in S} \alpha_i x \geq \beta_i \;\wedge\; \gamma x + \delta \leq y$$

   has a real solution iff $I_1 \neq \emptyset$ and some $i \in I_1$, $x = \beta_i/\alpha_i$ satisfies $(*)$ or $I_1 = \emptyset$ and $-\infty$ satisfies $(*)$.

2. *Suppose $\gamma < 0$. Then*

   $$(*) \qquad \bigwedge_{i \in S} \alpha_i x \geq \beta_i \;\wedge\; \gamma x + \delta \leq y$$

   has a real solution iff $I_{-1} \neq \emptyset$ and some $i \in I_{-1}$, $x = \beta_i/\alpha_i$ satisfies $(*)$ or $I_{-1} = \emptyset$ and ∞ satisfies $(*)$.

3. *Suppose $\gamma = 0$. Then the following assertions are equivalent:*

(i)

$$(*) \qquad \bigwedge_{i \in S} \alpha_i x \geq \beta_i \ \wedge \ \gamma x + \delta \leq y$$

has a real solution.

(ii) $I_1 \neq \emptyset$ *and some* $i \in I_1,\ x = \beta_i/\alpha_i$ *satisfies* $(*)$ *or* $I_1 = \emptyset$ *and* $-\infty$ *satisfies* $(*)$.

(iii) $I_{-1} \neq \emptyset$ *and some* $i \in I_{-1},\ x = \beta_i/\alpha_i$ *satisfies* $(*)$ *or* $I_{-1} = \emptyset$ *and* ∞ *satisfies* $(*)$.

Proof. In each case the implication yielding the solvability of $(*)$ from the respective condition is trivial.

For the converse direction we argue as follows: For fixed real values of $u_1, \ldots, u_m$, $x_1, \ldots, x_{r-1}, x_{r+1}, \ldots, x_n$ and hence of β_i and δ_i each constraint $\alpha_i x \geq \beta_i$ defines either the interval $(-\infty, \beta_i/\alpha_i]$, $\mathbf{R}$, $\emptyset$, or $[\beta_i/\alpha_i, \infty)$ depending on the sign $-1, 0, +1$ of α_i $(i \in S)$. Since by hypothesis the set M of real solutions of

$$(*) \qquad \bigwedge_{i \in S} \alpha_i x \geq \beta_i$$

$(*)$ is non-empty, this set is of one of the following forms

1. $M = \mathbf{R}$
2. $M = (-\infty, \beta_i/\alpha_i]$ and $\alpha_i < 0$
3. $M = [\beta_i/\alpha_i, \infty)$ and $\alpha_i > 0$
4. $M = [\beta_i/\alpha_i,\ \beta_j/\alpha_j]$ and $\alpha_j < 0 < \alpha_i$

Now we distinguish the 3 possible signs of γ:

1. $\gamma > 0$: Then we get the following solutions of $(*)$ in the respective cases $(i) - (iv)$: $-\infty, -\infty, \beta_i/\alpha_i, \beta_i/\alpha_i$.
2. $\gamma < 0$: Then we get the following solutions of $(*)$ in the respective cases $(i) - (iv)$: $\infty, \beta_i/\alpha_i, \infty, \beta_j/\alpha_j$.
3. $\gamma = 0$: Then we get the following solutions of $(*)$ in the respective cases $(i) - (iv)$: $-\infty, -\infty, \beta_i/\alpha_i, \beta_i/\alpha_i$ or alternatively $(i) - (iv)$: $\infty, \beta_i/\alpha_i, \infty, \beta_j/\alpha_j$.
$\square$

The theorem suggests the following algorithm LINOPT1 for the elimination of a single variable in a weakly parametric linear optimization problem. Its correctness follows immediately from the assertions of the theorem. As a subalgorithm it uses a procedure SIMPLIFY that takes as input any finite system P of polynomial inequalities, evaluates the variable-free parts of P to the boolean values 'true' or 'false', deletes all values 'true' from P, and replaces P by '$\{-1 \geq 0\}$ ' in case a 'false' occurs.

Algorithm LINOPT1

Input: A system $P = \{ \ p_i(u_1, \ldots, u_m, \ x_1, \ldots, x_n) \geq 0 \ \mid \ i \in S \ \}$ of weakly parametric linear constraints and a weakly parametric linear objective function $q(\underline{u}, \underline{x})$, both of $\underline{u}$-degree $\leq D$, and some $1 \leq r \leq n$.

Output: A finite familiy $\{ \ (P_j, t_j, q_j) \ \mid \ j \in J \ \}$ of triplets (P_j, t_j, q_j) such that

(i) $|J| \leq |S|$.

(ii) P_j is a system $\{ \ p_{ji}(\underline{u}, \ x_1, \ldots, x_{r-1}, x_{r+1}, \ldots, x_n) \geq 0 \ \mid \ i \in S_j \}$ of weakly parametric linear constraints of $\underline{u}$-degree $\leq D$ with $|S_j| \leq |S|$.

(iii) $t_j = t_j(\underline{u}, x_1, \ldots, x_{r-1}, x_{r+1}, \ldots, x_n)$ is a weakly parametric linear term of $\underline{u}$-degree $\leq D$ or $t_j = \pm\infty$.

(iv) $q_j = q_j(\underline{u}, \ x_1, \ldots, x_{r-1}, x_{r+1}, \ldots, x_n)$ is a weakly parametric linear objective function of $\underline{u}$-degree $\leq D$ or $q_j = -\infty$.

(v) If $q_j = -\infty$, then $t_j = \pm\infty$; if $t_j = \pm\infty$, then $J = \{1\}$ and so $j = 1$.

(vi) For arbitrary real values of $\underline{u}, \ x_1, \ldots, x_{r-1}, x_{r+1}, \ldots, x_n$ and $M = \{ \ x_r \in \mathbf{R} \mid p_i(\underline{u}, \underline{x}) \geq 0$ for $\ i \in S \ \}$ the following assertions hold

- $M \neq \emptyset \iff$ for some $j \in J, \ P_j(\underline{u}, \ x_1, \ldots, x_{r-1}, x_{r+1}, \ldots, x_n)$ holds in $\mathbf{R}$.
- If $M \neq \emptyset$ and $q(\underline{u}, \underline{x})$ has a unique minimum with respect to x_r in M, then for some $j \in J, \ P_j(\underline{u}, \ x_1, \ldots, x_{r-1}, \ x_{r+1}, \ldots, x_n)$ holds in $\mathbf{R}$, and $p_{ji} = p_i(\underline{u}, x_1, \ldots, x_{r-1}, t_j, x_{r+1}, \ldots, x_n)$, for all $\ i \in S_j$; moreover, $t_j(\underline{u}, \ x_1, \ldots, x_{r-1}, x_{r+1}, \ldots, x_n) \ \in \ M$, and $q_j(\underline{u}, \ x_1, \ldots, x_{r-1}, x_{r+1}, \ldots, x_n) = q(\underline{u}, \ x_1, \ldots, x_{r-1}, \ t_j, x_{r+1}, \ldots, x_n) = \min\{ \ q(\underline{u}, \ \underline{x}) \mid x_r \in M \ \}$.
- If q has no minimum with respect to x_r in M, then $J = \{1\}, \ t_1 = \pm\infty$ and $q_1 = -\infty$. If q is independent of x_r (i.e. $c_r = 0$) and M is unbounded, then $|J| = 1, \ t_1 = \pm\infty$, and $q_1 = q$.

begin
Put each $p_i \geq 0$ into normal form $\alpha_i x_r \geq \beta_i$ and put q into normal form
$\gamma x_r + \delta$ with respect to x_r;
$I_{-1} := \ \{ \ i \in S \mid \alpha_i < 0 \ \}$;
$I_0 := \ \{ \ i \in S \mid \alpha_i = 0 \ \}$;
$I_1 := \ \{ \ i \in S \mid \alpha_i > 0 \ \}$;
if $\gamma > 0$ **then**
if $I_1 \neq \emptyset$ **then**
$J := I_1$;
for each $j \in J$ **do**
$t_j := \beta_j / \alpha_j$ for $j \in J$;
$P_j :=$ SIMPLIFY($\{ \ p_i[t_j/x_r] \geq 0 \mid i \in S, \ i \neq j \ \}$);

```
            q_j := q[t_j/x_r]
    else   % (case I_1 = ∅)
        J := { 1 };
        t_1 := ±∞;
        P_1 := { −β_i ≥ 0 | i ∈ I_0 };
        q_1 := −∞
  elseif γ = 0  then
      if I_1 = ∅  then
        J := { 1 };
        t_1 := ±∞;
        P_1 := { −β_i ≥ 0 | i ∈ I_0 };
        q_1 := δ
      elseif I_{−1} = ∅  then
          J := { 1 };
          t_1 := ±∞;
          P_1 := { −β_i ≥ 0 | i ∈ I_0 };
          q_1 := δ
      elseif |I_1| ≤ |I_{−1}|  then
          J := I_1;
          for each j ∈ J do
              t_j := β_j/α_j;
              P_j := SIMPLIFY({ p_i[t_j/x_r] ≥ 0 | i ∈ S,  i ≠ j });
              q_j := q[t_1/x_r]
      else  %  (case |I_1| > |I_{−1}|)
          J := I_{−1};
          for each j ∈ J do
              t_j := β_j/α_j;
              P_j := SIMPLIFY({ p_j[t_j/x_r] ≥ 0 | i ∈ S,  i ≠ j });
              q_j := q[t_j/x_r]
  else  % (case γ < 0)
      if I_{−1} ≠ ∅  then
        J := I_{−1};
        foreach j ∈ J do
          t_j := β_j/α_j;
          P_j := SIMPLIFY({ p_i[t_j/x_r] ≥ 0 | i ∈ S,  i ≠ j });
          q_j := q[t_j/x_r]
      else
          J := { 1 };
          t_1 := ±∞;
          P_1 := { −β_i ≥ 0 | i ∈ I_0 };
          q_1 := −∞
end
```

An iteration of the algorithm LINOPT1 creates a finite tree of triplets as in LINOPT1 with respect to the successive variables $x_n, x_{n-1}, \dots, x_1$. So the depth of this tree is bounded by n. In order to minimize the storage requirements, the following algorithm traverses this tree in a depth-first manner. Whenever it arrives at a leaf of the tree, it stores the relevant information in an output list L.

Algorithm LINOPT

Input: A weakly parametric system $P = \{\ p_i(\underline{u}, \underline{x}) \geq 0 \mid i \in I\ \}$ of linear constraints and a weakly parametric linear objective function $q(\underline{u}, \underline{x})$, both of $\underline{u}$-degree $\leq D$.

Output: A list L of triplets $(C(\underline{u}),\ T,\ \mathrm{val}(\underline{u}))$ such that

(i) $C(\underline{u})$ is a finite system of polynomial inequalities in $\underline{u}$ of degree $\leq D$.

(ii) $T = (t_1(\underline{u}),\ t_2(\underline{u}, x_1), \dots,\ t_n(\underline{u},\ x_1, \dots, x_n))$, where each t_i is a weakly parametric linear term of $\underline{u}$-degree $\leq D$ or $\pm\infty$.

(iii) $\mathrm{val}(\underline{u})$ is a polynomial of degree $\leq D$ or $-\infty$.

(iv) For arbitrary real values of $\underline{u}$ the following assertions hold:

- If $C(\underline{u})$ fails for all triplets (C, T, val) in L, then the system P is infeasible for the given value of $\underline{u}$.
- Let $M = \{\ (C, T, \mathrm{val})$ in $L \mid C(\underline{u})$ holds and $val(\underline{u}) \neq -\infty\ \}$. If $M \neq \emptyset$, then $q(\underline{u}, \underline{x})$ assumes a minimum under the constraints P and this minimum equals $\min\{\ val(\underline{u}) \mid (C, T, \mathrm{val}) \in M\ \}$. Moreover, this minimum is assumed at the point in $\mathbf{R}^n$ obtained from the value of $\underline{u}$ by successive back-substitution according to the list T.
- If $M = \emptyset$ and $N = \{\ (C, T, \mathrm{val})$ in $L \mid C(\underline{u})$ holds $\ \} \neq \emptyset$, then for all $(C, T, \mathrm{val}) \in N$, $\mathrm{val}(\underline{u}) = -\infty$, the system P is feasible and describes an unbounded polyhedron, and q has no minimum under the constraints P.

begin
 $Path :=\ (\mathrm{LINOPT1}(P, q, n))$;
 $Substpath :=\ ()$;
 $L :=\ ()$;
 $m :=\ 1$;
 while $Path \neq ()$ **do**
  $\{\ (P_j, t_j, q_j) \mid j \in J\ \} :=\ \mathrm{first}(Path)$;
  $Path :=\ \mathrm{rest}(Path)$;

$k :=$ some element of J;
$J := J \setminus \{ k \}$;
$Path :=$ cons($\{(P_j, t_j, q_j) \mid j \in J\}$, $Path$);
if $P_k \neq (-1 \geq 0)$ **then**
  if $m = n$ **then**
    $L :=$ cons($(P_k,$ cons$(t_k, Substpath), q_k), L)$
  else
      $r := n - m$;
      $\{ (P'_\ell, t'_\ell, q'_\ell) \mid \ell \in L \} :=$ LINOPT1(P_k, q_k, r);
      $Substpath :=$ cons$(t_k, Substpath)$;
      $Path :=$ cons($\{ P'_\ell, t'_\ell, q'_\ell) \mid \ell \in L \}, Path)$;
      $m := m + 1$;
  if $J = \emptyset$ **then**
    $Path :=$ rest $(Path)$;
    $Substpath :=$ rest $(Substpath)$;
    $m := m - 1$
end

end

Correctness and Termination of the algorithm: Consider the set Tr of all triplets (P_j, t_j, q_j) obtained from (P, q) by iterated application of LINOPT1 with respect to the variables $x_1, \ldots, x_{n-m+1}$ $(1 \leq m \leq n)$. Then the triplet satisfies the specification given for the output of LINOPT1, but with respect to the variables $x_1, \ldots, x_{n-m}$ instead of $x_1, \ldots, x_{r-1},\ x_{r+1}, \ldots, x_n$. Call m the level of such a triplet. Define a binary relation $\rightarrow$ on Tr to hold between a triplet (P_j, t_j, q_j) of level m and $(P_{j'}, t_{j'}, q_{j'})$ of level $m+1$ if the latter occurs in the output of LINOPT1 applied to (P_j, q_j). Then $(Tr, \rightarrow)$ forms a finite tree of depth n (provided all the index sets occuring in iterated applications of LINOPT1 are kept disjoint). Define a **tube** T of Tr to be a set of nodes of Tr obtained from a path P of Tr by taking all immediate successors of all nodes in P and deleting all nodes in P.

We claim that each node of Tr is selected at most once in the second line of the while-loop. To prove this fact we note that after entering the while loop the set of all triplets occuring in families of 'Path' always forms a tube.

Moreover, whenever new nodes enter 'Path' at a later stage, these nodes a children of a parent-mode in the former value of 'Path' that has been cancelled in this step. Hence no node can be selected more than once in LINOPT.

Since Tr is finite, this proves the termination of LINOPT.

Another loop invariant is the fact that the subscripts of the terms in 'Substpath' describe exactly the path P from which the tube 'Path' is derived. Using this fact and the specification one can show as a loop invariant that each triplet of level m occuring in the first entry of 'Path' satisfies a specification similar that of the output triplet of LINOPT, but with respect to the variables $\underline{u},\ x_1,\ldots,x_{n-m}$. So upon termination, every node of Tr has been selected exactly once and, since triplets from 'Path' are added to 'L' only in case $m = n$, the output-specification holds. $\square$

Remarks

(i) In case a sequence contains entries ∞ or $-\infty$ the back substitution has to be performed in such a way that a positive (negative) real of sufficiently large absolute value is to be substituted for $\infty(-\infty)$. The correctness of this choice can be tested using the constraints P. In principle, more explicit points could easily be obtained in these cases by an obvious modification of the algorithm LINOPT1 at the expense of additional complexity.

(ii) If the input of LINOPT contains no parameter, it is not necessary to store the whole output list L, but only a single triplet from L. For this purpose, whenever a new triplet is appended to L in LINOPT one checks, whether the value of val in the new entry is smaller than in the previous entry; if yes, then the new triplet is stored; otherwise the old triplet is kept and the new triplet is discarded.

In other words, the line of code

$L :=$ cons $((P_k,$ cons $(t_k, \mathit{Substpath}), q_k), L)$

is changed to

if $L = ((C, T, \mathrm{val}))$ with val $> q_k$ **then**
$\quad L := ((P_k,$ cons $(t_k, \mathit{Substpath}), q_k))$

(iii) If the objective function $q(\underline{u}, \underline{x})$ in the input of LINOPT is taken as a constant function, then the output of the algorithm constitutes a parametric feasibility test; moreover, it produces parametrized sample points for the polyhedron described by the linear constraints P, provided this is non-empty for fixed value of the parameters $\underline{u}$.

In this case the algorithm could easily be extended to accept also strict inequalities in the input. The corresponding algorithm would use the virtual substitution of linear terms involving also infinitesimals in the manner of [Loos & Weispfenning].

(iv) If no parameters are present in the input of LINOPT, then in the list L of triplets $(C, T,$ val$)$ produced as an output all systems C of constraints are empty, backsubstitution using T produces a real number, and val is a real number or $-\infty$. If L is empty, then the system P of constraints is infeasible. If $-\infty$ occurs as some value of val, then all values of val will be $-\infty$, the polyhedron described by P is unbounded, and $q(\underline{x})$ has no minimum on this polyhedron. If all values of val are reals (and hence in fact rational numbers), then the least of these numbers is the minimum of $q(\underline{x})$ on the polyhedron described by P, and back-substitution via the corresponding sequence T produces a point in this polyhedron, where the minimum is attained.

In fact, in the parameter-free case, the elimination of the last variable by LINOPT1 can be avoided in favour of a direct substitution of the minimal or maximal bound in the constraints.

(v) LINOPT may also be employed in a different manner, more in the spirit of the Fourier–Motzkin method (compare [Dantzig], section 4—4, [Williams]). Instead of entering the objective function q explicitly, one may enter the additional weakly parametric constraint $y - q \geq 0$ (with a new variable y) to P to get P' and put the new target function $q' = 0$, i.e., use LINOPT as a feasibility test. Then if Q has a minimum point under the constraints P, such a point is found from the output of LINOPT as follows: For every triplet $(C, T, 0)$ in the output list L, group C into two disjoint parts: One part C_1 involving only the parameters $\underline{u}$ and the remaining part C_2, where the inequalities involving y have been normalized to the form $y \geq h(\underline{u})$. Put $V_{(C,T,0)} = \max\{\ h(\underline{u}) \mid (y \geq h(\underline{u}) \in C_2\ \}$. For given real values of $\underline{u}$ let $(C', T', 0) \in L$ be chosen such that $V_{(C',T',0)} = \min\{\ V_{(C,T,0)} \mid (C, T, 0) \in L\ \}$. Then $V_{(C',T',0)}$ is the minimum value of q under the constraints P, and the point in $\mathbf{R}^n$ obtained by back-substitution from T is a minimum point w.r.t. P.

This option has the advantage of producing never more and frequently less test points in each elimination step at the low cost of handling one extra final variable. Practical experience with the REDUCE–implementation in [Kappert] suggest that for non-parametric problems this option is always superior to the direct approach.

(vi) One may wonder why we admit parameters in the weakly parametric objective function q, since the problem of finding a minimum point for q is obviously not affected by an additive constant. Using successive elimination of variables with weakly parametric constraints, we are however forced to admit parameters in q, since they may enter the objective function already after the first elimination of a variable.

(vii) For given value of $\underline{u}$, we may consider the set V of all points in $\mathbf{R}^n$ obtained by back substitution from those term sequences T in the output of LINOPT, that contain no symbol $\pm\infty$, and for which the corresponding set $C(\underline{u})$ evaluates to 'true'. Then all points of V are vertices of the polyhedron Π described by P. This follows from the fact that all these points are in Π and by construction also in the intersection of n of the hyperplanes whose corresponding halfspaces define P.

Notice that the performance of LINOPT may depend crucially on the order of the variables $x_1, \ldots, x_n$, which may be chosen arbitrarily. Consider, e.g., the following two-variable parameter-free linear optimization problem:

Let $s > 1$, $P = \{ y \geq 0, \ y \geq x - 1 \} \cup \{ y \geq m_i(x - c_i) \mid \ 1 \leq i \leq s \}$ $q = 2y - x$ with $-2 < m_1 < \ldots < m_s < 0$, $0 < c_1 < \ldots < c_s < 1$.

If we order $(x, y) = (x_1, x_2)$, i.e., if we eliminate y first, the first elimination step (application of LINOPT1) will produce an index set J of size $(s + 2)$, where s can be arbitrarily large. For the reverse order $(y, x) = (x_1, x_2)$ LINOPT1 produces only a single triplet, which after elimination of x_1 yields the desired minimum point of $q(x, y)$. (The situation is easily visualized by a diagram.) This example shows moreover that it is in general not advisable to give priority in elimination to those variables x_r, where the maximum of the absolute values of the coefficients c_i occurs (choice of steepest descent along the coordinate axes). In the example this would be y.

So in order to optimize the performance of LINOPT we may proceed as follows.

Let SELECTVAR be an algorithm satisfying the following specification:

Algorithm: SELECTVAR

Input: A weakly parametric system $P = \{ p_i(\underline{u}, \underline{x}) \geq 0 \mid i \in I \}$ of linear constraints and a weakly parametric linear objective function $q(\underline{u}, \underline{x})$.

Output: Some r, $1 \leq r \leq n$ such that the number of significant occurences of x_r in P is minimal, where the number of significant occurences of x_r in P w.r.t. q is defined by

$$\mathrm{NSO}(x_r) = \begin{cases} |\{ i \in I \mid a_{ir} > 0 \}| & : \ c_r > 0 \\ |\{ i \in I \mid a_{ir} < 0 \}| & : \ c_r < 0 \\ \min(|\{ i \in I \mid a_{ir} > 0 \}|, \ |\{ i \in I \mid a_{ir} < 0 \}|) & : \ c_r = 0 \end{cases}$$

Let LINOPTSELECTVAR be the algorithm LINOPT with the following change:

Replace the assignment '$r := n - m + 1$' by '$r :=$ SELECTVAR(P_k, q_k)'. Then this modified algorithm satisfies the same specification as LINOPT with the only exception that in (ii) the variables x_j occuring in t_i have to be adapted according to the successive choices of indices made by SELECTVAR.

3 Complexity and comparison with the Fourier-Motzkin method

In this section we compute a worst-case upper complexity bound for the algorithm LINOPT and compare these bound to the upper and lower worst case bound for the elimination method of Fourier–Motzkin.

The classical elimination method initiated by Fourier in [Fourier] and rediscovered by Motzkin in [Motzkin] is basically a method for testing the feasibility of a parameter-free or weakly parametric system of linear constraints. Using the trick of a sliding objective function (compare Remark (iii) following LINOPT, and [Dantzig], section 4—4) it can also be used to minimize a linear objective function. Various applications of the Fourier-Motzkin method to parametric problems have been studied in [Eaves & Rothblum, Eaves & Rothblum, Eaves & Rothblum]. Due to the doubly exponential worst-case complexity (see below) of the method and its extreme space requirements in practice, it is not competitive to the simplex algorithm and its variants.

Instead of LINOPT1 the Fourier–Motzkin elimination uses the following algorithm FM1 for the elimination of a single variable x_r:

Algorithm FM1

Input: A weakly parametric system

$$P = \{\ p_i(u_1, \ldots, u_m,\ x_1, \ldots, x_n) \geq 0 \mid i \in S\ \}$$

of linear constraints of u-degree $\leq D$ and some $1 \leq r \leq n$.

Output: A triplet (L, U, P') such that

(i) L and U are systems of weakly parametric linear terms in $x_1, \ldots, x_{r-1},\ x_{r+1}, \ldots, x_n$ of u-degree $\leq D$. The index sets of both L and U comprise together at most $|S|$ elements.

(ii) P' is a weakly parametric system of linear constraints in $x_1, \ldots, x_{r-1},\ x_{r+1}, \ldots, x_n$ of u-degree $\leq D$ with an index set of at most $|S|^2/4$ elements.

(iii) For arbitrary real values of $\underline{u},\ x_1, \ldots, x_{r-1},\ x_{r+1}, \ldots, x_n$ and $M = \{x_r \in \mathbf{R} \mid p_i(\underline{u}, \underline{x}) \geq 0 \ \text{ for } \ i \in S\}$ the following assertions hold:

- $M \neq \emptyset \Leftrightarrow P'$ holds.
- If $M \neq \emptyset$, then $M = \{x_r \in \mathbf{R} \mid \max(L) \leq x_r \leq \min(U)\}$.

begin

Put each $p_i \geq 0$ into normal form $\alpha_i x_r \geq \beta_i$ with respect to $x = x_r$;
$I_{-1} := \{i \in S \mid \alpha_i < 0\}$;
$I_0 := \{i \in S \mid \alpha_i = 0\}$;

$I_1 := \{i \in S \mid \alpha_i > 0\}$;
$L := \{\beta_i \mid i \in I_1\}$;
$U := \{\beta_i \mid i \in I_{-1}\}$;
$P' := \{\beta_i - \beta_j \geq 0 \mid i \in I_{-1},\ j \in I_1\} \ \cup \ \{-\beta_k \mid k \in I_0\}$

end

The correctness of the algorithm is obvious. (Notice that the number of inequalities in P' is given by $k(n-\ell-k)+\ell$, where $k = |I_1|,\ \ell = |I_0|$. So for fixed n, ℓ we have $\max(k(n-\ell-k)) = (n-\ell)^2/4$, and for fixed $n \geq 2,\ \max((n-\ell)^2/4 + \ell \mid 0 \leq \ell \leq n) = n^2/4$.)

In order to eliminate all variables x_i, the algorithm is iterated w.r.t. the successive variables $x_n, \ldots, x_1$ by applying FM1 to the last component of the previous output. (Note that the order of the variables can again be chosen arbitrarily.) This results in the following algorithm FM.

Algorithm FM

Input: A weakly parametric system $P = \{\ p_i(\underline{u}, \underline{x}) \geq 0 \mid i \in S\ \}$ of linear constraints of $\underline{u}$-degree $\leq D$.

Output: A list R of pairs $(L(\underline{u}, x_1, \ldots, x_{n-i}), U(\underline{u}, x_1, \ldots, x_{n-i})\quad (i = 1 \ldots n)$ and a system $Q(\underline{u})$ of polynomial inequalities of the form $q(\underline{u}) \geq 0$, such that

(i) Each L and each U is a finite system of weakly parametric terms in the variables displayed of $\underline{u}$-degree $\leq D$. The number of terms in $L \cup U$ is bounded by

$$\frac{(|S|)^{2^{(i-1)}}}{2^{(2^i-2)}}$$

for the i-th entry (L, U) of R.

(ii) Each $q \in Q$ has degree $\leq D$ in $\underline{u}$. The index set of Q has at most

$$\frac{(|S|)^{2^n}}{2^{(2^{n+1}-2)}}$$

many elements.

(iii) For arbitrary real values of $\underline{u}$ the following assertions hold:

- If $Q(\underline{u})$ fails, then the system P is infeasible for the given value of $\underline{u}$.

- If $Q(\underline{u})$ holds, then the system P is feasible for the given value of $\underline{u}$. Moreover, a point in the polyhedron described by P can be obtained by the following back-substitution process: Choose a real number y_1 such that

$$\max(L) \leq y_1 \leq \min(U)$$

for the last pair (L, U) in R. Substitute this value y_1 for x_1 in the next to last pair (L', U') in R to obtain a pair (L'', U'') and choose a real number y_2 such that

$$\max(L'') \leq y_2 \leq \min(U''),$$

and so on. The resulting point $\underline{y} = (y_1, \ldots, y_n)$ will then satisfy P.

begin
    $m := n;$
    $R := ();$
    $Q := P;$
    while $m \geq 0$ **do**
        $(L, U, P') := FM1(Q, x_m);$
        $R :=$ cons $((L, U),\ R);$
        $Q := P';$
        $m := m - 1$
    end
end

The bounds on the size of the output are easily proved by induction on n from the corresponding bounds in FM1. Notice that the list R is required not only in the course of the algorithm, but also in the output in order to admit the construction of a point in the polyhedron described by P by back-substitution. So the total number of polynomials in the output of FM is bounded in the worst case by

$$\sum_{i=1}^{n} \frac{|S|^{2^{(i-1)}}}{2^{(2^i-2)}} + \frac{|S|^{2^n}}{2^{(2^{n+1}-2)}}$$

and hence (for $|S| \geq 2$) more roughly by

$$(n+1)\Big(\frac{|S|}{2}\Big)^{2^n}.$$

Next we show that the first bound is also a worst-case lower bound. For this purpose it suffices to find for every $n \in \mathbf{N}$ an input P such that

- x_n has no zero coefficient in any p_i.
- The number of positive coefficients of x_n in the p_i's equals the number of negative coefficients of x_n in the p_i's.
- These two properties are preserved under the passage from P to P' obtained by an application of FM1.

We describe the input P consisting of 2^n linear constraints by its $(2^n \times (n+1))$-matrix A of the coefficients a_{ij}: The column of entries a_{i0} can be chosen arbitrarily.

For the remaining columns we assume for the moment that our base field is a non-archimedean extension $\mathbf{Q}^*$ of $\mathbf{Q}$ containing an element ω such that $\mathbf{Q} < \omega$. Call $\alpha \in \mathbf{Q}^*$ **finite**, if there exists $q \in \mathbf{Q}$ with $|\alpha| \leq q$.

Then the j-th column of A $(1 \leq j \leq n)$ looks as follows:
The first $2^{(j-1)}$ entries a_{ij} are of the form $-1 + k_{ij}/\omega^{2j}$, where k_{ij} are pairwise different elements of $\mathbf{Q}^*$, such that both k_{ij} and k_{ij}^{-1} are finite. The next $2^{(j-1)}$ entries a_{ij} are of the form $\omega^{2j+1} + h_{ij}$ for pairwise different finite h_{ij} in $\mathbf{Q}^*$. The remaining entries a_j are arbitrary pairwise different finite elements of $\mathbf{Q}^*$.

Then after one application of FM1 (eliminating x_n), this structure is preserved, except that the number of rows has increased as in the worst-case from $|S| = 2^n$ to $|S'| = \left(\frac{2^n}{2}\right)^2$, and in the structure description the number $2^{(j-1)}$ has to be multiplied by the factor $2^n/2$.

So the iterated elimination of variables using FM1 will always remain in the worst case. This shows that the worst-case behaviour occurs in this series of examples.

The provisional hypothesis that the input coefficients contain an 'infinitely large' ω can be eliminated by choosing $\omega \in \mathbf{Q}$ sufficiently large w.r.t. the remaining 'finite' coefficients. For a formal proof of this fact one uses the 'Tarski principle' for real closed fields.

Let us compare this bound with a corresponding worst case bound for the algorithm LINOPT applied as a feasibility test.

Let *Tr* be the tree described in the correctness argument for LINOPT as before, we assume $q \equiv 0$. Then an induction on m using the specification of LINOPT1 shows that the total number of nodes of level m in *Tr* is bounded in the worst case by $\left(\frac{|S|}{2}\right)^m$ (provided $|S| \geq 2$). For $m = n$ we thus obtain at most $\left(\frac{|S|}{2}\right)^n$ elements in the output list L. So the total number of polynomial inequalities in the output list L is bounded by $|S| \cdot \left(\frac{|S|}{2}\right)^n$ and hence the total number of polynomials in the output is bounded by $\frac{|S|^n}{2^n}(|S| + n + 1)$. The total number of polynomials to be stored in the lists 'Path' and 'Substpath' ar any given time during the execution of the algorithm is bounded by $n(|S|+2)$ and n, respectively. So the total number of polynomial to be stored during the execution of the algorithm LINOPT never exceeds

$$\frac{|S|^n}{2^n}(|S|+n+1) \ + \ n(|S|+3).$$

Hence we get a singly exponential worst-case bound in comparison to the doubly exponential worst-case bound for the Fourier-Motzkin method. Apart from this difference, the output is in both cases obtained by similar operations, i.e., normalization with respect to some variables and substitution of linear terms for variables. So the time-complexities of both methods are in similar relation to each other.

If LINOPT is applied to a non-trivial objective function q, then the size of the index set in the output of LINOPT1 may increase in the worst-case from $\frac{|S|}{2}$ to $|S|$. Accordingly, the output list L has at most $|S|^n$ elements, the total number of polynomials in L is bounded by $|S|^n\ (|S|+n+1)$, and the total number of polynomials to be stored during the execution of the algorithm LINOPT is bounded by

$$|S|^n(|S|+n+1) \ + \ n(|S|+3).$$

So again we have a singly exponential complexity in terms of the number of polynomials in the output.

Using instead $y-q \geq 0$ as a new constraint as in remark (v) following LINOPT are even get the better bound of the feasability test. For the time complexity one has then to add the time for eliminating the parameter y which does not change the outcome essentially.

The same argument will apply to the modified and extended algorithms LINOPTPAR, QUADOPT, QUADOPTPAR. In each case, the algorithms LINOPT1PAR, QUADOPT1, QUADOPT1PAR replacing LINOPT1 will produce a family of triplets, where the index set J has at most $|S|+1$ elements. Accordingly the respective output list L will have at most $(|S|+1)^n$ elements, the total number of polynomials in the output will be bounded by

$$(|S|+1)^n(|S|+n+1),$$

and the total number of polynomials to be stored during the execution of the respective algorithms will be bounded by

$$(|S|+1)^n(|S|+n+1) \ + \ n(|S|+3).$$

Notice that this worst-case bound is essentially the same for linear and quadratic objective functions q.

Finally, we show that the problem of weakly parametric linear optimization is inherently exponential provided the output of the algorithms is of a form similar to the output of

LINOPT. So we assume that the algorithm provides upon input of a weakly parametric linear optimization problem at least a finite list of pairs consisting of a system $C(\underline{u})$ in the parameters and a polynomial $\mathrm{val}(\underline{u})$ providing the corresponding minimal value of the objective function q.

Consider the following series of weakly parametric inputs: $\{x_i \geq u_i | 1 \leq i \leq n\} \cup \{x_i \geq v_i | 1 \leq i \leq n\}$, $q = x_1 + \ldots + x_n$ in the parameters $\underline{u}, \underline{v}$. Obviously, $\min q = \max(u_1, v_1) + \ldots + \max(u_n, v_n)$ under the given constraints. Assume for a contradiction, that some algorithm solving the problem has an output of less than 2^n pairs. Then there exist one pair $(C(\underline{u}, \underline{v}),\ \mathrm{val}(\underline{u}, \underline{v}))$ such that $C(\underline{u}, \underline{v})$ describes a set M with non empty interior M° in $\mathbf{R}^n$ so that for some $1 \leq i \leq n$, $M_1 = M^\circ \cap \{(\underline{u}, \underline{v}) \mid u_i > v_i\} \neq \emptyset$ and $M_2 = M^\circ \cap \{(\underline{u}, \underline{v}) \mid u_i < v_i\} \neq \emptyset$. Choose open balls $B_i \subseteq M_i$ such that all $\mathrm{sgn}(u_j - v_j)$ has a constant value $\neq 0$ on B_i for all $1 \leq j \leq n$. Then $\mathrm{val}(\underline{u}, \underline{v})$ equals two different polynomial functions on B_1 and B_2, a contradiction.

By a slightly more subtle argument we can show that the feasibility problem is inherently exponential, provided one admits quadratic constraints of the form $x_i^2 - x_i = 0$ and a single additive parameter u.

Consider the following system S of constraints in $x_1, \ldots, x_{2n}$ with parameter n:

$x_i^2 - x_i = 0 \qquad \text{for} \quad 1 \leq i \leq n$

$x_{n+1} = 1$

$x_{n+i+1} = x_{n+i} + x_{n+i} \qquad \text{for} \quad 1 \leq i < n$

$x_1 x_{n+1} + \ldots + x_n x_{2n} - u = 0$

Then S is feasible iff $x_i \in \{0, 1\}$ for $1 \leq i \leq n$ and $x_{n+i} = 2^{i-1}$ for $1 \leq i < n$, and so iff u is an integer with $0 \leq u \leq 2^n - 1$. It has been shown in [Weispfenning 1], section 6, that any boolean combination of polynomial inequalities in u defining the set of all integers between 0 and $2^n - 1$ is necessarily of length exponential in n.

4 Linear optimization for parametric data

In this section we generalize the results of section 2 for linear constraints and linear objective functions, where **all** coefficients may be polynomials – or in fact rational functions – in the parameters $\underline{u}$. The main difference is that in contrast to the weakly parametric case, we are no longer able to distinguish cases according to the signs of coefficients. As a consequence, the set of test terms that are substituted for the variable x_r to be eliminated has to include the union of the sets of test terms that have been found previously. Another point is the fact that in contrast to the test terms used previously, the parameters $\underline{u}$ will now occur in the denominators. So we have to take care to handle these denominators in a correct way not depending on the actual values that may be substituted for the parameters $\underline{u}$. This can be achieved, e.g., by the convention that $c/0 =: 0$ (compare [Loos & Weispfenning]).

Note that throughout this section we admit rational functions in $\underline{u}$ as coefficients for linear terms.

Theorem 4.1 (Parametric linear optimization) *Let*

$$(*) \qquad p_i(\underline{u}, x_1, \ldots, x_n) \geq 0 \quad (i \in S)$$

be a finite system of parametric constraints and let $q(\underline{u}, x_1, \ldots, x_n)$ be a parametric quadratic objective function, both of $\underline{u}$-degree $\leq D$, and let $1 \leq r \leq n$. Suppose the normal form of $p_i(\underline{u}, \underline{x}) \geq 0$ with respect to $x = x_r$ is $\alpha_i(\underline{u})x \geq \beta_i(\underline{u})$ and the normal form of $q(\underline{u}, \underline{x})$ with respect to $x = x_r$ is $\gamma(\underline{u})x + \delta(\underline{u})$.

Then for arbitrary real values of $u_1, \ldots, u_m, x_1, \ldots, x_{r-1}, x_{r+1}, \ldots, x_n$, where all p_i and q is defined, and an arbitrary real number y the following equivalence holds:

$$(*) \qquad \bigwedge_{i \in S} \alpha_i x \geq \beta_i \ \wedge \ \gamma x + \delta \leq y$$

has a real solution iff some $i \in S$, $x = \beta_i/\alpha_i$ satisfies $(*)$ or ∞ or $-\infty$ satisfies $(*)$.

Proof. For given real values of $u_1, \ldots, u_m, x_1, \ldots, x_{r-1}, x_{r+1}, \ldots, x_n$, the case distinction on the signs of γ and the α_i can be performed as in theorem 2.1. So one can argue as in the proof of this theorem. □

In the corresponding algorithm LINOPT1PAR below, we dispense with the formal substitution of $\pm\infty$, since this would increase the complexity considerably (compare [Loos & Weispfenning]). This weakens the significance of the output slightly.

Algorithm LINOPT1PAR

Input: A system $P = \{ \ p_i(u_1, \ldots, u_m, \ x_1, \ldots, x_n) \geq 0 \ \mid \ i \in S \ \}$ of parametric linear constraints and a parametric linear objective function $q(\underline{u}, \underline{x})$, both of $\underline{u}$-degree $\leq D$, and some $1 \leq r \leq n$.

Output: A finite familiy $\{ \ (P_j, t_j, q_j) \ \mid \ j \in J \ \}$ of triplets (P_j, t_j, q_j) such that

(i) $|J| \leq |S| + 1$.

(ii) P_j is a system $\{\ p_{ji}(\underline{u},\ x_1,\ldots,x_{r-1},x_{r+1},\ldots,x_n) \geq 0 \ \mid\ i \in S_j\ \}$ of parametric linear constraints of $\underline{u}$-degree $\leq 2D$ with $|S_j| \leq |S|$.

(iii) $t_j = t_j(\underline{u},\ x_1,\ldots,x_{r-1},x_{r+1},\ldots,x_n)$ is a parametric linear term of $\underline{u}$-degree $\leq 2D$

(iv) $q_j = q_j(\underline{u},\ x_1,\ldots,x_{r-1},x_{r+1},\ldots,x_n)$ is a parametric linear objective function of $\underline{u}$-degree $\leq 2D$.

(v) For arbitrary real values of $\underline{u},\ x_1,\ldots,x_{r-1},x_{r+1},\ldots,x_n$, where the input is defined, and for $M = \{\ x_r \in \mathbf{R} \mid p_i(\underline{u},\underline{x}) \geq 0 \ \text{ for } \ i \in S\ \}$ the following assertions hold

- $M \neq \emptyset \Longleftrightarrow$ for some $j \in J,\ P_j(\underline{u},\ x_1,\ldots,x_{r-1},x_{r+1},\ldots,x_n)$ holds in $\mathbf{R}$.
- If $M \neq \emptyset$ and $q(\underline{u},\underline{x})$ has a minimum with respect to x_r in M, then for some $j \in J,\ P_j(\underline{u},\ x_1,\ldots,x_{r-1},x_{r+1},\ldots,x_n)$ holds in $\mathbf{R}$; moreover, $t_j(\underline{u},\ x_1,\ldots,x_{r-1},x_{r+1},\ldots,x_n) \in M$, and $q_j(\underline{u},\ x_1,\ldots,x_{r-1},x_{r+1},\ldots,x_n) = q(\underline{u},\ x_1,\ldots,x_{r-1},t_j,x_{r+1},\ldots,x_n) = \min\{\ q(\underline{u},\ \underline{x}) \mid x_r \in M\ \}$.

begin

  Put each $p_i \geq 0$ into normal form $\alpha_i x \geq \beta_i$ and put q into normal form
    $\gamma x + \delta$ with respect to $x = x_r$;
  $I_{\pm 1} := \{\ i \in S \mid \alpha_i \neq 0\ \}$;
  $J := I_{\pm 1} \cup \ \{\ *\ \}$;
  for each $j \in J$ **do**
    $t_j := \beta_j/\alpha_j$ for $j \in I_{\pm 1}$;
    $t_* := 0$;
    $P_j :=$ SIMPLIFY($\{\ p_i[t_j/x_r] \geq 0 \mid i \in S\ \}$);
    $q_j := q[t_j/x_r]$

end

As in the previous section we may iterate this algorithm by traversing the corresponding tree in a depth-first manner. The only change that has to be made in LINOPT is the replacement of the call to LINOPT1 by a corresponding call to LINOPT1PAR. The resulting algorithm LINOPTPAR satisfies the following specification:

Algorithm LINOPTPAR

Input: A parametric system $P = \{\ p_i(\underline{u},\underline{x}) \geq 0 \mid i \in I\ \}$ of linear constraints and a parametric linear objective function $q(\underline{u},\underline{x})$, both of $\underline{u}$-degree $\leq D$.

Output: A list L of triplets $(C(\underline{u}),\ T,\ \text{val}(\underline{u}))$ such that

(i) $C(\underline{u})$ is a finite system of rational function inequalities in $\underline{u}$ of degree $\leq 2^{(n+1)}D$.

(ii) $T = (t_1(\underline{u}),\ t_2(\underline{u}, x_1), \ldots,\ t_n(\underline{u},\ x_1, \ldots, x_{n-1}))$, where each t_i is a parametric linear term of $\underline{u}$-degree $\leq 2^{(n-i+2)}D$.

(iii) $\mathrm{val}(\underline{u})$ is a rational function of degree $\leq 2^{(n+1)}D$.

(iv) For arbitrary real values of $\underline{u}$ for which the input is defined, the following assertions hold:

- If for all triplets (C, T, val) in L, $C(\underline{u})$ fails, then the system P is infeasible for the given value of $\underline{u}$.
- Let $M = \{\ (C, T, \mathrm{val}) \text{ in } L \mid C(\underline{u}) \text{ holds }\ \}$. If $M \neq \emptyset$, then the system P is feasible for the given value of $\underline{u}$. Moreover,for every triplet $(C, T, \mathrm{val}) \in M$ the point in $\mathbf{R}^n$ obtained from the value of $\underline{u}$ by successive back-substitution according to the list T satisfies P.
- If $q(\underline{u}, \underline{x})$ assumes a minimum under the constraints P, then $M \neq \emptyset$ and this minimum equals $\min\{\ val(\underline{u}) \mid (C, T, \mathrm{val}) \in M\ \}$. Moreover, this minimum is assumed at a point in $\mathbf{R}^n$ obtained from the value of $\underline{u}$ by successive back-substitution according to the list T.

Remarks

(i) If the objective function $q(\underline{u}, \underline{x})$ in the input of LINOPTPAR is taken as a constant function, then the output of the algorithm constitutes a parametric feasibility test; moreover, it produces parametrized sample points for the polyhedron described by the linear constraints P, provided this is non-empty for fixed value of the parameters $\underline{u}$.

(ii) If LINOPT1PAR is modified so that $*$ is never added to the index set J then LINOPTPAR applied to a non-parametric system P of constraints and the zero function q yields a list of all vertices of the polyhedron Π described by P. (Compare the last remark following the algorithm LINOPT.) For a survey on other algorithms solving this problem see [Mattheiss & Rubin].

(iii) In contrast to the weakly parametric situation, an optimization of LINOPTPAR by SELECTVAR will in general not be possible, since we can not distinguish the signs of the parametric coefficients independently of the paramaters $\underline{u}$. If, however, some non-parametric coefficients are present in the input, we may instead use the following variant SELECTVARPAR of SELECTVAR for optimization.

Algorithm: SELECTVARPAR

Input: A parametric system $P = \{ p_i(\underline{u}, \underline{x}) \geq 0 \mid i \in I \}$ of linear constraints and a parametric linear objective function $q(\underline{u}, \underline{x})$.

Output: Some r, $1 \leq r \leq n$ such that the number of significant occurences of x_r in P is minimal.

where the number of significant occurences of x_r in P w.r.t. q is defined by

$$\mathrm{NSO}(x_r) = \begin{cases} |\{ i \in I \mid a_{ir} \neq 0 \}| & : \quad c_r \notin \mathbf{Q} \\ (|\{ i \in I \mid a_{ir} > 0 \}| \ + \ |\{ i \in I \mid a_{ir} \notin \mathbf{Q} \,|) & : \quad 0 < c_r \in \mathbf{Q} \\ (|\{ i \in I \mid a_{ir} < 0 \}| \ + \ |\{ i \in I \mid a_{ir} \notin \mathbf{Q} \}|) & : \quad 0 > c_r \in \mathbf{Q} \\ \min((|\{ i \in I \mid a_{ir} > 0 \}| \ + \ |\{ i \in I \mid a_{ir} \notin \mathbf{Q} \}|), & : \\ (|\{ i \in I \mid a_{ir} < 0 \}| \ + \ |\{ i \in I \mid a_{ir} \notin \mathbf{Q}|)) & : \quad c_r = 0 \end{cases}$$

5 Quadratic optimization for weakly parametric data

In this section we consider the problem of minimizing a weakly parametric **quadratic** objective function and a weakly parametric system of linear constraints.

The objective function $q(\underline{u}, x_1, \ldots, x_n)$ is **quadratic** (with respect to $\underline{x}$) if it is of the form

$$q = \sum_{i,j=1}^{n} c_{ij}(\underline{u}) x_i x_j + \sum_{i=1}^{n} c_i(\underline{u}) x_i + c_0(\underline{u}),$$

where $c_{ij}(\underline{u})$, $c_(\underline{u})$, $c_0(\underline{u})$ are polynomials in $\underline{u}$ with rational coefficients. As before the $\underline{u}$-**degree** of q is the maximum of the total degrees of all $c_{ij}(\underline{u})$, $c_i(\underline{u})$, $c_0(\underline{u})$.

We call $q(\underline{u}, \underline{x})$ **weakly parametric** if all c_{ij}, c_i except possibly $c_0(\underline{u})$ are rational constants.

Notice that we do **not** assume $q(\underline{u}, \underline{x})$ to be **convex**.

For $1 \leq r \leq n$, $q(\underline{u}, \underline{x})$ is in normal form with respect to $x = x_r$ if q is written as

$$q = \epsilon x^2 + \gamma x + \delta,$$

where

$$\epsilon = c_{rr}, \ \gamma = \sum_{i \neq r} c_{ir} x_i + c_r, \ \delta = \sum_{i,j \neq r} c_{ij} x_i x_j + \sum_{i \neq r} c_i x_i + c_0(\underline{u}).$$

As previously, $\epsilon \in \mathbf{Q}$ for weakly parametric q. Notice, however, that in contrast to the weakly parametric linear case, γ is no longer a rational constant, but a linear polynomial in all x_i $(i \neq r)$ with rational coefficients. So for fixed real values of $\underline{u}$, $x_1, \ldots, x_{r-1}$, $x_{r+1}, \ldots, x_n$, the graph of q with respect to x_r represents a convex (concave) parabola, if $\epsilon \geq 0$ $(\epsilon < 0)$, and a straight line if $\epsilon = 0$.

Let now I be a closed interval and suppose q has a minimum on I with respect to x_r at $x_r = z \in I$. Then in case $\epsilon > 0, z$ is either an endpoint of I or a local minimum of q, i.e. $z = -\gamma/2\epsilon$; in case $\epsilon \leq 0, z$ must be an endpoint of I. So the argument used for the proof of theorem 4.1 may be repeated verbatim using the additional test point $-\gamma/2\epsilon$ in case $\epsilon > 0$. The crucial point here is that $-\gamma/2\epsilon$ is a linear polynomial in $x_1, \ldots, x_{r-1}$, $x_{r+1}, \ldots, x_n$ with rational coefficients. So the substitution of $-\gamma/2\epsilon$ for x_r in a weakly parametric linear or quadratic term will again yield a weakly parametric linear or quadratic term, respectively. As a result we obtain the following theorem:

Theorem 5.1 (Weakly parametric quadratic optimization) *Let*

$$(*) \qquad p_i(\underline{u}, x_1, \ldots, x_n) \geq 0 \quad (i \in S)$$

be a finite system of weakly parametric constraints and let $q(\underline{u}, x_1, \ldots, x_n)$ be a weakly parametric quadratic objective function, both of $\underline{u}$-degree $\leq D$, and let $1 \leq r \leq n$. Suppose the normal form of $p_i(\underline{u}, \underline{x}) \geq 0$ with respect to $x = x_r$ is $\alpha_i x \geq \beta_i$ and the normal form of $q(\underline{u}, \underline{x})$ with respect to $x = x_r$ is $\epsilon x^2 + \gamma x + \delta$. Group the subscripts of the constraints into pairwise disjoint sets $I_{\pm 1}, I_0$ such that

$$I_{\pm 1} = \{ \ i \in S \mid \alpha_i \neq 0 \ \}, \quad I_0 = \{ \ i \in S \mid \alpha_i = 0 \ \}.$$

Then for arbitrary real values of $u_1, \ldots, u_m, x_1, \ldots, x_{r-1}, x_{r+1}, \ldots, x_n$ and an arbitrary real number y the following equivalences hold:

1. *Suppose $\epsilon > 0$. Then*

$$(*) \qquad \bigwedge_{i \in S} \alpha_i x \geq \beta_i \ \wedge \ \epsilon x^2 + \gamma x + \delta \leq y$$

has a real solution iff $-\gamma/2\epsilon$ satisfies $(*)$ or $I_{\pm 1} \neq \emptyset$ and some $i \in I_1$, $x = \beta_i/\alpha_i$ satisfies (*). $I_1 = \emptyset$ and $-\infty$ satisfies $(*)$.

2. *Suppose $\epsilon \leq 0$. Then*

$$(*) \qquad \bigwedge_{i \in S} \alpha_i x \geq \beta_i \ \wedge \ \epsilon x^2 + \gamma x + \delta \leq y$$

has a real solution iff $I_{\pm 1} \neq \emptyset$ and some $i \in I_{\pm 1}$, $x = \beta_i/\alpha_i$ satisfies $(*)$ or ∞ or $-\infty$ satisfies $(*)$.

The **proof** is similar to that of theorem 3.1, taking into account the remarks preceeding the theorem. $\square$

The corresponding algorithm is QUADOPT1:

Algorithm QUADOPT1

Input: A weakly parametric system $P = \{\ p_i(u_1, \ldots, u_m,\ x_1, \ldots, x_n) \geq 0 \ \mid\ i \in S\ \}$ of linear constraints and a weakly parametric quadratic objective function $q(\underline{u}, \underline{x})$, both of $\underline{u}$-degree $\leq D$, and some $1 \leq r \leq n$.

Output: A finite familiy $\{\ (P_j, t_j, q_j) \ \mid\ j \in J\ \}$ of triplets (P_j, t_j, q_j) such that

(i) $|J| \leq |S|$.

(ii) P_j is a system $\{\ p_{ji}(\underline{u},\ x_1, \ldots, x_{r-1}, x_{r+1}, \ldots, x_n) \geq 0 \ \mid\ i \in S_j\ \}$ of weakly parametric linear constraints of $\underline{u}$-degree $\leq D$ with $|S_j| \leq |S|$.

(iii) $t_j = t_j(\underline{u},\ x_1, \ldots, x_{r-1}, x_{r+1}, \ldots, x_n)$ is a weakly parametric linear term of $\underline{u}$-degree $\leq D$.

(iv) $q_j = q_j(\underline{u},\ x_1, \ldots, x_{r-1}, x_{r+1}, \ldots, x_n)$ is a weakly parametric quadratic objective function of $\underline{u}$-degree $\leq D$.

(v) For arbitrary real values of $\underline{n},\ x_1, \ldots, x_{r-1}, x_{r+1}, \ldots, x_n$ and $M = \{\ x_r \in \mathbf{R} \mid p_i(\underline{u}, \underline{x}) \geq 0$ for $\ i \in S\ \}$ the following assertions hold:

- $M \neq \emptyset \iff$ for some $j \in J,\ P_j(\underline{u},\ x_1, \ldots, x_{r-1}, x_{r+1}, \ldots, x_n)$ holds in $\mathbf{R}$.
- If $M \neq \emptyset$ and $q(\underline{u}, \underline{x})$ has a minimum with respect to x_r in M, then for some $j \in J,\ P_j(\underline{u},\ x_1, \ldots, x_{r-1}, x_{r+1}, \ldots, x_n)$ holds in $\mathbf{R}$, and $p_{ji} = p_i(\underline{u}, x_1, \ldots, x_{r-1}, t_j, x_{r+1}, \ldots, x_n)$ for all $i \in S_j$; moreover, $t_j(\underline{u},\ x_1, \ldots, x_{r-1}, x_{r+1}, \ldots, x_n) \in M$, and $q_j(\underline{u},\ x_1, \ldots, x_{r-1}, x_{r+1}, \ldots, x_n) = q(\underline{u},\ x_1, \ldots, x_{r-1}, t_j, x_{r+1}, \ldots, x_n) = \min\{\ q(\underline{u},\ \underline{x}) \mid x_n \in M\ \}$.

begin

  Put each $p_i \geq 0$ into normal form $\alpha_i x \geq \beta_i$ and
    put q into normal form $\epsilon x^2 + \gamma x + \delta$ with respect to $x = x_r$;
  $I_{\pm 1} :=\ \{\ i \in S \mid \alpha_i \neq 0\ \}$;
  $I_0 :=\ \{\ i \in S \mid \alpha_i = 0\ \}$;
  if $\epsilon > 0$ **then**
    $J := I_{\pm 1} \cup\ \{\ *\ \}$;
    for each $j \in J$ **do**
      $t_j := \beta_j / \alpha_j$ for $j \in I_{\pm 1}$;
      $t_* := -\gamma / 2\epsilon$;
      $P_j :=$ SIMPLIFY($\{\ p_i[t_j / x_r] \geq 0 \mid i \in S\ \}$);
      $q_j := q[t_j / x_r]$

```
else  % (case ϵ ≤ 0)
    if I_{±1} ≠ ∅ then
      J := I_{±1};
      for each j ∈ J  do
            t_j := β_j/α_j;
            P_j := SIMPLIFY({ p_i[t_j/x_r] ≥ 0 | i ∈ S,  i ≠ j });
            q_j := q[t_j/x_r]
    else J := { * };
        t_* := 0;
        P_* := SIMPLIFY({ p_i[0/x_r] ≥ 0 | i ∈ S, });
        q_* := q[0/x_r]
end
```

As in the previous sections we may iterate this algorithm by traversing the corresponding tree in a depth-first manner. The only change that has to be made in LINOPT is again the replacement of the call to LINOPT1 by a corresponding call to QUADOPT1. The resulting algorithm QUADOPT satisfies the following specification:

Algorithm QUADOPT

Input: A weakly parametric system $P = \{ \ p_i(\underline{u}, \underline{x}) \geq 0 \mid i \in I \ \}$ of linear constraints and a weakly parametric quadratic objective function $q(\underline{u}, \underline{x})$, both of $\underline{u}$-degree $\leq D$.

Output: A list L of triplets $(C(\underline{u}),\ T,\ \mathrm{val}(\underline{u}))$ such that

(i) $C(\underline{u})$ is a finite system of polynomial inequalities in $\underline{u}$ of degree $\leq D$.

(ii) $T = (t_1(\underline{u}),\ t_2(\underline{u}, x_1), \ldots,\ t_n(\underline{u},\ x_1, \ldots, x_{n-1}))$, where each t_i is a weakly parametric linear term of $\underline{u}$-degree $\leq D$.

(iii) $\mathrm{val}(\underline{u})$ is a polynomial of degree $\leq D$.

(iv) For arbitrary real values of $\underline{u}$ the following assertions hold:

- If for all triplets (C, T, val) in L, $C(\underline{u})$ fails, then the system P is infeasible for the given value of $\underline{u}$.
- Let $M = \{ \ (C, T, \mathrm{val}) \text{ in } L \mid C(\underline{u}) \text{ holds } \}$. If $M \neq \emptyset$, then the system P is feasible for the given value of $\underline{u}$. Moreover,for every triplet $(C, T, \mathrm{val}) \in M$ the point in $\mathbf{R}^n$ obtained from the value of $\underline{u}$ by successive back-substitution according to the list T satisfies P.

- If $q(\underline{u}, \underline{x})$ assumes a minimum under the constraints P, then $M \neq \emptyset$ and this minimum equals $\min\{ \, val(\underline{u}) \mid (C, T, \mathrm{val}) \in M \, \}$. Moreover, this minimum is assumed at a point in $\mathbf{R}^n$ obtained from the value of $\underline{u}$ by successive back-substitution according to the list T.

Remarks

1. Notice that as in the case of the algorithm LINOPTPAR, the algorithm QUADOPT does not provide a necessary and sufficient criterion for the existence of a minimum of q under the constraints P. This weakness is due to the fact that we have no way of deciding the sign of the coefficient γ of q in QUADOPT1, since this coefficient may still involve the remaining x_i as parameters. If, however, a unique minimum exists, for given values of the parameters, then the output provides this minimum point.

2. In contrast to the situation, where the objective function is linear, there seems to be no a priori strategy for the choice of the order of the x_i in the present case, except for a preference of those x_r that have few non-zero coefficients. Moreover, the variant of including the objective function q in the form of a constraint $y - q \geq 0$ as in LINOPT is impossible now due to the non-linearity of q.

6 Parametric Quadratic Optimization

Here the changes that occur in comparison to the weakly parametric quadratic case are similar to those in the corresponding linear case. So again we must admit rational functions in the parameters $\underline{u}$ in the coefficients of linear and quadratic terms and we cannot perform case distinctions with respect to the sign of the highest coefficient $\epsilon(\underline{u})$ of q with respect to some variable x_r.

So the algorithm QUADOPT1PAR is described by the following modification of QUADOPT1:

Algorithm QUADOPT1PAR

begin
Put each $p_i \geq 0$ into normal form $\alpha_i x \geq \beta_i$ and
put q into normal form $\epsilon x^2 + \gamma x + \delta$ with respect to $x = x_r$;
$I_{\pm 1} := \{ \, i \in S \mid \alpha_i \neq 0 \, \}$;
$I_0 := \{ \, i \in S \mid \alpha_i = 0 \, \}$;
$J := I_{\pm 1} \cup \{ \, * \, \}$;
for each $j \in J$ **do**

$t_j := \beta_j/\alpha_j$ for $j \in I_{\pm 1}$;
$t_* := -\gamma/2\epsilon$;
$P_j :=$ SIMPLIFY($\{\ p_i[t_j/x_r] \geq 0 \mid i \in S\ \}$);
$q_j := q[t_j/x_r]$

end

The algorithm QUADOPTPAR is as before obtained by replacing the call to LINOPT1 in LINOPT by a call to QUADOPTPAR.

The resulting specification of QUADOPTPAR reads as follows:

Algorithm QUADOPTPAR

Input: A parametric system $P = \{\ p_i(\underline{u}, \underline{x}) \geq 0 \mid i \in I\ \}$ of linear constraints and a parametric quadratic objective function $q(\underline{u}, \underline{x})$, both of $\underline{u}$-degree $\leq D$.

Output: A list L of triplets $(C(\underline{u}),\ T,\ \mathrm{val}(\underline{u}))$ such that

(i) $C(\underline{u})$ is a finite system of rational function inequalities in $\underline{u}$ of degree $\leq 2^{(n+1)}D$.

(ii) $T = (t_1(\underline{u}),\ t_2(\underline{u}, x_1), \ldots,\ t_n(\underline{u},\ x_1, \ldots, x_{n-1}))$, where each t_i is a parametric linear term of $\underline{u}$-degree $\leq 2^{(n-i+2)}D$.

(iii) $\mathrm{val}(\underline{u})$ is a rational function of degree $\leq 2^{(n+1)}D$.

(iv) For arbitrary real values of $\underline{u}$ for which the input is defined, the following assertions hold:

- If for all triplets (C, T, val) in L, $C(\underline{u})$ fails, then the system P is infeasible for the given value of $\underline{u}$.
- Let $M = \{\ (C, T, \mathrm{val}) \text{ in } L \mid C(\underline{u}) \text{ holds }\ \}$. If $M \neq \emptyset$, then the system P is feasible for the given value of $\underline{u}$. Moreover,for every triplet $(C, T, \mathrm{val}) \in M$ the point in $\mathbf{R}^n$ obtained from the value of $\underline{u}$ by successive back-substitution according to the list T satisfies P.
- If $q(\underline{u}, \underline{x})$ assumes a minimum under the constraints P, then $M \neq \emptyset$ and this minimum equals $\min\{\mathrm{val}(\underline{u}) \mid (C, T, \mathrm{val}) \in M\ \}$. Moreover, this minimum is assumed at a point in $\mathbf{R}^n$ obtained from the value of $\underline{u}$ by successive back-substitution according to the list T.

To conclude we remark that this algorithm (with the obvious modifications) can also be applied to a parametric objective function q that is a quotient of two linear parametric polynomials in the x_i, since the condition that $\frac{\partial q}{\partial x_j} = 0$ leads again to a test-point that is linear in all x_i.

7 Examples

The following examples are computed by the REDUCE–implementation of the algorithms described above in [Kappert]. All CPU-timings are taken on a SUN Sparc-10 in milliseconds.

Example 1.

The following series of examples is taken from [Borgwardt], page 16, and is due to Klee and Minty, 1972. It constitutes a series of non-parametric examples where the simplex method requires $2^n - 1$ pivot steps. On the other hand each constraint in such example involves at most two variables. So it is known that related feasibility problem is solved even by the Fourier-Motzkin method in subexponential time. In fact an improved elimination algorithm can solve the problem in polynomial time (see [Chandru]). It turns out that LINOPT solves the problem in time linear in n if rational number operations are counted in unit time.

Problem: Minimize $q = -x_n$ under the $2n$ linear constraints

$$\begin{array}{rcl} 0 & \leq x_1 \leq & 1 \\ \epsilon x_1 & \leq x_2 \leq & 1 - \epsilon x_1 \\ \epsilon x_2 & \leq x_3 \leq & 1 - \epsilon x_2 \\ \dots & \leq \dots \leq & \dots \\ \epsilon x_{n-1} & \leq x_n \leq & 1 - \epsilon x_{n-1} \end{array}$$

where $\epsilon \in \mathbf{Q}$ is fixed, $0 < \epsilon < 1/2$.

If the variables are eliminated in the order x_1 first,..., x_n last, then LINOPT will substitute 0 for $x_1, \dots, x_{n-1}$, and 1 for x_n and thus produce the correct output that $(0, \dots, 0, 1)$ is a minimum point of q under the given constraints. Under the reverse order of elimination the performance of LINOPT can also be traced by hand and is only marginally less efficient.

The REDUCE-output for the Klee-Minty-cube with 50 variables is as follows (where z is the artificial varible for the value of the objective function):

Output:

$\{\{true, \{z_0 = -1, x_1 = 0, x_2 = 0, x_3 = 0, x_4 = 0, x_5 = 0, x_6 = 0, x_7 = 0, x_8 = 0, x_9 = 0, x_{10} = 0,$

$x_{11} = 0, x_{12} = 0, x_{13} = 0, x_{14} = 0, x_{15} = 0, x_{16} = 0, x_{17} = 0, x_{18} = 0, x_{19} = 0, x_{20} = 0,$
$x_{21} = 0, x_{22} = 0, x_{23} = 0, x_{24} = 0, x_{25} = 0, x_{26} = 0, x_{27} = 0, x_{28} = 0, x_{29} = 0, x_{30} = 0,$
$x_{31} = 0, x_{32} = 0, x_{33} = 0, x_{34} = 0, x_{35} = 0, x_{36} = 0, x_{37} = 0, x_{38} = 0, x_{39} = 0, x_{40} = 0,$
$x_{41} = 0, x_{42} = 0, x_{43} = 0, x_{44} = 0, x_{45} = 0, x_{46} = 0, x_{47} = 0, x_{48} = 0, x_{49} = 0, x_{50} = 1\},$
$-1\}\}$

Time: 4488 ms plus GC time: 85 ms

Example 2.

This feasibility problem was kindly communicated to the author by H. Melenk ([Melenk]) as an instance, where the REDUCE-implementation of the Fourier-Motzkin method at the ZIB Berlin could not eliminate more than 6 variables. The original example was non-parametric with $a = 8$.

Input:

$\exists x_1 \exists x_2 \exists x_3 \exists x_4 \exists x_5 \exists x_6 \exists x_7 \exists x_8 \exists x_9 \exists x_{10} \exists x_{11} \exists x_{12} \exists x_{13} \exists x_{14}$
$\exists x_{15}($
$x_1 + x_{10} + x_{11} + x_{12} + x_{13} + x_{14} + x_{15} + x_2 + x_3 + x_4 + x_5 + x_6 + x_7 + x_8 + x_9 \leq a$
$\wedge$
$-x_1 + x_{10} - x_{11} - x_{12} - x_{13} + x_{14} - x_{15} + x_2 + x_3 + x_4 - x_5 - x_6 - x_7 + x_8 + x_9 \leq 0$
$\wedge$
$x_1 + x_{10} - x_{11} - x_{12} + x_{13} - x_{14} - x_{15} - x_2 + x_3 + x_4 - x_5 + x_6 + x_7 - x_8 - x_9 \leq 0$
$\wedge$
$x_1 - x_{10} - x_{11} + x_{12} - x_{13} - x_{14} - x_{15} + x_2 - x_3 + x_4 + x_5 - x_6 + x_7 - x_8 + x_9 \leq 0$
$\wedge$
$x_1 - x_{10} + x_{11} - x_{12} - x_{13} - x_{14} - x_{15} + x_2 + x_3 - x_4 + x_5 + x_6 - x_7 + x_8 - x_9 \leq 0$
$\wedge$
$-x_1 + x_{10} + x_{11} + x_{12} - x_{13} - x_{14} + x_{15} - x_2 + x_3 + x_4 + x_5 - x_6 - x_7 - x_8 - x_9 \leq 0$
$\wedge$
$-x_1 - x_{10} + x_{11} - x_{12} + x_{13} - x_{14} + x_{15} + x_2 - x_3 + x_4 - x_5 + x_6 - x_7 - x_8 + x_9 \leq 0$
$\wedge$
$-x_1 - x_{10} - x_{11} + x_{12} + x_{13} - x_{14} + x_{15} + x_2 + x_3 - x_4 - x_5 - x_6 + x_7 + x_8 - x_9 \leq 0$
$\wedge$
$x_1 - x_{10} + x_{11} - x_{12} - x_{13} + x_{14} + x_{15} - x_2 - x_3 + x_4 - x_5 - x_6 + x_7 + x_8 - x_9 \leq 0$
$\wedge$
$x_1 - x_{10} - x_{11} + x_{12} - x_{13} + x_{14} + x_{15} - x_2 + x_3 - x_4 - x_5 + x_6 - x_7 - x_8 + x_9 \leq 0$
$\wedge$
$x_1 + x_{10} - x_{11} - x_{12} + x_{13} + x_{14} + x_{15} + x_2 - x_3 - x_4 + x_5 - x_6 - x_7 - x_8 - x_9 \leq 0$
$\wedge$
$-x_1 - x_{10} - x_{11} + x_{12} + x_{13} + x_{14} - x_{15} - x_2 - x_3 + x_4 + x_5 + x_6 - x_7 + x_8 - x_9 \leq 0$
$\wedge$
$-x_1 - x_{10} + x_{11} - x_{12} + x_{13} + x_{14} - x_{15} - x_2 + x_3 - x_4 + x_5 - x_6 + x_7 - x_8 + x_9 \leq 0$
$\wedge$
$-x_1 + x_{10} + x_{11} + x_{12} - x_{13} + x_{14} - x_{15} + x_2 - x_3 - x_4 - x_5 + x_6 + x_7 - x_8 - x_9 \leq 0$
$\wedge$
$x_1 + x_{10} + x_{11} + x_{12} + x_{13} - x_{14} - x_{15} - x_2 - x_3 - x_4 - x_5 - x_6 - x_7 + x_8 + x_9 \leq 0$

$\wedge$
$-x_1 + x_{10} - x_{11} - x_{12} - x_{13} - x_{14} + x_{15} - x_2 - x_3 - x_4 + x_5 + x_6 + x_7 + x_8 + x_9 \leq 0$
)

Output: $a \geq 0$

Time: 669987 ms plus GC time: 14569 ms

For the original non-parametric problem, where $a = 8$, the REDUCE-timings are much better, since the algorithm stops once it has found a condition 'TRUE'. Here is corresponding output:

Output: true

Time: 527 ms

Example 3.

This example was kindly communicated to the author by g. Meisinger ([Meisinger]). The original example was non-parametric with $a = 0,\ b = 14$.

Input:

$\exists x_{20} \exists x_{21} \exists x_{40} \exists x_{41} \exists x_{42} \exists x_8 \exists x_9 \exists x_{17} \exists x_{18} \exists x_2 \exists x_{34} \exists x_{37} \exists x_5$
$\exists x_1 \exists x_{32} \exists x_{33} \exists x_4 \exists x_{36} ($
$-6x_{10} + 18x_{20} - 6x_{21} - 6x_{40} + 3x_{41} - x_{42} - 36x_8 + 18x_9 \geq a \wedge$
$-15x_{17} + 15x_{18} - 21x_{20} + 3x_{21} + 7x_{40} - x_{41} + 42x_8 - 6x_9 \geq a \wedge$
$-3x_{10} + 5x_{18} - 10x_2 - x_{21} - 5x_{34} + x_{37} - 10x_5 \geq a \wedge$
$4x_{10} - 10x_{18} + 20x_2 + 3x_{21} + 4x_{34} - x_{37} - 10x_5 \geq a \wedge$
$7x_1 + 7x_{16} - x_{17} - 5x_2 - 7x_{32} + x_{33} + 7x_4 - x_5 - 7x_8 + x_9 \geq b \wedge$
$-9x_{36} + 3x_{37} + 6x_{40} - 3x_{41} + x_{42} \geq a \wedge$
$15x_{17} - 15x_{18} + 21x_{20} - 3x_{21} - 14x_{40} + 2x_{41} \geq a \wedge$
$-9x_{20} + 3x_{21} + 6x_{40} - 3x_{41} + x_{42} \geq a \wedge$
$9x_{18} - 3x_{21} - 6x_{34} + 3x_{37} - x_{42} \geq a \wedge$
$-3x_{16} + 2x_{32} - x_{33} + x_{34} - x_{36} + x_{40} \geq a \ \wedge \ -7x_8 + x_9 \geq a \wedge$
$-4x_{10} + x_{20} - 3x_{36} + 10x_4 + 2x_{40} - x_{41} + x_{42} - 8x_8 + 4x_9 \geq a \wedge$
$4x_{10} - 3x_{20} + x_{21} - 3x_{37} + 9x_{39} - 30x_4 - 6x_{40} + 3x_{41} - x_{42} + 10x_5 + 24x_8 - 12x_9 \geq a$
$\wedge\ 3x_4 - x_5 \geq a \ \wedge \ -7x_1 - x_{17} + 2x_2 + x_{33} - x_5 + x_9 \geq a)$

Output: true

Time: 272 ms

Example 4.

This is the three-dimensional transportation problem, with the bounds (3,2,2) for the range of the 3 indices (see [Loos & Weispfenning] for a more detailed discussion).

Input:
$\exists x_{322}\exists x_{222}\exists x_{122}\exists x_{321}\exists x_{221}\exists x_{121}\exists x_{312}\exists x_{212}\exists x_{112}\exists x_{311}$
$\exists x_{211}\exists x_{111}(-a_{11} + x_{111} + x_{112} = 0 \ \wedge \ -a_{12} + x_{121} + x_{122} = 0 \wedge$
$-a_{21} + x_{211} + x_{212} = 0 \ \wedge \ -a_{22} + x_{221} + x_{222} = 0 \ \wedge \ -a_{31} + x_{311} + x_{312} = 0$
$\wedge -a_{32} + x_{321} + x_{322} = 0 \ \wedge \ -b_{11} + x_{111} + x_{121} = 0 \wedge$
$-b_{12} + x_{112} + x_{122} = 0 \ \wedge \ -b_{21} + x_{211} + x_{221} = 0 \ \wedge \ -b_{22} + x_{212} + x_{222} = 0$
$\wedge -b_{31} + x_{311} + x_{321} = 0 \ \wedge \ -b_{32} + x_{312} + x_{322} = 0 \wedge$
$-c_{11} + x_{111} + x_{211} + x_{311} = 0 \ \wedge \ -c_{12} + x_{112} + x_{212} + x_{312} = 0 \wedge$
$-c_{21} + x_{121} + x_{221} + x_{321} = 0 \ \wedge \ -c_{22} + x_{122} + x_{222} + x_{322} = 0 \ \wedge \ x_{111} \geq 0$
$\wedge \ x_{112} \geq 0 \ \wedge \ x_{121} \geq 0 \ \wedge \ x_{122} \geq 0 \ \wedge \ x_{211} \geq 0 \ \wedge \ x_{212} \geq 0 \ \wedge \ x_{221} \geq 0 \ \wedge \ x_{222} \geq 0 \wedge$
$x_{311} \geq 0 \ \wedge \ x_{312} \geq 0 \ \wedge \ x_{321} \geq 0 \ \wedge \ x_{322} \geq 0)$

Output:

$((a_{12} - a_{31} - b_{11} + b_{32} + c_{11} - c_{22} \geq 0 \wedge$
$a_{11} + a_{12} - b_{11} + b_{32} - c_{12} - c_{22} \leq 0 \wedge$
$a_{12} - a_{31} - b_{11} - b_{21} + b_{32} + c_{11} - c_{22} \leq 0 \wedge$
$b_{11} + b_{21} + b_{31} - c_{11} - c_{21} = 0 \ \wedge \ a_{31} + a_{32} - b_{31} - b_{32} = 0 \wedge$
$a_{11} + a_{21} + a_{31} - c_{11} - c_{12} = 0 \ \wedge \ a_{11} + a_{12} - b_{11} - b_{12} = 0 \wedge$
$a_{11} + a_{12} - b_{11} + b_{22} + b_{32} - c_{12} - c_{22} = 0 \wedge$
$a_{12} + a_{22} - a_{31} - b_{11} - b_{21} + b_{32} + c_{11} - c_{22} = 0 \ \wedge \ a_{31} - b_{32} + c_{22} \geq 0 \wedge$
$a_{12} - a_{31} - b_{11} + b_{32} - c_{22} \leq 0 \ \wedge \ b_{31} \geq 0 \ \wedge \ a_{31} \geq 0 \wedge$
$a_{12} - a_{31} + b_{32} - c_{22} \geq 0 \ \wedge \ a_{11} + a_{12} - a_{31} - b_{11} + b_{32} - c_{22} \geq 0 \wedge$
$a_{31} - b_{32} \leq 0)$
$\vee \ (a_{12} - a_{31} - b_{11} + b_{32} + c_{11} - c_{22} \geq 0 \wedge$
$a_{11} + a_{12} - b_{11} + b_{32} - c_{12} - c_{22} \leq 0 \wedge$
$a_{12} - a_{31} - b_{11} - b_{21} + b_{32} + c_{11} - c_{22} \leq 0 \wedge$
$b_{11} + b_{21} + b_{31} - c_{11} - c_{21} = 0 \ \wedge \ a_{31} + a_{32} - b_{31} - b_{32} = 0 \wedge$
$a_{11} + a_{21} + a_{31} - c_{11} - c_{12} = 0 \ \wedge \ a_{11} + a_{12} - b_{11} - b_{12} = 0 \wedge$

$a_{11} + a_{12} - b_{11} + b_{22} + b_{32} - c_{12} - c_{22} = 0 \wedge$
$a_{12} + a_{22} - a_{31} - b_{11} - b_{21} + b_{32} + c_{11} - c_{22} = 0 \;\wedge\; c_{22} \geq 0 \wedge$
$a_{12} - b_{11} - c_{22} \leq 0 \;\wedge\; a_{31} - b_{31} - b_{32} \leq 0 \;\wedge\; b_{32} \geq 0 \;\wedge\; a_{12} - c_{22} \geq 0 \wedge$
$a_{11} + a_{12} - b_{11} - c_{22} \geq 0 \;\wedge\; a_{31} - b_{32} \geq 0)$
$\vee\; (a_{12} - a_{31} - b_{11} + b_{32} + c_{11} - c_{22} \geq 0 \wedge$
$a_{11} + a_{12} - b_{11} + b_{32} - c_{12} - c_{22} \leq 0 \wedge$
$a_{12} - a_{31} - b_{11} - b_{21} + b_{32} + c_{11} - c_{22} \leq 0 \wedge$
$b_{11} + b_{21} + b_{31} - c_{11} - c_{21} = 0 \;\wedge\; a_{31} + a_{32} - b_{31} - b_{32} = 0 \wedge$
$a_{11} + a_{21} + a_{31} - c_{11} - c_{12} = 0 \;\wedge\; a_{11} + a_{12} - b_{11} - b_{12} = 0 \wedge$
$a_{11} + a_{12} - b_{11} + b_{22} + b_{32} - c_{12} - c_{22} = 0 \wedge$
$a_{12} + a_{22} - a_{31} - b_{11} - b_{21} + b_{32} + c_{11} - c_{22} = 0 \;\wedge\; a_{11} + a_{12} - b_{11} \geq 0 \wedge$
$a_{11} \geq 0 \;\wedge\; a_{11} + a_{12} - a_{31} - b_{11} + b_{31} + b_{32} - c_{22} \geq 0 \wedge$
$a_{11} + a_{12} - b_{11} + b_{32} - c_{22} \geq 0 \;\wedge\; a_{11} - b_{11} \leq 0 \;\wedge\; a_{11} + a_{12} - b_{11} - c_{22} \leq 0 \wedge$
$a_{11} + a_{12} - a_{31} - b_{11} + b_{32} - c_{22} \leq 0)$
$\vee\; (a_{12} - a_{31} - b_{11} + b_{32} + c_{11} - c_{22} \geq 0 \wedge$
$a_{11} + a_{12} - b_{11} + b_{32} - c_{12} - c_{22} \leq 0 \wedge$
$a_{12} - a_{31} - b_{11} - b_{21} + b_{32} + c_{11} - c_{22} \leq 0 \wedge$
$b_{11} + b_{21} + b_{31} - c_{11} - c_{21} = 0 \;\wedge\; a_{31} + a_{32} - b_{31} - b_{32} = 0 \wedge$
$a_{11} + a_{21} + a_{31} - c_{11} - c_{12} = 0 \;\wedge\; a_{11} + a_{12} - b_{11} - b_{12} = 0 \wedge$
$a_{11} + a_{12} - b_{11} + b_{22} + b_{32} - c_{12} - c_{22} = 0 \wedge$
$a_{12} + a_{22} - a_{31} - b_{11} - b_{21} + b_{32} + c_{11} - c_{22} = 0 \;\wedge\; a_{12} \geq 0 \;\wedge\; b_{11} \geq 0 \wedge$
$a_{12} - a_{31} + b_{31} + b_{32} - c_{22} \geq 0 \;\wedge\; a_{12} + b_{32} - c_{22} \geq 0 \;\wedge\; a_{11} - b_{11} \geq 0 \wedge$
$a_{12} - c_{22} \leq 0 \;\wedge\; a_{12} - a_{31} + b_{32} - c_{22} \leq 0)$
$\vee\; (a_{11} - b_{11} \geq 0 \;\wedge\; a_{12} \geq 0 \;\wedge\; b_{11} \geq 0 \;\wedge\; b_{11} + b_{21} + b_{31} - c_{11} - c_{21} = 0 \wedge$
$a_{31} + a_{32} - b_{31} - b_{32} = 0 \;\wedge\; a_{11} + a_{21} + a_{31} - c_{11} - c_{12} = 0 \wedge$
$a_{11} + a_{12} - b_{11} - b_{12} = 0 \;\wedge\; a_{11} + a_{12} - b_{11} + b_{22} + b_{32} - c_{12} - c_{22} = 0 \wedge$
$a_{12} + a_{22} - a_{31} - b_{11} - b_{21} + b_{32} + c_{11} - c_{22} = 0 \wedge$
$a_{12} - a_{31} + b_{32} - c_{22} \leq 0 \;\wedge\; b_{11} - c_{11} \leq 0 \;\wedge\; b_{31} \geq 0 \;\wedge\; a_{31} \geq 0 \wedge$
$a_{11} + a_{31} - b_{11} - c_{12} \leq 0 \;\wedge\; b_{11} + b_{21} - c_{11} \geq 0 \;\wedge\; a_{31} - b_{32} \leq 0)$
$\vee\; (a_{11} - b_{11} \geq 0 \;\wedge\; a_{12} \geq 0 \;\wedge\; b_{11} \geq 0 \;\wedge\; b_{11} + b_{21} + b_{31} - c_{11} - c_{21} = 0 \wedge$
$a_{31} + a_{32} - b_{31} - b_{32} = 0 \;\wedge\; a_{11} + a_{21} + a_{31} - c_{11} - c_{12} = 0 \wedge$
$a_{11} + a_{12} - b_{11} - b_{12} = 0 \;\wedge\; a_{11} + a_{12} - b_{11} + b_{22} + b_{32} - c_{12} - c_{22} = 0 \wedge$
$a_{12} + a_{22} - a_{31} - b_{11} - b_{21} + b_{32} + c_{11} - c_{22} = 0 \;\wedge\; a_{12} - c_{22} \leq 0 \wedge$
$a_{31} + b_{11} - b_{32} - c_{11} \leq 0 \;\wedge\; a_{31} - b_{31} - b_{32} \leq 0 \;\wedge\; b_{32} \geq 0 \wedge$
$a_{11} - b_{11} + b_{32} - c_{12} \leq 0 \;\wedge\; a_{31} + b_{11} + b_{21} - b_{32} - c_{11} \geq 0 \;\wedge\; a_{31} - b_{32} \geq 0)$
$\vee\; (a_{11} - b_{11} \geq 0 \;\wedge\; a_{12} \geq 0 \;\wedge\; b_{11} \geq 0 \;\wedge\; b_{11} + b_{21} + b_{31} - c_{11} - c_{21} = 0 \wedge$
$a_{31} + a_{32} - b_{31} - b_{32} = 0 \;\wedge\; a_{11} + a_{21} + a_{31} - c_{11} - c_{12} = 0 \wedge$
$a_{11} + a_{12} - b_{11} - b_{12} = 0 \;\wedge\; a_{11} + a_{12} - b_{11} + b_{22} + b_{32} - c_{12} - c_{22} = 0 \wedge$
$a_{12} + a_{22} - a_{31} - b_{11} - b_{21} + b_{32} + c_{11} - c_{22} = 0 \wedge$
$a_{12} - a_{31} - b_{11} - b_{21} + b_{32} + c_{11} - c_{22} \leq 0 \;\wedge\; b_{21} \geq 0 \wedge$
$b_{11} + b_{21} + b_{31} - c_{11} \geq 0 \;\wedge\; a_{31} + b_{11} + b_{21} - c_{11} \geq 0 \wedge$
$a_{11} + a_{31} + b_{21} - c_{11} - c_{12} \leq 0 \;\wedge\; a_{31} + b_{11} + b_{21} - b_{32} - c_{11} \leq 0 \wedge$

$b_{11} + b_{21} - c_{11} \leq 0)$
$\vee\ (a_{11} - b_{11} \geq 0\ \wedge\ a_{12} \geq 0\ \wedge\ b_{11} \geq 0\ \wedge\ b_{11} + b_{21} + b_{31} - c_{11} - c_{21} = 0\ \wedge$
$a_{31} + a_{32} - b_{31} - b_{32} = 0\ \wedge\ a_{11} + a_{21} + a_{31} - c_{11} - c_{12} = 0\ \wedge$
$a_{11} + a_{12} - b_{11} - b_{12} = 0\ \wedge\ a_{11} + a_{12} - b_{11} + b_{22} + b_{32} - c_{12} - c_{22} = 0\ \wedge$
$a_{12} + a_{22} - a_{31} - b_{11} - b_{21} + b_{32} + c_{11} - c_{22} = 0\ \wedge$
$a_{11} + a_{12} - b_{11} + b_{32} - c_{12} - c_{22} \leq 0\ \wedge\ a_{11} + a_{31} - c_{11} - c_{12} \leq 0\ \wedge$
$a_{11} + a_{31} - b_{11} - b_{31} - c_{12} \leq 0\ \wedge\ a_{11} - b_{11} - c_{12} \leq 0\ \wedge$
$a_{11} + a_{31} + b_{21} - c_{11} - c_{12} \geq 0\ \wedge\ a_{11} - b_{11} + b_{32} - c_{12} \geq 0\ \wedge$
$a_{11} + a_{31} - b_{11} - c_{12} \geq 0)$
$\vee\ (a_{12} - a_{31} - b_{11} + b_{32} + c_{11} - c_{22} \leq 0\ \wedge\ a_{11} + a_{31} - c_{11} - c_{12} \leq 0\ \wedge$
$b_{21} \geq 0\ \wedge\ b_{11} + b_{21} + b_{31} - c_{11} - c_{21} = 0\ \wedge\ a_{31} + a_{32} - b_{31} - b_{32} = 0\ \wedge$
$a_{11} + a_{21} + a_{31} - c_{11} - c_{12} = 0\ \wedge\ a_{11} + a_{12} - b_{11} - b_{12} = 0\ \wedge$
$a_{11} + a_{12} - b_{11} + b_{22} + b_{32} - c_{12} - c_{22} = 0\ \wedge$
$a_{12} + a_{22} - a_{31} - b_{11} - b_{21} + b_{32} + c_{11} - c_{22} = 0\ \wedge\ a_{12} - b_{11} + c_{11} \geq 0\ \wedge$
$c_{11} \geq 0\ \wedge\ b_{31} \geq 0\ \wedge\ a_{31} \geq 0\ \wedge\ b_{11} - c_{11} \geq 0\ \wedge\ a_{11} - c_{11} \geq 0\ \wedge\ a_{31} - b_{32} \leq 0)$
$\vee\ (a_{12} - a_{31} - b_{11} + b_{32} + c_{11} - c_{22} \leq 0\ \wedge\ a_{11} + a_{31} - c_{11} - c_{12} \leq 0\ \wedge$
$b_{21} \geq 0\ \wedge\ b_{11} + b_{21} + b_{31} - c_{11} - c_{21} = 0\ \wedge\ a_{31} + a_{32} - b_{31} - b_{32} = 0\ \wedge$
$a_{11} + a_{21} + a_{31} - c_{11} - c_{12} = 0\ \wedge\ a_{11} + a_{12} - b_{11} - b_{12} = 0\ \wedge$
$a_{11} + a_{12} - b_{11} + b_{22} + b_{32} - c_{12} - c_{22} = 0\ \wedge$
$a_{12} + a_{22} - a_{31} - b_{11} - b_{21} + b_{32} + c_{11} - c_{22} = 0\ \wedge$
$a_{12} - a_{31} - b_{11} + b_{32} + c_{11} \geq 0\ \wedge\ a_{31} - b_{32} - c_{11} \leq 0\ \wedge\ a_{31} - b_{31} - b_{32} \leq 0\ \wedge$
$b_{32} \geq 0\ \wedge\ a_{31} + b_{11} - b_{32} - c_{11} \geq 0\ \wedge\ a_{11} + a_{31} - b_{32} - c_{11} \geq 0\ \wedge$
$a_{31} - b_{32} \geq 0)$
$\vee\ (a_{12} - a_{31} - b_{11} + b_{32} + c_{11} - c_{22} \leq 0\ \wedge\ a_{11} + a_{31} - c_{11} - c_{12} \leq 0\ \wedge$
$b_{21} \geq 0\ \wedge\ b_{11} + b_{21} + b_{31} - c_{11} - c_{21} = 0\ \wedge\ a_{31} + a_{32} - b_{31} - b_{32} = 0\ \wedge$
$a_{11} + a_{21} + a_{31} - c_{11} - c_{12} = 0\ \wedge\ a_{11} + a_{12} - b_{11} - b_{12} = 0\ \wedge$
$a_{11} + a_{12} - b_{11} + b_{22} + b_{32} - c_{12} - c_{22} = 0\ \wedge$
$a_{12} + a_{22} - a_{31} - b_{11} - b_{21} + b_{32} + c_{11} - c_{22} = 0\ \wedge\ a_{11} + a_{12} - b_{11} \geq 0\ \wedge$
$a_{11} \geq 0\ \wedge\ a_{11} + b_{31} - c_{11} \geq 0\ \wedge\ a_{11} + a_{31} - c_{11} \geq 0\ \wedge\ a_{11} - b_{11} \leq 0\ \wedge$
$a_{11} + a_{31} - b_{32} - c_{11} \leq 0\ \wedge\ a_{11} - c_{11} \leq 0)$
$\vee\ (a_{12} - a_{31} - b_{11} + b_{32} + c_{11} - c_{22} \leq 0\ \wedge\ a_{11} + a_{31} - c_{11} - c_{12} \leq 0\ \wedge$
$b_{21} \geq 0\ \wedge\ b_{11} + b_{21} + b_{31} - c_{11} - c_{21} = 0\ \wedge\ a_{31} + a_{32} - b_{31} - b_{32} = 0\ \wedge$
$a_{11} + a_{21} + a_{31} - c_{11} - c_{12} = 0\ \wedge\ a_{11} + a_{12} - b_{11} - b_{12} = 0\ \wedge$
$a_{11} + a_{12} - b_{11} + b_{22} + b_{32} - c_{12} - c_{22} = 0\ \wedge$
$a_{12} + a_{22} - a_{31} - b_{11} - b_{21} + b_{32} + c_{11} - c_{22} = 0\ \wedge\ a_{12} \geq 0\ \wedge\ b_{11} \geq 0\ \wedge$
$b_{11} + b_{31} - c_{11} \geq 0\ \wedge\ a_{31} + b_{11} - c_{11} \geq 0\ \wedge\ a_{11} - b_{11} \geq 0\ \wedge$
$a_{31} + b_{11} - b_{32} - c_{11} \leq 0\ \wedge\ b_{11} - c_{11} \leq 0)$
$\vee\ (a_{11} - b_{11} \leq 0\ \wedge\ a_{11} + a_{12} - b_{11} \geq 0\ \wedge\ a_{11} \geq 0\ \wedge$
$b_{11} + b_{21} + b_{31} - c_{11} - c_{21} = 0\ \wedge\ a_{31} + a_{32} - b_{31} - b_{32} = 0\ \wedge$
$a_{11} + a_{21} + a_{31} - c_{11} - c_{12} = 0\ \wedge\ a_{11} + a_{12} - b_{11} - b_{12} = 0\ \wedge$
$a_{11} + a_{12} - b_{11} + b_{22} + b_{32} - c_{12} - c_{22} = 0\ \wedge$
$a_{12} + a_{22} - a_{31} - b_{11} - b_{21} + b_{32} + c_{11} - c_{22} = 0\ \wedge$

$a_{11} + a_{12} - a_{31} - b_{11} + b_{32} - c_{22} \leq 0 \ \wedge \ a_{11} - c_{11} \leq 0 \ \wedge \ b_{31} \geq 0 \ \wedge \ a_{31} \geq 0 \ \wedge$
$a_{31} - c_{12} \leq 0 \ \wedge \ a_{11} + b_{21} - c_{11} \geq 0 \ \wedge \ a_{31} - b_{32} \leq 0)$
$\vee \ (a_{11} - b_{11} \leq 0 \ \wedge \ a_{11} + a_{12} - b_{11} \geq 0 \ \wedge \ a_{11} \geq 0 \ \wedge$
$b_{11} + b_{21} + b_{31} - c_{11} - c_{21} = 0 \ \wedge \ a_{31} + a_{32} - b_{31} - b_{32} = 0 \ \wedge$
$a_{11} + a_{21} + a_{31} - c_{11} - c_{12} = 0 \ \wedge \ a_{11} + a_{12} - b_{11} - b_{12} = 0 \ \wedge$
$a_{11} + a_{12} - b_{11} + b_{22} + b_{32} - c_{12} - c_{22} = 0 \ \wedge$
$a_{12} + a_{22} - a_{31} - b_{11} - b_{21} + b_{32} + c_{11} - c_{22} = 0 \ \wedge$
$a_{11} + a_{12} - b_{11} - c_{22} \leq 0 \ \wedge \ a_{11} + a_{31} - b_{32} - c_{11} \leq 0 \ \wedge \ a_{31} - b_{31} - b_{32} \leq 0 \ \wedge$
$b_{32} \geq 0 \ \wedge \ b_{32} - c_{12} \leq 0 \ \wedge \ a_{11} + a_{31} + b_{21} - b_{32} - c_{11} \geq 0 \ \wedge \ a_{31} - b_{32} \geq 0)$
$\vee \ (a_{11} - b_{11} \leq 0 \ \wedge \ a_{11} + a_{12} - b_{11} \geq 0 \ \wedge \ a_{11} \geq 0 \ \wedge$
$b_{11} + b_{21} + b_{31} - c_{11} - c_{21} = 0 \ \wedge \ a_{31} + a_{32} - b_{31} - b_{32} = 0 \ \wedge$
$a_{11} + a_{21} + a_{31} - c_{11} - c_{12} = 0 \ \wedge \ a_{11} + a_{12} - b_{11} - b_{12} = 0 \ \wedge$
$a_{11} + a_{12} - b_{11} + b_{22} + b_{32} - c_{12} - c_{22} = 0 \ \wedge$
$a_{12} + a_{22} - a_{31} - b_{11} - b_{21} + b_{32} + c_{11} - c_{22} = 0 \ \wedge$
$a_{12} - a_{31} - b_{11} - b_{21} + b_{32} + c_{11} - c_{22} \leq 0 \ \wedge \ b_{21} \geq 0 \ \wedge$
$a_{11} + b_{21} + b_{31} - c_{11} \geq 0 \ \wedge \ a_{11} + a_{31} + b_{21} - c_{11} \geq 0 \ \wedge$
$a_{11} + a_{31} + b_{21} - c_{11} - c_{12} \leq 0 \ \wedge \ a_{11} + a_{31} + b_{21} - b_{32} - c_{11} \leq 0 \ \wedge$
$a_{11} + b_{21} - c_{11} \leq 0)$
$\vee \ (a_{11} - b_{11} \leq 0 \ \wedge \ a_{11} + a_{12} - b_{11} \geq 0 \ \wedge \ a_{11} \geq 0 \ \wedge$
$b_{11} + b_{21} + b_{31} - c_{11} - c_{21} = 0 \ \wedge \ a_{31} + a_{32} - b_{31} - b_{32} = 0 \ \wedge$
$a_{11} + a_{21} + a_{31} - c_{11} - c_{12} = 0 \ \wedge \ a_{11} + a_{12} - b_{11} - b_{12} = 0 \ \wedge$
$a_{11} + a_{12} - b_{11} + b_{22} + b_{32} - c_{12} - c_{22} = 0 \ \wedge$
$a_{12} + a_{22} - a_{31} - b_{11} - b_{21} + b_{32} + c_{11} - c_{22} = 0 \ \wedge$
$a_{11} + a_{12} - b_{11} + b_{32} - c_{12} - c_{22} \leq 0 \ \wedge \ a_{11} + a_{31} - c_{11} - c_{12} \leq 0 \ \wedge$
$a_{31} - b_{31} - c_{12} \leq 0 \ \wedge \ c_{12} \geq 0 \ \wedge \ a_{11} + a_{31} + b_{21} - c_{11} - c_{12} \geq 0 \ \wedge$
$b_{32} - c_{12} \geq 0 \ \wedge \ a_{31} - c_{12} \geq 0)$
$)$

Time: 1428 ms

Example 5.

This example, AFIRO, is one of a series of benchmark examples for linear programming provided to us by ZIB, Berlin.

Input:

$\exists vx_{01} \exists vx_{02} \exists vx_{03} \exists vx_{04} \exists vx_{06} \exists vx_{07} \exists vx_{08} \exists vx_{09} \exists vx_{10} \exists vx_{11}$
$\exists vx_{12} \exists vx_{13} \exists vx_{14} \exists vx_{15} \exists vx_{16} \exists vx_{22} \exists vx_{23} \exists vx_{24} \exists vx_{25} \exists vx_{26}$
$\exists vx_{28} \exists vx_{29} \exists vx_{30} \exists vx_{31} \exists vx_{32} \exists vx_{33} \exists vx_{34} \exists vx_{35} \exists vx_{36} \exists vx_{37}$
$\exists vx_{38} \exists vx_{39} (vx_{01} \geq 0 \ \wedge \ vx_{02} \geq 0 \ \wedge \ vx_{03} \geq 0 \ \wedge \ vx_{04} \geq 0 \ \wedge \ vx_{06} \geq 0 \ \wedge$
$vx_{07} \geq 0 \ \wedge \ vx_{08} \geq 0 \ \wedge \ vx_{09} \geq 0 \ \wedge \ vx_{10} \geq 0 \ \wedge \ vx_{11} \geq 0 \ \wedge \ vx_{12} \geq 0 \ \wedge \ vx_{13} \geq 0 \ \wedge$

$vx_{14} \geq 0 \ \wedge \ vx_{15} \geq 0 \ \wedge \ vx_{16} \geq 0 \ \wedge \ vx_{22} \geq 0 \ \wedge \ vx_{23} \geq 0 \ \wedge \ vx_{24} \geq 0 \ \wedge \ vx_{25} \geq 0 \ \wedge$
$vx_{26} \geq 0 \ \wedge \ vx_{28} \geq 0 \ \wedge \ vx_{29} \geq 0 \ \wedge \ vx_{30} \geq 0 \ \wedge \ vx_{31} \geq 0 \ \wedge \ vx_{32} \geq 0 \ \wedge \ vx_{33} \geq 0 \ \wedge$
$vx_{34} \geq 0 \ \wedge \ vx_{35} \geq 0 \ \wedge \ vx_{36} \geq 0 \ \wedge \ vx_{37} \geq 0 \ \wedge \ vx_{38} \geq 0 \ \wedge \ vx_{39} \geq 0 \ \wedge$
$-vx_{01} + vx_{02} + vx_{03} = 0 \ \wedge \ -53vx_{01} + 50vx_{04} = 0 \ \wedge \ vx_{01} - 80 \leq 0 \ \wedge$
$-5vx_{02} + 7vx_{14} \leq 0 \ \wedge \ -vx_{06} - vx_{07} - vx_{08} - vx_{09} + vx_{14} + vx_{15} = 0 \ \wedge$
$-53vx_{06} - 53vx_{07} - 48vx_{08} - 43vx_{09} + 50vx_{16} = 0 \ \wedge$
$vx_{06} - vx_{10} - 80 \leq 0 \ \wedge \ vx_{07} - vx_{11} \leq 0 \ \wedge \ vx_{08} - vx_{12} \leq 0 \ \wedge \ vx_{09} - vx_{13} \leq 0$
$\wedge \ -vx_{22} + vx_{23} + vx_{24} + vx_{25} = 0 \ \wedge \ -43vx_{22} + 100vx_{26} = 0 \ \wedge$
$vx_{22} - 500 \leq 0 \ \wedge \ -5vx_{23} + 7vx_{36} \leq 0 \ \wedge$
$-43vx_{28} - 43vx_{29} - 39vx_{30} - 37vx_{31} + 100vx_{38} = 0 \ \wedge$
$vx_{28} + vx_{29} + vx_{30} + vx_{31} - vx_{36} + vx_{37} + vx_{39} - 44 = 0 \ \wedge$
$vx_{28} - vx_{32} - 500 \leq 0 \ \wedge \ vx_{29} - vx_{33} \leq 0 \ \wedge \ vx_{30} - vx_{34} \leq 0 \ \wedge$
$vx_{31} - vx_{35} \leq 0 \ \wedge$
$2364vx_{10} + 2386vx_{11} + 2408vx_{12} + 2429vx_{13} - 1000vx_{25} + 2191vx_{32}$
$+2219vx_{33} + 2249vx_{34} + 2279vx_{35} \leq 0$
$\wedge \ -1000vx_{03} + 109vx_{22} \leq 0 \ \wedge$
$-1000vx_{15} + 109vx_{28} + 108vx_{29} + 108vx_{30} + 107vx_{31} \leq 0 \ \wedge$
$301vx_{01} - 1000vx_{24} \leq 0 \ \wedge$
$301vx_{06} + 313vx_{07} + 313vx_{08} + 326vx_{09} - 1000vx_{37} \leq 0 \ \wedge$
$vx_{04} + vx_{26} - 310 \leq 0 \ \wedge \ vx_{16} + vx_{38} - 300 \leq 0 \ \wedge$
$10vx_{02} + 8vx_{14} + 15vx_{23} + 12vx_{36} - 250vx_{39} + 25z \geq 0)$

$$\frac{-10vx_{02} - 8vx_{14} - 15vx_{23} - 12vx_{36} + 250vx_{39}}{25}$$

Output:

$\{\{(875z + 406659 \geq 0 \ \wedge \ 70000z + 31917303 \leq 0), \{z = \frac{-406659}{875}, vx_{01} = 80,$
$vx_{22} = 500, vx_{23} = \frac{11898}{25}, vx_{06} = 80, vx_{09} = 0, vx_{08} = 0, vx_{07} = 0, vx_{10} = 0,$
$vx_{11} = 0, vx_{12} = 0, vx_{14} = \frac{255}{14}, vx_{02} = \frac{51}{2}, vx_{24} = \frac{602}{25}, vx_{13} = 0, vx_{28} = 0,$
$vx_{31} = 0, vx_{32} = 0, vx_{29} = 0, vx_{33} = 0, vx_{30} = 0, vx_{34} = 0, vx_{35} = 0,$
$vx_{36} = \frac{11898}{35}, vx_{37} = \frac{13438}{35}, vx_{03} = \frac{109}{2}, vx_{04} = \frac{424}{5}, vx_{15} = \frac{865}{14},$
$vx_{16} = \frac{424}{5}, vx_{25} = 0, vx_{26} = 215, vx_{38} = 0, vx_{39} = 0\}, \frac{-406659}{875}\}\}$

Time: 762025 ms plus GC time: 8194 ms

The remaining examples concern a quadratic objective function.

Example 6.

Find the minimum point of $q = 3x^2+3y^2-2xy$ under the constraints $P = \{-x-y+u \geq 0,\ x-v \geq 0,\ y-v \geq 0\}$ in dependence of the parameters. The graph of q is a parabolic ellipsoid centered at the origin with main axes of the ellipse-sections along the diagonals. The constraints describe a triangle with sides parallel to the coordinate axes and to the second diagonal.

Input:

$\exists x \exists y (u - x - y \geq 0 \ \wedge \ -v + x \geq 0 \ \wedge \ -v + y \geq 0)$

$3x^2 - 2xy + 3y^2$

Output:

$$\begin{aligned}
&\{\{u - 2v \geq 0, \{x = v, y = v\}, 4v^2\},\\
&\{u - 2v \geq 0, \{x = u - v, y = v\}, 3u^2 - 8uv + 8v^2\},\\
&\{u - 2v \geq 0, \{x = \frac{u}{2}, y = \frac{u}{2}\}, u^2\},\\
&\{(u - 4v \geq 0 \ \wedge \ v \geq 0), \{x = 3v, y = v\}, 24v^2\},\\
&\{(3u - 4v \geq 0 \ \wedge \ v \leq 0), \{x = v, y = \frac{v}{3}\}, \frac{8v^2}{3}\},\\
&\{(3u - 4v \geq 0 \ \wedge \ u - 4v \geq 0), \{x = \frac{3u}{4}, y = \frac{u}{4}\}, \frac{3u^2}{2}\},\\
&\{(u \geq 0 \ \wedge \ v \leq 0), \{x = 0, y = 0\}, 0\}\}
\end{aligned}$$

Time: 85 ms

Example 7.

This example is taken from [Gabasov & Kirillova].

Input:

$\exists x_1 \exists x_2 \exists x_3 \exists x_4 (2x_1 + x_2 + x_3 = 6 \ \wedge \ -x_1 + x_2 + x_4 = 5 \ \wedge \ x_1 \geq 0 \ \wedge \ x_2 \geq 0 \ \wedge$
$x_3 \geq 0 \ \wedge \ x_4 \geq 0)$

$$\frac{x_1^2 - 2x_1x_2 - 6x_1 + 2x_2^2 - 8x_2}{2}$$

Output:

$$\{\{true, \{x_1 = \frac{25}{13}, x_2 = \frac{28}{13}, x_3 = 0, x_4 = \frac{62}{13}\}, \frac{-313}{26}\}\}$$

Time: 68 ms

Example 8.

This example is taken from [Murty].

Input:

$$\exists y_1 \exists y_2 \exists y_3 \exists y_4 (y_1 + y_2 + y_3 + y_4 = 1 \ \wedge \ y_1 \geq 0 \ \wedge \ y_2 \geq 0 \ \wedge \ y_3 \geq 0 \ \wedge \ y_4 \geq 0)$$

$10y_1^2 + 8y_1y_2 + 22y_1y_3 + 34y_1y_4 + 10y_1 + 16y_2^2 + 40y_2y_3 + 40y_2y_4 + 16y_2 + 29y_3^2 + 66y_3y_4 + 24y_3 + 41y_4^2 + 28y_4 + 5$

Output:

$$\{\{true, \{y_1 = \frac{5}{6}, y_2 = \frac{1}{6}, y_3 = 0, y_4 = 0\}, \frac{49}{2}\}\}$$

Time: 68 ms

Example 9.

This example is taken from [Gottfried & Weisman].

Input:

$$\exists x_1 \exists x_2 \exists x_3 (2x_1 + x_2 + x_3 \geq 20 \ \wedge \ x_1 + x_3 \geq 10 \ \wedge \ x_1 \geq 0 \ \wedge \ x_2 \geq 0 \ \wedge \ x_3 \geq 0)$$

$$\frac{2x_1^2 + 6x_2^2 + 3x_3}{2}$$

Output:

$\{\{true, \{x_1 = \frac{3}{2}, x_2 = \frac{1}{4}, x_3 = \frac{67}{4}\}, \frac{441}{16}\}\}$

Time: 34 ms

Example 10.

This example is taken from [Foulds].

Input:

$\exists x_1 \exists x_2 (4x_1 + 5x_2 \leq 20 \ \wedge \ x_1 \geq 0 \ \wedge \ x_2 \geq 0)$

$x_1^2 + x_1x_2 - 3x_1 + x_2^2 - 2x_2$

Output:

$\{\{true, \{x_1 = \frac{4}{3}, x_2 = \frac{1}{3}\}, \frac{-7}{3}\}\}$

Time: 17 ms

References

[Borgwardt] K.H. Borgwardt, The Simplex Method, Algorithms and Combinatorics 1, Springer Verlag, 1987.

[Černikov] R.N. Černikov, The Solution of Linear Programming Problems by Elimination of Unknowns, Doklady Akademii Nauk 139 (1961), pp. 1314-1317. [Translation in: Soviet Mathematics Doklady 2 (1961), pp. 1099-1103.]

[Chandru] V. Chandru, Variable elimination in linear constraints, The Computer Journal 36 (1993), pp. 463-472.

[Chvátal] V. Chvátal, Linear Programming, A Series of Books in the Mathematical Sciences, Victor Klee, ed., W.H. Freeman and Company, New York/San Francisco, 1983.

[Collins & Hong] G.E. Collins, H. Hong, Partial cylindrical algebraic decomposition for quantifier elimination, J. Symb. Comp. 12 (1991), pp. 299-328.

[Dantzig] G. B. Dantzig, Lineare Programmierung und Erweiterungen, Springer-Verlag, 1966.

[Dantzig & Eaves] G.B. Dantzig, B.C. Eaves, Fourier- Motzkin Elimination and its Dual, Journal of Combinatorial Theory, Series A 14 (1973), pp. 288-297.

[Dines] L.L. Dines, Systems of linear inequalities, Annals of Mathematics, 20 (1919), pp. 191-199.

[Eaves & Rothblum] B.C. Eaves, U.G. Rothblum, A theory on extending algorithms for parametric problems, Mathematics of Operations Research, **Vol. 14, No.3**, August 1989, USA, pp. 502–533.

[Eaves & Rothblum] B.C. Eaves, U.G. Rothblum, Dines-Fourier-Motzkin quantifier elimination and an application of corresponding transfer principles over ordered fields, Mathematical Programming **53**, North-Holland, 1992, pp. 307-321.

[Eaves & Rothblum] B.C. Eaves, U.G. Rothblum, Formulation of linear problems and solution by a universal machine, Mathematical Programming, to appear.

[Fourier] J.B.J. Fourier, 1826-1: Solution d'une question particulière du calcul des inégalités, 1826, and excerpts from Histoire de l'Académie, 1823, 1824, Oeuvres II, pp. 317-328.

[Foulds] L.R. Foulds, Optimization Techniques, Springer-Verlag, New York–Heidelberg–Berlin.

[Hong 1] H. Hong, Quantifier elimination for formulas constrained by quadratic equation, The Computer Journal 36, 5 (1993), pp. 440-449.

[Hong 2] H. Hong, Simple solution formula construction in cylindrical algebraic decomposition based quantifier elimination, in ISSAC'92, P. S. Wang editor, ACM 1992, pp. 177 - 188.

[Gabasov & Kirillova] R.F. Gabasov, F.M. Kirillova, Methods of optimization, Optimization Software, Inc. Publications Division, New York.

[Gottfried & Weisman] B.S. Gottfried, J. Weisman, Introduction to optimization theory, Prentice-Hall, Inc., Englewood Cliffs, New Jersey.

[Jeroslow] R.G. Jeroslow, Asymptotic linear programming, Operations Research **21**, 1973, pp. 1128–1141.

[Jeroslow] R.G. Jeroslow, Linear programs dependent on a single parameter, Discrete Mathematics **6**, 1973, North-Holland, pp. 119–140.

[Kappert] M. Kappert, Diplomarbeit, Universität Passau, in preparation.

[Loos & Weispfenning] R. Loos, V. Weispfenning, Applying linear quantifier elimination, The Computer Journal 36 (1993), pp. 450-462.

[Meisinger] G. Meisiger, Universität Passau, Private communication, 1994.

[Melenk] H. Melenk, ZIB, Berlin, Private communication, 1993.

[Motzkin] T.S. Motzkin, 1936-1: Beiträge zur Theorie der linearen Ungleichungen, Dissertation Universität Zürich,

[Murty] K.G. Murty, Linear complementarity, Linear and nonlinear programming, Heldermann Verlag, Berlin.

[Mattheiss & Rubin] T.H. Mattheiss, D.S. Rubin, A survey and comparison of methods for finding all vertices of convex polyhedral sets, Math. Oper. Res. 5 (1980), pp. 167-185.

[Schrijver] A. Schrijver, Theory of linear and integer programming, Wiley, Chichester, 1986.

[Seidel] R. Seidel, Constructing higher-dimensional convex hulls at logarithmic cost per face, Proc. 18th Ann. ACM Symposium Theory Comput. (1986), pp. 404-413.

[Sturm] Th. Sturm, REDUCE-implementation of the algorithms in [Loos & Weispfenning], Passau, 1993.

[Weispfenning 1] V. Weispfenning, The complexity of linear problems in fields, J. Symb. Comp. 5 (1988), pp. 3-27.

[Weispfenning 2] V. Weispfenning, A new approach to quantifier elimination for real algebra, Proc. Collins Symposium Linz Oct. 1993, to appear.

[Weispfenning 3] V. Weispfenning, Aspects of quantifier elimination in algebra, in: Universal Algebra and its links with logic ..., Proc. 25, Arbeitstagung Darmstadt 1983, Heldermann V., Berlin, pp. 85-105.

[Williams] H.P. Williams, Fourier's method of linear programming and its dual, American Mathematical Monthly 93 (1986),), pp. 681-695.