



A Dynamic Algorithm for Constructing the Dual Representation of a Polyhedral Cone

Sergey O. Semenov  and Nikolai Yu. Zolotykh 

Lobachevsky State University of Nizhny Novgorod,
Gagarin Avenue 23, Nizhny Novgorod 603600, Russia
{sergey.semenov,nikolai.zolotykh}@itmm.unn.ru

Abstract. We propose a dynamic version of the double description method for generating the extreme rays of a polyhedral cone. The dynamic version of the algorithm supports online input of inequalities. Some modifications of the method were implemented and the results of computational experiments are presented. On a series of problems, our implementation of the algorithm showed higher performance results in comparison with the known analogues.

Keywords: System of linear inequalities · Convex hull · Cone · Polyhedron · Double description method

1 Introduction

It is well-known that any convex polyhedron $P \subseteq F^d$, where F is an ordered field, can be represented in any of the following two ways:

- (1) as the set $P = \{x \in F^d : Ax \leq b\}$ of solutions to a system of linear inequalities, where $A \in F^{m \times d}$, $b \in F^m$ (*facet description*);
- (2) as the sum of the conical hull of a set of vectors v_1, \dots, v_s in F^d and the convex hull of a set of points w_1, \dots, w_n in F^d (*vertex description*).

The problem of finding the representation (1) given the representation (2) is called *the convex hull problem*. According to the classical theorem of Weyl this problem is equivalent (dual) to the problem of constructing the representation (2) given the representation (1). These two problems are referred to as *finding the dual representation of a polyhedron*.

The problem of constructing the dual representation of a convex polyhedron plays a central role in the theory of systems of linear inequalities and computational geometry [11, 21]. For some applications, the representation (1) is convenient, while in other cases the representation (2) is more usable, therefore, it is important to quickly move from one description to another. The importance

This work was supported by the Russian Science Foundation Grant No. 17-11-01336.

of studying this problem is also emphasized by the fact that it has a variety of applications, the most known of which are linear and integer programming [18], combinatorial optimization [19, 21] and global optimization [15]. Some of the newer applications are biological kinetics [20], analysis and verification of software and hardware [5], identification of dynamic systems [13] and computer algebra [17, 22].

From an algorithmic or theoretical points of view it is convenient to consider these problems only for polyhedral cones. Any polyhedral cone $C \subseteq F^d$ can be represented in two equivalent ways:

- (1) as the set $C = \{x \in F^d : Ax \geq 0\}$ of solutions to a homogeneous system of linear inequalities, where $A \in F^{m \times d}$, or
- (2) as the conical hull of a set of vectors v_1, \dots, v_s .

There is a standard method for reducing the problem of finding the dual representation for convex polyhedra to the corresponding one for polyhedral cones. For example, in order to find representation (2) for a polyhedron $P = \{x \in F^d : Ax \leq b\}$ it is sufficient to solve the corresponding problem for the polyhedral cone $\{x = (x_0, x_1, \dots, x_d) \in F^{d+1} : bx_0 - Ax \geq 0, x_0 \geq 0\}$, and then set $x_0 = 1$ (see Section 1.5 in [21]).

There are several known algorithms for solving problems of finding the dual representation. One of the most popular ones is *the double description method* (DDM) [16], also known as the Motzkin–Burger algorithm [11] or Chernikova’s algorithm [12]. The double description method generally outperforms the other algorithms when applied to degenerate inputs and/or outputs [3].

There are multiple known programs implementing various modifications of the double description method. Among the most well-known are:

- cdd [14] (www.inf.ethz.ch/personal/fukudak/cdd_home);
- SKELETON [23] (www.uic.unn.ru/~zny/skeleton);
- QSkeleton [9] (github.com/sbastrakov/qskeleton);
- Parma Polyhedra Library [5] (bugseng.com/products/ppl).

Implementations of other algorithms solving the given problem should also be noted:

- QHull [6] (www.qhull.org);
- lrs [2, 4] (cgm.cs.mcgill.ca/~avis/C/lrs.html);
- pd [10] (www.cs.unb.ca/~bremner/software/pd).

In this paper we consider the *dynamic* problem of finding the dual representation. This problem appears in many of the applications listed above. For definiteness, we will deal with the problem of constructing a description (2) if the description (1) is given. In dynamic problem the full list of constraints is not known in advance and the constraints come to the input of the algorithm online as the computation proceeds, and the dual description must be computed at each iteration for the current system of linear inequalities. Such framework does not allow the use of many of the heuristics proposed by various authors

(see the references above) to accelerate the algorithm. Nevertheless, we propose such an algorithm (as a version of the double description method) for the dynamic problem, and our algorithm is usually not inferior in performance to other offline algorithms.

The problem of efficient removal of constraints from the double description is considered in [1, 7]. Other issues concerning the dynamic problem are studied in [8].

2 Preliminaries

The material in this section is based on [11, 18, 21]. A *polyhedral cone*, or simply a *cone*, in F^d is defined as a set

$$C = \{x \in F^d : Ax \geq 0\},$$

where F is an ordered field, $A \in F^{m \times d}$. The system of linear inequalities $Ax \geq 0$ is said to *define* the cone C . A cone is called *pointed*, if it contains no zero subspaces. It is well-known that for a cone to be pointed it is necessary and sufficient that $\text{rank } A = d$, where $\text{rank } A$ denotes the rank of matrix A . Any polyhedral cone C can be defined as the conical hull of a finite set of vectors v_1, v_2, \dots, v_s in F^d , i. e.

$$C = \{x = \alpha_1 v_1 + \alpha_2 v_2 + \dots + \alpha_s v_s : \alpha_i \geq 0 \ (i = 1, \dots, s)\}.$$

By writing vectors v_1, v_2, \dots, v_s as rows of matrix $V \in F^{s \times d}$ the conical hull can be defined as

$$C = \{x = \alpha V, \alpha \in F^s, \alpha \geq 0\},$$

where α is a row vector. The set of vectors v_1, \dots, v_s are said to *generate* the cone C .

A non-zero vector $u \in C$ is referred to as a *ray* of the cone C . Two rays u and v are *equal* (written as $u \simeq v$) if for some $\alpha > 0$ it is true that $u = \alpha v$. A ray $u \in C$ is said to be *extreme* if the condition $u = \alpha v + \beta w$, where $\alpha \geq 0, \beta \geq 0$ and $v, w \in C$ implies $u \simeq v \simeq w$. Suppose that P is a convex subset of F^d , and for some $a \in F^d, \alpha \in F$, it holds that $P \subseteq \{x : ax \leq \alpha\}$. Then $P \cap \{x : ax = \alpha\}$ is called a *face* of the set P . Two different extreme rays u and v of a pointed cone C are said to be *adjacent*, if no minimal face containing both rays contains any other extreme rays of the cone C .

The problem of constructing the set of vectors generating polyhedral cone $C = \{x \in F^d : Ax \geq 0\}$ is reduced to finding the extreme generators of a pointed cone by transition to the orthogonal complement L^\perp of the maximal subspace $L = \{x \in F^d : Ax = 0\}$ contained inside C . Unfortunately, in our study we cannot take advantage of this fact (as is often the case), since in the process of adding new inequalities, the space L may change.

Notations: $I_{d \times d}$ is the identity matrix of size $d \times d$, $O_{s \times m}$ is the zero matrix of size $s \times m$, A_i is the i -th row of matrix A , A_{ij} is the element from the i -th row and j -th column of matrix A .

3 Dynamic Double Description Method

The dynamic variant of the double description method is based on the regular double description method [11]. The procedure takes a matrix $A \in F^{m \times d}$ as its input. The output is matrices $U \in F^{t \times d}$ and $V \in F^{s \times d}$, the rows of which compose a basis of maximal subspace $L = \{x \in F^d : Ax = 0\}$ and an irreducible set of vectors generating the cone $C = \{x \in F^d : Ax \geq 0\}$ respectively.

procedure DDM-DYN(A)

Input: $A \in F^{m \times d}$

Output: the basis U of maximal subspace $L = \{x \in F^d : Ax = 0\}$

and the set of vectors V generating the cone $C = \{x \in F^d : Ax \geq 0\}$

$U \leftarrow I_{d \times d}$

$V \leftarrow O_{0 \times d}$

$Q \leftarrow O_{0 \times d}$

for $i = 1, 2, \dots, m$

$p \leftarrow U \cdot A_i^\top$

$q \leftarrow V \cdot A_i^\top$

if $p = 0$

insert the column $\text{BOOL}(q)$ into Q

$J_+ \leftarrow \{j : q_j > 0\}$

$J_- \leftarrow \{j : q_j < 0\}$

$V_{\text{new}} \leftarrow O_{0 \times d}$

$Q_{\text{new}} \leftarrow O_{0 \times m}$

$E \leftarrow \text{ADJACENT}(Q, J_+, J_-)$

for $\{j_1, j_2\} \in E$

append the row $\text{NORMALIZE}(q_{j_1}V_{j_2} - q_{j_2}V_{j_1})$ to V_{new}

append the row $\text{NORMALIZE}(Q_{j_1} \vee Q_{j_2})$ to Q_{new}

append the zero column to Q_{new}

remove the rows J_- from V and from Q

append the rows of V_{new} to V and the rows of Q_{new} to Q

else

find j_0 with $p_{j_0} \neq 0$

if $p_{j_0} < 0$

$U_{j_0} \leftarrow -U_{j_0}$

$p_{j_0} \leftarrow -p_{j_0}$

for $j = 1, 2, \dots, d$

if $j \neq j_0$ **and** $p_j \neq 0$

$U_j \leftarrow \text{NORMALIZE}(p_{j_0}U_j - p_jU_{j_0})$

for $j = 1, 2, \dots, s$

if $q_j \neq 0$

$V_j \leftarrow \text{NORMALIZE}(p_{j_0}V_j - q_jU_{j_0})$

insert the row U_{j_0} into V

remove the j_0 -th row from U

insert the zero column into Q

insert the $(0, 0, \dots, 0, 1)$ row into Q

The normalization function NORMALIZE used in the algorithm is necessary to prevent unlimited growth of elements. Its implementation may vary, the one used in this study for integer calculations divides each element of the input vector by their greatest common divisor. The ADJACENT function checks a pair of extreme rays for adjacency.

Typically modifications of the double description method alter some of the following three parameters:

- (1) the order in which the rows of the primal representation are considered
- (2) the test for adjacency of two extreme rays
- (3) the moment when the extreme rays are tested for adjacency.

Since the dynamic version of the double description method has to support online input of inequalities, only modifications of the latter two aspects are applicable to this algorithm. Two versions of adjacency test were implemented: the combinatorial test [16], and the graph test [23].

```

function ADJACENT.COMBINATORIAL( $Q, J_1, J_2$ )
  Input:  $Q \in F^{s \times m}$ 
            $J_1, J_2 \subseteq \{1, 2, \dots, s\}$ 
  Output:  $E = \{\{j_1, j_2\} : j_1 \in J_1, j_2 \in J_2, j_1 \text{ and } j_2 \text{ - adjacent rays}\}$ 
            $E = \emptyset$ 
  for  $j_1 \in J_1$ 
    for  $j_2 \in J_2$ 
       $Z \leftarrow \{\ell : Q_{j_1 \ell} = 0 \wedge Q_{j_2 \ell} = 0\}$ 
      if  $|Z| \geq r - 2$ 
        if  $\forall k = 1, 2, \dots, s, k \neq j_1 \wedge k \neq j_2 \exists \ell \in Z : Q_{k \ell} = 1$ 
           $E \leftarrow E \cup \{\{j_1, j_2\}\}$ 
  return  $E$ 

```

```

function ADJACENT.GRAPH( $Q, J_1, J_2$ )
  Input:  $Q \in F^{s \times m}$ 
            $J_1, J_2 \subseteq \{1, 2, \dots, s\}$ 
  Output:  $E = \{\{j_1, j_2\} : j_1 \in J_1, j_2 \in J_2, j_1 \text{ and } j_2 \text{ are adjacent rays}\}$ 
            $E \leftarrow \emptyset$ 
  for  $j_1 \in J_1$ 
     $D \leftarrow \emptyset$ 
    for  $j \in \{1, 2, \dots, m\}$ 
       $Z \leftarrow \{\ell : Q_{j_1 \ell} = 0 \wedge Q_{j \ell} = 0\}$ 
      if  $|Z| \geq r - 2$ 
         $D \leftarrow D \cup \{j\}$ 
    for  $j_2 \in D \cap J_2$ 
      if  $\forall k = 1, 2, \dots, s, k \neq j_1 \wedge k \neq j_2 \exists \ell \in Z : Q_{k \ell} = 1$ 
         $E \leftarrow E \cup \{\{j_1, j_2\}\}$ 
  return  $E$ 

```

Some modifications of the double description method were proposed (e.g. [14]) where the set of adjacent extreme rays is maintained and rebuilt immediately after the list of extreme rays is updated. Instead of iterating over all pairs of rays (u, v) for which $uA_i^T \cdot vA_i^T < 0$ to generate new extreme rays the algorithm iterates over all pairs of adjacent extreme rays. Below is an adaptation of such a modification for the dynamic version of the algorithm which will be referred to as M1.

procedure DDM-DYN.M1(A)

Input: $A \in F^{m \times d}$

Output: the basis U of maximal subspace $L = \{x \in F^d : Ax = 0\}$

and the set of vectors V generating the cone $C = \{x \in F^d : Ax \geq 0\}$

$U \leftarrow I_{d \times d}$

$V \leftarrow O_{0 \times d}$

$Q \leftarrow O_{0 \times d}$

$E \leftarrow \emptyset$

for $i = 1, 2, \dots, m$

$p \leftarrow U \cdot A_i^T$

$q \leftarrow V \cdot A_i^T$

if $p = 0$

insert the column $\text{BOOL}(q)$ into Q

$E_{\text{new}} \leftarrow \emptyset$

$E_{\text{old}} \leftarrow \emptyset$

for $\{j_1, j_2\} \in E$

if $q_{j_1} > 0 \wedge q_{j_2} < 0$

append row $\text{NORMALIZE}(q_{j_1}V_{j_2} - q_{j_2}V_{j_1})$ to V_{new}

append row $\text{NORMALIZE}(Q_{j_1} \vee Q_{j_2})$ to Q_{new}

$j \leftarrow$ index of the new rows after their insertion

$E_{\text{new}} \leftarrow E_{\text{new}} \cup \{\{j_1, j\}\}$

$E_{\text{old}} \leftarrow E_{\text{old}} \cup \{\{j_1, j_2\}\}$

else if $q_{j_1} < 0 \wedge q_{j_2} > 0$

append row $\text{NORMALIZE}(q_{j_2}V_{j_1} - q_{j_1}V_{j_2})$ to V_{new}

append row $\text{NORMALIZE}(Q_{j_1} \vee Q_{j_2})$ to Q_{new}

$j \leftarrow$ index of the new rows after their insertion

$E_{\text{new}} \leftarrow E_{\text{new}} \cup \{\{j, j_2\}\}$

$E_{\text{old}} \leftarrow E_{\text{old}} \cup \{\{j_1, j_2\}\}$

else if $q_{j_1} \leq 0 \wedge q_{j_2} \leq 0$

$E_{\text{old}} \leftarrow E_{\text{old}} \cup \{\{j_1, j_2\}\}$

$E \leftarrow E \cup E_{\text{new}} \setminus E_{\text{old}}$

$J_- \leftarrow \{j : q_j < 0\}$

$J_{\pm} \leftarrow \{j : q_j = 0\}$

remove rows J_- from V and from Q

update row indices in E and J_{\pm}

append rows of V_{new} to V and insert their new indices into J_{\pm}

append rows of Q_{new} to Q

$E \leftarrow E \cup \text{ADJACENT.M1}(Q, J_{\pm})$

```

else
  find  $j_0$  with  $p_{j_0} \neq 0$ 
  if  $p_{j_0} < 0$ 
     $U_{j_0} \leftarrow -U_{j_0}$ 
     $p_{j_0} \leftarrow -p_{j_0}$ 
  for  $j = 1, 2, \dots, d$ 
    if  $j \neq j_0$  and  $p_j \neq 0$ 
       $U_j \leftarrow \text{NORMALIZE}(p_{j_0}U_j - p_jU_{j_0})$ 
  for  $j = 1, 2, \dots, s$ 
    if  $q_j \neq 0$ 
       $V_j \leftarrow \text{NORMALIZE}(p_{j_0}V_j - q_jU_{j_0})$ 
  append row  $U_{j_0}$  to  $V$ ,  $j \leftarrow$  its new index
   $E \leftarrow E \cup \{\{j', j\} : j' \in \{1, 2, \dots, j-1\}\}$ 
  remove row  $j_0$  from  $U$ 
  append the zero column to  $Q$ 
  append the  $(0, 0, \dots, 0, 1)$  row into  $Q$ 

```

ADJACENT.M1(Q, J_{\pm}) is a simple modification of ADJACENT(Q, J_1, J_2) that iterates over $\{\{j_1, j_2\} : j_1 \in J_{\pm}, j_2 \in J_{\pm}, j_1 \neq j_2\}$ rather than $\{(j_1, j_2) : j_1 \in J_1, j_2 \in J_2\}$.

4 Computational Results

A C++ implementation of the dynamic double description method and its modifications presented above has been developed. The computational experiments were performed on a computer with Intel(R) Core(TM) i7-8700K CPU at 3.70 GHz, Microsoft Windows 10 operating system, using the Microsoft Visual Studio 2017 compiler. The experiments were run using the problem instances described in [14].

Tables 1 and 2 present the performance comparison of DDM-DYN, its modification DDM-DYN.M1 and SKELETON [23], with/without its PlusPlus modification. Since computation time depends significantly on the order in which the rows of the primal representation are considered and which is fixed in the case of the dynamic algorithm, SKELETON was used with the minindex order of consideration. Note that the PlusPlus modification of SKELETON reduces the number of adjacency checks by relying on the fact that the entire primal representation is known ahead of time and, therefore, it cannot be adopted for use with online input of inequalities.

Figures 1, 2 and 3 demonstrate the dependence of the number of adjacency checks on the number of iteration made by DDM-DYN and DDM-DYN.M1 on cube18, mit729-9 and ccc7 problems. The number of the checks varies from one problem instance to another and heavily impacts total computation time.

Table 1. Performance comparison of DDM-DYN and SKELETON, with combinatorial adjacency test (s)

Problem	Input	Output	DDM-DYN	DDM -DYN.M1	SKELETON	SKELETON, PlusPlus
cube16	32×17	65536×17	3.512	7.210	12.199	3.354
cube18	36×19	262144×19	45.184	103.514	207.037	27.069
mit729-9	729×9	4862×9	164.727	120.745	274.795	255.393
ccc7	63×22	38780×22	14793.694	15413.77	16016.7	3437.64

Table 2. Performance comparison of DDM-DYN and SKELETON, with graph adjacency test (s)

Problem	Input	Output	DDM-DYN	DDM -DYN.M1	SKELETON	SKELETON, PlusPlus
cube16	32×17	65536×17	4.316	4.961	4.777	4.489
cube18	36×19	262144×19	48.215	54.552	50.698	48.498
mit729-9	729×9	4862×9	874.575	106.253	289.037	258.991
ccc7	63×22	38780×22	>5 h	2988.16	3041.86	1854.63

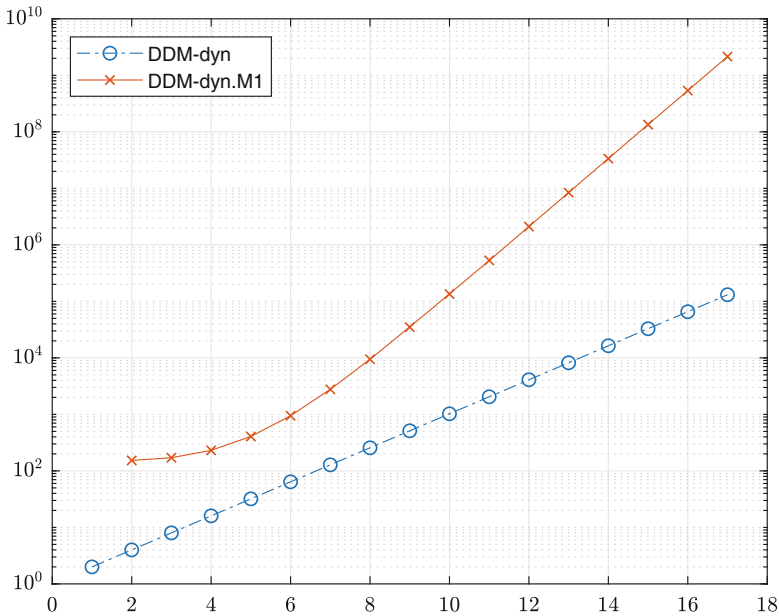


Fig. 1. Number of adjacency checks on cube18

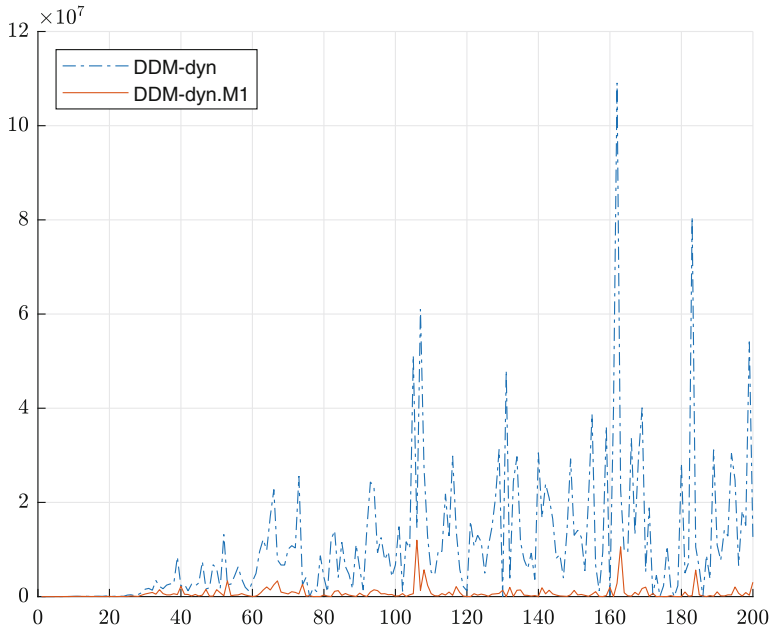


Fig. 2. Number of adjacency checks on mit729-9 (first 200 iterations)

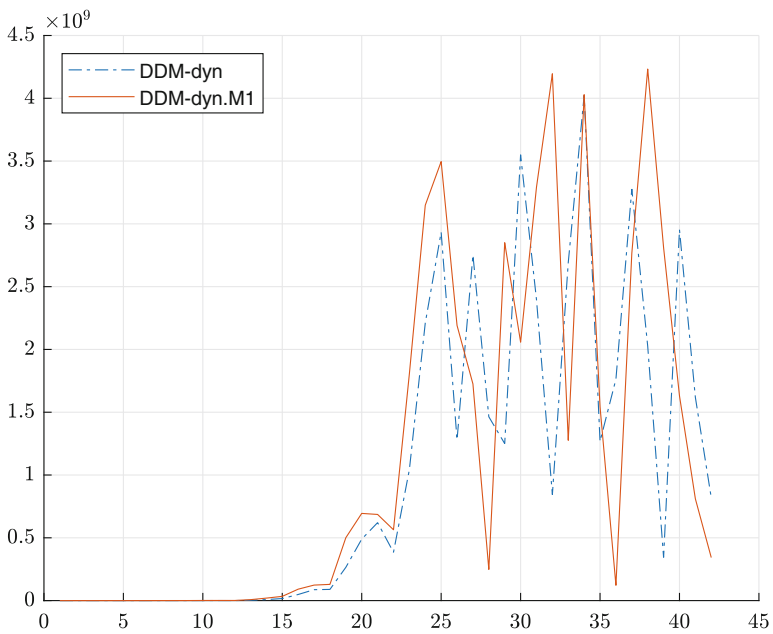


Fig. 3. Number of adjacency checks on ccc7

5 Conclusion

A dynamic version of the double description method for finding extreme rays of a polyhedral cone with online input of inequalities has been proposed. Two known modifications of the algorithm (graph adjacency test and maintaining the edge set to generate new extreme rays) have been adopted for its dynamic variant and tested on multiple problem instances. The results of computational experiments demonstrate better performance with certain configurations than that of SKELETON on a number of problems if the same limitations of input being unknown ahead of time are imposed on both implementations.

References

1. Amato, G., Scozzari, F., Zaffanella, E.: Efficient constraint/generator removal from double description of polyhedra. *Electron. Notes Theor. Comput. Sci.* **307**, 3–15 (2014)
2. Avis, D.: A revised implementation of the reverse search vertex enumeration algorithm. In: Kalai, G., Ziegler, G.M. (eds.) *Polytopes-Combinatorics and Computation*, vol. 29, pp. 177–198. Springer, Basel (2000). https://doi.org/10.1007/978-3-0348-8438-9_9
3. Avis, D., Bremner, D., Seidel, R.: How good are convex hull algorithms? *Comput. Geom.* **7**(5–6), 265–301 (1997)
4. Avis, D., Fukuda, K.: A pivoting algorithm for convex hulls and vertex enumeration of arrangements and polyhedra. *Discret. Comput. Geom.* **8**(3), 295–313 (1992)
5. Bagnara, R., Hill, P.M., Zaffanella, E.: The Parma polyhedra library: toward a complete set of numerical abstractions for the analysis and verification of hardware and software systems. *Sci. Comput. Program.* **72**(1–2), 3–21 (2008)
6. Barber, C.B., Dobkin, D.P., Huhdanpaa, H.: The quickhull algorithm for convex hulls. *ACM Trans. Math. Softw. (TOMS)* **22**(4), 469–483 (1996)
7. Bastrakov, S.I., Zolotykh, N.Y.: Elimination of inequalities from a facet description of a polyhedron. *Trudy Inst. Mat. i Mekh. UrO RAN* **21**(3), 37–45 (2015). (in Russian)
8. Bastrakov, S.I., Zolotykh, N.Y.: On the dynamic problem of computing generators of a polyhedral cone. *Vestn. Yuzhno-Ural. Gos. Un-ta. Ser. Matem. Mekh. Fiz.* **9**(1), 5–12 (2017). (in Russian)
9. Bastrakov, S.I., Zolotykh, N.Y.: Fast method for verifying Chernikov rules in Fourier-Motzkin elimination. *Comput. Math. Math. Phys.* **55**(1), 160–167 (2015)
10. Bremner, D., Fukuda, K., Marzetta, A.: Primal-dual methods for vertex and facet enumeration. *Discret. Comput. Geom.* **20**(3), 333–357 (1998)
11. Chernikov, S.: *Linear Inequalities*. Nauka, Moscow (1968). (in Russian)
12. Chernikova, N.: Algorithm for finding a general formula for the non-negative solutions of system of linear inequalities. *U.S.S.R. Comput. Math. Math. Phys.* **5**(2), 228–233 (1965)
13. Demenkov, M., Filimonov, N.: Polyhedral barrier regulator design using non-monotonic Lyapunov function. In: *2016 International Conference Stability and Oscillations of Nonlinear Control Systems (Pyatnitskiy’s Conference)*, pp. 1–3. IEEE (2016)

14. Fukuda, K., Prodon, A.: Double description method revisited. In: Deza, M., Euler, R., Manoussakis, I. (eds.) CCS 1995. LNCS, vol. 1120, pp. 91–111. Springer, Heidelberg (1996). https://doi.org/10.1007/3-540-61576-8_77
15. Horst, R., Pardalos, P.M., Van Thoai, N.: Introduction to Global Optimization. Springer, Dordrecht (2000)
16. Motzkin, T., Raiffa, H., Thompson, G., Thrall, R.: The double description method. In: Kuhn, H., Tucker, A.W. (eds.) Contributions to Theory of Games, vol. 2. Princeton University Press, Princeton (1953)
17. Perry, J.: Exploring the dynamic Buchberger algorithm. In: Proceedings of the 2017 ACM on International Symposium on Symbolic and Algebraic Computation, pp. 365–372. ACM (2017)
18. Schrijver, A.: Theory of Linear and Integer Programming. Wiley, New York (1998)
19. Schrijver, A.: Combinatorial Optimization: Polyhedra and Efficiency, vol. 24. Springer, Heidelberg (2003)
20. Terzer, M., Stelling, J.: Large-scale computation of elementary flux modes with bit pattern trees. *Bioinformatics* **24**(19), 2229–2235 (2008)
21. Ziegler, G.M.: Lectures on Polytopes, vol. 152. Springer, Heidelberg (2012)
22. Zolotykh, N.Y., Kubarev, V.K., Lyalin, S.S.: Double description method over the field of algebraic numbers. *Vestn. Udmurtsk. Univ. Mat. Mekh. Komp. Nauki* **28**(2), 161–175 (2018). (in Russian)
23. Zolotykh, N.: New modification of the double description method for constructing the skeleton of a polyhedral cone. *Comput. Math. Math. Phys.* **52**(1), 146–156 (2012)