Fourier–Motzkin Elimination Method

LEONID KHACHIYAN
Department Computer Sci., Rutgers University, Piscataway, USA

MSC2000: 52B12, 68Q25

Article Outline

Keywords

Fourier–Motzkin elimination; Linear inequalities; Linear programming; First order theory of real addition with order; Semilinear set; Projection; Output-polynomial

Consider a system of m linear inequalities in n real variables

$$Ax \leq b, \qquad (1)$$

where $x = (x_1, \dots, x_n)^\intercal \in \mathbf{R}^n$ is the vector of unknowns and A, b are a given real matrix and vector. Let $X = \{x \in \mathbf{R}^n : Ax \leq b\}$ be the solution set of the system, and let $X^{[k]}$ denote the projection of X onto the linear space spanned by the last $n - k$ coordinates:

$$\begin{aligned} X^{[k]} &= \{(x_{k+1}, \dots, x_n) \in \mathbb{R}^{n-k}: \\ &\exists (x_1, \dots, x_k) \in \mathbb{R}^k \\ &\text{s.t. } (x_1, \dots, x_n) \in X\}. \end{aligned}$$

The *Fourier–Motzkin method* [3,4,5,8,10,12,14,15] successively eliminates variables $x_1, \dots, x_{n-1}$ from (1) and computes matrices $A^{[k]}$ and vectors $b^{[k]}$ such that

$$X^{[k]} = \left\{x^{[k]} \in \mathbb{R}^{n-k}:\ A^{[k]}x^{[k]} \leq b^{[k]}\right\}, \qquad k = 1, \dots, n-1,$$

where $x^{[k]} = (x_{k+1}, \dots, x_n)^\intercal$.

In order to eliminate variable x_1, we first multiply each of the m inequalities of (1) by an appropriate positive scalar to make each entry in the first column of A equal to $\pm$ 1 or 0. We can thus assume without loss of generality that the original system of inequalities has the form

$$\begin{aligned} +1 \cdot x_1 + \alpha_i(x^{[1]}) &\leq 0, \quad i \in M_+, \\ -1 \cdot x_1 + \alpha_i(x^{[1]}) &\leq 0, \quad i \in M_-, \\ 0 \cdot x_1 + \alpha_i(x^{[1]}) &\leq 0, \quad i \in M_0, \end{aligned}$$

where $\alpha_i(x^{[1]}) = \alpha_{i2}x_2 + \cdots + \alpha_{in}x_n + \beta_i$ are given affine forms of $x^{[1]} = (x_2, \dots, x_n)^\intercal \in \mathbf{R}^{n-1}$ and M_+, M_-, M_0 are disjoint sets of (indices of) inequalities partitioning the entire set of inequalities in (1):

$$M_+ \cup M_- \cup M_0 = \{1, \dots, m\}.$$

It is easy to see that for each fixed $x^{[1]}$, the inequalities with indices $i \in M_+ \cup M_-$ can be satisfied by some real x_1 if and only if each upper bound $-\alpha_i(x^{[1]})$, $i \in M_+$ on x_1 exceeds each lower bound $\alpha_j(x^{[1]})$, $j \in M_-$ on the same variable, i. e., $-\alpha_i(x^{[1]}) \geq \alpha_j(x^{[1]})$ for all $i \in M_+$ and $j \in M_-$. Combining these $|M_+|\,|M_-|$ inequalities with the remaining $|M_0|$ inequalities of (1) that do not depend on x_1, we arrive at the system of $|M_+|\,|M_-| + |M_0|$ linear inequalities

$$\begin{aligned} &\alpha_i(x^{[1]}) + \alpha_j(x^{[1]}) \leq 0, \quad (i, j) \in M_+ \times M_-, \\ &\alpha_i(x^{[1]}) \leq 0, \quad i \in M_0, \end{aligned}$$

whose solutions set is $X^{[1]}$. The above system can be written as $A^{[1]}x^{[1]} \leq b^{[1]}$ with appropriate matrix $A^{[1]}$ and vector $b^{[1]}$. This gives $X^{[1]} = \{x^{[1]} \in \mathbf{R}^{n-1} : A^{[1]}x^{[1]} \leq b^{[1]}\}$. Eliminating variable x_2 from $A^{[1]}x^{[1]} \leq b^{[1]}$ we obtain a similar description $X^{[2]} = \{x^{[2]} \in \mathbf{R}^{n-2} : A^{[2]}x^{[2]} \leq b^{[2]}\}$ for the second projection and so on. After $n - 1$ steps of the above procedure we have $n - 1$ matrices $A^{[k]}$ and vectors $b^{[k]}$ such that $X^{[k]} = \{x^{[k]} \in \mathbf{R}^{n-k} : A^{[k]}x^{[k]} \leq b^{[k]}\}$, $k = 1, \dots, n-1$.

Solution of Systems of Linear Inequalities and Linear Programming Problems

If the solution set $X = \{x \in \mathbf{R}^n : Ax \leq b\}$ is nonempty, then so are all the projections $X^{[k]} \subseteq \mathbf{R}^{n-k}$, $k = 1, \dots, n-1$, and vice versa. In particular, if $Ax \leq b$ is feasible, then

$$X^{[n-1]} = \left\{x^{[n-1]} \in \mathbb{R}:\ A^{[n-1]}x^{[n-1]} \leq b^{[n-1]}\right\}$$

is a nonempty interval on the scalar variable $x^{[n-1]} = x_n$. Given $A^{[n-1]}$ and $b^{[n-1]}$, we can easily find a point $\overline{x}_n \in X^{[n-1]}$. Then, substituting $x_n = \overline{x}_n$ into $A^{[n-2]}x^{[n-2]} \leq b^{[n-2]}$, we obtain a new feasible system of linear inequalities whose solution set is the interval $\left\{x_{n-2} \in \mathbb{R}\colon\ (x_{n-1}, \overline{x}_n) \in X^{[n-2]}\right\}$. Solving this one-variable system yields a point $\overline{x}^{[n-2]} = (\overline{x}_{n-1}, \overline{x}_n) \in X^{[n-2]}$, which can be substituted in $A^{[n-3]}x^{[x-3]} \leq b^{[n-3]}$ etc. By repeating such backward substitutions, the Fourier–Motzkin method can compute a solution $(\overline{x}_1, \ldots, \overline{x}_n)$ to any feasible system of linear inequalities $Ax \leq b$. 'Historically, it is the 'pre-linear programming' method to solve linear inequalities' [14].

Now suppose that the input system is infeasible, i. e. $X = \{x \in \mathbf{R}^n\colon Ax \leq b = \emptyset\}$. As was pointed out in [10], the Fourier–Motzkin method can then find nonnegative real multipliers $p_1, \ldots, p_m$ such that

$$pA = 0, \quad pb = -1, \quad p = (p_1, \ldots, p_m) \geq 0. \quad (2)$$

To see this, observe that each inequality in $A^{[1]}x^{[1]} \leq b^{[1]}$ is a positive combination of at most two inequalities of the original system. Since a nonnegative combination of nonnegative combinations of some inequalities is a nonnegative combination of the same inequalities, we conclude that each inequality in each system $A^{[k]}x^{[k]} \leq b^{[k]}$, $k = 1, \ldots, n-1$, is a nonnegative combination of the input inequalities. Considering that $A^{[n-1]}x^{[n-1]} \leq b^{[n-1]}$ is an infeasible system of linear inequalities in one variable, $A^{[n-1]}x^{[n-1]} \leq b^{[n-1]}$ is easily seen to contain one or two inequalities whose positive combination yields the infeasible inequality $0 \cdot x_n \leq -1$. This is equivalent to (2). In particular, the Fourier–Motzkin method provides a simple algorithmic proof of the Farkas lemma (cf. ► Farkas lemma; ► Farkas lemma: Generalizations): (1) is feasible if and only if (2) is infeasible.

The Fourier–Motzkin method can also be used to solve the general linear programming problem

$$\xi^* = \max\left\{c^\top x\colon\ Ax \leq b,\ x \in \mathbb{R}^n\right\}. \quad (3)$$

For instance, we can eliminate n variables $x = (x_1, \ldots, x_n)$ from $Ax \leq b, x_{n+1} - c^\top x \leq 0$ to determine the interval $X^{[n]} = \{x_{n+1} : x_{n+1} \leq \xi^*\}$. Then, letting $x_{n+1} = \xi^*$ and solving the resulting system yields an optimal solution.

It should be mentioned that there are far more efficient linear programming algorithms. Note, however, that (3) calls for projecting $X = \{x \in \mathbf{R}^n\colon Ax \leq b\}$ on a one-dimensional subspace. After an appropriate linear transformation, the Fourier–Motzkin method can project $X = \{x \in \mathbf{R}^n\colon Ax \leq b\}$ on any given subspace in $\mathbf{R}^n$.

Complexity of the Fourier–Motzkin Method

Let m_k denote the number of inequalities in the kth system $A^{[k]}x^{[k]} \leq b^{[k]}$ generated by the Fourier–Motzkin method. Since $m_1 = |M_+|\,|M_-| + |M_0| \leq m^2$, we have $m_k \leq m_{k-1}^2$ for all k. So the number of inequalities is at most squared at each step of the method, which implies that m_k is bounded by a doubly exponential function in k, say $m_k \leq m^{2^k}$. The following example shows that with sufficiently many variables, the kth step of the method can produce

$$m_k = m^{2^{k(1-o(1))}}$$

inequalities.

Example 1 [14] Let $n = 2^k + k + 2$ and consider a system of linear inequalities $Ax \leq b$ which contains as left-hand sides $m = 8\binom{n}{3}$ linear forms $\pm x_{i_1} \pm x_{i_2} \pm x_{i_3}$ for all $1 \leq i_1 < i_2 < i_3 \leq n$. By induction on $j = 1, \ldots, k$ it is easy to show that after eliminating the first j variables, the resulting system includes among its left-hand sides all the forms $\pm x_{i_1} \pm \cdots \pm x_{i_s}$ with $k+1 \leq i_1 < \cdots < i_s \leq n$ and $s = 2^j + 2$. In particular, for $j = k$ we have at least $2^{2^k+2} = m^{2^{k(1-o(1))}}$ inequalities in $A^{[k]}x^{[k]} \leq b^{[k]}$.

Let us now return to the first step of the algorithm where we replace $Ax \leq b$ by the $|M_+|\,|M_-| + |M_0|$ new inequalities $A^{[1]}x^{[1]} \leq b^{[1]}$. As was pointed out already by J.B.J. Fourier, 'it nearly always happens that a rather large number of these new inequalities are redundant' and 'their removal greatly simplifies the problem' [8]. If the redundant inequalities are systematically removed at each step of the algorithm, the number m_k of inequalities generated by kth step of the Fourier–Motzkin method is bounded by an exponential function in k. Assume without loss of generality that $X = \{x \in \mathbf{R}^n\colon Ax \leq b\}$ is full-dimensional, then each projection $X^{[k]}$ is also full-dimensional and m_k is the number of facets of $X^{[k]}$. Therefore m_k is bounded by the total number of i-faces of X for $i \geq n - k - 1$. Hence

$$m_k \leq \sum_{i=1}^{k+1} \binom{m}{i} \sim \frac{m^{k+1}}{(k+1)!} \quad \text{for } m \to \infty.$$

(This rough estimate can be improved by using the upper bound theorem [11]; in particular, m_k cannot grow faster than $m^{\lfloor n/2 \rfloor}$.) In the example below, $X^{[k]}$ has

$$m_k \geq \frac{m^{k+1}}{(k+1)^{k+1}}$$

facets.

Example 2 Let $s \geq 2$ be a natural number. Consider the system of $m = (k+1)\,s$ linear inequalities

$$\begin{aligned} y_{ij} &\geq x_i, \quad i = 1, \dots, k, \quad j = 1, \dots, s, \\ x_1 + \dots + x_k &\geq z_l, \quad l = 1, \dots, s, \end{aligned}$$

where x_i, y_{ij}, and z_l are real variables. The elimination of $x_1, \dots, x_k$ results in $s^{k+1} = (m/(k+1))^{k+1}$ inequalities

$$y_{1f(1)} + \dots + y_{kf(k)} \geq z_l, \quad l = 1, \dots, s,$$

where f ranges over the set of all s^k mappings from $\{1, \dots, k\}$ to $\{1, \dots, s\}$. None of the inequalities above is redundant. For instance,

$$y_{11} + \dots + y_{k1} \geq z_1$$

is violated by $y_{11} = \dots = y_{k1} = 0$ and $z_1 = \dots = z_s = 1$, whereas all the other inequalities can be satisfied by giving the remaining variables y_{ij} a high value.

Since detecting the redundancy of an inequality can be done via linear programming (or by maintaining a list of vertices and extreme directions of $X^{[k]}$ with the *double description method* [4,13], see also [9,15] and references herein), the Fourier–Motzkin method runs in exponential space and time. It is natural to ask whether given $X = \{x \in \mathbf{R}^n \colon Ax \leq b\}$ and a number $k \in \{1, \dots, n-1\}$, an irredundant description for $X^{[k]} = \{x^{[k]} \in \mathbf{R}^{n-k} \colon A^{[k]} x^{[k]} \leq b^{[k]}\}$ can be computed in *output-polynomial time*, i. e. by an algorithm that runs in time polynomial in the total input and output size. This question is open even in the bit model of computation for rational A and b, when redundant inequalities can be detected in polynomial time. A related problem is the generation of all vertices for $X = \{x \in \mathbf{R}^n \colon Ax \leq b\}$. The vertex generation problem (or its dual, the *convex hull problem*) can also be solved by the double description method, see e. g. [1], but the question as to whether there is an output-polynomial vertex generation algorithm remains open.

Finally, we mention that the Fourier–Motzkin method can be modified to a quantifier-elimination method for arbitrary *semilinear sets*

$$\begin{aligned} \mathcal{X}^{[k]} = \{(x_{k+1}, \dots, x_n) \in \mathbb{R}^{n-k} &\colon \\ (Q_1 x_1 \in \mathbb{R}) \cdots (Q_k x_k \in \mathbb{R}) & \\ \mathcal{F}(x_1, \dots, x_n) \text{ true}\}, & \quad (4) \end{aligned}$$

where $Q_1, \dots, Q_k \in \{\exists, \forall\}$ are existential and/or universal quantifiers and $\mathcal{F}(x_1, \dots, x_n)$ is a given Boolean function of m threshold predicates

$$\mathcal{F}_i(x) = \begin{cases} \text{true} & \text{if } a_i^\top x \leq b_i, \\ \text{false} & \text{otherwise}, \end{cases}$$

with given coefficients $a_i \in \mathbf{R}^n$ and $b_i \in \mathbf{R}$, $i = 1, \dots, m$. In particular, if $Q_1, \dots, Q_k$ are all existential quantifies and $\mathcal{F} = \mathcal{F}_1 \wedge \dots \wedge \mathcal{F}_m$, we obtain the previously considered problem of projecting the polyhedral set $X = \{x \in \mathbf{R}^n \colon a_i^\top x \leq b_i,\ i = 1, \dots, m\}$ onto the space spanned by the last $n - k$ coordinates. In general, (4) can be transformed into an equivalent quantifier-free representation $\mathcal{X}^{[k]} = \{(x_{k+1}, \dots, x_n) : \mathcal{G}(x_{k+1}, \dots, x_n) \text{ true}\}$, where $\mathcal{G}$ is some Boolean formula whose atoms are new threshold predicates of $(x_{k+1}, \dots, x_n) \in \mathbf{R}^{n-k}$. This can be done, for instance, as follows [6,7]. To eliminate the rightmost quantifier $Q_k x_k \in \mathbf{R}$, write each threshold inequality involving x_k in the form $x_k \leq \alpha_i(x^{(k)})$ or $x_k \geq \alpha_i(x^{(k)})$, where the $\alpha_i{}'$ are given affine forms of the remaining variables $x^{(k)} = (x_1, \dots, x_{k-1}, x_{k+1}, \dots, x_n)$. Replace the infinite range $x_k \in \mathbf{R}$ by the finite set S of sample points $x_k = (\alpha_i(x^{(k)}) + \alpha_j(x^{(k)}))/2$ and $x_k = \pm\infty$. Now it is easy to see that the expression $(\exists x_k \in \mathbf{R})\mathcal{F}(x_1, \dots, x_n)$ is equivalent to the quantifier-free disjunction $\vee_{x_k \in S} \mathcal{F}(x_1, \dots, x_n)$ and that $(\forall x_k \in \mathbf{R})\mathcal{F}(x_1, \dots, x_n)$ can be replaced by the equivalent conjunction $\wedge_{x_k \in S}\mathcal{F}(x_1, \dots, x_n)$. Quantifies $Q_{k-1}x_{k-1}, \dots, Q_1 x_1$ can be eliminated in the same way. For a discussion of faster algorithms that eliminate *blocks* of consecutive identically quantified variables see [2].

See also

► Linear Programming

References

1. Avis D, Bremner B, Seidel R (1997) How good are convex hull algorithms. Comput Geom Th Appl 7:265–302

2. Basu S (1997) An improved algorithm for quantifier elimination over real closed fields. In: Proc. 38th IEEE Symp. Foundations of Computer Sci., pp 56–65
3. Chvátal V (1983) Linear programming. Freeman, New York
4. Dantzig GB, Eaves BC (1975) Fourier–Motzkin elimination and its dual. J Combin Th A 14:288–297
5. Dines LL (1918/9) Systems of linear inequalities. Ann of Math 2(20):191–199
6. Eaves BC, Rothblum UG (1992) Dines–Fourier–Motzkin quantifier-elimination and applications of corresponding transfer principles over ordered fields. Math Program 53:307–321
7. Ferrante J, Rackoff C (1975) A decision procedure for the first order theory of real addition with order. SIAM J Comput 1:69–76
8. Fourier JBJ (1973) Analyse de travaux de l'Académie Royale de Sci., pendant l'anné 1824. In: Kohler DA (ed) Oper Res, 10, pp 38–42, Transl. of a Report by Fouries on his work on linear inequalities.
9. Fukuda K, Prodon A (1996) Double description method revisited. Lecture Notes Computer Sci 1120:91–111
10. Kuhn HW (1956) Solvability and consistency for linear equations and inequalities. Amer Math Monthly 63:217–232
11. McMullen P (1970) The maximal number of faces of a convex polytope. Mathematika 17:179–184
12. Motzkin TS (1936) Beiträge zur Theorie der linearen Ungleichungen. Doktorarbeit, Univ. Basel. Transl. in Cantor D, Gordon B, Rothschild B (eds) (1983): Contribution to the theory of linear inequalities: Selected papers. Birkhäuser, pp 81–103
13. Motzkin TS, Raifa H, Thompson GL, Thrall RM (1953) The double description method. In: Kuhn HW, Tucker AW (eds) Contributions to the Theory of Games, vol II. Ann Math Stud. 28 Princeton Univ. Press, Princeton, pp 81–103
14. Schrijver A (1986) Theory of linear and integer programming. Wiley/Interscience, New York
15. Ziegler GM (1994) Lectures on polytopes. Springer, Berlin

Fractional Combinatorial Optimization

FCO

TOMASZ RADZIK
King's College London, London, UK

MSC2000: 90-08, 90C27, 90C32, 68Q25, 68R05

Article Outline

Keywords

Fractional optimization; Cost-to-time ratio; Newton method; Dinkelbach method; Megiddo parametric search

A *fractional combinatorial optimization problem* (FCOP) is a combinatorial optimization problem with an objective function which is a ratio of two (nontrivial) functions. Instances of a FCOP can be expressed in the general form:

$$\begin{cases} \max & \frac{f(\mathbf{x})}{g(\mathbf{x})}, \\ \text{for} & \mathbf{x} \in \mathcal{X}, \end{cases} \tag{1}$$

where $\mathcal{X} \subseteq \{0, 1\}^p$ is a set of (vectors representing) certain combinatorial structures, and f and g are real-valued functions defined on $\mathcal{X}$. Numbers $f(\mathbf{x})$, $g(\mathbf{x})$, and $f(\mathbf{x})/g(\mathbf{x})$ are usually called the *cost*, the *weight*, and the *mean-weight cost* of structure $\mathbf{x}$. A minimization FCOP is equivalent to the corresponding maximization problem, if the cost function f can be replaced with function $-f$. The FCOPs which appear in the literature on combinatorial optimization include: the *minimum ratio spanning-tree* problem [2,13,14]; the *maximum profit-to-time ratio cycle* problem and the equivalent *minimum cost-to-time ratio cycle* problem [1,3,6,11,12,13,14]; the *minimum mean cycle* problem [1,10,11]; the *maximum mean-weight cut* problem [16]; the *maximum mean cut* problem [5,9]; and the *fractional 0–1 knapsack* problem [7,8].

Consider, as an example, the minimum cost-to-time ratio cycle problem (MRCP). An instance of this problem consists of a directed graph $G = (V, E)$, where $E = \{e_1, \ldots, e_m\}$ is the set of edges, and numbers c_i and t_i associated with each edge e_i, for $i = 1, \ldots, m$. The objective is to find a simple cycle Γ in G which minimizes the ratio of $\sum\{c_i : e_i \in \Gamma\}$ to $\sum\{t_i : e_i \in \Gamma\}$. To express this instance of the MRCP in the form (1), let $\mathcal{X} \subseteq \{0, 1\}^m$ be the set of the characteristic vectors of the simple cycles in G, and for $\mathbf{x} = (x_1, \ldots, x_m) \in \{0, 1\}^m$, let $f(\mathbf{x}) = -(c_1x_1 + \cdots + c_mx_m)$ and $g(\mathbf{x}) = t_1x_1 + \cdots + t_mx_m$. The MRCP models the following *tramp steamer*