Chapter V: Ellipsoid Methods

In this lecture, we will learn our first polynomial-time algorithm to solve convex optimization problems. From a historical point of view, this method is famous for being the first algorithm that could solve linear programs (LP) in polynomial-time, which generated a lot of excitement in the late 70's (cf. Section 5).

From a practical point of view, we point out that this method suffers from many numerical issues, and hence it is not suited to solve even medium-sized problems. The interior-point methods, which we will study in details in a later chapter, are much better for practical purposes. Nevertheless, the ellipsoid method remains a formidable theoretical tool, as it only relies on the availability of an oracle that separates feasible from infeasible solutions. In particular, it does not depend on the number of constraints used to define the feasible set, a property often used to show the existence of polynomial-time algorithms for combinatorial optimization problems.

The ellipsoid method can, in fact, be seen as a generalization of the bisection method to higher dimension. Let us first recall the bisection method for convex optimization problems with a single variable:

$$\operatorname*{\mathbf{minimize}}_{x\in[\ell_0,u_0]} f(x).$$

At the kth iteration, we know that the optimal solution lies in the interval $[\ell_{k-1}, u_{k-1}]$, and we evaluate $f'(x_k)$, where $x_k = \frac{1}{2}(\ell_{k-1} + u_{k-1})$ is the midpoint of the interval. If $f'(x_k) < 0$, then the optimum must lie in the interval $[\ell_{k-1}, x_k]$, so we set $\ell_k = \ell_{k-1}, u_k = x_k$ and we iterate. Otherwise, the optimum lies in $[x_k, u_{k-1}]$, so we set $\ell_k = x_k, u_k = u_{k-1}$ and we iterate. The basic property of this algorithm is that we keep a set $S_k = [\ell_k, u_k]$ that is guaranteed to contain the optimum through the iterations, and its "volume" (the length of the interval) is divided by 2 at each iteration, which ensures a very fast convergence.

In more dimensions, ellipsoids will play the role of the intervals, and we need to define an operation on ellipsoids which plays the role of *halving the intervals.*

1 Halving Ellipsoids

Since a half ellipsoid is not an ellipsoid itself, we must rely on something more complex. A result attributed to Löwner, and completed by John in 1948, states that every convex body $K \subset \mathbb{R}^n$ (i.e., a compact convex set having non-empty interior) is contained in a unique ellipsoid E of minimal volume. Moreover, the ellipsoid obtained by shrinking E by a factor $\frac{1}{n}$ around its center is contained in K. This result falls out of the scope of this lecture, but we can define the Löwner-John ellipsoid of a convex body:

> **Definition 1.** (Löwner John Ellipsoid). Let K be a compact convex set of nonempty interior. The *Löwner-John* ellipsoid of K is the unique ellipsoid E of minimal volume satisfying $E \supseteq K$.

In general, it can be quite complicated to compute the Löwner ellipsoid of an arbitrary convex body K. However, there is a simple explicit formula for the case where K is a half-ellipsoid:

> **Proposition 1.** *(Löwner-John Ellipsoid of a half-ellipsoid).*
> *Let* $E = E(\boldsymbol{a}, Q) := \{\boldsymbol{x} \in \mathbb{R}^n : (\boldsymbol{x} - \boldsymbol{a})^T Q^{-1}(\boldsymbol{x} - \boldsymbol{a}) \leq 1\}$, *for some* $\boldsymbol{a} \in \mathbb{R}^n$ *and* $Q \in \mathbb{S}^n_{++}$. *Let* $\boldsymbol{h} \in \mathbb{R}^n \setminus \{\boldsymbol{0}\}$, *and define the half-ellipsoid*
>
> $$H = E \cap \{\boldsymbol{x} \in \mathbb{R}^n : \boldsymbol{h}^T\boldsymbol{x} \leq \boldsymbol{h}^T\boldsymbol{a}\}.$$

Define $\boldsymbol{b} := \frac{1}{\sqrt{\boldsymbol{h}^T Q \boldsymbol{h}}} Q\boldsymbol{h}$. *Then, the Löwner-John ellipsoid of* H *is* $E' = E(\boldsymbol{a}', Q')$, *where*

$$\boldsymbol{a}' := \boldsymbol{a} - \frac{1}{(n+1)}\boldsymbol{b}$$
$$Q' := \frac{n^2}{n^2-1}\left(Q - \frac{2}{n+1}\boldsymbol{b}\boldsymbol{b}^T\right)$$

The trick to prove this result is to reduce to the case where E is the unit ball and the halfspace is $\{\boldsymbol{x} : x_1 \leq 0\}$, which makes the calculations much easier. Then, the result can be obtained by taking the affine transformation that maps the unit ball to E. Details can be found in [1].

We also need a lemma which shows that the volume of the Löwner-John ellipsoid of a half-ellipsoid is within a constant fraction of the original volume. To prove this lemma, we will use a classical result of linear algebra, which allows to take rank-one updates of determinants:

Lemma 2. *(Matrix-determinant Lemma). Let* A *be an* $n \times n$*-invertible matrix, and let* $\boldsymbol{u}, \boldsymbol{v} \in \mathbb{R}^n$. *Then,*

$$\det(A + \boldsymbol{u}\boldsymbol{v}^T) = (1 + \boldsymbol{v}^T A^{-1}\boldsymbol{u})\ \det A.$$

Proof. We first note that $\det(I_n + \boldsymbol{u}\boldsymbol{v}^T) = (1 + \boldsymbol{v}^T\boldsymbol{u})$, which is a consequence of the identity

$$\begin{bmatrix} I_n & \boldsymbol{0}_n \\ \boldsymbol{v}^T & 1 \end{bmatrix} \cdot \begin{bmatrix} I_n + \boldsymbol{u}\boldsymbol{v}^T & \boldsymbol{u} \\ \boldsymbol{0}_n^T & 1 \end{bmatrix} \cdot \begin{bmatrix} I_n & \boldsymbol{0} \\ -\boldsymbol{v}^T & 1 \end{bmatrix} = \begin{bmatrix} I_n & \boldsymbol{u} \\ 0 & 1 + \boldsymbol{v}^T\boldsymbol{u} \end{bmatrix}.$$

Then, the result follows from $\det(A + \boldsymbol{u}\boldsymbol{v}^T) = \det A(I_n + A^{-1}\boldsymbol{u}\boldsymbol{v}^T) = \det A\ \det(I_n + (A^{-1}\boldsymbol{u})\boldsymbol{v}^T)$. □

Lemma 3. *Let* $E' = E(\boldsymbol{a}', Q')$ *be the Löwner-John ellipsoid of* $E(\boldsymbol{a}, Q) \cap \{\boldsymbol{x} \in \mathbb{R}^n : \boldsymbol{h}^T\boldsymbol{x} \leq \boldsymbol{h}^T\boldsymbol{a}\}$. *Then,*

$$\text{volume}(E') < e^{-\frac{1}{2(n+1)}}\ \text{volume}(E).$$

Proof. First, we recall that $E(\boldsymbol{a}, Q) := \{\boldsymbol{x} \in \mathbb{R}^n : (\boldsymbol{x}-\boldsymbol{a})^T Q^{-1}(\boldsymbol{x}-\boldsymbol{a}) \leq 1\}$ is the image of the unit ball by the affine map $\boldsymbol{x} \mapsto Q^{1/2}\boldsymbol{x} + \boldsymbol{a}$. Hence, $\text{volume}(E(\boldsymbol{a}, Q)) = \det Q^{\frac{1}{2}}\gamma_n$, where γ_n is the volume of the unit ball in $\mathbb{R}^n$.

Then, we make use of the matrix-determinant lemma in order to express $\det Q'$ as a function of $\det Q$:

$$\det Q' = \left(\frac{n^2}{n^2-1}\right)^n \det(Q - \frac{2}{n+1}\boldsymbol{b}\boldsymbol{b}^T) = \left(\frac{n^2}{n^2-1}\right)^n (1 - \frac{2}{n+1}\boldsymbol{b}^T Q^{-1}\boldsymbol{b})\ \det Q = \left(\frac{n^2}{n^2-1}\right)^n \frac{n-1}{n+1}\det Q.$$

Therefore, we have

$$\frac{\text{volume}(E')}{\text{volume}(E)} = \sqrt{\frac{\det Q'}{\det Q}} = \sqrt{\left(\frac{n^2}{n^2-1}\right)^n \frac{n-1}{n+1}} = \sqrt{\left(\frac{n^2}{n^2-1}\right)^{n-1}\left(\frac{n}{n+1}\right)^2} = \left(1 - \frac{1}{n+1}\right)\cdot\left(1 + \frac{1}{n^2-1}\right)^{(n-1)/2}.$$

Finally, using the inequality $1 + t \leq e^t$ (and the inequality is strict for all $t \neq 0$), we get

$$\frac{\text{volume}(E')}{\text{volume}(E)} < \exp\left(-\frac{1}{n+1} + \frac{n-1}{2(n^2-1)}\right) = e^{-\frac{1}{2(n+1)}}$$

□

2 Solving feasibility problems

We are now ready to describe and prove the convergence of a simple variant of the ellipsoid algorithm, which solves the following feasibility problem:

$$\textit{Given a convex set } K \subseteq \mathbb{R}^n\textit{, find } \boldsymbol{x} \in K \textit{ or assert that } K \textit{ is empty.} \tag{$F(K)$}$$

In what follows, we assume that a separation oracle $\texttt{SEP}_K$ is available: given $\boldsymbol{x} \in \mathbb{R}^n$, the oracle either

- asserts that $\boldsymbol{x} \in K$;
- or returns a separating hyperplane between K and $\{\boldsymbol{x}\}$, i.e., a vector $\boldsymbol{h} \in \mathbb{R}^n$ such that

$$\boldsymbol{h}^T \boldsymbol{z} \leq \boldsymbol{h}^T \boldsymbol{x}, \quad \forall \boldsymbol{z} \in K.$$

In addition, the ellipsoid algorithm requires two parameters to solve $F(K)$:
(i) a real $R > 0$ such that $K \subseteq B(\mathbf{0}, R)$, the ball of center $\mathbf{0}$ and radius R;
(ii) a real $r > 0$ such that either $K = \emptyset$ or K contains a ball of radius r.

Algorithm 1 (ELLIPSOID ALGORITHM)

Input: Separation oracle $\texttt{SEP}_K$, $r, R > 0$;
Output: Some $\boldsymbol{x} \in K$, or asserts "$K = \emptyset$".

$P_0 \leftarrow R^2 I_n$, $\boldsymbol{x}_0 \leftarrow \mathbf{0}_n$ ▷ Hence, Assumption (i) implies $K \subseteq E_0 = E(\boldsymbol{x}_0, Q_0)$
$N \leftarrow \lfloor 2n(n+1)\log(R/r) \rfloor$
for $k = 0, 1, \ldots, N$ **do**
  Run the separation oracle $\texttt{SEP}_K(\boldsymbol{x}_k)$.
  if the oracle returns a separating hyperplane $\boldsymbol{h} \neq \mathbf{0}$ **then**
    Define $(Q_{k+1}, \boldsymbol{x}_{k+1})$ using Proposition 1, so $E_{k+1} = E(\boldsymbol{x}_{k+1}, Q_{k+1})$ is the Löwner-John ellipsoid of
$$E_k \cap \{\boldsymbol{x} \in \mathbb{R}^n : \boldsymbol{h}^T \boldsymbol{x} \leq \boldsymbol{h}^T \boldsymbol{x}_k\}.$$
  else
    return $\boldsymbol{x}_k$ ▷ In this case, the separation oracle certified $\boldsymbol{x}_k \in K$, so we are done.
  end if
end for
return "$K = \emptyset$"

Theorem 4. *Under assumptions (i) and (ii), the ellipsoid algorithm described above correctly solves Problem $F(K)$, by making $O(n^2 \log(R/r))$ calls to the oracle.*

Proof. If the algorithm returns a vector $\boldsymbol{x}$, then we know that $\boldsymbol{x} \in K$, so there is nothing to show. Otherwise, by construction, we have $E_k \supseteq K$ at all iterations k, so in particular, $E_{N+1} \supseteq K$. Therefore,

$$\text{volume}(K) \leq \text{volume}(E_{N+1}) < e^{-\frac{N+1}{2(n+1)}} \text{volume}(E_0) \leq e^{-\frac{2n(n+1)\log(R/r)}{2(n+1)}} R^n \gamma_n = r^n \gamma_n,$$

where the strict inequality follows from Lemma 3, and γ_n is the volume of the unit ball in $\mathbb{R}^n$. We have shown that the volume of K is smaller than the volume of a ball of radius r. Hence, by assumption (ii), K must be empty. □

3 Solving convex optimization problems

We first make a few comments on what it means to solve a convex optimization problem.

- The first thing to notice is that there is no hope to design an algorithm that solves convex optimiation problems *exactly* using finite-precision arithmetics. To see this, consider the convex optimization problem
$$\max\{x : x^2 \leq 2\}.$$
Although this problem has integer data, the only solution to this problem is $x^* = \sqrt{2} \notin \mathbb{Q}$.

- Therefore, we must content ourselves with approximate solutions of convex optimization problems. An algorithm is said to solve a convex optimization problem in polynomial time if *it returns an $\epsilon-$suboptimal solution*, in time polynomial with respect to the input length of the problem and $\log \frac{1}{\epsilon}$.

- Finally, there are many technical details implied by the use of finite-precision arithmetics. This is a problem, since the formula to update the ellipsoid involves a square root, which needs to be approximated when we use finite-precision arithmetics. We will skip this issue in this section, and simply assume that exact arithmetics is used.

Now, we consider a convex optimization problem of the form

$$p^* = \inf_{\boldsymbol{x} \in K} \boldsymbol{c}^T \boldsymbol{x}, \tag{P}$$

where $K \subseteq \mathbb{R}^n$ is convex body and $\|\boldsymbol{c}\| = 1$. If we have an efficient routine to solve the feasibility problem $F(K)$, then we can solve the convex optimization problem P to arbitrary precision by doing a binary search to find the smallest δ such that the convex set $K(\delta) := K \cap \{\boldsymbol{x} \in \mathbb{R}^n : \boldsymbol{c}^T \boldsymbol{x} \leq \delta\}$ is nonempty. However, if we want to use the ellipsoid algorithm from the previous section, we need to choose r' so that $K \cap \{\boldsymbol{x} \in \mathbb{R}^n : \boldsymbol{c}^T \boldsymbol{x} \leq \delta\}$ contains a ball or radius r', even when δ gets very close to the optimum value, say $\delta = p^* + \epsilon$. This is possible, as long as K satisfies the conditions (i) and (ii) from the previous section for two **given** reals $R > 0$ and $r > 0$.

Proposition 5. *Let K be a convex body satifying (i) and (ii) for $r, R > 0$, and let $0 < \epsilon < R$. Then, either K is empty, or the ϵ-suboptimal set for (P) contains a ball of radius $\frac{r\epsilon}{2R+r}$.*

Proof. We can assume that K contains a ball $B = B(\boldsymbol{x}_0, r)$ of center $\boldsymbol{x}_0 \in K$ and radius r, (otherwise, K is empty, so there is nothing to show). Since K is compact and linear functions are continuous, Problem (P) attains its minimum at some $\boldsymbol{x}^* \in K$. Without loss of generality, we assume $\boldsymbol{x}^* = 0$ (by shifting K), so $p^* = 0$ and the ϵ-suboptimal set reads $K_\epsilon := K \cap \{\boldsymbol{x} \in \mathbb{R}^n : \ \boldsymbol{c}^T \boldsymbol{x} \leq \epsilon\}$. Moreover, observe that $\boldsymbol{c}^T \boldsymbol{x}_0 \geq 0$, as $\boldsymbol{c}^T \boldsymbol{x}_0 < p^* = 0$ would contradict that $\boldsymbol{x}^* = \boldsymbol{0}$ is an optimal solution.

By convexity of K, the ball $B(t\,\boldsymbol{x}_0, t\,r)$ is contained in K for all $t \in [0, 1]$. Now, we search the largest $t \in [0, 1]$ such that $B(t\,\boldsymbol{x}_0, t\,r) \subseteq K_\epsilon \iff B(t\,\boldsymbol{x}_0, t\,r) \subseteq \{\boldsymbol{x} \in K : \ \boldsymbol{c}^T \boldsymbol{x} \leq \epsilon\}$. This inclusion can be rewritten as

$$\max_{\boldsymbol{x} \in B(t\,\boldsymbol{x}_0, t\,r)} \boldsymbol{c}^T \boldsymbol{x} \leq \epsilon,$$

or, after the change of variable $\boldsymbol{x} = t(\boldsymbol{x}_0 + r\boldsymbol{z})$,

$$t\boldsymbol{c}^T \boldsymbol{x}_0 + t\,r \max_{\boldsymbol{z} \in B(\boldsymbol{0}, 1)} \boldsymbol{c}^T \boldsymbol{z} \leq \epsilon.$$

The above maximization problem has optimal solution $\boldsymbol{z}^* = \boldsymbol{c}$, which is an easy consequence of the Cauchy-Schwarz inequality: $\forall \boldsymbol{z} \in B(0, 1), \boldsymbol{c}^T \boldsymbol{z} \leq \|\boldsymbol{c}\|\|\boldsymbol{z}\| \leq \|\boldsymbol{c}\| = \boldsymbol{c}^T \boldsymbol{z}^* = 1$. Hence, the inclusion $B(t\,\boldsymbol{x}_0, t\,r) \subseteq K_\epsilon$ is equivalent to $t(\boldsymbol{c}^T \boldsymbol{x}_0 + r) \leq \epsilon$. We have thus shown that K_ϵ contains a ball of radius $t^*\,r$, where $t^* = \min(1, \epsilon(\boldsymbol{c}^T \boldsymbol{x}_0 + r)^{-1})$.

Finally, to obtain a bound that does not depend on $\boldsymbol{c}^T\boldsymbol{x}_0$, we observe that $\boldsymbol{c}^T\boldsymbol{x}_0 \leq \|\boldsymbol{c}\|\|\boldsymbol{x}_0\| \leq 2R$, so that $t^* r = \min(r, r\epsilon(\boldsymbol{c}^T\boldsymbol{x}_0 + r)^{-1}) \geq \min(r, \frac{r\epsilon}{2R+r}) = \frac{r\epsilon}{2R+r}$ (the last equality follows from the assumption $\epsilon < R$), which concludes the proof.

□

Rather than relying on binary search to solve (P), it is possible to modify the ellipsoid algorithm as follows, so that it directly returns an $\epsilon-$approximate solution: Whenever the oracle certifies $\boldsymbol{x}_k \in K$, do not stop the computations, but instead store $f^{\text{best}} = \boldsymbol{c}^T\boldsymbol{x}_k$, and update the ellipsoid to be the Löwner-John ellipsoid of $E_k \cap \{\boldsymbol{x} \in \mathbb{R}^n : \boldsymbol{c}^T\boldsymbol{x} \leq \boldsymbol{c}^T\boldsymbol{x}_k\}$. In this way, E_{k+1} must contain the $(f^{\text{best}} - p^*)$-suboptimal set, and this property will remain true for all subsequent iterations.

We run this modified ellipsoid algorithm for $N' = \lfloor 2n(n+1)\log\left(\frac{R}{\epsilon}(2\frac{R}{r}+1)\right)\rfloor$ iterations. The condition $\epsilon < R$ guarantees that $N' \geq N$, and so we can return that "the problem is infeasible" (i.e., $K = \emptyset$) if a feasible solution has not been found during the first N iterations. Otherwise, similarly as in the proof of Theorem 4, the volume of $E_{N'+1}$ is smaller than the volume of a ball of radius $\epsilon/(2R/r+1)$. On the other hand, we know that $E_{N'+1}$ contains the $(f^{\text{best}} - p^*)-$suboptimal set; Hence by Proposition 5, it contains a ball of radius $\frac{r(f^{\text{best}}-p^*)}{2R+r}$. This shows:

$$\frac{r(f^{\text{best}} - p^*)}{2R+r} \leq \frac{\epsilon}{2R/r+1} \iff f^{\text{best}} \leq p^* + \epsilon.$$

Thus, we have shown:

Theorem 6. *If constants R and r are known, such that the convex body K satisfies assumptions (i)-(ii), then we can find an $\epsilon-$suboptimal solution of (P), or assert that this problem is infeasible, by making $O(n^2 \log \frac{R}{\min(r,\epsilon)})$ calls to the separation oracle.*

4 Weak Optimization & Separation

In the previous section, we have given a polynomial-time algorithm for convex optimization problems with two additional assumptions on the feasible set K. While the assumption (i) is easy to satisfy in practice, (we can generally put a large bound on all optimization variables), it is much more tricky to guarantee assumption (ii) (K is either empty or contains a ball of radius $r > 0$).

When using finite-precision arithmetics, it would make sense to set r to the machire precision, but then, it means that we are not able to differentiate infeasible problems from problems with a very small feasibility region. In particular, finite-precision arithmetics makes it unrealistic to have an *exact separation oracle*, as assumed in the previous sections. To cope with this issue, Grötschel, Lovász and Schrijver [2] proposed to work with a *weak separation oracle*:

Definition 2. (Weak separator). A weak separation oracle for $K \subset \mathbb{R}^n$ takes $\boldsymbol{x} \in \mathbb{Q}^n$ and $\epsilon \in \mathbb{Q}_{++}$ as input, and either

1. asserts that $\boldsymbol{x}$ is $\epsilon-$almost in K, i.e., $d(\boldsymbol{x}, K) := \inf_{\boldsymbol{z}\in K} \|\boldsymbol{x} - \boldsymbol{z}\| \leq \epsilon$.
2. or returns a hyperplane $\boldsymbol{h}$ with $\|\boldsymbol{h}\|_\infty = 1$ that "almost" separates $\boldsymbol{x}$ from all points that are deep in K, i.e.,
$$\boldsymbol{h}^T\boldsymbol{x} \leq \boldsymbol{h}^T\boldsymbol{z} + \epsilon, \quad \forall \boldsymbol{z} \in \mathbb{R}^n, B(\boldsymbol{z}, \epsilon) \subseteq K.$$

Then, they defined a relaxed version of the optimization problem P, in which we only need to separate points that are ϵ-almost/ϵ-deep in K:

Definition 3. (Weak optimization). Given $\epsilon > 0$, the weak optimization version of Problem (P) consists in either

1. asserting that K does not contain any ball or radius ϵ.
2. or returning a point $\boldsymbol{x}^*$ that is $\epsilon-$almost in K, i.e., $d(\boldsymbol{x}^*, K) := \inf_{\boldsymbol{z}\in K} \|\boldsymbol{x}^* - \boldsymbol{z}\| \leq \epsilon$, and such that $\boldsymbol{x}^*$ is almost optimal compared to $\epsilon-$deep points:

$$\boldsymbol{c}^T\boldsymbol{x}^* \leq \boldsymbol{c}^T\boldsymbol{z} + \epsilon, \quad \forall \boldsymbol{z} \in \mathbb{R}^n, B(\boldsymbol{z}, \epsilon) \subseteq K.$$

The main result of Grötschel, Lovász and Schrijver is as follows:

Theorem 7. *Given R and a polynomial-time weak separation oracle for $K \subseteq B(0, R)$, we can solve the weak optimization problem in polynomial time.*

The proof of this result essentially follows the ideas presented in this lecture, but the formulas are much more complicated, because we use the weak separator, and we need to "blow-up" the ellipsoids in order to ensure that $E_k \subseteq K$, in order to compensate for the lack of accuracy caused by the rounding; see [2] for details. Interestingly, there is also a converse result, so that the problems of weak optimization and weak separation are polynomially inter-reducible (up to the knowledge of a sufficiently large $R > 0$):

Theorem 8. *Given a polynomial-time weak optimization oracle for a convex set K, we can solve the weakly separation problem for K in polynomial time.*

5 Final remarks and historical notes

The ellipsoid method was initially proposed by Yudin and Nemirovki [5] and Shor [6] in the 70's. The method attracted a lot of interest after the work of Khachian [3], who used it to design the first polynomial-time algorithm to solve linear programming: An essential contribution of Khachian was to show that we can reduce linear programming to a problem using finite-precision arithmetics only, with a number of digits polynomially bounded by the input size of the problem. Later, Grötschel, Lovász and Schrijver (see [2]) discussed the implications of this result for combinatorial optimization problems, using linear programs with exponentially many constraints that can be *separated* in polynomial time.

Last but not least, we mention that the ellipsoid method is just one of many "cutting plane methods", which differ in the way a set $E_k \supseteq K$ is maintained (not necessarily an ellipsoid), and how the query point $\boldsymbol{x}_k$ is selected. While it was the first method to solve the feasibility problem in polynomial time (its total complexity is $O(n^2 \log \frac{R}{r}(SO + n^2))$, where SO is the complexity of the separation oracle), more efficient methods exist: The best cutting plane method known to date is due to Lee, Sidford and Wong [4] and solves the same problem in $O(n \log \frac{nR}{r} SO + n^3 \log^{O(1)} \frac{nR}{r})$.

References

[1] Bland, R.G., Goldfarb, D., & Todd, M.J. (1981). The Ellipsoid Method: A survey. *Operations research*, 29(6), 1039–1091.

[2] Grötschel, M., Lovász, L. & Schrijver, A. (1988). Geometric Algorithms and Combinatorial Optimization.

[3] Khachian, L.G. (1979). A polynomial algorithm in linear plogramming. *Soviet Math. Dokl.*, 20, 191–194.

[4] Lee, Y.T., Sidford, A., & Wong, S.C.W. (2015). A faster cutting plane method and its implications for combinatorial and convex optimization. In proceedings of FOCS 2015, pp. 1049-1065.

[5] Yudin, D.B. and Nemirovski, A.S. (1976). Informational complexity and effective methods of solution for convex extremal problems. *Ekonomika i Matematicheskie Metody*, 12, 357–369.

[6] Shor, N. Z. (1977). Cut-off method with space extension in convex programming problems. *Cybernetics*, 13(1), 94–96.