Robust-to-Dynamics Linear Programming

Amir Ali Ahmad and Oktay Günlük

***Abstract*— We consider a class of robust optimization problems that we call "robust-to-dynamics optimization" (RDO). The input to an RDO problem is twofold: (i) a mathematical program (e.g., an LP, SDP, IP, etc.), and (ii) a dynamical system (e.g., a linear, nonlinear, discrete, or continuous dynamics). The objective is to maximize over the set of initial conditions that forever remain feasible under the dynamics. The focus of this paper is on the case where the optimization problem is a linear program and the dynamics are linear. We establish some structural properties of the feasible set and prove that if the linear system is asymptotically stable, then the RDO problem can be solved in polynomial time. We also outline a semidefinite programming based algorithm for providing upper bounds on robust-to-dynamics linear programs.**

***Index Terms*— Robust optimization, linear programming, semidefinite programming, dynamical systems.**

I. Introduction

The field of robust optimization (RO) deals with the scenario where the constraint functions of an optimization problem are not exactly known and a decision has to be made which is feasible irrespective of what the true functions turn out to be. Broadly speaking, a robust optimization problem is a problem of the form

$$\min_x \{f(x) : g_i(x) \leq 0 \ \forall g_i \in \mathcal{G}_i, i = 1, \ldots, m\}, \quad (1)$$

where $\mathcal{G}_i$ is a prescribed "uncertainty set" for the i-th constraint function. Over the last couple of decades, our understanding of computational tractability of robust optimization problems has matured quite a bit. A rather complete catalog is available which tells us for which types of optimization problems and which types of uncertainty sets the robust counterpart of the problem becomes tractable (as in solvable in polynomial time) versus intractable (typically meaning NP-hard). For example, if the original problem is a linear program (LP), i.e., the functions f, g_i are all linear, and if the sets $\mathcal{G}_i$ in parameter space are polyhedral, then the robust counterpart can be written as a polynomially-sized LP. If the uncertainty sets are instead ellipsoidal, then the robust counterpart can be written as a second order cone program and solved to arbitrary accuracy in polynomial time. On the other hand, if the original problem is for example a convex quadratically constrained quadratic program and the uncertainty sets are polyhedral, then the robust counterpart is NP-hard. A full collection of characterizations of this type is available in [1] and [2] and references therein.

Amir Ali Ahmadi is with the Department of Operations Research and Financial Engineering at Princeton University. Email: a_a_a@princeton.edu. He has been partially supported for this work by the AFOSR Young Investigator Program Award. Oktay Günlük is with the Mathematical Programming Group of the Mathematical Sciences division of IBM Research. Email: gunluk@us.ibm.com.

Our goal in this paper is to initiate a similar algorithmic study for a special type of robust optimization problems that interact with dynamical systems. We call these problems *robust to dynamics optimization problems* (RDO). An RDO problem has a very clean an natural mathematical formulation. It has two pieces of input:

1) an optimization problem:

$$\min_x \{f(x) : x \in \Omega\} \quad (2)$$

2) a dynamical system:

$$\begin{array}{lll} x_{k+1} & = g(x_k) & \text{(in discrete time)} \\ \text{or } \dot{x} & = g(x) & \text{(in continuous time).} \end{array} \quad (3)$$

Here, we have $x \in \mathbb{R}^n$, $\Omega \subseteq \mathbb{R}^n$, $f, g : \mathbb{R}^n \to \mathbb{R}$; x_k denotes the state at time step k, and $\dot{x}$ is the derivative of x with respect to time. RDO is then the following optimization problem:

$$\min_{x_0} \{f(x_0) : x_k \in \Omega, k = 0, 1, 2, \ldots\} \quad (4)$$

in discrete time, or

$$\min_{x_0} \{f(x_0) : x(t; x_0) \in \Omega, \forall t \geq 0\} \quad (5)$$

in continuous time. In the latter case, the notation $x(t; x_0)$ denotes the solution of the differential equation $\dot{x} = g(x)$ at time t, starting at the initial condition $x_0 \in \mathbb{R}^n$. In words, we are optimizing an objective function over the set of initial conditions[1] that never leave the set Ω under the dynamics. Problems of this type can appear in any real-life situation where our current decisions—which need to optimize an objective function—initiate a trajectory of a dynamical system that we desire to constrain in future time. One such setting e.g. arises in the study of robust model predictive control [3].

Various RDO problems can be defined depending on what combination of the optimization problem in (2) and the dynamics in (3) one chooses to consider:

Optimization Problem	**Dynamics**
Linear Program	Linear
Semidefinite Program	Nonlinear
Geometric Program	Uncertain
Integer Program	Time-varying
Robust Linear Program	Discrete/continuous/hybrid of both
⋮	⋮

[1]By picking f appropriately we can also penalize future values of the state; e.g., $f(x_0) = \hat{f}(x_0) + \hat{f}(g(x_0))$ in the discrete time case.

In this paper, we consider the case where the optimization problem is a linear program and the dynamics is linear and in discrete time. Already a host of interesting mathematical questions arise in this setting, which is arguably the simplest one possible. In [4], we extend these results to the case where the dynamics is uncertain and time-varying (in a switched-systems model), and to the case where some of the functions in the optimization problem are convex quadratics.

II. Robust to linear dynamics linear programming

The *robust to linear dynamics linear programming* problem (R-LD-LP) is the following optimization problem:

$$\min_{x_0}\{c^T x_0 : Ax_k \le b, k = 0, 1, 2, \ldots; x_{k+1} = Gx_k\}. \quad (6)$$

The input to this problem is $c \in \mathbb{F}^n, A \in \mathbb{F}^{m\times n}, b \in \mathbb{F}^m, G \in \mathbb{F}^{n\times n}$, where we either have $\mathbb{F} = \mathbb{R}$ (for generality of presentation) or $\mathbb{F} = \mathbb{Q}$ (when we would like to study the complexity of the problem in the bit model of computation). We assume throughout the paper that the polyhedron $\{Ax \le b\}$ is bounded; i.e., it is a polytope. Problem (6) has a simple geometric interpretation: we are interested in optimizing a linear function not over the entire polyhedra $\{Ax \le b\}$, but over a subset of it that does not leave the polytope under the application of G, G^2, G^3, etc. So the feasible set of R-LD-LP is the following set

$$\mathcal{S} := \bigcap_{k=0}^{\infty} \{x|\ AG^k x \le b\}. \quad (7)$$

Let us give an example and draw some pictures.

Example 2.1: Consider an instance of R-LD-LP with

$$A = \begin{bmatrix} -1 & 0 \\ 0 & -1 \\ 0 & 1 \\ 1 & 1 \end{bmatrix}, b = \begin{bmatrix} 1 \\ 1 \\ 1 \\ 3 \end{bmatrix}, c = \begin{bmatrix} -1 \\ 0 \end{bmatrix}, G = \begin{bmatrix} 0.6 & -0.4 \\ 0.8 & 0.5 \end{bmatrix}. \quad (8)$$

The caption of Figure 1 explains the analysis of this instance. The feasible set of this R-LD-LP is plotted in Figure 1 **(g)**. The optimal value is achieved at its right most corner and is equal to 1.1492.

Several natural questions arise:

1) Does convergence to the set $\mathcal{S}$ happen in a finite number of steps?
2) Is the set $\mathcal{S}$ always polyhedral?
3) If (or when) it is polyhedral, how many facets do we need to describe it?
4) Can problem (6) be solved in polynomial time?
5) It is clear that any LP resulting from some truncation of (6) (fixing an upper bound on k) gives a lower bound on R-LD-LP. But how can we compute upper bounds?

These questions motivate what we present in the sequel.

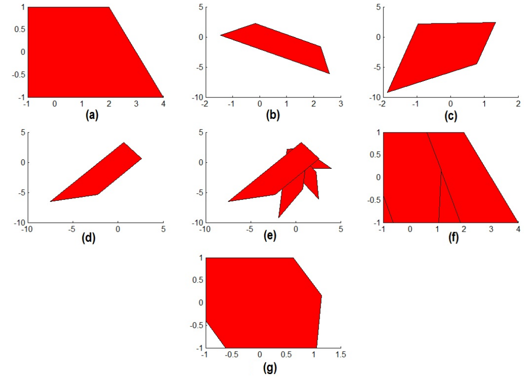

Fig. 1. Plots correspond to the instance of R-LD-LP given in (8): **(a)** The polytope $\{Ax \le b\}$. **(b)** the polytope $\{AGx \le b\}$. **(c)** The polytope $\{AG^2x \le b\}$. **(d)** The polytope $\{AG^3x \le b\}$. **(e)** Polytopes in (a)-(d) plotted together. **(f)** The original polytope $\{Ax \le b\}$ (largest and in red), two cuts added by the polytope $\{AGx \le b\}$ (black lines with negative slope), a single cut added by $\{AG^2x \le b\}$ (black line with positive slope); no cut is added in this case by $\{AG^3x \le b\}$. **(g)** $\bigcap_{k=0}^{3}\{x|\ AG^kx \le b\}$; this turns out to be equal to $\bigcap_{k=0}^{\infty}\{x|\ AG^kx \le b\}$, i.e., the feasible set of our R-LD-LP.

A. Some structural results

Let us start with a simple lemma. A set $\mathcal{C}$ is said to be *invariant* (with respect to a particular dynamical system), if points starting in $\mathcal{C}$ remain in it forever.

Lemma 2.1: The set $\mathcal{S}$ defined in (7) is closed, convex, and invariant.

Proof: Convexity is obvious, as $\mathcal{S}$ is given by (an infinite) intersection of polyhedra. Invariance is also a trivial implication of the definition: if $x \in \mathcal{S}$, then $Gx \in \mathcal{S}$. To prove that $\mathcal{S}$ is closed, consider a sequence $\{x_j\} \to \hat{x}$, with each $x_j \in \mathcal{S}$. Suppose we had $\hat{x} \notin \mathcal{S}$. This means that $AG^k\hat{x} > b$ for some k. But this implies that $AG^kx > b$, for all x sufficiently close to $\bar{x}$, including some elements of the sequence $\{x_j\}$. ∎

Our next lemma gives a way of detecting termination as we add cuts and is also simple to prove.

Lemma 2.2: Suppose for some r we have

$$\bigcap_{k=0}^{r}\{x|\ AG^kx \le b\} = \bigcap_{k=0}^{r+1}\{x|\ AG^kx \le b\}. \quad (9)$$

Then,

$$\bigcap_{k=0}^{\infty}\{x|\ AG^kx \le b\} = \bigcap_{k=0}^{r}\{x|\ AG^kx \le b\}.$$

Proof: We observe that condition (9) implies that the set $\mathcal{S}_r := \bigcap_{k=0}^{r}\{x|\ AG^kx \le b\}$ is invariant. If not, there would exist an $x \in \mathcal{S}_r$ with $Gx \notin \mathcal{S}_r$. But this implies that $x \notin \mathcal{S}_{r+1}$, which is a contradiction. Invariance of S_r implies that $\mathcal{S}_r = \mathcal{S}$. ∎

Lemma 2.3: Condition (9) can be checked in polynomial time.

Proof: We clearly have $\mathcal{S}_{r+1} \subseteq \mathcal{S}_r$. To check the inclusion $\mathcal{S}_r \subseteq \mathcal{S}_{r+1}$, we can solve m LP feasibility problems that try to find a point in S_r that violates one of the m defining inequalities of S_{r+1}. The desired inclusion is true if and only if all of these LPs are infeasible. ■

Now that we know finite termination can be efficiently detected, a natural question is whether finite termination always occurs. The answer is negative (and in more than one way!).

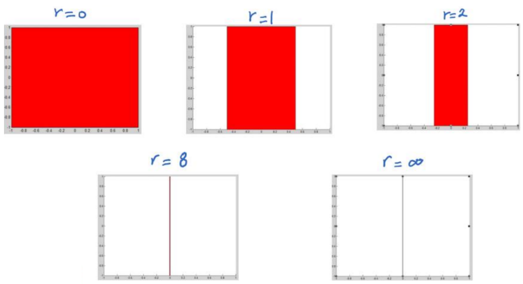

Fig. 2. The counterexample in the proof of Theorem 2.4.

Theorem 2.4: Let $\mathcal{S}_r := \bigcap_{k=0}^{r}\{x|\ AG^k x \le b\}$. Then the convergence of $\mathcal{S}_r$ to $\mathcal{S}$ in (7) may not be finite.

Proof: Consider an instance of R-LD-LP with $G = \begin{pmatrix} 2 & 0 \\ 0 & \frac{1}{2} \end{pmatrix}$, and let the initial polytope $\{Ax \le b\}$ be the unit square on the plane. Then it is straightforward to see that convergence is not finite; see Figure 2. ■

Even though in our previous example convergence did not happen in a finite number of steps, the feasible set of R-LD-LP still turned out to be polyhedral. Is this always the case? The answer is negative.

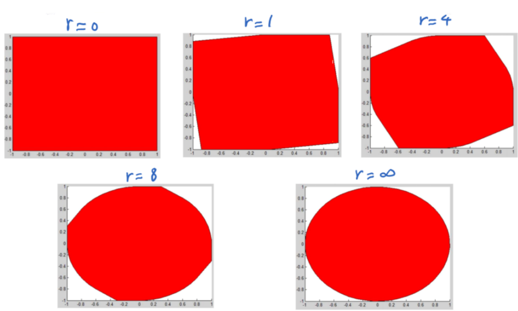

Fig. 3. The counterexample in the proof of Theorem 2.5.

Theorem 2.5: The feasible set of R-LD-LP may not be polyhedral.

Proof: Consider the following instance of R-LD-LP in $\mathbb{R}^2$:

$$A = \begin{bmatrix} 1 & 0 \\ -1 & 0 \\ 0 & 1 \\ 0 & -1 \end{bmatrix}, b = \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \end{bmatrix} \quad \text{and} \quad G = \begin{bmatrix} \cos(\theta) & \sin(\theta) \\ -\sin(\theta) & cos(\theta) \end{bmatrix}, \tag{10}$$

where $\theta/2\pi$ is irrational. In other words, letting $B = [-1, 1]^2$ and using polar coordinates with the convention $x_k = (r_k, \phi_k)$, the feasible region consists of points $x_0 = (r_0, \phi_0)$ such that $(r_0, \phi_0 + k\theta) \in B$ for all non-negative integers k.

Notice that the closed disk D centered at the origin with radius 1 is contained in B and consequently $(r_0, \phi_0 + k\theta) \in B$ for all k provided that $r_0 \le 1$. On the other hand, consider a point $x_0 = (r_0, \phi_0)$ such that $r_0 > 1$, and let C be the circle centered at the origin with radius r_0. Clearly, all $x_k \in C$. Furthermore, note that for any fixed $r_0 > 1$, there exists $2\pi > \beta_1 > \beta_2 \ge 0$ such that none of the points in C between (r_0, β_1) and (r_0, β_2) belong to B. But, for large enough $k > 1$, we have $\phi_0 + k\theta \in [\beta_1, \beta_2]$ as θ and 2π are rationally independent (see [5, Chapter 3, Theorem 1]). Consequently, x_0 is feasible if and only if $x_0 \in D$.

Figure 3 depicts an example of the phenomenon described in this proof. ■

We remark that some entries in the matrix G presented in the proof we just gave are irrational. It is an interesting open problem to determine whether one can have an instance of R-LD-LP, with rational A, b, G, where the final feasible set is not polyhedral. We have been unable to construct such an example.

Here is another question: Is the feasible set of R-LD-LP polyhedral if the spectral radius $\rho(G)$ of the matrix G is less than one? The answer in this case is positive. In fact a stronger result is proven in the next section.

III. A POLYNOMIAL-TIME ALGORITHM

In this section we present a polynomial-time algorithm for R-LD-LP in (6) in the case where the dynamics is asymptotically stable, i.e., $\rho(G) < 1$. Arguably, this is the most interesting setting. Note that if $\rho(G) > 1$, then almost all trajectories shoot out to infinity and hence under our assumption that the polyhedron $\{Ax \le b\}$ is bounded, the feasible set of R-LD-LP will not be full dimensional. The boundary case $\rho(G) = 1$ is more tricky. Here trajectories can stay bounded or go to infinity depending on the geometric/algebraic multiplicity of the eigenvalues with absolute value one. Even in the bounded case, we have shown already in Theorem 2.5 that the feasible set of R-LD-LP may not be polyhedral. Hence the optimal value of R-LD-LP may not even be a rational number (consider, e.g., Figure 3 with $c = (1, 1)^T$).

In the case where $\rho(G) < 1$, we must have the condition that the origin is inside the polytope $\{Ax \le b\}$ or else the feasible set of R-LD-LP will be empty. We make the slightly stronger assumption that the origin is strictly in the interior of the polytope.

Theorem 3.1: R-LD-LP (see (6)) can be solved in polynomial time when $\rho(G) < 1$ and the origin strictly in the interior of $\{Ax \le b\}$.

Proof: We will show that there exists an integer r, with size polynomial in the data (A, b, c, G), such that the feasible set of R-LD-LP given in (7) in fact coincides with its finite truncation of length r:

$$\mathcal{S}_r = \bigcap_{k=0}^{r}\{x|\ AG^k x \le b\}.$$

Hence, we can instead minimize $c^T x$ over this polyhedron. In view of the fact that this is a polynomially-sized LP, and that LPs can be solved in polynomial time [6], the theorem would be established.

Our algorithm for finding the integer r is broken down into four simple steps:

- Find an ellipsoid $\{x^T Px \leq 1\}$, defined by a positive definite matrix P, which is invariant under the dynamics $x_{k+1} = Gx_k$; i.e., P satisfies the linear matrix inequality $G^T PG \preceq P$.
- Find a scalar $\alpha_2 > 0$ such that
$$\{Ax \leq b\} \subseteq \{x^T Px \leq \alpha_2\}.$$
- Find a scalar $\alpha_1 > 0$ such that
$$\{x^T Px \leq \alpha_1\} \subseteq \{Ax \leq b\}.$$
- Find a "shrinkage factor" $\gamma \in (0, 1)$, which gives a lower bound on the amount our ellipsoid shrinks in every iteration. This would be a scalar satisfying $G^T PG \preceq \gamma P$.
- Let
$$r = \lceil \frac{\log \frac{\alpha_1}{\alpha_2}}{\log \gamma} \rceil. \tag{11}$$

The idea is that all the points inside the outer ellipsoid $\{x^T Px \leq \alpha_2\}$, and in particular all the points in $\{Ax \leq b\}$, are guaranteed to be within the inner ellipsoid $\{x^T Px \leq \alpha_1\}$ after at most r steps. Once they are in the inner ellipsoid, they can never leave it because of the invariance condition. Hence, if any point were to leave the polyhedron $\{Ax \leq b\}$, it needs to do it before r steps.

What we need to do now is to show that $P, \gamma, \alpha_1, \alpha_2$ can be computed in polynomial time.

- **Computation of P.** To find an invariant ellipsoid for G, we solve the linear system
$$G^T PG - P = -I, \tag{12}$$
where I is the $n \times n$ identity matrix. This is the well-known Lyapunov equation. Since $\rho(G) < 1$, this equation is guaranteed to have a unique solution. Note that since this is a linear system, the entries of the solution P will be rational numbers of polynomial size. Further, we claim that P will automatically turn out to be positive definite (hence the sublevel sets $\{x^T Px \leq \alpha\}$ will be compact). To see this, suppose we had $y^T Py \leq 0$, for some $y \in \mathbb{R}^n, y \neq 0$. Multiplying (12) from left and right by y^T and y, we see that $y^T G^T PGy \leq -y^T y < 0$. In fact, $y^T (G^k)^T PG^k y \leq -y^T y < 0$, for all $k \geq 1$. But since $\rho(G) < 1$ implies that the linear system is asymptotically stable, we must have $G^k y \to 0$, and hence $y^T (G^k)^T PG^k y \to 0$, a contradiction.
In short, we can get a rational positive definite matrix P of polynomial size just by solving the linear system (12). This defines the shape of our invariant ellipsoid.
- **Computation of γ.** We claim that we can let our shrinkage factor be equal to
$$\gamma = 1 - \frac{1}{\max_i \{P_{ii} + \sum_{j \neq i} |P_{i,j}|\}}.$$
Note that $P_{ii} \geq 1$ because (12) and positive definiteness of P imply that $P - I \succeq 0$. Hence γ is indeed a number in $[0, 1)$. To prove that this choice of γ works, observe that by (12) we have the following inequality for all x:
$$\begin{aligned} x^T G^T PGx &= x^T Px - x^T x \\ &\leq x^T Px(1 - \eta), \end{aligned} \tag{13}$$
where η is any number such that
$$\eta x^T Px \leq x^T x,$$
(for all x). The largest such η is exactly $\frac{1}{\lambda_{max}(P)}$. Since we do not want to deal with eigenvalues, we simply observe that an upper bound on $\lambda_{max}(P)$, via Gershgorin's circle theorem, is
$$\lambda_{max}(P) \leq \max_i \{P_{ii} + \sum_{j \neq i} |P_{i,j}|\}.$$
- **Computation of α_2.** By solving, e.g., n LPs, we can place our polytope $\{Ax \leq b\}$ in a box; i.e., compute $2n$ scalars l_i, u_i such that
$$\{Ax \leq b\} \subseteq \{l_i \leq x_i \leq u_i\}.$$
We then bound $x^T Px = \sum_{i,j} P_{i,j} x_i x_j$ term by term to get α_2:
$$\alpha_2 = \sum_{i,j} \max\{P_{i,j} u_i u_j, P_{i,j} l_i l_j, P_{i,j} u_i l_j, P_{i,j} l_i u_j\}.$$
This ensures that $\{l_i \leq x_i \leq u_i\} \subseteq \{x^T Px \leq \alpha_2\}$. Hence, $\{Ax \leq b\} \subseteq \{x^T Px \leq \alpha_2\}$.
- **Computation of α_1.** For $i = 1, \ldots, m$, we compute a scalar η_i by solving the convex program
$$\eta_i := \min_x \{a_i^T x : x^T Px \leq 1\},$$
where a_i is the i-th row of the constraint matrix A. This problem has a closed form solution:
$$\eta_i = -\sqrt{a_i^T P^{-1} a_i}.$$
Note that P^{-1} exists since $P \succ 0$. We then let
$$\alpha_1 = \min_i \{\frac{b_i^2}{\eta_i^2}\}.$$
This ensures that for each i, the minimum of $a_i^T x$ over $\{x^T Px \leq \alpha_1\}$ is larger or equal to b_i. Hence, $\{x^T Px \leq \alpha_1\} \subseteq \{Ax \leq b\}$.
We finally observe that all computations described above can be carried out in polynomial time.

■

IV. Finding an Invariant Inner Ellipsoid by Semidefinite Programming

In this section, we describe a semidefinite programming (SDP) based algorithm to obtain upper bounds on the optimal value of an R-LD-LP when $\rho(G) < 1$. Although we have already presented a polynomial time algorithm for this task in the previous section, an algorithm for finding upper bounds is valuable in two regards.

- First, in a practical situation, we may not want to do all the computation needed to find the integer r in (11) and

then solve an LP that adds cuts all the way to level r. Instead, we may want to only go through the first few levels of the LP and obtain a lower bound on the optimal value of R-LD-LP. If we could also get our hands on an upper bound from another algorithm, then we would have an idea of how far we are from optimality. If we are happy with the gap, we could just stop there.
- Second, when we extend our problem in [4] to the case where the dynamics is still linear, but uncertain and time-varying, then it is hopeless to look for an exact polynomial time algorithm of the type presented in Section III. Nevertheless, the SDP-based algorithm outlined below goes through and allows us to obtain upper bounds.

In addition to these reasons, the convexification tricks [4] used to arrive at the SDP are in our opinion quite interesting and could be of value elsewhere.

So what is our strategy for obtaining upper bounds on R-LD-LP? We will look for an ellipsoid that has following properties:

1) It is invariant under the dynamics $x_{k+1} = Gx_k$.
2) It is contained in the polytope $\{Ax \leq b\}$.
3) Among all ellipsoids that have the previous properties, this is one that gives the minimum value of $c^T x_0$ as x_0 ranges over the ellipsoid.

Here is a mathematical description of this optimization problem:

$$\begin{array}{l} \min_{x,P}\{c^T x : \\ P \succ 0, G^T P G \preceq P, x^T P x \leq 1, \\ [\forall z, z^T P z \leq 1 \implies Az \leq b\ ]\}. \end{array} \tag{14}$$

There are at least two issues with this formulation. First, the constraint $[\forall z, z^T P z \leq 1 \implies Az \leq b]$ needs to be rewritten to remove the universal quantifier. Second, the decision variables x and P are multiplying each other in the constraint $x^T P x \leq 1$, which is a problem. Nevertheless, one can get around these issues and formulate this problem *exactly* as an SDP. The proof is presented in [4], but we give the statement here in the case where the polytope $\{Ax \leq b\}$ is origin symmetric[2]. The main ingredients of the proof are the S-lemma [7], Schur complements, polar duality theory of convex sets, and duality of linear dynamical systems under transposition of the matrix G.

Theorem 4.1: In the case where the polytope $\{Ax \leq b\}$ is origin symmetric, the optimization problem in (14) can be rewritten as the following SDP:

$$\begin{array}{l} \min_{x,Q}\{c^T x : \\ Q \succ 0, GQG^T \preceq Q, \begin{pmatrix} Q & x \\ x^T & 1 \end{pmatrix} \succeq 0, a_i^T Q a_i \leq 1, i = 1, \ldots, m\}, \end{array} \tag{15}$$

where a_i is the transpose of the i-th row of A.

[2]Origin symmetry means that x is in the polytope if and only if $-x$ is in the polytope. This assumption can be removed at the expense of a more elaborate SDP [4] .

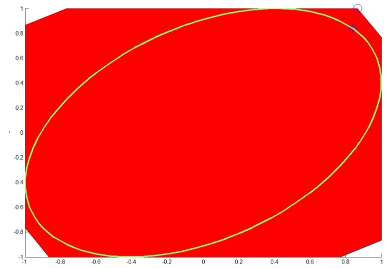

Fig. 4. An inner ellipsoid found by semidefinite programming; the optimal solution over the ellipsoid (*); the true optimal solution of R-LD-LP (o).

Example 4.1: Consider the following instance of R-LD-LP in $\mathbb{R}^2$:

$$A = \begin{bmatrix} 1 & 0 \\ -1 & 0 \\ 0 & 1 \\ 0 & -1 \end{bmatrix}, b = \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \end{bmatrix} \quad \text{and} \quad G = \frac{4}{5}\begin{bmatrix} \cos(\theta) & \sin(\theta) \\ -\sin(\theta) & cos(\theta) \end{bmatrix}, \tag{16}$$

with $\theta = \frac{\pi}{6}$ and $c = (-1, -1)^T$.

Figure 4 depicts the true feasible set of R-LD-LP, the invariant ellipsoid that approximates it from the inside, and the optimal points achieved by minimizing $c^T x$ over the two sets. The true optimal value of R-LD-LP is 1.6795 and the upper bound we are getting from the SDP in (15) is 1.8660.

References

[1] D. Bertsimas, D. B Brown, and C. Caramanis. Theory and applications of robust optimization. *SIAM review*, 53(3):464–501, 2011.

[2] A. Ben-Tal, L. El Ghaoui, and A. Nemirovski. *Robust optimization*. Princeton University Press, 2009.

[3] A. Bemporad and M. Morari. Robust model predictive control: A survey. In *Robustness in identification and control*, pages 207–226. Springer, 1999.

[4] A. A. Ahmadi and O. Günlük. An algorithmic study of robust-to-dynamics optimization. In preparation, 2015.

[5] I.P. Cornfeld, S.V. Fomin, and Y.G. Sinai. *Ergodic Theory*. Grundlehren der mathematischen Wissenschaften. Springer-Verlag New York, 1982.

[6] L. G Khachiyan. Polynomial algorithms in linear programming. *USSR Computational Mathematics and Mathematical Physics*, 20(1):53–72, 1980.

[7] I. Pólik and T. Terlaky. A survey of the S-lemma. *SIAM Review*, 49(3):371–418, 2007.