



HAL
open science

Convergent algorithms for a class of convex semi-infinite programs

Martina Cerulli, Antoine Oustry, Claudia d'Ambrosio, Leo Liberti

► **To cite this version:**

Martina Cerulli, Antoine Oustry, Claudia d'Ambrosio, Leo Liberti. Convergent algorithms for a class of convex semi-infinite programs. *SIAM Journal on Optimization*, inPress, 32 (4), pp.2493-2526. 10.1137/21M1431047 . hal-03339887v2

HAL Id: hal-03339887

<https://hal.science/hal-03339887v2>

Submitted on 22 Feb 2022

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

1 **CONVERGENT ALGORITHMS FOR A CLASS OF CONVEX SEMI-INFINITE**
2 **PROGRAMS***

3 MARTINA CERULLI[†], ANTOINE OUSTRY[‡], CLAUDIA D’AMBROSIO[§], AND LEO LIBERTI[§]

4 **Abstract.** We focus on convex semi-infinite programs with an infinite number of quadratically parametrized
5 constraints. In our setting, the lower-level problem, i.e., the problem of finding the constraint that is the most
6 violated by a given point, is not necessarily convex. We propose a new convergent approach to solve these semi-
7 infinite programs. Based on the Lagrangian dual of the lower-level problem, we derive a convex and tractable
8 restriction of the considered semi-infinite programming problem. We state sufficient conditions for the optimality of
9 this restriction. If these conditions are not met, the restriction is enlarged through an Inner-Outer Approximation
10 Algorithm, and its value converges to the value of the original semi-infinite problem. This new algorithmic approach
11 is compared with the classical Cutting Plane algorithm. We also propose a new rate of convergence of the Cutting
12 Plane algorithm, directly related to the iteration index, derived when the objective function is strongly convex, and
13 under a strict feasibility assumption. We successfully test the two methods on two applications: the constrained
14 quadratic regression and a zero-sum game with cubic payoff. Our results are compared to those obtained using the
15 approach proposed in [29], as well as using the classical relaxation approach based on the KKT conditions of the
16 lower-level problem.

17 **Key words.** Semi-infinite programming, Semidefinite programming, Cutting Plane, Convergent algorithms

18 **AMS subject classifications.** 90C34, 90C22, 90C46

19 **1. Introduction.** A Semi-Infinite Programming (SIP) problem is an optimization problem
20 with a finite number of decision variables, and an infinite number of parametrized constraints. In
21 this paper we consider a *standard* SIP problem, for which the parameter set is independent from
22 the variables, as opposed to the *generalized* SIP problem, where such set is allowed to depend on
23 the decision variables. We further assume that the SIP problem is convex with respect to (w.r.t.)
24 the decision variable x , and has infinitely many constraints which are quadratic and possibly non-
25 convex w.r.t. the parameter y . More precisely, we assume that the objective function $F(x)$ is
26 continuous and convex in x , where x is the array of decision variables, constrained to be in the
27 feasible set $\mathcal{X} \subset \mathbb{R}^m$. The constraint functions are convex in x and possibly non-convex quadratic
28 in the parameter y . The parameter set is the polytope

$$\mathcal{F} = \{y \in \mathbb{R}^n : Ay \leq b\} = \{y \in \mathbb{R}^n : \forall j \leq r, a_j^\top y \leq b_j\},$$

29 where r is an integer, A is a $r \times n$ matrix, a_j is the j -th row of the matrix A , and b a r -dimensional
30 vector. As already introduced, the set \mathcal{F} does not depend on x , i.e., the standard SIP problem is
31 considered. The Mathematical Programming (MP) formulation we study is as follows:

$$(SIP) \quad \begin{cases} \min_{x \in \mathcal{X}} & F(x) \\ \text{s.t.} & h(x) \leq \frac{1}{2}y^\top Q(x)y + q(x)^\top y \quad \forall y \in \mathcal{F}, \end{cases}$$

32 where $F(x)$ and $h(x)$ are continuous convex functions in the variables x , and both the $n \times n$ matrix
33 $Q(x)$ and the n -dimensional vector $q(x)$ depend linearly on x . We remark that $-h(x)$ may be
34 interpreted as the constant term of the quadratic function in y , in the right hand side. Being 0_n

*This research was partly funded by the European Union’s Horizon 2020 research and innovation program under the Marie Skłodowska-Curie grant agreement n. 764759 ETN “MINOA”.

[†]ESSEC Business School of Paris, Cergy-Pontoise, France (cerulli@essec.edu)

[‡]École des Ponts, 77455, Marne-la-Vallée, France.

[§]LIX - CNRS, École Polytechnique, Institut Polytechnique de Paris, 91120, Palaiseau, France (oustry@lix.polytechnique.fr, dambrosio@lix.polytechnique.fr, liberti@lix.polytechnique.fr)

38 the $n \times n$ null matrix, we further introduce the following matrices

$$\begin{aligned}
 39 \quad & \bullet \mathcal{Q}(x) = \frac{1}{2} \begin{pmatrix} Q(x) & q(x) \\ q(x)^\top & 0 \end{pmatrix}, \\
 40 \quad & \bullet \mathcal{G}(x) = \frac{1}{2} \begin{pmatrix} Q(x) & q(x) \\ q(x)^\top & -2h(x) \end{pmatrix} = \mathcal{Q}(x) - \begin{pmatrix} 0_n & 0 \\ 0 & h(x) \end{pmatrix}, \\
 41 \quad & \bullet \mathcal{A}_j = \frac{1}{2} \begin{pmatrix} 0_n & a_j \\ a_j^\top & 0 \end{pmatrix}, \quad \forall j \in \{1, \dots, r\},
 \end{aligned}$$

42 and the set $\mathcal{P} = \left\{ M(y) = \begin{pmatrix} yy^\top & y \\ y^\top & 1 \end{pmatrix} : y \in \mathcal{F} \right\} \subset \mathbb{R}^{(n+1) \times (n+1)}$, that we will use in the paper to
 43 obtain different formulations of problem (SIP). Here are the assumptions we make on (SIP).

44 ASSUMPTION 1. *The objective function $F(x)$ is convex and J -Lipschitz continuous on \mathcal{X} .*

45 ASSUMPTION 2. *\mathcal{X} is convex and compact.*

46 ASSUMPTION 3. *The functions $q(x)$ and $Q(x)$ are linear, i.e., the function $\mathcal{Q}(x)$ is linear.*

47 ASSUMPTION 4. *The function $h(x)$ is convex and Lipschitz continuous on \mathcal{X} .*

48 ASSUMPTION 5. *The set \mathcal{F} is compact, and a scalar $\rho > 0$ is known such that the set \mathcal{F} is
 49 included in the centered l_2 -ball with radius ρ .*

50 Assumptions 1, 2, 3 and 4 guarantee that the SIP problem is convex.

51 In the following, given a formulation (P) of an optimization problem, we denote its optimal
 52 value by $\text{val}(\text{P})$, and we will use the term *reformulation* to describe a formulation having the same
 53 set of optima of (P), i.e., what is defined as *exact reformulation* in [26, Definition 10]. With the
 54 term *relaxation*, we will refer to a formulation having a feasible set which contains the feasible set
 55 of (P) [26, Definition 13]. Instead, we will use the term *restriction* when referring to a formulation
 56 having a feasible set which is included in the feasible set of (P). Finally, a formulation (P) is defined
 57 *finite* when it has a finite number of variables and constraints.

58 As detailed in [38], the key to the theoretical as well as algorithmic handling of SIP problems
 59 lies in their bilevel structure. Indeed, the set of infinitely many parametrized constraints (SIP) is
 60 equivalent to $0 \leq \phi(x)$, where $\phi(x) = \min_{y \in \mathcal{F}} g(x, y)$ is the so-called *value function*. This allows
 61 writing the constraints in problem (SIP) as the *lower-level problem* of a bilevel program, as long
 62 as $g(x, y) = \frac{1}{2} y^\top Q(x) y + q(x)^\top y - h(x)$. In contrast to the *upper-level problem* which consists in
 63 minimizing $F(x)$ over the feasible set $\{x \in \mathcal{X} \mid 0 \leq \phi(x)\}$, in the lower-level problem x plays the
 64 role of an m -dimensional parameter, and y is the decision variable. Given this equivalence, in the
 65 following, we will refer to the lower level and the upper level of the bilevel reformulation of (SIP)
 66 as “the lower level” and “the upper level” of (SIP), respectively. Whereas the upper level of (SIP)
 67 is convex under the assumptions above, the lower level is not necessarily convex w.r.t. y .

68 Our first contribution is a new convergence rate of a classical Cutting Plane (CP) algorithm
 69 [8, 12, 22, 43] solving (SIP). While such algorithm and its convergence are well known in SIP, we de-
 70 rive a new convergence rate in terms of the number of iterations, under the additional assumptions
 71 that $F(x)$ is strongly convex and that there exists an upper-level solution strictly satisfying the
 72 constraint involving the lower-level problem. As a second contribution, we propose a new approach
 73 to solve problem (SIP). A tractable restriction with a finite number of constraints is obtained by
 74 dualizing, using Lagrangian duality and Semidefinite Programming (SDP), the problem $\min_{y \in Y} g(x, y)$,
 75 i.e., the problem of finding the most violated constraint among the infinite number of constraints
 76 of the corresponding SIP problem. If $g(x, y)$ is convex in y , i.e., if $Q(x)$ is positive semidefinite

77 (PSD), the obtained formulation is not a restriction, but a reformulation of (SIP). The dualiza-
78 tion technique has been used in the SIP literature in [10, 25], and inspired by approaches from
79 robust optimization [4, Section 1.3-1.4]. However, SDP together with Lagrangian duality has never
80 been used to dualize a quadratic programming lower-level problem to the best of our knowledge.
81 Moreover, the finite single-level formulation we obtain with this approach is convex, contrary to
82 the pre-existing methods in [10, 25]. When $g(x, y)$ is not convex in y , we still have convergence
83 guarantees. Indeed, we introduce a new Inner-Outer Approximation (IOA) algorithm, that pro-
84 gressively enlarges the restriction set so as to generate a sequence of feasible points the values of
85 which converge to the value of (SIP).

86 The rest of the paper is organized as follows. We review the relevant literature in Section 2. A
87 CP algorithm for solving formulation (SIP) is presented in Section 3, and a new rate of convergence
88 is derived in Subsection 3.1. A finite restriction/reformulation of problem (SIP) is introduced and
89 discussed in Section 4, in order to present a new convergent algorithm, the IOA algorithm, in
90 Subsection 4.5. Applications are introduced in Section 5. Numerical results, obtained by applying
91 both solution approaches to these applications, are presented in Section 6: our results, compared
92 to those obtained by solving our formulations with the algorithm proposed in [29], illustrate the
93 interest of the proposed method. Finally, Section 7 concludes the paper.

94 **2. Literature review.** Despite the difficulty in solving SIP problems for their infinitely many
95 constraints, many algorithms have been proposed in literature [11, 19, 31]. Most of them consist in
96 generating a sequence of finite problems, with different techniques.

97 The discretization approach [18, 30, 41] consists in replacing the infinite constraint parameter set
98 by a finite subset which samples it finely: this leads to a relaxation of the original problem, the value
99 of which converges towards the value of the original problem when the mesh gets finer. This method
100 is commonly used for parameters sets of low dimensions, but faces the curse of dimensionality when
101 the number of parameters increases. Indeed, the cost per iteration increases drastically as the size
102 of the considered finite subset grows. When the considered finite subset of constraints is increased
103 at each iteration by adding the most violated constraint, such discretization method for convex
104 problems corresponds to the Kelley algorithm [22], also known as cutting plane algorithm.

105 Reduction based methods [19, 31], under some strong assumptions, replace the infinitely many
106 SIP constraints by finitely many constraints which locally are sufficient to describe the SIP feasible
107 set. In order to do this, all the local minima of the lower-level problem must be computed, and this
108 is the bottleneck of this type of method.

109 Interior point methods for solving linear or convex SIP problems are suggested in [35, 37], and
110 for solving SIP problems with convex lower level in [40]. In [21], proximal penalty approaches have
111 been proposed, and a deletion procedure of inactive constraints is suggested.

112 A further class of methods is the so-called exchange method family [17, 31, 24, 47]. The notion
113 “exchange algorithm” refers to the fact that in every step some new constraints are added and
114 some of the old constraints may be deleted, i.e., an exchange of constraints takes place. When no
115 constraint is deleted and only one new constraint is added at each iteration, the exchange method
116 can be seen as a CP algorithm [22]. A generalized CP algorithm designed to solve SIP problems
117 is described in [8], and its convergence is established in the general case of continuous functions
118 and compact feasible set. In [43], a CP algorithm for solving a convex SIP problems with a strictly
119 convex quadratic objective function is presented. The relaxed CP approach for solving convex
120 quadratic SIP problems is studied in [12]. At each iteration, an approximate solution is computed,
121 using “inexact” minimizers for generating new cuts. Extending the central CP algorithm proposed
122 in [15] for solving linear SIP problems, a central CP method is proposed in [23] for convex SIP

123 problems, ensuring a linear rate of convergence w.r.t. the values of the objective function. In [6],
 124 an acceleration procedure of the central CP algorithm is proposed for linear SIP programs and a
 125 faster convergence rate is obtained.

126 Some methods overestimate the optimal objective value of the lower level by solving a restriction
 127 of the SIP problem. These restrictions are such that they are finite, or at least easy to reformulate
 128 to finite problems. In [7], the first deterministic algorithm for the global optimization of non-convex
 129 SIP problems is proposed. The upper-level variables are decided by a branch-and-bound algorithm.
 130 The lower bound is given by a discretization-based relaxation. The upper bound is obtained via
 131 natural interval extension underestimating the value of the lower-level problem. This yields a
 132 restriction of the SIP problem, the feasible points of which are also feasible for the SIP problem
 133 (and, thus, provide an upper bound to the optimal objective value). The key assumption for a finite
 134 convergence of this algorithm to an approximate optimal solution is the assumptions that there are
 135 Slater points in the SIP problem. Later on, under the same assumption and assuming continuity
 136 of the involved functions, a method to generate feasible points of the SIP problem is proposed
 137 in [29], solving restrictions of the discretization-based relaxations. A converging upper bounding
 138 procedure similar to the strategy proposed by [8] is used, and combined with an outer approximation
 139 of the feasible set. This generates infeasible iterates giving rigorous lower bounds to the optimal
 140 objective value. Recently, a branch-and-bound algorithm for the solution of SIP problems with a
 141 box-constrained lower level was proposed in [28]. In [13, 39], a convexification method is proposed
 142 which adaptively constructs convex relaxations of the lower-level problem, replaces the relaxed
 143 lower-level problems equivalently by their KKT conditions, and solves the resulting mathematical
 144 restrictions with complementarity constraints. This approximation produces feasible solutions for
 145 the original problem, under the continuity assumption and the existence of a Slater point in the
 146 SIP problem.

147 Another class of algorithms for SIP is based on Lagrangian penalty functions and Trust-Region
 148 methods [9, 42]. However, in the context of problem (SIP), as for the reduction based methods,
 149 they would require to compute all the local minima of problem $\min_{y \in Y} g(x, y)$. In the case where g is
 150 not convex in y , the enumeration of all local minima is intractable even for medium-scale instances.

151 Lower-level duality has been already used in [10, 25], leveraging on approaches from robust
 152 optimization [5]. However, contrary to what is proposed in this paper, the existing dual approaches
 153 lead to non-convex problems, and do not use SDP. In [10], several strategies are used to reformulate
 154 generalized SIP problems into non-convex finite minimization problems by exploiting Wolfe duality
 155 for the convex lower-level problems. In [25], the authors tackle generalized SIP problems where
 156 the convex quadratic lower-level problem has a fixed Hessian matrix Q , which does not depend on
 157 the variable x . Instead, in the present paper, we consider standard SIP problems with a linear
 158 function $Q(x)$ for the lower-level Hessian (as stated in Assumption 3). Back to [25], the authors use
 159 the Lagrangian dual of the lower level to obtain a non-convex restriction with a finite number of
 160 variables and constraints. In the latter problem, the convex envelopes of the non-convex functions
 161 in the objective and constraints can be easily computed. As a consequence, an approximate solution
 162 of the original problem can be obtained by solving finitely many convex problems.

163 In this paper, we prove a new convergence rate for the classical CP algorithm [22, 32] to solve
 164 (SIP) in the case where the objective is strongly convex, and under a strict feasibility condition.
 165 Our convergence rate is directly related to the iteration index k , which is something new w.r.t.
 166 what is usually proved in SIP literature, where the linear rate of convergence is related to an index
 167 that is independent of the iteration k (see [31, Theorem 4.3]). Furthermore, we exploit SDP and
 168 Lagrangian duality (we use the so-called *lower-level dualization approach*) to obtain a convex single-

169 level restriction of problem (SIP). If the lower level is convex in y for any value of x , the obtained
 170 formulation is a reformulation of (SIP). We further prove a sufficient condition on an optimal
 171 solution \bar{x} of this single-level formulation, which can be checked *a posteriori* to state that \bar{x} is an
 172 optimal solution of problem (SIP). We finally present a new algorithm based on the *lower-level*
 173 *dualization approach*, called IOA algorithm, that generates a sequence of feasible solutions of (SIP),
 174 the values of which are proved to converge to $\text{val}(\text{SIP})$.

175 **3. Cutting plane algorithm.** We detail in this section a CP algorithm for solving formula-
 176 tion (SIP). We also include a proof of convergence for this algorithm in Appendix A, as well as a
 177 convergence rate in Section 3.1, obtained by introducing a dual view of the CP algorithm. While
 178 CP methods and their convergence are broadly known in the SIP literature, the convergence rate
 179 we prove under stricter assumptions is something new w.r.t. the state of the art.

Algorithm 3.1 CP algorithm for (SIP)

Input: $\epsilon \geq 0$

0: Let $k \leftarrow 0$.

1: **while** true **do**

2: Being y^ℓ the solution of the lower-level problem solved at iteration ℓ , solve the problem

$$(R_k) \quad \begin{cases} \min_{x \in \mathcal{X}} & F(x) \\ \text{s.t.} & h(x) \leq \frac{1}{2}(y^\ell)^\top Q(x)y^\ell + q(x)^\top y^\ell, \quad \ell \in \{0, \dots, k-1\} \end{cases}$$

obtaining a solution x^k .

3: Compute an optimal solution y^k of the lower-level problem for $x = x^k$.

4: **if** $h(x^k) \leq \frac{1}{2}(y^k)^\top Q(x^k)y^k + q(x^k)^\top y^k + \epsilon$ **then**

5: Return (x^k, y^k) .

6: **else**

7: $k \leftarrow k + 1$

8: **end if**

9: **end while**

180 At the first iteration of Algorithm 3.1, the relaxed problem (R_0) is given by $\min_{x \in \mathcal{X}} F(x)$,
 181 which considers minimizing the objective function without any constraint parametrized by y . This
 182 problem has a finite value according to the compactness of set \mathcal{X} . At each iteration, Algorithm 3.1
 183 defines the feasible set of the upper-level problem by means of cuts in the variables x . The resulting
 184 (R_k) problems are relaxations of (SIP), and their feasible sets are decreasing in the sense of the
 185 inclusion, bounded because included in the feasible set of R_0 , and closed as intersections of closed
 186 sets. Thus, each problem (R_k) admits a minimum. Moreover, the sequence $F(x^k)$ is increasing,
 187 and $F(x^k) \leq \text{val}(\text{SIP})$ holds for any k . At step 3, the problem solved to find a new cut is

188 $(P_{x^k}) \quad \min_{y \in \mathbb{R}^n} \left\{ \frac{1}{2}y^\top Q(x^k)y + q(x^k)^\top y \mid Ay \leq b \right\}.$

189 This problem is a quadratic program that is either convex or non-convex depending on the positive
 190 semi-definiteness of the matrix $Q(x^k)$. In order to find global optima of (P_{x^k}) , regardless of the
 191 definiteness of $Q(x^k)$ (in turn depending on the value of x^k), a global optimization algorithm should
 192 be employed. Step 5 returns the optimal solution of formulation (SIP). The reader is referred to
 193 Appendix A for a proof of the convergence of this CP algorithm.

194 **3.1. A convergence rate for the CP algorithm.** In this section, we give a convergence
 195 rate of the CP Algorithm 3.1, under two additional assumptions on the SIP problem. The proofs
 196 of all the lemmata introduced in this section are in Appendix B.

197 First of all, using the notation introduced in Section 1, we remark that (SIP) can be formulated
 198 as

$$199 \text{ (SIP')} \quad \begin{cases} \min_{x \in \mathcal{X}} & F(x) \\ \text{s.t.} & 0 \leq \langle \mathcal{G}(x), Y \rangle, \forall Y \in \mathcal{P}. \end{cases}$$

200 We define $\mathcal{K} = \text{cone}(\mathcal{P}) \subset \mathbb{R}^{(n+1) \times (n+1)}$ as the convex cone generated by \mathcal{P} , and $\mathcal{L}(x, Y) = F(x) - \langle \mathcal{G}(x), Y \rangle$ as the Lagrangian function defined over $\mathcal{X} \times \mathcal{K}$. We remark that, $\forall x \in \mathcal{X}$, the following holds:

$$\sup_{Y \in \mathcal{K}} \mathcal{L}(x, Y) = \begin{cases} F(x) & \text{if } 0 \leq \langle \mathcal{G}(x), Y \rangle, \forall Y \in \mathcal{P} \\ +\infty & \text{else.} \end{cases}$$

201 Hence, problem (SIP') can be expressed as the saddle-point problem $\min_{x \in \mathcal{X}} \sup_{Y \in \mathcal{K}} \mathcal{L}(x, Y)$. At this
 202 point, we make the following further assumption.

203 **ASSUMPTION 6.** *The objective function $F(x)$ is μ -strongly-convex, i.e., $F(x) - \frac{\mu}{2}\|x\|^2$ is convex.*

204 Assumption 6 is quite strong, but we remark that, if the original objective function is just convex, it
 205 is always possible to enforce this assumption by “regularizing” the SIP problem adding a ℓ_2 penalty
 206 to the primal objective function, i.e., minimizing $F(x) + \frac{\mu}{2}\|x\|^2$ instead of $F(x)$. The Lagrangian
 207 function $\mathcal{L}(x, Y)$ is linear (thus, continuous and concave) w.r.t. Y for all $x \in \mathcal{X}$ and is continuous
 208 and convex w.r.t. x for all $Y \in \mathcal{K}$. The convexity w.r.t. x follows from Assumptions 1–4 and from
 209 the fact that $Y_{n+1, n+1} \geq 0$ for any $Y \in \mathcal{K}$. Since the set \mathcal{X} is convex (Assumption 2) and the set
 210 \mathcal{K} is convex too, Sion’s minimax theorem [36] is applicable and the following holds:

$$211 \quad \min_{x \in \mathcal{X}} \sup_{Y \in \mathcal{K}} \mathcal{L}(x, Y) = \sup_{Y \in \mathcal{K}} \min_{x \in \mathcal{X}} \mathcal{L}(x, Y).$$

212 Defining the dual function $\theta(Y) = \min_{x \in \mathcal{X}} \mathcal{L}(x, Y)$, we know that

$$213 \text{ (3.1)} \quad \text{val(SIP')} = \sup_{Y \in \mathcal{K}} \theta(Y).$$

214 Notice that the dual function $\theta(Y)$ is concave, as a minimum of linear functions in Y . As a
 215 direct application of [20, Corollary VI.4.4.5], the dual function $\theta(Y)$ is differentiable because of
 216 the uniqueness of $\arg \min_{x \in \mathcal{X}} \mathcal{L}(x, Y)$, which is, in turn, a consequence of the strong convexity of
 217 $\mathcal{L}(x, Y)$ w.r.t. x that follows from Assumption 6. Moreover, the gradient of the dual function is
 218 $\nabla \theta(Y) = -\mathcal{G}(x)$, where $x = \arg \min_{x \in \mathcal{X}} \mathcal{L}(x, Y)$. The differentiability of θ implies, in particular, that
 219 θ is continuous. We prove now that we can replace the sup operator with the max operator in the
 220 formulation (3.1), under the following additional assumption.

221 **ASSUMPTION 7.** *There exists $\hat{x} \in \mathcal{X}$, s.t., $\forall y \in \mathcal{F}$, $g(\hat{x}, y) = \frac{1}{2}y^\top Q(\hat{x})y + q(\hat{x})^\top y - h(\hat{x}) > 0$.*

222 **LEMMA 3.1.** *Under Assumption 7, the dual problem of (SIP') has an optimal solution Y^* .*

223 *Proof.* Proof in Appendix B.1. □

224 According to this lemma, the dual version of problem (SIP'), thus, reads

$$225 \text{ (DSIP)} \quad \max_{Y \in \mathcal{K}} \theta(Y).$$

226 This concave maximization problem on the convex cone \mathcal{K} is the Lagrangian dual of the prob-

227 lem (SIP'). Indeed, in this section, we are dualizing the whole problem (SIP), contrary to Section 4,
 228 where we will dualize the lower-level problem only. We are now going to see that the CP algorithm
 229 3.1 can be interpreted, from a dual perspective, as a cone constrained Fully Corrective Frank-Wolfe
 230 (FCFW) algorithm [27] solving the dual problem (DSIP). We prove that, during the execution of
 231 the CP Algorithm 3.1, the dual variables obtained when solving the relaxation (R_k) instantiate
 232 the iterates of a FCFW algorithm. In the following, the sets $B_k \subset \mathbb{R}^{(n+1) \times (n+1)}$ are finite sets,
 233 composed of rank-one matrices of the form $M(y)$.

234 The initialization of the CP can be seen, in the dual perspective, as the initialization of the
 235 FCFW algorithm, with $B_0 \leftarrow \emptyset$ and $Y^0 = 0$. Then, the generic iteration k is described in Table 1.

	Primal perspective: CP	Link	Dual perspective: FCFW
Step 1	Solve (R_k) and store the solution x^k	Duality	Solve the dual problem on $\text{cone}(B_k)$, i.e., $\max_{Y \in \text{cone}(B_k)} \theta(Y)$, and store the solution Y^k , the associated x^k and the gradient $\nabla\theta(Y^k) = -\mathcal{G}(x^k)$
Step 2	Solve the lower-level problem (P_{x^k}) $\min_{y \in \mathcal{F}} \frac{1}{2}y^\top Q(x^k)y + q(x^k)^\top y$ and store the solution y^k	$Z^k = M(y^k)$	Solve the problem $\max_{Z \in \mathcal{P}} \langle \nabla\theta(Y^k), Z \rangle$ and store the solution Z^k
Step 3a	If $h(x^k) \leq \frac{1}{2}(y^k)^\top Q(x^k)y^k + q(x^k)^\top y^k + \epsilon$, Return (x^k, y^k)	Reformulation	If $\langle \nabla\theta(Y^k), Z^k \rangle \leq \epsilon$, Return (Y^k, x^k)
Step 3b	If $h(x^k) > \frac{1}{2}(y^k)^\top Q(x^k)y^k + q(x^k)^\top y^k + \epsilon$, build (R_{k+1}) as (R_k) with the additional ineq. $h(x) \leq \frac{1}{2}(y^k)^\top Q(x)y^k + q(x)^\top y^k$.	Reformulation	If $\langle \nabla\theta(Y^k), Z^k \rangle > \epsilon$, set $B_{k+1} \leftarrow B_k \cup \{Z^k\}$.

Table 1: The k -th iteration of the CP (Algorithm 3.1), and of the FCFW algorithm

236 The different steps summarized in Table 1 can be explained as follows:

- 237 • *Step 1*: At iteration k , the set B_k represents, from a dual perspective, the set of CPs in
 238 the primal relaxation (R_k). The dual problem of (R_k) is in fact a restriction of (DSIP) on
 239 $\text{cone}(B_k)$, which is a polyhedral subcone of \mathcal{K} , since the following holds:

$$\begin{aligned}
 \max_{Y \in \text{cone}(B_k)} \theta(Y) &= \max_{Y \in \text{cone}(B_k)} \min_{x \in \mathcal{X}} (F(x) - \langle \mathcal{G}(x), Y \rangle) \\
 &= \min_{x \in \mathcal{X}} \max_{Y \in \text{cone}(B_k)} (F(x) - \langle \mathcal{G}(x), Y \rangle) \\
 &= \min_{x \in \mathcal{X}} \{F(x) \text{ s.t. } 0 \leq \langle \mathcal{G}(x), Z \rangle, \forall Z \in B_k\},
 \end{aligned}$$

241 which we recognize being the master problem (R_k). The absence of duality gap is, also
 242 in this case, a direct application of Sion's Theorem [36]. The new dual solution Y^k is
 243 obtained solving this restriction of (DSIP) on $\text{cone}(B_k)$, and the primal solution $x^k =$
 244 $\arg \min_{x \in \mathcal{X}} \mathcal{L}(x, Y^k)$ gives the gradient of the dual function in Y^k , i.e., $\nabla\theta(Y^k) = -\mathcal{G}(x^k)$.

- 245 • *Step 2*: Finding the SIP constraint that is the most violated by x^k is equivalent to finding

246 the furthest point of \mathcal{P} in the direction $\nabla\theta(Y^k)$. Indeed, the following holds:

$$247 \quad (3.2) \quad \max_{Z \in \mathcal{P}} \langle \nabla\theta(Y^k), Z \rangle = -\min_{Z \in \mathcal{P}} \langle \mathcal{G}(x^k), Z \rangle$$

$$248 \quad (3.3) \quad = -\min_{y \in \mathcal{F}} \left\{ \frac{1}{2} y^\top Q(x^k) y + q(x^k)^\top y - h(x^k) \right\},$$

249 and any optimal solution Z^k in problem (3.2) has the form $Z^k = M(y^k)$, with y^k optimal
250 in problem (3.3).

251 • *Step 3a:* The CP feasibility test $h(x^k) \leq \frac{1}{2}(y^k)^\top Q(x^k)y^k + q(x^k)^\top y^k + \epsilon$ is equivalent to the
252 dual optimality condition $\langle \nabla\theta(Y^k), Z^k \rangle \leq \epsilon$, according to the equality $\nabla\theta(Y^k) = -\mathcal{G}(x^k)$.

253 • *Step 3b:* Increasing the set of atoms $B_{k+1} \leftarrow B_k \cup \{Z^k\}$ is the *dual* of adding the corre-
254 sponding cut (with y^k s.t. $Z^k = M(y^k)$) to (R_k) , which creates the relaxation (R_{k+1}) .

255 The following lemma states a property of the iterates Y^k for all k .

256 LEMMA 3.2. *For any $k \in \mathbb{N}$, $\langle \nabla\theta(Y^k), Y^k \rangle = 0$.*

257 *Proof.* Proof in Appendix B.2. □

259 Based on the dual interpretation of the CP algorithm, we are now going to state a convergence rate
260 for this algorithm. We begin with two technical lemmata.

LEMMA 3.3. *It exists $L > 0$ s.t. function θ is L -smooth, i.e., for all $Y, Y' \in \mathcal{K}$,*

$$\|\nabla\theta(Y) - \nabla\theta(Y')\|_2 \leq L\|Y - Y'\|_2,$$

261 *which means that function $\nabla\theta$ is L -Lipschitz continuous.*

262 *Proof.* Proof in Appendix B.3. □

263 The following lemma is a consequence of the L -smoothness of θ .

264 LEMMA 3.4. *Being L the smoothness constant associated with θ , for any $Y, Z \in \mathcal{K}$ and $\gamma \geq 0$,*

$$265 \quad \theta(Y + \gamma Z) \geq \theta(Y) + \gamma \langle \nabla\theta(Y), Z \rangle - \frac{L\|Z\|^2}{2} \gamma^2.$$

266 *Proof.* Proof in Appendix B.4. □

267 We define the constant $T = \max_{Z \in \mathcal{P}} \|Z\|^2$. According to Lemma 3.1, (DSIP) admits an optimal

268 solution Y^* . We define τ as the last element of the optimal dual solution Y^* , i.e., $\tau = Y_{n+1, n+1}^*$.

269 This scalar plays a central role in the convergence rate analysis, conducted in the following theorem.

270 THEOREM 3.5. *Under Assumptions 1-7, if Algorithm 3.1 executes iteration $k \in \mathbb{N}$, then*

$$271 \quad (3.4) \quad \delta_k \leq \frac{2LT\tau^2}{k+2},$$

272 *where δ_k is the objective error $\text{val}(\text{SIP}) - F(x^k) \geq 0$.*

273 *Proof.* We emphasize that at each iteration k , $\theta(Y^k) = F(x^k)$, thus δ_k may also be seen as the
274 optimality gap in the dual problem (DSIP) (i.e., $\delta_k = \text{val}(\text{SIP}) - F(x^k) = \theta(Y^*) - \theta(Y^k)$). We prove
275 the inequality (3.4) by induction.

Base case: $k = 0$. Since θ is concave, we have that $\delta_0 = \theta(Y^*) - \theta(Y^0) \leq \langle \nabla\theta(Y^0), Y^* - Y^0 \rangle =$
 $\langle \nabla\theta(Y^0), Y^* \rangle$, with the last equality following from $Y^0 = 0$. We remark that $\langle \nabla\theta(Y^0), Y^* \rangle =$
 $\langle \nabla\theta(Y^0) - \nabla\theta(Y^*), Y^* \rangle$ since $\langle \nabla\theta(Y^*), Y^* \rangle = 0$ by optimality of Y^* . Hence,

$$\delta_0 \leq \langle \nabla\theta(Y^0) - \nabla\theta(Y^*), Y^* \rangle \leq \|\nabla\theta(Y^0) - \nabla\theta(Y^*)\| \|Y^*\|,$$

276 where the last inequality is the Cauchy-Schwarz inequality. Using the L -Lipschitzness of $\nabla\theta$
277 (Lemma 3.3), we know that $\|\nabla\theta(Y^0) - \nabla\theta(Y^*)\| \leq L\|Y^0 - Y^*\| = L\|Y^*\|$. Finally, we deduce

278 that, since $Y^* \in \tau \text{conv}(\mathcal{P})$, $\delta_0 \leq L\|Y^*\|^2 \leq LT\tau^2$.

279 *Induction.* We suppose that the algorithm runs $k + 1$ iterations, and that the property (3.4)
 280 is true for k . Using Lemma 3.4, we can compute a lower bound on the progress made during the
 281 iteration of index $k + 1$:

$$282 \quad \theta(Y^{k+1}) \geq \theta(Y^k + \gamma Z^k) \geq \theta(Y^k) + \gamma \langle \nabla \theta(Y^k), Z^k \rangle - \frac{L\|Z^k\|^2}{2} \gamma^2,$$

283 for any $\gamma \geq 0$. Multiplying by -1 , adding $\theta(Y^*)$ to both left and right hand sides of the above
 284 inequality, and using $\|Z^k\|^2 \leq T$, we have that

$$285 \quad (3.5) \quad \delta_{k+1} \leq \delta_k - \gamma \langle \nabla \theta(Y^k), Z^k \rangle + \frac{LT}{2} \gamma^2,$$

286 for any $\gamma \geq 0$. We remark that the value T is independent from k . By concavity of θ , $\delta_k = \theta(Y^*) -$
 287 $\theta(Y^k) \leq \langle \nabla \theta(Y^k), Y^* - Y^k \rangle$. By Lemma 3.2, we have $\langle \nabla \theta(Y^k), Y^k \rangle = 0$. Thus, $\delta_k \leq \langle \nabla \theta(Y^k), Y^* \rangle$.
 288 As $Y_{n+1, n+1}^* = \tau$, we know that $Y^* \in \tau \text{conv}(\mathcal{P})$, and, therefore,

$$289 \quad (3.6) \quad \delta_k \leq \max_{Z \in \tau \text{conv}(\mathcal{P})} \langle \nabla \theta(Y^k), Z \rangle = \max_{Z \in \tau \mathcal{P}} \langle \nabla \theta(Y^k), Z \rangle = \tau \langle \nabla \theta(Y^k), Z^k \rangle,$$

290 where the last equality follows from the definition of Z^k . Combining Eqs. (3.5) and (3.6), we obtain
 291 $\delta_{k+1} \leq \delta_k - \gamma \tau^{-1} \delta_k + \frac{LT}{2} \gamma^2$, for every $\gamma \geq 0$. Factoring and setting $\tilde{\gamma} = \gamma \tau^{-1}$ (for any $\tilde{\gamma} \geq 0$) yields:

$$292 \quad (3.7) \quad \delta_{k+1} \leq (1 - \tilde{\gamma}) \delta_k + \frac{LT\tau^2}{2} \tilde{\gamma}^2.$$

293 We have derived a lower bound on the optimality gap at iteration k . Applying Eq. (3.7) with
 294 $\tilde{\gamma} = \frac{2}{k+2}$, we obtain:

$$295 \quad \delta_{k+1} \leq \left(1 - \frac{2}{k+2}\right) \delta_k + \frac{LT\tau^2}{2} \frac{4}{(k+2)^2} \leq \frac{k}{k+2} \frac{2LT\tau^2}{k+2} + \frac{LT\tau^2}{2} \frac{4}{(k+2)^2},$$

297 with the second inequality coming from the application of (3.4), which holds for k by the induction
 298 hypothesis. Finally, we deduce that

$$299 \quad \delta_{k+1} \leq \frac{2LT\tau^2}{k+2} \left(\frac{k}{k+2} + \frac{1}{k+2}\right) \leq \frac{2LT\tau^2}{k+2} \frac{k+1}{k+2} \leq \frac{2LT\tau^2}{k+2} \frac{k+2}{k+3} = \frac{2LT\tau^2}{k+3},$$

301 where the third inequality follows from the observation that $\frac{k+1}{k+2} \leq \frac{k+2}{k+3}$. Hence, the property (3.4)
 302 is true for $k + 1$ as well. This concludes the proof. \square

303 We remark that the convergence rate defined in (3.4) is directly related to the iteration index k ,
 304 which is something different w.r.t. what is usually proved for existing CP algorithms solving SIP
 305 problems [6, 23, 31], where the rate of convergence is not directly controlled by k .

306 **4. Lower-level dualization approach and Inner-Outer approximation algorithm.** A
 307 possible way to deal with the SIP problem (SIP) is what we call *lower-level dualization approach*,
 308 which consists in replacing the constraint involving the quadratic lower-level problem with one
 309 involving its dual. In particular, we consider a strong dual of an SDP relaxation of the lower-level
 310 problem (or a reformulation if the latter is convex), which is something new w.r.t. the existing SIP
 311 literature. We recall that the lower-level problem of (SIP), for any $x \in \mathcal{X}$, reads:

$$312 \quad (\text{P}_x) \quad \begin{cases} \min_{y \in \mathbb{R}^n} & \frac{1}{2} y^\top Q(x) y + q(x)^\top y \\ \text{s.t.} & a_j^\top y \leq b_j, \quad \forall j \in \{1, \dots, r\}, \end{cases}$$

313 where the objective function $f(x, y) = \frac{1}{2} y^\top Q(x) y + q(x)^\top y$ is convex if $Q(x)$ is PSD. In Subsec-
 314 tion 4.1, we introduce the classical SDP relaxation (reformulation, if the lower level is convex) of the

315 lower-level problem regularized by a ball constraint, and, then, in Subsection 4.2, we introduce the
 316 SDP dual of this relaxation (reformulation, resp.). In Subsection 4.3 we present a single-level finite
 317 formulation, (SIPR), obtained applying the so-called *lower-level dualization approach* to the problem
 318 (SIP). This formulation is a reformulation of (SIP) if $Q(x)$ is PSD for any $x \in \mathcal{X}$. Otherwise, an *a*
 319 *posteriori* sufficient condition on a computed solution \bar{x} of (SIPR) introduced in Subsection 4.4 can
 320 be verified. If \bar{x} satisfies such condition, one can state that \bar{x} is an optimal solution of (SIP). If not,
 321 an IOA algorithm is proposed in Subsection 4.5, which generates a sequence of converging feasible
 322 solutions of (SIPR). The proofs of all the lemmata introduced in this section are in Appendix C.

323 **4.1. SDP relaxation/reformulation of the lower-level problem.** In this section, we
 324 reason for any fixed value of the decision vector $x \in \mathcal{X}$. We denote by $\langle A, B \rangle = \text{Tr}(A^\top B)$ the
 325 Froebenius product of two square matrices A and B with same size. Using the matrices \mathcal{Q} , and \mathcal{A}_j
 326 introduced in Section 1, under Assumption 5, the problem

$$327 \quad (4.1) \quad \left\{ \begin{array}{ll} \min_{Y \in \mathbb{R}^{(n+1) \times (n+1)}} & \langle \mathcal{Q}(x), Y \rangle \\ \text{s.t.} & \langle \mathcal{A}_j, Y \rangle \leq b_j \quad \forall j \in \{1, \dots, r\} \\ & \text{Tr}(Y) \leq 1 + \rho^2 \\ & Y_{n+1, n+1} = 1 \\ & Y \succeq 0 \\ & \text{rank}(Y) = 1, \end{array} \right.$$

is a reformulation of (P_x) , because any feasible matrix Y has the form $Y = \begin{pmatrix} y \\ 1 \end{pmatrix} \begin{pmatrix} y \\ 1 \end{pmatrix}^\top$ with $y \in \mathcal{F}$,
 and, therefore, $\langle \mathcal{Q}(x), Y \rangle = f(x, y)$. The constraint $\text{Tr}(Y) \leq 1 + \rho^2$, derives from Assumption 5 as
 follows:

$$\|y\|_2^2 \leq \rho^2 \Leftrightarrow \text{Tr}(yy^\top) \leq \rho^2 \Leftrightarrow \text{Tr}(Y) \leq \rho^2 + 1,$$

328 being $\text{Tr}(Y) = \text{Tr}(yy^\top) + 1$. This constraint does not play any role at this point, but will be useful
 329 thereafter to come up with a dual SDP problem with no duality gap (see Section 4.2). If we relax
 330 the non-convex constraint $\text{rank}(Y) = 1$ in problem (4.1), we obtain:

$$331 \quad (\text{SDP}_x) \quad \left\{ \begin{array}{ll} \min_{Y \in \mathbb{R}^{(n+1) \times (n+1)}} & \langle \mathcal{Q}(x), Y \rangle \\ \text{s.t.} & \langle \mathcal{A}_j, Y \rangle \leq b_j \quad \forall j \in \{1, \dots, r\} \\ & \text{Tr}(Y) \leq 1 + \rho^2 \\ & Y_{n+1, n+1} = 1 \\ & Y \succeq 0, \end{array} \right.$$

332 which is a SDP relaxation of (P_x) , as proved in the following Lemma 4.1. If $Q(x)$ is PSD, Lemma 4.1
 333 states that (SDP_x) has the same optimal objective function value of (P_x) , the rank-constraint
 334 relaxation notwithstanding.

335 **LEMMA 4.1.** *Under Assumption 5, $\text{val}(\text{SDP}_x) \leq \text{val}(P_x)$. If $Q(x)$ is PSD, $\text{val}(\text{SDP}_x) = \text{val}(P_x)$.*

336 *Proof.* Proof in Appendix C.1. □

337 **4.2. Dual SDP problem.** As already done in Section 4.1, also in this section we reason for
 338 any fixed value of $x \in \mathcal{X}$. Let E be a $(n+1) \times (n+1)$ matrix s.t. $E_{n+1, n+1} = 1$ and $E_{ij} = 0$
 339 everywhere else. Let I_{n+1} be the $(n+1) \times (n+1)$ identity matrix. The following SDP problem

$$340 \quad (\text{DSDP}_x) \quad \left\{ \begin{array}{ll} \max_{\lambda \in \mathbb{R}_+^r, \alpha \in \mathbb{R}_+, \beta \in \mathbb{R}} & -b^\top \lambda - \alpha(1 + \rho^2) - \beta \\ \text{s.t.} & \mathcal{Q}(x) + \sum_{j=1}^r \lambda_j \mathcal{A}_j + \alpha I_{n+1} + \beta E \succeq 0, \end{array} \right.$$

341 is the dual of problem (SDP_x) , as the following lemma states.

342 LEMMA 4.2. *Formulations (SDP_x) and (DSDP_x) are a primal-dual pair of SDP problems and*
 343 *strong duality holds, i.e., $\text{val}(\text{SDP}_x) = \text{val}(\text{DSDP}_x)$.*

344 *Proof.* Proof in Appendix C.2.

345 **4.3. SDP restriction/reformulation of the SIP problem.** Leveraging on Section 4.1
 346 and Section 4.2, which focus on the lower-level problem (P_x) , its SDP relaxation (SDP_x) and the
 347 respective dual problem (DSDP_x) , we propose a single-level finite restriction of problem (SIP) . It
 348 is a reformulation of (SIP) if $Q(x)$ is PSD for any $x \in \mathcal{X}$.

349 THEOREM 4.3. *The finite formulation*

$$350 \quad (\text{SIPR}) \quad \begin{cases} \min_{x, \lambda, \alpha, \beta} & F(x) \\ \text{s.t.} & x \in \mathcal{X} \\ & h(x) \leq -b^\top \lambda - \alpha(1 + \rho^2) - \beta \\ & Q(x) + \sum_j \lambda_j \mathcal{A}_j + \alpha I_{n+1} + \beta E \succeq 0 \\ & x \in \mathbb{R}^m, \lambda \in \mathbb{R}_+^r, \alpha \in \mathbb{R}_+, \beta \in \mathbb{R}, \end{cases}$$

351 *is a restriction of problem (SIP) . If $Q(x)$ is PSD for any $x \in \mathcal{X}$, it is a reformulation of (SIP) .*

352 *Proof.* Let $\text{Feas}(\text{SIP})$ and $\text{Feas}(\text{SIPR})$ be the feasible sets of (SIP) and (SIPR) , respectively. Since
 353 (SIP) and (SIPR) share the same objective function, proving for any $x \in \mathbb{R}^m$ the implication

$$354 \quad (4.2) \quad (\exists \lambda \in \mathbb{R}_+^r, \alpha \in \mathbb{R}_+, \beta \in \mathbb{R} : (x, \lambda, \alpha, \beta) \in \text{Feas}(\text{SIPR})) \implies x \in \text{Feas}(\text{SIP}),$$

355 will prove the first part of the theorem. For any $x \in \mathcal{X}$, we have:

$$356 \quad (4.3) \quad h(x) \leq \text{val}(\text{SDP}_x) \implies h(x) \leq \text{val}(\text{P}_x) \iff x \in \text{Feas}(\text{SIP}),$$

357 where the first implication stems from Lemma 4.1, which stipulates that $\text{val}(\text{SDP}_x) \leq \text{val}(\text{P}_x)$.
 358 Applying Lemma 4.2, we obtain that:

$$359 \quad (4.4) \quad h(x) \leq \text{val}(\text{SDP}_x) \iff h(x) \leq \text{val}(\text{DSDP}_x).$$

360 For any $x \in \mathcal{X}$, we have that

$$361 \quad (4.5) \quad h(x) \leq \text{val}(\text{DSDP}_x) \iff \exists \lambda \in \mathbb{R}_+^r, \alpha \in \mathbb{R}_+, \beta \in \mathbb{R} : \begin{cases} h(x) \leq -b^\top \lambda - \alpha(1 + \rho^2) - \beta \\ Q(x) + \sum_{j=1}^r \lambda_j \mathcal{A}_j + \alpha I_{n+1} + \beta E \succeq 0. \end{cases}$$

362 The equivalence (4.5) just expresses the fact that the maximization problem (DSDP_x) has a value
 363 exceeding $h(x)$ if and only if it has a feasible solution with value exceeding $h(x)$. Hence, from (4.4),
 364 and (4.5), the following equivalences hold:

$$365 \quad (4.6) \quad h(x) \leq \text{val}(\text{SDP}_x) \iff \exists \lambda \in \mathbb{R}_+^r, \alpha \in \mathbb{R}_+, \beta \in \mathbb{R} : \begin{cases} h(x) \leq -b^\top \lambda - \alpha(1 + \rho^2) - \beta \\ Q(x) + \sum_{j=1}^r \lambda_j \mathcal{A}_j + \alpha I_{n+1} + \beta E \succeq 0 \end{cases}$$

$$366 \quad \iff \exists \lambda \in \mathbb{R}_+^r, \alpha \in \mathbb{R}_+, \beta \in \mathbb{R}, (x, \lambda, \alpha, \beta) \in \text{Feas}(\text{SIPR}).$$

367 The equivalence (4.6), together with implication (4.3), proves the implication (4.2).

368 If $Q(x)$ is PSD for any $x \in \mathcal{X}$, we can replace the implication (4.3) by the equivalence

$$369 \quad (4.7) \quad h(x) \leq \text{val}(\text{SDP}_x) \iff h(x) \leq \text{val}(\text{P}_x) \iff x \in \text{Feas}(\text{SIP}).$$

370 This, together with equivalence (4.6), proves that

$$371 \quad \exists \lambda \in \mathbb{R}_+^r, \alpha \in \mathbb{R}_+, \beta \in \mathbb{R} : (x, \lambda, \alpha, \beta) \in \text{Feas}(\text{SIPR}) \iff x \in \text{Feas}(\text{SIP}),$$

372 meaning that (SIPR) is a reformulation of (SIP) , since the objective function is the same. \square

375 Assumptions 1, 2, 3, and 4 imply that the single-level finite problem (SIPR) is convex.

376 **4.4. Optimality of the SDP restriction: a sufficient condition.** Theorem 4.3 states
 377 that, if $Q(x) \succeq 0$ for all $x \in \mathcal{X}$, the single-level finite formulation (SIPR) is an exact reformulation
 378 of the problem (SIP). In this section, we show that, even if this *a priori* condition is not satisfied,
 379 an *a posteriori* condition on the computed solution \bar{x} of (SIPR) enables us to state that \bar{x} is an
 380 optimal solution of (SIP).

381 **THEOREM 4.4.** *Let \bar{x} be a solution of the single-level formulation (SIPR). Assuming that*
 382 *$Q(\bar{x}) \succ 0$, then \bar{x} is optimal in (SIP).*

383 *Proof.* Given a closed convex set S , according to the definition [20, Def. III.5.1.1], the tangent
 384 cone to S at x (denoted by $T_S(x)$) is the set of directions $u \in \mathbb{R}^m$ such that it exists a sequence
 385 $(x_k)_{k \in \mathbb{N}}$ in S , and a positive sequence $(t_k)_{k \in \mathbb{N}}$ s.t. $t_k \rightarrow 0$ and $\frac{x_k - x}{t_k} \rightarrow u$. Moreover, according to
 386 the definition [20, Def. III.5.2.4], the normal cone $N_S(x)$ to S at x is the polar cone of the tangent
 387 cone $T_S(x)$, i.e., $N_S(x) = T_S(x)^\circ$. We define the closed convex set C (resp. \hat{C}) as the feasible set of
 388 formulation (SIP) (resp. (SIPR)).

389 Since $Q(\bar{x}) \succ 0$ and $\det Q(x)$ is continuous, it exists $r > 0$ s.t. for all x in the open ball of
 390 radius r with center \bar{x} (denoted by $B(\bar{x}, r)$), $Q(x) \succeq 0$. According to Lemma 4.1, this means that
 391 for all x in $\mathcal{X} \cap B(\bar{x}, r)$, $\text{val}(\mathbf{P}_x) = \text{val}(\mathbf{SDP}_x)$. Hence, we deduce that, for any $x \in \mathcal{X} \cap B(\bar{x}, r)$, x is
 392 feasible in (SIP) if and only if x is feasible in (SIPR). In other words, $C \cap B(\bar{x}, r) = \hat{C} \cap B(\bar{x}, r)$.
 393 According to the aforementioned definition of the tangent and normal cones, we further deduce
 394 that $T_C(\bar{x}) = T_{\hat{C}}(\bar{x})$, and $N_C(\bar{x}) = T_C(\bar{x})^\circ = T_{\hat{C}}(\bar{x})^\circ = N_{\hat{C}}(\bar{x})$.

395 We know that \bar{x} is optimal in (SIPR), i.e., $\bar{x} \in \arg \min_{x \in \hat{C}} F(x)$. Since F is a finite-valued convex
 396 function, and \hat{C} is a closed and convex set, Theorem [20, Th. VII.1.1.1] holds, and we can deduce
 397 that $0 \in \partial F(\bar{x}) + N_{\hat{C}}(\bar{x})$. Using the equality $N_C(\bar{x}) = N_{\hat{C}}(\bar{x})$, we have that $0 \in \partial F(\bar{x}) + N_C(\bar{x})$ too.
 398 Applying the same Theorem [20, Th. VII.1.1.1] with the closed and convex constraint set C , we
 399 know that $0 \in \partial F(\bar{x}) + N_C(\bar{x})$ implies that $\bar{x} \in \arg \min_{x \in C} F(x)$, meaning that \bar{x} is optimal in (SIP). \square

400 If this sufficient condition is satisfied, although solving a problem with a different feasible set,
 401 i.e., restriction (SIPR), a guarantee of global optimality for the original problem (SIP) is obtained.

402 **4.5. Inner-Outer Approximation algorithm.** If neither the lower level is convex, nor the
 403 sufficient optimality condition in Theorem 4.4 is satisfied, we do not directly obtain an optimal
 404 solution of the SIP problem by solving (SIPR). Yet, in this section, we present an algorithm based
 405 on the *lower-level dualization approach* that allows us to construct a sequence of feasible solutions
 406 of the SIP problem, the values of which converge to the SIP optimal value.

407 For $k \in \mathbb{N}^*$, we consider two finite sequences $x^1, \dots, x^{k-1} \in \mathcal{X}$ and $v_1, \dots, v_{k-1} \in \mathbb{R}$ s.t.
 408 $v_\ell = \text{val}(\mathbf{P}_{x^\ell})$. Since, for all $\ell = 1, \dots, k-1$, the inequality

$$409 \quad \forall y \in \mathcal{F}, \quad \frac{1}{2} y^\top Q(x^\ell) y + q(x^\ell)^\top y \geq v_\ell,$$

410 holds, the following SDP problem is still a relaxation of (\mathbf{P}_x) , for any $x \in \mathcal{X}$:

$$411 \quad (\mathbf{SDP}_x^k) \quad \begin{cases} \min_{Y \in \mathbb{R}^{(n+1) \times (n+1)}} & \langle Q(x), Y \rangle \\ \text{s.t.} & \langle \mathcal{A}_j, Y \rangle \leq b_j \quad \forall j \in \{1, \dots, r\} \\ & \langle Q(x^\ell), Y \rangle \geq v_\ell \quad \forall \ell \in \{1, \dots, k-1\} \\ & \text{Tr}(Y) \leq 1 + \rho^2 \\ & Y_{n+1, n+1} = 1 \\ & Y \succeq 0. \end{cases}$$

412 For ease of reading, in the following, we denote by $V_{\mathbf{P}}(x) = \text{val}(\mathbf{P}_x)$ the optimal value of prob-

413 lem (\mathbf{P}_x) , by $V_{\text{SDP}}(x) = \text{val}(\text{SDP}_x)$ the optimal value of problem (SDP_x) , and by $V_{\text{SDP}}^k(x) =$
 414 $\text{val}(\text{SDP}_x^k)$ the optimal value of problem (SDP_x^k) . With this notation $v_\ell = \text{val}(\mathbf{P}_{x^\ell}) = V_{\mathbf{P}}(x^\ell)$.
 415 We underline that the function $V_{\text{SDP}}^k(x)$ depends on the finite sequence $x_1, \dots, x_{k-1} \in \mathcal{X}$. Being
 416 η_ℓ the Lagrangian multiplier associated to the constraint $\langle Q(x^\ell), Y \rangle \geq v_\ell$, the strong SDP dual of
 417 problem (SDP_x^k) is

$$(4.8) \quad \begin{cases} \max_{\substack{\lambda \in \mathbb{R}_+^r, \alpha \in \mathbb{R}_+, \\ \beta \in \mathbb{R}, \eta \in \mathbb{R}_+^{k-1}}} & -b^\top \lambda - \alpha(1 + \rho^2) - \beta + \sum_{\ell=1}^{k-1} \eta_\ell v_\ell \\ \text{s.t.} & Q(x) + \sum_{j=1}^r \lambda_j \mathcal{A}_j + \alpha I_{n+1} + \beta E - \sum_{\ell=1}^{k-1} \eta_\ell Q(x^\ell) \succeq 0. \end{cases}$$

419 Hence, for any $\hat{x} \in \mathcal{X}$, $h(\hat{x}) \leq V_{\text{SDP}}^k(\hat{x})$ holds if and only if $\Omega^k(\hat{x}) \neq \emptyset$, where $\Omega^k(\hat{x})$ is defined as

$$(4.20) \quad \Omega^k(\hat{x}) := \{(\lambda, \alpha, \beta, \eta) \in \mathbb{R}_+^r \times \mathbb{R}_+ \times \mathbb{R} \times \mathbb{R}_+^{k-1} : h(\hat{x}) \leq -b^\top \lambda - \alpha(1 + \rho^2) - \beta + \sum_{\ell=1}^{k-1} \eta_\ell v_\ell \wedge \\
 (4.21) \quad Q(\hat{x}) + \sum_{j=1}^r \lambda_j \mathcal{A}_j + \alpha I_{n+1} + \beta E - \sum_{\ell=1}^{k-1} \eta_\ell Q(x^\ell) \succeq 0\}.$$

422
 423

Algorithm 4.1 IOA algorithm for **(SIP)**

Input: $\underline{\mu} > 0, \bar{\mu} > 0, d \geq 0, \epsilon \geq 0, k \leftarrow 0$

- 0: Solve the restriction **(SIPR)**, obtaining a solution \hat{x}^0 .
- 1: **if** $Q(\hat{x}^0) \succ 0$ **then**
- 2: Return \hat{x}^0 .
- 3: **else**
- 4: $k \leftarrow 1$.
- 5: **while true do**
- 6: Choose any $\mu_k \in [\underline{\mu}, \bar{\mu}]$.
- 7: Being Y^ℓ the extended solution of problem (\mathbf{P}_{x^ℓ}) , solve

$$(4.9) \quad \begin{cases} \min_{x, \hat{x}, \lambda, \alpha, \beta, \eta} & F(x) + F(\hat{x}) + \frac{\mu_k}{2} \|x - \hat{x}\|^2 \\ \text{s.t.} & h(x) \leq \langle Q(x), Y^\ell \rangle, \quad \ell \in \{1, \dots, k-1\} \\ & (\lambda, \alpha, \beta, \eta) \in \Omega^k(\hat{x}), \quad x, \hat{x} \in \mathcal{X}, \end{cases}$$

obtaining two solutions x^k, \hat{x}^k .

- 8: Solve (\mathbf{P}_{x^k}) obtaining a value v_k , a solution y^k , and an extended solution $Y^k = M(y^k)$.
 - 9: **if** $\|x^k - \hat{x}^k\| \leq d$ and $h(x^k) \leq v_k + \epsilon$ **then**
 - 10: Return \hat{x}^k .
 - 11: **else**
 - 12: $k \leftarrow k + 1$.
 - 13: **end if**
 - 14: **end while**
 - 15: **end if**
-

424 Algorithm 4.1 is the pseudocode of the IOA algorithm. It starts by solving the restriction
 425 **(SIPR)** and checks whether the condition presented in Theorem 4.4 is satisfied or not. If yes, the

426 algorithm stops returning the solution which is optimal for both (SIPR) and (SIP). Otherwise,
 427 it performs a sequence of iterations, until the stopping criteria are satisfied, i.e., $\|x^k - \hat{x}^k\| \leq d$
 428 and $h(x^k) \leq v_k + \epsilon$. At each iteration the convex optimization problem (4.9) is solved. This
 429 problem is a *coupling* between the minimization of F on a relaxed set and the minimization of F
 430 on a restricted set. Indeed, x belongs to an outer-approximation (relaxation), whereas \hat{x} belongs
 431 to an inner-approximation (restriction) of SIP feasible set, since \hat{x} satisfies $V_{\text{SDP}}^k(\hat{x}) \geq h(\hat{x})$. The
 432 minimization of F over these two sets is *coupled* by a proximal term that penalizes the distance
 433 between x and \hat{x} . After solving the master problem (4.9), the lower-level problem (P_x) is solved for
 434 $x = x^k$. The solution of such problem is used to restrict the outer-approximation, and to enlarge
 435 the inner-approximation. We are now going to prove the convergence of Algorithm 4.1. We begin
 436 with two technical lemmata.

437 LEMMA 4.5. *It exists $\zeta \in \mathbb{R}_+$ such that, for any sequence $(x^k)_{k \in \mathbb{N}^*} \in \mathcal{X}$, the value function*
 438 V_{SDP}^k *is ζ -Lipschitz over \mathcal{X} for all $k \in \mathbb{N}^*$. Moreover, for any sequence $(x^k)_{k \in \mathbb{N}^*} \in \mathcal{X}$:*

- 439 1. $\forall x \in \mathcal{X}, \quad V_{\text{SDP}}(x) = V_{\text{SDP}}^1(x) \leq V_{\text{SDP}}^2(x) \leq \dots \leq V_{\text{SDP}}^k(x) \leq V_{\text{P}}(x),$
- 440 2. $\forall \ell \leq k - 1, \quad V_{\text{SDP}}^k(x^\ell) = V_{\text{P}}(x^\ell) = v_\ell.$

441 *Proof.* Proof in Appendix C.3. □

442 LEMMA 4.6. *Under Assumptions 1–5, and denoting by x^* an optimal solution of problem (SIP),*
 443 *if Algorithm 4.1 runs iteration k , then $F(x^k) \leq F(x^*) + \mu_k(x^k - \hat{x}^k)^\top(x^* - x^k)$.*

444 *Proof.* Proof in Appendix C.4. □

445 Before proving Theorem 4.7, we must assume that Slater condition holds for the restric-
 446 tion (SIPR), even if there is no need to know the corresponding Slater point to run the algorithm.

447 ASSUMPTION 8. *It exists $x^S \in \mathcal{X}$ that is strictly feasible in (SIPR), i.e., $V_{\text{SDP}}(x^S) > h(x^S)$.*

448 Under Assumptions 1–5 and Assumption 8 the IOA algorithm converges, as stated in the
 449 following theorem.

450 THEOREM 4.7. *Under Assumptions 1–5 and Assumption 8, if $d = \epsilon = 0$, Algorithm 4.1*

- 451 • *either terminates in finite time and the last iterate \hat{x}^k is an optimal solution of (SIP),*
- 452 • *or generates an infinite sequence (\hat{x}^k) of feasible solutions in (SIP) s.t. $F(\hat{x}^k) \rightarrow \text{val}(\text{SIP})$.*

453 *Proof.* First of all, we emphasize that iterate \hat{x}^k , for any $k \in \mathbb{N}$, is feasible in (SIP) since
 454 $V_{\text{SDP}}^k(\hat{x}_k) - h(\hat{x}_k) \geq 0$ by definition of \hat{x}^k , and since $V_{\text{P}}(\hat{x}_k) \geq V_{\text{SDP}}^k(\hat{x}_k)$ according to Lemma 4.5:
 455 this proves that $V_{\text{P}}(\hat{x}_k) - h(\hat{x}^k) \geq 0$.

456 We start by considering the first case, where Algorithm 4.1 stops. If it stops before entering
 457 the loop, i.e., if $Q(\hat{x}^0) \succ 0$, we can apply Theorem 4.4 and conclude that \hat{x}^0 is an optimal solution
 458 of (SIP). If it stops at iteration k during the loop, this means that $x^k = \hat{x}^k$ since $d = 0$. Applying
 459 Lemma 4.6, we deduce that $F(\hat{x}^k) = F(x^k) \leq F(x^*) + \mu_k(x^k - \hat{x}^k)^\top(x^* - x^k) = F(x^*)$, where x^*
 460 is any optimal solution of (SIP). Therefore, the first part of the theorem is proved.

461 Now, we consider the second case: Algorithm 4.1 does not stop, and generates infinite sequences
 462 $(x^k)_{k \in \mathbb{N}^*}$ and $(\hat{x}^k)_{k \in \mathbb{N}^*}$. Using the notation $x^- := \max\{0, -x\}$ to denote the negative part of x , we
 463 claim that $(V_{\text{P}}(x^k) - h(x^k))^- \rightarrow 0$; for sake of brevity, we do not detail the proof of this vanishing
 464 here, but it uses exactly the same arguments used in the proof of Theorem A.1 (in Appendix A),
 465 relying on the compactness of sets \mathcal{X} and \mathcal{F} as well as on the continuity of the involved functions.
 466 We prove now that $(V_{\text{SDP}}^k(x^k) - h(x^k))^- \rightarrow 0$. The sequence $\nu_k := V_{\text{SDP}}^k(x^k) - h(x^k)$ is bounded
 467 since (i) $V_{\text{SDP}}(x^k) - h(x^k) \leq V_{\text{SDP}}^k(x^k) - h(x^k) \leq V_{\text{P}}(x^k) - h(x^k)$ due to Lemma 4.5 and (ii) $V_{\text{SDP}} - h$

468 as well as $V_{\mathcal{P}} - h$ are continuous (thus bounded) on the compact set \mathcal{X} . Therefore, $(\nu_k)^-$ is bounded
 469 too, and we must prove that the only possible limit for any converging subsequence of $(\nu_k)^-$ is 0.

470 We define any convergent subsequence extracted from $(\nu_k)^-$ as $(\nu_{\psi_0(k)}^-)$, where $\psi_0 : \mathbb{N}^* \rightarrow \mathbb{N}^*$
 471 is an increasing application, and ν^* the limit of $(\nu_{\psi_0(k)}^-)$. Since $(x^{\psi_0(k)})$ is bounded, it exists a
 472 converging subsequence $(x^{\psi(k)})$. We have then $\nu_{\psi(k)} = V_{\text{SDP}}^{\psi(k)}(x^{\psi(k)}) - h(x^{\psi(k)})$ for all $k \in \mathbb{N}^*$.
 473 We add and subtract $V_{\text{SDP}}^{\psi(k)}(x^{\psi(k-1)}) + h(x^{\psi(k-1)})$ to the right hand side of the equation, obtaining:
 474 $\nu_{\psi(k)} = \left(V_{\text{SDP}}^{\psi(k)}(x^{\psi(k-1)}) - h(x^{\psi(k-1)}) \right) + \left(V_{\text{SDP}}^{\psi(k)}(x^{\psi(k)}) - V_{\text{SDP}}^{\psi(k)}(x^{\psi(k-1)}) \right) + \left(h(x^{\psi(k-1)}) - h(x^{\psi(k)}) \right)$. As
 475 ψ is an increasing function, we have $\psi(k-1) \leq \psi(k) - 1$, and thus, applying Lemma 4.5, we deduce
 476 that $V_{\text{SDP}}^{\psi(k)}(x^{\psi(k-1)}) = V_{\mathcal{P}}(x^{\psi(k-1)})$. This is why, since $V_{\text{SDP}}^{\psi(k)}$ is ζ -Lipschitz (Lemma 4.5),

$$477 \quad \nu_{\psi(k)} = \left(V_{\mathcal{P}}(x^{\psi(k-1)}) - h(x^{\psi(k-1)}) \right) + \left(V_{\text{SDP}}^{\psi(k)}(x^{\psi(k)}) - V_{\text{SDP}}^{\psi(k)}(x^{\psi(k-1)}) \right) + \left(h(x^{\psi(k-1)}) - h(x^{\psi(k)}) \right)$$

$$478 \quad \geq \left(V_{\mathcal{P}}(x^{\psi(k-1)}) - h(x^{\psi(k-1)}) \right) - \zeta \|x^{\psi(k)} - x^{\psi(k-1)}\| + \left(h(x^{\psi(k-1)}) - h(x^{\psi(k)}) \right).$$

480 Being the negative part function decreasing and subadditive, we deduce that:

$$481 \quad (4.10) \quad \nu_{\psi(k)}^- \leq \left(V_{\mathcal{P}}(x^{\psi(k-1)}) - h(x^{\psi(k-1)}) \right)^- + \left(h(x^{\psi(k-1)}) - h(x^{\psi(k)}) - \zeta \|x^{\psi(k)} - x^{\psi(k-1)}\| \right)^-.$$

482 As $x^{\psi(k)}$ is converging and h is continuous, we have that $h(x^{\psi(k-1)}) - h(x^{\psi(k)}) - \zeta \|x^{\psi(k)} - x^{\psi(k-1)}\| \rightarrow 0$. Hence the negative part of this term, appearing in Eq. (4.10), vanishes as well.
 483 Since $\left(V_{\mathcal{P}}(x^{\psi(k-1)}) - h(x^{\psi(k-1)}) \right)^-$ is extracted from $\left(V_{\mathcal{P}}(x^k) - h(x^k) \right)^-$, which converges to zero, we
 484 deduce that the whole expression in the right hand side of Eq. (4.10) vanishes when $k \rightarrow \infty$. Thus,
 485 $\nu_{\psi(k)}^- \rightarrow 0$ and, by uniqueness of the limit, $\nu^* = 0$. As a conclusion, $\nu_k^- = \left(V_{\text{SDP}}^k(x^k) - h(x^k) \right)^- \rightarrow 0$.

487 Using Assumption 8, we introduce a Slater point $x^S \in \mathcal{X}$ such that $V_{\text{SDP}}(x^S) - h(x^S) = c > 0$.
 488 We also introduce $\lambda_k := \nu_k^- / (c + \nu_k^-)$. We notice that $\lambda_k \rightarrow 0$, since $\nu_k^- \rightarrow 0$. We define $\bar{x}_k =$
 489 $(1 - \lambda_k)x^k + \lambda_k x^S$, i.e., a convex combination of x^k and x^S . We emphasize that (\bar{x}_k, \bar{x}_k) is feasible
 490 in problem (4.9) at iteration k since

- 491 • \bar{x}_k satisfies the constraints on x , because both x^k and x^S satisfy the convex constraints
 - 492 $h(x) \leq \langle Q(x), Y^\ell \rangle$ for $\ell \in \{1, \dots, k-1\}$, and, by convex combination, so does \bar{x}_k ;
 - 493 • \bar{x}_k satisfies the constraint on the \hat{x} -component, since a solution $(\lambda, \alpha, \beta, \eta) \in \Omega^k(\bar{x}_k)$ exists.
- 494 Indeed, by concavity of $V_{\text{SDP}}^k - h$, we have that $V_{\text{SDP}}^k(\bar{x}_k) - h(\bar{x}_k) \geq (1 - \lambda_k)(V_{\text{SDP}}^k(x^k) - h(x^k)) + \lambda_k(V_{\text{SDP}}(x^S) - h(x^S)) = (1 - \lambda_k)\nu_k + \lambda_k c \geq -(1 - \lambda_k)\nu_k^- + \lambda_k c$. By construction
 495 of λ_k , $\lambda_k c - (1 - \lambda_k)\nu_k^- = 0$ and thus $V_{\text{SDP}}^k(\bar{x}_k) - h(\bar{x}_k) \geq 0$. This means that the value of
 496 problem (4.8) is greater than $h(\bar{x}_k)$, and thus that it exists $(\lambda, \alpha, \beta, \eta) \in \Omega^k(\bar{x}_k)$.

498 As the objective value of (\bar{x}_k, \bar{x}_k) in the problem (4.9) is $2F(\bar{x}_k)$, by optimality of (x^k, \hat{x}^k) :

$$499 \quad (4.11) \quad F(x^k) + F(\hat{x}^k) + \frac{\mu_k}{2} \|x^k - \hat{x}^k\|^2 \leq 2F((1 - \lambda_k)x^k + \lambda_k x^S) \leq 2(1 - \lambda_k)F(x^k) + 2\lambda_k F(x^S),$$

500 with the second inequality following from the convexity of F . We also notice that (\hat{x}^k, \hat{x}^k) is feasible
 501 in the problem (4.9) at iteration k , thus $F(x^k) + F(\hat{x}^k) + \frac{\mu_k}{2} \|x^k - \hat{x}^k\|^2 \leq 2F(\hat{x}^k)$, which means

$$502 \quad (4.12) \quad F(x^k) + \frac{\mu_k}{2} \|x^k - \hat{x}^k\|^2 \leq F(\hat{x}^k).$$

503 Summing Eq. (4.11) with Eq. (4.12), we obtain that $2F(x^k) + \mu_k \|x^k - \hat{x}^k\|^2 \leq 2(1 - \lambda_k)F(x^k) +$
 504 $2\lambda_k F(x^S)$, and thus $\mu_k \|x^k - \hat{x}^k\|^2 \leq 2\lambda_k (F(x^S) - F(x^k))$. Using that $0 < \underline{\mu} \leq \mu_k$,

$$505 \quad (4.13) \quad \|x^k - \hat{x}^k\| \leq \sqrt{\underline{\mu}^{-1} (2\lambda_k (F(x^S) - F(x^k)))}$$

506 holds. Since $\lambda_k \rightarrow 0$, and $F(x^S) - F(x^k)$ is bounded, we deduce from Eq. (4.13) that

$$507 \quad (4.14) \quad \|x^k - \hat{x}^k\| \rightarrow 0.$$

508 Taken an optimal solution of (SIP), x^* , since \hat{x}^k is feasible in (SIP) as stated above, and since F
509 is J -Lipschitz according to Assumption 1, we deduce that $F(x^*) \leq F(\hat{x}^k) \leq F(x^k) + J\|x^k - \hat{x}^k\|$.
510 According to Lemma 4.6, we know that $F(x^k) \leq F(x^*) + \mu_k(x^k - \hat{x}^k)^\top(x^* - x^k)$, which implies,
511 according to the Cauchy-Schwartz inequality, that

$$512 \quad (4.15) \quad F(x^*) \leq F(\hat{x}^k) \leq F(x^*) + \mu_k \|x^k - \hat{x}^k\| \|x^* - x^k\| + J \|x^k - \hat{x}^k\|.$$

513 Since $\|x^* - x^k\|$ is bounded, we deduce from Eq. (4.14) that $F(x^*) + \mu_k \|x^k - \hat{x}^k\| \|x^* - x^k\| + J \|x^k -$
514 $\hat{x}^k\| \rightarrow F(x^*)$ and thus, $F(\hat{x}^k) \rightarrow F(x^*) = \text{val}(\text{SIP})$. \square

515 Based on the previous Theorem, we deduce that the Algorithm 4.1 stops in finite time if the
516 tolerance parameters are positive. We prove this result in the following corollary, in which we use
517 the notation $\text{diam}(\mathcal{X}) := \max_{x_1, x_2 \in \mathcal{X}} \|x_1 - x_2\|$.

518 **COROLLARY 4.8.** *Under Assumptions 1–5, and Assumption 8, if $\epsilon > 0$ and $d > 0$, Algorithm 4.1*
519 *terminates in finite time and returns a solution \hat{x}^k feasible in (SIP); moreover $F(\hat{x}^k) \leq \text{val}(\text{SIP}) +$*
520 *$d(\mu_k \text{diam}(\mathcal{X}) + J)$, where J is the Lipschitz constant for F .*

521 *Proof.* Proof in Appendix C.5. \square

522 **5. Applications.** In this section, we present two problems that can be modeled as (SIP).
523 For each of these, we present both the SIP formulation, and the corresponding single-level finite
524 formulation (SIPR).

525 **5.1. Constrained quadratic regression.** We consider a quadratic statistical model with
526 Gaussian noise linking a vector $w \in \mathbb{R}^n$ of explanatory variables, i.e., the features vector, and an
527 output $z \in \mathbb{R}$ as follows: $z = \frac{1}{2}w^\top \bar{Q}w + \bar{q}^\top w + \bar{c} + \epsilon$, where $\bar{Q} \in \mathbb{R}^{n \times n}$ s.t. $\bar{Q} = \bar{Q}^\top$, $\bar{q} \in \mathbb{R}^n$, $\bar{c} \in \mathbb{R}$
528 and $\epsilon \sim \mathcal{N}(0, \sigma^2)$. Let us suppose that the parameters of this model are unknown, except an *a priori*
529 bound $B \in \mathbb{R}_+$ on their magnitude. Moreover, we are given a dataset $(w_i, z_i)_{1 \leq i \leq P} \in (\mathbb{R}^n \times \mathbb{R})^P$.
530 Note that w_i is an n -dimensional vector, for any $i = 1, \dots, P$. The problem of finding the maximum
531 likelihood estimator for $\bar{Q} \in \mathbb{R}^{n \times n}$, $\bar{q} \in \mathbb{R}^n$, $\bar{c} \in \mathbb{R}$ just consists in computing the triplet $(Q, q, c) \in$
532 $\mathbb{R}^{n \times n} \times \mathbb{R}^n \times \mathbb{R}$ that minimizes the least-squares error $\sum_{i=1}^P (z_i - \frac{1}{2}w_i^\top Qw_i - q^\top w_i - c)^2$. We consider that
533 (i) the features vector belongs to a given polytope $\mathcal{F} \subset \mathbb{R}^n$, (ii) the noiseless value $\frac{1}{2}y^\top \bar{Q}y + \bar{q}^\top y + \bar{c}$
534 is nonnegative for any $y \in \mathcal{F}$. Hence, this inverse problem is a “constrained quadratic regression
535 problem” that may be written as:

$$536 \quad (5.1) \quad \begin{cases} \min_{Q, q, c} & \sum_{i=1}^P (z_i - \frac{1}{2}w_i^\top Qw_i - q^\top w_i - c)^2 \\ \text{s.t.} & Q = Q^\top \\ & 0 \leq \frac{1}{2}y^\top Qy + q^\top y + c \quad \forall y \in \mathcal{F} \\ & \|Q\|_\infty \leq B, \|q\|_\infty \leq B, |c| \leq B \\ & Q \in \mathbb{R}^{n \times n}, q \in \mathbb{R}^n, c \in \mathbb{R}. \end{cases}$$

537 Formulation (5.1) is a SIP problem. In particular, this model fits in the general setting of formulation
538 (SIP), where the matrix Q is itself the upper-level variable of dimensions $n \times n$. As in Section 4, we
539 assume that $\mathcal{F} = \{y \in \mathbb{R}^n : a_j^\top y \leq b_j, \forall j = 1, \dots, r\}$ is included in the centered ℓ_2 -ball with radius

540 $\rho > 0$, and we use the notation $\mathcal{A}_j = \begin{pmatrix} 0_n & \frac{a_j}{2} \\ \frac{a_j^\top}{2} & 0 \end{pmatrix}$ for all $j \in \{1, \dots, r\}$. Then

$$541 \quad (5.2) \quad \left\{ \begin{array}{l} \min_{Q, q, c, \lambda, \alpha, \beta} \sum_{i=1}^P (z_i - \frac{1}{2} w_i^\top Q w_i - q^\top w_i - c)^2 \\ \text{s.t.} \quad Q = Q^\top \\ -c \leq -b^\top \lambda - \alpha(1 + \rho^2) - \beta \\ \frac{1}{2} \begin{pmatrix} Q + 2\alpha I_n & q \\ q^\top & 2(\beta + \alpha) \end{pmatrix} + \sum_{j=1}^r \lambda_j \mathcal{A}_j \succeq 0 \\ \|Q\|_\infty \leq B, \|q\|_\infty \leq B, |c| \leq B \\ Q \in \mathbb{R}^{n \times n}, q \in \mathbb{R}^n, c \in \mathbb{R} \\ \lambda \in \mathbb{R}_+^r, \alpha \in \mathbb{R}_+, \beta \in \mathbb{R}. \end{array} \right.$$

542 is the (SIPR) formulation corresponding to (5.1). Formulation (5.2) is feasible, because the all-
543 zero solution satisfies every constraint. In general, (5.2) is a restriction of (5.1) since Q may not
544 necessarily be PSD. The set $\Omega^k(\hat{Q}, \hat{q}, \hat{c})$ that will be used in the IOA algorithm is:

$$545 \quad \Omega^k(\hat{Q}, \hat{q}, \hat{c}) := \left\{ (\lambda, \alpha, \beta, \eta) \in \mathbb{R}_+^r \times \mathbb{R}_+ \times \mathbb{R} \times \mathbb{R}_+^{k-1} : -\hat{c} \leq -b^\top \lambda - \alpha(1 + \rho^2) - \beta + \sum_{\ell=1}^{k-1} \eta_\ell v_\ell \wedge \right. \\ 546 \quad \left. \frac{1}{2} \begin{pmatrix} \hat{Q} + 2\alpha I_n & \hat{q} \\ \hat{q}^\top & 2(\beta + \alpha) \end{pmatrix} + \sum_{j=1}^r \lambda_j \mathcal{A}_j - \sum_{\ell=1}^{k-1} \frac{1}{2} \eta_\ell \begin{pmatrix} Q_\ell & q_\ell \\ q_\ell^\top & 0 \end{pmatrix} \succeq 0 \right\},$$

547 where (Q_ℓ, q_ℓ) is the solution x_ℓ obtained by solving problem (4.9) at iteration ℓ , and v_ℓ is the
548 value of the lower-level problem: $\min_{y \in \mathcal{F}} \frac{1}{2} y^\top Q_\ell y + q_\ell^\top y$.

549 In order to benchmark our approaches, we can solve the following relaxation of (5.1) obtained
550 by replacing the lower-level problem by its KKT conditions:

$$551 \quad (5.3) \quad \left\{ \begin{array}{l} \min_{Q, q, c, y, \gamma} \sum_{i=1}^P (z_i - \frac{1}{2} w_i^\top Q w_i - q^\top w_i - c)^2 \\ \text{s.t.} \quad Q = Q^\top \\ -c \leq \frac{1}{2} y^\top Q y + q^\top y \\ Ay \leq b \\ Qy + q + A^\top \gamma = 0 \\ \gamma^\top (Ay - b) = 0 \\ \|Q\|_\infty \leq B, \|q\|_\infty \leq B, |c| \leq B \\ Q \in \mathbb{R}^{n \times n}, q \in \mathbb{R}^n, c \in \mathbb{R}, y \in \mathbb{R}^n, \gamma \in \mathbb{R}_+^r, \end{array} \right.$$

552 where γ is the KKT multiplier vector associated to the lower-level constraints $Ay \leq b$. Problem (5.3)
553 is a non-convex polynomial problem involving multivariate polynomials of degree up to three. We
554 also compare our results with those obtained by the global optimization algorithm proposed in [29].

555 **5.2. Zero-sum game with cubic payoff.** In this section, we are interested in solving a two-
556 player zero-sum game that is related to an undirected graph $\mathcal{G} = (V, E)$. We assume that player
557 1 benefits from a strategical advantage on player 2, which will be explained more precisely later.
558 We let n denote the cardinality of V . Each player positions a resource on each node $i \in V$. After
559 normalization, we can consider that the action set of both players is $\Delta_n = \{x \in \mathbb{R}_+^n : \sum_{i=1}^n x_i = 1\}$.

560 A two-player zero-sum game is a two-player game s.t., for every strategy $x \in \Delta_n$ of player 1, and for
561 every strategy $y \in \Delta_n$ of player 2, the payoffs of the two players sum to zero. If we define $P_i(x, y)$
562 as the payoff of player i related to the strategy pair (x, y) , we thus have that $P_1(x, y) = -P_2(x, y)$.
563 Since the payoffs sum to zero, we can write the zero-sum game by specifying only one *game payoff*.

567 Player 1 wishes to minimize it, and player 2 wishes to maximize it. The game payoff $P(x, y)$ related
 568 to the pair of strategies $(x, y) \in \Delta_n \times \Delta_n$ is the sum of:

- 569 • the opposite of a term describing the “proximity” between x and y in the graph, $x^\top My$,
 570 where $M \in \mathbb{R}^{n \times n}$ is a matrix having $M_{ij} = 1$ if $i = j$ or $\{i, j\} \in E$, and $M_{ij} = 0$ otherwise,
- 571 • the quadratic costs that player 1 has to pay to deploy his resources on the graph: $c_1(x) =$
 572 $\frac{1}{2}x^\top Q_1 x + q_1^\top x$,
- 573 • the opposite of the quadratic costs that player 2 has to pay to deploy her resources on the
 574 graph, and that is influenced by player 1 strategy: $c_2(x, y) = \frac{1}{2}y^\top Q_2(x)y + q_2^\top y$. In this
 575 sense, player 1 has a strategic advantage over player 2.

576 Hence, this zero-sum game can then be written as $\min_{x \in \Delta_n} \max_{y \in \Delta_n} -x^\top My + c_1(x) - c_2(x, y)$. Loosely
 577 speaking, player 1 trades off his costs for placing his resource where player 2’s one is (i.e., maximizing
 578 the proximity) and for augmenting player 2’s costs. In the meantime, player 2 tries to *avoid* player
 579 1, while minimizing her own costs. From player 1’s perspective, this problem can be cast as the
 580 following SIP formulation:

$$581 \quad (5.4) \quad \begin{cases} \min_{x, z} & \frac{1}{2}x^\top Q_1 x + q_1^\top x + z \\ \text{s.t.} & -z \leq \frac{1}{2}y^\top Q_2(x)y + (q_2 + M^\top x)^\top y \quad \forall y \in \Delta_n \\ & x \in \Delta_n, z \in \mathbb{R}. \end{cases}$$

583 This latter formulation clearly fits in the general setting of formulation (SIP). Hence, we apply the
 584 methodology of Section 4 with $r = n + 2$, $\rho = 1$, $a_1 = \mathbf{1}$ ($\mathbf{1}$ is the all-ones n -dimensional vector),
 585 $b_1 = 1$, $a_2 = -\mathbf{1}$, $b_2 = -1$, and $\forall j \in \{1, \dots, n\} \quad a_{j+2} = -e_j$ (e_j is the j -th vector of the standard
 586 basis in \mathbb{R}^n) and $b_j = 0$. The dual variable is $\lambda \in \mathbb{R}_+^{n+2}$. In this application, the single-level finite
 587 formulation (SIPR) reads

$$588 \quad (5.5) \quad \begin{cases} \min_{x, z, \lambda, \alpha, \beta} & z + \frac{1}{2}x^\top Q_1 x + q_1^\top x \\ \text{s.t.} & -z \leq -\lambda_1 + \lambda_2 - 2\alpha - \beta \\ & \frac{1}{2} \begin{pmatrix} Q_2(x) + 2\alpha I_n & W(x, \lambda) \\ W(x, \lambda)^\top & 2\beta + 2\alpha \end{pmatrix} \succeq 0 \\ & x \in \Delta_n, z \in \mathbb{R} \\ & \lambda \in \mathbb{R}_+^{n+2}, \alpha \in \mathbb{R}_+, \beta \in \mathbb{R}, \end{cases}$$

590 where $W(x, \lambda) = q_2 + M^\top x - \sum_{j=1}^n \lambda_{j+2} e_j + (\lambda_1 - \lambda_2)\mathbf{1}$. If $Q_2(x) \succeq 0$ is PSD for any $x \in \Delta_n$,
 591 formulation (5.5) is a reformulation of (5.4). Otherwise, it is just a restriction of (5.4). In any case,
 592 such formulation is feasible, because for given vectors $x \in \Delta_n$, $\lambda \in \mathbb{R}_+^{n+2}$, and scalar $\beta \in \mathbb{R}$, taking
 593 arbitrary large scalars α and z , the two constraints are satisfied.

594 The set $\Omega^k(\hat{x}, \hat{z})$ that will be used in the IOA algorithm, as a constraint in problem (4.9), is:

$$595 \quad \Omega^k(\hat{x}, \hat{z}) := \left\{ (\lambda, \alpha, \beta, \eta) \in \mathbb{R}_+^{n+2} \times \mathbb{R}_+ \times \mathbb{R} \times \mathbb{R}_+^{k-1} : -\hat{z} \leq -\lambda_1 + \lambda_2 - 2\alpha - \beta + \sum_{\ell=1}^{k-1} \eta_\ell v_\ell \wedge \right. \\ \left. \frac{1}{2} \begin{pmatrix} Q_2(\hat{x}) + 2\alpha I_n & W(\hat{x}, \lambda) \\ W(\hat{x}, \lambda)^\top & 2\beta + 2\alpha \end{pmatrix} - \sum_{\ell=1}^{k-1} \frac{1}{2} \eta_\ell \begin{pmatrix} Q_2(x_\ell) & q_2 + M^\top x_\ell \\ (q_2 + M^\top x_\ell)^\top & 0 \end{pmatrix} \succeq 0 \right\},$$

596 where x_ℓ is the solution obtained by solving problem (4.9) at iteration ℓ , and v_ℓ is the value
 599 of the lower-level problem: $\min_{y \in \Delta_n} \frac{1}{2}y^\top Q_2(x_\ell)y + (q_2 + M^\top x_\ell)^\top y$. As for the first application, we
 600 benchmark our two approaches both with the KKT-based relaxation/reformulation (depending on

601 the convexity of the lower-level problem), and with the algorithm proposed by Mitsos in [29]. Given
 602 the KKT multipliers γ_1 and γ_2 associated respectively to the lower-level constraint $\sum_{i=1}^n y_i = 1$, and
 603 the nonnegativity constraint $y \geq 0$, the single-level finite formulation obtained by replacing the
 604 lower level of (5.4) by its KKT conditions, is

$$\begin{cases}
 \min_{x,z,y,\gamma_1,\gamma_2} & z + \frac{1}{2}x^\top Q_1 x + q_1^\top x \\
 \text{s.t.} & -z \leq \frac{1}{2}y^\top Q_2(x)y + (q_2 + M^\top x)^\top y \\
 & Q_2(x)y + q_2 + M^\top x + \gamma_1 \mathbf{1} - I_n \gamma_2 = 0 \\
 & -\gamma_2^\top (I_n y) = 0 \\
 & x \in \Delta_n, y \in \Delta_n, z \in \mathbb{R}, \gamma_1 \in \mathbb{R}, \gamma_2 \in \mathbb{R}_+^n.
 \end{cases}
 \tag{5.6}$$

607 The KKT multiplier γ_1 is associated to an equality constraint, hence it can be either nonnegative
 608 or negative, and we have no complementarity constraint involving it in formulation (5.6). This
 609 relaxation/reformulation of problem (5.4), as well as (5.6), is a non-convex polynomial optimization
 610 problem involving multivariate polynomials of degree up to three.

611 **6. Numerical results.** In this section we present the numerical results obtained by testing
 612 several instances of the two applications presented in Section 5, available online at the public
 613 repository <https://github.com/aoustry/SIP-with-QP-LL>.

614 For the constrained quadratic regression (Section 5.1), we solve twenty randomly generated
 615 instances. Each of these instances is generated by choosing the statistical parameters $\bar{Q}, \bar{q}, \bar{c}$ at
 616 random, drawing $P = 4000$ random features vectors $w_i \in \mathbb{R}^n$, and then computing the associated
 617 outputs $z_i \in \mathbb{R}$ with a centered Gaussian noise. The data $(w_i, z_i)_{1 \leq i \leq P}$ are produced with \bar{Q} PSD
 618 for ten instances, named *PSD_inst#* in Table 2, and are produced with an indefinite \bar{Q} for ten
 619 instances, named *notPSD_inst#* in Table 2.

620 For the zero-sum game with cubic payoff application (Section 5.2), we test twenty-two instances
 621 where the matrix M is taken from the DIMACS graph coloring challenge¹. We randomly generate
 622 Q_1 in a way such that it is PSD, as well as the coefficients of the linear function $Q_2(x)$ such that
 623 $Q_2(x)$ is PSD for all feasible x in the instances named *#_PSD* in Table 3. Regarding the instances
 624 named *#_notPSD* in Table 3, no particular precaution is taken to enforce that $Q_2(x)$ is PSD. Hence,
 625 the sign of the eigenvalues of $Q_2(x)$ depends on x . The code that generates all the instances is
 626 available online, in the aforementioned repository.

627 The global solutions of SIP formulations are found using the CP algorithm (Algorithm 3.1
 628 presented in Section 3), the IOA algorithm (Algorithm 4.1 presented in Section 4.5), and the global
 629 solution algorithm proposed in [29], that we call “Mitsos Algorithm” in this section. We also
 630 benchmark these algorithms with the traditional relaxation/reformulation approach based on the
 631 KKT conditions of the lower-level problem.

632 The CP algorithm is implemented using the Python programming language [45]. Both the
 633 master problem (R_k) and the lower-level problem (P_{x^k}) are solved using the global QP solver
 634 Gurobi [16]. The tolerance for the feasibility error $\epsilon = (h(x^k) - \text{val}(P_{x^k}))^+$ is set to 10^{-6} .

635 The IOA algorithm is also implemented in Python. We use the conic optimization solver Mosek
 636 [2] to solve (SIPR) at step 0 as well as the master problem (4.9) at step 7. The global solver Gurobi
 637 is used to solve the problem (P_{x^k}) at step 8. The tolerances d and ϵ , used in the stopping criteria,
 638 are set to 10^{-6} . An *a priori* knowledge on the convex nature of the lower-level problem gives the

¹ <https://mat.tepper.cmu.edu/COLOR/instances.html>

639 guarantee that formulation (SIPR) has the same value of formulation (SIP). In such case, one can
 640 just solve (SIPR) to obtain an optimal solution of (SIP). Yet, this prior knowledge is not common
 641 to all the possible applications of our approaches, hence we decide to treat all the instances in the
 642 same way, by running the sequence of instructions described in Algorithm 4.1.

643 Mitsos algorithm [29], used to benchmark the proposed approaches, is implemented using
 644 Python. As already said in Section 2, the algorithm generates, at each iteration, a lower and
 645 an upper bound of the optimal value of (SIP). The relaxation solved to get a lower bound is ob-
 646 tained by approximating the infinite constraint parameter set by a progressively finer finite subset.
 647 The formulation solved to get an upper bound is obtained by restricting the infinite constraints
 648 right hand side by $\epsilon_g > 0$ and considering a successively finer discretization of the parameter set.
 649 For arbitrary combinations of the discretized parameter set and ϵ_g , this formulation is neither a re-
 650 striction, nor a relaxation of the SIP problem. However, the existence of a SIP-Slater point ensures
 651 that the algorithm finitely generates feasible iterates, the objective value of which converges to the
 652 optimal value. In our implementation, at each iteration, both the relaxation, and the restriction of
 653 the SIP problem, as well as the lower level for the corresponding iterate, are solved using Gurobi.

654 We implement the KKT relaxations/reformulations with the AMPL modeling language [14],
 655 and solve them using Gurobi, for sake of fairness in the comparison. The KKT formulations are
 656 particularly hard to solve, mainly because of the complementarity constraints. Indeed, for most of
 657 the tested instances, Gurobi does not terminate within the time limit. For these instances, we just
 658 display, in italic font, the lower bound given by the optimal value of the best relaxation of the KKT
 659 formulation found by Gurobi within the time limit.

660 For all the approaches, Gurobi is run with its default settings. The tests were performed on a
 661 computer with a 2.70GHz Intel(R) Core(TM) i7 quad-core and with 16 GB of RAM. For all the
 662 approaches we set a time limit (t.l.) of 18.000 seconds (5 hours).

663 The results for Application 1 and Application 2 are reported in Table 2 and Table 3 respectively.
 664 The headings are the following:

- 665 • “ n ” is the number of the lower-level variables; “time(s)” is the computing time in seconds;
 666 “it” is the number of iterations (for the IOA algorithm, such number is 0 when the sufficient
 667 condition at step 1 is verified and the algorithm does not enter the loop);
- 668 • for CP and Mitsos algorithms “obj/LB-UB” is, respectively, either the optimal value of SIP
 669 formulation, or a pair of values corresponding to: the best lower bound (LB) and the best
 670 feasible solution, i.e., upper bound (UB), found by the algorithm within the time limit;
- 671 • for the IOA algorithm “obj/UB” is, respectively, either the optimal value of the SIP for-
 672 mulation (the sufficient condition at step 1 is verified), or the best value $F(\hat{x}^k)$ found by
 673 the algorithm within the time limit;
- 674 • for CP and IOA algorithms “% (P_{x^k})” is the percentage of the total computing time, i.e.,
 675 time(s), used to solve (P_{x^k});
- 676 • for the KKT approach, “obj/LB” is, respectively, either the optimal value of the KKT
 677 formulation, or the best lower bound of such value found by the solver Gurobi within the
 678 time limit, which is a lower bound for the SIP optimal value too.

679 In Table 2 and Table 3, the minimum required times are reported in bold for each instance.

680 As expected, the optimal values found by the three considered global methods are the same for
 681 all the instances (when the algorithm stops before the time limit is hit). In terms of computational
 682 time, the IOA algorithm is more efficient than the other approaches for all the instances where the
 683 restriction is proven to be optimal during the preliminary step of this algorithm. When solving the
 684 other instances, CP shows the best performance, although the number of iterations needed by the

Instances		CP algorithm				IOA algorithm				KKT	Mitsos algorithm [29]		
Name	n	obj/LB-UB	time(s)	it	% ($P_{\pm k}$)	obj/UB	time(s)	it	% ($P_{\pm k}$)	obj/LB	obj/LB-UB	time(s)	it
PSD_inst1	5	358.64	0.70	6	3.4	358.64	0.27	0	-	355.78	358.64	1.26	6
PSD_inst2	5	365.60	0.32	3	3.6	365.60	0.23	0	-	363.85	365.60	0.56	3
PSD_inst3	5	363.43	0.91	8	3.4	363.43	0.22	0	-	359.16	363.43	1.78	8
PSD_inst4	5	353.90	0.54	5	3.6	353.90	0.22	0	-	353.19	353.90	0.97	5
PSD_inst5	10	391.21	4.81	17	1.1	391.21	0.60	0	-	359.48	391.21	10.14	17
PSD_inst6	10	397.59	4.92	17	1.0	397.59	0.63	0	-	353.55	397.59	10.45	17
PSD_inst7	13	440.84	8.70	19	0.7	440.84	1.01	0	-	358.19	440.84	25.1	19
PSD_inst8	13	382.17	2581	17	99.7	382.17	2734	15	98.6	345.52	382.17	6193	17
PSD_inst9	15	564.84 - 622.88	t.l.	5	100	572.77	1.62	0	-	351.95	564.84 - inf	t.l.	5
PSD_inst10	15	526.22 - 545.34	t.l.	8	100	528.93	1.44	0	-	346.43	526.22 - inf	t.l.	8
notPSD_inst1	5	358.47	0.22	2	4.4	358.47	2.03	4	0.8	345.12	358.47	0.28	2
notPSD_inst2	5	378.28	0.22	2	4.95	378.28	2.04	4	0.5	370.89	378.28	0.28	2
notPSD_inst3	5	345.81	0.12	1	3.5	345.81	0.66	1	0.3	345.81	345.81	0.18	1
notPSD_inst4	5	353.25	0.11	1	4.3	353.25	1.10	2	0.3	353.25	353.25	0.14	1
notPSD_inst5	10	503.88	5.17	18	8.4	503.88	32.2	18	1.3	360.42	503.88	11.3	18
notPSD_inst6	10	482.96	31.6	36	68.0	482.96	84.2	35	32.4	357.48	482.96	65.4	36
notPSD_inst7	13	647.08	119	61	77.2	647.08	211	54	37.8	351.31	647.08	263	61
notPSD_inst8	13	588.19	566	77	92.8	588.19	700	74	73.6	358.28	588.19	977	77
notPSD_inst9	15	1126.44	687	97	89.9	1126.44	922	92	63.4	345.44	1126.44	1356	97
notPSD_inst10	15	580.60	595	64	92.0	580.60	711	60	70.9	350.60	580.60	1047	64

Table 2: Numerical results of the first application

Instances		CP algorithm				IOA algorithm				KKT	Mitsos algorithm [29]		
Name	n	obj/LB-UB	time(s)	it	% ($P_{\pm k}$)	obj/UB	time(s)	it	% ($P_{\pm k}$)	obj/LB	obj/LB-UB	time(s)	it
jean_PSD	80	-0.0760	49.5	183	23.5	-0.0760	19.9	0	-	-1.0121	-0.0760	128	171
myciel4_PSD	23	-0.3643	4.81	390	31.0	-0.3643	0.09	0	-	-1.0154	-0.3643	139	371
myciel5_PSD	47	-0.3164	21.6	684	13.1	-0.3164	1.51	0	-	-1.0171	-0.3164	580	633
myciel6_PSD	95	-0.2841	399	2203	2.8	-0.2841	42.0	0	-	-1.0207	-0.2841	6738	2008
myciel7_PSD	191	-0.2608	7498	5586	0.5	-0.2608	3452	0	-	-1.9246	-0.2608 - -0.2608	t.l.	3268
queen5.5_PSD	25	-0.5536	1.73	165	39.4	-0.5536	0.12	0	-	-1.0163	-0.5536	19.7	151
queen6.6_PSD	36	-0.4619	8.98	511	22.3	-0.4619	0.37	0	-	-1.0185	-0.4619	168	458
queen7.7_PSD	49	-0.4054	31.0	937	12.1	-0.4054	1.60	0	-	-1.0204	-0.4054	602	863
queen8.8_PSD	64	-0.3614	97.0	1578	7.0	-0.3614	4.43	0	-	-1.0215	-0.3614	1662	1416
queen8.12_PSD	96	-0.3000	1194	4138	1.9	-0.3000	36.4	0	-	-1.0217	-0.3000	14153	3570
queen9.9_PSD	81	-0.3247	351	2637	3.4	-0.3247	14.9	0	-	-1.0216	-0.3247	5027	2357
jean_notPSD	80	2.3979	6.82	7	99.5	2.3979	195	8	4.0	1.4095	2.3979	16.4	7
myciel4_notPSD	23	0.5198	43.5	40	99.8	0.5198	52.8	41	82.9	-0.2441	0.5198	102.6	40
myciel5_notPSD	47	1.2779	42.4	37	99.7	1.2779	86.3	33	35.3	0.3167	1.2779	103	37
myciel6_notPSD	95	2.9378	236	35	99.9	2.9378	2223	38	11.8	1.7319	2.9378	615	35
myciel7_notPSD	191	6.2486	773	23	99.9	6.2932	t.l.	4	0.06	-9.2171	6.2486	1320	23
queen5.5_notPSD	25	0.3800	21.2	51	99.4	0.3800	29.5	44	57.2	-0.3318	0.3800	42.4	51
queen6.6_notPSD	36	0.8511	293	73	99.9	0.8511	350	68	81.5	-0.0377	0.8511	751	73
queen7.7_notPSD	49	1.3510	69.8	44	99.7	1.3510	161	40	40.7	0.3615	1.3510	174	44
queen8.8_notPSD	64	1.8122	543	33	100	1.8122	1001	42	70.1	0.7866	1.8122	1113	33
queen8.12_notPSD	96	2.8102	1049	34	100	2.8102	2525	32	32.4	1.6273	2.8102	1935	34
queen9.9_notPSD	81	2.2979	2424	46	100	2.2979	2613	39	69.9	1.2042	2.2979	4545	46

Table 3: Numerical results of the second application

685 three methods is always comparable. Indeed, IOA iterations are, in average, more time consuming
 686 than the other algorithms and hence, for these instances, the computational time for IOA is larger
 687 even if the number of iterations of IOA is often less w.r.t. CP and Mitsos algorithm. As regards
 688 Mitsos algorithm, it turns out to be slower than CP. When compared to the IOA algorithm, it is
 689 sometimes better in terms of computational time, as shown in Table 4. We recall that, as the IOA
 690 algorithm, Mitsos algorithm computes a sequence of feasible solutions, the value of which converges
 691 to the optimal value, whereas the iterates in the CP algorithm are only asymptotically feasible.

692 The instance “PSD_inst8” is interesting, since the restriction (5.2) is obviously optimal (its

Instances		IOA algorithm	Mitsos algorithm [29]
First application	PSD_inst#	100%	0%
	notPSD_inst#	40%	60%
Second application	#_PSD	100%	0
	#_notPSD	60%	40%

Table 4: Percentage of instances for which each approach requires less computational time than the other

693 value is also 382.17), but the IAO algorithm is not able to detect it immediately. Indeed, matrix
 694 Q^0 found solving the formulation (5.2) for “PSD_inst8” is not positive definite, but only positive
 695 semidefinite, thus the algorithm needs to enter the loop and runs 15 iterations before stopping.

696 As concerns the KKT formulation, we see that the quality of the lower bounds computed by
 697 Gurobi within the time limit is very bad. Indeed, this formulation is particularly hard to solve
 698 mainly because of the complementarity constraints.

699 To understand the causes of the computational time required by the IOA and the CP algorithms,
 700 we can look at “% (P_{x^k})” columns of Table 2 and 3. As regards the CP algorithm, for the first
 701 application, the time required to perform step 3 of the CP algorithm (i.e., to solve (P_{x^k})) is longer
 702 than the time required to perform step 2 (i.e., to solve (R_k)) only for the bigger instances ($n \geq 13$
 703 for instances with a convex lower level and $n \geq 10$ for instances with a non-convex lower level).
 704 In fact, when n grows, more time is needed to solve a possibly non-convex QP problem having
 705 Q and q as coefficients, rather than a convex QP having Q and q as variables. When n is small,
 706 it is different: even if the inner problem is quadratic non-convex, it has a small size so it is not
 707 harder to solve than the master problem. For the second application, the time required to solve
 708 the lower-level problem is longer than the time required to solve the master problem only for the
 709 instances having a non-convex lower level, i.e., the second half of the Table 3 rows. Indeed, when
 710 $Q_2(x^k)$ is not PSD, problem (P_{x^k}) is possibly non-convex and it becomes harder to solve than the
 711 master problem. As regards the IOA algorithm, we see that the percentage of time required to
 712 solve problem (P_{x^k}) depends on the instance. Actually, the difficulty of the lower-level problem
 713 may also vary, for a same instance, between iterates, depending, e.g., on the number of the negative
 714 eigenvalues of $Q(x^k)$. In general, the value in the column “% (P_{x^k})” for the IOA algorithm is always
 715 less than the corresponding value in the column of the CP algorithm. This means, as expected,
 716 that the master program is more costly for IOA than for CP.

717 **7. Conclusion.** We focus on a convex semi-infinite program having an infinite number of
 718 quadratically parameterized constraints. We consider two independent approaches to deal with such
 719 SIP problems. First, we focus on a classical cutting plane algorithm for solving the SIP formulation.
 720 We propose for it a new convergence rate, in the case where the objective function is strongly-
 721 convex and under a Slater assumption. Our new convergence rate presents the nice property to be
 722 directly related to the iteration index k , which is something new w.r.t. what is usually proved in
 723 SIP literature, where the linear rate of convergence is not controlled by k (see [31, Theorem 4.3]).
 724 Second, a new convex finite formulation (SIPR) obtained via the *lower-level dualization approach*
 725 provides a feasible solution \hat{x} , which is optimal either if the quadratic lower-level problem is convex,
 726 or if a sufficient condition we introduce on \hat{x} (that can be computed a posteriori) is verified. Based
 727 on the *lower-level dualization approach*, we present a new convergent algorithm, named Inner-Outer
 728 Approximation algorithm, which solves at each iteration a relaxation of the restriction (SIPR). Our
 729 computational experiments on small and medium-scale instances show the superiority, in terms of
 730 solution time, of the Inner-Outer Approximation algorithm for the instances where it is able to
 731 certify the optimality of the restriction. As concerns the other cases, the cutting plane approach is

732 faster, but the IOA often requires less iterations, and provides a feasible solution at each iteration.
 733 Both methods find an optimal solution of the SIP problem with good accuracy. We also compare the
 734 performances of the proposed approaches with the “Mitsos algorithm” [29], which provides feasible
 735 solutions at each iteration, and the KKT relaxation approach, which only provides loose lower
 736 bounds. A possible extension of our work could be implementing the Inner-Outer Approximation
 737 algorithm with the lower-level problem solved with an “on-demand” accuracy at each iteration.
 738 A rule for the update of the proximal parameter μ in the Inner-Outer Approximation algorithm
 739 should be studied to improve the performances of the algorithm itself. These possibilities will be
 740 addressed in future works.

741

REFERENCES

742 [1] S. ABBOTT, Understanding Analysis, Undergraduate Texts in Mathematics, Springer New York, 2016, <https://doi.org/10.1007/978-1-4939-2712-8>.
 743
 744 [2] M. APS, The MOSEK python optimizer API manual. Version 9.2.36, 2021, <https://docs.mosek.com/9.2/pythonapi/index.html>.
 745
 746 [3] B. BANK, J. GUDDAT, D. KLATTE, B. KUMMER, AND K. TAMMER, Non-linear parametric optimization, Springer,
 747 1982, <https://doi.org/10.1007/978-3-0348-6328-5>.
 748 [4] A. BEN-TAL, L. EL GHAOU, AND A. NEMIROVSKI, Robust optimization, Princeton university press, 2009.
 749 [5] A. BEN-TAL AND A. NEMIROVSKI, Lectures on Modern Convex Optimization, Society for Industrial and Applied
 750 Mathematics, 2001, <https://doi.org/10.1137/1.9780898718829>.
 751 [6] B. BETRÓ, An accelerated central cutting plane algorithm for linear semi-infinite programming, Mathematical
 752 Programming, 101 (2004), pp. 479–495, <https://doi.org/10.1007/s10107-003-0492-5>.
 753 [7] B. BHATTACHARJEE, P. LEMONIDIS, W. H. GREEN JR, AND P. I. BARTON, Global solution of
 754 semi-infinite programs, Mathematical Programming, 103 (2005), pp. 283–307, <https://doi.org/10.1007/s10107-005-0583-6>.
 755
 756 [8] J. W. BLANKENSHIP AND J. E. FALK, Infinitely constrained optimization problems, Journal of Optimization
 757 Theory and Applications, 19 (1976), pp. 261–281, <https://doi.org/10.1007/BF00934096>.
 758 [9] I. D. COOPE AND G. A. WATSON, A projected lagrangian algorithm for semi-infinite programming, Mathematical
 759 Programming, 32 (1985), pp. 337–356, <https://doi.org/10.1007/BF01582053>.
 760 [10] M. DIEHL, B. HOUSKA, O. STEIN, AND P. STEUERMANN, A lifting method for generalized semi-infinite programs
 761 based on lower level wolfe duality, Computational Optimization and Applications, 54 (2013), pp. 189–210,
 762 <https://doi.org/10.1007/s10589-012-9489-4>.
 763 [11] H. DJELASSI, A. MITSOS, AND O. STEIN, Recent advances in nonconvex semi-infinite programming: Applications
 764 and algorithms, EURO Journal on Computational Optimization, 9 (2021), <https://doi.org/10.1016/j.ejco.2021.100006>.
 765
 766 [12] S. FANG, C. LIN, AND S. WU, Solving quadratic semi-infinite programming problems by using relaxed
 767 cutting-plane scheme, Journal of Computational and Applied Mathematics, 129 (2001), pp. 89–104,
 768 [https://doi.org/10.1016/S0377-0427\(00\)00544-6](https://doi.org/10.1016/S0377-0427(00)00544-6).
 769 [13] C. A. FLOUDAS AND O. STEIN, The adaptive convexification algorithm: A feasible point method for
 770 semi-infinite programming, SIAM Journal on Optimization, 18 (2008), pp. 1187–1208, <https://doi.org/10.1137/060657741>.
 771
 772 [14] R. FOURER, D. M. GAY, AND B. W. KERNIGHAN, AMPL: A Modeling Language for Mathematical Programming,
 773 Cengage Learning, Boston, MA, 2002.
 774 [15] P. GRIBIK, A central-cutting-plane algorithm for semi-infinite programming problems, in Semi-infinite pro-
 775 gramming, Springer, 1979, pp. 66–82, <https://doi.org/10.1007/BFb0003884>.
 776 [16] L. GUROBI OPTIMIZATION, Gurobi optimizer reference manual, 2021, <http://www.gurobi.com>.
 777 [17] R. HETTICH, A review of numerical methods for semi-infinite optimization, Semi-infinite programming and
 778 applications, (1983), pp. 158–178, https://doi.org/10.1007/978-3-642-46477-5_11.
 779 [18] R. HETTICH, An implementation of a discretization method for semi-infinite programming, Mathematical Pro-
 780 gramming, 34 (1986), pp. 354–361, <https://doi.org/10.1007/BF01582235>.
 781 [19] R. HETTICH AND K. O. KORTANEK, Semi-infinite programming: Theory, methods, and applications, SIAM
 782 Review, 35 (1993), pp. 380–429, <https://doi.org/10.1137/1035089>.
 783 [20] J. HIRIART-URRUTY AND C. LEMARÉCHAL, Convex analysis and minimization algorithms I: Fundamentals,
 784 vol. 305, Springer-Verlag Berlin Heidelberg, 2013, <https://doi.org/10.1007/978-3-662-02796-7>.
 785 [21] A. KAPLAN AND R. TICHATSCHKE, A regularized penalty method for solving convex semi-infinite programs,

- Optimization, 26 (1992), pp. 215–228, <https://doi.org/10.1080/02331939208843853>.
- [22] J. KELLEY, JR., The cutting-plane method for solving convex programs, Journal of the society for Industrial and Applied Mathematics, 8 (1960), pp. 703–712, <https://www.jstor.org/stable/2099058>.
- [23] K. O. KORTANEK AND H. NO, A central cutting plane algorithm for convex semi-infinite programming problems, SIAM Journal on optimization, 3 (1993), pp. 901–918, <https://doi.org/10.1137/0803047>.
- [24] P. LAURENT AND C. CARASSO, An algorithm of successive minimization in convex programming, RAIRO. Analyse numérique, 12 (1978), pp. 377–400, http://www.numdam.org/item/M2AN_1978__12_4_377_0/.
- [25] E. LEVITIN AND R. TICHATSCHKE, A branch-and-bound approach for solving a class of generalized semi-infinite programming problems, Journal of Global Optimization, 13 (1998), pp. 299–315, <https://doi.org/10.1023/A:1008245113420>.
- [26] L. LIBERTI, S. CAFIERI, AND F. TARISSAN, Reformulations in mathematical programming: A computational approach, in Foundations of Computational Intelligence Volume 3: Global Optimization, A. Abraham et al., eds., Springer, Berlin, Heidelberg, 2009, pp. 153–234, https://doi.org/10.1007/978-3-642-01085-9_7.
- [27] F. LOCATELLO, M. TSCHANNEN, G. RÄTSCH, AND M. JAGGI, Greedy algorithms for cone constrained optimization with convergence guarantees, arXiv preprint, (2017), <http://arxiv.org/abs/1705.11041>.
- [28] A. MARENDET, A. GOLDSZTEJN, G. CHABERT, AND C. JERMANN, A standard branch-and-bound approach for nonlinear semi-infinite problems, European Journal of Operational Research, 282 (2020), pp. 438–452, <https://doi.org/10.1016/j.ejor.2019.10.025>.
- [29] A. MITSOS, Global optimization of semi-infinite programs via restriction of the right-hand side, Optimization, 60 (2011), pp. 1291–1308, <https://doi.org/10.1080/02331934.2010.527970>.
- [30] R. REEMTSEN, Discretization methods for the solution of semi-infinite programming problems, Journal of Optimization Theory and Applications, 71 (1991), pp. 85–103, <https://doi.org/10.1007/BF00940041>.
- [31] R. REEMTSEN AND S. GÖRNER, Numerical methods for semi-infinite programming: A survey, in Reemtsen and Rückmann [32], pp. 195–275, https://doi.org/10.1007/978-1-4757-2868-2_7.
- [32] R. REEMTSEN AND J. RÜCKMANN, eds., Semi-Infinite Programming, Springer, Boston, 1998.
- [33] R. T. ROCKAFELLAR AND R. J.-B. WETS, Variational analysis, vol. 317, Springer, Berlin, Heidelberg, 2009, <https://doi.org/10.1007/978-3-642-02431-3>.
- [34] G. ROMANO, New results in subdifferential calculus with applications to convex optimization, Applied Mathematics and Optimization, 32 (1995), pp. 213–234, <https://doi.org/10.1007/BF01187900>.
- [35] U. SCHÄTTLER, An interior-point method for semi-infinite programming problems, Annals of Operations Research, 62 (1996), pp. 277–301, <https://doi.org/10.1007/BF02206820>.
- [36] M. SION, On general minimax theorems, Pacific Journal of mathematics, 8 (1958), pp. 171–176, <https://doi.org/pjm/1103040253>.
- [37] G. SONNEVEND, A new class of a high order interior point method for the solution of convex semiinfinite optimization problems, in Computational Optimal Control, Springer, 1994, pp. 193–211, https://doi.org/10.1007/978-3-0348-8497-6_16.
- [38] O. STEIN, Bi-level strategies in semi-infinite programming, vol. 71 of Nonconvex Optimization and Its Applications, Springer Boston, MA, 2013, <https://doi.org/10.1007/978-1-4419-9164-5>.
- [39] O. STEIN AND P. STEUERMANN, The adaptive convexification algorithm for semi-infinite programming with arbitrary index sets, Mathematical programming, 136 (2012), pp. 183–207, <https://doi.org/10.1007/s10107-012-0556-5>.
- [40] O. STEIN AND G. STILL, Solving semi-infinite optimization problems with interior point techniques, SIAM J. Control and Optimization, 42 (2003), pp. 769–788, <https://doi.org/10.1137/S0363012901398393>.
- [41] G. STILL, Discretization in semi-infinite programming: the rate of convergence, Mathematical programming, 91 (2001), pp. 53–69, <https://doi.org/10.1007/s101070100239>.
- [42] Y. TANAKA, M. FUKUSHIMA, AND T. IBARAKI, A globally convergent SQP method for semi-infinite nonlinear optimization, Journal of Computational and Applied Mathematics, 23 (1988), pp. 141–153, [https://doi.org/10.1016/0377-0427\(88\)90276-2](https://doi.org/10.1016/0377-0427(88)90276-2).
- [43] R. TICHATSCHKE AND V. NEBELING, A cutting-plane method for quadratic semi infinite programming problems, Optimization, 19 (1988), pp. 803–817, <https://doi.org/10.1080/02331938808843393>.
- [44] H. TUY, Convex Analysis and Global Optimization, vol. 22, Springer, Boston, MA, 2 ed., 1998, <https://doi.org/10.1007/978-3-319-31484-6>.
- [45] G. VAN ROSSUM AND F. L. DRAKE, JR., Python tutorial, Centrum voor Wiskunde en Informatica Amsterdam, The Netherlands, 1995.
- [46] L. VANDENBERGHE AND S. BOYD, Semidefinite programming, SIAM review, 38 (1996), pp. 49–95, <https://doi.org/10.1137/1038003>.
- [47] L. ZHANG, S.-Y. WU, AND M. A. LÓPEZ, A new exchange method for convex semi-infinite programming, SIAM Journal on optimization, 20 (2010), pp. 2959–2977, <https://doi.org/10.1137/090767133>.

844 **Appendices** We report in this section the proof of convergence of Algorithm 3.1, as well as
 845 the proofs of all the lemmata and corollaries introduced in the paper. While the convergence of the
 846 cutting plane algorithm is well-known in literature (even if we prove here the convergence of our
 847 specific Algorithm 3.1 from scratch, for sake of completeness), the other proofs reported in these
 848 appendices are new results.

849 **Appendix A. Convergence proof of CP algorithm.** In this section, a convergence
 850 proof for Algorithm 3.1 is given. Since $Q(x)$ and $q(x)$ are linear w.r.t. x , the function $f(x, y) =$
 851 $\frac{1}{2}y^\top Q(x)y + q(x)^\top y$ is continuously differentiable, and therefore Lipschitz continuous on the compact
 852 set $\mathcal{X} \times \mathcal{F}$ (see Assumptions 2 and 5), with $H > 0$ an associated Lipschitz constant. Moreover,
 853 $x \mapsto \text{val}(\mathbf{P}_x)$ is continuous over the compact feasible set \mathcal{X} , as shown, e.g., in [33, Th. 1.17], and [3,
 854 Sec. 4.2]. Based on these observations, we prove the convergence of the algorithm.

855 **THEOREM A.1.** *Assume that \mathcal{X} and \mathcal{F} are compact and that $\epsilon = 0$, Algorithm 3.1 either ter-*
 856 *minates in $K \in \mathbb{N}^*$ iterations, in which case x^K is a solution of (SIP), or generates an infinite*
 857 *sequence $(x^k)_{k \in \mathbb{N}^*}$ with the following convergence guarantees:*

- 858 • *feasibility error:* $\epsilon_k = (\text{val}(\mathbf{P}_{x^k}) - h(x^k))^- \rightarrow 0$,
- 859 • *objective error:* $\delta_k = \text{val}(\mathbf{SIP}) - F(x^k) \rightarrow 0$.

860 *Proof.* If Algorithm 3.1 terminates at iteration $K \in \mathbb{N}^*$, x^K is feasible in (SIP), i.e., $x^K \in \mathcal{X}$ and
 861 $\text{val}(\mathbf{P}_{x^K}) \geq h(x^K)$, which implies that $F(x^K) \geq \text{val}(\mathbf{SIP})$. At the same time $F(x^K) = \text{val}(R_K) \leq$
 862 $\text{val}(\mathbf{SIP})$, being (R_k) a relaxation of (SIP) by definition. Thus, $F(x^K) = \text{val}(\mathbf{SIP})$, and x^K is an
 863 optimal solution of (SIP).

864 Let us suppose now that the stopping test is never satisfied. In this context, we prove first
 865 the convergence of the feasibility error ϵ_k towards 0. For any $k \in \mathbb{N}^*$, we have that $\text{val}(\mathbf{P}_{x^k}) =$
 866 $\frac{1}{2}y^{k\top} Q(x^k)y^k + q(x^k)^\top y^k = f(x^k, y^k)$, thus $\epsilon_k = (f(x^k, y^k) - h(x^k))^-$. Since $f(x, y)$, $h(x)$ and the
 867 negative part function are continuous, and since both x^k and y^k are bounded, the sequence ϵ_k is
 868 also bounded. According to Bolzano-Weierstrass theorem [1], this bounded sequence has at least
 869 a convergent subsequence. In the following, we define any convergent subsequence extracted from
 870 ϵ_k as $\epsilon_{\psi_0(k)}$, where $\psi_0 : \mathbb{N}^* \mapsto \mathbb{N}^*$ is an increasing application. Defining as $\epsilon_* \in \mathbb{R}$ the limit of this
 871 convergent subsequence, we will show that this limit value is in fact 0.

872 The sequence $(y^{\psi_0(k)}, \epsilon_{\psi_0(k)})$ is a subsequence of the bounded sequence (y^k, ϵ_k) , therefore it is
 873 bounded. According to the Bolzano-Weierstrass theorem, sequence $(y^{\psi_0(k)}, \epsilon_{\psi_0(k)})$ has a convergent
 874 subsequence $(y^{\psi(k)}, \epsilon_{\psi(k)})$. Since $\epsilon_{\psi(k)}$ is a convergent subsequence of $\epsilon_{\psi_0(k)}$, $\epsilon_{\psi(k)} \rightarrow \epsilon_*$ holds.
 875 Because $\psi(k-1) < \psi(k)$ by definition of ψ , the cut related to $y^{\psi(k-1)}$ is a constraint of problem
 876 $R_{\psi(k)}$ (added by Algorithm 3.1 at iteration $k-1$). Thus, $f(x^{\psi(k)}, y^{\psi(k-1)}) - h(x^{\psi(k)}) \geq 0$, and

$$\begin{aligned} 877 f(x^{\psi(k)}, y^{\psi(k)}) - h(x^{\psi(k)}) &= f(x^{\psi(k)}, y^{\psi(k)}) - f(x^{\psi(k)}, y^{\psi(k-1)}) + f(x^{\psi(k)}, y^{\psi(k-1)}) - h(x^{\psi(k)}) \\ 878 &\geq f(x^{\psi(k)}, y^{\psi(k)}) - f(x^{\psi(k)}, y^{\psi(k-1)}). \end{aligned}$$

879 Being the negative part function decreasing, $\epsilon_{\psi(k)} = (f(x^{\psi(k)}, y^{\psi(k)}) - h(x^{\psi(k)}))^-$ is less than or
 880 equal to $(f(x^{\psi(k)}, y^{\psi(k)}) - f(x^{\psi(k)}, y^{\psi(k-1)}))^-$. Therefore $\epsilon_{\psi(k)} \leq |f(x^{\psi(k)}, y^{\psi(k)}) - f(x^{\psi(k)}, y^{\psi(k-1)})|$.
 881 From this last result and the fact that f is H -Lipschitz continuous, we deduce that

$$882 \text{(A.1)} \quad \epsilon_{\psi(k)} \leq H \left\| \begin{pmatrix} x^{\psi(k)} \\ y^{\psi(k)} \end{pmatrix} - \begin{pmatrix} x^{\psi(k-1)} \\ y^{\psi(k-1)} \end{pmatrix} \right\| = H \|y^{\psi(k)} - y^{\psi(k-1)}\|.$$

883 As $y^{\psi(k)}$ is convergent, we know that $\|y^{\psi(k)} - y^{\psi(k-1)}\| \rightarrow 0$. Being $\epsilon_{\psi(k)}$ nonnegative, we deduce
 884 from Eq. (A.1) that $\epsilon_{\psi(k)} \rightarrow 0$, and thus, $\epsilon_* = 0$.

885 We proved that the sequence ϵ_k is bounded, and that any converging subsequence converge
 886 towards 0, thus we can conclude that ϵ_k converges towards 0 itself, according to a well-known
 887 result in analysis [1]. Based on this result, we prove now the second point, i.e., the convergence of
 888 objective error. We know that

$$889 \quad (\text{A.2}) \quad \forall k \in \mathbb{N}^* \quad F(x^k) \in [F(x^1), \text{val}(\text{SIP})],$$

890 therefore the increasing sequence $F(x^k)$ is bounded, and thus, converging. Since x^k is bounded, we
 891 can derive a converging subsequence $x^{\phi(k)} \rightarrow x^*$ with $\phi : \mathbb{N}^* \mapsto \mathbb{N}^*$ being an increasing func-
 892 tion. The associated feasibility error is $\epsilon_{\phi(k)} = (\text{val}(\mathbf{P}_{x^{\phi(k)}}) - h(x^{\phi(k)}))^-$. On the one hand,
 893 being $\epsilon_{\phi(k)}$ a subsequence of ϵ_k which converges towards zero, $\epsilon_{\phi(k)} \rightarrow 0$. On the other hand,
 894 $\epsilon_{\phi(k)} \rightarrow (\text{val}(\mathbf{P}_{x^*}) - h(x^*))^-$ holds by continuity of $x \mapsto \text{val}(\mathbf{P}_x)$ and $h(x)$. By uniqueness of the
 895 limit, $(\text{val}(\mathbf{P}_{x^*}) - h(x^*))^- = 0$. Therefore, $x^* \in \mathcal{X}$ is feasible in (SIP) and $F(x^*) \geq \text{val}(\text{SIP})$. From
 896 (A.2) we also know that $F(x^*) \leq \text{val}(\text{SIP})$, and thus $F(x^*) = \text{val}(\text{SIP})$. We can conclude that $F(x^k)$
 897 is bounded and admits a unique limit point which is $\text{val}(\text{SIP})$. Hence, $\delta_k \rightarrow 0$. \square

898 Appendix B. Proofs of Lemmata in Section 3.

899 **B.1. Proof of Lemma 3.1.** We denote by $\hat{x} \in \mathcal{X}$ the primal feasible solution s.t. $g(\hat{x}, y) =$
 900 $\frac{1}{2}y^\top Q(\hat{x})y + q(\hat{x})^\top y - h(\hat{x}) > 0$ for all $y \in \mathcal{F}$. Since the set \mathcal{F} is compact and the function $g(\hat{x}, y)$
 901 is continuous in y and positive, it exists $c > 0$ s.t. $g(\hat{x}, y) \geq c$ for all $y \in \mathcal{F}$. For any $Y \in \mathcal{K}$, we
 902 have that $Y = \sum_{k=1}^p \lambda_k M(y^k)$, for an integer $p \in \mathbb{N}$, vectors $y^1, \dots, y^p \in \mathcal{F}$ and nonnegative scalars
 903 $\lambda_1, \dots, \lambda_p \in \mathbb{R}_+$. Since $\langle \mathcal{G}(\hat{x}), M(y) \rangle = \frac{1}{2}y^\top Q(\hat{x})y + q(\hat{x})^\top y - h(\hat{x})$ for any $y \in \mathcal{F}$, the following
 904 holds by linearity:

$$905 \quad \langle \mathcal{G}(\hat{x}), Y \rangle = \langle \mathcal{G}(\hat{x}), \sum_{k=1}^p \lambda_k M(y^k) \rangle = \sum_{k=1}^p \lambda_k \langle \mathcal{G}(\hat{x}), M(y^k) \rangle \geq \sum_{k=1}^p \lambda_k c = Y_{n+1, n+1} c.$$

907 Moreover, by definition of θ , for any $Y \in \mathcal{K}$ $\theta(Y) = \min_{x \in \mathcal{X}} F(x) - \langle \mathcal{G}(x), Y \rangle \leq F(\hat{x}) - \langle \mathcal{G}(\hat{x}), Y \rangle \leq$
 908 $F(\hat{x}) - Y_{n+1, n+1} c$. We take then a maximizing sequence $(Y^k)_{k \in \mathbb{N}}$ of problem (3.1). Defining $V_{\text{SIP}'} =$
 909 $\text{val}(\text{SIP}')$, we know that $\theta(Y^k) \rightarrow V_{\text{SIP}'}$ and hence, it exists $j \in \mathbb{N}$ s.t. for all $k \geq j$, $\theta(Y^k) \geq V_{\text{SIP}'} - 1$.
 910 This implies that, for all $k \geq j$, $0 \leq Y_{n+1, n+1}^k \leq \frac{F(\hat{x}) - V_{\text{SIP}'} + 1}{c}$. Defining $\bar{\tau} = \frac{F(\hat{x}) - V_{\text{SIP}'} + 1}{c}$, we deduce
 911 that $\forall k \geq j$, Y^k belongs to $\bar{\tau} \text{conv}(\mathcal{P})$, which is compact. Thus, the sequence $(Y^k)_{k \in \mathbb{N}}$ admits an
 912 accumulation point Y^* , s.t. $\theta(Y^*) = V_{\text{SIP}'}$ by continuity of θ .

913 **B.2. Proof of Lemma 3.2.** This property follows from the 1st order optimality condition at
 914 1 of the differentiable function $w(t) = \theta(tY^k)$. Indeed, $w'(1) = \langle \nabla \theta(Y^k), Y^k \rangle = 0$, because (i) 1 is
 915 optimal for w since $Y^k \in \arg \max_{Y \in \text{cone}(B_k)} \theta(Y)$, (ii) 1 lies in the interior of the definition domain of w .

916 **B.3. Proof of Lemma 3.3.** For the purpose of this proof, we introduce the linear operator
 917 \mathcal{Q}^* , defined as the adjoint operator of the linear (by Assumption 3) operator $\mathcal{Q}(x)$. With this
 918 notation, we have that $\langle \mathcal{Q}(x), Y \rangle = x^\top (\mathcal{Q}^* Y)$. We also denote by $\|\mathcal{Q}^*\|_{\text{op}}$ the operator norm of \mathcal{Q}^* .
 919 We notice that the image of the bounded set \mathcal{X} by the subdifferential mapping $\partial h(\mathcal{X}) = \bigcup_{x \in \mathcal{X}} \partial h(x)$
 920 is bounded according to Th.6.2.2 in [20, Chap.]. Hence $D \geq 0$ exists such that

$$921 \quad (\text{B.1}) \quad \forall x \in \mathcal{X}, \forall s \in \partial h(x), \quad \|s\|_2 \leq D.$$

922 Given $Y, Y' \in \mathcal{K}$, we are now going to prove that $\|\nabla \theta(Y) - \nabla \theta(Y')\|_2 \leq L \|Y - Y'\|_2$ for a constant
 923 L that is independent from Y and Y' . Being $i_{\mathcal{X}}(x)$ the indicator function of the set \mathcal{X} , we introduce
 924 the functions $w(x) = \mathcal{L}(x, Y) + i_{\mathcal{X}}(x)$ and $w'(x) = \mathcal{L}(x, Y') + i_{\mathcal{X}}(x)$. According to Assumptions
 925 6, as well as 2, 3, and 4 we remark that application w (resp. w') is μ -strongly convex because
 926 it is the sum of the μ -strongly convex function $F(x)$ and the function $-\langle \mathcal{G}(x), Y \rangle + i_{\mathcal{X}}(x)$ (resp.

927 $-\langle \mathcal{G}(x), Y' \rangle + i_{\mathcal{X}}(x)$ convex in x . Being u (resp. u') the unique (for strong convexity) minimum of
 928 function w (resp. w') the optimality conditions of function w , and w' respectively read

929 (B.2)
$$0 \in \partial w(u),$$

930 (B.3)
$$0 \in \partial w'(u').$$

931 We remark that $w'(x) = F(x) + i_{\mathcal{X}}(x) + Y'_{n+1, n+1}h(x) - x^\top(Q^*Y')$. The function $F(x) + i_{\mathcal{X}}(x)$ is
 932 convex in x as a sum of convex functions; the function $Y'_{n+1, n+1}h(x)$ is convex in x since $h(x)$
 933 is convex and $Y'_{n+1, n+1} \geq 0$ by definition of cone \mathcal{K} ; $-x^\top(Q^*Y')$ is linear (convex) in x . The
 934 intersection of the relative interiors of the domains of these convex functions is $\text{ri}(\mathcal{X})$. Being \mathcal{X}
 935 a finite-dimensional convex set, $\text{ri}(\mathcal{X}) \neq \emptyset$ [44, Prop. 1.9]. Hence, the subdifferential of the sum
 936 is the sum of the subdifferentials [34, Th. 2.1], and the subdifferential of function w' at u' reads
 937 $\partial w'(u') = \partial(F + i_{\mathcal{X}})(u') - Q^*Y' + Y'_{n+1, n+1}\partial h(u')$. Based on this decomposition, it follows from (B.3)
 938 that $\exists g_0 \in \partial(F + i_{\mathcal{X}})(u')$, $g_1 \in \partial h(u')$ such that

940 (B.4)
$$g_0 - Q^*Y' + Y'_{n+1, n+1}g_1 = 0.$$

941 Additionally, since $w(x) = F(x) + i_{\mathcal{X}}(x) - x^\top(Q^*Y) + Y_{n+1, n+1}h(x)$, and $g_0 \in \partial(F + i_{\mathcal{X}})(u')$, $g_1 \in \partial h(u')$,
 942 $g_0 - Q^*Y + Y_{n+1, n+1}g_1 \in \partial w(u')$. Combining this with Eq. (B.4), we deduce:

943 (B.5)
$$Q^*(Y' - Y) + (Y_{n+1, n+1} - Y'_{n+1, n+1})g_1 \in \partial w(u').$$

944 Applying Th.6.1.2 in [20, Chap.], the μ -strong convexity of w gives that, for any $s_1 \in \partial w(u)$ and
 945 $s_2 \in \partial w(u')$, $\langle s_2 - s_1, u' - u \rangle \geq \mu\|u - u'\|_2^2$. Moreover, due to the Cauchy-Schwartz inequality,
 946 $\|s_1 - s_2\|_2\|u - u'\|_2 \geq \langle s_2 - s_1, u' - u \rangle$. Therefore, $\|s_2 - s_1\|_2 \geq \mu\|u - u'\|_2$ holds for any $s_1 \in \partial w(u)$
 947 and $s_2 \in \partial w(u')$. Since $0 \in \partial w(u)$ according to (B.2), and $Q^*(Y' - Y) + (Y_{n+1, n+1} - Y'_{n+1, n+1})g_1 \in$
 948 $\partial w(u')$ according to (B.5), we deduce that $\|Q^*(Y' - Y) + (Y_{n+1, n+1} - Y'_{n+1, n+1})g_1 - 0\|_2 \geq \mu\|u - u'\|_2$.
 949 According to the triangle inequality $\|Q^*(Y' - Y)\|_2 + |Y_{n+1, n+1} - Y'_{n+1, n+1}|\|g_1\|_2 \geq \mu\|u - u'\|_2$, and
 950 thus, since $\|Y - Y'\|_2 \geq |Y_{n+1, n+1} - Y'_{n+1, n+1}|$, $\|Q^*\|_{\text{op}}\|Y - Y'\|_2 + \|Y - Y'\|_2\|g_1\|_2 \geq \mu\|u - u'\|_2$.

951 Defining $B = \|Q^*\|_{\text{op}} + D$ and using that $\|g_1\|_2 \leq D$, which holds for (B.1), we know that
 952 $B\|Y - Y'\|_2 \geq \mu\|u - u'\|_2$. According to Assumption 4, $h(x)$ is Lipschitz continuous and so are
 953 $q(x)$ and $Q(x)$ by Assumption 3. Hence, it exists a constant $K > 0$ such that $\mathcal{G}(x)$ is K -Lipschitz
 954 continuous. We deduce that $K\|u - u'\|_2 \geq \|\mathcal{G}(u) - \mathcal{G}(u')\|_2$, and, consequently, $\|Y - Y'\|_2 \geq$
 955 $\frac{\mu}{BK}\|\mathcal{G}(u) - \mathcal{G}(u')\|_2$. We define the constant $L = \frac{BK}{\mu}$, which is clearly independent from Y, Y', u and
 956 u' . Since $\nabla\theta(Y) = -\mathcal{G}(u)$ and $\nabla\theta(Y') = -\mathcal{G}(u')$, we deduce that $L\|Y - Y'\|_2 \geq \|\nabla\theta(Y) - \nabla\theta(Y')\|_2$,
 957 which concludes the proof.

958 **B.4. Proof of Lemma 3.4.** For any $Y, Z \in \mathcal{K}$ and $\gamma > 0$, we obtain by integration that

959 (B.6)
$$\theta(Y + \gamma Z) - \theta(Y) = \int_0^\gamma \langle \nabla\theta(Y + tZ), Z \rangle dt = \gamma \langle \nabla\theta(Y), Z \rangle + \int_0^\gamma \langle \nabla\theta(Y + tZ) - \nabla\theta(Y), Z \rangle dt.$$

 960 Since $\langle \nabla\theta(Y + tZ) - \nabla\theta(Y), Z \rangle \geq -|\langle \nabla\theta(Y + tZ) - \nabla\theta(Y), Z \rangle|$, using Cauchy-Schwartz inequality
 961 and L -smoothness of θ , we know that $\langle \nabla\theta(Y + tZ) - \nabla\theta(Y), Z \rangle \geq -\|\nabla\theta(Y + tZ) - \nabla\theta(Y)\|_2\|Z\|_2 \geq$
 962 $-tL\|Z\|_2^2$. Combining this with Eq. (B.6), we deduce that $\theta(Y + \gamma Z) - \theta(Y) \geq \gamma \langle \nabla\theta(Y), Z \rangle -$
 963 $\int_{t=0}^\gamma tL\|Z\|_2^2 dt$, which yields finally that $\theta(Y + \gamma Z) - \theta(Y) \geq \gamma \langle \nabla\theta(Y), Z \rangle - \frac{L\|Z\|_2^2}{2}\gamma^2$.

964 **Appendix C. Proofs of Lemmata in Section 4.**

965 **C.1. Proof of Lemma 4.1.** The inequality $\text{val}(\text{SDP}_x) \leq \text{val}(\text{P}_x)$ follows from the relaxation
 966 of the rank-constraint. We now assume that $Q(x)$ is PSD and prove that $\text{val}(\text{SDP}_x) \geq \text{val}(\text{P}_x)$ holds.
 967 Given a matrix Y feasible for (SDP_x) , we denote by $u_1, \dots, u_{n+1} \in \mathbb{R}^{n+1}$ a basis of eigenvectors
 968 of Y (which is PSD) and their respective eigenvalues $v_1, \dots, v_{n+1} \in \mathbb{R}_+$. Let us introduce the two
 969 following index sets: $I = \{i \in \{1, \dots, n+1\} : (u_i)_{n+1} \neq 0\}$ and $J = \{i \in \{1, \dots, n+1\} : (u_i)_{n+1} =$
 970 $0\}$. We have then: $I \cup J = \{1, \dots, n+1\}$. Moreover,

- 971 • if $i \in I$: we define the nonnegative scalar $\mu_i = v_i(u_i)_{n+1}^2$ and $y_i \in \mathbb{R}^n$ s.t. $u_i = (u_i)_{n+1} \begin{pmatrix} y_i \\ 1 \end{pmatrix}$
 972 • if $i \in J$: we define the nonnegative scalar $\nu_i = v_i$ and $z_i \in \mathbb{R}^n$ s.t. $u_i = \begin{pmatrix} z_i \\ 0 \end{pmatrix}$.

With this notation, we have that

$$Y = \sum_{i=1}^{n+1} v_i u_i u_i^\top = \sum_{i \in I} v_i (u_i)_{n+1}^2 \begin{pmatrix} y_i \\ 1 \end{pmatrix} \begin{pmatrix} y_i \\ 1 \end{pmatrix}^\top + \sum_{i \in J} v_i \begin{pmatrix} z_i \\ 0 \end{pmatrix} \begin{pmatrix} z_i \\ 0 \end{pmatrix}^\top = \sum_{i \in I} \mu_i \begin{pmatrix} y_i y_i^\top & y_i \\ y_i^\top & 1 \end{pmatrix} + \sum_{i \in J} \nu_i \begin{pmatrix} z_i z_i^\top & \mathbf{0} \\ \mathbf{0}^\top & 0 \end{pmatrix},$$

973 where $\mathbf{0}$ is the null n -dimensional vector (whereas 0_n is the $n \times n$ null matrix). Let us define the
 974 vector $\bar{y} = \sum_{i \in I} \mu_i y_i$. Its obj. value in (\mathbf{P}_x) is smaller than the obj. value of Y in (\mathbf{SDP}_x) . In fact:

$$975 \quad (\text{C.1}) \quad \langle \mathcal{Q}(x), Y \rangle = \sum_{i \in I} \mu_i f(x, y_i) + \frac{1}{2} \sum_{i \in J} \nu_i z_i^\top \mathcal{Q}(x) z_i \geq \sum_{i \in I} \mu_i f(x, y_i) \geq f(x, \sum_{i \in I} \mu_i y_i) = f(x, \bar{y}).$$

976 The first inequality is due to $\mathcal{Q}(x) \succeq 0$ and $\nu_i \geq 0$. The second inequality derives from $\sum_{i \in I} \mu_i =$
 977 $Y_{n+1, n+1} = 1$, and from the convexity of function $f(x, y)$ (Jensen inequality). Moreover, since Y
 978 is feasible in (\mathbf{SDP}_x) , for each $j \in \{1, \dots, r\}$ we have $b_j \geq \langle \mathcal{A}_j, Y \rangle = \sum_{i \in I} \mu_i a_j^\top y_i = a_j^\top \bar{y}$, which
 979 means that \bar{y} is feasible in (\mathbf{P}_x) too. This implies that $f(x, \bar{y}) \geq \text{val}(\mathbf{P}_x)$ and together with (C.1),
 980 that $\langle \mathcal{Q}(x), Y \rangle \geq \text{val}(\mathbf{P}_x)$. This being true for any matrix Y feasible in (\mathbf{SDP}_x) , we conclude that
 981 $\text{val}(\mathbf{SDP}_x) \geq \text{val}(\mathbf{P}_x)$. This proves that $\text{val}(\mathbf{SDP}_x) = \text{val}(\mathbf{P}_x)$.

982 **C.2. Proof of Lemma 4.2.** The Lagrangian of problem (\mathbf{SDP}_x) is defined over $Y \in S_{n+1}^+(\mathbb{R})$,
 983 $\lambda \in \mathbb{R}_+^r, \alpha \in \mathbb{R}_+, \beta \in \mathbb{R}$ and reads $L_x(Y, \lambda, \alpha, \beta) = \langle \mathcal{Q}(x), Y \rangle + \sum_{j=1}^r [\lambda_j (\langle \mathcal{A}_j, Y \rangle - b_j)] + \alpha(\text{Tr}(Y) - 1 -$
 984 $\rho^2) + \beta(Y_{n+1, n+1} - 1) = - \sum_{j=1}^r \lambda_j b_j - \alpha(1 + \rho^2) - \beta + \langle \mathcal{Q}(x) + \sum_{j=1}^r \lambda_j \mathcal{A}_j + \alpha I_{n+1} + \beta E, Y \rangle$. The Lagrangian
 985 dual problem of (\mathbf{SDP}_x) is: $\max_{\lambda, \alpha, \beta} \min_Y L_x(Y, \lambda, \alpha, \beta)$. According to Lagrangian expression, it can thus
 986 be written as

$$987 \quad \max_{\substack{\lambda \in \mathbb{R}_+^r \\ \alpha \in \mathbb{R}_+ \\ \beta \in \mathbb{R}}} \left(- \sum_{j=1}^r \lambda_j b_j + \alpha(1 + \rho^2) + \beta \right) + \min_{Y \in S_{n+1}^+(\mathbb{R})} \left(\langle \mathcal{Q}(x) + \sum_{j=1}^r \lambda_j \mathcal{A}_j + \alpha I_{n+1} + \beta E, Y \rangle \right).$$

988 We notice that

$$989 \quad \min_{Y \in S_{n+1}^+(\mathbb{R})} \langle \mathcal{Q}(x) + \sum_{j=1}^r \lambda_j \mathcal{A}_j + \alpha I_{n+1} + \beta E, Y \rangle = \begin{cases} 0 & \text{if } (\mathcal{Q}(x) + \sum_{j=1}^r \lambda_j \mathcal{A}_j + \alpha I_{n+1} + \beta E) \succeq 0 \\ -\infty & \text{otherwise.} \end{cases}$$

990 This proves that the dual problem of (\mathbf{SDP}_x) can be formulated as (\mathbf{DSDP}_x) . To prove that
 991 $\text{val}(\mathbf{SDP}_x) = \text{val}(\mathbf{DSDP}_x)$, we prove that the Slater condition, which is a sufficient condition for
 992 strong duality [46], holds for the dual problem (\mathbf{DSDP}_x) , exploiting the Lagrangian multiplier
 993 associated to the constraint $\text{Tr}(Y) \leq 1 + \rho^2$. We denote by m_x the minimum eigenvalue of $\mathcal{Q}(x)$.
 994 By definition of m_x , the matrix $\mathcal{Q}(x) + (1 - m_x)I_{n+1}$ is positive definite. This is why $(\lambda, \alpha, \beta) =$
 995 $(0, \dots, 0, 1 - m_x, 0)$ is a strictly feasible point of (\mathbf{DSDP}_x) . Hence, the Slater condition holds.

996 **C.3. Proof of Lemma 4.5.** We begin by proving the points 1 and 2, before proving that
 997 the value functions are ζ -Lipschitz. Given any sequence $(x^k)_{k \in \mathbb{N}^*} \in \mathcal{X}$, for all $k \in \mathbb{N}^*$ we have that
 998 $v_k = V_{\mathbf{P}}(x^k)$ and we define \mathcal{U}^k the constraint set in (\mathbf{SDP}_x^k) , i.e., the set

$$999 \quad \mathcal{U}^k := \{Y \succeq 0 : (\forall j \in \{1, \dots, r\}, \langle \mathcal{A}_j, Y \rangle \leq b_j) \wedge (\forall \ell \in \{1, \dots, k-1\}, \langle \mathcal{Q}(x^\ell), Y \rangle \geq v_\ell) \wedge$$

1000 $(\text{Tr}(Y) \leq 1 + \rho^2) \wedge (Y_{n+1, n+1} = 1)\}.$

1002 Thus $V_{\text{SDP}}^k(x) = \min_{Y \in \mathcal{U}^k} \langle \mathcal{Q}(x), Y \rangle$. We remark that for $k = 1$, the second set of constraints is
 1003 empty, thus \mathcal{U}^1 is exactly the constraint set of (SDP_x) , and therefore $V_{\text{SDP}}^1(x) = V_{\text{SDP}}(x)$ for any
 1004 $x \in \mathcal{X}$. Moreover, $\mathcal{U}^{k+1} \subset \mathcal{U}^k$. This is why $V_{\text{SDP}}^k(x) \leq V_{\text{SDP}}^{k+1}(x)$ for any $k \in \mathbb{N}^*$ and any $x \in \mathcal{X}$.
 1005 Finally, as explained in Section 4.5, problem (SDP_x^k) is a relaxation of (P_x) for any $k \in \mathbb{N}^*$, thus
 1006 $V_{\text{SDP}}^k(x) \leq V_{\text{P}}(x)$. With this, we proved point 1.

1007 We fix $k \in \mathbb{N}^*$ and we reason for any $\ell \in \mathbb{N}^*$ s.t. $\ell \leq k - 1$. On the one side, we know that
 1008 $V_{\text{SDP}}^k(x^\ell) \leq V_{\text{P}}(x^\ell)$ from the previous point. On the other side, we know that, for any $Y \in \mathcal{U}^k$,
 1009 $\langle \mathcal{Q}(x^\ell), Y \rangle \geq v_\ell = V_{\text{P}}(x^\ell)$, which implies that $V_{\text{SDP}}^k(x^\ell) = \min_{Y \in \mathcal{U}^k} \langle \mathcal{Q}(x^\ell), Y \rangle \geq V_{\text{P}}(x^\ell)$. Hence,
 1010 $V_{\text{SDP}}^k(x^\ell) = V_{\text{P}}(x^\ell)$, which proves point 2.

1011 We recall that \mathcal{Q}^* is the adjoint operator of the linear operator \mathcal{Q} and $\|\mathcal{Q}^*\|_{\text{op}}$ is the linear
 1012 operator of \mathcal{Q}^* . We define $\Gamma := \max_{Y \in \mathcal{U}^1} \|Y\|_2$, and $\zeta := \|\mathcal{Q}^*\|_{\text{op}} \Gamma$, with ζ not depending on k or on
 1013 the choice of sequence $(x^k)_{k \in \mathbb{N}^*}$. We remark that the function $-V_{\text{SDP}}^k(x)$ reads $\max_{Y \in \mathcal{U}^k} \langle x, -\mathcal{Q}^*(Y) \rangle$,
 1014 with \mathcal{U}^k being a compact set and $\langle x, -\mathcal{Q}^*(Y) \rangle$ a linear function in x for any $Y \in \mathcal{U}^k$. Applying
 1015 [20, Th. VI.4.4.2], we deduce that $-V_{\text{SDP}}^k$ is convex and that, for any $x \in \mathcal{X}$, the subdifferential of
 1016 $-V_{\text{SDP}}^k$ at x is

$$(C.2) \quad \partial(-V_{\text{SDP}}^k)(x) = \{-\mathcal{Q}^*(Y) : (Y \in \mathcal{U}^k) \wedge (-V_{\text{SDP}}^k(x) = \langle x, -\mathcal{Q}^*(Y) \rangle)\}.$$

1018 Combining this with the observation that $\mathcal{U}^k \subset \mathcal{U}^1$, we deduce that $\partial(-V_{\text{SDP}}^k)(x) \subset \{-\mathcal{Q}^*(Y) :$
 1019 $Y \in \mathcal{U}^1\}$. Hence, for any $x \in \mathcal{X}$, and any $s \in \partial(-V_{\text{SDP}}^k)(x)$, we know that

$$(C.3) \quad \|s\| \leq \|\mathcal{Q}^*\|_{\text{op}} \Gamma = \zeta.$$

1021 Let us take any pair $(x, \hat{x}) \in \mathcal{X} \times \mathcal{X}$. Applying convexity inequalities to $-V_{\text{SDP}}^k$, we deduce that for
 1022 any $s \in \partial(-V_{\text{SDP}}^k)(x)$ and $\hat{s} \in \partial(-V_{\text{SDP}}^k)(\hat{x})$, the following holds: $V_{\text{SDP}}^k(\hat{x}) - V_{\text{SDP}}^k(x) \leq (\hat{x} - x)^\top s$ and
 1023 $V_{\text{SDP}}^k(x) - V_{\text{SDP}}^k(\hat{x}) \leq (x - \hat{x})^\top \hat{s}$. We know from the Cauchy-Schwartz inequality and from Eq. (C.3)
 1024 that $(\hat{x} - x)^\top s \leq \zeta \|\hat{x} - x\|$ and $(x - \hat{x})^\top \hat{s} \leq \zeta \|\hat{x} - x\|$.

1025 We deduce that $\|V_{\text{SDP}}^k(\hat{x}) - V_{\text{SDP}}^k(x)\| \leq \zeta \|\hat{x} - x\|$.

1026 **C.4. Proof of Lemma 4.6.** We analyze the variation of the objective function w.r.t. the
 1027 variable x . Since $x^* \in \mathcal{X}$ is a feasible value for variable x , the direction $h = x^* - x^k$ is admissible
 1028 at x^k in the problem (4.9). As $F(x)$ is convex over \mathbb{R}^n , the directional derivative $F'(x^k, h) =$
 1029 $\lim_{t \rightarrow 0^+} \frac{F(x^k + th) - F(x^k)}{t}$ is well-defined. By optimality of x^k , the directional derivative of function
 1030 $F(x) + \frac{\mu_k}{2} \|x - \hat{x}^k\|^2$ in the direction h is non-negative, i.e., $F'(x^k, h) + \mu_k (x^k - \hat{x}^k)^\top h \geq 0$. By
 1031 convexity of $F(x)$, we also have that $F(x^*) - F(x^k) \geq F'(x^k, h)$. Combining this with the previous
 1032 inequality yields $F(x^k) \leq F(x^*) + \mu_k (x^k - \hat{x}^k)^\top (x^* - x^k)$.

1033 **C.5. Proof of Corollary 4.8.** We reason by contradiction: let us assume that the algorithm
 1034 generates an infinite sequence. We know that this implies that $(V_{\text{P}}(x^k) - h(x^k))^- \rightarrow 0$ and $\|x^k -$
 1035 $\hat{x}^k\| \rightarrow 0$. Moreover, since the algorithm does not stop, for all $k \in \mathbb{N}^*$, either $(V_{\text{P}}(x^k) - h(x^k))^- > \epsilon$
 1036 or $\|x^k - \hat{x}^k\| > d$. By case disjunction, we can deduce that either it exists an infinite number of k
 1037 such that $(V_{\text{P}}(x^k) - h(x^k))^- > \epsilon$ and thus $\epsilon = 0$ or an infinite number of k such that $\|x^k - \hat{x}^k\| > d$
 1038 and thus $d = 0$. Hence, if $\epsilon > 0$ and $d > 0$, the algorithm terminates in finite time. As stated in
 1039 Theorem 4.7, the iterate \hat{x}^k is feasible. The fact that $F(\hat{x}^k) \leq \text{val}(\text{SIP}) + d(\mu_k \text{diam}(\mathcal{X}) + J)$ directly
 1040 follows from Eq. (4.15) since $\|x^k - \hat{x}^k\| \leq d$ and $\|x^* - x^k\| \leq \text{diam}(\mathcal{X})$.