CHAPTER 1

A ten page introduction to conic optimization

This background chapter gives an introduction to conic optimization. We do not give proofs, but focus on important (for this thesis) tools and concepts.

1.1. Optimization and computational hardness

Optimization is about maximizing or minimizing a function over a set. The set is typically described more implicitly than just an enumeration of its elements, and the structure in this description is essential in developing good optimization techniques. The set is known as the *feasible set* and its elements the *feasible solutions*. The function is called the *objective function* and its values the *objective values*. We write a minimization problem as $p = \inf_{x \in S} f(x)$, and we often use p to refer to the optimization problem as a whole instead of just its optimal value. We are not only interested in finding the optimal value, but also in finding optimal solutions, and if this is too difficult (or if they do not exist) we seek close to optimal feasible solutions. An important topic is finding certificates asserting the solution's optimality or quality of approximation. In fact, when solving an optimization problem we often intend to find an optimal or near optimal feasible solution together with a certificate.

Linear programming is foundational in conic optimization. Consider the problem of finding a vector x satisfying a linear system $Ax = b$. We can find such an x by Gaussian elimination, but when we also require the entries of x to be nonnegative, then we need algorithms which go further then just solving linear systems. Linear programming is the optimization analogue of this feasibility problem: In a linear program we optimize a linear functional over all nonnegative vectors satisfying some linear system. It is the positivity condition, and not the fact that we are optimizing a functional, that moves a linear problem into the realm of optimization: Using complementary slackness (see Section 1.3) we can add variables and constraints to a linear program so that all its feasible solutions are optimal. Alternatively, we can constrain a minimization problem's objective value to be at most some number b, and then bisect on b to solve the optimization problem by solving a sequence of feasibility problems.

When we discuss the hardness (in some computational model) of solving or approximating a class of optimization problems, we need to define an explicit encoding of the feasible sets and objective functions. In this way it is clear what constitutes the input data for the algorithms. This is important because the efficiency of an algorithm is determined by the dependence of the running time on the input size. Geometrically, linear programming is the optimization of a linear functional over a polyhedron, and although a polyhedron can be described in different ways, when we discuss hardness we assume a facial description. This means the polyhedron is given by all vectors x satisfying some linear inequality $Ax \geq b$. We can, however, use any description which is easily transformable into and derivable from this one, such as the description from the previous paragraph. Linear programs can be solved efficiently in practice by simplex methods, although it is not known whether there exists a polynomial time simplex method. The ellipsoid method can solve a rational linear program in polynomial time (in the bit model) but appears to be too slow in practice. In Section 1.4 we discuss interior point methods which are fast in practice and which can be made to run in polynomial time.

If a linear program's input is rational, then its optimal value is a rational number whose bit size is bounded by a fixed polynomial in the input bit size [**56**, Corollary 10.2a]. For semidefinite programming, which is a broad generalization of linear programming, there exist rational instances whose optimal value is an algebraic number of high degree [**49**]. It is not known whether a polynomial time algorithm for semidefinite programing exists. However, if the feasible set of a semidefinite program contains a ball of radius r and is contained in a ball of radius R, then for each $\varepsilon > 0$ we can find an ε-optimal solution in polynomial time (where ε, r, and R are part of the input of the algorithm). An *ε-optimal* solution is a feasible solution whose objective value is within ε of the optimum.

We distinguish between convex and nonconvex optimization problems, where a convex optimization problem has a convex feasible set and convex (concave) objective function in case it is a minimization (maximization) problem. Convex problems have the advantage that local optima are globally optimal, although this does not mean that they are necessarily easy to solve.

1.2. Lifts and relaxations

When optimization problems are difficult, we can try to use their description to derive easier optimization problems which give information about the original problems. Lifts provide one such technique. A *lift* of an optimization problem is another optimization problem with a surjective map P from its feasible set onto the original problem's feasible set, and whose objective function is given by composing the original objective function with P. This technique originated from the observation that there exist polytopes which are projections of higher dimensional polytopes with drastically simpler facial structure. Lifts contain all information of the original problems; they have the same optimal value and we can project their optimal solutions to optimal solutions of the original problem.

Typically we do not lift a single problem, but we systematically lift an entire class of problems. When the worst case instances in this class are inherently difficult to solve – for instance, the class is NP-hard and P $\neq$ NP – then it appears to be difficult for lifts to recognize the easy problems; that is, all of them will be hard to solve. More successful in this respect are relaxations. A *relaxation* of a problem $\inf_{x \in A} f(x)$ is another problem $\inf_{x \in B} g(x)$ together with an injective map $R \colon A \hookrightarrow B$ such that $g \circ R \leq f$. Relaxations are often obtained by relaxing the constraint set, in which case R is the identity. For example, by removing the integrality constraints in an integer linear program we obtain the linear programming relaxation. Also common are Lagrangian relaxations which we discuss in Section 1.3.

A lift of a relaxation is a relaxation, and we will encounter instances which are naturally interpreted in this way. When R is surjective, the relaxation is a lift, and R and P are each others inverses. Even when R is not surjective, it can happen that it maps optimal solutions to optimal solutions, and in this case we say the relaxation is *sharp*. Any optimization problem $\inf_{x\in S} f(x)$ admits a sharp convex relaxation $\inf_{x\in C} g(x)$ by taking C to be the convex hull of the basis elements δ_x of the vector space $\mathbb{R}^S$ of finitely supported functions $S \to \mathbb{R}$ and g to be the linear functional satisfying $g(\delta_x) = f(x)$ for $x \in S$.

1.3. Conic programming and duality

In a *conic program* we optimize a linear functional over the intersection of a closed convex cone with an affine space. A convex cone K is a subset of a real vector space E such that $ax + by \in K$ for all $a, b \geq 0$ and $x, y \in K$. We define the affine space by the set of solutions to the equation $Ax = b$, where A is a linear operator from E to another real vector space F, and b is an element from F. The objective function is a linear functional $c\colon E \to \mathbb{R}$. A conic program (in standard form) reads

$$p = \inf \left\{ c(x) : x \in K,\, Ax = b \right\}.$$

We can write any convex optimization problem $\inf_{x\in S} f(x)$ as a conic program: First write it as a minimization problem with linear objective $g(x, b) = b$ and convex feasible set $S' = \{(x, b) : x \in S,\, f(x) \leq b\}$, then write it as a conic program over the cone $\{(x, t) : t \geq 0,\, x \in tS'\}$. The power of conic programming, however, lies in the fact that we only need a few classes of convex cones to express a wide variety of optimization problems. The type of optimization problem is encoded by the cone and the problem data by the affine space and objective function. Linear programs are conic programs over a nonnegative orthant cone $\mathbb{R}^n_{\geq 0}$, and semidefinite programs use a cone of positive semidefinite matrices.

Positivity — as modeled by the cone constraints in a conic program — is fundamental in convex optimization. A second fundamental concept is duality. We first discuss Lagrangian duality, which is based on removing constraints and penalizing violations of those constraints in the objective. Consider a problem of the form

$$q = \inf \Big\{ f(x) : x \in S,\, g_i(x) = 0 \text{ for } i \in [l],\, h_j(x) \geq 0 \text{ for } j \in [m] \Big\},$$

where $[l] = \{1, \ldots, l\}$. We call this the primal problem. For simplicity we assume all functions to be real-valued and continuously differentiable, and we assume S to be an open subset of $\mathbb{R}^n$. We define the *Lagrangian* by

$$L\colon S \times \mathbb{R}^l \times \mathbb{R}^m_{\leq 0} \to \mathbb{R},\, (x, u, v) \mapsto f(x) + \sum_{i=1}^{l} u_i g_i(x) + \sum_{j=1}^{m} v_j h_j(x).$$

When $m = 0$, the constrained stationary points of f correspond precisely to the stationary points of L. The geometric explanation is that $\nabla_u L = 0$ forces x to be feasible, and $\nabla_x L = 0$ forces the direction of steepest descent of f at x to be a normal vector of the feasible set. The entries of the vector u in a stationary point (x, u) of L are called *Lagrange multipliers*. In the general case where $m > 0$ the situation is more subtle. The constrained stationary points of L are known as *Karush-Kuhn-Tucker points*. For each such point (x, u, v), the vector x is a constrained stationary point of f. In general not all constrained stationary points of f can be obtained in this way, but there are sufficient conditions known as global constraint qualifications under which this is true. The most well-known is Slater's condition which requires the problem to be convex and to admit a strictly (all inequalities are strictly satisfied) feasible point. When the functions $f, g_1, \ldots, g_m$ are convex, the set S is convex, and the functions $h_1, \ldots, h_m$ are linear, then the problem is convex. In convex problems the global constrained minima are precisely the constrained stationary points.

We define the *Lagrangian dual function*

$$R\colon \mathbb{R}^l \times \mathbb{R}^m_{\leq 0} \to \mathbb{R},\ R(u, v) = \inf_{x \in S} L(x, u, v),$$

so that $R(u, v)$ is a relaxation of q for each u and $v \leq 0$. The *Lagrangian dual problem* is given by maximizing this function over its domain:

$$q^* = \sup_{(u,v) \in \mathbb{R}^l \times \mathbb{R}^m_{\leq 0}} R(u, v).$$

The primal problem can be written as

$$\inf_{x \in S} \sup_{(u,v) \in \mathbb{R}^l \times \mathbb{R}^m_{\leq 0}} L(x, u, v),$$

so that we simply interchange sup and inf to go from the primal to the dual problem. A global constraint qualification such as Slater's condition guarantees the optimal values are the same.

To apply Lagrangian duality to general conic programs we extend the above discussion to conic constraints. In q^* the objective function is an optimization problem itself, and the reduction to a more explicit form requires problem specific information. An advantage of conic programming is that all nonlinearities are contained in the cone constraint, and an explicit description of the dual cone is all we need for an explicit description of the dual program. The dual program is a conic program over the dual cone, and the situation is symmetric in the sense that we recover the original problem by taking the dual again.

Let E^* and F^* be the algebraic duals of E and F; that is, the vector spaces of real-valued linear functionals on E and F. Then $c \in E^*$. We have two nondegenerate bilinear pairings $E \times E^* \to \mathbb{R}$ and $F \times F^* \to \mathbb{R}$, each denoted and defined by $\langle x, y\rangle = y(x)$. The dual cone K^* is defined by $\{y \in E^* : \langle x, y\rangle \geq 0 \text{ for } x \in K\}$. The adjoint operator $A^*\colon F^* \to E^*$ is defined by $A^* f = f \circ A$ for all $f \in F^*$. The Lagrangian of the conic program p is naturally given by

$$L\colon K^* \times E \to \mathbb{R},\ (y, x) \mapsto c(x) - \langle x, y\rangle,$$

so that the Lagrangian dual program becomes

$$p^* = \sup\{\langle b, y\rangle : y \in F^*,\ c - A^* y \in K^*\}.$$

To reconstruct the primal from the dual we write the dual as a conic program in standard form, take the dual, and write this in standard form. The symmetry here becomes more apparent when we write both programs in a more geometric form. For e an element such that $Ae = b$ and $P = \ker(A)$, the primal and dual become

$$\inf\{\langle x, c\rangle : x \in (e + P) \cap K\} \quad \text{and} \quad \sup\{\langle y, e\rangle : y \in P^{\perp} \cap (K^* + c)\},$$

so that both programs optimize a linear functional over the intersection of a (translated) cone with an (affine) linear subspace.

When the vector spaces E and F are infinite dimensional, their algebraic duals are so large that the algebraic dual conic programs have too many variables and constraints to be useful. Instead we endow E and F with topologies and restrict E^* and F^* to contain only *continuous* linear functionals. We require these topologies to agree with the data in the sense that c and A are continuous, so that c is in E^* and the adjoint A^* maps E^* into F^*. We also require these topologies to be Hausdorff and locally convex so that there are — by the Hahn–Banach theorem — enough continuous linear functionals to separate points. This insures nondegeneracy of the bilinear forms, so that (E, E^*) and (F, F^*) are *dual pairs.* We form the dual cone and the dual conic program in the same way as before, and if we equip E and F with very strong topologies, such as the topologies of algebraically open sets, then we get the same duals as in the algebraic case. To keep the situation symmetric we equip E^* and F^* with weak* topologies; that is, we give them the weakest topologies for which all linear functionals $x \mapsto \langle x, y\rangle$ are continuous. Using nondegeneracy of the pairings we see that $(E^*)^*$ and $(F^*)^*$ are isomorphic to E and F, and by identifying them we obtain $(A^*)^* = A$, $(K^*)^* = K$, and $(p^*)^* = p$.

Suppose x is feasible for p and y is feasible for p^*. We always have $p \geq p^*$, which we call *weak duality* and which follows from $\langle x, c\rangle \geq \langle x, A^*y\rangle = \langle Ax, y\rangle = \langle b, y\rangle$. We also always have *complementary slackness*, which says $\langle x, c - A^*y\rangle = 0$ if and only if both x and y are optimal and have the same objective value. There can be a strictly positive *duality gap* $p - p^*$, and we say *strong duality* holds when this gap is 0. As for the constraint qualifications in Lagrangian duality, we have sufficient conditions for strong duality. To Slater's condition corresponds the following interior point condition: If the interior of K admits a primal feasible point and the primal problem is bounded, then $p = p^*$, and the supremum in the dual is attained.

In infinite dimensional spaces there are many interesting cones whose interior is empty, which means we cannot use an interior point condition. We have the following alternative closed cone condition: If the cone $\{(Ax, \langle x, c\rangle) : x \in K\}$ is closed in $F \times \mathbb{R}$ and there is a primal feasible solution, then $p = p^*$, and if in addition the primal is bounded, then the infimum in the primal is attained [**8**]. Choosing stronger topologies on E and F makes it easier for strong duality to hold: K will have more interior points and $F \times \mathbb{R}$ more closed sets. But the duality gap cannot always be closed by choosing a stronger topology; even finite dimensional problems such as semidefinite programs can have a strictly positive duality gap. Notice that the interior point condition benefits from E having a stronger topology, while the closed cone condition benefits from F having a stronger topology (and indirectly by E having a stronger topology to keep A continuous). The crucial ingredient in the proofs of these conditions is the Hahn–Banach separation theorem. This theorem says that if we have a closed convex set and a point, then either the point lies in the set or it can be strictly separated from it by two parallel hyperplanes in between the set and the point. This resembles the situation that given strong duality, a number λ either is an upper bound on the optimal objective of a minimization problem, or there is a dual feasible solution whose objective is in between λ and the optimal objective.

1.4. Semidefinite programming and interior point methods

The *positive semidefinite cone* $\mathcal{S}^n_{\succeq 0}$ consists of the positive semidefinite matrices of size $n \times n$. A *positive semidefinite matrix* is a real symmetric matrix whose eigenvalues are nonnegative, or equivalently, a matrix which can be written as RR^{T} where R is a real rectangular matrix. Such an R can be found efficiently by performing a Cholesky factorization, which moreover gives R in lower triangular form which is useful for solving a system of the form $RR^{\mathsf{T}}x = b$. The positive semidefinite cones are convex and for $n \geq 2$ they are not polyhedral; the extreme rays are spanned by the rank one matrices xx^{T}, where $x \in \mathbb{R}^n$. The positive semidefinite cones are self dual, where the dual pairings, denoted by $\langle \cdot, \cdot \rangle$, are defined by taking the trace of the matrix product. We view $\mathcal{S}^n_{\succeq 0}$ as a subset of the $n(n+1)/2$ dimensional vector space $\mathcal{S}_n$ of $n \times n$ real symmetric matrices. The interior of $\mathcal{S}^n_{\succeq 0}$ is the cone $S^n_{\succ 0}$ of positive definite matrices, which are real symmetric matrices with strictly positive eigenvalues. The cones $\mathcal{S}^n_{\succeq 0}$ and $\mathcal{S}^n_{\succ 0}$ induce partial orders, denoted $\succeq$ and $\succ$, on the vector space $\mathcal{S}^n$. The *Schur complement condition* says that if A, B, and C are matrices with A invertible, then $\left(\begin{smallmatrix} A & B \\ B^{\mathsf{T}} & C \end{smallmatrix}\right) \succeq 0$ if and only if $A \succ 0$ and $C - B^{\mathsf{T}} A^{-1} B \succeq 0$.

A semidefinite program is a conic program over a cone of positive semidefinite matrices. We write such a program as

$$p = \inf \Big\{ \langle X, C \rangle : X \in \mathcal{S}^n_{\succeq 0},\, \langle X, A_i \rangle = b_i \text{ for } i \in [m] \Big\},$$

where $C, A_1, \ldots, A_m \in \mathcal{S}^n$ and $b_1, \ldots, b_m \in \mathbb{R}$. The dual program is given by

$$p^* = \sup \Big\{ \langle b, y \rangle : y \in \mathbb{R}^m,\, C - \sum_{i=1}^m y_i A_i \in \mathcal{S}^n_{\succeq 0} \Big\}.$$

Checking and certifying whether a matrix is or is not positive semidefinite is easy, and this indicates that semidefinite programming is easy. Indeed, interior point methods are highly effective in solving semidefinite programs by reducing such a problem to a sequence of problems of approximating stationary points of functions from nearby points.

For this we use *Newton's method*, which is an iterative method to find roots of multivariate vector functions and stationary points of multivariate scalar functions. Given a continuously differentiable function $g\colon \mathbb{R}^n \to \mathbb{R}^n$ and a point close enough to a root r, Newton's method generates a sequence of points rapidly converging to r by applying successive Newton steps. A *Newton step* moves a point to the root of the linear approximation of g at that point. To find a stationary point of a twice continuously differentiable function $f\colon \mathbb{R}^n \to \mathbb{R}$ we apply the above method to the gradient ∇f. In this case a Newton step maps a point x to the stationary point of the second order Taylor approximation of f at x; that is, it maps x to $x - (Hf(x))^{-1} \nabla f(x)$, where Hf is the Hessian. If the domain of f is an affine space in $\mathbb{R}^n$, then we use Lagrange multipliers to optimize the Taylor approximation subject to linear equality constraints. In our applications the linear systems to be solved to determine the Newton steps will have a positive definite matrix so that we can use a Cholesky factorization. Although a Cholesky factorization can be computed efficiently, this is a relatively expensive step in interior point methods, so we typically only perform a single Newton step when we invoke Newton's method.

The function $\beta\colon S^n_{\succ 0} \to \mathbb{R}$ defined by $\beta(X) = -\log(\det(X))$ is strongly convex and grows to infinity as X nears the boundary of the cone. This is an example of a *barrier functional*, which lies at the heart of any interior point method. We use this to define the primal and dual central paths $\{X_\eta\}_{\eta\geq 0}$ and $\{(y_\eta, Z_\eta)\}_{\eta\geq 0}$, where X_η and (y_η, Z_η) are the unique optimal solutions to the *barrier problems*

$$p_\eta = \min\big\{\langle X, C\rangle + \eta\beta(X) : X \in S^n_{\succ 0},\, \langle X, A_i\rangle = b_i \text{ for } i \in [m]\big\}$$

and

$$p^*_\eta = \max\Big\{\langle b, y\rangle - \eta\beta(Z) : y \in \mathbb{R}^m,\, Z \in S^n_{\succ 0},\, Z = C - \sum_{i=1}^m y_i A_i\Big\}.$$

To guarantee the existence and uniqueness of optimal solutions we assume strict feasibility of p and p^* and linear independence of the matrices A_i. The central paths converge to optimal solutions of p and p^* as η tends to 0.

In the (short-step) *primal barrier method* we first solve an auxiliary problem to find a primal feasible solution X close to the primal central path; that is, close to X_η for some η. Then we iteratively decrease η and apply a constrained Newton step to X for the function $\langle X, C\rangle + \eta\beta(X)$ and the constraints $\langle X, A_i\rangle = b_i$ for $i \in [m]$. If we decrease η slowly enough, this results in a sequence of matrices which lie close to the central path and for which it is guaranteed that they are positive definite. As $\eta \to 0$ they converge towards the optimal solution $\lim_{\eta\downarrow 0} X_\eta$, and by choosing the right parameters this algorithm finds, for each $\varepsilon > 0$, an ε-optimal solution in polynomial time.

In *primal-dual methods* we maintain both primal and dual iterates which are allowed to violate the affine constraints. To find new iterates we use both primal and dual information, and this results in excellent performance in practice. The main observation is that the Lagrangian

$$L_\eta\colon \mathcal{S}^n \times \mathbb{R}^m \times \mathcal{S}^n_{\succ 0} \to \mathbb{R},\ (X, y, Z) \mapsto \langle b, y\rangle - \eta\beta(Z) + \langle C - \sum_{i=1}^m y_i A_i - Z, X\rangle$$

of p^*_η has (X_η, y_η, Z_η) as unique stationary point. The stationarity condition

$$0 = \nabla_Z L_\eta(X_\eta, y_\eta, Z_\eta) = -\eta Z_\eta^{-1} + X_\eta$$

can be written as $X_\eta Z_\eta = \eta I$ so that $\eta = \langle X_\eta, Z_\eta\rangle/n$. Since $\langle X_\eta, Z_\eta\rangle$ is the duality gap, this tells us how fast the primal and dual central paths converge to optimality as $\eta \downarrow 0$. Moreover, this formula allows us to compute an η value for iterates which do not lie on the central paths or are not even feasible.

The basic idea in primal-dual methods is to start with arbitrary positive definite matrices X and Z and corresponding vector y. Then we iteratively set η to half (which is a good heuristic) the corresponding η value and perform a Newton step for the function L_η. Although $\mathcal{S}^n_{\succ 0}$ is convex and X, Z, Z_η, and X_η are all positive definite, after a Newton step the iterates X and Z might not be positive definite because Newton iterates do not lie on a straight line segment. We perform a line search in the direction of the Newton step to select positive definite iterates.

In the above method we take an optimizing Newton step for L_η, which is the same as taking a root finding Newton step for ∇L_η. In practice we often use variations which are obtained by first rewriting the equation $\nabla_Z L_\eta(X, y, Z) = 0$ as for instance $ZX - \eta I = 0$. In this variation we have to symmetrize the Z matrix after each Newton step because the product XZ of two symmetric matrices is not necessarily symmetric, so we have to apply Newton's root finding method to maps whose domain and codomain is $\mathcal{S}^n \times \mathbb{R}^m \times \mathbb{R}^{n\times n}$ instead of $\mathcal{S}^n \times \mathbb{R}^m \times \mathcal{S}^n$. This reformulation of the nonlinear gradient condition is used in the CSDP solver, which uses a predictor-corrector variant of the above algorithm [**12**].

These interior point methods can be generalized to methods for *symmetric cones*, which have been classified as being products of Lorentz cones, real, complex, and quaternionic positive semidefinite cones, and one exceptional cone. Semidefinite programming is the main case in the sense that a conic program over a product of cones from these families can be transformed (in quadratic time) into a semidefinite program. A conic program over a product of positive semidefinite cones is a semidefinite program where we take direct sums of the data matrices with zero blocks at appropriate places. This also shows linear programming is a special case of semidefinite programming. A second order cone program transforms into a semidefinite program using a Schur complement. The complex plane embeds into the algebra of real antisymmetric 2×2 matrices by mapping $x+iy$ to the matrix $\left(\begin{smallmatrix} x & y \\ -y & x \end{smallmatrix}\right)$. To transform a complex semidefinite program into a semidefinite program we simply replace each entry in the data matrices by such a block. For the quaternionic case we do the same using an embedding of the quaternions in the algebra of real antisymmetric 4×4 matrices. Of course, the computational complexity of solving a resulting semidefinite program can be much higher than the original problem, and for especially linear and second order cone programming we use specialized solvers. Moreover, semidefinite programming solvers work with products of semidefinite cones; that is, they exploit the block structure in semidefinite programs.

1.5. Symmetry in semidefinite programming

Problems $p = \inf_{x\in S} f(x)$ often contain symmetry because the underlying data has symmetry or because the modeling method introduces symmetry. Exploiting this symmetry can significantly reduce the problem size and remove degeneracies which is relevant for numerical stability. We say p has symmetry if f is invariant for a nontrivial action of a group Γ on S. More specifically we say p is Γ-invariant. When S is convex, f is linear, and Γ is compact, then we can use this symmetry to derive a simpler optimization problem. For this we let μ be the normalized Haar measure of Γ and notice that for each $x \in S$ the group average $\bar{x} = \int \gamma x \, d\mu(\gamma)$ also lies in S, is invariant under the action of Γ, and satisfies $f(\bar{x}) = f(x)$. We obtain a reduced optimization problem $p^\Gamma = \inf_{x\in S^\Gamma} f(x)$, where S^Γ is the set of Γ-invariant vectors in S. Note that in the above argument convexity is essential; a general symmetric optimization problem does not admit symmetric optimal solutions.

Given a unitary representation ρ of a finite group Γ on $\mathbb{C}^n$; that is, a group homomorphism $\rho\colon \Gamma \to U(\mathbb{C}^n)$, we get an action of Γ on the space of Hermitian $n\times n$-matrices by $\gamma X = \rho(\gamma)^* X \rho(\gamma)$. This action is eigenvalue preserving, so it preserves positive semidefiniteness, and a complex semidefinite program p is invariant whenever its objective and affine space are invariant. We obtain p^Γ by restricting to the cone of Γ-invariant, complex, positive semidefinite matrices.

There are several equivalent ways to simplify p^Γ. The matrix $*$-algebra $(\mathbb{C}^{n\times n})^\Gamma$ is $*$-isomorphic to a direct sum $\oplus_{i=1}^d \mathbb{C}^{m_i\times m_i}$ [**5**], and since $*$-isomorphisms between unital $*$-algebras preserve eigenvalues, this provides a block diagonalization of p^Γ as a conic program over a product of smaller complex positive semidefinite cones. Another viewpoint, where we use the representation more explicitly, is that invariant matrices X commutate with ρ: for each $\gamma \in \Gamma$ we have $\rho(\gamma)^* X = X\rho(\gamma)$. Schur's lemma [**28**] provides a coordinate transform $T\colon \mathbb{C}^n \to \mathbb{C}^n$ such that T^*XT has identical block structure for all $X \in (\mathbb{C}^{n\times n})^\Gamma$. This is a block diagonal structure with d diagonal blocks where the ith block is again block diagonal and consists of identical blocks of size m_i. Applying this transformation and removing redundant blocks yields the same block diagonalization as above. Here d is the number of inequivalent irreducible subrepresentations of ρ and m_i is the number of equivalent copies of the ith of these representations. A third approach applies when ρ maps into the set of permutation matrices. Then we view an invariant matrix as an invariant kernel $[n] \times [n] \to \mathbb{C}$ and apply Bochner's theorem to obtain a diagonalization with the kernel's Fourier coefficients as blocks; see Chapter 2.

1.6. Moment hierarchies in polynomial optimization

When constructing a relaxation we need to find a balance between its complexity and the quality of the bounds it gives. For an in general NP-hard optimization problem of the form

$$p = \inf_{x\in S} f(x), \quad S = \{x \in \mathbb{R}^n : g(x) \geq 0 \text{ for } g \in \mathcal{G}\},$$

where $\{f\} \cup \mathcal{G}$ is a finite set of polynomials, we use moment techniques to define a hierarchy of semidefinite programs which give increasingly good bounds. The program p admits the sharp relaxation $\inf_{\mu\in\mathcal{P}(S)} \mu(f)$, where $\mathcal{P}(S)$ is the set of probability measures on S. Let $y_\alpha = \mu(x^\alpha)$ where $x^\alpha = x_1^{\alpha_1} \cdots x_n^{\alpha_n}$. The moment sequence $\{y_\alpha\}_{\alpha\in\mathbb{N}_0^n}$ satisfies $y_0 = 1$ and is of *positive type*. This means the infinite *moment matrix* $M(y)$, defined by $M(y)_{\alpha,\beta} = y_{\alpha+\beta}$, is positive semidefinite (all its finite principal submatrices are positive semidefinite). Moreover, the *localizing matrices* $M(y * g)$, where $y * g$ is the convolution $(y * g)_\alpha = \sum_\gamma y_{\alpha+\gamma} g_\gamma$, are also positive semidefinite.

We obtain a relaxation by optimizing over truncated moment sequences y which satisfy only finitely many of these constraints. Let $M_t(y)$ be the submatrix of $M(y)$ whose entries are indexed by (α, β) with $|\alpha|, |\beta| \leq t$. Let $M_t^g(y)$ be the partial matrix whose entries are indexed by (α, β) with $|\alpha|, |\beta| \leq t$ where the (α, β) entry is given by $(y * g)_{\alpha+\beta}$ if $|\alpha + \beta| \leq 2t - \deg(g)$ and remains unspecified otherwise.

By $M_t^g(y) \succeq 0$ we mean that $M_t^g(y)$ can be completed to a positive semidefinite matrix.

For $t \geq \lceil \deg(f)/2 \rceil$, we have the relaxation

$$L_t = \inf \Big\{ \sum_\alpha f_\alpha y_\alpha : y \in \mathbb{R}^{\{\alpha : |\alpha| \leq 2t\}},\, y_0 = 1,\, M_t(y) \succeq 0,\, M_t^g(y) \succeq 0 \text{ for } g \in \mathcal{G} \Big\}.$$

This is a strengthened variation on the Lasserre hierarchy [**41**]. This gives a nondecreasing sequence of lower bounds on p and under mild conditions on the set $\mathcal{G}$ these bounds converge to p.

In the case where we enforce the variables to be binary by using the constraints $x_i^2 - x_i \geq 0$ and $x_i - x_i^2 \geq 0$ for $i \in [n]$, we can simplify the hierarchy. For each feasible y the localizing matrix corresponding to a constraint $x_i^2 - x_i \geq 0$ is both positive and negative definite, and hence equal to zero. It follows that $y_\alpha = y_{\bar{\alpha}}$ for each $\alpha \in \mathbb{N}_0^n$, where $\bar{\alpha}$ is obtained from α by replacing all nonzero entries by ones. By restricting the vectors to be of this form and removing the polynomials $x_i^2 - x_i$ and $x_i - x_i^2$ from $\mathcal{G}$ we simplify the hierarchy. We may assume all polynomials to be square free and we index their entries by subsets of $[n]$ instead of 0/1 vectors. The moment matrix of a real vector y indexed by elements from $[n]_{2t} = \{S \subseteq [n] : |S| \leq 2t\}$ is now defined as $M(y)_{J,J'} = y_{J \cup J'}$ for $J, J' \in [n]_t$, and we modify the truncated/localizing matrices in the same way. The hierarchy becomes

$$L_t = \inf \Big\{ \sum_{S \in [n]_{2t}} f_S\, y_S : y \in \mathbb{R}^{[n]_{2t}},\, y_\emptyset = 1,\, M_t(y) \succeq 0,\, M_t^g(y) \succeq 0 \text{ for } g \in \mathcal{G} \Big\}.$$

In [**42**] it is shown that the relaxation is sharp already for $t = n$.

The maximum independent set problem, which asks for a largest set of pairwise nonadjacent vertices in a finite graph $G = (V, E)$, can be written as a polynomial optimization problem with a binary variable x_v for each vertex $v \in V$ and a constraint $x_u + x_v \leq 1$ for each edge $\{u, v\} \in E$. In [**43**] it is shown that for $t \geq 2$ the t-th step of the (maximization version of the) Lasserre hierarchy reduces to

$$\vartheta_t(G) = \max \Big\{ \sum_{x \in V} y_{\{x\}} : y \in \mathbb{R}^{[n]_{2t}},\, y_\emptyset = 1,\, M_t(y) \succeq 0,\, y_S = 0 \text{ for } S \text{ dependent} \Big\}.$$

Our strengtened version reduces to this hierarchy for all $t \geq 1$. This hierarchy converges to the indepence number $\alpha(G)$ in $\alpha(G)$ steps. The map $P \colon \mathbb{R}^{V_{2t}} \to \mathbb{R}^V$ defined by $P(y)_v = y_{\{v\}}$ identifies $\vartheta_t(G)$ as a lift (see Section 1.2) of the relaxation $\max\{\sum_{x \in V} x_v : x \in P(F_t)\}$ where F_t is the feasible set of $\vartheta_t(G)$. The first step is equivalent to the Lovász ϑ-number [**54**, Theorem 67.10] which a well-known relaxation in combinatorial optimization. When the edge set is invariant under a group action on the vertices, this is a good example where the symmetrization procedure from the previous section applies.