Minimax and Convex-Concave Games

Arpita Ghosh and Stephen Boyd
EE392o, Stanford University

Autumn, 2003

Here we consider *games*, which are optimization problems with more than one decision maker (or player, in the terminology used in games), often with conflicting goals. The general theory of games is quite complicated, but some classes of games are closely related to convex optimization, and as a result have a nice theory, and are computationally tractable. These results for games are closely related to so-called min-max theorems, which state that, in some special cases, you can switch around the order of minimization and maximization without changing the value. One very useful consequence is that we can solve certain robust and minimax problems very effectively.

These notes cover some of these topics; see also the material in Boyd and Vandenberghe [BV03].

1 The matrix game

A very basic game is a two-player, zero-sum, matrix game [Int02]. A matrix game is specified by a payoff matrix $P \in \mathbf{R}^{m \times n}$. Players 1 and 2 have m and n strategies respectively, and each player picks his strategy without knowledge of the strategy of his opponent.

When player 1 picks strategy k, the possible payoffs are specified by the entries in the kth row of the matrix, and if player 2 picks strategy (or column) l, then player 1 makes a payment P_{kl} to player 2. (The game is zero sum since the payoffs to the players are equal and opposite.) Player 1 would like to choose his strategy k so as to maximize his payoff $-P_{kl}$, *i.e.*, to pick row k so as to make P_{kl} as small as possible, whereas player 2 would like to pick column l so as to make his payoff P_{kl} as large as possible.

1.1 Pure Strategies

Each player would like to choose his strategy to guarantee himself the best possible payoff, *regardless of what his opponent does.* So player 1 must expect that if he picks row k, player 2 will choose l such that P_{kl} is the largest entry in row k. This means he must choose as his strategy that row k which minimizes the row maxima, *i.e.*, the k that solves the problem:

$$\text{minimize}_k \quad \max_l P_{kl}. \tag{1}$$

Strategy	1	2	3	Row maxima
1	-1	1	0	1
2	0	4	6	6
Column minima	-1	1	0	$1 = 1$

Table 1: A two-person, zero-sum game with a saddle point.

Strategy	1	2	3	Row maxima
1	6	-2	3	6
2	-4	5	4	5
Column minima	-4	-2	3	$5 \neq 3$

Table 2: A two-person, zero-sum game with no saddle point.

Similarly, player 2 chooses a strategy assuming that if he picks column l, player 1 will choose his strategy k so as to minimize P_{kl}. So player 2 will choose that value of l which maximizes the column minima, *i.e.*, the l that solves

$$\text{maximize}_l \quad \min_k P_{kl}. \tag{2}$$

The inequality

$$\max_l \ \min_k \quad P_{kl} \leq \min_k \ \max_l \quad P_{kl} \tag{3}$$

always holds, and has an obvious interpretation: it is always better to go second, *i.e.*, to make your choice knowing your opponent's choice. In general, (3) need not hold with equality; if it does, then the game is said to be strictly determined, and the number $\max_l \min_k P_{kl}$ is said to be the *value* of the game. In a strictly determined game, there is no advantage to knowing your opponent's strategy.

Table 1 shows a payoff matrix for which (1) and (2) are equal, and table 2 shows a payoff matrix for which the maximum of the row minima is not equal to the minimum of the column maxima.

1.2 Mixed Strategies

We saw above that not every matrix game has a solution in terms of pure strategies. A *mixed* strategy is a probability distribution on a player's set of choices, according to which the player makes his choice, randomly and independently of the other player's choice. Suppose $u = (u_1, \ldots, u_n)$ and $v = (v_1, \ldots, v_n)$ are the mixed strategies of player 1 and player 2; u_i is the probability that player 1 will choose strategy i, v_j is the probability that player 2 uses strategy j. Then, for payoff matrix P, the *expected payoff* from player 1 to player 2 is

$$\sum u_i v_j P_{ij} = u^T P v. \tag{4}$$

Now the optimization variables for players 1 and 2 are respectively the vectors u and v, where u and v are both probability distributions, *i.e.*, $\mathbf{1}^T u = 1$, $\mathbf{1}^T v = 1$, $u \succeq 0$, $v \succeq 0$.

Player 1 wishes to choose u to minimize $u^T P v$, while player 2 wishes to choose v to maximize $u^T P v$. Note that the set of strategies is no longer a finite set, since the strategy now is the vector u (or v, for player 2), rather than a choice of index $i \in 1, 2, \ldots, n$ ($j \in 1, 2, \ldots, m$ for player 2).

Reasoning as before, player 1 chooses u to solve the problem

$$\begin{array}{ll} \text{minimize} & \max_{i=1,\ldots,m}(P^T u)_i \\ \text{subject to} & u \succeq 0, \quad \mathbf{1}^T u = 1, \end{array} \tag{5}$$

which is equivalent to the LP

$$\begin{array}{ll} \text{minimize} & t \\ \text{subject to} & u \succeq 0, \quad \mathbf{1}^T u = 1, \\ & P^T u \preceq t\mathbf{1}. \end{array} \tag{6}$$

Denote the optimal value of this problem by $p_1^\star$. This is the smallest expected payoff player 1 can arrange to have, assuming that player 2 knows the strategy of player 1, and plays to his own maximum advantage.

Similarly, player 2 chooses v to solve the problem

$$\begin{array}{ll} \text{maximize} & \min_{i=1,\ldots,n}(Pv)_i \\ \text{subject to} & v \succeq 0, \quad \mathbf{1}^T v = 1; \end{array} \tag{7}$$

this is equivalent to the LP

$$\begin{array}{ll} \text{maximize} & \nu \\ \text{subject to} & v \succeq 0, \quad \mathbf{1}^T v = 1, \\ & Pv \succeq \nu\mathbf{1}. \end{array} \tag{8}$$

Denote the optimal value of this problem by $p_2^\star$. This is the largest expected payoff player 2 can guarantee getting, assuming that player 1 knows the strategy of player 2. We can interpret the difference, $p_1^\star - p_2^\star$ (which is nonnegative), as the advantage conferred on a player by knowing the opponent's strategy.

It can be shown ([BV03, §5.2.5]) that the LPs (6) and (8) are duals, and using strong duality (since the LPs are feasible), we have $p_1^\star = p_2^\star$. That is, any zero-sum matrix game with mixed strategies has a unique value $p = p_1^\star = p_2^\star$, which can be found as the solution to either of the linear programs (5) or (7). The optimal mixed strategy for player 1, $u_{\rm opt}$, is found from the solution of (5). Since v is the dual variable associated with the inequality $P^T u \preceq t\mathbf{1}$, the optimal strategy for player 2, $v_{\rm opt}$, can be obtained from the optimal dual variables for (5), or directly by solving (7). Note that the optimal mixed strategies need not be unique; however, we assume that a player is only interested in finding *an* optimal strategy, since all optimal strategies lead to the same expected value of the game.

1.3 Optimal server location on network

We consider a network with n nodes, specified by an undirected graph $\mathcal{G}$. A *request* originates at a node on the network. The request is served by a *server*, also located at a node on the network. The *delay* in serving the request is given by the (shortest) distance between the server and request nodes. The goal is to place the server is such a way that we minimize the delay. If the server knows, in advance, which node the request will come from, the server simply places itself at the same node; in this case the delay is zero.

Now suppose the server does not know which node the request will come from, and wants to place itself to minimize the worst-case delay. For this, the server should place itself at a node which minimizes the maximum distance to any other node on the graph. Such a node called a *center* of the graph. The center node need not be unique; it only has the property that no matter where the request originates, the delay is never greater than $d_{\min}$, the distance from the center to the node farthest from it.

Suppose now that the location of the request is drawn from a probability distribution, and the cost is measured by expected delay. (For example, requests come repeatedly, and we consider the long term average of the delay.) The goal of the server now is to choose its position so as to minimize the *expected* delay. Here again, as in the deterministic case, there are two situations. If the probability distribution of the requests v_0 is known to the server, then to minimize the expected delay, the server simply chooses a probability distribution that minimizes the expected delay, *i.e.*, the u that solves

$$\begin{array}{ll} \text{minimize} & (Pv_0)^T u \\ \text{subject to} & \mathbf{1}^T u = 1. \end{array} \tag{9}$$

One solution to this LP is $u = e_i$, where $i = \operatorname{argmin}_j \, (Pv_0)_j$. That is, when the request distribution is known, there is always a pure strategy for the server which minimizes the expected delay. (The set of all solutions is the set of all probability distributions with support on $\mathcal{S}$, where $\mathcal{S} = \{i : (Pv_0)_i = \min_j \, (Pv_0)_j\}$; each solution specifies a probability distribution according to which the server could choose its position to minimize expected delay.)

Now consider the situation where the distribution of requests is *not* known to the server. Suppose that the goal is to find a server distribution that minimizes the worst possible expected delay, over all possible request distributions. In this case the probability distribution according to which the server should locate itself is the solution of a min-max problem, which can be formulated as a game.

The server and request are the players, and the moves or choices are the n nodes on the graph. The server wants to minimize the payoff, which is the delay, and the (adversarial) request wants to maximize it. When the request chooses to originate at node i, and the server is placed at node j, the payoff is P_{ij}, where P_{ij} is the length of the shortest path between nodes i and j. The optimal (mixed) strategy for the server (and the worst possible distribution of requests) is the solution of the matrix game with payoff matrix P.

We now consider the specific example, with 11 nodes, shown in figure 1. The payoff

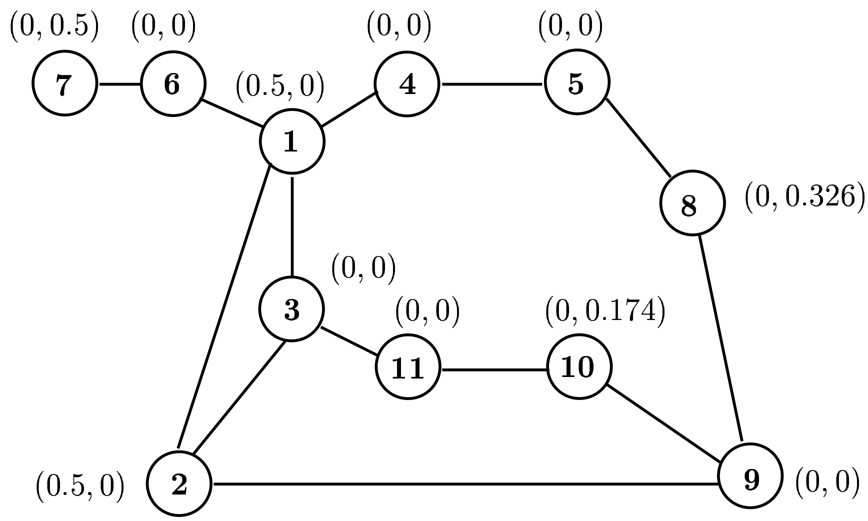

Figure 1: A network with 11 nodes. Next to each node i is the two-tuple $(u_i^\star, v_i^\star)$, where $u_i^\star$ is the probability of the server placing itself at node i, and $v_i^\star$ is the probability of the request occuring at node i, where $(u^\star, v^\star)$ is a solution of the game.

matrix P for this graph is

$$P = \begin{bmatrix} 0 & 1 & 1 & 1 & 2 & 1 & 2 & 3 & 2 & 3 & 2 \\ 1 & 0 & 1 & 2 & 3 & 2 & 3 & 2 & 1 & 2 & 2 \\ 1 & 1 & 0 & 1 & 2 & 2 & 3 & 3 & 2 & 2 & 1 \\ 1 & 2 & 1 & 0 & 1 & 2 & 3 & 2 & 3 & 3 & 2 \\ 2 & 3 & 2 & 1 & 0 & 3 & 4 & 1 & 2 & 3 & 3 \\ 1 & 2 & 2 & 2 & 3 & 0 & 1 & 4 & 3 & 4 & 3 \\ 2 & 3 & 3 & 3 & 4 & 1 & 0 & 5 & 4 & 5 & 4 \\ 3 & 2 & 3 & 2 & 1 & 4 & 5 & 0 & 1 & 2 & 3 \\ 2 & 1 & 2 & 3 & 2 & 3 & 4 & 1 & 0 & 1 & 2 \\ 3 & 2 & 2 & 3 & 3 & 4 & 5 & 2 & 1 & 0 & 1 \\ 2 & 2 & 1 & 2 & 3 & 3 & 4 & 3 & 2 & 1 & 0 \end{bmatrix}. \qquad (10)$$

Nodes 1, 2, 3 and 4 are all centers of the graph, and the distance from a center to the node farthest from it is 3. If the request is unknown and deterministic, the server can place itself at any of these centers, and be assured of a delay no greater than 3, no matter where the request node is located.

Suppose the request is not static, but instead is known to be equally likely to come from any node. Then to minimize the expected delay, the server should place itself at node 1 or 3 (or any combination of nodes 1 and 3 with probabilities that add to one). The corresponding minimum expected delay is 1.636.

When the request distribution is not known to the server, the optimal mixed strategy is found as the solution to the LP (6), with P as specified. An optimal probability distribution for the server is

$$u^\star = (0.486, 0.077, 0, 0, 0, 0.104, 0.035, 0.298, 0, 0, 0). \qquad (11)$$

Note that the optimal u need not be unique. In this case, for example, the strategy $u' = (0.5, 0.5, 0, 0, 0, 0, 0, 0, 0, 0, 0)$ also solves the LP (6), and is optimal for the server. Similarly, a worst possible probability distribution for the request nodes is

$$v^\star = (0, 0, 0, 0, 0, 0, 0.500, 0, 0.326, 0, 0.174). \tag{12}$$

The value of the game, which is the smallest worst-case expected delay, is 2.5. That is, with this service distribution $u^\star$, for *any* possible request distribution, the expected delay will be at most 2.5.

We noted before that the vectors u and v need not be unique. The set of all optimal u and v is the set of all primal and dual optimal solutions to the LP (6). That is, every primal (dual) solution to (6) is an optimal strategy for the server (request), and conversely any server (request) mixed strategy that solves the game must also be primal (dual) optimal for (6). Observe that not every server strategy which optimizes the delay for the optimal request strategy solves the game, *i.e.*, not every solution to

$$\begin{array}{ll} \text{minimize} & (Pv^\star)^T u \\ \text{subject to} & \mathbf{1}^T u = 1 \end{array} \tag{13}$$

is a solution of the game. This is because although such a solution minimizes the expected delay when the request distribution is $v^\star$, it does not guarantee that the expected delay with *any* request distribution is less than or equal to the value of the game. This happens because the (bilinear) objective function is not strictly convex-concave in u and v, so that for a fixed value of u, the optimizing v need not be unique (and vice versa).

2 Bilinear problems

Observe that we can now solve min-max problems with bilinear objective and separable polyhedral constraints on the variables using the same ideas as above; here, as before, the minimax is equal to the maximin. Consider the problem

$$\begin{array}{lll} \text{minimize}_x & \text{maximize}_y & x^T P y \\ \text{subject to} & & Ax \preceq b, \\ & & Cy \preceq d; \end{array} \tag{14}$$

Let us assume that the feasible set is non-empty and bounded. We would like to solve this problem, as well as show that its value is the same as the maximin problem

$$\begin{array}{lll} \text{maximize}_y & \text{minimize}_x & x^T P y \\ \text{subject to} & & Ax \preceq b, \\ & & Cy \preceq d; \end{array} \tag{15}$$

i.e., the order of optimization does not matter.

We will solve (14) by transforming it to an LP using duality. Fix x, then

$$\begin{array}{ll} \text{maximize}_y & (xP^T)^T y \\ \text{subject to} & Cy \preceq d \end{array} \tag{16}$$

is an LP, the optimal value of which is a function of x. The dual of this LP is

$$\begin{array}{ll} \text{minimize}_\lambda & d^T\lambda \\ \text{subject to} & C^T\lambda = P^T x; \\ & \lambda \succeq 0. \end{array} \tag{17}$$

Since we assumed the feasible set of (14) is non-empty and bounded, strong duality holds for (16) and its dual (17), and their optimal values are finite and equal. So we can rewrite (14) as

$$\begin{array}{ll} \text{minimize}_{x,\lambda} & d^T\lambda \\ \text{subject to} & C^T\lambda = P^T x, \\ & Ax \preceq b, \\ & \lambda \succeq 0; \end{array} \tag{18}$$

which is an LP, and can be efficiently solved.

To show that (14) and (15) have the same value, we will rewrite (15) using duality as we did earlier, and demonstrate that the resulting LP is the dual of (18). The dual of

$$\begin{array}{ll} \text{minimize}_x & (yP)^T x \\ \text{subject to} & Ax \preceq b \end{array} \tag{19}$$

is the following problem

$$\begin{array}{ll} \text{maximize}_\nu & -b^T\nu \\ \text{subject to} & A^T\nu + Py = 0; \\ & \nu \succeq 0, \end{array} \tag{20}$$

and using strong duality as before, we can rewrite (15) as

$$\begin{array}{ll} \text{maximize}_{y,\nu} & -b^T\nu \\ \text{subject to} & A^T\nu + Py = 0; \\ & Cy \preceq d, \\ & \nu \succeq 0. \end{array} \tag{21}$$

It is easy to show that (18) and (21) are duals of each other. Start with (18), and write the Lagrangian

$$\begin{array}{rcl} \mathcal{L}(x, \lambda, \mu_1, \mu_2, \mu_3) & = & d^T\lambda + \mu_1^T(Ax - b) + \mu^T\lambda + \mu_3(P^T x - C^T\lambda) \\ & = & (d^T + \mu_2^T - \mu_3 C^T)\lambda + (\mu_1^T A + \mu_3^T P^T)x - \mu_1^T b \end{array} \tag{22}$$

So

$$\inf_{x,\lambda} \mathcal{L} = \begin{cases} -\mu_1^T b; & -C^T\mu_3 + \mu + 2 + d = 0, A^T\mu_1 - P\mu_3 = 0 \\ -\infty; & \text{otherwise} \end{cases} \tag{23}$$

and the dual problem is

$$\begin{array}{ll} \text{maximize} & -b^T\mu_1 \\ \text{subject to} & -C\mu_3 + \mu_2 + d = 0 \\ & A^T\mu_1 + P\mu_3 = 0 \\ & \mu_1 \succeq 0 \\ & \mu_2 \succeq 0 \end{array} \tag{24}$$

which can be rewritten, eliminating μ_2, as

$$\begin{array}{ll} \text{maximize} & -b^T \mu_1 \\ \text{subject to} & C\mu_3 \preceq d \\ & A^T \mu_1 + P\mu_3 = 0 \\ & \mu_1 \succeq 0, \end{array} \tag{25}$$

which is identical to (21). So the two problems (18) and (21) are duals. Because of our assumption that the polyhedra $Ax \preceq b$ and $Cy \preceq d$ are feasible and bounded, the problems (18) and (21) are also feasible, and so strong duality obtains. We can solve (18) to obtain x and the optimal value, and y from the dual variable corresponding to the equality constraint $C^T\lambda = P^T x$, or obtain y directly from the solution of the LP (21).

2.1 Robust LP

To solve bilinear progams with separable polyhedral constraints, we use strong duality to transform a minimax problem into a minimin problem, which is a linear program. A similar idea can be used to rewrite a robust LP, where two variables are coupled through a bilinear constraint, as a single linear program. Specifically, consider the robust LP

$$\begin{array}{ll} \text{minimize} & c^T x \\ \text{subject to} & \sup_{u_i \in \mathcal{U}_i} (\bar{a}_i + B_i u_i)^T x + b_i \leq 0, \quad i = 1, \ldots, m, \end{array} \tag{26}$$

where $x \in \mathbf{R}^n$ is the optimization variable, and $u_i \in \mathbf{R}^{n_i}$ is the *uncertainty* vector. The problem data are $c \in \mathbf{R}^n$, $\bar{a}_i \in \mathbf{R}^n$, $B_i \in \mathbf{R}^{n \times n_i}$ and $b \in \mathbf{R}^m$. The uncertainty polyhedrons are defined as

$$\mathcal{U}_i \triangleq \{u \in \mathbf{R}^{n_i} \mid D_i u \preceq d_i\}, \quad i = 1, \ldots, m,$$

where $D_i \in \mathbf{R}^{m_i \times n_i}$ and $d_i \in \mathbf{R}^{m_i}$. We will assume that the $\mathcal{U}_i$ are non-empty and bounded. In *principle* we can solve the robust LP (26) by solving the linear program

$$\begin{array}{ll} \text{minimize} & c^T x \\ \text{subject to} & (\bar{a}_i + B_i w_i)^T x + b_i \leq 0, \quad \forall\, w_i \in V(\mathcal{U}_i), \quad i = 1, \ldots, m, \end{array} \tag{27}$$

where $V(\mathcal{U}_i)$ denotes the set of vertices of the polyhedron $\mathcal{U}_i$. But the cardinality of $V(\mathcal{U}_i)$ grows exponentially with m_i, and so (27) is computationally tractable only when m_i, $i = 1, \ldots, m$ are very small.

Using strong duality, we will transform the robust linear program (26) into a linear program whose number of variables grows linearly with m_i, $i = 1, \ldots, m$. Let us first express the robust LP problem (26) as

$$\begin{array}{ll} \text{minimize} & c^T x \\ \text{subject to} & f_i(x) \leq 0, \quad i = 1, \ldots, m, \end{array} \tag{28}$$

where $f_i(x)$ is the optimal value of the LP

$$\begin{array}{ll} \text{maximize} & (x^T B_i) w_i + (\bar{a}_i^T x + b_i) \\ \text{subject to} & D_i w_i \preceq d_i. \end{array} \tag{29}$$

It is easy to show that the Lagrange dual of the linear program (29) is given by

$$\begin{array}{ll} \text{minimize} & d_i^T z_i + (\bar{a}_i^T x + b_i) \\ \text{subject to} & D_i^T z_i = B_i^T x, \quad z_i \geq 0. \end{array} \tag{30}$$

Since $\mathcal{U}_i$ is bounded and non-empty, (29) is feasible (with a finite optimal value), and by strong duality, (29) and (30) have the same optimum value. So $f_i(x) \leq 0$ if and only if there exists a z_i such that $\bar{a}_i^T x + d_i^T z_i + b_i \leq 0$, $D_i^T z_i = B_i^T x$, and $z_i \geq 0$. Thus, the robust linear program (26) is equivalent to the following linear program

$$\begin{array}{ll} \text{minimize} & c^T x \\ \text{subject to} & D_i^T z_i = B_i^T x, \quad i = 1, \ldots, m, \\ & \bar{a}_i^T x + d_i^T z_i + b_i \leq 0, \quad i = 1, \ldots, m, \\ & z_i \geq 0, \quad i = 1, \ldots, m. \end{array} \tag{31}$$

3 Convex-concave games

An unconstrained (zero-sum, two-player) game on $\mathbf{R}^p \times \mathbf{R}^q$ is defined by its *payoff function* $f : \mathbf{R}^{p+q} \to \mathbf{R}$. As before, the meaning is that player 1 chooses a value (or move) $u \in \mathbf{R}^p$, and player 2 chooses a value (or move) $v \in \mathbf{R}^q$; based on these choices, player 1 makes a payment to player 2, in the amount $f(u, v)$. The goal of player 1 is to minimize this payment, while the goal of player 2 is to maximize it.

We say that $(u^\star, v^\star)$ is a *solution* of the game, or a *saddle-point* for the game, if for all u, v,

$$f(u^\star, v) \leq f(u^\star, v^\star) \leq f(u, v^\star).$$

At a saddle point, neither player can do better by unilaterally changing his strategy; for $u = u^\star$, $v^\star$ maximizes $f(u^\star, v)$ (which is a function only of v), and for $v = v^\star$, $u^\star$ is the (not necessarily unique) minimizer of $f(u, v^\star)$. (Note that a saddle point here is a pure strategy, as opposed to the mixed strategy saddle points in matrix games).

The game is called *convex-concave* if for each v, $f(u, v)$ is a convex function of u, and for each u, $f(u, v)$ is a concave function of v. When f is differentiable (and convex-concave), a saddle-point for the game is characterized by $\nabla f(u^\star, v^\star) = 0$. This is easy to see: since f is a convex function of u, the optimality condition for $u^\star$ to be a minimum for $f(u, v^\star)$ is that $\nabla_u f(u^\star, v^\star) = 0$; similarly the condition for $v^\star$ to be a maximizing point for the (concave) function $f(u^\star, v)$ is that $\nabla_v f(u^\star, v^\star) = 0$.

For a game with a twice-differentiable payoff function, the solution of the game can be computed using an infeasible start Newton method; see [BV03, §10.3.4] for details.

3.1 Equality constraints

Suppose that we add linear equality constraints for the variables u and v. If the equality constraints are separable in the variables u and v, the conditions for $(u^\star, v^\star)$ to be a saddle point are again not hard to derive. Specifically, suppose we want to find a saddle point for the game with (convex-concave) payoff function $f(u, v)$, subject to the constraints $Au = b$, $Cu = d$. Then, the KKT conditions for the point $u^\star$ to solve

$$\begin{array}{ll} \text{minimize}_u & f(u, v^\star) \\ \text{subject to} & Au = b, \end{array} \tag{32}$$

are

$$\begin{array}{c} \nabla_u f(u^\star, v^\star) + A^T \nu_1 = 0, \\ Au^\star = b, \end{array} \tag{33}$$

where ν_1 is the dual variable associated with the equality constraint $Au = b$. Similarly, the condition for $v^\star$ to solve

$$\begin{array}{ll} \text{maximize}_v & f(u^\star, v) \\ \text{subject to} & Cv = d, \end{array} \tag{34}$$

are

$$\begin{array}{c} \nabla_v f(u^\star, v^\star) + C^T \nu_2 = 0, \\ Cv^\star = d. \end{array} \tag{35}$$

To find the saddle point, we need to simultaneously solve the system of equations (33)-(35), for which we use the infeasible start Newton method. Here, the equation defining the Newton step (derived using the first order approximation) is

$$\begin{bmatrix} \nabla^2_{uu} f & 0 & A^T & 0 \\ 0 & \nabla^2_{vv} f & 0 & C^T \\ A & 0 & 0 & 0 \\ 0 & C & 0 & 0 \end{bmatrix} \begin{bmatrix} \Delta u \\ \Delta v \\ \Delta \nu_1 \\ \Delta \nu_2 \end{bmatrix} = \begin{bmatrix} -\nabla f_u - A^T \nu_1 \\ -\nabla f_v - C^T \nu_2 \\ b - Au \\ d - Cv \end{bmatrix}. \tag{36}$$

All derivatives are evaluated at the current iterate, and the vector $(\Delta u, \Delta v, \Delta \nu_1, \Delta \nu_2)$ is the Newton step.

3.2 Inequality constraints

Suppose that we now add inequality constraints for the variables u and v to the original unconstrained problem. In this case, we can use a barrier method to solve for the saddle point. For simplicity, let us assume that we have no equality constraints. Specifically, consider the convex-concave game with inequality constraints,

$$\begin{array}{lll} \text{minimize}_u \ \text{maximize}_v & f_0(u, v) & \\ \text{subject to} & f_i(u) \le 0, & i = 1, \dots, m \\ & \tilde{f}_i(v) \le 0, & i = 1, \dots, \tilde{m}. \end{array} \tag{37}$$

As before, we will assume that the constraint functions f_i and $\tilde{f}_i$ are convex and differentiable, and the objective function f_0 is differentiable and convex-concave, Also, for simplicity we assume that $\textbf{dom}\, f_0 = \mathbf{R}^n \times \mathbf{R}^{\tilde{n}}$.

3.2.1 Solution using barrier method

As for convex optimization problems, a barrier method can be used to solve a convex-concave game with inequality constraints.

Let $t > 0$, and

$$f_t(u,v) = f_0(u,v) - \sum_{i=1}^{m} \log(-f_i(u)) + \sum_{i=1}^{\tilde{m}} \log(-\tilde{f}_i(v)).$$

The function $\frac{1}{t} f_t$ approximates the objective function in (37) as $t \to \infty$; the barrier term is an approximation to the indicator function corresponding to the inequality constraints. To solve the convex-concave game (37), we solve a sequence of unconstrained games that approximate the original problem more and more closely as $t \to \infty$.

We can see that $f_t(u,v)$ is convex-concave in (u,v), from the convex-concave property of f_0; convexity of $-\log(-f_i)$, and concavity of $\log(-\tilde{f}_i)$. We will assume that it has a unique saddle-point, $(u^\star(t), v^\star(t))$, which can be found using the infeasible-start Newton method.

As in the barrier method for solving a convex optimization problem, we can derive a simple bound on the suboptimality of $(u^\star(t), v^\star(t))$, which depends only on the problem dimensions, and decreases to zero as t increases. Let U and V denote the feasible sets for u and v,

$$U = \{u \mid f_i(u) \leq 0,\ i = 1, \ldots, m\}, \qquad V = \{v \mid \tilde{f}_i(v) \leq 0,\ i = 1, \ldots, \tilde{m}\}.$$

We will show that

$$\begin{aligned} f_0(u^\star(t), v^\star(t)) &\leq \inf_{u \in U} f_0(u, v^\star(t)) + \frac{m}{t}, \\ f_0(u^\star(t), v^\star(t)) &\geq \sup_{v \in V} f_0(u^\star(t), v) - \frac{\tilde{m}}{t}, \end{aligned}$$

and therefore

$$\inf_{u \in U} f_0(u, v^\star(t)) - \sup_{v \in V} f_0(u^\star(t), v) \leq \frac{m + \tilde{m}}{t}.$$

Since $(u^\star(t), v^\star(t))$ is a saddle-point of the function

$$t f_0(u,v) - \sum_{i=1}^{m} \log(-f_i(u)) + \sum_{i=1}^{\tilde{m}} \log(-\tilde{f}_i(v)),$$

its gradient with respect to u, and also with respect to v, vanishes there:

$$\begin{aligned} t \nabla_u f_0(u^\star(t), v^\star(t)) + \sum_{i=1}^{m} \frac{1}{-f_i(u^\star(t))} \nabla f_i(u^\star(t)) &= 0 \\ t \nabla_v f_0(u^\star(t), v^\star(t)) + \sum_{i=1}^{\tilde{m}} \frac{-1}{-\tilde{f}_i(v^\star(t))} \nabla \tilde{f}_i(v^\star(t)) &= 0. \end{aligned}$$

It follows that $u^\star(t)$ minimizes

$$f_0(u, v^\star(t)) + \sum_{i=1}^{m} \lambda_i f_i(u)$$

over u, where $\lambda_i = 1/(-tf_i(u^\star(t)))$, *i.e.*, for all u, we have

$$f_0(u^\star(t), v^\star(t)) + \sum_{i=1}^m \lambda_i f_i(u^\star(t)) \leq f_0(u, v^\star(t)) + \sum_{i=1}^m \lambda_i f_i(u).$$

The lefthand side is equal to $f_0(u^\star(t), v^\star(t)) - m/t$, and for all $u \in U$, the second term on the righthand side is nonpositive, so we have

$$f_0(u^\star(t), v^\star(t)) \leq \inf_{u \in U} f_0(u, v^\star(t)) + m/t.$$

A similar argument shows that

$$f_0(u^\star(t), v^\star(t)) \geq \sup_{v \in V} f_0(u^\star(t), v) - m/t.$$

Combining the bounds from (3.2.1) and (3.2.1), we obtain the bound on the suboptimality of $(u^\star(t), v^\star(t))$:

$$\inf_{u \in U} f_0(u, v^\star(t)) - \sup_{v \in V} f_0(u^\star(t), v) \leq \frac{m + \tilde{m}}{t}. \tag{38}$$

This bound on the suboptimality of $(u^\star(t), v^\star(t))$ can be used in exit condition for the barrier method. Equality constraints on u and v can be easily handled by combining the results of (38) and (33)-(35).

3.3 An example of a convex-concave game

This example is from a simple communications problem. We consider m Gaussian communications channels, with signal power $p_i \geq 0$ and noise (or interference) power $n_i \geq 0$. The capacity of channel i is proportional to $\log(1 + \beta_i p_i/(\sigma_i + n_i))$, where β_i is positive constant, and $\sigma_i > 0$ is the receiver noise. Our objective is the total capacity,

$$f(p, n) = \sum_{i=1}^m \log\left(1 + \frac{\beta_i p_i}{\sigma_i + n_i}\right). \tag{39}$$

It can be verified that f is concave in p and convex in n.

The goal is to allocate a given total power P across the channels in order to maximize the sum rate f. If the noise powers n_i are known, then the optimal allocation of powers to maximize the capacity can be formulated as convex optimization problem, maximizing f with respect to p, with n fixed. This problem can be solved by standard methods, or by a special method called *waterfilling* ([CT91, §10.4], [BV03, §5.5.3]).

Now suppose that the noise powers are not known, but we do know that the total noise power is N. The allocation of the noise powers can be thought of as made by an adversary, whose aim is to reduce the total capacity of the systems. Another interpretation is that we want to allocate signal powers so that, even with the worst possible allocation of noise powers, we obtain the largest sum rate.

In this case, the capacity of the channel is a function of the signal powers p_i and the noise powers n_i, where both p and n satisfy a power constraint: $\sum_i p_i = P$, $\sum_i n_i = N$. The user would like to allocate the p_i so as to maximize the channel capacity irrespective of the noise, and the adversary would like to allocate the n_i so as to minimize the capacity. The optimal allocation of powers to the m channels can be found by solving a game with the noise and signal powers as players. The objective function f is smooth, convex in n for every p and concave in p for every n. Specifically, we would like to solve the following game

$$\begin{array}{lll} \text{maximize}_p & \text{minimize}_n & \sum_{i=1}^m \log(1+\frac{\beta_i p_i}{\sigma_i+n_i}) \\ \text{subject to} & & \mathbf{1}^T p = P, \\ & & \mathbf{1}^T n = N, \\ & & p \succeq 0, n \succeq 0. \end{array} \tag{40}$$

At a saddle point of the game, the value of $p^\star$ must maximize the capacity for the noise distribution $n^\star$; similarly $n^\star$ must be the minimizer of $f(p^\star, n)$. It is well known from information theory that the optimal $p^\star$ satisfies the following condition:

$$p_i^\star = (\nu - N_i)^+, \tag{41}$$

where ν is chosen to satisfy

$$\sum_i (\nu - N_i)^+ = P, \tag{42}$$

and N_i is the effective noise power on each channel, $N_i = \frac{\sigma_i+n_i^\star}{\beta_i}$. This solution is called waterfilling, since the way the power distributes itself among the various channels is identical to the way water distributes itself in a vessel with uneven base. We can easily compute the optimal value of ν since the lefthand side is increasing in ν, so we can, *e.g.*, use bisection.

In a similar way we can derive a semi-analytical expression for $n^\star$, as the minimizer of $f(p^\star, n)$, since the objective is separable. To do this, we rewrite the function f to *implicity* include the inequality constraint $n \succeq 0$:

$$f(p^\star, n) = \begin{cases} \sum_{i=1}^m \log(1+\frac{\beta_i p_i^\star}{\sigma_i+n_i}) & n \succeq 0 \\ +\infty & n \not\succeq 0 \end{cases} \tag{43}$$

Now we have the following optimization problem

$$\begin{array}{ll} \text{minimize}_p & f(p^\star, n) \\ \text{subject to} & \mathbf{1}^T n = N, \end{array} \tag{44}$$

which has only one (equality) constraint. Associating the dual variable μ with this constraint, we can write the Lagrangian

$$\begin{aligned} L(n,\mu) &= f(p^\star, n) + \mu(N - \mathbf{1}^T n) \\ &= \textstyle\sum_{i=1}^m (f_i(n_i) - \mu n_i) + \mu N, \end{aligned}$$

where

$$f_i(n_i) = \begin{cases} \log(1+\frac{\beta_i p_i^\star}{\sigma_i+n_i}) & n_i \geq 0 \\ +\infty & n_i < 0 \end{cases} \tag{45}$$

So

$$g(\mu) = \sum_{i=1}^m \min_{n_i}(f_i(n_i) - \mu n_i) + \mu N. \tag{46}$$

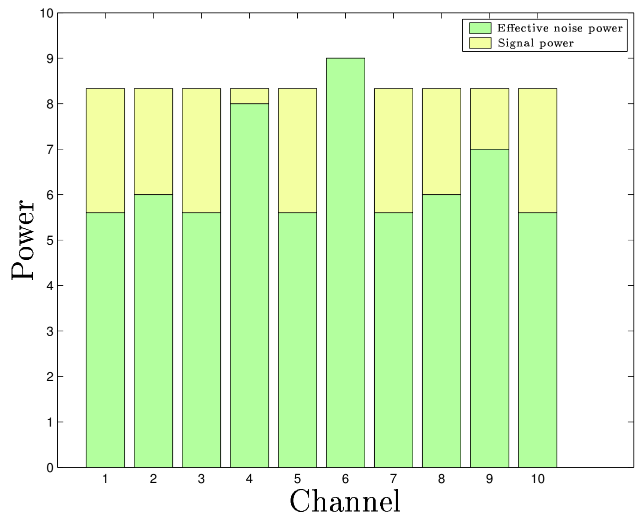

Figure 2: A game with 10 channels. The height of patch i is the effective noise in channel i at the saddle point, $(\sigma_i + n_i^\star)/\beta_i$. The height of the water above each patch is the optimal value of $p_i^\star$.

Since g is separable in n_i, we can separately minimize each term. The minimizing n_i is given by

$$n_i = \max\{(-\frac{\beta_i p_i}{2} + \frac{1}{2}\sqrt{(\beta_i p_i)^2 - 4\frac{\beta_i p_i}{\mu}} - \sigma_i), 0\}. \tag{47}$$

Substituting these equations for n_i back into the constraint $\sum n_i = N$, we see that μ is the (unique) real number that solves

$$\sum_{i=1}^{m} \max\{(-\frac{\beta_i p_i}{2} + \frac{1}{2}\sqrt{(\beta_i p_i)^2 - 4\frac{\beta_i p_i}{\mu}} - \sigma_i), 0\} = N. \tag{48}$$

(Again, the lefthand side is increasing in μ, and can be found by bisection.)

Since the solution $(p^\star, n^\star)$ of the game (40) is a saddle point of the game, $p^\star$ must be the waterfilling solution for the effective noise corresponding to $n^\star$, and $n^\star$ must be the solution to the minimization described above.

We use the barrier method to solve a specific instance of the game (40) with 10 channels. The problem parameters are chosen as $P = 20$, $N = 10$, $\sigma = (2, 6, 5, 8, 3, 9, 5, 6, 7, 3)$ and $\beta_i = 1$, $i = 1, \ldots, m$. The optimal allocation of signal powers is

$$p^\star = (2.734, 2.333, 2.733, 0.334, 2.733, 0.000, 2.733, 2.333, 1.333, 2.733),$$

and the worst noise is

$$n^\star = (3.6, 0, 0.6, 0, 2.6, 0, 0.6, 0, 0, 2.6).$$

The value of the game, *i.e.*, the capacity of the channels evaluated at $(p^\star, n^\star)$, is $C^\star = 2.860$. Figure 2 illustrates the distribution of the optimal $p^\star$ for this game, together with the effective noise $(\sigma_i + n_i^\star)/\beta_i$ for the optimal $n^\star$.

If the noise powers are distributed uniformly across the channels, then the capacity for the allocation $p^\star$ of signal powers is 3.206. As expected, this capacity is greater than $C^\star$ – since $(p^\star, n^\star)$ is a saddle point, *any* distribution of noise powers n will lead to higher capacity than $n^\star$. The best capacity with the uniform distribution of noise powers is 3.335, which, again as expected, is greater than both $C^\star$ and 3.206.

If the signal powers are equally distributed across all the channels, *i.e.*, $p_i = P/m = 2$, then the channel capacity with the noise allocation $n^\star$ is 2.777. This is smaller than $C^\star$ as it should be, since $C^\star$ is the maximum capacity that can be achieved if the noise distribution is $n^\star$. With the worst possible noise distribution for uniformly distributed signal powers, the capacity is only a little less than 2.777; in general this capacity will be less than both $C^\star$ and the capacity obtained with $n^\star$ and the uniform signal power distribution.

Another way one might expect to solve (40) is by *alternate waterfilling* ([Yu02, §3.3.3]). Since $p^\star$ maximizes $f(p, n^\star)$ and $n^\star$ minimizes $f(p^\star, n)$, we iteratively compute the optimal signal powers for a given noise distribution, then compute the minimizing noise powers from this power distribution. That is, starting with an initial value n_0, for this n_0 (given σ and β), perform waterfilling to find the maximizing p_0. Then, for this p_0, find the vector n which maximizes $f(p_0, n)$, and so on, till the optimal values of the two problems are equal within a specified tolerance ϵ. Here, alternate waterfilling converges, and yields the pair $(p^\star, n^\star)$ specified above, which is the solution of the game.

References

[BV03] S. Boyd and L. Vandenberghe. *Convex Optimization.* Cambridge University Press, 2003.

[CT91] T. Cover and J. Thomas. *Elements of Information Theory.* Wiley Series in Telecommunications. John Wiley and Sons, Inc., 1991.

[Int02] M. Intrilligator. *Mathematical Optimization and Economic Theory.* Classics in Applied Mathematics. SIAM, 2002.

[Yu02] W. Yu. *Competition and Cooperation in Multiuser Communication Environments.* PhD thesis, Stanford University, 2002.