2.2 Linear programs

A **linear program** is an optimization problem in which the objective and all constraints are linear. It has the form

$$\begin{array}{lll} \text{minimize} & c^T x & \\ \text{subject to} & a_i^T x \geq b_i, & i \in M_1 \\ & a_i^T x \leq b_i, & i \in M_2 \\ & a_i^T x = b_i, & i \in M_3 \\ & x_j \geq 0, & j \in N_1 \\ & x_j \leq 0, & j \in N_2 \end{array}$$

where $c \in \mathbb{R}^n$ is a cost vector, $x \in \mathbb{R}^n$ is a vector of decision variables, and constraints are given by $a_i \in \mathbb{R}^n$ and $b_i \in \mathbb{R}$ for $i \in \{1, \ldots, m\}$. Index sets $M_1, M_2, M_3 \subseteq \{1, \ldots, m\}$ and $N_1, N_2 \subseteq \{1, \ldots, n\}$ are used to distinguish between different types of constraints.

An equality constraint $a_i^T x = b_i$ is equivalent to the pair of constraints $a_i^T \leq b_i$ and $a_i^T x \geq b_i$, and a constraint of the form $a_i^T x \leq b_i$ can be rewritten as $(-a_i)^T x \geq -b_i$. Each occurrence of an unconstrained variable x_j can be replaced by $x_j^+ + x_j^-$, where x_j^+ and x_j^- are two new variables with $x_j^+ \geq 0$ and $x_j^- \leq 0$. We can thus write every linear program in the *general form*

$$\min \{c^T x : Ax \geq b, x \geq 0\} \tag{2.1}$$

where $x, c \in \mathbb{R}^n$, $b \in \mathbb{R}^m$, and $A \in \mathbb{R}^{m \times n}$. Observe that constraints of the form $x_j \geq 0$ and $x_j \leq 0$ are just special cases of constraints of the form $a_i^T x \geq b_i$, but we often choose to make them explicit.

A linear program of the form

$$\min \{c^T x : Ax = b, x \geq 0\} \tag{2.2}$$

is said to be in *standard form*. The standard form is of course a special case of the general form. On the other hand, we can also bring every general form problem into the standard form by replacing each inequality constraint of the form $a_i^T x \leq b_i$ or $a_i^T x \geq b_i$ by a constraint $a_i^T x + s_i = b_i$ or $a_i^T x - s_i = b_i$, where s_i is a new so-called **slack variable**, and an additional constraint $s_i \geq 0$.

The general form is typically used to discuss the theory of linear programming, while the standard form is more convenient when designing algorithms for linear programming.

Example 2.3. Consider the following linear program, as illustrated in Figure 2:

$$\begin{array}{ll} \text{minimize} & -(x_1 + x_2) \\ \text{subject to} & x_1 + 2x_2 \leq 6 \\ & x_1 - x_2 \;\;\leq 3 \\ & x_1, x_2 \quad\; \geq 0 \end{array}$$

2.4 Complementary slackness

An important relationship between primal and dual solutions is provided by conditions known as **complementary slackness**. Complementary slackness requires that slack does not occur simultaneously in a variable, of the primal or dual, and the corresponding constraint, of the dual or primal. Here, a variable is said to have slack if its value is non-zero, and an inequality constraint is said to have slack if it does not hold with equality. It is not hard to see that complementary slackness is a necessary condition for optimality. Indeed, if complementary slackness was violated by some variable and the corresponding constraint, reducing the value of the variable would reduce the value of the Lagrangian, contradicting optimality of the current solution. Recall that the variables of the dual correspond to the Lagrange multipliers. The following result formalizes this intuition.

Theorem 2.4. *Let x and λ be feasible solutions for the primal (2.1) and the dual (2.3), respectively. Then x and λ are optimal if and only if they satisfy complementary slackness, i.e. if*

$$(c^T - \lambda^T A)x = 0 \quad \text{and} \quad \lambda^T(Ax - b) = 0. \tag{2.4}$$

Proof. Since x and λ are feasible, (2.4) holds if and only if $(c^T - \lambda^T A)x + \lambda^T(Ax - b) = 0$. But this is equivalent to $c^T x = \lambda^T b$, which holds if and only if x and λ are optimal. □

2.5 Shadow prices

A more intuitive understanding of Lagrange multipliers can be obtained by again viewing (1.1) as a family of problems parameterized by $b \in \mathbb{R}^m$. As before, let $\phi(b) = \inf\{f(x) : h(x) \leq b, x \in \mathbb{R}^n\}$. It turns out that at the optimum, the Lagrange multipliers equal the partial derivatives of ϕ.

Theorem 2.5. *Suppose that f and h are continuously differentiable on $\mathbb{R}^n$, and that there exist unique functions $x^* : \mathbb{R}^m \to \mathbb{R}^n$ and $\lambda^* : \mathbb{R}^m \to \mathbb{R}^m$ such that for each $b \in \mathbb{R}^m$, $h(x^*(b)) = b$ and $f(x^*(b)) = \phi(b) = \inf\{f(x) - \lambda^*(b)^T(h(x) - b) : x \in \mathbb{R}^n\}$. If x^* and λ^* are continuously differentiable, then*

$$\frac{\partial \phi}{\partial b_i}(b) = \lambda_i^*(b).$$

Proof. We have that

$$\begin{aligned}\phi(b) &= f(x^*(b)) - \lambda^*(b)^T(h(x^*(b)) - b) \\ &= f(x^*(b)) - \lambda^*(b)^T h(x^*(b)) + \lambda^*(b)^T b.\end{aligned}$$

3.3 The simplex tableau

We can understand the simplex method in terms of the so-called **simplex tableau**, which stores all the information required to explore the set of basic solutions.

Let $A \in \mathbb{R}^{m\times n}$, $b \in \mathbb{R}^m$, and $x \in \mathbb{R}^n$ such that $Ax = b$. Let B be a *basis*, i.e. a set $B \subseteq \{1, \dots, n\}$ with $|B| = m$, corresponding to a choice of m non-zero variables. Then

$$A_B x_B + A_N x_N = b,$$

where $A_B \in \mathbb{R}^{m\times m}$ and $A_N \in \mathbb{R}^{m\times(n-m)}$ respectively consist of the columns of A indexed by B and those not indexed by B, and x_B and x_N respectively consist of the rows of x indexed by B and those not indexed by B. Moreover, if x is a basic solution, then there is a basis B such that $x_N = 0$ and $A_B x_B = b$, and if x is a basic feasible solution, there is a basis B such that $x_N = 0$, $A_B x_B = b$, and $x_B \geq 0$.

For x with $Ax = b$ and basis B, we have that $x_B = A_B^{-1}(b - A_N x_N)$, and thus

$$\begin{aligned} f(x) = c^T x &= c_B^T x_B + c_N^T x_N \\ &= c_B^T A_B^{-1}(b - A_N x_N) + c_N^T x_N \\ &= c_B^T A_B^{-1} b + (c_N^T - c_B^T A_B^{-1} A_N) x_N. \end{aligned}$$

Suppose that we want to maximize $c^T x$ and find that

$$c_N^T - c_B^T A_B^{-1} A_N \leq 0 \quad \text{and} \quad A_B^{-1} b \geq 0. \tag{3.3}$$

Then, for any feasible $x \in \mathbb{R}^n$, it holds that $x_N \geq 0$ and therefore $f(x) \leq c_B^T A_B^{-1} b$. The basic solution x^* with $x_B^* = A_B^{-1} b$ and $x_N^* = 0$, on the other hand, is feasible and satisfies $f(x^*) = c_B^T A_B^{-1} b$. It must therefore be optimal.

If alternatively $(c_N^T - c_B^T A_B^{-1} A_N)_i > 0$ for some i, then we can increase the value of the objective by increasing $(x_N)_i$. Either this can be done indefinitely, which means that the maximum is unbounded, or the constraints force some of the variables in the basis to become smaller and we have to stop when the first such variable reaches zero. In that case we have found a new BFS and can repeat the process.

Assuming that the LP is feasible and has a bounded optimal solution, there exists a basis B^* for which (3.3) is satisfied. The basic idea behind the simplex method is to start from an initial BFS and then move from basis to basis until B^* is found. The information required for this procedure can conveniently be represented by the so-called **simplex tableau**. For a given basis B, it takes the following form:[1]

[1] The columns of the tableau have been permuted such that those corresponding to the basis appear on the left. This has been done just for convenience: in practice we will always be able to identify the columns corresponding to the basis by the embedded identity matrix.

following tableau:

-1	-2	-1	1	0	-3
-2	1	3	0	1	-4
2	3	4	0	0	0

In the dual simplex algorithm the pivot is selected by picking a row i such that $a_{i0} < 0$ and a column $j \in \{j' : a_{ij'} < 0\}$ that minimizes $-a_{0j}/a_{ij}$. Pivoting then works just like in the primal algorithm. In the example we can pivot on a_{21} to obtain

0	$-\frac{5}{2}$	$-\frac{5}{2}$	1	$-\frac{1}{2}$	-1
1	$-\frac{1}{2}$	$-\frac{3}{2}$	0	$-\frac{1}{2}$	2
0	4	7	0	1	-4

and then on a_{12} to obtain

0	1	1	$-\frac{2}{5}$	$\frac{1}{5}$	$\frac{2}{5}$
1	0	-1	$-\frac{1}{5}$	$-\frac{2}{5}$	$\frac{11}{5}$
0	0	3	$\frac{8}{5}$	$\frac{1}{5}$	$-\frac{28}{5}$

We have reached the optimum of $28/5$ with $x_1 = 11/5$, $x_2 = 2/5$, and $x_3 = 0$.

It is worth pointing out that for problems in which all constraints are inequality constraints, the optimal dual solution can also be read off from the final tableau. For problems of this type, the last m columns of the extended constraint matrix A correspond to the slack variables and therefore contain values 1 or -1 on the diagonal and 0 everywhere else. For the same reason, the last m columns of the vector c^T are 0. The values of the dual variables, each of them with opposite sign of the slack variable in the corresponding constraint, thus appear in the last m columns of the vector $(c^T - \lambda^T A)$ in the last row of the final tableau. In our example, we have $\lambda_1 = 8/5$ and $\lambda_2 = 1/5$.

4.3 Gomory's cutting plane method

Another situation where the dual simplex method can be useful is when we need to add constraints to an already solved LP. While such constraints can make the primal solution infeasible, they do not affect feasibility of the dual solution. We can therefore simply add the constraint and continue running the dual LP algorithm from the current solution until the primal solution again becomes feasible. The need to add constraints to an LP for example arises naturally in Gomory's cutting plane approach for solving integer programs (IPs). An IP is a linear program with the additional requirement that variables should be integral.

Assume that for a given IP we have already found an optimal (fractional) solution x^* with basis B, and let a_{ij} denote the entries of the final tableau, i.e. $a_{ij} = (A_B^{-1} A_j)_i$

7 Ellipsoid Method

7.1 Ellipsoid method

Consider a polytope $P = \{x \in \mathbb{R}^n : Ax \geq b\}$, given by a matrix $A \in \mathbb{Z}^{m \times n}$ and a vector $b \in \mathbb{Z}^m$. Assume for now that P is bounded and either empty or full-dimensional. Here, P is called **full-dimensional** if $\mathrm{Vol}(P) > 0$. The ellipsoid method takes the following steps to decide whether P is non-empty:

1. Let U be the largest absolute value among the entries of A and b, and define
   $x_0 = 0, \quad D_0 = n(nU)^{2n} I, \quad E_0 = E(x_0, D_0),$
   $V = (2\sqrt{n})^n (nU)^{n^2}, \quad v = n^{-n}(nU)^{-n^2(n+1)},$
   $t^* = \lceil 2(n+1)\log(V/v) \rceil.$
2. For $t = 0, \ldots, t^*$, do the following:
   1. If $t = t^*$ then stop: P is empty.
   2. If $x_t \in P$ then stop; P is non-empty.
   3. Find a violated constraint, i.e. a row j such that $a_j^T x_t < b_j$.
   4. Let $E_{t+1} = E(x_{t+1}, D_{t+1})$ with

$$x_{t+1} = x_t + \frac{1}{n+1} \frac{D_t a_j}{\sqrt{a_j^T D_t a_j}},$$

$$D_{t+1} = \frac{n^2}{n^2 - 1}\left(D_t - \frac{2}{n+1}\frac{D_t a_j a_j^T D_t}{a_j^T D_t a_j}\right).$$

The ellipsoid method is a so-called **interior point method**, because it traverses the interior of the feasible set rather than following its boundary.

7.2 Proof of correctness

Observe that E_0 is a ball centered at the origin. Given Theorem 6.2, and assuming that (i) $P \subseteq E_0$ and $\mathrm{Vol}(E_0) < V$ and that (ii) P is either empty or $\mathrm{Vol}(P) > v$, correctness of the ellipsoid method is easy to see: it either finds a point in P, thereby proving that P is non-empty, or an ellipsoid $E_{t^*} \supseteq P$ with $\mathrm{Vol}(E_{t^*}) < e^{-t^*/2(n+1)}\mathrm{Vol}(E_0) < (v/V)\mathrm{Vol}(E_0) < v$, in which case P must be empty.

We now show that the above assumptions hold, starting with the inclusion of P in E_0 and the volume of E_0. We use the following lemma.

Lemma 7.1. *Suppose $A \in \mathbb{Z}^{m \times n}$, $b \in \mathbb{R}^m$ and $m \geq n$. Let U be the largest absolute value among the entries of A and b. Then every extreme point x of the polytope $P = \{x' \in \mathbb{R}^n : Ax' \geq b\}$ satisfies $-(nU)^n \leq x_i \leq (nU)^n$ for all $i = 1, \ldots, n$.*

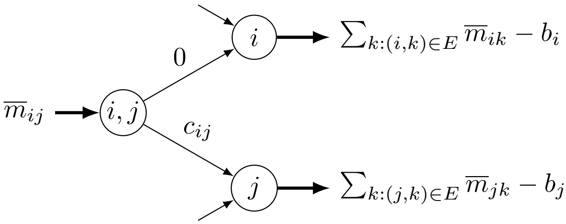


Figure 9: Representation of flow conservation constraints by a transportation problem

a feasible flow in the minimum cost flow problem. Let the flows on edges (ij, i) and (ij, j) be $\overline{m}_{ij} - x_{ij}$ and x_{ij}, respectively. The total flow into vertex i then is $\sum_{k:(i,k)\in E}(\overline{m}_{ik} - x_{ik}) + \sum_{k:(k,i)\in E} x_{ki}$, which must be equal to $\sum_{k:(i,k)\in E} \overline{m}_{ik} - b_i$. This is the case if and only if $b_i + \sum_{k:(k,i)\in E} x_{ki} - \sum_{k:(i,k)\in E} x_{ik} = 0$, which is the flow conservation constraint for vertex i in the original problem. □

9.2 Network simplex method in tableau form

When solving a transportation problem using the network simplex method, it is convenient to write it down in a tableau of the following form, where λ_i for $i = 1, \dots, n$ and μ_j for $j = 1, \dots, m$ are the dual variables corresponding to the flow conservation constraints for suppliers and consumers, respectively:

	μ_1	$\cdots$	μ_m	
λ_1	x_{11} c_{11}	$\cdots$ $\cdots$	x_{1m} c_{1m}	s_1
$\vdots$	$\vdots$ $\vdots$	$\ddots$ $\ddots$	$\vdots$ $\vdots$	$\vdots$
λ_n	x_{n1} c_{n1}	$\cdots$ $\cdots$	x_{nm} c_{nm}	s_n
	d_1	$\cdots$	d_m	

Consider the Hitchcock transportation problem given by the following tableau:

5	3	4	6	8
2	7	4	1	10
5	6	2	4	9
6	5	8	8	

increase and decrease the flow for edges along the cycle. In particular, increasing x_{21} by θ increases x_{12} and decreases x_{11} and x_{22} by the same amount. The is shown on the right of Figure 10. Increasing x_{21} by the maximum amount of $\theta = 3$ and re-computing the values of the dual variables λ_1 and μ_j, we obtain left hand tableau below.

Now, $c_{24} < \lambda_2 - \mu_4$, and we can increase x_{24} by 7 to obtain the right hand tableau below, which satisfies $c_{ij} \geq \lambda_i - \mu_j$ for all $(i,j) \notin T$ and therefore is optimal.

	-5	-3	-7	-9
0	3 $\boxed{5}$	5 $\boxed{3}$	7 $\boxed{4}$	9 $\boxed{6}$
-3	3 $\boxed{2}$	0 $\boxed{7}$	7 $\boxed{4}$	6 $\boxed{1}$
-5	0 $\boxed{5}$	-2 $\boxed{6}$	1 $\boxed{2}$	8 $\boxed{4}$

	-5	-3	-2	-4
0	3 $\boxed{5}$	5 $\boxed{3}$	2 $\boxed{4}$	4 $\boxed{6}$
-3	3 $\boxed{2}$	0 $\boxed{7}$	-1 $\boxed{4}$	[illegible] $\boxed{1}$
0	5 $\boxed{5}$	3 $\boxed{6}$	8 $\boxed{2}$	1 $\boxed{4}$

9.3 Assignment problem

An instance of the assignment problem is given by n agents and n jobs, and costs c_{ij} for assigning job j to agent i. The goal is to assign exactly one job to each agent to

$$\begin{aligned} \text{minimize} \quad & \sum_{i=1}^{n}\sum_{j=1}^{n} c_{ij}x_{ij} \\ \text{subject to} \quad & x_{ij} \in \{0,1\} \quad \text{for all } i,j = 1,\dots,n \\ & \sum_{j=1}^{n} x_{ij} = 1 \quad \text{for all } i = 1,\dots,n \\ & \sum_{i=1}^{n} x_{ij} = 1 \quad \text{for all } j = 1,\dots,n \end{aligned} \tag{9.1}$$

Except for the integrality constraints, this is a special case of the Hitchcock transportation problem. All basic solutions of the **LP relaxation** of this problem, which is obtained by replacing the integrality constraint $x_{ij} \in \{0,1\}$ by $0 \leq x_{ij} \leq 1$, are spanning tree solutions and therefore integral. Thus, both the network simplex method and the general simplex method yield an optimal solution of the original problem when applied to the LP relaxation. This is not necessarily the case, for example, for the ellipsoid method.

This problem is also known as the **weighted bipartite matching problem**. In the next lecture we will look at a polynomial time algorithm for solving this problem. As a preliminary, we state the following lemma.

Lemma 9.2. *A feasible solution $\{x_{ij}\}$ to (9.1) is optimal if there exist $\{\lambda_i\}$, $\{\mu_j\}$ such that $\lambda_i - \mu_j \leq c_{ij}$ for all i,j, and $\lambda_i - \mu_j = c_{ij}$ if $x_{ij} = 1$.*

10 Maximum Flows and Perfect Matchings

10.1 Maximum flow problem

Consider a flow network (V, E) with a single source 1, a single sink n, and finite capacities $\overline{m}_{ij} = C_{ij}$ for all $(i, j) \in E$. We will also assume for convenience that $\underline{m}_{ij} = 0$ for all $(i, j) \in E$. The **maximum flow problem** then asks for the maximum amount of flow that can be sent from vertex 1 to vertex n, i.e. the goal is to

$$\begin{aligned} &\text{maximize} && \delta \\ &\text{subject to} && \sum_{j:(i,j)\in E} x_{ij} - \sum_{j:(j,i)\in E} x_{ji} = \begin{cases} \delta & \text{if } i = 1 \\ -\delta & \text{if } i = n \\ 0 & \text{otherwise} \end{cases} \\ &&& 0 \le x_{ij} \le C_{ij} \quad \text{for all } (i,j) \in E. \end{aligned} \tag{10.1}$$

To see that this is again a special case of the minimum cost flow problem, set $c_{ij} = 0$ for all $(i, j) \in E$, and add an additional edge $(n, 1)$ with infinite capacity and cost $c_{n1} = -1$. Since the new edge $(n, 1)$ has infinite capacity, any feasible flow of the original network is also feasible for the new network. Cost is clearly minimized by maximizing the flow across the edge $(n, 1)$, which by the flow conservation constraints for vertices 1 and n maximizes flow through the original network. This is called a **circulation problem**, because there are no sources or sinks but flow merely circulates in the network.

10.2 Max-flow min-cut theorem

Consider a flow network $G = (V, E)$ with capacities C_{ij} for all $(i, j) \in E$. A **cut** of G is a partition of V into two sets, and the capacity of a cut is defined as the sum of capacities of all edges across the partition. Formally, for $S \subseteq V$, the capacity of the cut $(S, V \setminus S)$ is

$$C(S) = \sum_{(i,j)\in E\cap(S\times(V\setminus S))} C_{ij}. \tag{10.2}$$

Assume that x is a feasible flow vector that sends δ units of flow from vertex 1 to vertex n. It is easy to see that δ is bounded from above by the capacity of any cut S with $1 \in S$ and $n \in V \setminus S$. Indeed, for $X, Y \subseteq V$, let

$$f(X, Y) = \sum_{(i,j)\in E\cap(X\times Y)} x_{ij}.$$

Then, for any $S \subseteq V$ with $1 \in S$ and $n \in V \setminus S$,

$$\delta = \sum_{i\in S}\left(\sum_{j:(i,j)\in E} x_{ij} - \sum_{j:(j,i)\in E} x_{ji}\right) \tag{10.3}$$

11 Shortest Paths and Minimum Spanning Trees

11.1 Bellman's equations

In the **single-destination shortest path problem** one is given a destination $t \in V$ and simultaneously looks for shortest paths from any vertex $i \in V \setminus \{t\}$ to t. It is equivalent to the minimum cost flow problem on the same network where one unit of flow is to be routed from each vertex $i \in V \setminus \{t\}$ to t, i.e. the one with supply $b_i = 1$ at every vertex $i \in V \setminus \{t\}$ and demand $b_t = -(|V| - 1)$ at vertex t.

Let λ_i for $i \in V$ be the dual solution corresponding to an optimal spanning tree solution of this flow problem, and recall that for every edge $(i, j) \in E$ with $x_{ij} > 0$,

$$\lambda_i = c_{ij} + \lambda_j.$$

By setting $\lambda_t = 0$ and adding these equalities along a path from i to t, we see that λ_i is equal to the length of a shortest path from i to t. Moreover, since $b_i = 1$ for all $i \in V \setminus \{t\}$, and given $\lambda_t = 0$, the dual problem is to

$$\text{maximize} \sum_{i \in V \setminus \{t\}} \lambda_i \quad \text{subject to } \lambda_i \leq c_{ij} + \lambda_j \text{ for all } (i, j) \in E.$$

In an optimal solution, λ_i will thus be as large as possible subject to the constraints, i.e. it will satisfy the so-called **Bellman equations**

$$\lambda_i = \min_{j:(i,j) \in E} \{c_{ij} + \lambda_j\} \quad \text{for all } i \in V \setminus \{t\},$$

with $\lambda_t = 0$. The intuition behind these equalities is that in order to find a shortest path from i to t, one should choose the first edge (i, j) on the path in order to minimize the sum of the length of this edge and that of a shortest path from j to t. This situation is illustrated in Figure 14.

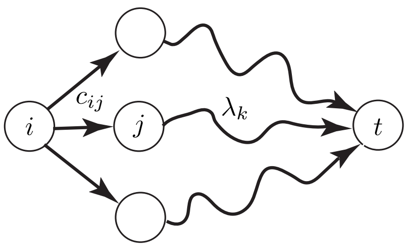


Figure 14: Illustration of the Bellman equations for the shortest path problem

11.4 Minimal spanning tree problem

The **minimum spanning tree problem** for a network (V, E) with associated costs c_{ij} for each edge $(i, j) \in E$ asks for a spanning tree of minimum cost, where the cost of a tree is the sum of costs of all its edges. This problem arises, for example, if one wishes to design a communication network that connects a given set of locations. The following property of minimum spanning trees will be useful.

Theorem 11.2. *Let (V, E) be a graph with edge costs c_{ij} for all $(i, j) \in E$. Let $U \subseteq V$ and $(u, v) \in U \times (V \setminus U)$ such that $c_{uv} = \min_{(i,j) \in U \times (V \setminus U)} c_{ij}$. Then there exists a spanning tree of minimum cost that contains (u, v).*

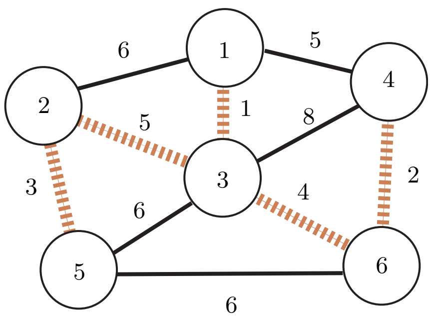
Proof. Let $T \subseteq E$ be a spanning tree of minimum cost. If $(u, v) \in T$ we are done. Otherwise, $T \cup \{(u, v)\}$ contains a cycle, and there must be another edge $(u', v') \in T$ such that $(u', v') \in U \times (V \setminus U)$. Then, $(T \cup \{(u, v)\}) \setminus \{(u', v')\}$ is a spanning tree, and its cost is no greater than that of T. □

Prim's algorithm uses this property to inductively construct a minimum spanning tree. It proceeds as follows.

1. Set $U = \{1\}$ and $T = \emptyset$.
2. If $U = V$, return T. Otherwise find an edge $(u, v) \in U \times (V \setminus U)$ such that $c_{uv} = \min_{(i,j) \in U \times (V \setminus U)} c_{ij}$.
3. Add v to U and (u, v) to T, and return to Step 2.

It is called a **greedy algorithm**, because it always chooses an edge of minimum cost.

Example. In this example, Prim's algorithm adds edges in the sequence $\{1, 3\}$, $\{3, 6\}$, $\{6, 4\}$, $\{3, 2\}$, $\{2, 5\}$.



After each iteration, we may compute and store for every $j \in V \setminus U$ a minimum cost edge to U. This only needs comparison between the previously stored edge and the edge to the vertex newly added to U. We then add to U the vertex that is closest to U. So each iteration needs time $O(|V|)$. The algorithm performs $|V| - 1$ iterations, so has overall running time of $O(|V|^2)$.

after the 3rd attempt is

$$x^\top C_2 x = x^\top \begin{pmatrix} 1 & \frac12 & \frac12 & \frac12 & \frac12 & 0 & \frac12 & 0 & \frac12 \\ \frac12 & 1 & \frac12 & 0 & \frac12 & \frac12 & \frac12 & \frac12 & 0 \\ \frac12 & \frac12 & 1 & \frac12 & 0 & \frac12 & 0 & \frac12 & \frac12 \\ \frac12 & 0 & \frac12 & 1 & \frac12 & \frac12 & \frac12 & \frac12 & 0 \\ \frac12 & \frac12 & 0 & \frac12 & 1 & \frac12 & 0 & \frac12 & \frac12 \\ 0 & \frac12 & \frac12 & \frac12 & \frac12 & 1 & \frac12 & 0 & \frac12 \\ \frac12 & \frac12 & 0 & \frac12 & 0 & \frac12 & 1 & \frac12 & \frac12 \\ 0 & \frac12 & \frac12 & \frac12 & \frac12 & 0 & \frac12 & 1 & \frac12 \\ \frac12 & 0 & \frac12 & 0 & \frac12 & \frac12 & \frac12 & \frac12 & 1 \end{pmatrix} x.$$

C_2 is not positive definite. (It's eigenvalues are $4, 1, 1, 1, 1, 1, 1, -\frac12, -\frac12$.) This means that the quadratic form $x^\top C_2 x$ has local minima. One such is given by $x^\top = (1/9)(1,1,1,1,1,1,1,1,1)$, which gives $x^\top C_2 x = 4/9$. But better is
$x^\top = (1/3)(1,0,0,0,0,1,0,1,0)$, which gives $x^\top C_2 x = 1/3$. How might we prove this is best?

Let J_2 be the 9×9 matrix of 1s. Note that for x to be a vector of probabilities, we must have $x^\top J x = 9$. As with the max-cut problem we think of relaxing $xx^\top$ to a matrix $X \succeq 0$ and consider the SDP

$$\text{minimize } \operatorname{tr}(C_2 X) \text{ s.t. } X \in S_n,\ X \geq 0,\ X \succeq 0 \ \text{ and } \ \operatorname{tr}(J_2 X) = 9.$$

One can numerically compute that the optimal value of this SDP. It is $1/3$. This provides a lower bound on the probability that the players do not rendezvous by the end of the 3rd attempt. This is achieved by $x^\top = (1/3)(1,0,0,0,0,1,0,1,0)$ — so this strategy does indeed minimize the probability that they have not yet met by the end of the 3rd attempt.

These ideas can be extended (Weber, 2008) to show that the expected time to rendezvous is minimized when players adopt a strategy in which they choose their first telephone at random, and if this does not connect them then on successive pairs of subsequent attempts they choose aa, bc or cb, each with probability $1/3$. Given that they fail to meet at the first attempt, the expected number of further attempts required is $5/2$. This is less than 3, i.e. the expected number of steps required if players simply try telphones at random at each attempt. There are many simply-stated but unsolved problems in the field of search games.

where $w \in \mathbb{R}$ and $y \in \mathbb{R}^n$. The Lagrangian has a finite maximum for $v \in \mathbb{R}$ and $x \in \mathbb{R}^m$ with $x \geq 0$ if and only if $\sum_{j=1}^n y_j = 1$, $\sum_{j=1}^n p_{ij} y_j \leq w$ for $i = 1, \dots, m$, and $y \geq 0$. The dual of (15.1) is therefore

$$\begin{aligned} &\text{minimize} \quad w \\ &\text{subject to} \quad \sum_{j=1}^n p_{ij} y_j \leq w \quad \text{for } i = 1, \dots, m, \quad \sum_{j=1}^n y_j = 1, \quad y \geq 0. \end{aligned}$$

It is easy to see that the optimal solution of the dual is $\min_{y \in Y} \max_{x \in X} p(x, y)$, and the theorem follows from strong duality. □

The number $\max_{x \in X} \min_{y \in Y} p(x, y) = \min_{y \in Y} \max_{x \in X} p(x, y)$ is called the **value** of the matrix game with payoff matrix P.

The solution of a matrix game can be found by solving the linear program (15.1). This problem can be simplified by first adding a constant to every element of P so that $P > 0$. This does not change the equilibrium of the game, but ensures that at the solution we must have $v > 0$. By setting $x' = x/v$ and noting that $1/v = \sum_i x'_i$, we can rewrite (15.1) as

$$\text{minimize} \quad \sum_{i=1}^m x'_i \quad \text{subject to} \quad \sum_{i=1}^m x'_i p_{ij} \geq 1 \text{ for } j = 1, \dots, n, \quad x' \geq 0.$$

Alternatively, we might apply a similar transformation to the dual and solve

$$\text{maximize} \quad \sum_{i=1}^n y'_i \quad \text{subject to} \quad \sum_{i=1}^n p_{ij} y'_i \leq 1 \text{ for } j = 1, \dots, n, \quad \bar{y}' \geq 0.$$

15.3 Equilibria of matrix games

The minimax theorem implies that every matrix game has an equilibrium, and in fact characterizes the set of equilibria of these games.

Theorem 15.2. *A pair of strategies $(x, y) \in X \times Y$ is an equilibrium of the matrix game with payoff matrix P if and only if it is a minimax point, i.e.*

$$\begin{aligned} \min_{y' \in Y} p(x, y') &= \max_{x' \in X} \min_{y' \in Y} p(x', y') \quad \textit{and} \\ \max_{x' \in X} p(x', y) &= \min_{y' \in Y} \max_{x' \in X} p(x', y'). \end{aligned} \tag{15.2}$$

Proof. For all $(x, y) \in X \times Y$,

$$\min_{y' \in Y} \max_{x' \in X} p(x', y') \leq \max_{x' \in X} p(x', y) \geq p(x, y) \geq \min_{y' \in Y} p(x, y') \leq \max_{x' \in X} \min_{y' \in Y} p(x', y'),$$

and the first and last term are equal by Theorem 15.1.

If (x, y) is an equilibrium, the second and third inequality hold with equality. This means that the first and last inequality have to hold with equality as well, and (15.2) follows.

On the other hand, if (15.2) is satisfied, then the first and last inequality hold with equality. This means that the second and third inequality have to hold with equality as well, so (x, y) is an equilibrium. □

Some other properties specific to matrix games are stated in the following theorem. These are that all equilibria yield the same payoffs and that any pair of strategies of the two players, such that each of them is played in some equilibrium, is itself an equilibrium.

Theorem 15.3. *Let $(x, y), (x', y') \in X \times Y$ be equilibria of the matrix game with payoff matrix P. Then $p(x, y) = p(x', y')$, and (x, y') and (x', y) are equilibria as well.*

Proof. Since equilibrium strategies are best responses to each other, we have that

$$p(x, y) \le p(x, y') \le p(x', y') \le p(x', y) \le p(x, y).$$

Since the first and last term are the same, the inequalities have to hold with equality and the first claim follows. Then,

$$\begin{aligned}
p(x, y') &= p(x', y') \ge p(z, y') && \text{for all } z \in X, \\
p(x, y') &= p(x, y) \le p(x, z) && \text{for all } z \in Y, \\
p(x', y) &= p(x, y) \ge p(z, y) && \text{for all } z \in X, \text{ and} \\
p(x', y) &= p(x', y') \ge p(x', z) && \text{for all } z \in X,
\end{aligned}$$

where the inequalities hold because (x, y) and (x', y') are equilibria. Thus (x, y') and (x', y) are pairs of strategies that are best responses to each other, and the second claim follows as well. □

Theorems 15.1, 15.2, and 15.3 together also imply that the set of equilibria of a matrix game is convex.

16 Solution of Two-person Games

16.1 Nash's theorem

Many of the results concerning equilibria of matrix games do *not* carry over to bimatrix games, with the exception of existence.

Theorem 16.1 (Nash, 1951)**.** *Every bimatrix game has an equilibrium.*

We use the following result.

Theorem 16.2 (Brouwer fixed point theorem)**.** *Let $f : S \to S$ be a continuous function, where $S \subseteq \mathbb{R}^n$ is closed, bounded, and convex. Then f has a fixed point.*

Proof of Theorem 16.1. Define X and Y as before, and observe that $X \times Y$ is closed, bounded, and convex. For $x \in X$ and $y \in Y$ define $s_i(x, y)$ and $t_j(x, y)$ as the additional payoff the two players could obtain by playing their ith or jth pure strategy instead of x or y, i.e.

$$\begin{aligned} s_i(x,y) &= \max\{0, p(e_i^m, y) - p(x,y)\} && \text{for } i = 1, \dots, m \text{ and} \\ t_j(x,y) &= \max\{0, q(x, e_j^n) - q(x,y)\} && \text{for } j = 1, \dots, n, \end{aligned}$$

where e_ℓ^k denotes the ℓth unit vector in $\mathbb{R}^k$. Further define $f : (X \times Y) \to (X \times Y)$ by letting $f(x, y) = (x', y')$ with

$$x'_i = \frac{x_i + s_i(x,y)}{1 + \sum_{k=1}^m s_k(x,y)} \qquad \text{and} \qquad y'_j = \frac{y_j + t_j(x,y)}{1 + \sum_{k=1}^n t_k(x,y)}$$

for $i = 1, \dots, m$ and $j = 1, \dots, n$. Function f is continuous, so by Theorem 16.2 is must have a fixed point, i.e. a pair of strategies $(x, y) \in X \times Y$ such that $f(x, y) = (x, y)$.

Further observe that there has to exist $i \in \{1, \dots, m\}$ such that $x_i > 0$ and $s_i(x, y) = 0$, since otherwise

$$p(x,y) = \sum_{k=1}^m x_k p(e_k^m, y) > \sum_{k=1}^m x_k p(x,y) = p(x,y).$$

Therefore, and since (x, y) is a fixed point,

$$x_i = \frac{x_i + s_i(x,y)}{1 + \sum_{k=1}^m s_k(x,y)}$$

and thus

$$\sum_{k=1}^m s_k(x,y) = 0.$$

This means that for $k = 1, \ldots, m$, $s_k(x, y) = 0$, and therefore

$$p(x, y) \geq p(e_k^m, y).$$

It follows that

$$p(x, y) \geq p(x', y) \quad \text{for all } x' \in X.$$

An analogous argument shows that $q(x, y) \geq q(x, y')$ for all $y' \in Y$, so (x, y) must be an equilibrium. □

Our requirement that a bimatrix game has a finite number of actions is crucial for this result. This can be seen very easily by considering a game where the set of actions of each player is the set of natural numbers, and players get a payoff of 1 if they choose a number that is greater than the one chosen by the other player, and zero otherwise.

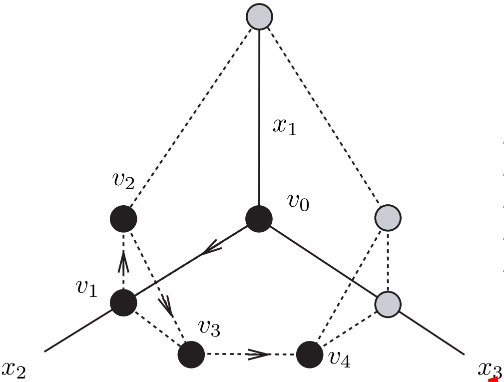
16.2 The complexity of finding an equilibrium

The proof of Theorem 16.1 relies on fixed points of a continuous function and does not give rise to a finite method for finding an equilibrium. Quite surprisingly, equilibrium computation turns out to be more or less a combinatorial problem.

Define the **support** of strategy $x \in X$ as $S(x) = \{i \in \{1, \ldots, m\} : x_i > 0\}$, and that of strategy $y \in Y$ as $S(y) = \{j \in \{1, \ldots, m\} : y_j > 0\}$. It is easy to see that a mixed strategy is a best response if and only if all pure strategies in its support are best responses: if one of them was not a best response, then the payoff could be increased by reducing the probability of that strategy, and increasing the probabilities of the other strategies in the support appropriately. In other words, randomization over the support of an equilibrium does not happen for the player's own sake, but to allow the other player to respond in a way that sustains the equilibrium.

It also follows from these considerations that finding an equilibrium boils down to finding its supports. Once the supports are known, the precise strategies can be computed by solving a set of equations, which in the two-player case are linear. For supports of sizes k and ℓ, there is one equation for each player stating that the probabilities sum up to one, and $k-1$ or $\ell-1$ equations, respectively, stating that the expected payoff is the same for every pure strategy in the support. Solving these $k+\ell$ equations in $k+\ell$ variables yields k values for player 1 and ℓ values for player 2. If the solution corresponds to a strategy profile and expected payoffs are maximized by the pure strategies in the support, then an equilibrium has been found. An equilibrium with supports of size two in the game of chicken would have to satisfy $x_1 + x_2 = 1$, $y_1 + y_2 = 1$, $2x_1 + 1x_2 = 3x_1 + 0x_2$, and $2y_1 + 1y_2 = 3y_1 + 0y_2$. The unique solution, $x_1 = x_2 = y_1 = y_2 = 1/2$, also satisfies the additional requirements and therefore is an equilibrium. No equilibrium with full supports exists in the prisoner's dilemma, because the corresponding system of equalities does not have a solution.

(or z_i) associated with an twice-represented strategy i, which is complementary to the variable z_i or (or x_i) that was decreased to 0 at the previous step.



	x_1	x_2	x_3	z_1	z_2	z_3
v_0 :	0	0	0	1	1	1
v_1 :	**0**	$\frac{1}{3}$	0	**0**	1	$\frac{1}{3}$
v_2 :	$\frac{1}{6}$	$\frac{1}{3}$	**0**	0	1	**0**
v_3 :	**0**	$\frac{1}{3}$	$\frac{1}{6}$	**0**	$\frac{1}{2}$	0
v_4 :	0	$\frac{1}{6}$	$\frac{1}{3}$	$\frac{1}{2}$	0	0

In this example there is only one equilibrium. If we had started with $i = 1$ we would have reached v_4 along a different path.

The algorithm will find one equilibrium, but if there is more than one it cannot guarantee to find them all. Starting with different i to be dropped we might reach the same equilibrium or a different equilibrium. If we start at an one equilibrium we will follow a path to a different equilibrium or to v_0.

There is an interesting corollary of this analysis.

Corollary 16.6. *A nondegenerate bimatrix game has an odd number of Nash equilibria.*

Proof. Let V be the set of vertices in which only Player I's first strategy might be missing (i.e. such that $x_1 z_1 > 0$). Every equilibrium of $P \times Q$ is a member of V (since equilibriums are vertices for which all strategies are represented). In the graph formed by vertices in V, each vertex has degree 1 or 2. So this graph consists of disjoint paths and cycles. The endpoints of the paths are the Nash equilibriums and the special vertex $(x, y) = (0, 0)$. There are an even number of endpoints, so the number of Nash equilibria must be odd. □

Consider for example the bimatrix game given by

$$P = \begin{pmatrix} 3 & 3 \\ 2 & 5 \\ 0 & 6 \end{pmatrix} \qquad \text{and} \qquad Q = \begin{pmatrix} 3 & 2 \\ 2 & 6 \\ 3 & 1 \end{pmatrix},$$

Indexing z and w by $M = \{1, \ldots, m\}$ and $N = \{m+1, \ldots, m+n\}$, respectively, the constraint $Q^T x + w = 1$ can be written in tableau form as follows:

x_1	x_2	x_3	w_4	w_5	
3	2	3	1	0	1
2	6	1	0	1	1

Assume that label 2 is dropped by increasing x_2 from 0. By pivoting we obtain the following tableau:

$\frac{7}{3}$	0	$\frac{8}{3}$	1	$-\frac{1}{3}$	$\frac{2}{3}$
$\frac{1}{3}$	1	$\frac{1}{6}$	0	$\frac{1}{6}$	$\frac{1}{6}$

The second row now corresponds to variable x_2 that has entered the basis. On the other hand, variable w_5 has left the basis. We thus want to turn to the constraint $Py + z = 1$ and drop the duplicate label 5. The initial tableau for this constraint looks as follows:

y_4	y_5	z_1	z_2	z_3	
3	3	1	0	0	1
2	5	0	1	0	1
0	6	0	0	1	1

By pivoting on the second column, corresponding to y_5, and on the third row, we pick up label 3 and obtain the following tableau:

3	0	1	0	$-\frac{1}{2}$	$\frac{1}{2}$
2	0	0	1	$-\frac{5}{6}$	$\frac{1}{6}$
0	1	0	0	$\frac{1}{6}$	$\frac{1}{6}$

Pivoting one more time in each of the two polytopes, we drop label 3 to pick up label 4:

$\frac{7}{8}$	0	1	$\frac{3}{8}$	$-\frac{1}{8}$	$\frac{1}{4}$
$\frac{3}{16}$	1	0	$-\frac{3}{48}$	$\frac{3}{16}$	$\frac{1}{8}$

and then drop label 4 to pick up label 2:

0	0	1	$-\frac{3}{2}$	$\frac{3}{4}$	$\frac{1}{4}$
1	0	0	$\frac{1}{2}$	$-\frac{5}{12}$	$\frac{1}{12}$
0	1	0	0	$\frac{1}{6}$	$\frac{1}{6}$

At this point we have a fully labelled pair. The final tableaus are the final two above. Reading off the values of x and y from the last column of each tableau and scaling them appropriately yields the equilibrium $x = (0, 1/3, 2/3)$, $y = (1/3, 2/3)$.

If $\phi_j(N) < \phi_j(N - \{i\})$, player j might threaten player i, "*Give me more or I will convince the others to exclude you and I will be better off.*" Player i has a valid counter-objection if he can point out that if he gets the others to exclude j then i will be better off by at least as much.

If every such objection has a counter-objection, then

$$\phi_i(N) - \phi_i(N - \{j\}) = \phi_j(N) - \phi_j(N - \{i\}).$$

The only solution to this is the Shapley value.

19.3 Bargaining theory

Bargaining theory investigates how agents should cooperate when non-cooperation may result in outcomes that are Pareto dominated. Formally, a (two-player) **bargaining problem** is a pair (F, d) where $F \subseteq \mathbb{R}^2$ is a convex set of **feasible outcomes** and $d \in F$ is a **disagreement point** that results if players fail to agree on an outcome. Here, convexity corresponds to the assumption that any lottery over feasible outcomes is again feasible. A **bargaining solution** then is a function that assigns to every bargaining problem (F, d) a unique element of F.

An example of a bargaining problem is the so-called ultimatum game given by $F = \{(v_1, v_2) \in \mathbb{R}^2 : v_1 + v_2 \leq 1\}$ and $d = (0, 0)$, in which two players receive a fixed amount of payoff if they can agree on a way to divide this amount among themselves. This game has many equilibria when viewed as a normal-form game, since disagreement results in a payoff of zero to both players. Players' preferences regarding these equilibria differ, and bargaining theory tries to answer the question which equilibrium should be chosen. More generally, a two-player normal-form game with payoff matrices $P, Q \in \mathbb{R}^{m \times n}$ can be interpreted as a bargaining problem where $F = \mathrm{conv}(\{(p_{ij}, q_{ij}) : i = 1, \ldots, m,\ j = 1, \ldots, n\})$, $d_1 = \max_{x \in X} \min_{y \in Y} p(x, y)$, and $d_2 = \max_{y \in Y} \min_{x \in X} q(x, y)$, given that $(d_1, d_2) \in F$. Here, $\mathrm{conv}(S)$ denotes the convex hull of set S.

Two kinds of approaches to bargaining exist in the literature: a strategic one that considers iterative procedures resulting in an outcome in F, and an axiomatic one that tries to identify bargaining solutions that possess certain desirable properties. We will focus on the axiomatic approach in this lecture.

19.4 Nash's bargaining solution

For a given bargaining problem (F, d), Nash proposed to

$$\begin{aligned} \text{maximize} \quad & (v_1 - d_1)(v_2 - d_2) \\ \text{subject to} \quad & v \in F \\ & v \geq d. \end{aligned} \tag{19.1}$$

21.2 The revenue equivalence theorem

The **symmetric independent private values model** (SIPV) concerns the auction of a single item, with risk neutral seller and bidders. Each bidder knows his own valuation of the item, which he keeps secret, and valuations of the bidders can be modelled as i.i.d. random variables. Important questions are

- what type of auction generates the most revenue for the seller?
- if seller or bidders are risk averse, which auction would they prefer?
- which auctions make it harder for the bidders to collude?

Let us begin with an intuitive result.

Lemma 21.1. *In any SIPV auction in which the bidders bid optimally and the item is awarded to the highest bidder, the bids are ordered the same as the valuations.*

Proof. Consider an auction satisfying the terms of the lemma. Let $e(p)$ be the minimal expected payment that a bidder can make if he wants to win the item with probability p. Notice that $e(p)$ must be a convex function of p, i.e. $e(\alpha p + (1-\alpha)p') \leq \alpha e(p) + (1-\alpha)e(p')$. This is because one strategy for winning with probability $\alpha p + (1-\alpha)p'$ is to bid so as to either win with probability p or p', doing these with probabilities α and $1-\alpha$ respectively. Since $e(p)$ is convex it is differentiable at all but a countable number of points. A bidder who has valuation θ and bids so as to win with probability p has expected profit $\pi(\theta) = p\theta - e(p)$. Assuming that p is chosen optimally, the relation between p and θ is determined by

$$\frac{\partial \pi}{\partial p} = \theta - e'(p) = 0. \qquad (21.1)$$

Since $e'(p)$ is nondecreasing in p, it follows that $p(\theta)$ must be nondecreasing in θ. As the item goes to the highest bidder, the probability of winning increases with the the bid, and so the optimal bid must be nondecreasing in the valuation θ. □

We say that two auctions have the same **bidder participation** if any bidder who finds it profitable to participate in one auction also finds it profitable to participate in the other. The following result is remarkable, as different auctions can have completely different rules and the bidders' optimal bidding strategies will differ.

Theorem 21.2 (Revenue equivalence theorem)**.** *The expected revenue obtained by the seller is the same for any two SIPV auctions that (a) award the item to the highest bidder, and (b) have the same bidder participation.*

Proof. Suppose there are n bidders. From (21.1) we have

$$\frac{d}{d\theta} e(p(\theta)) = e'(p)p'(\theta) = \theta p'(\theta).$$