MATHEMATICS OF OPERATIONS RESEARCH SUMMARY

ANDREW TULLOCH

1. Optimization

Definition. *An optimization problem has the standard form,* $\min f(x)$ *s.t.* $h(x) = b, x \in X$. *The set* $X(b) = \{x \in X : h(x) = b\}$ *is called the feasible set, and a problem is feasible if* $X(b)$ *is non-empty and is bounded if* $f(x)$ *is bounded from below no* $X(b)$. *A vector* $x^\star$ *is optimal if it is in the feasible set and minimizes* f *among all vectors in the feasible set.*

Definition. *The* ***Lagrangian*** *associated with is*

$$L(x, \lambda) = f(x) - \lambda^T(h(x) - b) \tag{1.1}$$

Theorem. Let $x \in X$ and $\lambda \in \mathbb{R}^m$ such that $L(x, \lambda) = \inf_{x' \in x} L(x', \lambda)$ and $h(x) = b$. Then x is an optimal solution.

Proof. $\min_{x' \in X(b)} f(x') \geq \min_{x' \in X}(f(x') - \lambda^T(h(x')) - b) = f(x) - \lambda^T(h(x) - b) = f(x)$ □

Theorem. To minimize $f(x)$ subject to $h(x) \leq b, x \in X$,

(i) Introduce a vector z of slack variables to obtain the equivalent problem $\min f(x)$ s.t. $h(x) + z = b, x \in X, z \geq 0$.
(ii) Compute $L(x, z, \lambda) = f(x) - \lambda^T(h(x) + z - b)$
(iii) Define $Y = \{\lambda \in \mathbb{R}^m | \inf_{x \in X, z \geq 0} L(x, z, \lambda) > -\infty\}$
(iv) For each $\lambda \in Y$, minimize $L(x, z, \lambda)$ subject only to regional constraints - so finding $x^\star(\lambda), z^\star(\lambda)$ satisfying $L(x^\star(\lambda), z^\star(\lambda), \lambda) = \inf_{x \in X, z \geq 0} L(x, z, \lambda)$.
(v) Find $\lambda^\star \in Y$ such that $(x^\star(\lambda), z^\star(\lambda^\star))$ is feasible - so $x^\star(\lambda^\star) \in X$, $z^\star(\lambda^\star) \geq 0$, and $h(x^\star(\lambda^\star)) + z^\star(\lambda^\star) = b$. By , $x^\star(\lambda^\star)$ is optimal.

Definition. *Denote* $\phi(b) = \inf_{x \in X(b)} f(x)$ *the solution of our optimization problem, and define the* ***Lagrange dual function*** $G : \mathbb{R}^m \to \mathbb{R}$ *as the minimum value of the Lagrangian over* X, *so* $g(\lambda) = \inf_{x \in X} L(x, \lambda)$. *For all* $\lambda \in \mathbb{R}^m$,

$$\inf_{x \in X(b)} f(x) = \inf_{x \in X(b)} L(x, \lambda) \geq \inf_{x \in X} L(x, \lambda) = g(\lambda) \tag{1.2}$$

so we have a lower bound on the optimal value of our problem. Thus, the dual problem is to ***maximize the lower bound****, thus* $\max g(\lambda)$ *s.t.* $\lambda \in Y$, *where* $Y = \{\lambda \in \mathbb{R}^m | g(\lambda) > -\infty\}$.

Theorem. We have the **weak duality theorem**,

$$\inf_{x \in X(b)} \geq \max_{\lambda \in Y} g(\lambda). \tag{1.3}$$

The primal problem satisfies **strong duality** if this holds with equality — so there exists λ such that $\phi(b) = g(\lambda)$.

Proof. $\inf_{x \in X(b)} f(x) = \inf_{x \in X(b)} L(x, \lambda) \geq \inf_{x \in X} L(x, \lambda) = g(\lambda)$. □

Definition. *Call a hyperplane* $\alpha : \mathbb{R}^m \to \mathbb{R}$ *a* ***supporting hyperplane*** *to* ϕ *at* b *if* $\alpha(c) = \phi(b) - \lambda^T(b - c)$ *and* $\phi(c) \geq \phi(b) - \lambda^T(b - c)$ *for all* $c \in \mathbb{R}^m$.

Theorem. There exists a (no-vertical) supporting hyperplane to ϕ at b if and only if the problem satisfies strong duality.

Proof. ($\Rightarrow$) is to note that

$$\phi(b) \leq \inf_{c \in \mathbb{R}^m} \phi(c) - \lambda^T(c - b) \tag{1.4}$$

$$= \inf_{c \in \mathbb{R}^m} \inf_{x \in X(c)} f(x) - \lambda^T(h(x) - b) \tag{1.5}$$

$$= \inf_{x \in X} L(x, \lambda) = g(\lambda). \tag{1.6}$$

($Leftarrow$) is to choose the maximizing λ that achieves equality, so $\phi(b) \leq f(x) - \lambda^T(h(x) - b)$ and minimize over $x \in X(c)$ obtains $\phi(b) - \lambda^T(b - c) \leq \phi(c)$. □

2. Linear Programming

Theorem. Suppose that ϕ is convex and $b \in \mathbb{R}$ lies in the interior of the set of points where ϕ is finite. Then there exists a (non-vertical) supporting hyperplane to ϕ at b.

Theorem. Consider the optimization problem, $\min f(x)$ s.t. $h(x) \leq b, x \in X$, and let ϕ be given by $\phi(b) = \inf_{x \in X(b)} f(x)$. Then ϕ is convex when X, f and h are convex.

Definition. *A linear program is in* ***general form*** *when written as*

$$\min\{c^T x | Ax \geq x, x \geq 0\}. \tag{2.1}$$

A linear program of the form $\min\{c^T x | Ax = b, x \geq 0\}$ *is said to be in* ***standard form****.*

Theorem. A linear program in general form can be written with slack variables as $\min\{c^T x | Ax - z = b, x, z \geq 0\}$. Then $X = \{(x, z) : x \geq 0, z \geq 0\}$, and the Lagrangian is $L((x, z), \lambda) = c^T x - \lambda^T(Ax - z - b) = (c^T - \lambda^T A)x + \lambda^T z + \lambda^T b$ with finite minimum over X if and only if $\lambda \in Y = \{\mu \in \mathbb{R}^m | c^T - \mu^T A \geq 0, \mu \geq 0\}$. Thus $g(\lambda) = \inf_{(x,z) \in X} L((x, z), \lambda) = \lambda^T b$. The dual problem is thus $\max\{b^T \lambda | A^T \lambda \leq c, \lambda \geq 0\}$.

Analogously, the dual of the **standard form** is $\max b^T \lambda | A^T \lambda \leq c$.

Theorem. Let x, λ be feasible solutions for the primal in general form and dual of general form, respectively. Then x, λ are optimal if and only if they satisfy complementary slackness - so $(c^T - \lambda^T A)x = 0$ and $\lambda^T(Ax - b) = 0$.

Proof. For x, λ optimal, we have $c^T x = \lambda^T b \leq c^T x - \lambda^T(Ax - b) \leq c^T x$, so holds with equality. □

Theorem. Suppose f, h are continuously differentiable on $\mathbb{R}^n$, and there exists unique function $x^\star : \mathbb{R}^m \to \mathbb{R}^n$ and $\lambda^\star : \mathbb{R}^m \to \mathbb{R}^m$ such that for each $b \in \mathbb{R}^m$, $h(x^\star(b)) = b$, $\lambda^\star(b) \leq 0$ and $f(x^\star(b)) = \phi(b) = \inf\{f(x) - \lambda^\star(b)^T(h(x) - b) | x \in \mathbb{R}^n\}$. If $x^\star$ and $\lambda^\star$ are continuously differentiable, then

$$\frac{\partial \phi}{\partial b_i}(b) = \lambda_i^\star(b). \tag{2.2}$$

Proof. Take derivatives of $\phi(b) = f(x^\star(b)) - \lambda^\star(b)^T(h(x^\star(b)) - b)$. □

3. The Simplex Method

Definition. *Consider the problem* $\max c^T x$ *s.t.* $Ax = b, x \geq 0$, *where* $A \in R^{m \times n}$ *and* $b \in R^m$. *Call a solution* $x \in \mathbb{R}^n$ *of the equation* $Ax = b$ ***basic*** *if at most* m *of its entries are non-zero - so there exists a set* $B \subseteq \{1, \ldots, n\}$ *with* $|B| = m$ *such that* $x_i = 0$ *if* $i \notin B$. *The set* B *is called the* ***basis,*** *and* x_i *is* ***basic*** *if* $i \in B$ *and* ***non-basic*** *if* $i \notin B$. *A basic solution* x *that also satisfies* $x \geq 0$ *is called a* ***basic feasible solution****.*

Theorem. x is a basic feasible solution of $Ax = b$ if and only if it is an extreme point of the set $X(b) = \{x : Ax = b, x \geq 0\}$.

Proof. Take convex combinations of points in the set $X(b) = \{x : Ax = b, x \geq 0\}$. Then we can show that any convex combination of $x, y \in X(b)$ with $x = \delta y + (1 - \delta)z$ must have $y = z$.

Alternatively, we can perturb a non BFS feasible solution. □

Theorem. If the linear program $\max c^T x$ s.t. $Ax = b, x \geq 0$ is feasible and bounded, then it has an optimal solution that is also a basic feasible solution.

Proof. Proceed by induction on the number of non-zero entries - finding x, y, z as before and choosing δ to reduce the number of non-zero entries in x. □

Definition. *The simplex method consists of the following steps.*

$$\begin{Bmatrix} (a_{ij}) & a_{i0} \\ a_{0j} & a_{00} \end{Bmatrix} \tag{3.1}$$

(i) Find an initial BFS with basis B.
(ii) Check whether $a_{0j} \leq 0$ *for every* j. *If yes, the current solution is optimal, so stop.*
(iii) Choose j *such that* $a_{0j} > 0$, *and choose* $i \in \{i' | a_{i'j} > 0\}$ *to minimize* a_{i0}/a_{ij}. *If* $a_{ij} \leq 0$ *for all* i, *the problem is unbounded, so stop. If multiple rows minimize* $\frac{a_{i0}}{a_{ij}}$, *the problem has a degenerate BFS.*
(iv) Update the tableau by multiplying row i *by* $\frac{1}{a_{ij}}$ *and adding* $-\frac{a_{ij}}{a_{ij}}$ *multiples of row* i *to each row* $k \neq i$. *Then return to step (ii).*

4. Advanced Simplex Procedures

Definition. *Two phase simplex method can be used where it is difficult to find an initial BFS. We proceed as follows:*

(i) *Bring the constraints into equality form. For each constraint in which the slack variable and the right hand side have opposite signs, or in which there is no slack variable, add a new artificial variable that has the same sign as the right hand side.*
(ii) *Minimize the sum of the artificial variables, starting from the BFS where the absolute value of the artificial variable for each constraint, or the slack variable in the case there is no artificial variable, is equal to that of the right hand side.*
(iii) *If some artificial variable has a positive value in the optimal solution, the original problem is infeasible — stop.*
(iv) *Now solve the original problem, starting from the BFS in Phase I.*

Definition. *We first find a dual-feasible solution, and proceed by selecting a row i such that $a_{i0} < 0$ and a column $j \in \{j' | a_{ij'} < 0\}$ that minimized $-\frac{a_{0j}}{a_{ij}}$. We can then pivot just like in the primal algorithm.*

Theorem (Gomory's Cutting-Plane method)**.** If we constrain our variables to be integral, we can first solve the linear program and add an extra constraint.

(i) Find an optimal (fractional) solution to the LP.
(ii) If $x^\star$ is not integral, there is some row i with a_{i0} not integral, and so for every feasible solution x,

$$x_i + \sum_{j \in \mathbb{N}} \lfloor a_{ij} \rfloor x_j \leq x_i + \sum_{j \in \mathbb{N}} a_{ij} x_j = a_{i0} \tag{4.1}$$

as $x \geq 0$ for feasibility.
(iii) If $x$is integral, the LHS is integral, and so the inequality holds if the right-hand side is rounded down, so

$$x_i + \sum_{j \in N} \lfloor a_{ij} \rfloor x_j \leq \lfloor a_{i0} \rfloor. \tag{4.2}$$

(iv) Thus, we can add this as a constraint and continue solving the simplex method, and obtain a modified version of our original problem. Iterating this results in a solution to the IP.

5. Complexity of Problems and Algorithms

Definition. *A problem is in P if there exists a Turing machine M and $k \in \mathbb{N}$ with the following property - for every $x \in \{0,1\}^\star$, if M is started with input x, then after $\mathcal{O}(|x|^k)$ steps it halts with output $f(x)$.*

$L \subseteq \{0,1\}^\star$ is in NP if there exists a Turing machine M and $k \in \mathbb{N}$ with the property that for every $x \in \{0,1\}^\star$, $x \in L$ if and only if there exists a certificate $y \in \{0,1\}^\star$ with $|y| = \mathcal{O}(|x|^k)$ such that M accepts (x, y) after $\mathcal{O}(|x|^k)$ steps.

6. The Complexity of Linear Programming

Theorem. Consider the linear problem of minimizing $-x_n$ subject to $\epsilon \leq x_1 \leq 1$, $\epsilon x_{i-1} \leq x_i \leq 1 - \epsilon x_{i-1}$ for $i = 2, \ldots, n$. Then there exists a pivoting rule and an initial BFS such the simplex methods requires $2^n - 1$ iterations before terminating.

Proof. Use the problem $\min -x_n$ such that $\epsilon \leq x_1 \leq 1$, $\epsilon x_{i-1} \leq x_i \leq 1 - \epsilon x_{i-1}$ for $i = 2, \ldots, n$ which traverses each vertex of the $\{0,1\}^n$ cube. □

Definition. *Given a symmetric positive definite matrix $D \in R^{n \times n}$ and $z \in \mathbb{R}^n$, the set of points*

$$E = E(z, D) = \{x \in \mathbb{R}^n | (x - z)^T D^{-1} (x - z) \leq 1\} \tag{6.1}$$

is called an ellipsoid with center z.

Theorem. Let $E = E(z, D)$ be an ellipsoid in $\mathbb{R}^n$ and $a \in \mathbb{R}^n$ nonzero. Consider the half-space $H = \{x \in \mathbb{R}^n | a^T x \geq a^T z\}$, and let $z' = z + \frac{1}{n+1} \frac{Da}{sqrt a^T Da}$, $D' = \frac{n^2}{n^2-1}(D - \frac{2}{n+1} \frac{Daa^T D}{a^T Da})$. Then D' is symmetric and positive definite, and so $E' = E(z', D')$ is an ellipsoid. Moreover, $E \cap H \subseteq E'$ and $\operatorname{Vol} E' < \exp(-\frac{1}{2(n+1)}) \operatorname{Vol} E$.

Theorem. Let $S : \mathbb{R}^n \to \mathbb{R}^n$ be an affine transformation given by $S(x) = Dx + b$, and let $L \subseteq \mathbb{R}^n$. Then $\operatorname{Vol} S(L) = |\det D| \operatorname{Vol} L$.

7. The Ellipsoid Method

Definition. *Consider a polytope $P = \{x \in \mathbb{R}^n | Ax \geq b\}$, with $A \in Z^{m \times n}$ and $b \in Z^m$. Assume that P is bounded and either empty or full-dimensional ($\operatorname{Vol} P > 0$). The ellipsoid method proceeds as follows to decide whether P is non-empty.*

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

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

Proo

8. Graphs and Flows

Definition. *$x \in \mathbb{R}^{n \times n}$ is a **minimum cost flow** of G if it is an optimal solution of the following optimization problem: $\min \sum_{(i,j) \in E} c_{ij} x_{ij}$ s.t. $b_i + \sum_{j:(j,i) \in E} x_{ji} = \sum_{j:(i,j) \in E} x_{ij}$ for all $i \in V$ (source/sink balance), and $\underline{m}_{ij} \leq x_{ij} \overline{m}_{ij}$ for all $(i,j) \in E$ (capacity constraint).*

Note that $\sum_{i \in V} b_i = 0$ is required for any feasible flows to exist.

The minimum cost flow is an LP, with constraints of the form $Ax = b$, where

$$a_{ik} = \begin{cases} 1 & k\text{-th edge starts at vertex } i \\ -1 & k\text{-th edge ends at vertex } i \\ 0 & \text{otherwise} \end{cases} \tag{8.1}$$

Definition. *As solution x to the minimum cost flow problem for a connected network $G = (V, E)$ is a **spanning tree solution** if there exists a spanning tree (V, T) of G and two sets $L, U \subseteq E$ with $L \cap U = \emptyset$ and $L \cup U = E \backslash T$ such that $x_{ij} = \underline{m}_{ij}$ if $(i,j) \in L$, and $x_{ij} = \overline{m}_{ij}$ if $(i,j) \in U$. These constraints determine the values for x_{ij} for $(i,j) \in T$.*

Theorem. A flow vector is a basic solution of a minimum cost flow problem if and only if it is a spanning tree solution.

Definition. *The **Lagrangian** of the minimum cost flow problem is*

$$L(x, \lambda) = \sum_{(i,j) \in E} c_{ij} x_{ij} - \sum_{i \in V} \lambda_i (\sum_{j:(i,j) \in E} x_{ij} - \sum_{j:(j,i) \in E} x_{ji} - b_i) \tag{8.2}$$

$$= \sum_{(i,j) \in E} (c_{ij} - \lambda_i + \lambda_j) x_{ij} + \sum_{i \in V}^{\lambda_i} b_i. \tag{8.3}$$

*Let $\overline{c}_{ij} = c_{ij} - \lambda_i + \lambda_j$ be the **reduced cost** of edge $(i,j) \in E$. We have the complementary slackness conditions $\overline{c}_{ij} > 0$ implies $x_{ij} = \underline{m}_{ij}$, $\overline{c}_{ij} < 0$ implies $x_{ij} = \overline{m}_{ij}$, and $\underline{m}_{ij} < x_{ij} < \overline{m}_{ij}$ implies $\overline{c}_{ij} = 0$.*

Assume that x is a BFS associated with sets T, U, and L. Then the system of equations $\lambda_{|V|} = 0$, $\lambda_i - \lambda_j = c_{ij}$ for all $(i,j) \in T$ has a unique solution, which allows us to compute $\overline{c}_{ij}$ for all edges $(i,j) \in E$. Note that by construction, $\overline{c}_{ij} = 0$ for all $(i,j) \in T$.

Theorem. If $\overline{c}_{ij} \geq 0$ for all $(i,j) \in L$, and $\overline{c}_{ij} \leq 0$ for all $(i,j) \in U$, then (x, λ) is dual-feasible and therefore optimal. Otherwise, find and edge (i,j) that violates these conditions, and observe this edge with the edges in T forms a unique cycle C. Since (i,j) is the only edge in C with non-zero reduced cost, we can decrease the objective by pushing flow along C to increase x_{ij} if $\overline{c}_{ij}$ is negative and decrease x_{ij} if $\overline{c}_{ij}$ is positive.

Let $\underline{B} \subseteq C$ denote the set of edges whose flow is to decrease, and $\overline{B} \subseteq C$ the set of edges whose flow is to increase. If the problem is uncapacitated and $\underline{B} = \emptyset$ or $\overline{B} = \emptyset$, the problem is unbounded. Otherwise, changing the flow by $\min\{\min_{(k,l) \in \underline{B}}\{x_{kl} - \underline{m}_{kl}\}, \min_{(k,l) \in \overline{B}}\{\overline{m}_{kl} - x_{kl}\}\}$ decreases this object as much as possible while maintaining prime feasibility. After this change, there will be an edge $(k,l) \in C$ whose flow is either $\underline{m}_{k,l}$ or $\overline{m}_{k,l}$. If $(k,l) \in T$, we obtain a new BFS with spanning tree $(T \backslash \{(k,l)\}) \cup \{(i,j)\}$. If instead $(k,l) = (i,j)$, we obtain a new BFS where (i,j) has moved from U to L, or vice versa.

To find an initial BFS, set all $\underline{m}_{ij} = 0$ (by introducing flows of forced capacity $\overline{m}'_{ij} = \underline{m}'_{ij} = \underline{m}_{ij}$) Introduce a dummy vertex $d \notin V$ and uncapacitated dummy edges $E' = \{(i,d) | i \in V, b_i \geq 0\} \cup \{(d,i) | i \in V, b_i < 0\}$ with cost equal to $\sum_{(i,j) \in E} c_{ij}$. A dummy edge has positive flow in an optimal solution of the new problem if and only if the original problem is infeasible. A feasible spanning tree solution is obtained by setting $T = E'$, $x_{id} = b_i$ for all $i \in V$ with $b_i > 0$, $d_{di} = -b_i$ for all $i \in V$ with $b_i < 0$, and $x_{ij} = 0$ otherwise.

Theorem. Consider a minimum cost flow problem that is feasible and bounded. If $b_i$is integral for all $i \in V$ and $\underline{m}_{ij}$ and $\overline{m}_{ij}$ are integral for all $(i, j) \in E$, then there exists an integral optimal solution. If $c_{ij}$is integral for all $(i, j) \in E$, then there exists an integral optimal solution to the dual.

Proof. We use no divisions, so integrality is preserved throughout iterations. □

9. Transportation and Assignment Problems

Definition. *We are given a set of supplies producing s_i units of a good and a set of consumers with demands d_j, with $\sum_{i=1}^n s_i = \sum_{j=1}^m d_j$. The cost of transporting from supplier i to $j$is c_{ij}. We formulate the problem of a minimum cost flow on the bipartite network as* $\min \sum_{i=1} \sum_{j=1}^m c_{ij} x_{ij}$ *s.t.* $\sum_{i=1}^n x_{ij} = d_j$, $\sum_{j=1}^m x_{ij} = s_i$, *and* $x_{ij} \geq 0$ *for all* i, j.

Theorem. Every minimum cost flow problem with finite capacities or non-negative costs has an equivalent transportation problem.

Proof. For every vertex $i \in V$, add a sink vertex with demand $\sum_k \overline{m}_{ik} - b_i$. For every edge $(i, j) \in E$, add a source vertex with supply $\overline{m}_{ij}$, an edge to vertex i with cost $c_{ij,i} = 0$, and an edge to vertex j with cost $c_{ij,j} = c_{ij}$.

Then take a feasible flow for the new graph with flows on $(ij, i), (ij, j)$ as $\overline{m}_{ij} - x_{ij}, x_{ij}$. Then the total flow constraint into vertex i is $\sum_{k:(i,j)\in E} \overline{m}_{ik} - b_i$ which is true 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 in the original problem. □

Theorem. For the transportation tableau, form the matrix with squares, λ_i on right, μ_j on the top, with $\lambda_i = \mu_j = c_{ij}$ satisfied for all $(i, j) \in T$, and infer the rest of λ_i, μ_j. Then if all $c_{ij} \geq \lambda_i - \mu_j$, then the flow is optimal. Otherwise, find violating edges, join to spanning tree, and push flow as much as possible along the cycle.

Theorem. In the assignment problem, we have to assign exactly one agent to one job - so the problem is $\min \sum_{i=1}^n \sum_{j=1}^n c_{ij} x_{ij}$ subject to $x_{ij} \in \{0, 1\}$, $\sum_{j=1}^n x_{ij} = 1$, $\sum_{i=1}^n x_{ij} = 1$ - since all solutions to the LP relation are spanning tree solutions (and thus integral), the network simplex method yields an optimal solution of the original problem when applied to the LP relaxation.

10. Maximum Flows and Perfect Matchings

Definition. *In a flow network (V, E) with a single source 1, a single sink n, and finite capacities $\underline{m}_{ij} = 0, \overline{m}_{ij} = C_{ij}$ for all $(i, j) \in E$. The maximum flow problem seeks to maximum δ such that*

$$\sum_{j:(i,j)\in E} x_{ij} - \sum_{j:(j,i)\in E} x_{ji} = \begin{cases} \delta & i = 1 \\ -\delta & i = n \\ 0 & \text{otherwise} \end{cases} \tag{10.1}$$

and $0 \leq x_{ij} \leq C_{ij}$ *for all* $(i, j) \in E$.

Definition. *The capacity of a cut $S \subseteq V$ is given by*

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

Theorem. Let δ be the optimal solution of for a network (V, E) with capacities C_{ij}. Then

$$\delta = \min\{C(S) | S \subseteq V, 1 \in S, n \in V\setminus S\} \tag{10.3}$$

Proof. Show that there exists a cut with capacity δ. Then taking S as 1 and the vertices where there exists an augmenting path to v from 1. Then $n \in V\setminus S$ by optimality (otherwise push flow along to n). Also, $\delta = f(S, V\setminus S) - f(V\setminus S, S) = f(S, V\setminus S) = C(S)$. □

Theorem. The **Ford-Fulkerson algorithm** proceeds to find a maximum flow by pushing along an augmenting path, until such a path cannot be found.

(i) Start with a feasible flow x.
(ii) If there is no augmenting path from 1 to n, stop.
(iii) Else, pick some augmenting path from 1 to n, push a maximum amount of flow along this path without violating any constraints. Then go to step (ii).

Definition. *Consider an alternative formulation of maximum flow as a minimum cost flow,* $\min -x_{n1}$ *subject to* $\sum_{j:(i,j)\in E'} x_{ij} - \sum_{j:(j,i)\in E'} x_{ji} = 0$ *for all* $i \in V$, $0 \leq x_{ij} \leq C_{ij}$ *for all* $(i, j) \in E$, $x_{n1} \geq 0$, *where* $E' = E \cup \{(n, 1)\}$. *The Lagrangian is*

$$L(x, \lambda) = (-1 - \lambda_n + \lambda_1)x_{n1} - \sum_{(i,j)\in E} (\lambda_i - \lambda_j)x_{ij}. \tag{10.4}$$

with bounded minimum when $x_{n1} > 0$ only if $\lambda_1 - \lambda_n = 1$.

Set $\lambda_1 = 1, \lambda_n = 0$. Then $g(\lambda) = \inf_x L(x, \lambda) = -\sum_{(i,j)\in E} \max(\lambda_i - \lambda_j, 0)C_{ij}$. Introducing $d_{ij} \geq \max(\lambda_i - \lambda_j, 0)$, we maximize $g(\lambda)$ by minimizing $\sum d_{ij}C_{ij}$ with $d_{ij} \geq \lambda_i - \lambda_j, d_{ij} \geq 0$, and obtain the dual with an optimal solution where $\lambda_i = \{0, 1\}$ for all $i \in V$, then the set $S = \{i \in V | \lambda_i = 1\}$ is a minimum cut, and the max-flow min-cut follows from strong duality.

Definition. *A matching of a graph is a set of edges that do not share any vertices. A matching is perfect if it covers every vertex - $|M| = \frac{|V|}{2}$. A graph is k-regular if every vertex has degree k.*

Theorem. A bipartite graph $G = (L \uplus R, E)$ with $|L| = |R|$ has a perfect matching if and only if $|N(X)| \geq |X|$ for every $X \subseteq L$, where $N(X) = \{j \in R | i \in X, (i, j) \in E\}$.

Proof. ($\Rightarrow$) is obvious. ($\Leftarrow$) follows by assuming that G has no perfect matching (so the max flow is less than $|L|$), and so choose the cut $S \subseteq L \uplus R \cup \{s\}$ with $C(S) < |L|$ and set $L_S = L \cap S, R_S = R \cap S$, and $L_T = L\setminus S$. Then by finiteness of the min-cut, all neighbors of $x \in L_S$ are in the cut, so $N(L_S) \subseteq R_S$. However, the capacity of the cut comes from $\{s\} \times L_T$, $R_S \times \{t\}$. So $oN(L_S)| \leq |R_s| = C(S) - |L_T| < |L| - L_T| = |L_S|$. □

11. Shortest Paths and Minimum Spanning Trees

Theorem. Let $\lambda_i(k)$ be the length of a shortest path from i to $t$that uses at most k edges. Then $\lambda_t(k) = 0$ for all $k \geq 0$, and $\lambda_i(0) = \infty$, $\lambda_i(k) = \min_{j:(i,j)\in E}(c_{ij} + \lambda_j(k-1))$ for all $i \in V\setminus\{t\}$ and $k \geq 1$. This is the **Bellman-Ford** algorithm.

Theorem. Consider a graph with vertices V and edge lengths $c_{ij} \geq 0$ for all $i, j \in V$. Fix $t \in V$ and let λ_i denote the length of a shortest path from $i \in V$ to t. Let $j \in V\setminus\{t\}$such that $c_{jt} = \min_{i\in V\setminus\{t\}} c_{it}$. Then $\lambda_j = c_{jt}$ and $\lambda_j = \min c_{jt} = \min_{i\in V\setminus\{t\}} \lambda_i$.

Theorem. 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).

12. Semidefinite Programming

Definition. *Let* $\langle C, X\rangle = \operatorname{tr} CX = \sum_{i=1}^n \sum_{j=1}^n c_{ij}x_{ij}$ *for some* $C, X \in \S^n$, *the set of symmetric matrices. A semidefinite program takes the form* $\min \langle C, X\rangle$ *subject to* $\langle A_i, X\rangle = b_i$ *for all* $i \in \{1, \ldots, m\}$, *and* $X \succeq 0$ *(positive semidefinite) where* $C, A_1, \ldots, A_m \in \S^n$ *and* $b \in \mathbb{R}^m$.

Equivalently, $\min c^T x$ *subject to* $B_0 + \sum_{i=1}^k x_i B_i \succeq 0$, *where* $B_i \in S^n$ *and* $c \in \mathbb{R}^k$.

Theorem. The Lagrangian of can be written as

$$L(X, \lambda, Z) = \langle C, X\rangle - \sum_{i=1}^m \lambda_i(\langle A_i, X\rangle - b_i) - \langle Z, X\rangle, \tag{12.1}$$

where the last term takes into account the constraint $X \succeq 0$. Then $g(\lambda, Z) = \inf_{X\in S^n} L(X, \lambda, Z) = \lambda^T b$ if $C - \sum_{i=1}^m \lambda_i A_i - Z = 0$, and $-\infty$ otherwise. Eliminating Z, we obtain the dual of (another semidefinite program), $\max \lambda^T b$ subject to $C - \sum_{i=1}^m \lambda_i A_i \succeq 0$.

Theorem. Consider the primal/dual linear programs $\min\{c^T x | Ax = b, x \geq 0\}$ and $\max\{b^T \lambda | A^T \lambda + z = c, z \geq 0\}$ If we use **primal-dual interior point methods**, we can augment the objective with a barrier function and solve with Newton's method. We augment with $\min c^T x - \mu \sum_{i=1}^n \log x_i | Ax = b$ and $\max\{b^T \lambda + \mu \sum_{j=1}^m \log z_j | A^T \lambda + z = c\}$ for a parameter $\mu > 0$.

Then (x, λ, z) is optimal for the modified primal/dual problems if $Ax = b, x \geq 0$, $A^T \lambda + z = c$, $z \geq 0$, $x_i z_i = \mu$, for all $i = 1, \ldots, n$.

13. Branch and Bound

Definition. *Assume we wish to solve* $\min f(x)$ *s.t.* $x \in X$ *for some feasible region* X*., we use divide and conquer on* X_i *with* $\cup_{i=1}^k X_i = X$*, and solve* $\min_{x\in X} f(x) = \min_{i=1}^k \min_{x\in X_i} f(x)$*.*

If we have a lower and upper bound on the optimal solution - functions l *and* u *such that for all* $X' \subseteq X$*,* $l(X') \leq \min_{x\in X'} f(x) \leq u(X')$ *, then we can efficiently prune our search space.*

The algorithm proceeds as follows:

(i) *Set* $U = \infty, L = \{X\}$*.*
(ii) *Pick* $Y \in L$*, remove from* L*, and split into* $k \geq 2$ *sets* $Y_1, \dots, Y_k$*.*
(iii) *For* $i \in \{1, \dots, k\}$*, compute* $l(Y_i)$*. If this yields* $x \in X$ *with* $l(Y_i) = f(x) < U$ *, set* U *to* $f(x)$*. If* $l(Y_i) < U$ *but no* $x \in X$ *is found, add* y_i *to* L*. If* $L = \emptyset$*, stop, the optimum value is* U*. Otherwise, go back to (ii).*

Theorem. Applied to an integer program, we can obtain a lower bound by solving the LP relaxation, and so eliminate large chunks of the search tree.

Definition. *The traveling salesman problem is the problem of finding a cycle in* G *that visits each vertex exactly once, of minimum overall cost.*

TO proceed, we can encode it as an integer program by introducing $x_{ij} \in \{0,1\}$*, whether the tour traverses edge* i*, and variables* $t_i \in \{0, \dots, n-1\}$ *indicating the position of the vertex* i *in the tour. If* $x_{ij} = 1$*, then* $t_j = t_i + 1$*. If* $x_{ij} = 0$*, then* $t_j \geq t_i - (n-1)$*. We need to constrain that there is exactly one edge entering and one edge leaving -* $\sum_{i=1}^n x_{ij} = 1$ *for* $j = 1, \dots, n$*, and* $\sum_{j=1}^n x_{ij} = 1$ *for* $i = 1, \dots, n$*.*

14. Heuristic Algorithms

Definition. *Assume we want solve the problem* $\min c(x)$ *s.t.* $x \in X$*, and that for any feasible solution* $x \in X$*, the cost* $c(x)$ *and a* ***neighborhood*** $N(x) \subseteq X$ *can be computed efficiently. Then* ***local search*** *proceeds as follows:*

(i) *Find an initial feasible solution* $x \in X$*.*
(ii) *Find a solution* $y \in N(x)$ *such that* $c(y) < c(x)$*.*
(iii) *If there is no solution - stop and return* x*, otherwise set the current solution* x *to* y *and return to (ii).*

Definition. *Simulated annealing considers a neighbor* y *of the current solution* x *and moves to the new solution with probability*

$$p_{xy} = \min(1, \exp(-\frac{c(y) - c(x)}{T})) \tag{14.1}$$

where $T \geq 0$ *is a parameter (the* ***temperature****) that can vary over the time.*

It can be shown that with detailed balance, we have

$$\pi_x = \frac{e^{-\frac{c(x)}{T}}}{\sum_{z\in X} e^{-\frac{c(z)}{T}}} \tag{14.2}$$

for every $x \in X$ *is a distribution and satisfies detailed balance, and so must be the stationary distribution. We have* $\frac{\pi_Y}{1-\pi_Y} \to \infty$ *as* $T \to 0$*, where* $Y \subseteq X$ *is the set of solutions with the minimum cost. We then decrease* T *for the Markov chain to reach the stationary distribution -e.g.* $T = \frac{c}{\log t}$*.*

15. Approximation Algorithms

Definition. *A function* $g : \{0.1\}^\star \to \{0,1\}^\star$ *is an* α *approximation for* o *if for some* $\alpha \geq 1$*, if for all* $x \in P$*,* $o(x, g(x)) \leq \alpha o(x, f(x))$*. In what follows we will be interested in algorithms that compute the function* g *in polynomial time, and refer to such an algorithm as a polynomial-time* α*-approximation algorithm.*

The class APX *are the class of problems where the exists an* α *approximation for some* α*. The class* $PTAS \subseteq APX$ *are problems that possess an* $(1+\epsilon)$ *approximation algorithm for any* $\epsilon > 0$*.*

Theorem. The max-cut problem asks for a cut on an undirected graph that maximizes the number of edges crossing from one side to the other. There exists a $\frac{1}{2}$ approximation with a simple greedy algorithm.

Use a universal hash function to obtain n^2 pairwise-independent samples, we can apply this to randomly cut a variable with probability $\frac{1}{2}$, and so obtain a cut with cost $\mathbb{E}(Q) = \mathbb{E}\Big(\sum_{(i,j)\in E} \mathbb{P}(Q_i \neq Q_j)\Big) = \sum_{(i,j)\in E} \mathbb{E}(\mathbb{P}(Q_i \neq Q_j)) = \frac{|E|}{2}$.

16. Non-Cooperative Games

Definition. *A normal-form game is a tuple* $\Gamma = (N, (A_i)_{i\in N}, (p_i)_{i\in N})$*, where* N *is a finite set of players, and* A_i *is a non-empty and finite set of actions available to* i*, and* $p_i : (\times_{i\in N} A_i) \to \mathbb{R}$ *is a function mapping each combination of actions to a payoff for* i*.*

A two-player game can be represented by $P, Q \in \mathbb{R}^{m\times n}$*, where* p_{ij}, q_{ij} *are the payoffs of players* $1, 2$ *when player* 1 *plays action* i *and player* 2 *plays action* j*.*

The set of possible ***strategies*** *of the two players are given as* $X = \{x \in \mathbb{R}^m_{\geq 0} | \sum_{i=1}^m x_i = 1\}$*,* $Y = \{y \in \mathbb{R}^n_{\geq 0} | \sum_{i=1}^n y_i = 1\}$*. A* ***pure strategy*** *is a strategy that chooses an action with probability one. The expected payoffs from playing* $(x, y) \in (X, Y)$ *are given as* $p(x, y) = x^T P y, q(x, y) = x^T Q y$*.*

For two strategies $x, x' \in X$*,* x *is said to dominate* x' *if for every strategy* $y \in Y$ *of the column player,* $p(x, y) > p(x', y)$*.*

A strategy that maximizes the payoff in the worst case, taken over all the other players strategies, is a ***maximin strategy****, and the payoff is the player's* ***security level****. It is sufficient to maximize the minimum payoff over all pure strategies of the other player (convex combinations), so choosing* x *such that* $\min_{j\in 1,\dots,n} \sum_{i=1}^m x_i p_{ij}$ *is as large as possible. Thus, this is a linear program* $\max v$ *subject to* $\sum_{i=1}^m x_i p_{ij} \geq v$ *for* $j = 1, \dots, n$*,* $\sum_{i=1}^m x_i = 1$*,* $x \geq 0$*.*

Strategy $x \in X$ *of the row player is a* ***best response*** *to strategy* $y \in Y$ *of the column player if for all* $x' \in X$*,* $p(x, y) \geq p(x', y)$*.*

Theorem. For the zero-sum game, where $q_{ij} = -p_{ij}$, we have **von Neumann's theorem**: $\max_{x\in X} \min_{y\in Y} p(x, y) = \min_{y\in Y} \max_{x\in X} p(x, y)$.

Proof. This can be shown by adding a slack variable $z \in \mathbb{R}^n$ with $z \geq 0$ and obtaining the Lagrangian

$$L(v, x, z, w, y) = v + \sum_{j=1}^n y_j (\sum_{i=1}^m x_{ij} - p_{ij} - z_j - v) - w(\sum_{i=1}^m x_i - 1) \tag{16.1}$$

$$= (1 - \sum_{j=1}^n y_j) v + \sum_{i=1}^m (\sum_{j=1}^n p_{ij} y_j - w) x_i - \sum_{j=1}^n y_j z_j + w \tag{16.2}$$

which has finite maximum 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 $y \geq 0$. So the dual is $\min w$ such that $sum_{j=1}^n p_{ij} y_j \leq w$ for $i = 1, \dots, m$, $\sum_{j=1}^n y_j = 1$, and $y \geq 0$. This has solution $\min_{y\in Y} \max_{x\in X} p(x, y)$ as required. This solution is called the **value** of a matrix game with payoff matrix P. □

17. Strategic Equilibrium

Definition. *A pair of strategies* $(x, y) \in X \times Y$ *with* x *a best response to* y *and* y *a best response to* x *is called an* ***equilibrium****.*

Theorem. 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 $\min_{y\in Y} p(x, y') = \max_{x\in X} \min_{y'\in Y} p(x', y')$, and $\max_{x'\in X} p(x', y) = \min_{y'\in Y} \max_{x'\in X} p(x', y')$.

Proof.

$$\min_{y'\in Y} \max_{x'\in X} p(x', y') \leq \max_{x'\in X} p(x', y) \geq p(x, y) \tag{17.1}$$

$$p(x, y) \geq \min_{y'\in Y} p(x, y') \leq \max_{x'\in X} \min_{y'\in Y} p(x', y'), \tag{17.2}$$

and first and last terms are equal from Von Neumann's theorem. □

Theorem. 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.

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.

Theorem (Nash's Theorem)**.** Every bimatrix game has an equilibrium.

Proof. To show this, define X, Y as before, and $X \times Y$ is closed, bounded, and convex. Then for $x \in X$, $y \in Y$, define $s_i(x, y)$ and $t_i(x, y)$ by the additionally payoff the two players could obtain by playing the i-th or j-th pure strategy instead of x or y - so $s_i(x, y) = \max\{0, p(e_i^m, y) - p(x, y)\}$, and $t_j(x, y) = \max\{0, q(x, e_j^n - q(x, y))\}$, and define $f : X \times Y \to X \times Y$ by $f(x, y) = (x', y')$, where

$$x'_i = \frac{x_i + s_i(x, y)}{1 = \sum_{k=1}^m s_k(x, y)} \tag{17.3}$$

$$y'_j = \frac{y_j + t_j(x, y)}{1 + \sum_{k=1}^n t_k(x, y)} \tag{17.4}$$

Note also there must exist $i \in \{1, \dots, m\}$ with $x_i > 0$ and $s_i(x, y) = 0$, as otherwise $p(x, y) = \sum_{k=1}^m x_k p(e_k^m, y) > \sum_{k=1}^m x_k p(x, y) = p(x, y)$. Thus, and as (x, y) is a fixed point,

$$x_i = \frac{x_i + s_i(x, y)}{1 + \sum_{k=1}^m s_k(x, y)} \tag{17.5}$$

and so $\sum_{k=1}^m s_k(x, y) = 0$, so for $k = 1, \dots, m$, $s_k(x, y) = 0$, and so $p(x, y) \geq p(e_k^m, y)$. So $p(x, y) \geq p(x', y)$ for all $x' \in X$. Analogously, $q(x, y) \geq q(x, y')$ for all $y' \in Y$, so (x, y) must be an equilibrium. □

Theorem. Given a bimatrix game, it is NP-complete to decide whether

(i) it has at least two equilibria
(ii) an equilibrium in which the expected payoff of the row player is at least a given amount,
(iii) an equilibrium in which the expected sum of the payoff of the two players is at least a given amount
(iv) an equilibrium with supports of a given minimum size,
(v) an equilibrium whose support includes a given pure strategy,
(vi) or an equilibrium whose support does not include a given pure strategy.

18. Equilibrium Computation

Definition. *Consider a bimatrix game with payoffs $P, Q \in \mathbb{R}^{m \times n}$, and assume WLOG that $P, Q > 0$. Then $M = \{1, \dots, m\}$ and $N = m + 1, \dots, m + n$, and define the sets X, Y of strategies accordingly.*

A pair $(x, y) \in X \times Y$ is an equilibrium if and only if all pure strategies in $S(x)$ are best responses to y and all pure strategies in $S(y)$ are best responses to x - so if for all $i \in M$, $x_i > 0$implies $(Py)_i = \max_{k \in M}(Py)_k$, and for all $j \in N$, $y_j > 0$ implies $(Q^T x)_j = \max_{k \in N}(Q^T x)_k$.

Definition. *A bimatrix game if **non-degenerate** if for every $(x, y) \in X \times Y$, $|S(x)| \geq |S(y)|$ if y is a best-response to x and $|S(y)| \geq |S(x)|$ if x is a best-response to y.*

Definition. *A pair of points is **fully-labelled** if $L(x) \cup L(y) = M \cup N$, where $L(x)$ are the index sets of the constraints that hold with equality - e.g. $\{i \in M | x_i = 0\} \cup \{j \in N | (Q^T x)_i = 1\}$ and vice versa for y.*

Theorem. A pair of extreme points $(x, y) \neq (0, 0)$ is an equilibrium if and only if it is fully-labelled.

Theorem. Every non-degenerate bimatrix game has an odd number of equilibria.

Theorem.
(i) Construct the tableau $Py + r = 1$, $Q^T x + s = 1$.
(ii) Choose a label l to drop.
(iii) Pivot as in simplex algorithm, following the chain of elements to drop.
(iv) When l enters the basis again, we have an equilibrium.

19. Cooperative Games

Definition. *A coalitional game is given by a set $N = \{1, \dots, n\}$ of players, a characteristic function $\nu : 2^N \to \mathbb{R}$ that maps each coalition to is value - the payoff the coalition can obtain by working together. An **imputation** of a game (N, ν) is a vector $x \in \mathbb{R}^n$ such that $x_i \geq \nu(\{i\})$ for all $i \in N$, and $\sum_{i=1}^n x_i = \nu(N)$. The first condition **individual rationality**, requires each player obtains the same payoff it would be able to obtain on it's own. The second is **economic efficiency** - no payoff is wasted.*

Definition. *An imputation x is in the core of game (N, ν) if $\sum_{i \in S} x_i \geq v(S)$ for all $S \subseteq N$.*

Definition. *A function $\lambda : 2^N \to [0, 1]$ is **balanced** if for every player the weights of all coalitions containing that player sum to 1 - so, for all $i \in N$, $\sum_{S \subseteq N \setminus \{i\}} \lambda(S \cup \{i\}) = 1$. A game (N, ν) is balanced if for every balanced function λ, $\sum_{S \subseteq N} \lambda(S)\nu(S) \leq \nu(N)$.*

Theorem. A game has a non-empty core if and only if it is balanced.

Proof. The core is non-empty if and only if $\min_{i \in N} x_i$ such that $\sum_{i \in S} x_i \geq v(S)$ for all $S \subseteq N$ has an optimal solution with value $v(N)$. The dual is $\max_{S \subseteq N} \lambda(S)v(S)$ such that $\sum_{S \subseteq N, i \in S} \lambda(S) = 1$ for all $i \in N$, $\lambda(S) \geq 0$ for all $S \subseteq N$. Note that λ is dual-feasible if and only if it is balanced. As primal and dual are both feasible, by strong duality the optimal objectives are the same. □

Definition. *The **excess** $e(S, x)$ of coalition $S \subseteq N$ for imputation x is the gain from leaving the grand coalition - $e(S, x) = \nu(S) - \sum_{i \in S} x_i$.*

For a given imputation x, let $S_1^x, \dots, S_{2^n-1}^x$ be an ordering of the coalitions such that $e(S_k^x, x) \geq e(S_{k+1}^x, x)$ for $k = 1, \dots, 2^n - 2$, and let $E(x) \in R^{2^n - 1}$ be the vector given by $E_k(x) = e(S_k^x, x)$. We say that $E(x)$ is lexicographically smaller than $E(y)$ if there exists $i \in \{1, \dots, 2^n - 1\}$ such that $E_k(x) = E_k(y)$ for $k = 1, \dots, i = 1$ and $E_i(x) < E_i(y)$. The nucleolus is then defined as the set of imputations x for which $E(x)$ is lexicographically minimal.

Theorem. The nucleolus of any coalitional game is a singleton.

Proof. □

Definition. *Call player $i \in N$ a **dummy** if its contribution to every coalition is exactly its value - if $\nu(S \cup \{i\}) = \nu(S) + \nu(\{i\})$ for all $S \subseteq N \setminus \{i\}$. Call two players i, j interchangeable if they contribute the same to every coalition - so $\nu(S \cup \{i\}) = \nu(S \cup \{j\})$ for all $S \subseteq N \setminus \{i, j\}$. Let a **solution** be a function $\phi : \mathbb{R}^{2^n} \to \mathbb{R}^n$ that maps every characteristic function ν to an imputation $\phi(\nu)$. Solutions ϕ is said to satisfy*

*(i) **dummies** if $\phi_i(v) = v(\{i\})$ whenever i is a dummy,*
*(ii) **symmetry** if $\phi_i(v) = \phi_j(v)$ whenever i, j are interchangeable, and **additivity** if $\phi(v + w) = \phi(v) + \phi(w)$.*

Theorem. The **Shapley value**, given by

$$\phi_i(v) = \sum_{S \subseteq N \setminus \{i\}} \frac{|S|!(|N| - |S| - 1)!}{|N|!}(\nu(S \cup \{i\}) - \nu(S)) \tag{19.1}$$

is the unique solution that satisfies dummies, symmetry, and additivity.

20. Bargaining

Definition. *A two-player bargaining problem is a pair F, d where $F \subseteq \mathbb{R}^n$ 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. A **bargaining solution** is a function that assigns to every bargaining problem (F, d) a unique element of F.*

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 $F = \mathrm{con}\{(p_{ij}, q_{ij}) | i \in M, j \in N\}$, $d_1 = \max_{x \in X} \min_{y \in Y} p(x, y)$, and $d_2 = \max_{y \in Y} \min_{x \in X} q(x, y)$.

Theorem. For a given bargaining problem, Nash proposes $\max(v_1 - d_1)(v_2 - d_2)$ such that $v \in F$, $v \geq de$.

Proof. To solve for a 2×2 bimatrix game,

(i) Find d_1, d_2, the minimax values for each player.
(ii) Plot the outcomes for each strategy pair on $\mathbb{R}^2$.
(iii) Find the line where the optimal solution must exist, and solve the one-dimensional optimization.

□

Definition. *A bargaining solution f is*

*(i) **Pareto efficient** if $f(F, d)$ is not Pareto dominated in F for any bargaining problem (F, d)*
*(ii) **symmetric** if $(f(F, d))_1 = (f(F, d))_2$ for every bargaining problem (F, d) such that $(y, x) \in F$ whenever $(x, y) \in F$ and $d_1 = d_2$.*
*(iii) **Invariant under positive affine transforms** if $f(F, d) = \alpha \circ f(F, d) + \beta$ for $\alpha, \beta \in \mathbb{R}^2$ with $\alpha > 0$ and any two bargaining problems (F, d) and (F', d') with $F' = \alpha F + \beta$, $d' = \alpha d + \beta$,*
*(iv) **independence or irrelevant alternatives** if $f(F, d) = f(F', d)$ for any two bargaining problems (F, d) and (F', d) such that $F' \subseteq F$ with $d \in F'$ and $f(F, d) \in F'$.*

Theorem. Nash's bargaining solution is the unique bargaining solution that is Pareto efficient, symmetric, invariant under positive affine transformations, and independent or irrelevant alternatives.

Proof. (⇒) is obvious. For (⇐), consider a bargaining solution that satisfies the axioms, and fix F, d. Let z be the Nash solution, and map the problem to F' affinely, taking z to $(\frac{1}{2}, \frac{1}{2})$, d to $(0, 0)$. Then show $f(F', 0) = (\frac{1}{2}, \frac{1}{2})$. First, show $v_1 + v_2 \leq 1$ - which follows by taking a convex combination of v with $v_1 + v_2 > 1$, and show that a small perturbation increases the objective function, contradicting optimality.

Taking the closure under symmetry, by Pareto optimality and symmetry, must have $f(F', 0) = (\frac{1}{2}, \frac{1}{2})$. □

21. Stable Matchings

Definition. *Consider a set S of students and a set A of potential advisors. Each student $i \in S$ has a strict linear order $\succ_i \subseteq A \times a$, each advisor $j \in A$ has a strict linear order $\succ_j \subseteq S \times S$. A matching is a function $\mu : S \cup A \to S \cup A$ such that $\mu(\mu(i)) = i$, $\mu(i) \in A$ for all $i \in S$, and $\mu(j) \in S$ for all $j \in A$.*

*A pair $(i, j) \in S \times A$ is a **blocking pair** for matching μ if i, j would rather be matched to each other than respective partners in μ - so $j \succ_i \mu(i)$ and $i \succ_j \mu(j)$. A matching is called **stable** if it does not have any blocking pairs.*

Theorem. Consider the **deferred acceptance procedure**,

(i) Let each student $i \in S$ propose to the advisor it ranks highest.
(ii) Match advisor $j \in A$ tentatively to the highest-ranked amongst the students that have proposed to it, if any, and reject the others for good.
(iii) Let each student that has been rejected but has not been rejected by all advisors propose to the next advisor. IF there are no such students, stop and return the matched pairs. Otherwise, return to (ii).

This procedure always terminates, and yields a stable matching when it does.

Proof. Any blocking pair must have been proposed to by a preferable student before the one it was eventually matched with. □

Theorem. Call $j \in A$ achievable for $i \in S$ if there exists a stable matching μ such that $\mu(i) = j$. A stable matching μ is called **student-optimal** if for all $i \in S$, $\mu(i)$ is most preferred among the advisors achievable for i.

Students-propose deferred acceptance yields a student-optimal stable matching.

Theorem. A matching μ is advisor pessimal if for all $j \in A$, $\mu(j)$ is least preferred among the students achievable for j.

Every student-optimal stable matching is advisor-pessimal.

Theorem. Fix a stable matching μ and let $x_{ij} = 1$ for $i \in S$ and $j \in A$ if $\mu(i) = j$, and $x_{ij} = 0$ otherwise. Then the constraints $\sum_{j \in A} x_{ij} = 1$ for all $i \in S$, $\sum_{i \in S} x_{ij} = 1$ for all $j \in A$, $x_{ij} + \sum_{k: j \succ_i k} x_{ik} + \sum_{k: i \succ_j k} x_{kj} \leq 1$ for all $i \in S$, $j \in A$, $x_{ij} \geq 0$ for all $i \in S$, $j \in A$.

Then P is the polytope described by the above constraints, and $P \neq \emptyset$. Moreover, a vector is an extreme point of P if and only if it is a stable matching.

22. Social Choice

Definition. *Let $N = \{1, \ldots, n\}$ be a set of agents, or **voters**, and $A = \{1, \ldots, m\}$ a set of alternatives. Assume $n, m \geq 2$ and finite. Assume each voter $i \in N$ has a strict linear order $\succ_i \in L(A)$, and the goal is to map the profile of individual preference orders to a social preference order. This is achieved by means of a social welfare function $f : L(A)^n \to L(A)$.*

A social welfare function is

*(i) **anonymous** if for every permutation $\pi \in S_n$ of the voters and preference profiles $\succ, \succ' \in L(A)^n$ such that $a \succ_i b$ if and only if $a \succ'_{\pi(i)} b$ for all $a, b \in A$, it holds that $f(\succ) = f(\succ')$. all*
*(ii) **neutral** if for every permutation $\pi \in S_m$ of the alternatives and all preference profiles $\succ, \succ' \in L(A)^n$ such that $a \succ_i b$ if and only if $\pi(a) \succ'_i \pi(b)$ for all $a, b \in A$, it holds that $a f(\pi)\ b$ if and only if $\pi(a) f(\pi') \pi(b)$ for all $a, b \in A$,*
*(iii) **monotone** if for all $\pi, \pi' \in L(A)^n$ and $a, b \in A$, $a f(\pi) b$ and $\{i \in N | a \succ_i b\} \subseteq \{i \in N | a \succ'_i b\}$ implies $a f(\succ_i) b$.*

Anonymity requires that voters are treated equally, symmetry requires that alternatives are treaded equally, and monotonicity requires that an alternative cannot become less preferred socially when it becomes more preferred by individuals. When the number of voters are odd, these intuitive fairness and welfare properties precisely characterize the majority rule.

Theorem. Consider an SWF $f : L(A)^n \to L(A)$, where $|A| = 2$ and n is odd. Then f is the majority rule if and only if it is anonymous, neutral, and monotone.

Definition. *A SWF $f : L(A)^{|N|} \to L(A)$ is*

*(i) **Pareto optimal** if for all $a, b \in A$ and every $\succ \in L(A)^n$ such that $a \succ_i b$ for all $i \in N$, it holds that $a \succ' b$, where $\succ' = f(\succ)$.*
*(ii) **independence of irrelevant alternatives** if for all $a, b \in A$ and all $\succ, \succ' \in L(A)^n$ such that $\succ_i \cap (\{a, b\} \times \{a, b\}) = \pi' \cap (\{a, b\} \times \{a, b\})$ for all $i \in N$, it holds that $f(\succ) \cap (\{a, b\} \times \{a, b\}) = f(\succ') \cap (\{a, b\} \times \{a, b\})$.*
*(iii) **dictatorial** if there exists $i \in N$ such that for all $\succ \in L(A)^n$, $f(\succ) = \succ_i$.*

Pareto optimality requires that alternative a is socially preferred over alternative b when every voter prefers a over b. Independence of irrelevant alternatives requires that the social preference with respect to a and b only depends on individual preferences with respect to a and b, but not on those with respect to other alternative.s Finally, an SWF is dictatorial if the social preference order is determined by a single voter.

It turns out that dictatorships at the only SWFs for three or more alternatives that are Pareto optimal and IIA.

Theorem. Consider an SWF $f : L(A)^n \to L(A)$, where $|A| \geq 3$. If f is Pareto optimal and IIA, then f is dictatorial.

23. Mechanism Design

Definition. *An SCF f is **manipulable** if there exist $i \in N$, $\succ \in L(A)^n$, and $\succ'_i \in L(A)$ such that $f((\succ_{-i}, \succ'_i) \succ_i f(\succ)$. SCF f is called **strategy-proof** if it is not manipulable.*

*- An SCF is **dictatorial** if there exists $i \in N$ such that for all $\succ \in L(A)^n$ and $a \in A \backslash \{f(\succ)\}, f(\succ) \succ_i a$. AN SCF f is **surjective** if for all $a \in A$, there exists $\succ \in L(A)^n$ such that $f(\succ) = a$.*

Theorem. Consider an SCF $f : L(A)^n \to L(A)$, where $|A| \geq 3$. If f is surjective and strategy-proof, then it is dictatorial.

Proo

Definition. *A mechanism design problem is a set A of alternatives and a set $N = \{1, \ldots, n\}$ of agents, each with a set Θ_i of possible types and a utility function $u_i : A \times \Theta_i \to \mathbb{R}$. A mechanism is a message space Σ_i for agent i and an outcome function $g : \times_{i \in N} \Sigma_i \to A$. A mechanism is called **direct** if the agents directly report their type to the mechanism - so$\Sigma_i = \Theta_i$ for all $i \in N$.*

*Mechanism $M = ((\Sigma_i)_{i \in N}, g)$ is said to **implement** SCF $f : \times_{i \in N} \Theta_i \to A$ (in weakly dominant strategies) if there exist functions $s_i : \Theta_i \to \Sigma_i$ for all $i \in N$ such that for every $\theta \in \Theta$, $g(s_1(\theta_1), \ldots, s_n(\theta_n)) = f(\theta)$, and for all $i \in N$, $\theta_i \in \Theta_i$ and for $\sigma \in \Sigma$, $u_i(g(s_i(\theta_i), \sigma_{-i}), \theta_i) \geq u_i(g(\sigma), \theta_i)$.*

*An SCF is called **implementable** if it is implemented by some mechanism.*

*A direct mechanism M is called **strategy-proof**, or **dominant strategy incentive compatible**, if for all $i \in N$, $\theta \in \Theta$, and $\theta'_i \in \Theta_i$, $u_i(g(\theta), \theta_i) \geq u_i(g(\theta'_i, \theta_{-i}), \theta_i)$.*

Theorem. A social choice function is implementable if and only if it is implemented in the truthful mechanism of a strategy-proof direct mechanism.

24. Mechanisms with Payments

Definition. *A mechanism is a pair (f, p) of a social choice function $\Theta \to A$ and a payment function $p : \Theta \to \mathbb{R}^n$. The utility of an agent i is $u_i(\theta', \theta_i) = v(f(\theta'), \theta_i) - p(\theta')$, where θ' is a profile of types revealed to the mechanism, θ_i is the true type of agent i, $v_i : A \times \Theta_i \to \mathbb{R}$ is valuation function over alternatives, and $p_i(\theta') = (p(\theta'))_i$.*

*The **social welfare** of an alternative $A \in A$ is $\sum_{i=1}^n v_i(a, \theta_i)$.*

Definition. *A mechanism f, p is a **Vickrey-Clark-Groves** mechanism if $f(\theta) \in \arg\max_{a \in A} \sum_{i=1}^n v_i(a, \theta_i)$, and $p_i(\theta) = h_i(\theta_{-i}) - \sum_{j \in N \backslash \{i\}} v_j(f(\theta), \theta_j)$ for all $i \in N$ where $h_i : \Theta_{-i} \to \mathbb{R}$ is a function that depends on the types of all agents except for i. The crucial component is the second term in p_i, the social welfare for all agents but i.*

Theorem. VCG mechanisms are strategy-proof.

Definition. *Mechanism (f, p) makes **no positive transfers** if $p_i(\theta) \geq 0$ for all $i \in N$ and $\theta \in \Theta$, and is **ex-post individually rational** if it always yields non-negative utilities for all agents - so $v_i(f(\theta)) - p_i(\theta) \geq 0$ for all $i \in N$ and $\theta \in \Theta$.*

*The **Clark pivot rule** is setting $h_i(\theta_{-i}) = \max_{a \in A} \sum_{j \in N \backslash \{i\}} v_j(a, \theta_j)$ such that the payment of agent i becomes $p_i(\theta) = \max_{a \in A} \sum_{j \in N \backslash \{i\}} v_j(a, \theta_j) - \sum_{j \in N \backslash \{i\}} v_j(f(\theta))$.*

Intuitively, the latter amount is equal to the externality i imposes on the other agents - the difference between their social welfare with and *without i's participation. The payment makes the agent internalize this externality.*

Theorem. A VCG mechanism with the Clarke pivot rule makes no positive transfers. If $v_i(a, \theta_i) \geq 0$ for all $i \in N$, $\theta_i \in \Theta_i$, and $a \in A$, it is also individually rational.

Theorem. A mechanism (f, p) is strategy-proof if and only if for every $i \in N$ and $\theta \in \Theta$, $p_i(\theta) = t_i(\theta_{-i}, f(\theta))$ and $f(\theta) = \arg\max_{a \in A(\theta_{-i})}\{v_i(\theta_i, a) - t_i(\theta_{-i}, a)\}$ where $t_i : \Theta_{-i} \times A \to \mathbb{R}$ is a **price function** and $A(\theta_{-i}) = \{f(\theta_i, \theta_{-i}) | \theta_i \in \Theta_i\}$ is the range of f given that the reported types of all agents but i are fixed to θ_{-i}.

Definition. *An SCF f satisfies **weak monotonicity** if for all $\theta \in \Theta$, $i \in N$, and $\theta_i' \in \Theta_i$, $f(\theta) = a \neq b = f(\theta_i, \theta_{-i})$ implies that $v_i(a, \theta_i) - v_i(b, \theta_i) \geq v_i(a, \theta_i') - v_i(b, \theta_i')$.*

Theorem. If mechanism (f, p) is strategy-proof, then f satisfies weak monotonicity. If SCF f satisfies weak monotonicity and for each $i \in N$,$\{(v_i(a, \theta_i))_{a \in A} | \theta_i \in \Theta_i\} \subseteq \mathbb{R}^{|A|}$ is a convex set, then there exists a payment function $p : \Theta \to \mathbb{R}^n$ such that (f, p) is strategy-proof.

Definition. *SCF f is called an **affine maximizer** if there exist $A' \subseteq A$, $w_i \in \mathbb{R}_{>0}$ for $i \in N$, and $c_a \in \mathbb{R}$ for $a \in A'$ such that for every $\theta \in \Theta$, $f(\theta) \in \arg\max_{a \in A'}(c_a + \sum_{i=1}^{n} w_i v_i(a, \theta_i))$.*

Theorem. Let f be an affine maximizer, and for each $i \in N$ and $\theta \in \Theta$, let $p_i(\theta) = h_i(\theta_{-i}) - \sum_{j \in N \setminus \{i\}} \frac{w_j}{w_i} v_j(f(\theta), \theta_j) - \frac{c_{f(\theta)}}{w_i}$ where $h_i : \Theta_{-i} \to \mathbb{R}$. Then (f, p) is strategy-proof.

Theorem. Let $|A| \geq 3$ and $\{(v_i(a, \theta_i))_{a \in A} | \theta \in \Theta\} = \mathbb{R}^{|A|}$ for every $i \in N$. Let $f : \theta \to A$ be a surjective SCF, and $p : \theta \to \mathbb{R}^n$ a payment function. If (f, p) is strategy-proof, then f is an affine maximizer.

References