Part IB — Optimisation
Definitions

Based on lectures by F. A. Fischer
Notes taken by Dexter Chua

Easter 2015

Lagrangian methods
General formulation of constrained problems; the Lagrangian sufficiency theorem. Interpretation of Lagrange multipliers as shadow prices. Examples. [2]

Linear programming in the nondegenerate case
Convexity of feasible region; sufficiency of extreme points. Standardization of problems, slack variables, equivalence of extreme points and basic solutions. The primal simplex algorithm, artificial variables, the two-phase method. Practical use of the algorithm; the tableau. Examples. The dual linear problem, duality theorem in a standardized case, complementary slackness, dual variables and their interpretation as shadow prices. Relationship of the primal simplex algorithm to dual problem. Two person zero-sum games. [6]

Network problems
The Ford-Fulkerson algorithm and the max-flow min-cut theorems in the rational case. Network flows with costs, the transportation algorithm, relationship of dual variables with nodes. Examples. Conditions for optimality in more general networks; *the simplex-on-a-graph algorithm*. [3]

Practice and applications
Efficiency of algorithms. The formulation of simple practical and combinatorial problems as linear programming or network problems. [1]

Contents

1 Introduction and preliminaries

1.1 Constrained optimization

Definition (Constrained optimization)**.** The general problem is of *constrained optimization* is

$$\text{minimize } f(x) \text{ subject to } h(x) = b,\ x \in X$$

where $x \in \mathbb{R}^n$ is the *vector of decision variables*, $f : \mathbb{R}^n \to \mathbb{R}$ is the *objective function*, $h : \mathbb{R}^n \to \mathbb{R}^m$ and $b \in \mathbb{R}^m$ are the *functional constraints*, and $X \subseteq \mathbb{R}^n$ is the *regional constraint*.

Definition (General and standard form)**.** The *general form* of a linear program is

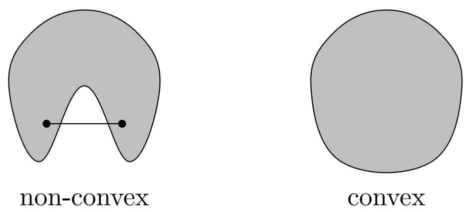
$$\text{minimize } c^T x \text{ subject to } Ax \geq b,\ x \geq 0$$

The *standard form* is

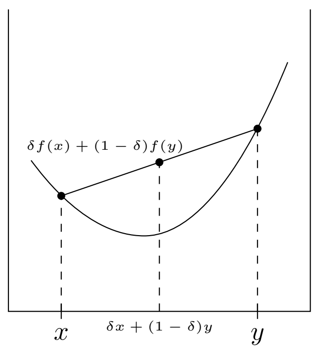
$$\text{minimize } c^T x \text{ subject to } Ax = b,\ x \geq 0.$$

1.2 Review of unconstrained optimization

Definition (Convex region)**.** A region $S \subseteq \mathbb{R}^n$ is *convex* iff for all $\delta \in [0, 1]$, $x, y \in S$, we have $\delta x + (1 - \delta)y \in S$. Alternatively, If you take two points, the line joining them lies completely within the region.



Definition (Convex function)**.** A function $f : S \to \mathbb{R}$ is *convex* if S is convex, and for all $x, y \in S$, $\delta \in [0, 1]$, we have $\delta f(x) + (1 - \delta)f(y) \geq f(\delta x + (1 - \delta)y)$.



A function is *concave* if $-f$ is convex. Note that a function can be neither concave nor convex.

Definition (Positive-semidefinite)**.** A matrix H is *positive semi-definite* if $v^T H v \geq 0$ for all $v \in \mathbb{R}^n$.

2 The method of Lagrange multipliers

Definition (Lagrangian)**.** The *Lagrangian* of a constraint (P) is defined as

$$L(x, \lambda) = f(x) - \lambda^T(h(x) - b).$$

for $\lambda \in \mathbb{R}^m$. λ is known as the *Lagrange multiplier*.

2.1 Complementary Slackness

2.2 Shadow prices

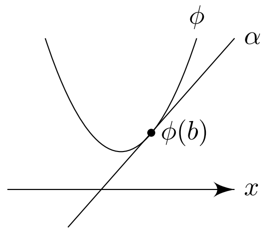
2.3 Lagrange duality

Definition (Strong duality)**.** (P) and (D) are said to satisfy *strong duality* if

$$\sup_{\lambda \in Y} g(\lambda) = \inf_{x \in X(b)} f(x).$$

2.4 Supporting hyperplanes and convexity

Definition (Supporting hyperplane)**.** A hyperplane $\alpha : \mathbb{R}^m \to \mathbb{R}$ is *supporting* to ϕ at b if α intersects ϕ at b and $\phi(c) \geq \alpha(c)$ for all c.



3 Solutions of linear programs

3.1 Linear programs

3.2 Basic solutions

Definition (Extreme point)**.** An *extreme point* $x \in S$ of a convex set S is a point that cannot be written as a convex combination of two distinct points in S, i.e. if $y, z \in S$ and $\delta \in (0, 1)$ satisfy

$$x = \delta y + (1 - \delta)z,$$

then $x = y = z$.

Definition (Basic solution and basis)**.** A solution $x \in \mathbb{R}^n$ is *basic* if it has at most m non-zero entries (out of n), i.e. if there exists a set $B \subseteq \{1, \cdots, n\}$ with $|B| = m$ such that $x_i = 0$ if $i \not\in B$. In this case, B is called the *basis*, and x_i are the *basic variables* if $i \in B$.

Definition (Non-degenerate solutions)**.** A basic solution is *non-degenerate* if it has exactly m non-zero entries.

Definition (Basic feasible solution)**.** A basic solution x is *feasible* if it satisfies $x \geq 0$.

3.3 Extreme points and optimal solutions

3.4 Linear programming duality

3.5 Simplex method

3.5.1 The simplex tableau

3.5.2 Using the Tableau

3.6 The two-phase simplex method

4 Non-cooperative games

4.1 Games and Solutions

Definition (Bimatrix game)**.** A two-player game, or *bimatrix game*, is given by two matrices $P, Q \in \mathbb{R}^{m\times n}$. Player 1, or the *row player*, chooses a row $i \in \{1, \cdots, m\}$, while player 2, the *column player*, chooses a column $j \in \{1, \cdots, n\}$. These are selected without knowledge of the other player's decisions. The two players then get payoffs P_{ij} and Q_{ij} respectively.

Definition (Strategy)**.** Players are allowed to play randomly. The set of *strategies* the row player can have is

$$X = \{x \in \mathbb{R}^m : x \geq 0, \sum x_i = 1\}$$

and the column player has strategies

$$Y = \{y \in \mathbb{R}^n : y \geq 0, \sum y_i = 1\}$$

Each vector corresponds to the probabilities of selecting each row or column.

A strategy profile $(x, y) \in X \times Y$ induces a lottery, and we write $p(x, y) = x^T P y$ for the expected payoff of the row player.

If $x_i = 1$ for some i, i.e. we always pick i, we call x a *pure strategy.*

Definition (Best response and equilibrium)**.** A strategy $x \in X$ is a *best response* to $y \in Y$ if for all $x' \in X$

$$p(x, y) \geq p(x', y)$$

A pair (x, y) is an *equilibrium* if x is the best response against y and y is a best response against x.

4.2 The minimax theorem

Definition (Zero-sum game)**.** A bimatrix game is a *zero-sum game*, or matrix game, if $q_{ij} = -p_{ij}$ for all i, j, i.e. the total payoff is always 0.

Definition (Value)**.** The *value* of the matrix game with payoff matrix P is

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

5 Network problems

5.1 Definitions

Definition (Directed graph/network)**.** A *directed graph* or *network* is a pair $G = (V, E)$, where V is the set of vertices and $E \subseteq V \times V$ is the set of edges. If $(u, v) \in E$, we say there is an edge from u to v.

Definition (Degree)**.** The degree of a vertex $u \in V$ is the number of $v \in V$ such that $(u, v) \in E$ or $(v, u) \in E$.

Definition (Walk)**.** An *walk* from $u \in V$ to $v \in V$ is a sequence of vertices $u = v_1, \cdots, v_k = v$ such that $(v_i, v_{i+1}) \in E$ for all i. An *undirected walk* allows $(v_i, v_{i+1}) \in E$ or $(v_{i+1}, v) \in E$, i.e. we are allowed to walk backwards.

Definition (Path)**.** A path is a walk where $v_1, \cdots, v_k$ are pairwise distinct.

Definition (Cycle)**.** A cycle is a walk where $v_1, \cdots, v_{k-1}$ are pairwise distinct and $v_1 = v_k$.

Definition (Connected graph)**.** A graph is *connected* if for any pair of vertices, there is an undirected path between them.

Definition (Tree)**.** A *tree* is a connected graph without (undirected) cycles.

Definition (Spanning tree)**.** The *spanning tree* of a graph $G = (V, E)$ is a tree (V', E') with $V' = V$ and $E' \subseteq E$.

5.2 Minimum-cost flow problem

5.3 The transportation problem

5.4 The maximum flow problem

Definition (Cut)**.** Suppose $G = (V, E)$ with capacities C_{ij} for $(i, j) \in E$. A *cut* of G is a partition of V into two sets.

For $S \subseteq V$, the *capacity* of the cut $(S, V \setminus S)$ is

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

All this clumsy notation says is that we add up the capacities of all edges from S to $V \setminus S$.