Nonlinear Optimization:
Discrete optimization

INSEAD, Spring 2006

Jean-Philippe Vert
Ecole des Mines de Paris

Jean-Philippe.Vert@mines.org
Motivations

\[
\begin{align*}
\text{minimize} & \quad f(x) \\
\text{subject to} & \quad x \in X, \\
& \quad g_j(x) \leq 0, \quad j = 1, \ldots, r,
\end{align*}
\]

where \( X \) is a \emph{finite} set (e.g., 0–1-valued vectors).

- Many problems involve integer constraints
- Applications in scheduling, resource allocation, engineering design...
- Diverse methodology for their solution, but an important subset of this methodology relies on the solution of \emph{continuous optimization subproblems}, as well as on \emph{duality}.
Outline

- Network optimization and unimodularity
- Examples of nonunimodular problems
- Branch-and-bound
- Lagrange relaxation
Network optimization and unimodularity
Network optimization

Let a directed graph with set of nodes $\mathcal{N}$ and set of arcs $(i, j) \in \mathcal{A}$.

An integer-constrained network optimization problem is:

$$\begin{align*}
\text{minimize} & \quad \sum_{(i,j) \in \mathcal{A}} a_{ij} x_{ij} \\
\text{subject to} & \quad \sum_{\{j \mid (i,j) \in \mathcal{A}\}} x_{ij} - \sum_{\{j \mid (j,i) \in \mathcal{A}\}} x_{ji} = s_i, \forall i \in \mathcal{N} \\
& \quad b_{ij} \leq x_{ij} \leq c_{ij}, \forall (i, j) \in \mathcal{A} \\
& \quad x_{ij} \in \mathbb{N}.
\end{align*}$$
Example: transportation optimization

- Nodes are *locations* (cities, warehouses, or factories) where a certain product is produced or consumed.
- Arcs are *transportation links* between the locations.
- $a_{i,j}$ is the *transportation cost* per unit transported between locations $i$ and $j$.
- The problem is to move the product from the production points to the consumption points at *minimum costs* while observing the capacity constraints of the transportation links.
- $s_i$ is the *supply* provided by node $i$ to the outside world. It is equal to the difference between the total flows coming in and out.
Example: shortest path

Given a starting node $s$ and a destination node $t$, let the “supply”:

$$s_i = \begin{cases} 
1 & \text{if } i = s, \\
-1 & \text{if } i = t, \\
0 & \text{otherwise.}
\end{cases}$$

and let the constraint $x_{ij} \in \{0, 1\}$.

Let $a_{ij}$ be the distance between locations $i$ and $j$.

Any feasible solution corresponds to a path between $s$ and $t$.

This problem is therefore that of finding the shortest path between $s$ and $t$. 

Nonlinear optimization ©2006 Jean-Philippe Vert, (Jean-Philippe.Vert@mines.org) – p.7/30
Relaxing constraints

- The most important property of the network optimization problem is that *the integer constraint can be neglected*
- The *relaxed problem* (a LP without integer constraint) has the same optimal value as the integer-constrained original
- Great significance since the relaxed problem can be solved using efficient linear (not integer) programming algorithms.
Unimodularity property

- A square matrix $A$ with integer components is **unimodular** if its determinant is 0, 1 or $-1$.

- If $A$ is invertible and unimodular, the inverse matrix $A^{-1}$ has integer components. Hence the solution $x$ of the system $Ax = b$ is integer for every integer vector $b$.

- A rectangular matrix with integer components is called **totally unimodular** if each of its square submatrices is unimodular.

- Key fact: A polyhedron $\{x | Ex = d, b \leq x \leq c\}$ has integer extreme points if $E$ is totally unimodular and $b, c$ and $d$ have integer components.
Unimodularity of network optimization

\[
\begin{align*}
\text{minimize} & \quad a^\top x \\
\text{subject to} & \quad Ex = d, \quad d \leq x \leq c.
\end{align*}
\]

The fundamental theorem of linear programming states that the solution to a linear program is an extreme point of the polyhedron of feasible points.

The constraint matrix for the network optimization problem is the \textit{arc incidence matrix} for the underlying graph. We can show that it is totally unimodular (by induction, left as exercise)

Therefore the problem is unimodular: the solution of the LP has integer values!

However, unimodularity is an exceptional property...
Example: shortest path as a LP

\[
\begin{align*}
\text{minimize} & \quad x_1 + x_2 + x_3 + x_4 + x_5 \\
\text{subject to} & \quad x_1 - x_2 = 1 , \\
& \quad x_3 - x_1 - x_5 = 0 , \\
& \quad x_2 + x_4 - x_3 = 0 , \\
& \quad x_5 - x_4 = -1 , \\
& \quad 0 \leq x_i \leq 1 , \; i = 1, \ldots, 5 .
\end{align*}
\]

See script \texttt{shortestpath.m}
Examples of nonunimodular problems
Generalized assignment problem

- $m$ jobs must be assigned to $n$ machines
- If job $i$ is performed at machine $j$ it costs $a_{ij}$ and requires $t_{ij}$ time units.
- Each job must be performed in its entirety at a single machine
- Goal: find a minimum cost assignment of the jobs to the machines, given the total available time $T_j$ at machine $j$. 
Formalization

- Let $x_{ij} \in \{0, 1\}$ indicate whether job $i$ is assigned to machine $j$.
- Each job must be assigned to some machine:
  \[ \sum_{j=1}^{n} x_{ij} = 1. \]
- Limit in the total working time of machine $j$:
  \[ \sum_{i=1}^{m} x_{ij} t_{ij} \leq T_j \]
- Total cost is
  \[ \sum_{i=1}^{m} \sum_{j=1}^{n} x_{ij} a_{ij} \]
minimize \[ \sum_{i=1}^{m} \sum_{j=1}^{n} x_{ij}a_{ij} \]

subject to \[ \sum_{j=1}^{n} x_{ij} = 1 , \quad i = 1, \ldots, m \]
\[ \sum_{i=1}^{m} x_{ij}t_{ij} \leq T_j , \quad j = 1, \ldots, n , \]
\[ x_{ij} \in \{0, 1\} , \quad i = 1, \ldots, m, \quad j = 1, \ldots, n . \]
Other problems

- **Facility location problem**: select a subset of locations from a given candidate set, and place in each of these locations a facility that will serve the needs of certain clients up to a given capacity bound (minimize the cost).

- **Traveling salesman problem**: find a minimum cost tour that visits each of $N$ given cities exactly once and returns to the starting city.

- **Separable resource allocation problems**: optimally produce a given amount of product using $n$ production units.

- (see Bertsekas sec. 5.5)
Approaches to discrete programming

- Enumeration of the finite set of all feasible solutions, and comparison to obtain an optimal solution (rarely practical)

- Constraint relaxation and heuristic rounding:
  - neglect the integer constraints
  - solve the problem using linear/nonlinear programming methods
  - if a noninteger solution is obtained, round it to integer using a heuristic
  - sometimes, with favorable structure, clever problem formulation, and good heuristic, this works remarkably well.
Branch-and-bound
Motivations

- Combines the preceding two approaches (enumeration and constraint relaxation)

- It uses constraint relaxation and solution of noninteger problems to obtain certain lower bounds that are used to discard large portions of the feasible set.

- In principle it can find an optimal (integer) solution, but this may require unacceptable long time.

- In practice, usually it is terminated with a heuristically obtained integer solution, often derived by rounding a noninteger solution.
Principle of branch-and-bound

- Consider minimizing $f(x)$ over a finite set $x \in X$.

- Let $Y_1$ and $Y_2$ be two subsets of $X$ for which we have bounds:

$$m_1 \leq \min_{x \in Y_1} f(x), \quad M_2 \geq \min_{x \in Y_2} f(x).$$

- If $M_2 \leq m_1$ then the solutions in $Y_1$ may be disregarded since their cost cannot be smaller than the cost of the best solution in $Y_2$. 
The branch-and-bound method uses suitable upper and lower bounds, and the bounding principle to eliminate substantial portions of $X$. It uses a tree, with nodes that correspond to subsets of $X$, usually obtained by binary partition.
Algorithm

Initialization: OPEN=\{X\}, UPPER=+\infty

While OPEN is nonempty
  - Remove a node Y from OPEN
  - For each child Y_i of Y, find the lower bound \( m_i \) and a feasible solution \( \bar{x} \in Y_i \).
  - If \( m_i < UPPER \) place \( Y_i \) in OPEN
  - If in addition \( f(\bar{x}) < UPPER \) set \( UPPER = f(\bar{x}) \) and mark \( \bar{x} \) as the best solution found so far.

Termination: the best solution so far is optimal.

Tight lower bounds \( m_i \) are important for quick termination!
Example: facility location

- $m$ clients, $n$ locations
- $x_{ij} \in \{0, 1\}$ indicates that client $i$ is assigned to location $j$ at cost $a_{ij}$.
- $y_j \in \{0, 1\}$ indicates that a facility is placed at location $j$ (at cost $b_j$)

minimize $\sum_{i=1}^{m} \sum_{j=1}^{n} x_{ij} a_{ij} + \sum_{j=1}^{n} b_j y_j$

subject to $\sum_{j=1}^{n} x_{ij} = 1$, $i = 1, \ldots, m$

$\sum_{i=1}^{m} x_{ij} t_{ij} \leq T_j y_j$, $j = 1, \ldots, n$,

$x_{ij} \in \{0, 1\}$, $i = 1, \ldots, m$, $j = 1, \ldots, n$,

$y_j \in \{0, 1\}$, $j = 1, \ldots, n$. 

Nonlinear optimization (c) 2006 Jean-Philippe Vert, (Jean-Philippe.Vert@mines.org) – p.23/30
It is convenient to select subsets of the form:

$$X(J_0, J_1) = \{ (x, y) \text{ feasible : } y_j = 0, \forall j \in J_0, y_j = 1, \forall j \in J_1 \} ,$$

where $J_0$ and $J_1$ are disjoint subsets of facility locations (i.e., for all solutions in $X(J_0, J_1)$, a facility is placed at locations in $J_1$, no facility is placed at the locations in $J_0$, and a facility may or may not be placed at the remaining locations).

For each subset $X(J_0, J_1)$ we can obtain a lower bound and a feasible solution by solving the linear program where all integer constraints are relaxed except that the variables $y_j, j \in J_0 \cup J_1$ are fixed at either 0 or 1.
Lagrangian relaxation
Motivations

- We have seen that obtaining lower bounds on the optimal value of a discrete optimization problem is important for branch-and-bound.

- Relaxing the discrete (integer) constraint is one approach to obtain such lower bounds, by transforming the integer problem into a LP or other continuous problem.

- Here we consider another important method called Lagrange relaxation, based on weak duality.
Lagrangian relaxation

- Remember that the dual of any problem (in particular the subproblem of a node of the branch-and-bound tree) is always concave, and its maximum provides a lower bound on the optimal solution of the problem by *weak duality*.

- In Lagrange relaxation, we use the dual optimal as a lower bound to the primal subproblem.

- Essential for applying Lagrangian relaxation is that the dual problem is easy to solve (e.g., LP).
Consider the problem:

\[
\begin{align*}
& \text{minimize} \quad f(x) \\
& \text{subject to} \quad Ax \leq b, \\
& \quad x \in X ,
\end{align*}
\]

where \( f \) is convex and \( X \) is a discrete subset of \( \mathbb{R}^n \). Let \( f^* \) be the optimal primal cost.

Which bound is the tightest between constraint and Lagrange relaxation?
Comparison (cont.)

- The lower bound provided by Lagrangian relaxation is:

\[ q^* = \sup_{\mu \geq 0} \inf_{x \in X} L(x, \mu), \]

where \( L \) is the Lagrangian.

- The lower bound provided by constraint relaxation is:

\[ \hat{f} = \inf_{Ax \leq b} f(x) \]

- By strong duality of the problem with relaxed constraints (\( f \) is convex) we know that:

\[ \hat{f} = \hat{g} = \sup_{\mu \geq 0} \inf_{x \in \mathbb{R}^n} L(x, \mu) \leq q^*. \]
Comparison (cont.)

- The lower bound obtained by Lagrangian relaxation is no worse than the lower bound obtained by constraint relaxation.
- However computing the dual function may be complicated (due to other constraints), and the maximization of the dual may be nontrivial (in particular it is typically nondifferentiable).