EC674 CHEAT SHEET

CHAPTER 1: INTRODUCTION

general linear programming problem: given a cost vector $\vec{c}=(c_1,\dots,c_n)$ and we seek to minimize a linear cost function $c'x = \sum_{i=1}^{n} c_i x_i$ over all n dimensional vectors $\vec{x}$ SUBJECT TO A SET OF LINEAR EQUALITY/INEQUALITY CONSTRAINTS:

$$\begin{bmatrix} \text{minimize } \vec{c}'\vec{x} & \\ \text{subject to } a_i'x \ge b_i & i\in M_1 \\ a_i'x \le b_i & i\in M_2 \\ a_i'x = b_i & i \in M_3 \\ x_j \ge 0 & j\in N_1 \\ x_j \le 0 & j \in N_2 \end{bmatrix}$$

- decision variables: $x_1,\dots,x_n$
- Feasible solution: a vector $\vec{x}$ satisfying all conditions
- objective function: $\vec{c}'\vec{x}$
- free variable: x_j s.t. $j\notin N_1$, $j\notin N_2$
- optimal solution: $\vec{x}^*$ s.t. $\vec{c}'\vec{x}^* \le \vec{c}'\vec{x}$ $\forall$ feasible x

standard form linear programming problem: a problem of the form

$$\begin{bmatrix} \text{minimize } \vec{c}'\vec{x} \\ \text{subject to } Ax=b,\ x\ge 0 \end{bmatrix}$$

TO REDUCE A PROBLEM TO STANDARD FORM:

a) **Elimination of free variables:** Given a free variable x_j, replace it w/ $x_j^+ - x_j^-$ where x_j^+, x_j^- are new variables w/ positivity constraints $x_j^+\ge 0$, $x_j^-\ge 0$

b) **Elimination of inequality constraints:** Given an inequality constraint of the form $\sum_{j=1}^{n} a_{ij}x_j \le b_i$, introduce new slack variable s_i and the standard form constraints: $\sum_{j=1}^{n} a_{ij}x_j + s_i = b_i$, $s_i\ge 0$.

convex: a function $f:\mathbb{R}^n \mapsto \mathbb{R}^n$ is called convex if for every $x,y\in\mathbb{R}^n$ and every $\lambda\in[0,1]$, we have $f(\lambda\vec{x}+(1-\lambda)\vec{y}) \ge \lambda f(\vec{x}) + (1-\lambda) f(\vec{y})$

concave: a function f is concave iff $-f$ is convex

THEOREM 1.1: Let $f_1,\dots,f_m:\mathbb{R}^n\mapsto\mathbb{R}$ be convex functions. Then the function f defined by $f(x)=\max_{i=1,\dots,m} f_i(x)$ is also convex

LPP formulation of minimizing a piecewise-linear convex function:

$$\begin{bmatrix} \text{minimize } \max_{i\in[1,\dots,m]}(c_i'x+d_i) \\ \text{s.t. } Ax\ge b \end{bmatrix} \implies \begin{bmatrix} \text{minimize } z \\ \text{s.t. } z\ge c_i'x+d_i \quad i=1,\dots,m \\ Ax\ge b \end{bmatrix}$$

Rewriting piecewise linear convex constraints:

$f(x)\le h$, $f(x)$ piecewise linear convex, $f(x)=\max_{i=1,\dots,m}(f_i'x+g_i)$ $\implies$ $f_i'x+g_i\le h$, $i=1,\dots,m$

problems involving absolute values: Consider a problem of the form

$$\begin{bmatrix} \text{minimize } \sum_{i=1}^{n} c_i|x_i| \\ \text{subject to } A\vec{x}\ge b \end{bmatrix}$$ (cost coefficients c_i assumed to be nonnegative)

Formulation #1: Note $|x_i|$ = smallest number satisfying $x_i\le z_i$, $-x_i\le z_i$, and obtain:

$$\begin{bmatrix} \text{minimize } \sum_{i=1}^{n} c_i z_i \\ \text{subject to } Ax\ge b,\ x_i\le z_i,\ -x_i\le z_i \quad i=1,\dots,n \end{bmatrix}$$

Formulation #2: introduce new variables x_i^+, x_i^- constrained to be nonnegative, and let $x_i = x_i^+ - x_i^-$ (x_i will be = to either x_i^+ or x_i^- depending on its sign) and replace every occurrence of $|x_i|$ with $x_i^+ + x_i^-$ and obtain the alternate formulation

$$\begin{bmatrix} \text{minimize } \sum_{i=1}^{n} c_i(x_i^+ + x_i^-) \\ \text{subject to } Ax^+ - Ax^- \ge b \\ x^+, x^- \ge 0 \end{bmatrix}$$ $(x^+ = (x_1^+\dots x_n^+),\ x^-=(x_1^-,\dots,x_n^-))$

- **column space:** given A an $m\times n$ matrix, the column space is the subspace of $\mathbb{R}^m$ spanned by the cols of A
- **row space:** the subspace of $\mathbb{R}^n$ spanned by the rows of A
- dimension of the row space is always equal to dim(col space), and is called the **rank** of A
- $\text{rank}(A)\le\min\{m,n\}$. A is full rank if $\text{rank}(A)=\min\{m,n\}$
- **nullspace:** the set $\{x\in\mathbb{R}^n \mid Ax=0\}$, a subspace of $\mathbb{R}^n$ w/ dimension $n-\text{rank}(A)$

possible outcomes for linear programming problems

a) There exists a unique optimal solution

b) There exists multiple optimal solutions (this set could be bounded or unbounded)

c) The optimal cost is $-\infty$, and no feasible solution is optimal

d) The feasible set is empty

★ e) An optimal solution does not exist even though the problem is feasible (never arises in LP)

Vectors and Matrices

- **matrix:** a matrix of dimensions $m\times n$ is an array of real numbers a_{ij} $A=\begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & [illegible] \\ \vdots & & & \vdots \\ a_{m1} & a_{m2} & \cdots & a_{mn}\end{bmatrix}$
  - a_{ij} or $[A]_{ij}$ the (i,j)th entry of A • A_j = the jth column • a_i: ith row
  - **row vector:** a matrix with $m=1$
  - **column vector:** a matrix with $n=1$, synonymous with vector
  - **transpose:** $[A']_{ij}=[A]_{ji}$
  - **inner product:** $x'y=y'x=\sum_{i=1}^{n} x_i y_i$ • **orthogonal:** $x'y=0$
  - **Euclidean norm:** $\|x\|=\sqrt{x'x}$ • **Schwarz inequality:** $|x'y|\le\|x\|\cdot\|y\|$ (equality holds when $y=cx$)
  - **matrix multiplication** AB: $[AB]_{ij}=\sum_{k=1}^{n}[A]_{ik}[B]_{kj}=a_i'B_j$
    - $(AB)C=A(BC)$ • $AB\ne BA$ • $(AB)'=B'A'$
    - $Ax=\begin{bmatrix} a_1'x \\ \vdots \\ a_m'x\end{bmatrix}$
  - **square matrix:** $m=n$ • **identity matrix:** square matrix w/ $a_{ii}=1$, $a_{ij}=0$, $j\ne i$ SATISFIES $IA=A$ $BI=B$
  - **invertible:** a matrix A is invertible if $\exists B$ s.t. $AB=BA=I$
    - $(A')^{-1}=(A^{-1})'$ • $(AB)^{-1}=B^{-1}A^{-1}$
  - **linear independence:** given a finite collection of vectors $x^1,\dots,x^K\in\mathbb{R}^n$, they are linearly dependent if $\exists$ real numbers $a_1,\dots,a_K$ (not all 0) s.t. $\sum_{k=1}^{K} a_k x^k = 0$
  - EQUIVALENT DEFINITION OF LINEAR INDEPENDENCE: none of $x^1,\dots,x^K$ is a linear combo of the others

THEOREM 1.2: Let A be a square matrix. The following statements are equivalent:

a) A, A' are invertible b) $\det(A)$ is nonzero c) Rows of A are linearly independent d) Columns of A are linearly independent. e) For all b, $Ax=b$ has a unique solution f) There exists some vector b s.t. $Ax=b$ has a unique solution

Cramer's rule: Assume A an invertible matrix. An explicit formula for $x=A^{-1}b$ is given by $x_j=\frac{\det(A^j)}{\det(A)}$, where A^j is A but the jth column is replaced by b

- **subspace:** a subset S of $\mathbb{R}^n$ s.t. $ax+by\in S$ $\forall x,y\in S$, $a,b\in\mathbb{R}$; proper subspace if $S\ne\mathbb{R}^n$
- **span:** the span of a finite # of vectors $x^1,\dots,x^K$ in $\mathbb{R}^n$ is the subspace of $\mathbb{R}^n$ defined as the set of all vectors y of the form $y=\sum_{k=1}^{K} a_k x^k$, where $a_k\in\mathbb{R}$ [y is a linear combo of x]
- **basis:** given a subspace S of $\mathbb{R}^n$, $S\ne\{0\}$, a basis of S is a collection of linearly independent vectors s.t. their span is equal to S
- **dimension** of S is the # of vectors in its basis (e.g. $\dim(\mathbb{R}^n)=n$)
- If S a subspace of $\mathbb{R}^n$ with dimension $m\le n$, $\exists$ $n-m$ linearly independent vectors orthogonal to S

THEOREM 1.3: Suppose the span S of $x^1,\dots,x^K$ has dimension m. Then:

a) $\exists$ a basis of S consisting of m of the vectors $x^1,\dots,x^K$

b) If $k\le m$ and $x^1,\dots,x^k$ are linearly independent we can form a basis of S by starting with $x_1,\dots,x^k$ and choosing $m-k$ of the vectors $x^{k+1},\dots,x^K$

- **affine subspace:** Let S_0 be a subspace of $\mathbb{R}^n$, x^0 be some vector. $S = S_0 + x^0 = \{x + x^0 \mid x \in S_0\}$

  $\dim(S) = \dim(S_0)$. Eg. [the set defined by $x^0 + \lambda_1 x^1 + \lambda_2 x^2 + \dots \lambda_k x^k$]

  Fix x_0 s.t. $Ax^0 = b$

  $[S = \{x \in \mathbb{R}^n \mid Ax = b\} \Rightarrow S_0 = \{y \mid Ay = 0\},\ S = \{y + x^0 \mid y \in S_0\}]$

SOME OPERATION COUNTS:

- Inverting a matrix is $O(n^3)$
- matrix-vector product: $O(n^2)$ (n×n matrix and n-dim vector)
- Solving systems of n equations in n unknowns is $O(n^3)$
- given $a, b \in \mathbb{R}^n$, $a'b$ takes $2n-1$ operations $O(n)$
- given A, B n×n matrices, AB takes $(2n-1)n^2$ ops

Definition 1.2: Algorithm Runtime Theory: Let f and g be functions that map positive #s to positive #s

a) $f(n) = O(g(n))$ if $\exists$ positive numbers n_0 and c s.t. $f(n) \le c\,g(n)$ for all $n \ge n_0$

b) We write $f(n) = \Omega(g(n))$ if $\exists$ positive numbers n_0 and c s.t. $f(n) \ge c\,g(n)$ $\forall n \ge n_0$

c) We write $f(n) = \Theta(g(n))$ if $f(n) = O(g(n))$ and $f(n) = \Omega(g(n))$

- **polynomial time algorithms** have running time $O(n^k)$ for some positive integer k
- **Exponential time algorithms** have running time $\Omega(2^{cn})$

CHAPTER 2: Geometry of Linear Programming

- **polyhedron:** A polyhedron is a set that can be described $\{x \in \mathbb{R}^n \mid Ax \ge b\}$, A an m×n matrix, b a vector in $\mathbb{R}^m$
- **bounded:** a set $S \subset \mathbb{R}^n$ is bounded if $\exists$ a constant k s.t. the absolute value of every component of every element of S is less than or equal to k
- Let a be a nonzero vector in $\mathbb{R}^n$ and let b be a scalar
  - The set $\{x \in \mathbb{R}^n \mid a'x = b\}$ is called a **hyperplane**
  - The set $\{x \in \mathbb{R}^n \mid a'x \ge b\}$ is called a **halfspace**
- **convex set:** a set $S \subseteq \mathbb{R}^n$ s.t. for any $x, y \in S$, $\lambda \in [0,1]$, $\lambda x + (1-\lambda) y \in S$.
- Let $x^1, \dots, x^k$ be vectors in $\mathbb{R}^n$ and let $\lambda_1, \dots, \lambda_k$ be nonnegative scalars who sum to 1.
  - The vector $\sum_{i=1}^{k} \lambda_i x^i$ is said to be a ~~convex combination~~ convex combination of the vectors $x^1, \dots x^k$
  - The **convex hull** of the vectors $x^1, \dots x^k$ is the set of all convex combinations of these vectors

THEOREM 2.1 a) The intersection of convex sets is convex

b) Every polyhedron is a convex set

c) A convex combination of a finite number of elements of a convex set is also convex

d) The convex hull of a finite number of vectors is a convex set

- **extreme point:** If P is a polyhedron, $x \in P$ is an extreme point of P if we cannot find two vectors $y, z \in P$, both different from x, and a scalar $\lambda \in [0,1]$, s.t. $x = \lambda y + (1-\lambda) z$
- **vertex:** given a polyhedron P, $x \in P$ is a vertex of P if there exists some c s.t. $c'x < c'y$ for all y satisfying $y \in P$ and $y \ne x$
- **active/binding constraint:** if a vector x^* satisfies $a_i' x^* = b_i$ for some i in M_1, M_2, or M_3, the corresponding constraint is active/binding

THEOREM 2.2 Let x^* be an element of $\mathbb{R}^n$, $I = \{i \mid a_i' x^* = b_i\}$ be the set of indices of constraints that are active at x^*. Then, the following are equivalent:

a) $\exists$ n vectors in the set $\{a_i \mid i \in I\}$, which are linearly independent

b) The span of the vectors a_i, $i \in I$, is all of $\mathbb{R}^n$

c) The system of equations $a_i' x = b_i$, $i \in I$, has a unique solution

DEFINITION: Consider a polyhedron P defined by linear equality and inequality constraints, and let x^* be $\in \mathbb{R}^n$.

a) The vector x^* is a **basic solution** if:

  i) All equality constraints are active

  ii) Out of the constraints that are active at x^*, n of them are linearly independent

b) If x^* is a basic solution that satisfies all the constraints (≥ 0 ones), it is a **basic feasible solution**

THEOREM 2.3: Let P be a nonempty polyhedron and let $x^* \in P$. The following are equivalent:

a) x^* is a vertex b) x^* is an extreme point c) x^* is a BFS

Corollary 2.1: Given a finite number of linear equality constraints, there can be only a finite # of basic or basic feasible solutions

- two distinct basic solutions to a set of linear constraints in $\mathbb{R}^n$ are said to be **adjacent** if we can find $n-1$ independent constraints that are active at both of them

THEOREM 2.4: Consider the constraints $Ax = b$ and $x \ge 0$ and assume the m×n matrix A has linearly independent rows.

A vector $x \in \mathbb{R}^n$ is a basic solution iff we have $Ax = b$ and there exists indices $B(1), \dots, B(m)$ s.t.

a) The columns $A_{B(1)}, \dots A_{B(m)}$ are linearly independent

b) If $i \ne B(1), \dots B(m)$, then $x_i = 0$

[Let B be the m×m matrix $B = \begin{bmatrix} | & & | \\ A_{B(1)} & \dots & A_{B(m)} \\ | & & | \end{bmatrix}$

$x_B = (x_{B(1)}, \dots x_{B(m)})^T$

Then $x_B = B^{-1} b$ (since $B x_B = b$)]

PROCEDURE FOR CONSTRUCTING BASIC SOLUTIONS

1. Choose m linearly independent columns $A_{B(1)}, \dots A_{B(m)}$
2. Let $x_i = 0$ for all $i \ne B(1), \dots B(m)$
3. Solve the system of m equations $Ax = b$ for the unknowns $x_{B(1)}, \dots x_{B(m)}$

- If x is a basic solution, the variables $x_{B(1)}, \dots x_{B(m)}$ are called ~~basic~~ basic variables, and the remaining variables are called nonbasic (constraints active: $= 0$)

THEOREM 2.5: Full Row Rank Assumption: Let $P = \{x \mid Ax = b, x \ge 0\}$, be a nonempty polyhedron where A is a matrix of dimensions m×n, with rows $a_1', \dots a_m'$. Suppose that $\operatorname{rank}(A) = k < m$ and rows $a_{i_1}', \dots a_{i_k}'$ are linearly independent. Consider the polyhedron $Q = \{x \mid a_{i_1}' x = b_{i_1}, \dots a_{i_k}' x = b_{i_k}, x \ge 0\}$. Then $Q = P$

- **degenerate:** a basic solution $x \in \mathbb{R}^n$ is said to be degenerate if more than n constraints are active at x
  - **IN STANDARD FORM POLYHEDRA:** $P = \{x \in \mathbb{R}^n \mid Ax = b, x \ge 0\}$ (A m×n), x a basic solution, then x is degenerate if more than $n-m$ of the components are zero
- **DEFINITION:** A polyhedron $P \subset \mathbb{R}^n$ contains a line if $\exists$ a vector $x \in P$ and a nonzero vector $d \in \mathbb{R}^n$ such that $x + \lambda d \in P$ for all scalars λ

Theorem 2.6: Suppose that the polyhedron $P = \{x \in \mathbb{R}^n \mid a_i' x \ge b_i, i = 1, \dots m\}$ is nonempty. The following are equivalent:

a) The polyhedron P has at least one extreme point b) The polyhedron does not contain a line

c) $\exists$ n vectors out of the family $a_1, \dots a_m$ which are linearly independent (we get these from nonzero constraints)

Corollary 2.2: Every nonempty bounded polyhedron and every nonempty polyhedron in standard form has at least one BFS.

Theorem 2.7: Consider the linear programming problem of minimizing $c'x$ over a polyhedron P. Suppose that P has at least one extreme point and that there exists an optimal solution. Then $\exists$ an optimal solution which is an extreme point of P

THEOREM 2.8: Consider the linear programming problem of minimizing $c'x$ over a polyhedron P. Suppose that P has at least one extreme point. Then, either the optimal cost is equal to $-\infty$ or $\exists$ an extreme point which is optimal

Corollary 2.3 Consider the linear programming problem of minimizing $c'x$ over a nonempty polyhedron. Then either the optimal cost is equal to $-\infty$ or $\exists$ an optimal solution

CHAPTER 3: THE SIMPLEX METHOD

- **Feasible direction:** Let x be an element of a polyhedron P. A vector $d \in \mathbb{R}^n$ is said to be a feasible direction at x, if there exists a positive scalar θ for which $x + \theta d \in P$
- **reduced cost:** let x be a basic solution and B be an associated basis matrix, and let c_B be the vector of costs of the basic variables. For each j, we define the reduced cost $\bar{c}_j$ of the variable x_j according to the formula $[\bar{c}_j = c_j - c_B' B^{-1} A_j] \Rightarrow \bar{c} = c - c_B B^{-1} A$

Theorem 3.1 Consider a basic feasible solution $\vec{x}$ associated with a basis matrix B, and let $\bar{c}$ be the corresponding vector of reduced costs

a) If $\bar{c} \ge 0$, then x is optimal

b) If x is optimal AND nondegenerate, then $\bar{c} \ge 0$

- **optimal basis matrix:** a basis matrix B is optimal if:

  a) $B^{-1} b \ge 0$ (FEASIBILITY)

  b) $\bar{c}' = c' - c_B' B^{-1} A \ge 0$ (OPTIMALITY)

THEOREM 3.2

a) The columns $A_{B(i)}$, $i \ne l$ (minimizing index), and A_j are linearly independent and therefore $\bar{B}$ is a basis matrix

b) The vector $y = x + \theta^* d$ is a BFS associated w/ basis matrix $\bar{B}$

OPTIMAL p given x^*: $p' = c_B' B^{-1}$

↳ reduced costs of slack variables

$\binom{n}{k} = \frac{n!}{k!(n-k)!}$

An iteration of the simplex method:

1. Start with a basis consisting of basic columns $A_{B(1)}, \ldots, A_{B(m)}$ and associated BFS $\vec{x}$.
2. Compute the reduced costs $\bar{c}_j = c_j - c_B' B^{-1} A_j$ for nonbasic indices j. If all nonnegative, current $\vec{x}$ is optimal and algorithm terminates. Else, choose j for which $\bar{c}_j < 0$
3. Compute $u = B^{-1} A_j$. if u has no positive component, $\theta^* = \infty$, optimal cost is $-\infty$
4. If u has some positive component, $\theta^* = \min_{\{i=1,\ldots,m \mid u_i > 0\}} \frac{x_{B(i)}}{u_i}$
5. Let l be such that $\theta^* = \frac{x_{B(l)}}{u_l}$. Form a new basis by replacing $A_{B(l)}$ with A_j. If y is the new BFS, the values of the new basic variables are $y_j = \theta^*$, $y_{B(i)} = x_{B(i)} - \theta^* u_i$

THEOREM 3.3: Assume the feasible set is nonempty and that every BFS is nondegenerate. Then the simplex method terminates in a finite # of iterations. At termination, either:

a) We have an optimal basis B and associated BFS which is optimal

b) We have found a vector d satisfying $Ad = 0$, $d \geq 0$, and $c'd < 0$, optimal cost $= -\infty$

Some pivot selection rules:

FOR ENTERING COLUMN:

a) Choose A_j w/ $\bar{c}_j < 0$ whose reduced cost is the most negative

b) Choose A_j w/ $\bar{c}_j < 0$ for which the corresponding cost decrease $\theta^* |\bar{c}_j|$ is largest

c) smallest subscript rule: choose the smallest j for which $\bar{c}_j$ is negative (using this rule for both the entering and exiting column, cycling can be avoided!)

• Elementary row operation: given a matrix, not necessarily square, the operation of adding a constant multiple of one row to another row (or the same)

ITERATION OF THE REVISED SIMPLEX METHOD

1. Start with a basis consisting of the basic columns $A_{B(1)}, \ldots, A_{B(m)}$, an associated basic feasible solution x, and the inverse B^{-1} of the basis matrix
2. Compute the row vector $p' = c_B' B^{-1}$ and the reduced costs $\bar{c}_j = c_j - p'A_j$. If they are all nonnegative, the current BFS is optimal, and the algorithm terminates. Else choose j for which $\bar{c}_j < 0$
3. Compute $u = B^{-1} A_j$. If no component of u is positive, optimal cost is $-\infty$, algorithm terminates
4. If some component of u is positive, let $\theta^* = \min_{\{i=1,\ldots,m \mid u_i > 0\}} \frac{x_{B(i)}}{u_i}$
5. Let l be such that $\theta^* = \frac{x_{B(l)}}{u_l}$. Form new basis by replacing $A_{B(l)}$ with A_j. If y is the new BFS, the values of the new basic variables are $y_j = \theta^*$, $y_{B(i)} = x_{B(i)} - \theta^* u_i$
6. Form the $m \times (m+1)$ matrix $[B^{-1} | u]$. Add to each one of its rows a multiple of the lth row to make the last column equal to the unit vector e_l. The first m columns of the result is the matrix $\bar{B}^{-1}$.

• STRUCTURE OF SIMPLEX TABLEAU

$-c_B' B^{-1} b$	$c' - c_B' B^{-1} A$
$B^{-1} b$	$B^{-1} A$

$\Rightarrow$

$-c_B' x_B$	$\bar{c}_1$	$\ldots$	$\bar{c}_n$
$x_{B(1)}$ $\vdots$ $x_{B(m)}$	$B^{-1}A_1$		$B^{-1}A_n$

← reduced costs

← u_j's

ITERATION OF FULL TABLEAU IMPLEMENTATION

1. Start with tableau associated with basis matrix B and corresponding BFS x.
2. Examine the reduced costs of the zeroth row of the tableau. If all nonnegative, current BFS is optimal & algorithm terminates; else choose j for which $\bar{c}_j < 0$
3. Consider $u = B^{-1} A_j$, jth column of tableau. If no component of u positive, optimal cost is $-\infty$
4. For each i for which u_i is positive, compute $\frac{x_{B(i)}}{u_i}$, let l be the index of the row corresponding to the smallest ratio. Column $A_{B(l)}$ exits the basis and the column A_j enters.
5. Add to each row of the tableau a constant multiple of the lth row (the pivot row) so that u_l (the pivot element) becomes 1 and all other entries in the pivot column become 0.

	Full Tableau	Revised Simplex
Memory	$O(mn)$	$O(m^2)$
Worst-case time	$O(mn)$	$O(mn)$
Best-case time	$O(mn)$	$O(m^2)$

Revised simplex method cannot be slower than full tableau, and could be much faster during most iterations

• Definition: a vector $u \in \mathbb{R}^n$ is said to be lexicographically larger (or smaller) than another vector $v \in \mathbb{R}^n$ if $u \neq v$ and the first nonzero component of $u - v$ is positive (or negative, respectively) and we write $u \overset{L}{>} v$ OR $u \overset{L}{<} v$

• Lexicographic pivoting rule

1. Choose an entering column A_j arbitrarily as long as its reduced cost $\bar{c}_j$ is negative. Let $u = B^{-1} A_j$ be the jth column of the tableau
2. For each i with $u_i > 0$, divide the ith row of the tableau (including entry in 0th column) by u_i and choose the lexicographically smallest row. If l is the lexic. smallest, lth basic variable $x_{B(l)}$ exits the basis

THEOREM 3.4: Suppose the simplex algorithm starts w/ all rows in the simplex tableau (besides the 0th row) lexicographically positive. Suppose the lexicographic pivoting rule is followed. Then:

a) Every row of simplex tableau other than 0th row remains lexicographically positive throughout the algorithm

b) The zeroth row strictly increases lexicographically at each iteration

c) The simplex method terminates after a finite # of iterations

Bland's Rule (smallest subscript pivoting rule)

1. Find the smallest j for which $\bar{c}_j$ is negative and have A_j enter the basis
2. Out of all x_i's that are tied in the test for choosing an exiting variable, choose one with smallest value of i

TWO-PHASE SIMPLEX METHOD

PHASE I:

1. By multiplying some of the constraints by -1, change the problem so that $b \geq 0$
2. Introduce artificial variables $y_1, \ldots, y_m$, if necessary, and apply the simplex method to the auxiliary problem with cost $\sum_{i=1}^m y_i$
3. If the optimal cost in the auxiliary problem is positive, original problem is infeasible. Algorithm terminates
4. If optimal cost in auxiliary problem is 0, a feasible solution to the original problem has been found. If no artificial variables ~~…~~ is in the final basis, the artificial variables and their corresponding columns are eliminated, and a feasible basis for the original problem is available.
5. If the lth basic variable is an artificial one, examine the lth entry of the columns $B^{-1}A_j$, $j = 1, \ldots, n$. If all these entries are 0, the lth row represents a redundant constraint and is eliminated. Else, the lth entry of the jth column is nonzero, apply a change of basis (w/ this entry serving as the pivot element): the lth basic variable exits and x_j enters the basis. Repeat until all artificial variables are driven out of the basis

PHASE II:

1. Let the final basis and tableau of Phase I be the initial basis and tableau for phase II
2. Compute reduced costs of all variables for initial basis using cost coefficients from the original problem
3. Apply the simplex method to the original problem

THEOREM 3.5: Consider the LPP of minimizing $-x_n$ s.t. $\epsilon \leq x_1 \leq 1$; $\epsilon x_{i-1} \leq x_i \leq 1 - \epsilon x_{i-1}$, $i = 2, \ldots, n$. Then:

a) The feasible set has 2^n vertices

b) The vertices can be ordered s.t. each one is adjacent to and has lower cost than the previous one

c) $\exists$ a pivoting rule under which the simplex method requires $2^n - 1$ changes of bases to terminate

• diameter of a polyhedron:

- $d(x,y)$ = minimum # of adjacent jumps to get from x to y
- diameter $D(P)$ = max of $d(x,y)$ over all pairs of vertices
- $\Delta(n,m)$ = maximum of $D(P)$ over all bounded polyhedra in $\mathbb{R}^n$ represented in terms of m linear equality constraints

$\Delta(2,m) = \lfloor \frac{m}{2} \rfloor$

• Hirsch Conjecture: $\Delta(n,m) \leq m - n$

CHAPTER 4: DUALITY THEORY

• a general primal/dual pair:

minimize $c'x$		maximize $p'b$		
subject to $a_i'x \geq b_i$	$i \in M_1$	subject to	$p_i \geq 0$	$i \in M_1$
$a_i'x \leq b_i$	$i \in M_2$		$p_i \leq 0$	$i \in M_2$
$a_i'x = b_i$	$i \in M_3$		p_i free	$i \in M_3$
$x_j \geq 0$	$j \in N_1$		$p'A_j \leq c_j$	$j \in N_1$
$x_j \leq 0$	$j \in N_2$		$p'A_j \geq c_j$	$j \in N_2$
x_j free	$j \in N_3$		$p'A_j = c_j$	$j \in N_3$

minimize $c'x$ s.t. $Ax = b$, $x \geq 0$ $\Longleftrightarrow$ max $p'b$ s.t. $p'A = c'$, $p \geq 0$

min $c'x$ s.t. $Ax \geq b$ $\Longleftrightarrow$ max $p'b$, $p'A = c'$, $p \geq 0$

PRIMAL	minimize	maximize	DUAL
constraints	$\geq b_i$	≥ 0	variables
	$\leq b_i$	≤ 0	
	$= b_i$	free	
variables	≥ 0	$\leq c_j$	constraints
	≤ 0	$\geq c_j$	
	free	$= c_j$	

THEOREM 4.1: If we transform the dual into an equivalent minimization problem and form its dual we obtain a problem equivalent to the original problem... the dual of the dual is the primal

THEOREM 4.2: Suppose we have transformed a LPP Π_1 to another LPP Π_2 by a sequence of the following transformations:

a) Replace a free variable w/ the difference of two nonnegative variables

b) Replace an inequality constraint involving a nonnegative slack variable

c) If some row of A in a feasible standard form problem is a linear combo of the other rows, eliminate the corresponding inequality constraint

Then, the duals Π_1 and Π_2 are equivalent (either both equivalent or have the same optimal cost.

THEOREM 4.3 (WEAK DUALITY): If x is a feasible solution to the primal problem and p a feasible solution to the dual problem, then $p'b \leq c'x$

COROLLARY 4.1:

a) If the optimal cost of the primal is $-\infty$, the dual must be infeasible

b) If the optimal cost of the dual is ∞, the primal problem must be infeasible

COROLLARY 4.2: Let x, p be feasible solutions to the primal and dual respectively & suppose $p'b = c'x$. Then x and p are optimal solutions to the primal and dual, respectively

THEOREM 4.4 (STRONG DUALITY): If a LPP has an optimal solution, so does its dual, & respective optimal costs are equal

	Finite optimum	Unbounded	Infeasible
Finite Optimum	✓	✗	✗
Unbounded	✗	✗	✓
Infeasible	✗	✓	✓

CLARK'S THM: unless both primal & dual are infeasible, at least one of them must have an unbounded feasible set

THEOREM 4.5 (Complementary slackness): Let x, p be feasible solutions to the primal/dual problem. x and p are optimal for their respective problems iff

$p_i(a_i'x - b_i) = 0 \; \forall i$, $(c_j - p'A_j) x_j = 0 \; \forall j$.

• Each component p_i of the optimal dual vector can be interpreted as the marginal cost (shadow price) per unit increase of the ith requirement b_i

ITERATION OF DUAL SIMPLEX METHOD

1. Start with a tableau associated w/ a basis matrix B, all reduced costs nonnegative
2. Examine $B^{-1}b$ in first column. If all nonnegative, we have optimal BFS & algorithm terminates. Else, choose l s.t. $x_{B(l)} < 0$
3. Consider lth row of tableau w/ elements $x_{B(l)}, v_1, \ldots, v_n$ (the pivot row). If $v_i \geq 0 \; \forall i$, the optimal dual cost is $+\infty$, algorithm terminates
4. Else, $\forall i$ s.t. $v_i < 0$, compute $\bar{c}_i / |v_i|$ & let j be the index of the column corresponding to the smallest ratio. $A_{B(l)}$ exits the basis and A_j takes its place
5. Add to each row of the tableau a multiple of the lth row (pivot row) s.t. v_j (pivot element) becomes 1 and all other entries of the pivot column become 0.

LEXICOGRAPHIC PIVOTING RULE FOR DUAL SIMPLEX METHOD

1. Choose any row l s.t. $x_{B(l)} < 0$ to be the pivot row
2. Determine the index j of the entering column as follows: for each column with $v_i < 0$, divide all entries by $|v_i|$, and then choose the lexicographically smallest column. If there is a tie, choose the one w/ the smallest index.

Some properties of Bases and Basic Solutions

a) Every basis determines a basic solution to the primal AND a corresponding basic solution to the dual, $p' = c_B' B^{-1}$

b) The dual basic solution is feasible iff all of the reduced costs are nonzero

c) Under this dual basic solution the reduced costs that are equal to zero correspond to active constraints in the dual problem.

d) The dual basic solution is degenerate iff some nonbasic variable has zero reduced cost

Theorem 4.6 (Farkas' Lemma): Let A be a matrix $m \times n$, $b \in \mathbb{R}^m$. Then exactly one of the following holds:

a) $\exists$ some $x \geq 0$ s.t. $Ax = b$

b) $\exists$ some p s.t. $p'A \geq 0'$, $p'b < 0$

Corollary 4.3: Let $A_1, \ldots, A_n$ and b be given vectors and suppose any vector p that satisfies $p'A_i \geq 0$, $i = 1, \ldots, n$, must also satisfy $p'b \geq 0$. Then b can be expressed as a nonnegative linear combo of $A_1, \ldots, A_n$.

THEOREM 4.7: Suppose $Ax \leq b$ has at least one solution and let d be some scalar. The following are equivalent:

a) Every feasible solution to the system $Ax \leq b$ satisfies $c'x \leq d$

b) There exists some $p \geq 0$ s.t. $p'A = c'$, $p'b \leq d$

THEOREM 4.8: The absence of arbitrage condition holds iff $\exists$ a nonnegative $\vec{q} = (q_1, \ldots, q_m)$ s.t. the price of each asset i is given by $p_i = \sum_{s=1}^m q_s r_{si}$

cone: a set $C \in \mathbb{R}^n$ s.t. $\lambda x \in C$ for all $\lambda \geq 0$, $x \in C$ [if C nonempty, $0 \in C$]

• polyhedral cone: polyhedron of the form $P = \{x \in \mathbb{R}^n \mid Ax \geq 0\}$

THEOREM 4.12: Let $C \subset \mathbb{R}^n$ be the polyhedral cone defined by the constraints $a_i'x \geq 0$, $i = 1, \ldots, m$. The following are equivalent:

a) The zero vector is an extreme point of C

b) The cone does not contain a line

c) $\exists$ n vectors in $a_1, \ldots, a_m$ which are linearly independent

recession cone: consider the nonempty polyhedron $P = \{x \in \mathbb{R}^n \mid Ax \geq b\}$. Fix $y \in P$. The recession cone at y is the set of all directions d along which we can move indefinitely away from y w/o leaving P.

$\{d \in \mathbb{R}^n \mid A(y + \lambda d) \geq b, \forall \lambda \geq 0\} = \{d \in \mathbb{R}^n \mid Ad \geq 0\}$ ← independent of starting point y

• rays: of polyhedron P are the nonzero elements of the recession cone

• DEFINITION: (a) A nonzero element x of a polyhedral cone $C \subset \mathbb{R}^n$ is called an extreme ray if there are $n-1$ linearly independent constraints active at x

b) An extreme ray of the recession cone associated with nonempty polyhedron P is also called an $\boxed{\text{extreme ray}}$ of P

THEOREM 4.13: Consider the problem of minimizing $c'x$ over a pointed polyhedral cone $C = \{x \in \mathbb{R}^n \mid a_i'x \geq 0, i = 1, \ldots, m\}$. The optimal cost is $-\infty$ iff some extreme ray d of C satisfies $c'd < 0$.

THEOREM 4.14: Consider minimizing $c'x$ subject to $Ax \geq b$ and assume the feasible set has at least one extreme point. The optimal cost is $-\infty$ iff some extreme ray d of the feasible set satisfies $c'd < 0$.

• If we have a standard form problem in which the optimal cost is $-\infty$, simplex provides termination w/ extreme ray

RESOLUTION THEOREM (4.15): Let $P = \{x \in \mathbb{R}^n \mid Ax \geq b\}$ be a nonempty polyhedron w/ at least one extreme point. Let $x^1, \ldots, x^k$ be the extreme points, and let $w^1, \ldots, w^r$ be a complete set of extreme rays of P. Let

$$Q = \{\textstyle\sum_{i=1}^k \lambda_i x^i + \sum_{j=1}^r \theta_j w^j \mid \lambda_i \geq 0, \theta_j \geq 0, \sum_{i=1}^k \lambda_i = 1\}. \text{ Then } Q = P$$

Corollary 4.4: A nonempty bounded polyhedron is the convex hull of its extreme points

Corollary 4.5: Assume the cone $C = \{x \mid Ax \geq 0\}$ is pointed. Then every element of C can be expressed as a nonnegative linear combination of the extreme rays of C.

• Finitely generated sets: sets of the form $Q = \{\sum_{i=1}^k \lambda_i x^i + \sum_{j=1}^r \theta_j w^j \mid \lambda_i \geq 0, \theta_j \geq 0, \sum_{i=1}^k \lambda_i = 1\}$

THEOREM 4.16: A finitely generated set is a polyhedron. The convex hull of finitely many vectors is a bounded polyhedron.

CHAPTER 5: SENSITIVITY ANALYSIS

• Let $P(b) = \{x \mid Ax = b, x \geq 0\}$ be the feasible set. $S = \{b \mid P(b) \text{ nonempty}\} = \{Ax \mid x \geq 0\}$. S is convex!!

• For any $b \in S$, define $F(b) = \min_{x \in P(b)} c'x$, optimal cost function of b.

THEOREM 5.1: The optimal cost $F(b)$ is a convex function of b on the set S.

• Subgradient: let F be a convex function defined on a convex set S. Let b^* be an element of S. Vector p is a subgradient of F at b^* if $F(b^*) + p'(b - b^*) \leq F(b)$ $\forall b \in S$ → unique if F linear near b^*

THEOREM 5.2: Suppose the LPP of minimizing $c'x$ s.t. $Ax = b^*$, $x \geq 0$ is feasible & the optimal cost is finite. Then a vector p is an optimal solution to the dual problem iff it is a subgradient of the optimal cost function F at b^*

THEOREM 5.3: Consider a feasible LPP in standard form.

a) The set T of all c for which the optimal cost is finite is convex

b) The optimal cost $G(c)$ is a concave function of c on the set T.

c) If for some value of c the primal problem has a unique optimal solution x^*, then G is linear in the vicinity of c and its gradient is equal to x^*.

PARAMETRIC PROGRAMMING EXAMPLE

minimize $(-7+2\theta)x_1 + (-3-\theta)x_2 + x_3$
s.t. $x_1 + 2x_2 - 3x_3 \le 5$
$2x_1 + x_2 - 4x_3 \le 7$
$x_1, x_2, x_3 \ge 0$

	x_1	x_2	x_3	x_4	x_5
0	$-3+2\theta$	$3-\theta$	1	0	0
x_4 = 5	1	2	-3	1	0
x_5 = 7	2	1	-4	0	1

If $-3+2\theta \ge 0 \Rightarrow \theta \ge 3/2$ and $3-\theta \ge 0 \Rightarrow \theta \le 3$, all reduced costs nonnegative, we have optimal solution $x=(0,0,0,5,7)$, and $g(\theta)=0$.
If $\theta \ge 3$, reduced cost of x_2 becomes negative, to let x_2 enter the basis, x_4 leave.

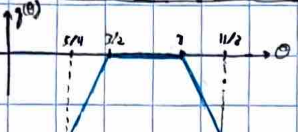
	x_1	x_2	x_3	x_4	x_5
$-7.5+2.5\theta$	$-4.5+2.5\theta$	0	$5.5-1.5\theta$	$-1.5+0.5\theta$	0
x_2 = 2.5	0.5	1	-1.5	0.5	0
x_5 = 4.5	1.5	0	-2.5	-0.5	1

Here, all reduced costs are nonnegative iff $3 \le \theta \le 11/3$. For this range of values, $g(\theta) = 7.5-2.5\theta$

If θ increased past 11/3, the reduced cost of x_3 becomes negative, but there is no positive pivot element in the third column of the tableau, so problem is unbounded, $g(\theta)=-\infty$
If $\theta < 3/2$, reduced cost of x_1 in original table becomes negative, so it enters the basis and x_5 leaves:

	x_1	x_2	x_3	x_4	x_5
$10.5-7\theta$	0	$4.5-2\theta$	$-5+4\theta$	0	$1.5-\theta$
x_4 = 1.5	0	1.5	-1	1	-0.5
x_1 = 3.5	1	0.5	-2	0	0.5

All reduced costs are nonnegative iff $5/4 \le \theta \le 3/2$, so for these values, $g(\theta) = -10.5+7\theta$
If $\theta \le 5/4$, reduced cost of x_3 is negative, but optimal cost $-\infty$



SENSITIVITY ANALYSIS SUMMARY

a) If a new variable is added, check its reduced cost. If it's negative, add a new column to the tableau & continue from there
b) If a new constraint is added, check whether it's violated. If so, form auxiliary problem and proceed from there
c) If an entry of b or c is changed by δ, we obtain an interval of values of δ for which the same basis remains optimal
d) If an entry of A is changed by δ, can perform similar analysis (tricky if change is in a basic column)
e) Assuming the dual problem is feasible, the optimal cost is a piecewise linear convex function of the vector b & subgradients of the optimal cost function correspond to optimal solutions to the dual problem.
f) If the primal problem is feasible, the optimal cost is a piecewise linear concave function of c

FULL TABLEAU SIMPLEX EXAMPLE

minimize $-10x_1 - 12x_2 - 12x_3$
s.t. $x_1 + 2x_2 + 2x_3 \le 20$
$2x_1 + x_2 + 2x_3 \le 20$
$2x_1 + 2x_2 + x_3 \le 20$
$x_1, x_2, x_3 \ge 0$

$\Rightarrow$ minimize $-10x_1 - 12x_2 - 12x_3$
s.t. $x_1 + 2x_2 + 2x_3 + x_4 = 20$
$2x_1 + x_2 + 2x_3 + x_5 = 20$
$2x_1 + 2x_2 + x_3 + x_6 = 20$
$x_1, \dots, x_6 \ge 0$

INITIAL BFS: $x=(0,0,0,20,20,20)$, $B(1)=4$, $B(2)=5$, $B(3)=6$

A	x_1	x_2	x_3	x_4	x_5	x_6
0	-10	-12	-12	0	0	0
x_4 = 20	1	2	2	1	0	0
x_5 = 20	2*	1	2	0	1	0
x_6 = 20	2	2	1	0	0	1

BASIC VARIABLES: $x_4 = x_5 = x_6 = 20$
NONBASIC VARIABLES: $x_1 = x_2 = x_3 = 0$
Choose pivot x_1, column = $u = (1,2,2)$
$\frac{x_{B(i)}}{u_i} = (20,10,10)$, choose $l=2$ (5)
5 exits
1 enters
4 active constraints
DEGEN!

D	x_1	x_2	x_3	x_4	x_5	x_6
100	0	-7	-2	0	5	0
x_4 = 10	0	3/2	1*	1	-1/2	0
x_1 = 10	1	1/2	1	0	1/2	0
x_6 = 0	0	1	-1	0	-1	1

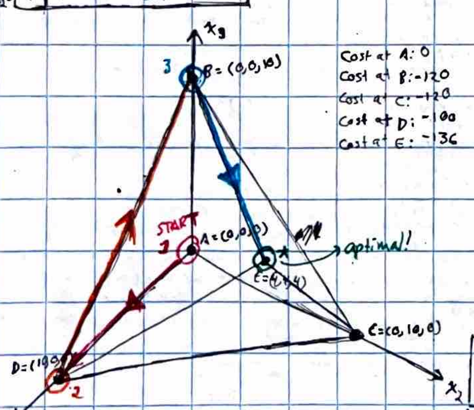
BASIC VARIABLES: $x_4 = x_1 = 10$, $x_6 = 0$
NONBASIC VARIABLES: $x_2 = x_3 = x_5 = 0$
choose pivot x_3, $u = (1,1,-1)$
$\frac{x_{B(i)}}{u_i} = (10,10,x)$, choose $l=1$
4 exits
3 enters
4 active constraints, DEGEN

B	x_1	x_2	x_3	x_4	x_5	x_6
120	0	-4	0	2	4	0
x_3 = 10	0	3/2	1	1	-1/2	0
x_1 = 0	1	-1	0	-1	1	0
x_6 = 10	0	5/2*	0	1	-3/2	1

BASIC VARIABLES: $x_1 = 0$, $x_3 = x_6 = 10$
NONBASIC VARIABLES: $x_2 = x_4 = x_5 = 0$
Choose pivot x_2, $u = (3/2, -1, 5/2)$
$\frac{x_{Bi}}{u_i} = (20/3, X, 4)$, choose $l=3$
6 exits, 2 enters

E	x_1	x_2	x_3	x_4	x_5	x_6
136	0	0	0	3.6	1.6	1.6
x_3 = 4	0	0	1	0.4	0.4	-0.6
x_1 = 4	1	0	0	-0.6	0.4	0.4
x_2 = 4	0	1	0	0.4	-0.6	0.4

BASIC VARIABLES: $x_1 = x_2 = x_3 = 4$
NONBASIC VARIABLES: $x_4 = x_5 = x_6 = 0$
$c'x = -136$, $x^* = (4,4,4,0,0,0)$

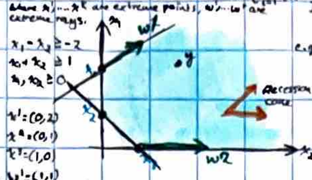


Cost at A: 0
Cost at B: -120
Cost at C: -120
Cost at D: -100
Cost at E: -136

Visualizing Representation Theorem

The resolution theorem states that any nonempty polyhedron with at least one extreme point can be expressed:
$Q = \left\{ \sum_{i=1}^{k} \lambda_i x^i + \sum_{j=1}^{r} \theta_j w^j \;\middle|\; \lambda_i, \theta_j \ge 0, \sum_{i=1}^{k} \lambda_i = 1 \right\}$
where $x^1 \dots x^k$ are extreme points, $w^1 \dots w^r$ are extreme rays.

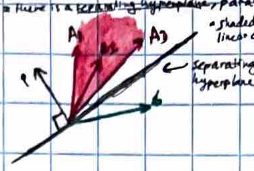
$x_1 - x_2 \ge -2$
$x_1 + x_2 \ge 1$
$x_1, x_2 \ge 0$
$x^1 = (0,2)$
$x^2 = (0,1)$
$x^3 = (1,0)$
$w^1 = (1,1)$
$w^2 = (1,0)$



e.g. $y = \begin{bmatrix}2\\2\end{bmatrix} = \begin{bmatrix}0\\1\end{bmatrix} + \begin{bmatrix}1\\1\end{bmatrix} + \begin{bmatrix}1\\0\end{bmatrix} = x^2 + w^1 + w^2$

Visualizing Farkas' Lemma

• Farkas' lemma states that exactly one of the following holds:
a) $\exists$ some $x \ge 0$ s.t. $Ax = b$ b) $\exists$ some p s.t. $p'A \ge 0$, $p'b < 0$
• b) $\Rightarrow$ there is a separating hyperplane, parameterized by p, between A and b



• shaded region represents all nonnegative linear combinations of columns of A

EXERCISE 3.18 Solution

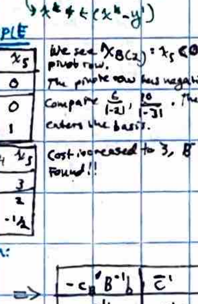
Consider the simplex method applied to a standard form problem and assume the rows of A are independent. T or F?
a) An iteration of the simplex method may move the feasible solution by a positive distance while leaving the cost unchanged.
FALSE. The entering variable always has negative reduced cost $\bar{c}_j$. The cost change is $\theta^* \bar{c}_j$. If a new feasible solution is obtained, must have $\theta^* \neq 0$, so cost change negative.
b) A variable that just left the basis cannot reenter the very next iteration.
TRUE. If x_j just exited, its reduced cost will be positive, so it will not be chosen to enter the basis.
c) A variable that has just entered the basis cannot leave the very next iteration.
FALSE. (3.8 in textbook)
d) If $\exists$ a nondegenerate optimal basis, $\exists$ a unique optimal basis
FALSE. Consider a problem w/ bounded feasible set and nonunique optimal solution
e) If x an optimal solution found by the simplex method, no more than m of its components can be positive, m = # of equality constraints
TRUE, since we assume solution is found by simplex method & therefore is a BFS.

PRACTICE EXAM SOLUTIONS

Problem 1 (T/F)
a) A problem in which a piecewise linear and concave function has to be minimized subject to linear constraints can be formulated as a LPP.
FALSE. Can only do this for convex functions
b) Consider optimizing $c'x$ s.t. $Ax=b$ and $|x_i| \ge \gamma$ $\forall i$, $\gamma \ge 0$. This can be formulated as a LPP.
FALSE. Constraints do not define a convex set
c) Consider $h: X \to \mathbb{R}$, $X \subseteq \mathbb{R}^n$. Define h^* as $h^*(\theta) \triangleq \sup \{\theta' z - h(z)\}$. Then h^* is convex.
TRUE. $h(z)$ becomes some scalar for each z, so $z \in X$ we have the max of a set of affine functions.
d) Consider max $\sum_{i=1}^{n} c_i x_i^2$ s.t. $\sum_i a_i x_i^2 \le b$, where $c_i, a_i \ge 0$ $\forall$ $i=1\dots n$. This can be formulated as a LPP.
TRUE. $y_i = x_i^2$, $y_i \ge 0$.
e) If $Ax \le b$ has a solution, then $p'b \ge 0$ is redundant in the system of equations $\{p'A=0, p\ge 0, p'b \ge 0\}$. TRUE. b can be expressed as a linear combo of the columns of A, so it is clear that if $p \ge 0$, $p'A=0$, then $p'b \ge 0$. (nonnegative)
f) In the simplex algorithm, if x_1 exits the basis at some iteration, then just after the pivot step the reduced cost $\bar{c}_1$ of x_1 will be strictly positive.
TRUE
g) If a standard form LP is unbounded, b can be adjusted to make the optimal cost finite. FALSE. Dual infeasible max $p'b$ s.t. $p'A \le c$, changing b does not change feasibility
h) At least one problem in any primal dual pair of LPs is always feasible. FALSE, both could be infeasible (table)
i) If an LPP in standard form is feasible, we can always find a BFS to start simplex.
TRUE, in standard form we generate at least one extreme point
j) If at the end of an iteration of the simplex method every nonbasic variable has positive reduced cost, the current BFS is the unique optimal solution. TRUE, if any $\bar{c}_j = 0$, would not be unique.

PROBLEM 2:
Consider the LP min $c'x$ s.t. $Ax=b$, $x \ge 0$.
a) Write down a set of necessary & sufficient conditions for (x,p) to be an optimal primal-dual pair, linear in x and p.
We need primal feasibility: $Ax=b$, $x \ge 0$
dual feasibility: $p'A \le c$
optimality: $c'x = p'b$
b) Suppose you know A, b, but not c. You are given x^* optimal solution of the primal. What is the set of c's consistent w/ x^* being optimal? What type of set is this.
$C = \{ c \in \mathbb{R}^n \mid \exists p : p'A \le c, c'x^* = p'b \}$
The set of all c's such that we can find a p satisfying feasibility & optimality conditions. This set is a cone!

PROBLEM 3: Consider a nonempty, bounded polyhedron $P = \{x \in \mathbb{R}^n \mid Ax \ge b\}$, $A \in \mathbb{R}^{m \times n}$, $b \in \mathbb{R}^m$. Assume P has at least one extreme point. We want to solve
$z_{LP1} = \min c'x$
s.t. $x \in P$, D, f, c given.
$Dx \ge f$
a) Assume LP in (1) has a unique optimal solution, denoted by x^*. Is x^* an extreme point of P?
It could be, but it is not necessarily: P
b) Consider the LP $z_{LP2} = \min c'x$ s.t. $x \in P$. Compare the optimal values z_{LP1}, z_{LP2}.
$\boxed{z_{LP1} \ge z_{LP2}}$
c) We want to express x^* as a convex combo of the extreme points of P. How many do we need? I.e. if $y^1 \dots y^k$ are the extreme points of P and $x^* = \sum_{i=1}^{s} \lambda_i y^i$, $\lambda_i \ge 0$, $\forall i$, $\sum_{i=1}^{s} \lambda_i = 1$, what is the maximum # of nonzero λ_i's.
We would need $s-(m+1)$
Pf. Define $Q = \{ \lambda \in \mathbb{R}^s \mid \lambda_i \ge 0, \sum \lambda_i = 1, \sum \lambda_i y^i = x^* \}$, a polyhedron
Caratheodory's Theorem: given a convex hull of vectors $P = \{ \sum_{i=1}^{n} \lambda_i A_i \mid \sum_{i=1}^{n} \lambda_i = 1, \lambda_1 \dots \lambda_n \ge 0 \}$, any $x \in P$ can be expressed as $\sum_{i=1}^{n} \lambda_i A_i$ where at most $m+1$ of the coefficients are nonzero.
We need n active constraints, we have m from def of P, 1 from $\sum \lambda_i = 1$, so need $n-(m+1)$ of the λ_i's to be 0.
d) Let $x^* = \sum_{i=1}^{r} \lambda_i y^i$, where $\lambda_i \ge 0$, $\sum_{i=1}^{r} = 1$ $(r = n-(m+1))$. Given x^*, y^1, we want $\lambda_1 \dots \lambda_r$ and $y^2 \dots y^r$.
Consider max t
s.t. $x = x^* + t(x^* - y^1)$
$x \in P$
. This has an optimal solution. Why?



Eventually you will hit an edge of P!
e) Let (x', t') be the optimal solution of the LP in d). Is it on a boundary or interior of P? Unique?
UNIQUE, on boundary
f) How do we proceed to complete algorithm.
Once we have found x', we need to find the y's that in convex combination form x'. We repeat this until we end up with all the y's composing x^*.

PRIMAL/DUAL STUFF

Consider min $c'x$
s.t. $Ax = b$
$x \ge 0$
Assume an optimal solution x^* exists. Introduce a relaxed problem in which the constraint $Ax=b$ is replaced by a penalty
min $c'x + p'(b-Ax)$
s.t. $x \ge 0$
Let $g(p)$ be the optimal cost of the relaxed problem. We expect $g(p)$ to be no larger than the optimal primal cost since we are allowing more options for x.
Indeed $g(p) = \min_{x \ge 0} [c'x + p'(b-Ax)] \le c'x^* + p'(b - Ax^*) = c'x^*$.
So each p leads to a lower bound $g(p)$ for optimal cost $c'x^*$. We want to find the tightest lower bound
max $g(p)$
s.t. no constraints
From the definition of $g(p)$:
$g(p) = \min_{x \ge 0} [c'x + p'(b-Ax)]$
$= p'b + \min_{x \ge 0} (c' - p'A)x$
$= \begin{cases} 0, & c' - p'A \ge 0 \\ -\infty & \text{else} \end{cases}$
So we only need to consider values of p for which $g(p)$ is not equal to $-\infty$. Then we have
max $p'b$
s.t. $p'A \le c$

CHANGE IN b

• Optimality conditions are not affected, just need to look at feasibility conditions
$B^{-1}(b + \delta e_i) \ge 0$
• Let g = the ith column of B^{-1}, $(\beta_{1i}, \dots, \beta_{mi})$.
• We have $x_B + \delta g \ge 0$
$x_{B(j)} + \delta \beta_{ji} \ge 0$, $j=1,\dots,m$
• So $\delta \ge \max_{\{j \mid \beta_{ji} > 0\}} (-x_{B(j)}/\beta_{ji})$, $\delta \le \min_{\{j \mid \beta_{ji} < 0\}} \left(\frac{-x_{B(j)}}{\beta_{ji}}\right)$

e.g.

	x_1	x_2	x_3	x_4
12	0	0	2	7
x_1 = 2	1	0	-3	2
x_2 = 2	0	1	5	-3

$B^{-1}_{(1)} = \begin{pmatrix}-3\\5\end{pmatrix}$ Basic vars: $x_1 = 2 - 3\delta$
$x_2 = 2 - 5\delta$
Basis is feasible if $-2/5 \le \delta \le 2/3$
OPTIMAL COST IN THE RANGE: $c_B' B^{-1}(b + \delta e_i)$
$= p'b + \delta p_i$
$p' = c_B' B^{-1}$
If δ outside range, apply dual simplex method

DUAL SIMPLEX EXAMPLE

	x_1	x_2	x_3	x_4	x_5
0	2	6	10	0	0
x_4 = 2	-2	4	1	1	0
x_5 = -1	4	-2*	-3	0	1

We see $x_{B(2)} = x_5 < 0$, so we choose the second row as our pivot row.
The pivot row has negative entries in the second and third column
Compare $\frac{6}{|-2|}$, $\frac{10}{|-3|}$. The smallest ratio is 3, so the second column enters the basis.

	x_1	x_2	x_3	x_4	x_5
-3	14	0	1	0	3
x_4 = 0	6	0	-5	1	2
x_2 = 1/2	-2	1	3/2	0	-1/2

Cost increased to 3, $B^{-1}b \ge 0$, so an optimal solution has been found!!

Given a LPP of the form:
minimize $c'x + 0y$
s.t. $Ax + Iy = b$
$x, y \ge 0$
$\Rightarrow$

$-c_B' B^{-1} b$	$\bar{c}'$	$-c_B' B^{-1}$
$B^{-1} b$	$B^{-1} A$	B^{-1}

E.g. New inequality constraint added

Consider min $-5x_1 - x_2 + 12x_3$
s.t. $3x_1 + 2x_2 + x_3 = 10$
$5x_1 + 3x_2 + x_4 = 16$
$x_1, \dots, x_4 \ge 0$
OST is

	x_1	x_2	x_3	x_4
12	0	0	2	7
x_1 = 2	1	0	-3	2
x_2 = 2	0	1	5	-3

Add constraint $x_1 + x_2 \ge 5$, violated by optimal solution
$x_1 + x_2 - x_5 = 5$
a = components of a_{m+1} associated w/ basic variables = (1,1).
Then
$a'B^{-1}A - a'_{m+1} = [1\,1]\begin{bmatrix}1 & 0 & -3 & 2\\ 0 & 1 & 5 & -3\end{bmatrix} - [1\,1\,0\,0] = [0\,0\,2\,-1]$
Our new tableau uses this as the final row:

	x_1	x_2	x_3	x_4	x_5
12	0	0	2	7	0
x_1 = 2	1	0	-3	2	0
x_2 = 2	0	1	5	-3	0
x_5 = -1	0	0	2	-1	1

Now apply dual simplex to solve

EC674 Study Sheet

GRAPHS

- **undirected graph**: $G=(N,E)$ consists of a set of nodes N and a set (E) of undirected arcs or **edges**, where an edge e is an unordered pair of distinct nodes. (self arcs not allowed)
- **degree** of a node is how many arcs incident to it. **degree** of an undirected graph = max of the degrees of its nodes
- **walk** from node i_1 to i_k in an undirected graph is a finite sequence of nodes $i_1, i_2, \dots, i_k$ s.t. $\{i_k, i_{k+1}\} \in E$, $k=1,2,\dots,k-1$
- **path**: walk w/ no repeated nodes
- **cycle**: walk $i_1,\dots,i_k$ s.t. the nodes $i_1, i_2, \dots, i_{k-1}$ are distinct & $i_1 = i_k$ (must have 3+ distinct nodes)
- an undirected graph is **connected** if for every two distinct nodes there is a path b/t them
- **directed graph**: $G=(N,A)$ consists of a set N of nodes and a set A of directed arcs, where a directed arc is an ordered pair of distinct nodes (i,j) (self arcs again not allowed)
- arc (i,j) is **outgoing** from node i and **incoming** to node j (**incident** to both)

$I(i) = \{j\in N \mid (j,i)\in A\}$, $O(i) = \{j \in N \mid (i,j)\in A\}$

- a directed graph is **connected** if the corresponding undirected graph is connected.
- **walk**: a sequence $i_1,\dots,i_k$ of nodes w/ an associated sequence $a_1,\dots,a_{k-1}$ of arcs s.t. for $k=1,\dots,k-1$, we have either $a_k=(i_k,i_{k+1})$ (a forward arc) or $a_k=(i_{k+1},i_k)$ (a backward arc).
- **path**: a walk w/ all distinct nodes $i_1,\dots,i_k$ • **cycle**: $i_1,\dots,i_{k-1}$ distinct and $i_1 = i_k$
- a walk, path, or cycle is **directed** if it only contains forward arcs.
- **Tree**: a connected, undirected graph w/ no cycles. **leaf**: tree node w/ degree = 1
- **Theorem 7.1**: a) every tree w/ more than one node has at least one leaf
  b) An undirected graph is a tree iff it is connected & has $|N|-1$ arcs
  c) For any two distinct nodes i and j in a tree, $\exists$ a unique path from i to j.
  d) If we start with a tree and add one arc, the resulting graph contains exactly one cycle
- **spanning tree**: given a connected, undirected graph $G=(N,E)$, let E_1 be a subset of E s.t. $T=(N,E_1)$ is a tree
- **Theorem 7.2**: let $G=(N,E)$ be a connected undirected graph and let E_0 be some subset of the set E of arcs. Suppose the arcs in E_0 do not form any cycles. Then the set E_0 can be augmented to a set $E_1 \supseteq E_0$ s.t. (N,E_1) is a spanning tree of G.

Network Flow problem

- **network**: a directed graph $G=(N,A)$ w/ some additional numerical information, such as the numbers b_i representing the external supply to each node $i\in N$, nonnegative #s u_{ij} representing the capacity of each arc $(i,j)\in A$, & c_{ij} representing the cost per unit flow along arc (i,j)
- use the f_{ij} to denote the amount of flow through arc (i,j)
- **source node**: $b_i>0$ – **sink node**: $b_i<0$ $\quad \boxed{\sum_{i\in N} b_i = 0}$
- conditions imposed on flow variables:

$$\left[\, b_i + \sum_{j\in I(i)} f_{ji} = \sum_{j\in O(i)} f_{ij} \;\; \forall i\in N \qquad 0\le f_{ij}\le u_{ij} \;\; \forall (i,j)\in A \,\right]$$

- **flow**: any vector w/ components f_{ij}, $(i,j)\in A$ • **feasible flow**: flow satisfying both constraints
- General minimum cost network flow problem $\left[\min \sum_{(i,j)\in A} c_{ij} f_{ij} \text{ over all feasible flows}\right]$
- **uncapacitated problem**: $u_{ij}=\infty \;\forall (i,j)\in A$
  - ↳ problem is in standard form b/c we only have equality & nonnegativity constraints
- There are several variants of the NFP that can be shown to be equivalent to each other:
  a) every NFP is equivalent to the **transportation problem**
  b) every NFP can be transformed to one w/ exactly one source and one sink
  c) every NFP can be reduced to one w/o sources or sinks (**circulation problems**)
  d) **node capacities**: if we have an upper bound d_i on the flow that can enter a node i, split it into two nodes & let d_i be the capacity of the arc b/w them

  e) If we add lower bounds on the arc flows, it can be reduced to an equiv. prob where every d_{ij} lower bound $=0$.
- **NFP matrix-vector notation**
- assume $N=\{1,\dots,n\}$, m = # of arcs. Fix ordering of the arcs & let f be a vector of flows that results when the components f_{ij} are ordered accordingly
- **node-arc incidence matrix**: a $n\times m$ matrix (rows correspond to nodes, cols to arcs), (i,k)th entry: $\left[a_{ik} = \begin{cases} 1 & i \text{ the start node for the } k\text{th arc} \\ -1 & i \text{ the end node of the } k\text{th arc} \\ 0 & \text{else}\end{cases}\right]$ $\quad \boxed{Af=b}$
- **circulation**: any flow vector f that satisfies $Af=0$ (flow conservation w/in network and zero supply/demand)
- consider a cycle C. Let F, B be the set of forward + backward arcs in the cycle. The flow vector h^C with components $h^C_{ij} = \begin{cases} 1 & (i,j)\in F \\ -1 & (i,j)\in B \\ 0 & \text{else}\end{cases}$ is the **simple circulation** associated w/ C and $\boxed{Ah^C=0}$
- **cost of cycle** C: $c'h^C = \sum_{(i,j)\in F} c_{ij} - \sum_{(i,j)\in B} c_{ij}$
- If f a flow vector, C a cycle, θ a scalar, we say the flow vector $f+\theta h^C$ is obtained from f by **pushing** θ units of flow around C & the resulting cost change is $\theta\, c'h^C$.

Network Simplex Algorithm

: applied to the uncapacitated NFP $\left[\text{minimize } c'f \text{ s.t. } Af=b,\ f\ge 0\right]$

- Note we have m flow vars and n equality constraints, opposite of our previous conventions
- **Assumptions**: a) $\sum_{i\in N} b_i = 0$ b) G is connected → has linearly indep. rows!
- **truncated node-arc incidence matrix** $\tilde{A}$: $(n-1)\times m$ matrix consisting of the first $n-1$ rows of A.
- **tree solution**: a flow vector f that can be constructed by the following procedure
  a) Pick a set $T\subset A$ of $n-1$ arcs that form a tree when direction is ignored
  b) Let $f_{ij}=0 \;\forall (i,j)\notin T$
  c) Use the flow conservation equation $\tilde{A}f=\tilde{b}$ to determine the flow variables f_{ij} for $(i,j)\in T$

  A tree solution satisfying $f\ge 0$ is a **feasible tree solution**
- **Theorem 7.3**: Let $T\subset A$ be a set of $n-1$ arcs that form a tree when direction is ignored. Then the system of linear equations $\tilde{A}f=\tilde{b}$ and $f_{ij}=0 \;\forall (i,j)\notin T$ has a unique solution
- **Corollary 7.1**: If the graph G is connected, then $\tilde{A}$ has linearly independent rows
- Let B be the $(n-1)\times(n-1)$ matrix resulting if we keep the $n-1$ columns of $\tilde{A}$ that correspond to the arcs in T. B is a basis matrix for the problem, and the corresponding f is a basic solution
- **Theorem 7.4**: a flow vector is a basic solution iff it is a tree solution.
- every basis matrix is triangular when its rows and columns are suitably reordered.
- **Change of basis**: picking a nonbasic variable ≡ picking an arc $(i,j)\notin T$. Then the arc (i,j) w/ some of the arcs in T form a cycle.
  - choose orientation of the cycle s.t. (i,j) a forward arc; F, B the sets of forward & backward arcs in the cycle
  - we want to increase the value of f_{ij} to some θ so adjust old basic vars to satisfy flow conservation. Accomplish by pushing θ units of flow around the cycle

$$(**)\quad \hat{f}_{kl} = \begin{cases} f_{kl}+\theta & (k,l)\in F \\ f_{kl}-\theta & (k,l)\in B \\ f_{kl} & \text{else}\end{cases} \qquad \text{Set } \theta \text{ as large as possible: } \theta^* = \min_{(k,l)\in B} f_{kl},\ \infty \text{ if } B \text{ empty}$$

- cost change from change of basis: $\theta\left(\sum_{(k,l)\in F} c_{kl} - \sum_{(k,l)\in B} c_{kl}\right)$ **reduced cost of** f_{ij}: $\bar{c}_{ij} = \sum_{(k,l)\in F} c_{kl} - \sum_{(k,l)\in B} c_{kl}$
- **reduced cost (alternative formulation)**: $\bar{c}' = c' - p'\tilde{A}$, $p_n=0$, $p' = c_B' B^{-1}$ $\Rightarrow \bar{c}_{ij} = c_{ij} - (p_i - p_j)$ • To calculate p_i, note that: $p_n=0$, $p_i - p_j = c_{ij} \;\forall (i,j)\in T$

SIMPLEX METHOD FOR UNCAPACITATED NFPs

1. Iteration starts w/ BFS f associated w/ a tree T
2. To compute p, solve the system of equations $p_i-p_j=c_{ij}$, $p_n=0$ proceeding from the roots to the leaves
3. Compute reduced costs $\bar{c}_{ij} = c_{ij}-(p_i-p_j)$ of all arcs $(i,j)\notin T$. If they are all nonnegative, current solution is optimal and algorithm terminates; else choose some (i,j) w/ $\bar{c}_{ij}<0$ to be brought into the basis
4. The entering arc (i,j) and the arcs in T form a unique cycle. If all arcs in the cycle are oriented the same way as (i,j), optimal cost is $-\infty$ & algorithm terminates.
5. Let B be the set of arcs in the cycle that are oriented in the opposite direction from (i,j). $\theta^* = \min_{(k,l)\in B} f_{kl}$ & push θ^* units of flow around the cycle. Determine a new flow vector according to (**). Remove from basis an old basic variable whose new value $=0$.

- **Theorem 7.5**: consider an uncapacitated network flow problem and assume the underlying graph is connected
  a) For every basis matrix B, the matrix B^{-1} has integer entries
  b) If the supplies b_i are integer, every basic solution has integer coordinates
  c) If the cost coefficients c_{ij} are integer, then every dual basic solution has integer coords
- **Corollary 7.2**: Consider an uncapacitated NFP and assume optimal cost is finite.
  a) If all supplies b_i are integer, $\exists$ an integer optimal flow vector
  b) If all cost coefficients c_{ij} are integer, $\exists$ an integer optimal solution to the dual prob

SIMPLEX FOR CAPACITATED NFPs:

1. Iteration starts w/ BFS associated w/ tree T and a partition of the remaining arcs into two sets D, U s.t. $f_{ij}=d_{ij}$ for $(i,j)\in D$, $f_{ij}=u_{ij}$ for $(i,j)\in U$.
2. Solve system of equations for $p_1,\dots,p_n$ proceeding from root to leaves
3. Compute reduced costs $\bar{c}_{ij} = c_{ij}-(p_i-p_j)$ of all arcs $(i,j)\notin T$. If $\bar{c}_{ij}\ge 0 \;\forall (i,j)\in D$ and $\bar{c}_{ij}\le 0$ for all $(i,j)\in U$, current BFS is optimal, DONE!
4. Let (i,j) be an arc s.t. $\bar{c}_{ij}<0$ for $(i,j)\in D$ or $\bar{c}_{ij}>0$ for $(i,j)\in U$. This arc (i,j) together with T form a unique cycle. Choose orientation: If $(i,j)\in D$, (i,j) should be a forward arc. If $(i,j)\in U$, (i,j) should be a backward arc.
5. Let F, B be forward & backward arcs in the cycle.

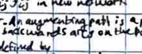
$$\theta^* = \min\left\{ \min_{(k,l)\in B} (f_{kl}-d_{kl}),\ \min_{(k,l)\in F}(u_{kl}-f_{kl}) \right\}. \text{ Compute new flow vector:}$$

$$\hat{f}_{kl} = \begin{cases} f_{kl}+\theta^* & \text{if } (k,l)\in F \\ f_{kl}-\theta^* & \text{if } (k,l)\in B \\ f_{kl} & \text{else}\end{cases} \qquad \text{update } T, D, U.$$

Maximum Flow problem

: given a directed graph $G=(N,A)$ and an arc capacity $u_{ij}\in[0,\infty]$ for each arc $(i,j)\in A$. Let s and t be source & sink nodes. We want to find the largest possible amount of flow that can be sent through the network from s to t.

- **Formulation**: max b_s s.t. $Af=b$, $b_t=-b_s$, $b_i=0 \;\forall i\ne s,t$, $0\le f_{ij}\le u_{ij}$
- **Decision variables**: f, b.
- **Reformulate into NFP**:
  - Let cost of every arc $=0$
  - New inf. capacity arc (t,s)
  - now just min $\sum_{(i,j)} c_{ij} f_{ij}$ in new network



$u_{ts}=\infty$, $c_{ts}=-1$

- **augmenting path**: let f be a feasible flow vector. An augmenting path is a path from s to t s.t. $f_{ij}<u_{ij}$ for all forward arcs and $f_{ij}>0 \;\forall$ backwards arcs on the path
- Amount of flow that can be no more than $\delta(P)$, defined by

$$\delta(P) = \min\left\{ \min_{(i,j)\in F}(u_{ij}-f_{ij}),\ \min_{(i,j)\in B} f_{ij}\right\} \quad F = \text{forward arcs},\ B = \text{backward arcs}$$

Ford-Fulkerson Alg

1. Start w/ a feasible flow f
2. Search for an augmenting path
3. If no augmenting path can be found alg. terminates
4. If augmenting path is found,
   a) If $\delta(P)<\infty$, push $\delta(P)$ units of flow along P & go back to step 2
   b) If $\delta(P)=\infty$, terminate

- **Theorem 7.8**: Suppose that all arc capacities u_{ij} are integer or infinite & the FF algorithm is initialized w/ an integer flow vector. Then the arc flow vars remain integer throughout the algorithm and, if the optimal val is finite, algorithm terminates in a finite # of steps.

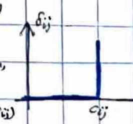
Labeling Algorithm:

1. The algorithm is initialized w/ $I=\{s\}$, s the only labeled node
2. Iteration starts w/ set I of labeled, unscanned nodes. If $t\in I$ or $I=\emptyset$, alg terminates. Else, choose $i\in I$ to be scanned & remove from I. Examine all arcs of form (i,j) or (j,i)
3. If $(i,j)\in A$, $f_{ij}<u_{ij}$ and j unlabeled, label j and add to I.
4. If $(j,i)\in A$, $f_{ji}>0$, j unlabeled, label j and add to I

- **Theorem 7.9**: The labeling algorithm runs in $O(|A|)$. At termination t labeled iff $\exists$ an augmenting path
- **cut**: a subset S of the nodes N s.t. $s\in S$ and $t\notin S$.
- **capacity** $C(S)$: Σ of capacities of arcs that cross from S to $\bar{S}$, i.e.

$$C(S) = \sum_{\{(i,j)\in A \mid i\in S,\, j\notin S\}} u_{ij}$$

- **Theorem 7.10**: a) If FF algorithm terminates b/c no aug path found, current flow optimal
  b) **Max-flow min-cut theorem**: value of max flow = min cut capacity
- **Network flow duality**: $\min c'f$ s.t. $Af=b$, $f\ge 0$ $\iff$ $\max p'b$ s.t. $p_i-p_j\le c_{ij} \;\forall (i,j)\in A$
- **Sensitivity**: increasing b_i by ε, resulting cost change is $\varepsilon(p_i-p_j)$, so in the absence of degen., p_i-p_j = marginal cost of sending an additional unit of flow from node i to node j.
- **Theorem 7.11 (Complementary slackness)**: For any uncapacitated NFP, the following are equiv:
  a) The vectors f and p are optimal solutions to the primal/dual
  b) f satisfies flow conservation equation $Af=b$, and for every arc (i,j) the pair (p_i-p_j, f_{ij}) satisfies the relation



arc should not carry flow if $p_i-p_j<c_{ij}$

Nonlinear Programming (NLP)

- euclidean norm: $\|x\| = \sqrt{x'x}$ • open ball around x w/ radius r: $\{y \mid \|y-x\|<r\}$
- $A\subset\mathbb{R}^n$ is compact iff closed & bounded
- consider function $f: A\to\mathbb{R}$
  - continuous at $x\in A$ if $\lim_{y\to x} f(y) = f(x)$
  - right cts: $\lim_{y\downarrow x} f(y) = f(x)$ • left cts: $\lim_{y\uparrow x} f(y) = f(x)$
  - lower semicts: $f(x)\le \liminf_{k\to\infty} f(x_k)$ $\forall x_k\to x$ (sequences x_k)
  - upper semicts: $f(x)\ge \limsup_{k\to\infty} f(x_k)$ $\forall x_k \to x$
  - coercive: if $\lim_{k\to\infty} f(x_k)=\infty$ for every sequence satisfying $\|x_k\|\to\infty$
- **Weierstrass Theorem**: Let $A\subset\mathbb{R}^n$, A closed & nonempty. Let $f: A\to\mathbb{R}$ be lower semicontinuous at all $x\in A$.
  a) If A compact, $\exists\, \bar{x}\in A$ s.t. $f(\bar{x})\le f(x) \;\forall x\in A$
  b) If f coercive, $\exists\, \bar{x}\in A$ s.t. $f(\bar{x})\le f(x) \;\forall x\in A$
- **Gradient**: $f:\mathbb{R}^n\to\mathbb{R} \Rightarrow \nabla f(x) = \left(\frac{\partial f(x)}{\partial x_1},\dots,\frac{\partial f(x)}{\partial x_n}\right)'$
- **Hessian**: $f:\mathbb{R}^n\to\mathbb{R}\Rightarrow \nabla^2 f(x) = \left[\frac{\partial^2 f(x)}{\partial x_i \partial x_j}\right]$
- **Taylor expansion**: $f(x+y) = f(x) + y'\nabla f(x) + \frac{1}{2} y'\nabla^2 f(x) y + o(\|y\|^2)$
- **UNCONSTRAINED OPTIMIZATION PROBLEM**: min $f(x)$ s.t. $x\in\mathbb{R}^n$
  - x^* a **local minimum** if $\exists \varepsilon>0$ s.t. $f(x^*)\le f(x) \;\forall x$ with $\|x-x^*\|<\varepsilon$
  - x^* a **global minimum** if $f(x^*)\le f(x)\;\forall x\in\mathbb{R}^n$
- **PROPOSITION**: Let x^* be an unconstrained local min of $f:\mathbb{R}^n\to\mathbb{R}$ cts diff'ble in an open set. Then $\nabla f(x^*)=0$ (1st order). If f twice cts diff'ble, $\nabla^2 f(x^*)$ positive semidefinite (2nd order) (**NECESSARY CONDITIONS**)
- **Necessary & sufficient conditions for convex functions**: given $f: C\to\mathbb{R}$ convex
  a) A local min of f over C is also a global min over C. If f strictly convex, at most one global min
  b) If f is convex, C is open, $\nabla f(x^*)=0$ is nec & suff for x^* to be a global min
- **sufficient conditions**: let $f:\mathbb{R}^n\to\mathbb{R}$ be twice cts diff'ble, $\nabla f(x^*)=0$, $\nabla^2 f(x^*)$ positive definite. Then x^* is a strict unconstrained local min of f; $\exists\gamma>0, \varepsilon>0$ s.t. $f(x)\ge f(x^*) + \frac{\gamma}{2}\|x-x^*\|^2$

Gradient methods:

- generic: $x^{k+1} = x^k + \alpha^k d^k$ s.t. if $\nabla f(x^k)\ne 0$ and d^k chosen so $\nabla f(x^k)' d^k < 0$
- Interesting class of gradient method: $x^{k+1} = x^k - \alpha^k D^k \nabla f(x^k)$
  - steepest descent: $D^k = I$ • Newton's method: $D^k = (\nabla^2 f(x^k))^{-1}$
  - Diagonally scaled steepest descent: $D^k = \operatorname{diag}(\dots)$
  - Modified Newton method
- Least squares problem: $\min f(x) = \frac{1}{2}\|g(x)\|^2 = \frac{1}{2}\sum_i g_i(x)^2$
  - Note that $\nabla f = \nabla g(x)\, g(x)$
  - Gauss-Newton method for least squares: $x^{k+1} = x^k - \alpha^k (\nabla g(x^k)\nabla g(x^k)')^{-1}\nabla g(x^k) g(x^k)$

Stepsize selection:

- minimization rule: $f(x^k+\alpha^k d^k) = \min_{\alpha\ge 0} f(x^k+\alpha d^k)$
- limited minimization rule: $f(x^k+\alpha^k d^k) = \min_{\alpha\in[0,s]} f(x^k+\alpha d^k)$
- constant stepsize: $\alpha^k = s$ • diminishing stepsize: $\alpha^k\to 0$ w/ $\sum_{k=0}^\infty \alpha^k = \infty$

Lagrange Multiplier Theory: we consider problems w/ equality constraints

min $f(x)$ s.t. $h_i(x)=0$, $i=1,\dots,m$

- **Necessary optimality conditions**: Let x^* be a local min. Assume x^* is regular. Then $\exists$ a unique Lagrange multiplier vector λ^* s.t. $\nabla f(x^*) + \sum_{i=1}^m \lambda_i^* \nabla h_i(x^*) = 0$

  If f, h are twice continuously differentiable

  $y'\left(\nabla^2 f(x^*) + \sum_{i=1}^m \lambda_i^* \nabla^2 h_i(x^*)\right) y \ge 0 \;\forall y\in V(x^*)$, where $V(x^*)$ is the subspace of first order feasible directions $\{y \mid \nabla h_i(x^*)'y = 0,\ i=1,\dots,m\}$
- **sufficient optimality conditions**
  - define the Lagrangian function $L(x,\lambda) = f(x) + \sum_{i=1}^m \lambda_i h_i(x)$

  Let x^*, λ^* satisfy $\nabla_x L(x^*,\lambda^*)=0$, $\nabla_\lambda L(x^*,\lambda^*)=0$

  $y'\nabla^2_{xx} L(x^*,\lambda^*) y > 0 \;\forall y\ne 0$ s.t. $\nabla h(x^*)'y=0$

  Then x^* is a strict local min; $\exists \gamma>0,\varepsilon>0$ s.t. $f(x)\ge f(x^*)+\frac{\gamma}{2}\|x-x^*\|^2$ $\forall x$ with $h(x)=0$, $\|x-x^*\|<\varepsilon$

Problems w/ equality & inequality constraints:

min $f(x)$ s.t. $h_1(x)=0,\dots,h_m(x)=0$, $g_1(x)\le 0,\dots,g_r(x)\le 0$, f,h,g cts diff'ble in an open set containing the feasible set

- Refine the **Lagrangian function**: $L(x,\lambda,\mu) = f(x) + \sum_{i=1}^m \lambda_i h_i(x) + \sum_{j=1}^r \mu_j g_j(x)$
- Let $A(x) = \{j \mid g_j(x)=0\}$
- **Karush-Kuhn-Tucker necessary optimality conditions**: Let x^* be a regular local minimum. Then there exist unique Lagrange multiplier vectors λ^*, μ^* s.t. $\nabla_x L(x^*,\lambda^*,\mu^*)=0$, $\mu_j^*\ge 0$, $\mu_j^*=0 \;\forall j\notin A(x^*)$

  If f,h,g twice diff'ble $y'\nabla^2_{xx}L(x^*,\lambda^*,\mu^*)y\ge 0\;\forall y\in V(x^*)$ where $V(x^*) = \{y \mid \nabla h_i(x^*)'y=0,\ \nabla g_j(x^*)'y=0,\ j\in A(x^*)\}$
- **Sufficient conditions**: $\nabla_x L(x^*,\lambda^*,\mu^*)=0$, $h(x^*)=0$, $g(x^*)\le 0$, $\mu_j^*\ge 0$, $\mu_j^*=0\;\forall j\notin A(x^*)$

  $y'\nabla^2_{xx}L(x^*,\lambda^*,\mu^*)y>0\;\forall y\ne 0$ … Then x^* a strict local minimum of the constrained problem

Barrier methods: min $f(x)$ s.t. $x\in X$, $g_j(x)\le 0$

- Define the interior of the feasible set $S=\{x\in X \mid g_j(x)<0\}$
- **Barrier function**: for $x\in S$: $B(x) = -\sum_j \ln(-g_j(x))$, $B(x) = -\sum_j \frac{1}{g_j(x)}$
- **Barrier method**: $x^k = \arg\min_{x\in S}\{f(x)+\varepsilon^k B(x)\}$, where $\{\varepsilon^k\}$ is a sequence converging to 0 and satisfying $0<\varepsilon^{k+1}<\varepsilon^k$, $k=0,1,\dots$
- **THEOREM**: Every limit point of $\{x^k\}$ is a global min of the constrained problem

Shortest Path Problem

: Given graph $G=(N,A)$ w/ n nodes, m arcs. Each arc (i,j) has a cost c_{ij} (allowed to be negative), also called length

- **length** of a walk is the sum of the lengths of its arcs
- a path from i to j is **shortest** if it has the min length of all paths w/ same origin & destination.
- **Shortest walk ≠ shortest path** (negative length cycle we can have walk w/ $-\infty$ length)
- length of any path is bounded by nC, $C=\max|c_{ij}|$
- If all cycles have nonnegative length, a shortest path = shortest walk
- Any cycle in a shortest walk must have 0 length, by removing it we get a shortest path
- **all-to-one shortest path**: find the shortest path from all origins to one dest.
- **all-pairs shortest path**: find shortest path for all possible origin-destination pairs
- **tree rooted at node n**: all $n-1$ arcs assigned directions so we have a directed path from every node i to n
- **tree of shortest paths**: a tree rooted at node n s.t. for every $i\ne n$, the path from i to n along the tree is a shortest path
- **Theorem 7.17**: Consider the shortest path problem in a directed graph w/ n nodes and the associated NFP (each node except n a source node w/ unit supply, n a sink node w/ demand $(n-1)$). Assume $\exists$ a path to n from every other node and node n has no outgoing arcs.
  a) If $\exists$ a negative length cycle, optimal cost in network flow problem $=-\infty$
  b) Suppose all cycles have nonnegative length. If a feasible tree solution is optimal, corresponding tree is a tree of shortest paths
  c) Suppose all cycles have nonnegative length. If we fix $p_n=0$, dual problem's unique solution p_i is the shortest path length from node i to n
- **Bellman's equation**: $p_i = \min_{k\in O(i)}\{c_{ik}+p_k\}$
- **Principle of optimality**: If a shortest path from i to n goes through an intermediate node k, the portion of the path from k to n is also a shortest path.
- If cycles have nonneg. length, can only assert that the shortest path lengths are a solution to Bellman's equation.
- **Bellman-Ford Algorithm**: $p_i(t)$ = shortest walk from i to n using t arcs.

  $p_i(1) = c_{in}$, $p_i(1)=\infty$ if no such walk exists; $p_n(t)=0$, $p_i(0)=\infty \;\forall i\ne n$

  Note that $p_i(t+1)\le p_i(t)$ and

  $p_i(t+1) = \min_{k\in O(i)}\{c_{ik}+p_k(t)\}$, $t=1,\dots,n-1$

  $p_i(n-1) = p_i^*$
- **Theorem 7.19**: Suppose that $c_{ij}\ge 0$ for all i,j. Let $l\ne n$ be s.t. $c_{ln} = \min_{i\ne n} c_{in}$. Then $p_l^* = c_{ln}$ and $p_l^*\le p_k^*$ for all $k\ne n$

Dijkstra Alg:

1. Find a node $l\ne n$ s.t. $c_{ln}\le c_{in}\;\forall i\ne n$ set $p_l^* = c_{ln}$
2. For every node $i\ne l, n$, set $c_{in} = \min\{c_{in}, c_{il}+c_{ln}\}$
3. Remove node l from the graph and apply the same steps to the new graph

Dantzig-Wolfe Decomposition algorithm

$$\begin{array}{ll} \min & c_1'x_1 + c_2'x_2 \\ \text{s.t.} & D_1x_1 + D_2x_2 = b_0 \\ & F_1x_1 = b_1 \\ & F_2x_2 = b_2 \\ & x_1, x_2 \ge 0 \end{array}$$

1. Iteration begins w/ m_0+2 extreme points and extreme rays of P_1, P_2 which lead to a BFS to the master problem, the corresponding inverse basis B^{-1} and the dual vector $p' = (q', r_1, r_2) = c_B'B^{-1}$
2. Form and solve two subproblems. If optimal costs are no smaller than r_1, r_2 respectively, algorithm terminates.
3. If optimal cost in ith subproblem is $-\infty$ we obtain an extreme ray w_i associated w/ a variable θ_i whose reduced cost is negative. This variable can enter the basis of the master problem
4. If the optimal cost for the ith subproblem is finite and $<r_i$, we obtain an extreme point x_i^j associated w/ variable λ_i^j whose reduced cost is negative. This variable can enter the basis in the master problem
5. Having chosen a variable to enter the basis, generate the column associated w/ that variable, carry out an iteration of the revised simplex method for the master, update B^{-1}, p.

$P_i = \{x_i\ge 0 \mid F_i x_i = b_i\}\ne\emptyset$

By resolution theorem: $x_i = \sum_{j\in J_i}\lambda_i^j x_i^j + \sum_{k\in K_i}\theta_i^k w_i^k$, $\sum_{j\in J_i}\lambda_i^j = 1$

Master Problem:

$$\min \sum_{j\in J_1}\lambda_1^j c_1'x_1^j + \sum_{k\in K_1}\theta_1^k c_1'w_1^k + \sum_{j\in J_2}\lambda_2^j c_2'x_2^j + \sum_{k\in K_2}\theta_2^k c_2'w_2^k$$

$$\text{s.t. } \sum_{j\in J_1}\lambda_1^j\begin{bmatrix} D_1x_1^j \\ 1 \\ 0\end{bmatrix} + \sum_{j\in J_2}\lambda_2^j\begin{bmatrix} D_2x_2^j\\ 0\\ 1\end{bmatrix} + \sum_{k\in K_1}\theta_1^k\begin{bmatrix}D_1w_1^k\\0\\0\end{bmatrix} + \sum_{k\in K_2}\theta_2^k\begin{bmatrix}D_2w_2^k\\0\\0\end{bmatrix} = \begin{bmatrix}b_0\\1\\1\end{bmatrix}$$

$\lambda_i^j\ge 0$, $\theta_i^k\ge 0$

Only m_0+2 constraints, huge # of vars

Subproblems: Let B basis, $p' = [q'\ r_1\ r_2]$

Reduced cost of λ_i^j: $(c_i' - q'D_i)x_i^j - r_i$

Reduced cost of θ_i^k: $(c_i'-q'D_i)w_i^k$

Prob #1: min $(c_1'-q'D_1)x_1$ s.t. $x_1\in P_1$ $\qquad$ Prob #2: min $(c_2'-q'D_2)x_2$ s.t. $x_2\in P_2$

Integer Programming

: We consider problems of the form

$$\begin{array}{ll}\min & c'x + d'y \\ \text{s.t.} & Ax + By = b \\ & x, y\ge 0 \\ & x \text{ integer}\end{array} \quad \text{(mixed IP)}$$

- Assume matrices & vectors are integers
  a) If no continuous variable (y) then IP
  b) If all variables 0-1 then ZOIP

Cutting plane algorithm

1. Solve the linear programming relaxation. Let x^* be an optimal solution
2. If x^* integer STOP
3. Else add linear inequality constraint to LP problem that all integer solutions to IP problem satisfy, but x^* does not. Repeat

Gomory Cut: $x_{B(i)} + \sum_{j\in N}\lfloor \bar{a}_{ij}\rfloor x_j \le \lfloor \bar{a}_{i0}\rfloor$

where $\bar{a}_j = (B^{-1}A_j)$, $\bar{a}_{i0} = (B^{-1}b)_i$; N = indices of nonbasic vars

Categories of IP algorithms:

a) **Exact algorithms**: guaranteed to find opt solution but may take exponential # of iterations (cutting plane, branch + bound, DP)
b) **approximation methods**: provide suboptimal solution together w/ a bound on degree of suboptimality
c) **heuristic algs**: suboptimal solution w/o guarantee on its quality (local search, simulated annealing)

Branch + Bound Algorithm

1. Select active subproblem F_i
2. If subproblem is infeasible, delete it, otherwise compute $b(F_i)$
3. If $b(F_i)\ge U$ delete subproblem
4. If $b(F_i)<U$ obtain opt sol of subproblem or break the corresponding subproblem into further subproblems which are added to the list of active subproblems

F = feasible set of solutions, partition into F_i

- We assume efficient algorithm s.t. for every F_i we get a lower bound $b(F_i)$
- **Zero-one knapsack problem alg**: $W_{i+1}(u) = \min\{W_i(u),\ W_i(u-c_{i+1})+w_{i+1}\}$

  $W_i(u)$ = least possible weight you need to accumulate to attain total value of u
  - Can be solved in $O(n^2 c_{max})$

Integer Programming Duality

- Consider the IP problem $\quad$ minimize $c'x$ s.t. $Ax\ge b$, $Dx\ge d$, x integer $\;(*)$ $\quad \boxed{Z_{LP}\le Z_D\le Z_{IP}}$
- Assume optimizing over the set $X=\{x \text{ integer} \mid Dx\ge d\}$ can be done efficiently. Introduce a dual variable for every constraint in $Ax\ge b$. Let $p\ge 0$ be a vector of dual vars (Lagrange multipliers) same dimension as b. For a fixed p, consider problem minimize $c'x + p'(b-Ax)$ s.t. $x\in X$, denote the optimal cost by $Z(p)$
- We say that we **relax** or **dualize** the constraints $Ax\ge b$. $Z(p)$ lower bound
- **Lemma 11.1**: If the problem (original) (*) has an optimal sol. and if $p\ge 0$, then $Z(p)\le Z_{IP}$
- Introduce the problem $\max_{p\ge 0} Z(p)$, **Lagrangean dual**
- **Theorem 11.2**: $Z_D\le Z_{IP}$
- **Theorem 11.3**: Assume that the system of linear inequalities $Dx\ge d$ has a feasible solution for a vector d and matrix D have integer entries. Let $X=\{x \text{ int} \mid Dx\ge d\}$. Let $CH(X)$ be the convex hull of the set X. Then $CH(X)$ a polyhedron
- **Theorem 11.4**: The optimal value Z_D of the Lagrangean dual is equal to the optimal cost of the following LP problem:

  minimize $c'x$ s.t. $Ax\ge b$, $x\in CH(X)$
- **Corollary 11.1**:
  a) We have $Z_{LP}=Z_D$ for all cost vectors c iff $CH(X\cap\{x\mid Ax\ge b\}) = CH(X)\cap\{x\mid Ax\ge b\}$
  b) We have $Z_{IP}=Z_D$ for all cost vectors c if $CH(X) = \{x \mid Dx\ge d\}$ (integer extreme points)

Interior point methods

- **Affine scaling algorithm**

1. (Initialization) $x^0>0$, $k=0$
2. (Compute dual estimates and reduced costs) Given $x^k>0$

   $X_k = \operatorname{diag}(x_1^k,\dots,x_n^k)$

   $p^k = (AX_k^2A')^{-1}AX_k^2c$

   $r^k = c - A'p^k$
3. (Optimality check) If $r^k\ge 0$ and $e'X_kr^k<\varepsilon$, STOP (near-optimal)
4. (Unboundedness) $-X_k^2r^k\ge 0$, optimal cost $=-\infty$
5. Update $x^{k+1} = x^k - \beta\frac{X_k^2r^k}{\|X_kr^k\|}$

Nonlinear programming HW #7

Problem 1: Consider $f_1(x) = x_1^2 + x_2^2$ and $f_2(x) = 10x_1^2 + x_2^2$. Implement steepest descent method w/ constant stepsize and minimization stepsize.

$\nabla f_1(x) = [2x_1, 2x_2]$, $\nabla f_2(x) = [20x_1, 2x_2]$

Direction of steepest descent method at step k is $d^k = -\nabla f(x^k) = [-2x_1^k, -2x_2^k]$

minimization stepsize at k: $\alpha^k = \arg\min_\alpha f(x^k + \alpha d^k)$

Problem 2: Find the minimum of the function $f(x) = (x_2 - x_1^2)^2$ among all points satisfying necessary conditions for an extremum.

$\nabla f(x) = [-2(x_2 - x_1^2)\cdot 2x_1,\ 2(x_2 - x_1^2)]'$

set $\nabla f(x) = 0 \Rightarrow x_2^2 = x_1 \ldots \Rightarrow x_2 = x_1^2$

For all points of the form $x_k^* = [x_1^*, x_2^*]$ we have $\nabla^2 f(x_k) = \begin{bmatrix} 2 & -4x_1 \\ -4x_1 & 8x_1^2 \end{bmatrix}$

For necessary conditions to hold, need $\nabla^2 f(x_k)$ to be pos semidef. Take $d'\nabla^2 f(x_k) d$, assume $d = [0, d_2]$, $d_2 \neq 0$.

$d'\nabla^2 f(x_k) d = 8x_1^2 d_2^2 \geq 0$

For arb. vector d, scale to $(1, d_2)$: $d'\nabla^2 f(x_k) d = 2(2x_1 d_2 - 1)^2 \geq 0$

So $\nabla^2 f(x_k)$ pos semidef $\forall x_k$, but not pos definite so sufficient conditions not satisfied.

Still, $f(x) \geq 0$, $f(x_k) = 0$, so we conclude the points $x_k = [x_1^k, x_2^k]$ are local minima.

Problem 3: Consider the problem min $f(x) = x_1 + x_2$ s.t. $h_1(x) = (x_1 - 1)^2 + x_2^2 - 1 = 0$, $h_2(x) = (x_1 - 2)^2 + x_2^2 - 4 = 0$

Find the local minimum. Do LaGrange multipliers exist?

$x^* = (0,0)$, $\nabla f(x^*) = (1,1)$, $\nabla h_1(x^*) = (-2, 0)$, $\nabla h_2(x^*) = (-4, 0)$

We cannot express $\nabla f(x^*)$ cost gradient as a linear combo of the constraint vectors. So, 1st order Lagrange multiplier condition can't hold for any λ_1^*, λ_2^*. LAGRANGE MULTIPLIERS DO NOT EXIST.

Why? The nullspace of 1st order feasible directions $V(x^*) = \{y \mid y_1 = 0, y_2 \in \mathbb{R}\}$ has a larger dimension than the true set of feasible directions. For a Lagrange multiplier to exist $\nabla f(x^*)$ must be $\perp$ to $V(x^*)$. PROBLEM: constraint gradients at x^* not linearly independent.

Problem 4: consider min $(x_1 - 9/4)^2 + (x_2 - 2)^2$ s.t. $x_2 - x_1^2 \geq 0$, $x_1 + x_2 \leq 6$, $x_1, x_2 \geq 0$

a) Write the Kuhn-Tucker optimality conditions & verify they are satisfied at $\bar{x} = (3/2, 9/4)$

In canonical form: min $(x_1 - 9/4)^2 + (x_2 - 2)^2$ s.t. $x_1^2 - x_2 \leq 0$, $-x_1 \leq 0$, $x_1 + x_2 - 6 \leq 0$, $-x_2 \leq 0$

KT first order optimality conditions given as:

$2x_1 - 9/2 + 2\mu_1 x_1 + \mu_2^* - \mu_3^* = 0$

$2x_2 - 4 - \mu_1^* + \mu_2^* - \mu_4^* = 0$

$\mu_j^* \geq 0$, $\mu_j^* = 0\ \forall j \notin A(x^*)$

$y'\begin{bmatrix} 2\mu_1 + 2 & 0 \\ 0 & 2 \end{bmatrix} y \geq 0\ \forall y \in V(x^*)$

where $V(x^*) = \{y \mid \nabla g_j(x^*)'y = 0,\ \forall j \in A(x^*)\}$

For $\bar{x} = (3/2, 9/4)$ only the first constraint is active so $\mu_2^* = \mu_3^* = \mu_4^* = 0$ so $\mu_1 = 1/2$ from 1st order conditions.

At $\bar{x}$ 2nd order conditions become $3y_1^2 + 2y_2^2 \geq 0\ \forall y \in V(\bar{x})$ where $V(\bar{x}) = \{y \mid 3y_1 - y_2 = 0\}$. satisfied ✓

b) Show $\bar{x}$ a global optimal solution

Since $\nabla^2 f(x) = \begin{bmatrix} 2 & 0 \\ 0 & 2 \end{bmatrix}$ is positive def for all x, $f(x)$ strictly convex and $\bar{x}$ global optimum.

HW #9 Network Flow

Problem #1: Each of n teams plays a total of k games. Assume every game ends in a win or loss and let x_i represent the total # of wins of team i. Let X be the set of all possible outcome vectors $(x_1, \ldots, x_n)$. We want to determine whether an arbitrary vector belongs to X (is a possible tournament outcome vector). Provide a network flow formulation of this problem.

Introduce nodes $T_1, \ldots, T_n$ corresponding to different teams. These are supply nodes and node T_i has supply x_i. For each unordered pair (i,j) of teams, introduce a node G_{ij}. These are demand nodes w/ demand k, the total # of games played between the two teams. Since the total # of games = the total # of wins, assume $\sum_{i=1}^n x_i = n(n-1)/2$.

# of games btw i,j that i wins.

This NFP is feasible iff $(x_1, \ldots, x_n)$ is a possible outcome vectors.

Problem #2: (Equivalence of capacitated NF and transportation problems). Consider a capacitated NFP defined by a graph $G = (N, A)$ and the data u_{ij}, c_{ij}, b_i. Assume each arc has finite capacity u_{ij}. Construct the related transportation problem: for every arc $(i,j) \in A$ we form a source node in the transportation problem w/ supply u_{ij}. For every node $i \in N$ construct a sink node with demand $\sum_{\{k \mid (i,k) \in A\}} u_{ik} - b_i$. At every supply node (i,j) there are two outgoing infinite capacity arcs: one goes to demand node i and its cost coefficient is 0, the other goes to demand node j and its cost coefficient is c_{ij}.

Show that there is a 1-1 correspondence b/w feasible flows in these two problems with the same cost.

Consider a feasible flow f in the original network problem w/ components f_{ij}. Build a corresponding flow g in the transportation problem: for each original arc (i,j), let $g_{(i,j)j} = f_{ij}$, $g_{(i,j)i} = u_{ij} - f_{ij}$. Check feasibility of g. For each source u_{ij},

$g_{(i,j)j} + g_{(i,j)i} = f_{ij} + (u_{ij} - f_{ij}) = u_{ij}$

For each sink i: $\sum_{\{k:(i,k)\in A\}} g_{(i,k)i} + \sum_{\{k:(k,i)\in A\}} g_{(k,i)i} = \sum_{\{k \mid (k,i) \in A\}} (u_{ik} - f_{ik}) + \sum_{\{k:(k,i)\in A\}} f_{ki} = \sum_{\{k \mid (k,i) \in A\}} u_{ik} - b_i$

So, g is a feasible solution to the trans. prob w/ same cost!

For the converse, let g be a feasible solution to the transportation problem. We build a solution to the NFP as follows: for each arc $(i,j) \in A$, let $f_{ij} = g_{(i,j)j}$. Then $f_{ij} \leq u_{ij}$. For each node i:

$\sum_j f_{ij} - \sum_j f_{ji} = \sum_{\{j:(j,i)\in A\}} g_{(j,i)i} - \sum_{\{j:(i,j)\in A\}} g_{(i,j)j} = \sum_{\{j:(j,i)\in A\}} g_{(j,i)i} - \sum_{\{j:(i,j)\in A\}} (u_{ij} - g_{(i,j)i})$

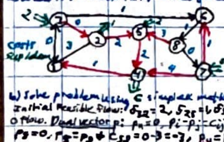
Taking into account the value of the demand at sink i, this expression is the same as

$\sum_{\{k:(i,k)\in A\}} u_{ik} - b_i - \sum_{\{j:(i,j)\in A\}} u_{ij} = -b_i$.

So f a feasible solution to the trans. problem w/ the same cost!

Problem #3: Consider the uncapacitated flow network problem.

a) What is the matrix A corresponding to this problem?



$$\begin{bmatrix} 1 & -1 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ -1 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & -1 & 0 & -1 & 0 & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 & 0 & 0 & 1 & 0 & 0 & -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & -1 & 0 & 0 & 1 & 1 & 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & -1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 1 & 1 & 0 \end{bmatrix}$$

b) Solve problem using simplex method starting with the red tree.

Initial feasible flow: $f_{32} = 2$, $f_{25} = 1$, $f_{54} = 3$, $f_{41} = 2$, $f_{74} = 5$, $f_{76} = 0$, $f_{85} = 0$. All other arcs have 0 flow. Dual vector p: $p_4 = 0$, $p_i - p_j = c_{ij}$

$p_8 = 0$, $p_5 = p_8 + c_{58} = 0 - 3 = -3$, $p_4 = p_5 + c_{45} = -3 - 2 = -5$, $p_7 = p_4 + c_{74} = -5 + 4 = -1$

$p_1 = p_4 + c_{14} = -5 - 1 = -6$, $p_2 = p_5 + c_{25} = -3 + 2 = -1$, $p_3 = p_2 - c_{32} = -1 + 0 = -1$, $p_6 = p_7 - 1 = -2$

Reduced costs of other arcs: $\bar{c}_{ij} = c_{ij} - (p_i - p_j)$

$\bar{c}_{13} = 0 - (-6 - (-1)) = 5$, $\bar{c}_{21} = 3 - (-1 - (-6)) = -2$, $\bar{c}_{36} = 1 - (-1 - (-2)) = 0$

$\bar{c}_{68} = 1 - (-2 - 0) = 3$, $\bar{c}_{87} = 0 - (0 - (-1)) = -1$

Arc $(2,1)$ has negative reduced costs so bring into basis. We can push $\theta^* = 1$ in the cycle $(2 \to 1 \to 4 \to 5 \to 2)$, limited by backward arc $(5,2)$. New flow vector:

$f_{21} = 1$, $f_{32} = 2$, $f_{41} = 1$, $f_{54} = 2$, $f_{74} = 5$, $f_{76} = 0$, $f_{85} = 0$, arc $(2,5)$ exits the basis.

New ps: $p_8 = 0$, $p_7 = -1$, $p_6 = -2$, $p_5 = -3$, $p_4 = -5$, $p_3 = -7$, $p_2 = -3$, $p_1 = -6$

New reduced costs: $\bar{c}_{13} = 3$, $\bar{c}_{25} = 2$, $\bar{c}_{36} = 2$, $\bar{c}_{68} = 3$, $\bar{c}_{87} = -1$

Arc $(7,8)$ has negative reduced cost, bring into basis. $\theta^* = 0$ can be pushed in the cycle $(8 \to 7 \to 4 \to 5 \to 8)$, limited by backward arc $(5,8)$. So, flow vector stays the same, but arc $(8,5)$ exits the basis. New p: $p_8 = 0$, $p_7 = 0$, $p_6 = -1$, $p_5 = -2$, $p_4 = -4$, $p_3 = -2$, $p_2 = -2$, $p_1 = -5$.

New reduced costs: $\bar{c}_{13} = 3$, $\bar{c}_{25} = 2$, $\bar{c}_{36} = 2$, $\bar{c}_{68} = 2$, $\bar{c}_{85} = 1$.

So optimal flow is above (highlighted), & optimal cost = 28

Problem #4: Consider the maximum flow problem.

a) Let p_i be a price variable associated w/ flow cons. constraint at node i. Let q_{ij} be the price variable associated w/ the capacity constraint at arc (i,j). Write a minimization problem w/ vars p_i, q_{ij} whose dual is the max flow problem.

Number the nodes s.t. source is the first node and sink is last. Let $e = (-1, 0, \ldots, 0, 1)$ and consider:

maximize $0'f + b_s$ s.t. $Af + eb_s = 0$, $f \leq u$, $f \geq 0$

minimize $0'p + u'q$ s.t. $p'A + q' \geq 0$, $e'p = 1$, $q \geq 0$, p free

The max problem is the maximum flow problem and is the dual of the second. Min problem only involves constraints $p_i - p_j + q_{ij} \geq 0$, $p_t - p_s = 1$, so only differences $p_i - p_j$ matter & we can set $p_s = 0$, $p_t = 1$.

b) Show that the optimal value in the min problem = min cut capacity & prove max-flow min-cut theorem. Consider an integer valued optimal solution to the min problem. q_{ij} nonneg. So set q_{ij} to smallest allowed value, $\max\{0, p_j - p_i\}$. Define a new solution defined by $\bar{p}_i = \min\{1, p_i\}$, $\bar{q}_{ij} = \max\{0, \bar{p}_j - \bar{p}_i\}$. Clearly $\bar{q}_{ij} \leq q_{ij}$. Since u_{ij} nonneg, new solution at least as good as preceding one & optimal. Repeat by letting $\bar{p}_i = \max\{0, \bar{p}_i\}$. Similarly show that it also optimal.

We just established $\exists$ an optimal solution s.t. $p_i \in \{0, 1\}$. Let S = set of i s.t. $p_i = 0$. Then S is a cut. $q_{ij} \neq 0$ only if $i \in S$, $j \notin S$, in which case $q_{ij} = 1$. So, optimal cost $\sum_{(i,j)} q_{ij} u_{ij}$ = cut capacity.

So max flow value = optimal cost of min problem = capacity of some cut. Value of max flow $\leq$ capacity of every cut, so max flow = min cut capacity.

HW #8: Integer Programming

Problem #1: Need to schedule n jobs on m machines that can work in parallel. Job j requires processing time p_{ij} on machine i. Assume jobs start processing at time 0 and let C_j be the completion time of job j. Formulate as an IP problem finding a schedule that minimizes $\sum_j C_j$.

Decision vars: $x_{ikj} = 1$ if job j scheduled as kth job on machine i.

If the ith to kth jobs on machine i are $j_1, j_2, \ldots, j_k$ in that order, we denote completion times $C_{j_1}, \ldots$ and we have

$C_{j_1} = C_{j_0} + p_{ij_1}$, $C_{j_2} = C_{j_1} + p_{ij_2} + \ldots$, $C_{j_k} = C_{j_{k-1}} + p_{ij_k} + \ldots - p_{ij}$

So $\sum_{h=1}^k C_{j_h} = k p_{ij_1} + (k-1) p_{ij_2} + \ldots + p_{ij_k}$, so kth to last job contributes k times its processing time to the sum.

We formulate the problem as: min $\sum_{i=1}^m \sum_{j=1}^n \sum_{k=1}^n k p_{ij} x_{ikj}$

s.t. $\sum_{i=1}^m \sum_{k=1}^n x_{ikj} = 1$, $j = 1, \ldots, n$

$\sum_{j=1}^n x_{ikj} \leq 1$, $i = 1, \ldots, m$, $k = 1, \ldots, n$

$x_{ikj} \in \{0, 1\}$

Problem #2: Consider the IP problem maximize $c'x$ s.t. $Ax \leq b$, $x \geq 0$, x integer, where A, b, c comprised of positive integers and let x^* be its optimal solution. Consider LP relaxation w/ $\hat{x}$ as the optimal solution.

a) Show $\lfloor \hat{x} \rfloor$ a feasible solution of the IP. Since $\hat{x}$ feasible solution of LP relaxation, $\hat{x} \geq 0$, $A\hat{x} \leq b$. A has positive elements & $\lfloor \hat{x} \rfloor \leq \hat{x}$, $A\lfloor \hat{x} \rfloor \leq A\hat{x} \leq b$. $\lfloor \hat{x} \rfloor \geq 0$ & integer, so feasible for IP.

b) Show cost of $\lfloor \hat{x} \rfloor$ can be no further from the optimal IP cost than $\sum_{i=1}^n c_i$.

Observe $c'\lfloor \hat{x} \rfloor \leq c'x^* \leq c'\hat{x}$ ← opt solution of LP relaxation is upper bound on optimal cost of problem. ↳ feasibility of $\lfloor \hat{x} \rfloor$ + opt of x^*.

So, suboptimality gap of $\lfloor \hat{x} \rfloor$ upper bounded by $c'(\hat{x} - \lfloor \hat{x} \rfloor) \leq \sum_{i=1}^n c_i$

HW #5 Large Scale Optimization

Problem #1: There are n foods & m nutrients. We are given an $m \times n$ matrix A w/ a_{ij} denoting the amount of nutrient i per unit of jth food. Let b_1 and b_2 be the nutritional requirements of two children. Let c be the cost vector for all the foods. Assume $a_{ij} \geq 0$, $c_j \geq 0$ $\forall i,j$. We must satisfy each child's needs at min cost, w/ cost spent on each child the same.

a) Provide a standard form formulation

minimize $c'x_1 + c'x_2$ s.t. $Ax_1 - z_1 = b_1$, $Ax_2 - z_2 = b_2$, $c'x_1 = c'x_2$, $x_1, x_2, z_1, z_2 \geq 0$

We have $2m + 1$ constraints, $2n + 2m$ variables

b) Construct Dantzig-Wolfe subproblems solved during iteration of master problem. Coupling constraint is $c'x_1 - c'x_2 = 0$. Let q be the dual variable associated w/ it. $D_i = c$, so $c' - (p_1)$... so first subproblem is minimize $(1-q)c'x_1$ s.t. $Ax_1 - z_1 = b_1$, $x_1, z_1 \geq 0$; minimize $(1+q)c'x_2$ s.t. $Ax_2 - z_2 = b_2$, $x_2, z_2 \geq 0$

c) Suggest a direct approach for solving this problem based on solution of two single child diet problems. Solve diet problem for each child. WLOG suppose optimal cost for child 1 $g_1 \leq$ the optimal cost g_2 for child 2. Buy additional food items for child 1 to bring cost from g_1 to g_2. It is guaranteed that buying more food will increase cost ($c_j \geq 0$) and will not lead to a violation of nutrition requirements ($a_{ij} \geq 0$). This solution is optimal since no solution could have cost lower than g_2.

Problem #2: Consider a LP problem of the form minimize $c_1'x_1 + c_2'x_2 + c_0'y$

s.t. $\begin{bmatrix} D_1 & 0 & F_1 \\ 0 & D_2 & F_2 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ y \end{bmatrix} \geq \begin{bmatrix} b_1 \\ b_2 \end{bmatrix}$, $x_1, x_2 \geq 0$

a) Form the dual problem and explain how Dantzig-Wolfe decomposition can be applied to it. What is the structure of the subproblems solved during a typical iteration? The dual problem is maximize $p_1'b_1 + p_2'b_2$

s.t. $p_1'D_1 \leq c_1'$

$p_2'D_2 \leq c_2$

$p_1'F_1 + p_2'F_2 = c_0$ ← coupling constraints

$p_1, p_2 \geq 0$

Subproblems involve optimization s.t. $p_i'D_i \leq c_i'$, $p_i \geq 0$

b) Rewrite the first set of constraints in the form $D_1x_1 + F_1y_1 \geq b_1$ and $D_2x_2 + F_2y_2 \geq b_2$, together w/ constraints relating y_1 to y_2. Discuss how to apply DW decomposition. Rewrite as:

minimize $c_1'x_1 + c_2'x_2 + 0.5c_0'y_1 + 0.5c_0'y_2$ s.t. $D_1x_1 + F_1y_1 \geq b_1$, $D_2x_2 + F_2y_2 \geq b_2$, $y_1 - y_2 = 0$, $x_1, x_2 \geq 0$

Take $y_1 - y_2 = 0$ as coupling constraint. ith subproblem involves optimizing w/ respect to x_i, y_i s.t. constraints $D_ix_i + F_iy_i \geq b_i$, $x_i \geq 0$

Problem #3: Consider an LPP of the form $\boxed{\text{minimize } c'x + d'y \text{ s.t. } Ax + Dy \leq b,\ Fx \leq f,\ y \geq 0}$

a) Suppose we have a very fast subroutine for solving problems of the form min $h'x$ s.t. $Fx \leq f$ for arb. cost vectors h. How would you decompose?

Applying DW decomp, we obtain subproblems of the form min $h'x$ s.t. $Fx \leq f$, where cost h may change from one iteration to another. We also get subproblems involving only the y variables, but since y vectors only constrained by $y \geq 0$, subproblems are easily solved.

b) Suppose we have fast subroutine for solving $\boxed{\min d'y \text{ s.t. } Dy \leq k,\ y \geq 0}$ for arbitrary vectors k. How to decompose?

Form the dual: $\boxed{\text{maximize } p'b + q'f \text{ s.t. } p'A + q'F = c',\ p'D \leq d',\ p \leq 0,\ q \leq 0}$

Apply DW decomp on dual. 1st constraint is coupling constraint. 2nd subprob involves optimization subject to the constraint $q \leq 0$ and is easily solved. The 1st subproblem involves optimization s.t. constraints $p'D \leq d'$, $p \leq 0$. Its generic form: $\boxed{\max p'h \text{ s.t. } p'D \leq d,\ p \leq 0}$ its dual $\boxed{\min d'y \quad Dy \leq h,\ y \geq 0}$

A generic fast subroutine can be solved by forming its dual and using the available subroutine.

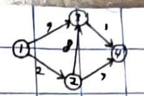
Problem #4: Consider a problem of the form $\min \max_{i=1,\ldots,m} (a_i'x - b_i)$ subject to no constraints.

a) Describe a cutting plane method for problems of this form. Given a subset I of the set of indices i, minimize $\max_{i \in I}(a_i'x - b_i)$, obtain optimal solution x^*, and let $z = \max_{i \in I}(a_i'x^* - b_i)$. If $z \geq a_i'x^* - b_i\ \forall i$, then x^* optimal solution of original problem. Else, we find some i s.t. $z < a_i'x^* - b_i$, add i to I, and continue.

b) Let x^* be an optimal solution to a relaxed problem in which only some of the $a_i'x - b_i$ terms have been retained. Describe a simple method for obtaining upper/lower bounds on the optimal cost for the original problem.

Let x^* be the optimal cost of the relaxed problem. $\max(a_i'x^* - b_i)$ is an upper bound on the optimal cost. Optimal cost of relaxed problem is a lower bound on the optimal cost.

BELLMAN FORD ALGORITHM EXAMPLE



Destination node = 4

$p_4(k) = 0\ \forall k$

$P_1(0) = 0$	$P_1(1) = \infty$	$P_1(2) = 9$ $P_1(3) = 9$
$P_2(0) = 0$	$P_2(1) = 7$	$P_2(2) = 7$ $P_2(3) = 7$
$P_3(0) = 0$	$P_3(1) = 1$	$P_3(2) = 1$ $P_3(3) = 1$

KKT Practice Problem

Among all possible rectangles w/ a given sum of lengths of edges, find the one that maximizes the surface area.

max $2(x_1x_2 + x_1x_3 + x_2x_3)$ s.t. $x_1, x_2, x_3 \geq 0$, $x_1 + x_2 + x_3 = \alpha$

min $-(x_1x_2 + x_1x_3 + x_2x_3)$ s.t. $x_1 + x_2 + x_3 = \alpha$, $x_1, x_2, x_3 \geq 0$

KKT:

min $f(x)$ s.t. $g_j(x) \leq 0$, $h_i(x) = 0$

$\nabla_x L = 0$, $\nabla_\lambda L = 0$, $\nabla_\mu L = 0$, $\mu_j g_j = 0\ \forall j$

$y'\nabla_{xx} L\, y \geq 0\ \forall y \in V(x) = \{y \mid \nabla h(x^*)'y = 0\}$

$L(x_1, x_2, x_3, \lambda, \mu) = -x_1x_2 - x_2x_3 - x_1x_3 + \lambda(x_1 + x_2 + x_3 - \alpha) + \mu_1x_1 + \mu_2x_2 + \mu_3x_3$

$\nabla_x L = \begin{bmatrix} -x_2 - x_3 + \lambda \\ -x_1 - x_3 + \lambda \\ -x_1 - x_2 + \lambda \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} \rightsquigarrow x_1 = x_2 = x_3$

$\nabla_\lambda L = x_1 + x_2 + x_3 - \alpha = 0$

$3x_1 = \alpha \Rightarrow x_1 = x_2 = x_3 = \frac{\alpha}{3}$

$\nabla_{xx} L = \begin{bmatrix} 0 & -1 & -1 \\ -1 & 0 & -1 \\ -1 & -1 & 0 \end{bmatrix}$, $\nabla_x h(x) = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix}$

$V(x^*) = \{y \mid y_1 + y_2 + y_3 = 0\}$

$[y_1\ y_2\ y_3] \begin{bmatrix} 0 & -1 & -1 \\ -1 & 0 & -1 \\ -1 & -1 & 0 \end{bmatrix} \begin{bmatrix} y_1 \\ y_2 \\ y_3 \end{bmatrix} = y_1^2 + y_2^2 + y_3^2 \geq 0$ ✓

$y_1 + y_2 + y_3 = 0 \Rightarrow y_1 = y_2 = y_3$

Basis $\mu_j \neq 0$, nonbasic $\Rightarrow$ active constraints