$\boxed{\text{Linear Programming \& the Simplex Algorithm}}$

Part I

P&S Chapter 2

Last Revised – Oct 6, 2004

Many Optimization problems can be written in the form

$$
\begin{array}{rclcccl}
\multicolumn{3}{l}{\min\ c'x} & & & & \text{Cost/Objective} \\
a_i'x & = & b_i & i & \in & M & \text{Constraint} \\
a_i'x & \geq & b_i & i & \in & \bar{M} & \text{Constraint} \\
x_j & \geq & 0 & j & \in & N & \text{Constraint} \\
x_j & \lessgtr & 0 & j & \in & \bar{N} & \text{Constraint}
\end{array}
$$

where
$x \in R^n$ and b are $n \times 1$ column matrices (vectors)
and the a_i' are m $1 \times n$ row matrices (vectors).
Often assume that the b, a_i are integers (or rational).

An $x \in R^n$ that satisfies the constraints is a feasible solution.

The problem is to find the minimal cost of a feasible solution.

Such a problem is known as a **Linear Program**.

One of the first problems ever formulated in this fashion was the **diet problem**; how to minimize the cost of a diet that contains a sufficient amount of each of m nutrients that can be constructed from an available supply of n different foods.

$$
\begin{aligned}
a_{i,j} &= \text{amount of } i\text{th nutrient in a unit of the } j\text{th food} \\
&\quad\; i = 1, \ldots, m,\; j = 1, \ldots, n \\
r_i &= \text{yearly requirement of } i\text{th nutrient} \\
&\quad\; i = 1, \ldots, m \\
x_j &= \text{yearly consumption of the } j\text{th food (in units)} \\
&\quad\; j = 1, \ldots, n \\
c_j &= \text{cost per unit of the } j\text{th food} \\
&\quad\; j = 1, \ldots, n
\end{aligned}
$$

Consider the $a_{i,j}$ as entries in matrix A and r_i, x_j as entries in column vectors r, x and c_j as entries in a row vector. Then the problem can be rewritten as

$$
\begin{aligned}
&\min\; c'x \\
Ax &\geq r \\
x &\geq 0
\end{aligned}
$$

There are actually three *different* accepted ways of writing LPs:

- General Form (Ex. Max-Flow)

$$
\begin{array}{rclccc}
\multicolumn{3}{c}{\min\ c'x} & & & \\
a_i'x & = & b_i & i & \in & M \\
a_i'x & \geq & b_i & i & \in & \bar{M} \\
x_j & \geq & 0 & j & \in & N \\
x_j & \lesseqgtr & 0 & j & \in & \bar{N}
\end{array}
$$

- Canonical Form (Ex. Diet-Problem)

$$
\begin{array}{rcl}
\multicolumn{3}{c}{\min\ c'x} \\
Ax & \geq & r \\
x & \geq & 0
\end{array}
$$

- Standard Form

$$
\begin{array}{rcl}
\multicolumn{3}{c}{\min\ c'x} \\
Ax & = & r \\
x & \geq & 0
\end{array}
$$

General Form	Canonical Form	Standard Form
$$\begin{array}{rcllcl} & \min c'x & & & & \\ a_i'x & = & b_i, & i & \in & M \\ a_i'x & \geq & b_i, & i & \in & \bar{M} \\ x_j & \geq & 0, & j & \in & N \\ x_j & \lesseqgtr & 0, & j & \in & \bar{N} \end{array}$$	$$\begin{array}{rcl} & \min c'x & \\ Ax & \geq & r \\ x & \geq & 0 \end{array}$$	$$\begin{array}{rcl} & \min c'x & \\ Ax & = & r \\ x & \geq & 0 \end{array}$$

Theorem: The General, Canonical and Standard Forms are all equivalent to each other.

Different forms are useful for different problem formulations and proving different theorems.

Proof: Problems in SF and CF are already in GF.

We must therefore show
(a) GF $\Rightarrow$ CF:
(b) GF $\Rightarrow$ SF:

General Form | Canonical Form | Standard Form

$$\begin{array}{rcllcl} & \text{min } c'x & & & & \\ a_i'x & = & b_i, & i & \in & M \\ a_i'x & \geq & b_i, & i & \in & \bar{M} \\ x_j & \geq & 0, & j & \in & N \\ x_j & \lesseqgtr & 0, & j & \in & \bar{N} \end{array}$$

$$\begin{array}{rcl} & \text{min } c'x & \\ Ax & \geq & r \\ x & \geq & 0 \end{array}$$

$$\begin{array}{rcl} & \text{min } c'x & \\ Ax & = & r \\ x & \geq & 0 \end{array}$$

GF $\Rightarrow$ CF:

We must eliminate unconstrained variables and equalities.

Variables:

Replace x_j s.t. $x_j \lesseqgtr 0$ with two *new* variables x_j^+, x_j^- s.t. $x_j^+, x_j^- \geq 0$ and $x_j = x_j^+ - x_j^-$.

More specifically, remove x_j and constraint $x_j \lesseqgtr 0$, add two variables x_j^+, x_j^- and conditions $x_j^+ \geq 0$, $x_j^- \geq 0$. Every occurrence $a_{i,j}x_j$ in an inequality is replaced by $a_{i,j}x_j^+ - a_{i,j}x_j^-$.

Equalities: Any equality $\sum_{j=1}^n a_{i,j}x_j = b_i$ can be replaced by two inequalities $\sum_{j=1}^n a_{i,j}x_j \geq b_i$ and $\sum_{j=1}^n (-a_{i,j})x_j \geq -b_i$.

General Form	Canonical Form	Standard Form
$\begin{array}{rcllcl} & \min c'x & & & & \\ a_i'x & = & b_i, & i & \in & M \\ a_i'x & \geq & b_i, & i & \in & \bar{M} \\ x_j & \geq & 0, & j & \in & N \\ x_j & \lesseqgtr & 0, & j & \in & \bar{N} \end{array}$	$\begin{array}{rcl} & \min c'x & \\ Ax & \geq & r \\ x & \geq & 0 \end{array}$	$\begin{array}{rcl} & \min c'x & \\ Ax & = & r \\ x & \geq & 0 \end{array}$

GF $\Rightarrow$ SF:

We must eliminate unconstrained variables and inequalities.

Unconstrained variables are eliminated as before, i.e., x_j replaced by x_j^+, x_j^-.

Inequalities:

Given inequality $\sum_{j=1}^n a_{i,j}x_j \geq b_i$ in GF introduce new surplus variable s_i and equation

$$\boxed{\sum_{j=1}^n a_{i,j}x_j - s_i = b_i, \quad s_i \geq 0.}$$

Note: If we ever have inequality of form $\sum_{j=1}^n a_{i,j}x_j \leq b_i$ we can introduce new slack variable s_i and equation

$$\boxed{\sum_{j=1}^n a_{i,j}x_j + s_i = b_i, \quad s_i \geq 0.}$$

Basic Feasible Solutions

$$\begin{aligned} &\min c'x \\ Ax &= r \\ x &\geq 0 \end{aligned}$$

Assume we are given an LP in standard form , where A is an $m \times n$ matrix with $m \leq n$. We will also make an assumption (shown unnecessary later)

Assumption 1: A has *rank* m, i.e., there are m linearly independent A_j columns in A.

A Basis of A is a linearly independent collection $\mathcal{B} = \{A_{j_1}, \ldots, A_{j_m}\}$. Alternatively, $\mathcal{B}$ can be thought of as an $m \times m$ nonsingular matrix $B = [A_{j_i}]$.

The Basic Solution corresponding to $\mathcal{B}$ is $x \in R^n$ s.t.

$$\begin{aligned} x_j &= 0 \text{ for } A_j \notin \mathcal{B} \\ x_{j_k} &= \text{the } k\text{th component of } B^{-1}b, \quad k = 1, \ldots, m. \end{aligned}$$

$x \in R^n$ is a basic feasible solution if it is a basic solution *and* a feasible solution.

$$
\begin{array}{lrcrcrcrcrcrcrcl}
\min & & & 2x_2 & + & x_4 & + & 5x_7 & & & & & & & & \\
& x_1 & + & x_2 & + & x_3 & + & x_4 & & & & & & & = & 4 \\
& x_1 & & & & & & & + & x_5 & & & & & = & 2 \\
& & & & & x_3 & & & & & + & x_6 & & & = & 3 \\
& & & 3x_2 & + & x_3 & & & & & & & + & x_7 & = & 6 \\
& x_1, & & x_2, & & x_3, & & x_4, & & x_5, & & x_6, & & x_7 & \geq & 0
\end{array}
$$

Example 1:

$\mathcal{B} = \{A_4, A_5, A_6, A_7\}$.

$B = I$.

Corresponding basic solution is

$x = (0, 0, 0, 4, 2, 3, 6)$.

This x is feasible.

Example 2:

$\mathcal{B}' = \{A_2, A_5, A_6, A_7\}$.

Corresponding basic solution is

$x' = (0, 4, 0, 0, 2, 3, -6)$.

This x' is *not* feasible.

Basic Feasible Solutions will be interesting to us since we will later be able to show that **there exists an optimal solution which is a basic feasible solution.**

Since a basic solution corresponds to a set of m linearly independent columns this transforms our continuous optimization problem (with an infinite number of solutions) into a combinatorial optimization problem (with a finite number of solutions).

A finite algorithm would then be to simply look at all $\binom{n}{m}$ subsets of m columns, calculate the corresponding basic solutions and check if they are feasible. Examine all basic feasible solutions in this fashion and return the one with minimum cost.

We will also see that a linear program corresponds to a polytope with basic feasible solutions corresponding to vertices.

The simplex method will permit us to solve the LP by starting at some vertex (BFS) and then walking from vertex to vertex on edges of the polytope, always improving the cost of the current solution. When the solution can no longer be improved by walking along some edge leaving the current vertex, we will be at an optimal solution.

The next few classes will be devoted to deriving all of the above.

Lemma: Let $x = (x_1, \ldots, x_n)$ be a basic solution. Then

$$|x_j| \leq m! \alpha^{m-1} \beta$$

where

$$\alpha = \max_{i,j} \{|a_{ij}|\} \quad \text{and} \quad \beta = \max_{j=1,\ldots,m} \{|b_j|\}$$

Proof: If x_j is not basic then $x_j = 0$.

If x_j is basic then it is the appropriate entry in $B^{-1}b$.

Each element in B^{-1} is an entry in a $(m-1) \times (m-1)$ determinant divided by $det(B)$. The entries in B^{-1} are all $\leq (m-1)! \alpha^{m-1}$ and $|det(B)| \geq 1$.

Since x_j is the sum of m entries in B^{-1} multiplied by entries in b, the lemma follows.

Lemma: Let x be a BFS of

$$\begin{aligned} Ax &= r \\ x &\geq 0 \end{aligned}$$

corresponding to basis $\mathcal{B}$. Then there exists cost vector c such that x is the unique optimal solution of the LP

$$\begin{aligned} &\min\ c'x \\ Ax &= r \\ x &\geq 0 \end{aligned}$$

Recall

Assumption 1: A has *rank* m, i.e., there are m linearly independent A_j columns in A.

We now add (again we will remove this later)

Assumption 2: The set F of feasible points is not empty.

Theorem: Under assumptions 1 and 2 at least one BFS exists.

Proof: Assume, in contradiction, that F contains a solution x with $t > m$ non-zero components and that x is solution with *largest* number of zero components. WLOG

$$x_1, \ldots, x_t > 0, \quad \text{and} \quad x_{t+1}, \ldots x_n = 0.$$

The first t columns of A then satisfy

$$A_1 x_1 + \cdots + A_t x_t = b.$$

Theorem: Under assumptions 1 and 2 at least one BFS exists.

Proof: (cont) $A_1x_1 + \cdots + A_tx_t = b.$

Let r be the rank of these t columns.
$r > 0$ since if $r = 0$, then BFS $x = 0$ in F.

Therefore(after possible row reordering) the matrix

$$\begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1r} \\ a_{21} & a_{22} & \cdots & a_{2r} \\ \vdots & \vdots & \ddots & \vdots \\ a_{r1} & a_{r2} & \cdots & a_{rr} \end{bmatrix}$$

is nonsingular.This permits solving $x_1, \ldots, x_r$ in terms of $x_{r+1}, \ldots x_t$. That is

$$x_j = \beta_j + \sum_{i=r+1}^{t} \alpha_{ij}x_i, \quad j = 1, \ldots, r$$

Proof: (cont)
$A_1x_1 + \cdots + A_tx_t = b$, and
$x_j = \beta_j + \sum_{i=r+1}^{t} \alpha_{ij}x_i, \quad j = 1, \ldots, r$

Now let

$$\theta = \min\{x_{r+1}, \theta_1\}$$

where

$$\theta_1 = \min_{\alpha_{r+1,i}>0} \left\{ \frac{x_i}{\alpha_{r+1,i}}, i = 1, \ldots, r \right\}$$

Construct new feasible solution

$$\hat{x}_j = \begin{cases} x_j - \theta & \text{if } j = r+1 \\ x_j & \text{if } j > r+1 \\ \beta_j + \sum_{i=r+1}^{t} \alpha_{ij}\hat{x}_i & \text{if } j < r+1 \end{cases}$$

Then, for $j \le r$, $\hat{x}_j = x_j - \alpha_{r+1,j}\theta$.

If $\theta = x_{r+1}$ then $\hat{x}_{r+1} = 0$;
If $\theta = \theta_1 = \frac{x_k}{\alpha_{r+1,k}}$ for some $k \le r$, then $\hat{x}_k = 0$.

So $\hat{x}$ is a *feasible* solution with one more zero component than x, contradiction.

So far we have made 2 assumptions. we now add a third (which we will also be able to get rid of later).

Assumption 1:
A has *rank* m, i.e., there are m linearly independent A_j columns in A.

Assumption 2:
The set F of feasible points is not empty.

Assumption 3:
The set of real numbers $\{c'x \ : \ x \in F\}$ is bounded from below.

We can now prove

Theorem:
Let Assumptions 1-3 hold for the LP

$$\begin{aligned} \min c'x \\ Ax = b \\ x \geq 0 \end{aligned}$$

Then the following LP is equivalent, in the sense that it has the same optimal value of its cost function:

$$\begin{aligned} \min c'x \\ Ax = b \\ x \geq 0 \\ x \leq M \end{aligned}$$

where

$$\begin{aligned} M &= (m+1)!\alpha^m \beta \\ \alpha &= \max\{|a_{ij}|, |c_j|\} \\ \beta &= \max\{|b_i|, |z|\} \end{aligned}$$

and z is the greatest lower bound of the set $\{c'x : Ax = b, x \geq 0\}$.

The Geometry of Linear Programs

We will now see how LPs correspond to polytopes and BFSs to vertices of the polytopes. This requires reviewing some basic linear algebra.

- Consider vector space R^d. A (linear) subspace S of R^d is a subset of R^d closed under vector addition and scalar multiplication.

- Equivalently, S is the set of points satisfying linear equations
  $S = \{x \in R^d \,:\, a_{j,1}x_1 + \cdots + a_{j,d}x_d = 0,\ j = 1, \ldots, m\}$

- $dim(S)$, the dimension of S, is the maximum number of linearly independent vectors in S.
  $dim(S) = d - rank([a_{ij}])$

- An affine subspace A is a linear subspace S translated by $u \in S$;
  $A = \{u + s \,:\, s \in S\}$.

- $dim(A) = dim(S)$

- Equivalently, A is set of points satisfying
  $A = \{x \in R^d \,:\, a_{j,1}x_1 + \cdots + a_{j,d}x_d = b_j,\ j = 1, \ldots, m\}$

- The dimension of *any subset* of R^d is smallest dimension of any affine subspace containing set, e.g., line segments have dimension 1.
  Any set of $k \le d + 1$ points has dimension $\le k - 1$.

- If A is an $m \times n$ matrix then dimension of feasible set defined by LP

$$Ax = b \quad \text{and} \quad x \ge 0$$

  is at most $d - m$.

An affine subspace of dimension $d-1$ is called a *hyperplane*. Equivalently, a hyperplane is a set of points satisfying

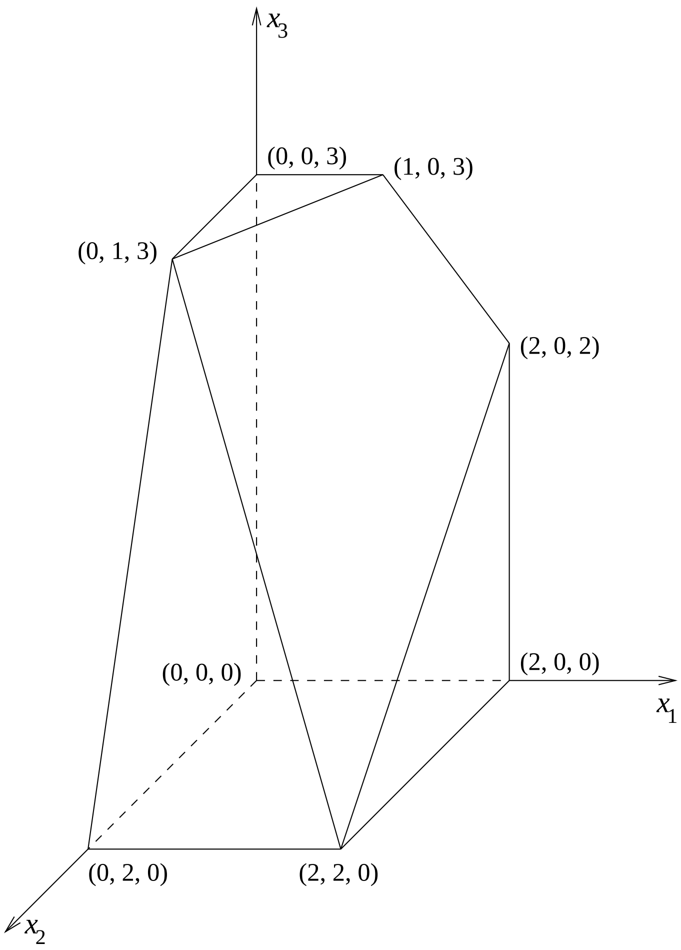
$$a_1x_1 + a_2x_2 + \cdots + a_dx_d = b$$

where not all $a_i = 0$. A hyperplane defines two *(closed) halfspaces*:

$$\begin{aligned} a_1x_1 + a_2x_2 + \cdots + a_dx_d &\geq b \\ a_1x_1 + a_2x_2 + \cdots + a_dx_d &\leq b \end{aligned}$$

A halfspace is *convex*. Since the intersection of convex sets is also convex, the intersection of halfspaces is also convex. The intersection of a *finite* number of halfspaces is called a *(convex) polytope*.

We will only be interested in polytopes in which all vector entries are nonnegative. By convention, d of the defining halfspaces will always be
$x_j \geq 0, \quad j = 1, \ldots, d$.



$$
\begin{array}{rcrcrcl}
x_1 & + & x_2 & + & x_3 & \leq & 4 \\
x_1 & & & & & \leq & 2 \\
 & & & & x_3 & \leq & 3 \\
 & & 3x_2 & + & x_3 & \leq & 6 \\
x_1 & & & & & \geq & 0 \\
 & & x_2 & & & \geq & 0 \\
 & & & & x_3 & \geq & 0
\end{array}
$$

let P be a polytope of dimension d and HS a half-space defined by hyperplane H. If the intersection $f = P \cap HS \subset H$ then
f is a *face* of P.
H is the *supporting hyperplane defining f.*

In particular.
A *facet* is a face of dimension $d - 1$.
A *vertex* is a face of dimension zero (a point).
An *edge* is a face of one dimension (a line segment).

Some observations:

The hyperplane defining a facet corresponds to a defining halfspace of the polytope. The converse is not true. Some defining halfspaces are *redundant* and can be discarded without changing the polytope.

An edge always connects two vertices.
Not every two vertices are connected by an edge.

Theorem:
(a) Every convex polytope is the convex hull of its vertices.
(b) If V is a finite set of points then $CH(V)$ is a convex polytope P. The set of vertices of P is a subset of V.

A polytope P can therefore be thought of in 3 ways:

1) As the convex hull of a finite set of polytopes.

2) As the intersection of many halfspaces, as long as the intersection is bounded.

3) An algebraic version of the above: let

$$Ax = b, \quad x \geq 0$$

define the feasible region F of some LP.
Assume A is an $m \times n$ matrix of rank m.
We may then assume (why?)

$$x_i = b_i - \sum_{j=1}^{n-m} a_{i,j}\, x_j, \quad i = n-m+1, \ldots, n.$$

This is the same as

$$\begin{aligned} b_i - \textstyle\sum_{j=1}^{n-m} a_{i,j}\, x_j &\geq 0, & i &= n-m+1, \ldots, n \\ x_j &\geq 0, & j &= 1, \ldots, n-m \end{aligned}$$

so F is a polytope in R^{n-m} defined by the intersection of n halfspaces.

3) (cont). We just saw that if A is a $m \times n$ matrix then feasible region of

$$Ax = b, \quad x \geq 0$$

is a polytope in R^{n-m} defined by the intersection of n halfspaces.

Now assume the converse, that P is a convex polytope in R^{n-m} defined by the intersection of n halfspaces.

$$h_{i,1}x_1 + \cdots + h_{i,n-m}x_{n-m} + g_i \leq 0, \quad i = 1, \ldots, n.$$

By our convention we may assume that first $n-m$ equations are:

$$x_i \geq 0, i = 1, \ldots, n-m$$

Introducing $n-m$ *slack variables* $x_{n-m+1}, \ldots, x_n$ gives

$$Ax = b, \quad x \geq 0$$

where $b_i = -g_i$, $m \times n$ matrix $A = [H|I]$ and $x \in R^n$.

So a polytope can be viewed as the feasible region of some LP.

Theorem:
let P be a convex polytope,
$F = \{x \,:\, Ax = b,\, x \geq 0\}$,
the corresponding feasible set of an LP and
$\widehat{x} = (x_1, \ldots, x_{n-m}) \in P$.
Then the following three statements are equivalent:

a) The point $\widehat{x}$ is a vertex of P.

b) If $\widehat{x} = \lambda\widehat{x}' + (1-\lambda)\widehat{x}''$ with $\widehat{x}', \widehat{x}'' \in P$, $0 < \lambda < 1$, then $\widehat{x} = \widehat{x}' = \widehat{x}''$

c) The corresponding vector $x \in F$ is a BFS of F.

Note: Given two different vertices u, u', their corresponding bases $\mathcal{B}, \mathcal{B}'$ must be different. But, two different bases $\mathcal{B}, \mathcal{B}'$ could correspond to the same vertex. If this happens, the corresponding BFS has more than $m-n$ zeros. Such a BFS is called degenerate.

Finiteness of Linear programming

Theorem:
(a) There is an optimal BFS in any instance of LP.

(b) Furthermore, if q BFSs are optimal, their convex combinations are also optimal.

Proof:
This is equivalent to proving there is an optimal vertex of P and that if q vertices are optimal, so is any convex combination of them.

Let d be the objective function, i.e., cost is $d'x$. Note that P is *closed* so $d'x$ attains its minimum in P.

To prove (a) let x_0 be an optimal point and $x_1, \ldots x_N$ the vertices of P. Then

$$x_0 = \sum_{i=1}^{N} \alpha_i x_i, \quad \text{where} \sum_{i=1}^{N} \alpha_i = 1, \; \alpha_i \geq 0$$

Let j be the vertex with lowest cost. Then

$$d' x_0 = d' \sum_{i=1}^{N} \alpha_i x_i \geq d' x_j \sum_{i=1}^{N} \alpha_i = d' x_j$$

so x_j is optimal.

Theorem:

(a) There is an optimal BFS in any instance of LP.

(b) Furthermore, if q BFSs are optimal, their convex combinations are also optimal.

Proof: (cont)

To prove (b) assume that vertices $x_{j_1}, \ldots, x_{jq}$ are optimal and let $y = \sum_{i=1}^{q} \alpha_i x_{j_i}$ be some convex combination of the vertices, i.e., $\sum_{i=1}^{N} \alpha_i = 1, \ \alpha_i \geq 0$. Then

$$d'y = \sum_{i=1}^{q} \alpha_i x_{j_i} = \sum_{i=1}^{q} \alpha_i (d' x_{j_i}) = d' x_{j_1}$$

and we are finished.

As discussed earlier we have just shown that **there always exists an optimal solution which is a basic feasible solution.**

Since a basic solution corresponds to a set of m linearly independent columns this transforms our continuous optimization problem (with an infinite number of solutions) into a combinatorial optimization problem (with a finite number of solutions).

A finite algorithm would then be to simply look at all $\binom{n}{m}$ subsets of m columns, calculate the corresponding basic solutions and check if they are feasible. Examine all basic feasible solutions in this fashion and return the one with minimum cost.

In the next section we will develop the simplex algorithm, a more efficient way of walking through the basic feasible solutions (vertices), one that always improves cost (and usually doesn't have to look at all of the BFSs before finding the optimal).