Linear Programming

Linear programming became important during World War II:

→ used to solve logistics problems for the military.

Linear Programming (LP) was the first widely used form of optimization in the process industries and is still the dominant form today.

Linear Programming has been used to:

→ schedule production,

→ choose feedstocks,

→ determine new product feasibility,

→ handle constraints for process controllers,

⋮

There is still a considerable amount of research taking place in this area today.

Linear Programming

All Linear Programs can be written in the form:

$$\min_{x_i} \quad c_1 x_1 + c_2 x_2 + c_3 x_3 + \ \dots \ + \ c_n x_n$$

subject to:

$$\begin{aligned}
a_{11}x_1 + a_{12}x_2 + a_{13}x_3 + \ \dots \ + \ a_{1n}x_n \quad &\leq \ b_1 \\
a_{21}x_1 + a_{22}x_2 + a_{23}x_3 + \ \dots \ + \ a_{2n}x_n \quad &\leq \ b_2 \\
\vdots \qquad\qquad\qquad &\quad \vdots \\
a_{m1}x_1 + a_{m2}x_2 + a_{m3}x_3 + \ \dots \ + \ a_{mn}x_n \ &\leq \ b_m
\end{aligned}$$

$$0 \ \leq \ x_i \ \leq \ x_{l,i}$$

$$b_i \ \geq \ 0$$

⇒ note that all of the functions in this optimization problem are linear in the variables x_i.

Linear Programming

In mathematical short-hand this problem can be re-written:

$$\begin{aligned} &\min_{\mathbf{x}} \quad \mathbf{c}^{\mathrm{T}}\, \mathbf{x} \\ &\text{subject to}: \\ &\qquad \mathbf{A}\,\mathbf{x} \le \mathbf{b} \\ &\qquad 0 \le \mathbf{x} \le \mathbf{x}_l \\ &\qquad \mathbf{b} \ge \mathbf{0} \end{aligned}$$

Note that:

1) the objective function is linear,
2) the constraints are linear,

convention

3) the variables are defined as non-negative,
4) all elements of **b** are non-negative.

convention

Linear Programming

A Simple Example:

A market gardener has a small plot of land she wishes to use to plant cabbages and tomatoes. From past experience she knows that her garden will yield about 1.5 tons of cabbage per acre or 2 tons of tomatoes per acre. Her experience also tells her that cabbages require 20 lbs of fertilizer per acre, whereas tomatoes require 60 lbs of fertilizer per acre. She expects that her efforts will produce $600/ton of cabbages and $750/ton of tomatoes.

Unfortunately, the truck she uses to take her produce to market is getting old, so she doesn't think that it will transport anymore than 3 tons of produce to market at harvest time. She has also found that she only has 60 lbs of fertilizer.

What combination of cabbages and tomatoes should she plant to maximize her profits?

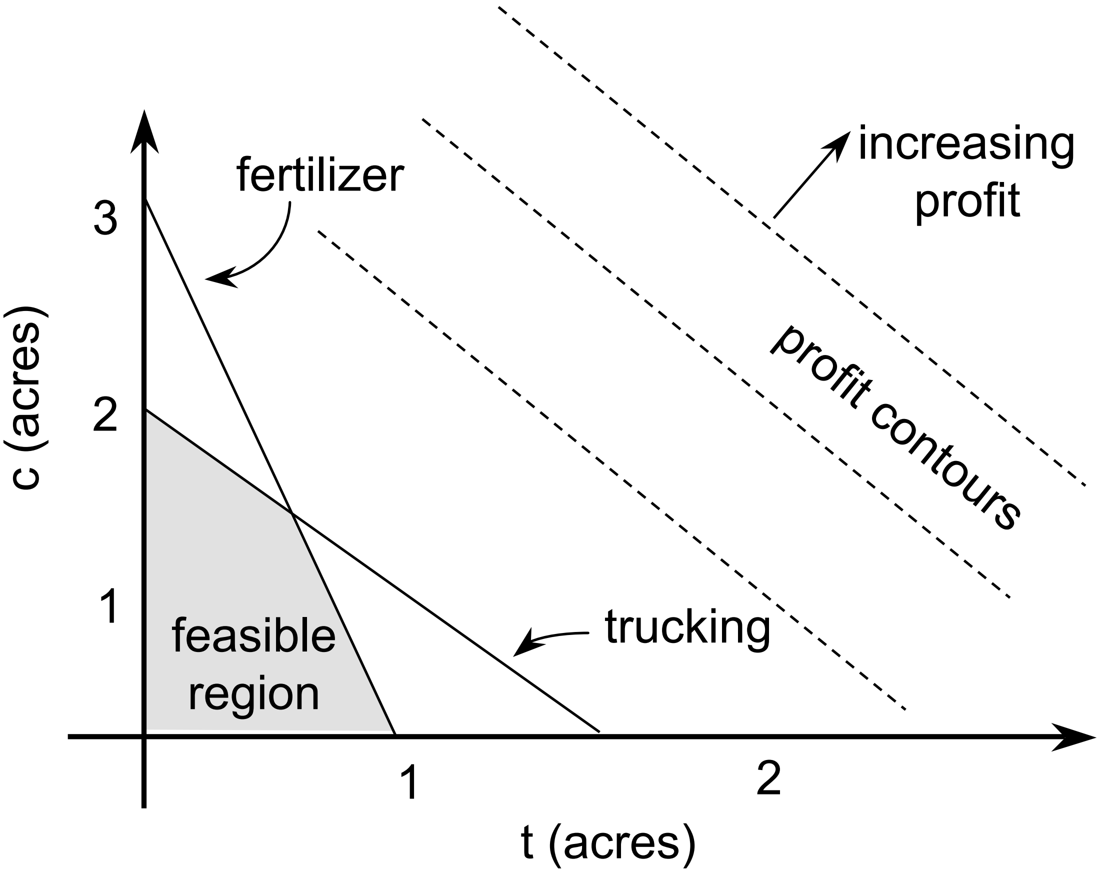
Linear Programming

The optimization problem is:

In the general matrix LP form:

Linear Programming

We can solve this problem graphically:



We know that for well-posed LP problems, the solution lies at a vertex of the feasible region. So all we have to do is try all of the constraint intersection points.

Linear Programming

Check each vertex:

c	t	P(**x**)
0	0	0
2	0	1800
0	1	1500
6/5	3/5	1980

So our market gardener should plant 1.2 acres in cabbages and 0.6 acres in tomatoes. She will then make $1980 in profit.

This graphical procedure is adequate when the optimization problem is simple. For optimization problems which include more variables and constraints this approach becomes impractical.

Linear Programming

Our example was well-posed, which means there was an unique optimum. It is possible to formulate LP problems which are ill-posed or degenerate. There are three main types of degeneracy which must be avoided in Linear Programming:

1) unbounded solutions,

- not enough independent constraints.

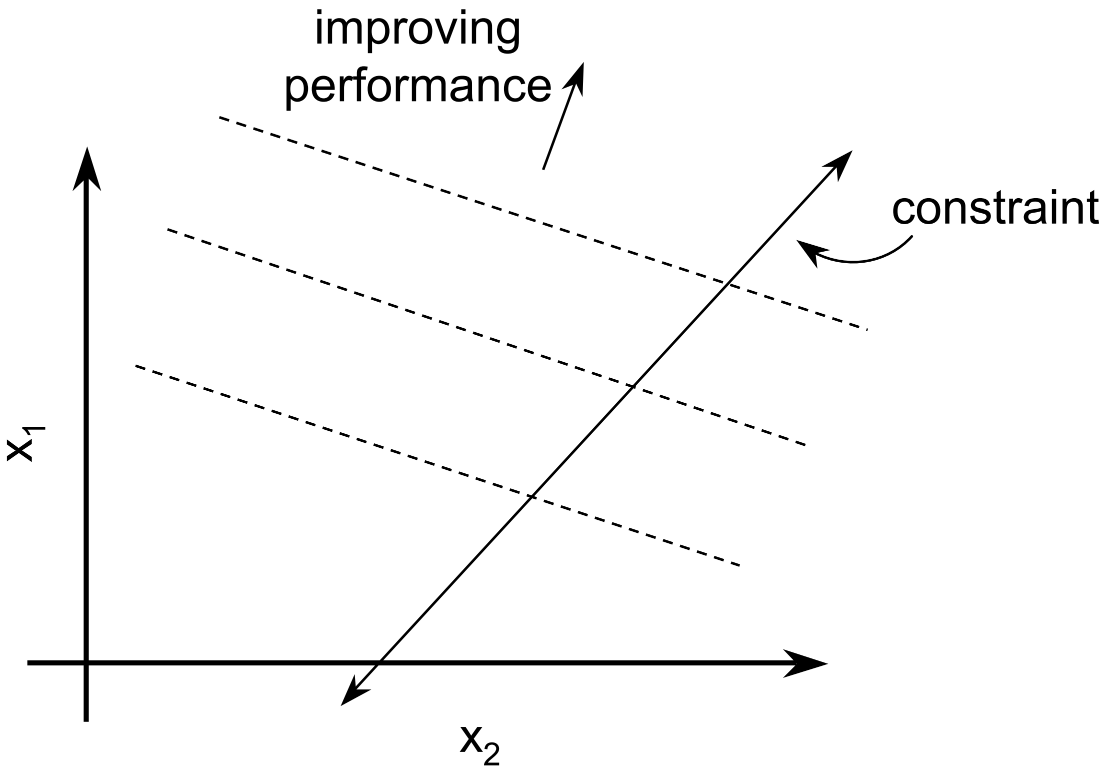
2) no feasible region,

- too many or conflicting constraints.

3) non-unique solution,

- too many solutions,
- dependence between profit function and a few of the constraints.

1) Unbounded Solution:



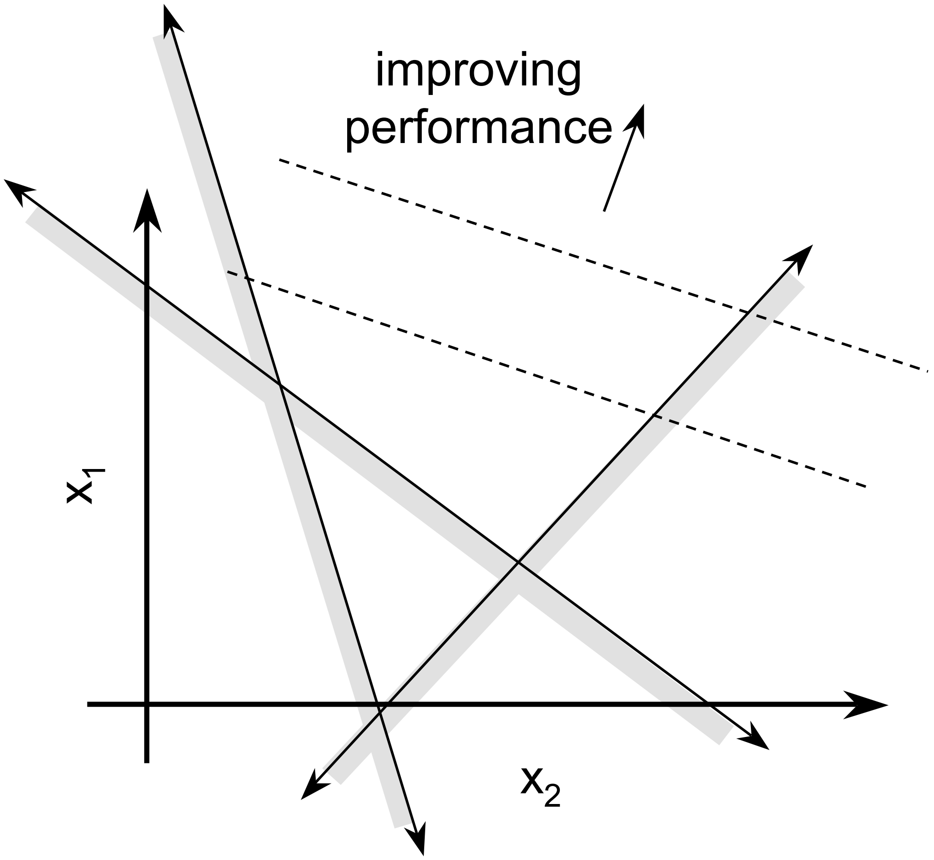
As we move up either the x_1-axis or the constraint the objective function improves without limit.

In industrial-scale problems with many variables and constraints, this situation can arise when:

→ there are too few constraints.

→ there is linear dependence among the constraints.

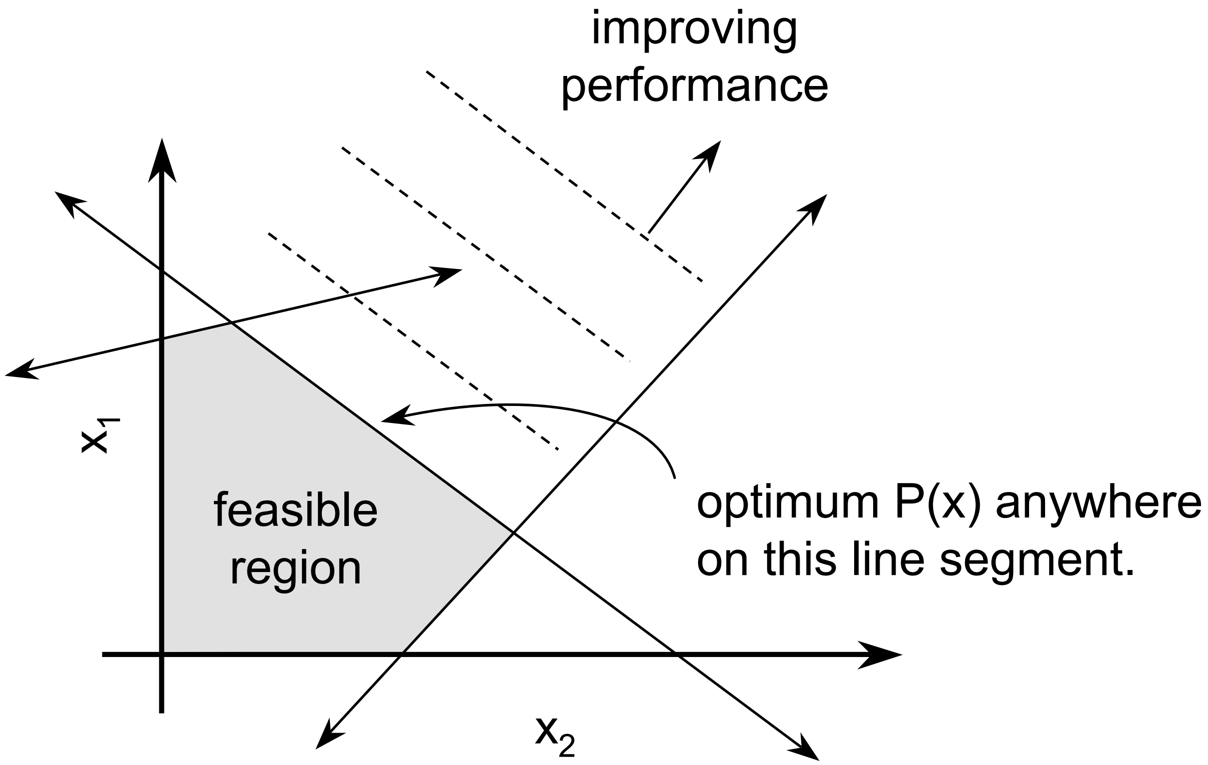
2) No Feasible Region:



Any value of **x** violates some set of constraints.

This situation occurs most often in industrial-scale LP problems when too many constraints are specified and some of them conflict.

3) Non-Unique Solution:



Profit contour is parallel to constraint or:

$$\nabla_x P \propto \nabla_x C_i$$

Linear Programming

Our market gardener example had the form:

$$\min_{\mathbf{x}} \quad -\begin{bmatrix} 900 & 1500 \end{bmatrix} \mathbf{x}$$

subject to :

$$\begin{bmatrix} 1.5 & 2 \\ 20 & 60 \end{bmatrix} \mathbf{x} \le \begin{bmatrix} 3 \\ 60 \end{bmatrix}$$

where : $\mathbf{x} \equiv \begin{bmatrix} \text{acres cabbages} & \text{acres tomatoes} \end{bmatrix}^T$.

We need a more systematic approach to solving these problems, particularly when there are many variables and constraints.

→ SIMPLEX method (Dantzig).

→ always move to a vertex which improves the value of the objective function.

Linear Programming

SIMPLEX Algorithm

1) Convert problem to standard form with:
   - positive right-hand sides,
   - lower bounds of zero.

2) Introduce slack / surplus variables:
   - change all inequality constraints to equality constraints.

3) Define an initial feasible basis:
   - choose a starting set of variable values which satisfy all of the constraints.

4) Determine a new basis which improves objective function:
   - select a new vertex with a better value of $P(\mathbf{x})$.

5) Transform the equations:
   - perform row reduction on the equation set.

6) Repeat Steps 4 & 5 until no more improvement in the objective function is possible.

Linear Programming

Market Gardener Revisited

The problem was written:

$$\max_{c,t} \quad 900\,c + 1500\,t$$

subject to:

trucking

$$1.5\,c + 2\,t \le 3$$

$$20\,c + 60\,t \le 60$$

fertilizer

1) in standard form, the problem is:

$$\min_{\mathbf{x}} \quad -\begin{bmatrix} 900 & 1500 \end{bmatrix}\mathbf{x}$$

subject to:

$$\begin{bmatrix} 1.5 & 2 \\ 20 & 60 \end{bmatrix}\mathbf{x} \le \begin{bmatrix} 3 \\ 60 \end{bmatrix}$$

$$\mathbf{x} \ge \mathbf{0}$$

where $\mathbf{x} \equiv \begin{bmatrix} c & t \end{bmatrix}^T$.

Linear Programming

Market Gardener Revisited

2) Introduce slack variables (or convert all inequality constraints to equality constraints):

$$\min_{c,t} \quad -900\,c - 1500\,t$$

$$\text{subject to:}$$

trucking

$$\begin{aligned} 1.5\,c + \ 2\,t \ + s_1 &= \ 3 \\ 20\,c + 60\,t \ + s_2 &= 60 \\ c, t, s_1, s_2 &\geq 0 \end{aligned}$$

fertilizer

or in matrix form:

$$\min_{\mathbf{x}} \quad -\begin{bmatrix} 900 & 1500 & 0 & 0 \end{bmatrix}\mathbf{x}$$

$$\text{subject to:}$$

$$\begin{bmatrix} 1.5 & 2 & 1 & 0 \\ 20 & 60 & 0 & 1 \end{bmatrix}\mathbf{x} \ = \ \begin{bmatrix} 3 \\ 60 \end{bmatrix}$$

$$\mathbf{x} \ \geq \ \mathbf{0}$$

where $\mathbf{x} \equiv \begin{bmatrix} c & t & s_1 & s_2 \end{bmatrix}^T$.

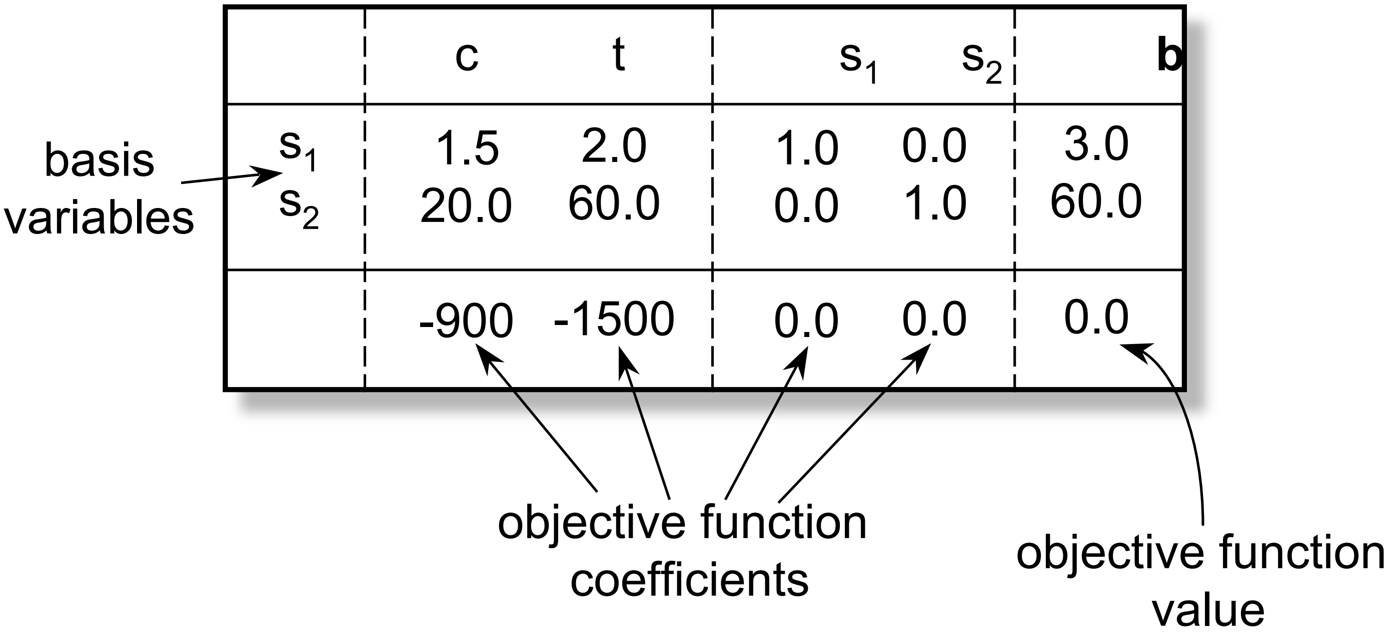
Linear Programming

Market Gardener Revisited

3) Define an initial basis and set up tableau:

- choose the slacks as the initial basic variables,
- this is equivalent to starting at the origin.

the initial tableau is:



	c	t	s_1	s_2	**b**
s_1	1.5	2.0	1.0	0.0	3.0
s_2	20.0	60.0	0.0	1.0	60.0
	-900	-1500	0.0	0.0	0.0

basis variables

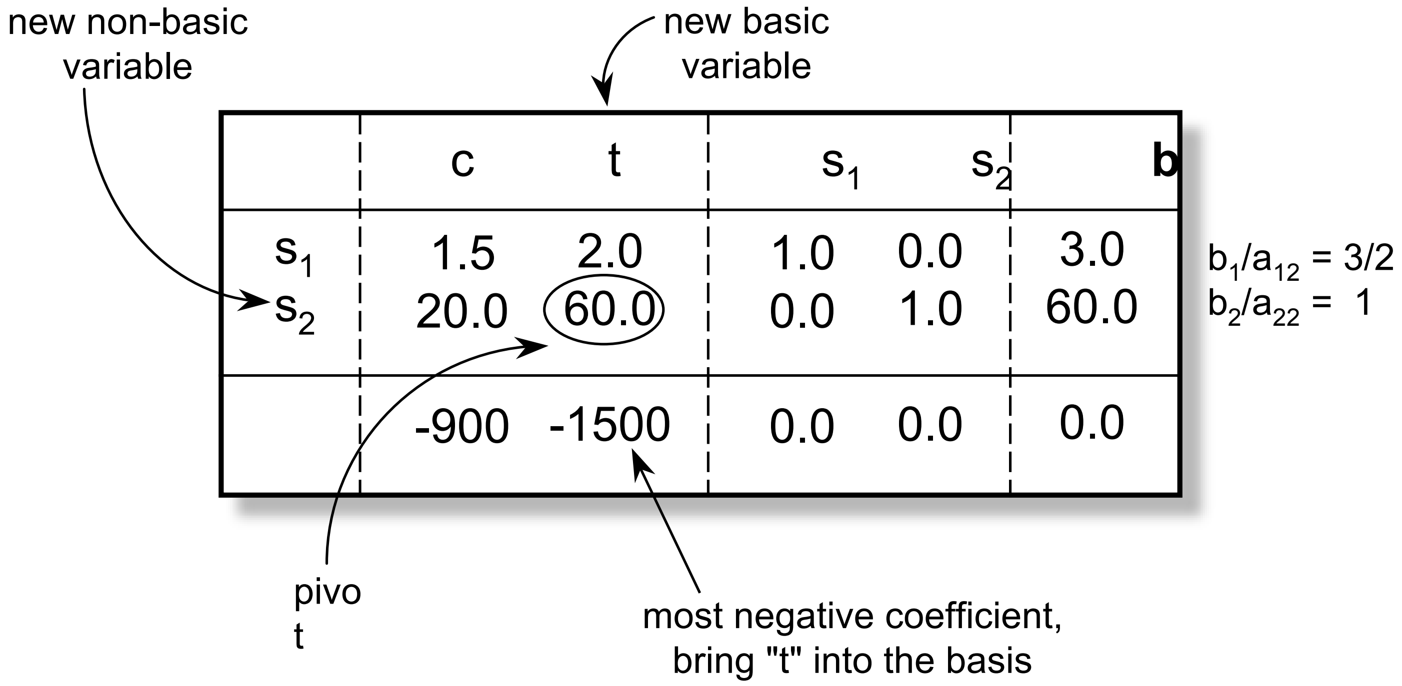
objective function coefficients

objective function value

Market Gardener Revisited

4) Determine the new basis:

- examine the objective function coefficients and choose a variable with a negative weight. (you want to decrease the objective function because you are minimizing. Usually we will choose the most negative weight, but this is not necessary).
- this variable will be brought into the basis.
- divide each element of **b** by the corresponding constraint coefficient of the new basic variable.
- the variable which will be removed from the basis is in the pivot row (given by the smallest positive ratio of b_i/a_{ij}).



	c	t	s_1	s_2	**b**
s_1	1.5	2.0	1.0	0.0	3.0
s_2	20.0	60.0	0.0	1.0	60.0
	-900	-1500	0.0	0.0	0.0

$b_1/a_{12} = 3/2$

$b_2/a_{22} = 1$

Linear Programming

Market Gardener Revisited

5) Transform the constraint equations:

- perform row reduction on the constraint equations to make the pivot element 1 and all other elements of the pivot column 0.

	c	t	s_1	s_2	**b**
s_1	5/6	0.0	1.0	-1/30	1.0
t	1/3	1.0	0.0	1/60	1.0
	-400	0.0	0.0	25.0	1500

i) new row #2 = row #2 / 60

ii) new row #1 = row #1 - 2*new row #2

iii) new row #3 = row #3 + 1500* new row #2

Linear Programming

Market Gardener Revisited

6) Repeat Steps 4 & 5 until no more improvement is possible:

pivot

	c	t	s_1	s_2	**b**
s_1	5/6	0.0	1.0	-1/30	1.0
t	1/3	1.0	0.0	1/60	1.0
	-400	0.0	0.0	25.0	1500

the new tableau is:

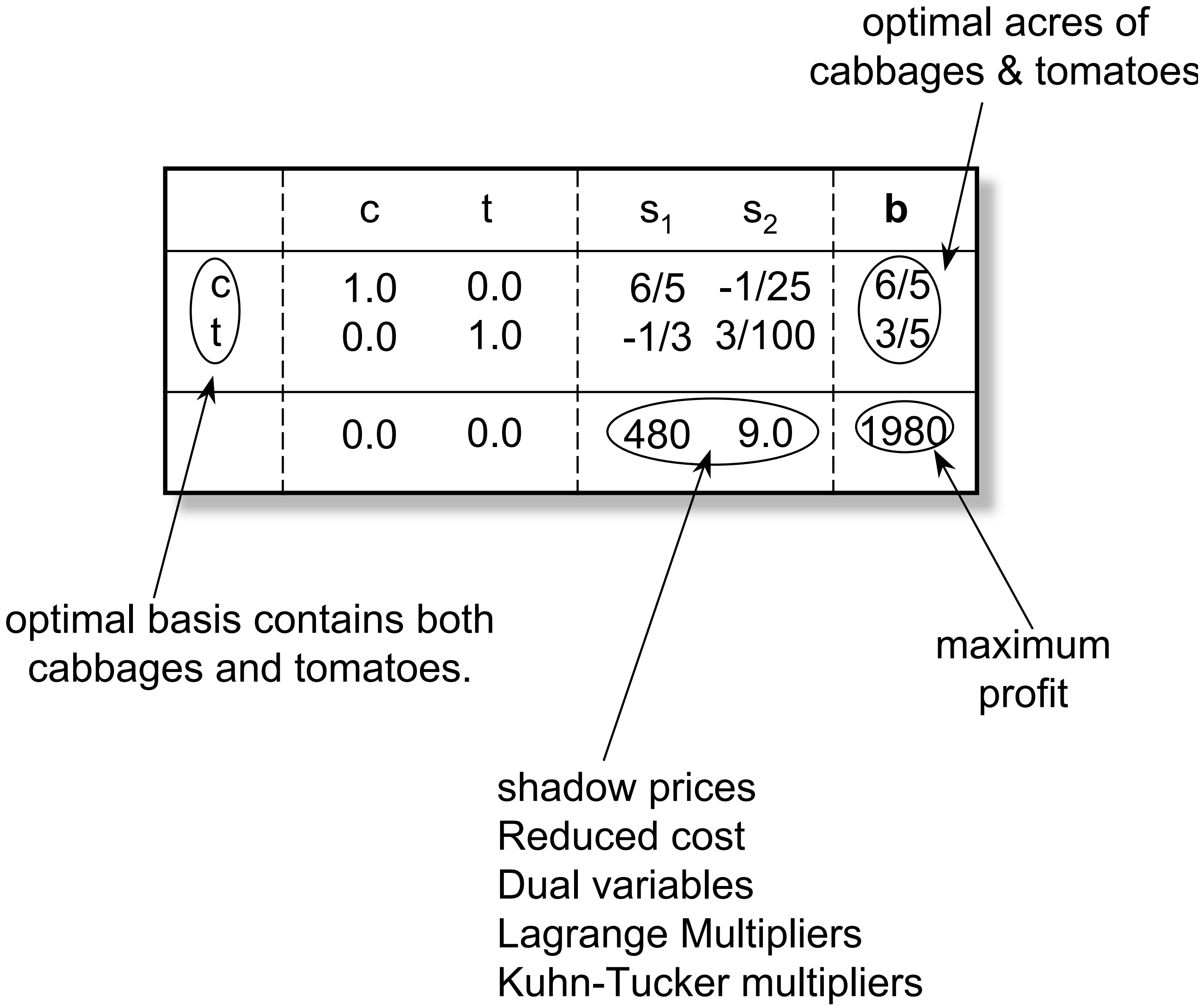
	c	t	s_1	s_2	**b**
c	1.0	0.0	6/5	-1/25	6/5
t	0.0	1.0	-1/3	3/100	3/5
	0.0	0.0	480	9.0	1980

no further improvements are possible, since there are no more negative coefficients in the bottom row of the tableau

Linear Programming

Market Gardener Revisited

The final tableau is:



optimal acres of
cabbages & tomatoes

	c	t	s_1	s_2	**b**
c t	1.0 0.0	0.0 1.0	6/5 -1/3	-1/25 3/100	6/5 3/5
	0.0	0.0	480	9.0	1980

optimal basis contains both
cabbages and tomatoes.

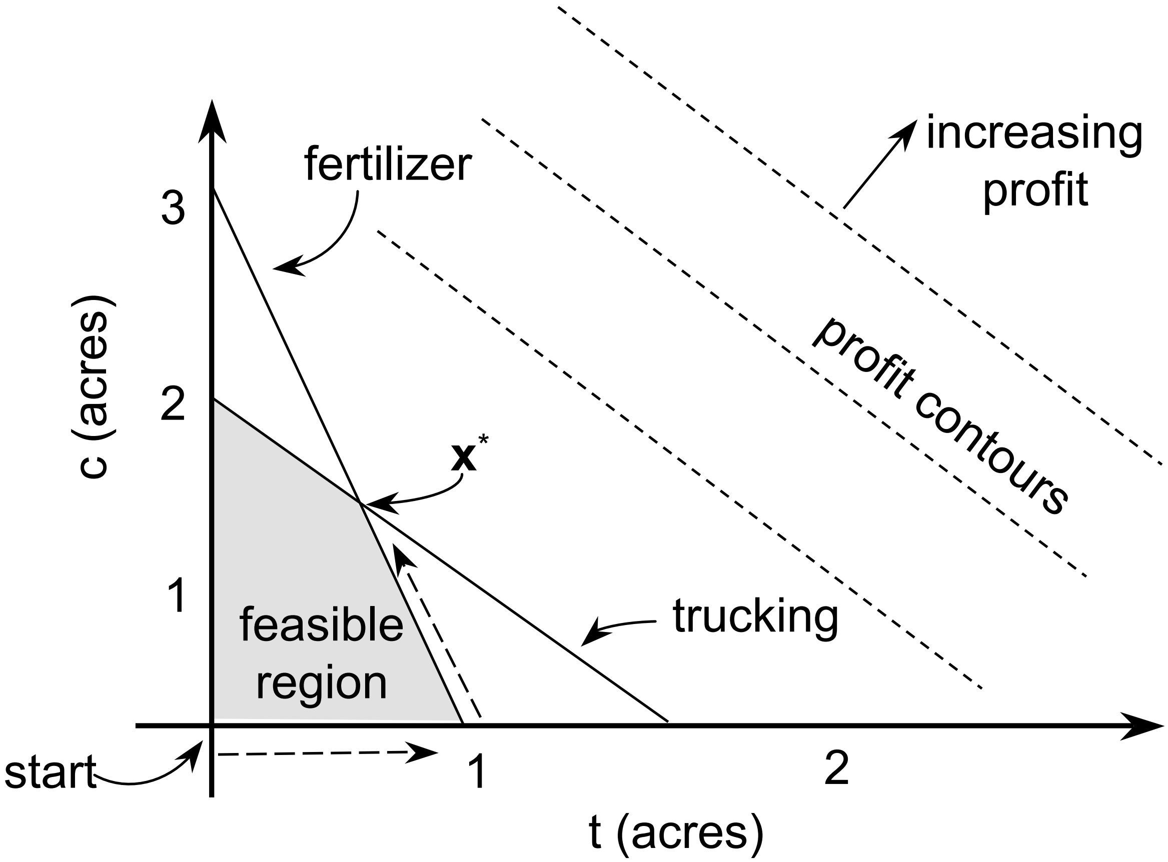
maximum
profit

shadow prices
Reduced cost
Dual variables
Lagrange Multipliers
Kuhn-Tucker multipliers

Linear Programming

Market Gardener Revisited

Recall that we could graph the market garden problem:



We can track how the SIMPLEX algorithm moved through the feasible region. The SIMPLEX algorithm started at the origin. Then moved along the tomato axis (this was equivalent to introducing tomatoes into the basis) to the fertilizer constraint. The algorithm then moved up the fertilizer constraint until it found the intersection with the trucking constraint, which is the optimal solution.

SIMPLEX Algorithm (Matrix Form)

To develop the matrix form for the Linear Programming problem:

$$\min_{\mathbf{x}} \quad \mathbf{c}^{\mathrm{T}}\mathbf{x}$$

subject to:

$$\begin{aligned} \mathbf{A}\mathbf{x} &\leq \mathbf{b} \\ \mathbf{x} &\geq \mathbf{0} \end{aligned}$$

we introduce the slack variables and partition **x**, **A** and **c** as follows:

$$\mathbf{x} = \begin{bmatrix} \mathbf{x}_B \\ --- \\ \mathbf{x}_N \end{bmatrix} \begin{array}{l} \text{basic} \\ \\ \text{non-basic} \end{array}$$

$$\mathbf{A} = \left[\mathbf{B} \;\vdots\; \mathbf{N}\right]$$

$$\mathbf{c} = \begin{bmatrix} \mathbf{c}_B \\ --- \\ \mathbf{c}_N \end{bmatrix}$$

Then, the Linear Programming problem becomes:

$$\min_{\mathbf{x}} \quad \mathbf{c}_B^{\mathrm{T}}\mathbf{x}_B + \mathbf{c}_N^{\mathrm{T}}\mathbf{x}_N$$

subject to:

$$\begin{aligned} \mathbf{B}\mathbf{x}_B + \mathbf{N}\mathbf{x}_N &= \mathbf{b} \\ \mathbf{x}_B,\, \mathbf{x}_N &\geq \mathbf{0} \end{aligned}$$

SIMPLEX Algorithm (Matrix Form)

Feasible values of the basic variables ($\mathbf{x}_B$) can be defined in terms of the values for non-basic variables ($\mathbf{x}_N$):

$$\mathbf{x}_B = \mathbf{B}^{-1}\left[\mathbf{b} - \mathbf{N}\mathbf{x}_N\right]$$

The value of the objective function is given by:

$$P(\mathbf{x}) = \mathbf{c}_B^{\mathrm{T}}\mathbf{B}^{-1}\left[\mathbf{b} - \mathbf{N}\mathbf{x}_N\right] + \mathbf{c}_N^{\mathrm{T}}\mathbf{x}_N$$

or:

$$P(\mathbf{x}) = \mathbf{c}_B^{\mathrm{T}}\mathbf{B}^{-1}\mathbf{b} + \left[\mathbf{c}_N^{\mathrm{T}} - \mathbf{c}_B^{\mathrm{T}}\mathbf{B}^{-1}\mathbf{N}\right]\mathbf{x}_N$$

Then, the tableau we used to solve these problems can be represented as:

	$\mathbf{x}_B^{\mathrm{T}}$	$\mathbf{x}_N^{\mathrm{T}}$	$\mathbf{b}$
$\mathbf{x}_B$	$\mathbf{I}$	$\mathbf{B}^{-1}\mathbf{N}$	$\mathbf{B}^{-1}\mathbf{b}$
	$\mathbf{0}$	$-\left[\mathbf{c}_N^{\mathrm{T}} - \mathbf{c}_B^{\mathrm{T}}\mathbf{B}^{-1}\mathbf{N}\right]$	$\mathbf{c}_B^{\mathrm{T}}\mathbf{B}^{-1}\mathbf{b}$

SIMPLEX Algorithm (Matrix Form)

The Simplex Algorithm is:

1) form the **B** and **N** matrices. Calculate $\mathbf{B}^{-1}$.

2) calculate the shadow prices (reduced costs) of the non-basic variables ($\mathbf{x}_N$):

$$-\left[\mathbf{c}_N^{\mathrm{T}} - \mathbf{c}_B^{\mathrm{T}}\mathbf{B}^{-1}\mathbf{N}\right]$$

3) calculate $\mathbf{B}^{-1}\,\mathbf{N}$ and $\mathbf{B}^{-1}\,\mathbf{b}$.

4) find the pivot element by performing the ratio test using the column corresponding to the most negative shadow price.

5) the pivot column corresponds to the new basic variable and the pivot row corresponds to the new non-basic variable. Modify the **B** and **N** matrices accordingly. Calculate $\mathbf{B}^{-1}$.

6) repeat steps 2 through 5 until there are no negative shadow prices remaining.

SIMPLEX Algorithm (Matrix Form)

This method is computationally inefficient because you must calculate a complete inverse of the matrix B at each iteration of the SIMPLEX algorithm. There are several variations on the Revised SIMPLEX algorithm which attempt to minimize the computations and memory requirements of the method. Examples of these can be found in:

Chvatal
Edgar & Himmelblau (references in §7.7)
Fletcher
Gill, Murray and Wright

Duality & Linear Programming

Many of the traditional advanced topics in LP analysis and research are based on the so-called method of Lagrange. This approach to constrained optimization:

. originated from the study of orbital motion.

. has proven extremely useful in the development of,

- optimality conditions for constrained problems,
- duality theory,
- optimization algorithms,
- sensitivity analysis.

Consider the general Linear programming problem:

$$\min_{\mathbf{x}} \quad \mathbf{c}^T\mathbf{x}$$

subject to:

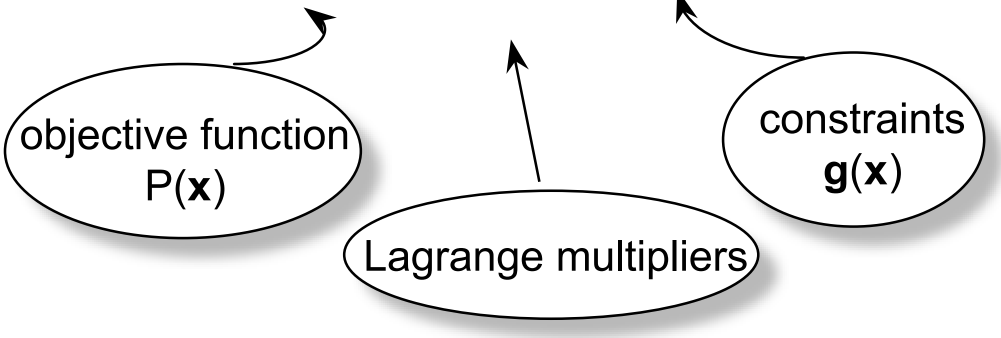
$$\mathbf{Ax} \geq \mathbf{b}$$

This problem is often called the Primal problem to indicate that it is defined in terms of the Primal variables (**x**).

Duality & Linear Programming

You form the Lagrangian of this optimization problem as follows:

$$L(\mathbf{x},\lambda) = \mathbf{c}^{\mathrm{T}}\mathbf{x} - \lambda^{\mathrm{T}}\left[\mathbf{A}\mathbf{x} - \mathbf{b}\right]$$



Notice that the Lagrangian is a scalar function of two sets of variables: the Primal variables **x**, and the Dual variables (or Lagrange multipliers) **λ**. Up until now we have been calling the Lagrange multipliers **λ** the shadow prices.

We can develop the Dual problem first be re-writing the Lagrangian as:

$$L(\mathbf{x},\lambda) = \mathbf{x}^{\mathrm{T}}\mathbf{c} - \left\lfloor \mathbf{x}^{\mathrm{T}}\mathbf{A}^{\mathrm{T}} - \mathbf{b}^{\mathrm{T}} \right\rfloor \lambda$$

which can be re-arranged to yield:

$$L(\lambda,\mathbf{x}) = \mathbf{b}^{\mathrm{T}}\lambda - \mathbf{x}^{\mathrm{T}}\left\lfloor \mathbf{A}^{\mathrm{T}}\lambda - \mathbf{c} \right\rfloor$$

This Lagrangian looks very similar to the previous one except that the Lagrange multipliers **λ** and problem variables **x** have switched places.

Duality & Linear Programming

In fact this re-arranged form is the Lagrangian for the maximization problem:

$$\max_{\lambda} \quad \mathbf{b}^{\mathrm{T}}\boldsymbol{\lambda}$$

subject to:

$$\mathbf{A}^{\mathrm{T}}\boldsymbol{\lambda} \le \mathbf{c}$$

$$\boldsymbol{\lambda} \ge \mathbf{0}$$

This formulation is often called the Dual problem to indicate that it is defined in terms of the Dual variables **λ**.

It is worth noting that in our market gardener example, if we calculate the optimum value of the objective function using the Primal variables $\mathbf{x}^*$:

$$P(\mathbf{x}^*)\Big|_{\mathrm{Primal}} = \mathbf{c}^{\mathrm{T}}\mathbf{x}^* = 900\left(\frac{6}{5}\right) + 1500\left(\frac{3}{5}\right) = 1980$$

and using the Dual variables $\boldsymbol{\lambda}^*$:

$$P(\lambda^*)\Big|_{\mathrm{Dual}} = \mathbf{b}^{\mathrm{T}}\lambda^* = 3(480) + 60(9) = 1980$$

we get the same result. There is a simple explanation for this that we will examine later.

Duality & Linear Programming

Besides being of theoretical interest, the Dual problem formulation can have practical advantages in terms of ease of solution. Consider that the Primal problem was formulated as:

$$\min_{\mathbf{x}} \quad \mathbf{c}^{\mathrm{T}}\mathbf{x}$$

subject to:

$$\mathbf{Ax} \geq \mathbf{b}$$

As we saw earlier, problems with inequality constraints of this form often present some difficulty in determining an initial feasible starting point. They usually require a Phase 1 starting procedure such as the Big 'M' method.

The Dual problem had the form:

$$\max_{\lambda} \quad \mathbf{b}^{\mathrm{T}}\lambda$$

subject to:

$$\mathbf{A}^{\mathrm{T}}\lambda \leq \mathbf{c}$$

$$\lambda \geq \mathbf{0}$$

Problems such as these are usually easy to start since $\boldsymbol{\lambda} = \mathbf{0}$ is a feasible point. Thus no Phase 1 starting procedure is required.

As a result you may want to consider solving the Dual problem, when the origin is not a feasible starting point for the Primal problem.

Optimality Conditions & Linear Programming

Consider the linear programming problem:

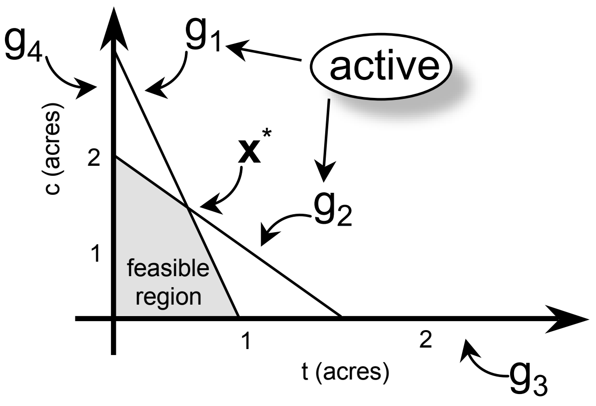
$$\min_{\mathbf{x}} \quad \mathbf{c}^{T}\,\mathbf{x}$$

subject to:

$$\left[\begin{array}{c}\mathbf{M}\\ \hdashline \mathbf{N}\end{array}\right]\mathbf{x} \;\geq\; \left[\begin{array}{c}\mathbf{b}_{M}\\ \hdashline \mathbf{b}_{N}\end{array}\right]\begin{array}{l}\text{active}\\ \\ \text{inactive}\end{array}$$

- if equality constraints are present, they can be used to eliminate some of the elements of **x**, thereby reducing the dimensionality of the optimization problem.
- rows of the coefficient matrix (**M**) are linearly independent.

In our market gardener example, the problem looked like:



g_1 - fertilizer
g_2 - trucking
g_3 - non-negativity tomatoes
g_4 - non-negativity cabbages

Optimality Conditions & Linear Programming

Form the Lagrangian:

$$L(\mathbf{x},\lambda) = \mathbf{c}^{\mathrm{T}}\mathbf{x} - \lambda_{\mathrm{M}}^{\mathrm{T}}\,[\mathbf{M}\mathbf{x} - \mathbf{b}_{\mathrm{M}}] - \lambda_{\mathrm{N}}^{\mathrm{T}}\,[\mathbf{N}\mathbf{x} - \mathbf{b}_{\mathrm{N}}]$$

At the optimum, we know that the Lagrange multipliers (shadow prices) for the inactive inequality constraints are zero (i.e. $\boldsymbol{\lambda}_{\mathrm{N}}=0$). Also, since at the optimum the active inequality constraints are exactly satisfied:

$$\mathbf{M}\mathbf{x}^{*} - \mathbf{b}_{M} = \mathbf{0}$$

Then, notice that at the optimum:

$$L(\mathbf{x}^{*},\lambda^{*}) = P(\mathbf{x}^{*}) = \mathbf{c}^{\mathrm{T}}\mathbf{x}^{*}$$

At the optimum, we have seen that the shadow prices for the active inequality constraints must all be non-negative (i.e. $\boldsymbol{\lambda}_{\mathrm{M}} \geq 0$). These multiplier values told us how much the optimum value of the objective function would increase if the associated constraint was moved into the feasible region, for a minimization problem.

Finally, the optimum ($\mathbf{x}^{*},\boldsymbol{\lambda}^{*}$) must be a stationary point of the Lagrangian:

$$\nabla_{\mathbf{x}} L(\mathbf{x}^{*},\lambda^{*}) = \nabla_{\mathbf{x}}\mathrm{P}(\mathbf{x}^{*}) - \left(\lambda_{M}^{*}\right)^{\mathrm{T}} \mathbf{g}_{M}(\mathbf{x}^{*}) = \mathbf{c}^{\mathrm{T}} - \left(\lambda_{M}^{*}\right)^{\mathrm{T}}\mathbf{M} = \mathbf{0}^{\mathrm{T}}$$

$$\nabla_{\lambda} L(\mathbf{x}^{*},\lambda^{*}) = \left(\mathbf{g}_{M}(\mathbf{x}^{*})\right)^{\mathrm{T}} = \left(\mathbf{M}\mathbf{x}^{*} - \mathbf{b}_{M}\right)^{\mathrm{T}} = \mathbf{0}^{\mathrm{T}}$$

Optimality Conditions & Linear Programming

Thus, necessary and sufficient conditions for an optimum of a Linear Programming problem are:

1) the rows of the active set matrix (**M**) must be linearly independent,

2) the active set are exactly satisfied at the optimum point $\mathbf{x}^*$:

$$\mathbf{M}\mathbf{x}^* = \mathbf{b}_M$$

3) the Lagrange multipliers for the inequality constraints are:

$$\begin{aligned} \boldsymbol{\lambda}_N^* &= \mathbf{0} \\ \boldsymbol{\lambda}_M^* &\geq \mathbf{0} \end{aligned}$$

this is sometimes expressed as:

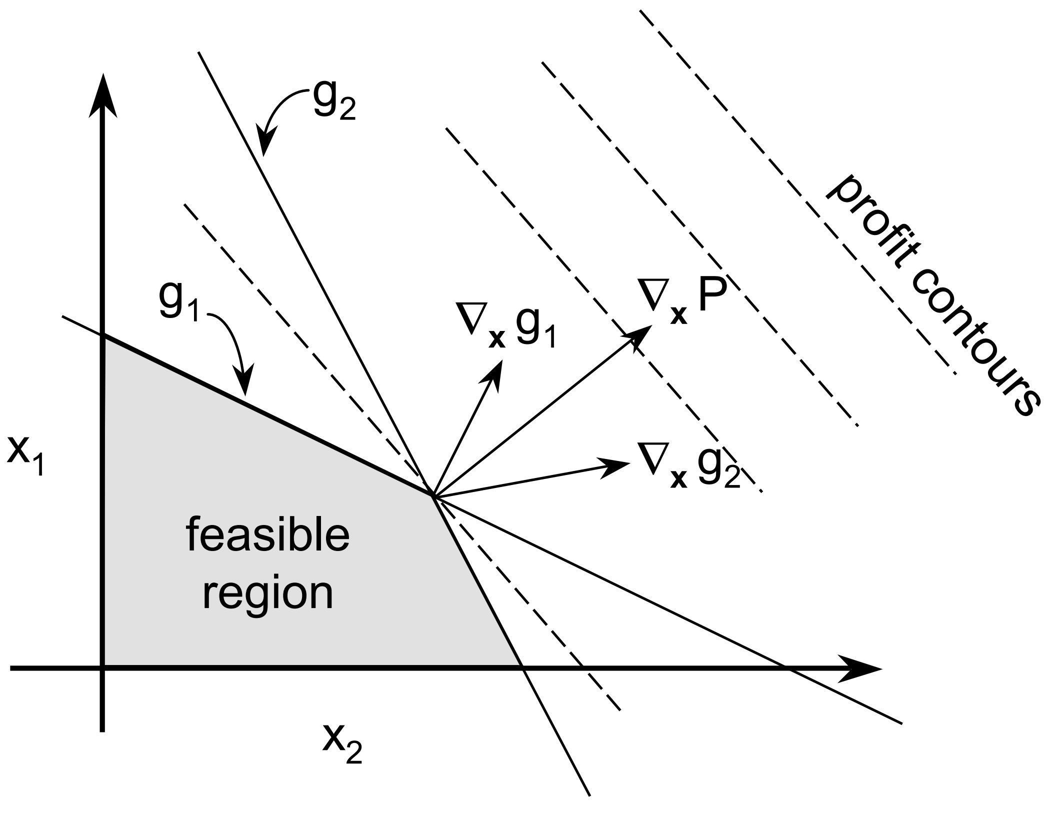
$$\left(\lambda_M^*\right)^{\mathrm{T}}\left[\mathbf{M}\mathbf{x}^* - \mathbf{b}_M\right] + \left(\lambda_N^*\right)^{\mathrm{T}}\left[\mathbf{N}\mathbf{x}^* - \mathbf{b}_N\right] = \mathbf{0}$$

4) the optimum point $(\mathbf{x}^*,\boldsymbol{\lambda}^*)$ is a stationary point of the Lagrangian:

$$\nabla_{\mathbf{x}} L(\mathbf{x}^*, \lambda^*) = \mathbf{0}^{\mathrm{T}}$$

Optimality Conditions & Linear Programming

This 2 variable optimization problem has an optimum at the intersection of the g_1 and g_2 constraints, and can be depicted as:



Linear Programming & Sensitivity Analysis

Usually there is some uncertainty associated with the values used in an optimization problem. After we have solved an optimization problem, we are often interested in knowing how the optimal solution will change as specific problem parameters change. This is called by a variety of names, including:

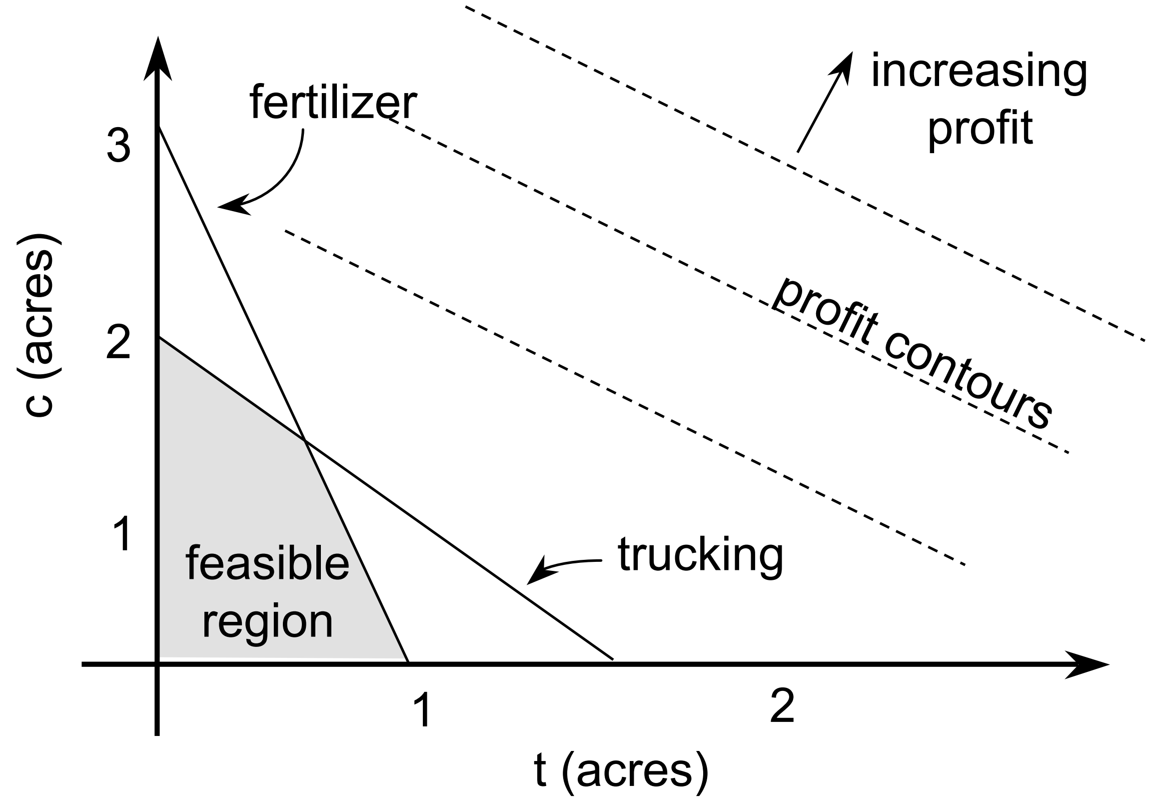
- sensitivity analysis,
- post-optimality analysis,
- parametric programming.

In our Linear Programming problems we have made use of pricing information, but we know such prices can fluctuate. Similarly the demand / availability information we have used is also subject to some fluctuation. Consider that in our market gardener problem, we might be interested in determining how the optimal solution changes when:

- the price of vegetables changes,
- the amount of available fertilizer changes,
- we buy a new truck.

Linear Programming & Sensitivity Analysis

In the market gardener example, we had:



Changes in the pricing information ($\mathbf{c}^T$) affects the slope of the objective function contours.

Changes to the right-hand sides of the inequality constraints (**b**), translates the constraints without affecting their slope.

Changes to the coefficient matrix of the constraints (**A**) affects the slopes of the corresponding constraints.

Linear Programming & Sensitivity Analysis

For the general Linear Programming problem:

$$\min_{\mathbf{x}} \quad \mathbf{c}^{\mathrm{T}}\mathbf{x}$$

subject to:

$$\mathbf{Ax} \geq \mathbf{b}$$

we had the Lagrangian:

$$L(\mathbf{x},\lambda) = P(\mathbf{x})\text{-}\lambda^{\mathrm{T}}\mathbf{g}(\mathbf{x}) = \mathbf{c}^{\mathrm{T}}\mathbf{x} - \lambda^{\mathrm{T}}\left[\mathbf{Ax} - \mathbf{b}\right]$$

or in terms of the active and inactive inequality constraints:

$$L(\mathbf{x},\lambda) = \mathbf{c}^{\mathrm{T}}\mathbf{x} - \lambda_M^{\mathrm{T}}\left[\mathbf{Mx} - \mathbf{b}_M\right] - \lambda_N^{\mathrm{T}}\left[\mathbf{Nx} - \mathbf{b}_N\right]$$

Recall that at the optimum:

$$\nabla_{\mathbf{x}}\ \mathrm{L}(\mathbf{x}^*,\lambda^*) = \mathbf{c}^{\mathrm{T}}\ \text{-}\ \lambda_M^{\mathrm{T}}\ \mathbf{M} = \mathbf{0}^{\mathrm{T}}$$

$$\nabla_{\lambda}\ \mathrm{L}(\mathbf{x}^*,\lambda^*) = \left[\mathbf{Mx}\ \text{-}\ \mathbf{b}_M\right] = \mathbf{0}^{\mathrm{T}}$$

$$\lambda_N^{\mathrm{T}} = \mathbf{0}$$

Notice that the first two sets of equations are linear in the variables of interest ($\mathbf{x}^*$,$\boldsymbol{\lambda}_M{}^*$) and they can be solved to yield:

$$\lambda_M^* \quad = (\mathbf{M}^{\mathrm{T}})^{-1}\ \mathbf{c}$$

$$\mathbf{x}^* \quad = \mathbf{M}^{-1}\ \mathbf{b}_M$$

Then, as we discussed previously, it follows that:

$$\mathrm{P}(\mathbf{x}^{*}) = \mathbf{c}^{\mathrm{T}}\mathbf{x}^{*} = \mathbf{c}^{\mathrm{T}}\mathbf{M}^{-1}\,\mathbf{b}_{M} = (\lambda_{M}^{*})^{\mathrm{T}}\,\mathbf{b}_{M} = \mathbf{b}_{M}^{\mathrm{T}}\lambda_{M}^{*}$$

We now have expressions for the complete solution of our Linear Programming problem in terms of the problem parameters.

In this course we will only consider two types of variation in our nominal optimization problem including:

1) changes in the right-hand side of constraints (**b**). Such changes occur with variation in the supply / demand of materials, product quality requirements and so forth.
2) pricing (**c**) changes. The economics used in optimization are rarely known exactly.

For the more difficult problem of determining the effects of uncertainty in the coefficient matrix (**A**) on the optimization results see:

Gal, T., *Postoptimal Analysis, Parametric Programming and Related Topics*, McGraw-Hill, 1979.

Forbes, J.F., and T.E. Marlin, Model Accuracy for Economic Optimizing Controllers: The Bias Update Case, *Ind. Eng. Chem. Res.*, **33**, pp. 1919-29, 1994.

Linear Programming & Sensitivity Analysis

For small (differential) changes in $\mathbf{b}_M$, which do not change the active constraint set:

shadow prices

$$\nabla_{\mathbf{b}}\, P(\mathbf{x}^*) = \nabla_{\mathbf{b}}\left[(\lambda^*)^{\mathrm{T}}\mathbf{b}\right] = (\lambda^*)^{\mathrm{T}}$$

$$\nabla_{\mathbf{b}}\, \mathbf{x}^* = \nabla_{\mathbf{b}}\left[\mathbf{M}^{-1}\mathbf{b}_M\right] = \begin{bmatrix}\mathbf{M}^{-1}\\ ---\\ \mathbf{0}\end{bmatrix}$$

inverse of constraint coefficient matrix

$$\nabla_{\mathbf{b}}\, \lambda^* = \nabla_{\mathbf{b}}\begin{bmatrix}\mathbf{M}^{-1}\mathbf{c}\\ ---\\ \mathbf{0}\end{bmatrix} = \mathbf{0}$$

Note that the Lagrange multipliers are not a function of $\mathbf{b}_M$ as long as the active constraint set does not change.

For small (differential) changes in $\mathbf{c}$, which do not change the active constraint set:

$$\nabla_{\mathbf{c}}\, P(\mathbf{x}^*) = \nabla_{\mathbf{c}}\left[\mathbf{c}^{\mathrm{T}}\mathbf{x}^*\right] = (\mathbf{x}^*)^{\mathrm{T}}$$

$$\nabla_{\mathbf{c}}\, \mathbf{x}^* = \nabla_{\mathbf{c}}\left[\mathbf{M}^{-1}\mathbf{b}_M\right] = \mathbf{0}$$

$$\nabla_{\mathbf{c}}\, \lambda^* = \nabla_{\mathbf{c}}\begin{bmatrix}(\mathbf{M}^{\mathrm{T}})^{-1}\mathbf{c}\\ ---\\ \mathbf{0}\end{bmatrix} = \begin{bmatrix}(\mathbf{M}^{\mathrm{T}})^{-1}\\ ---\\ \mathbf{0}\end{bmatrix}$$

Note that $\mathbf{x}^*$ is not an explicit function of the vector **c** and will not change so long as the active set remains fixed. If a change in **c** causes an element of $\boldsymbol{\lambda}$ to become negative, then $\mathbf{x}^*$ will jump to a new vertex.

Also it is worth noting that the optimal values $P(\mathbf{x}^*)$, $\mathbf{x}^*$ and $\boldsymbol{\lambda}^*$ are all affected by changes in the elements of the constraint coefficient matrix (**M**). This is beyond the scope of the course and the interested student can find some detail in the references previously provided.

There is a simple geometrical interpretation for most of these results. Consider the 2 variable Linear Programming problem:

$$\min_{x_1,x_2} \qquad c_1x_1 + c_2x_2$$

subject to:

$$\begin{bmatrix} g_1(\mathbf{x}) \\ g_2(\mathbf{x}) \\ \vdots \end{bmatrix} = \begin{bmatrix} a_{11}x_1 + a_{12}x_2 \\ a_{21}x_1 + a_{22}x_2 \\ \vdots \end{bmatrix} = \mathbf{Ax} \geq \mathbf{b}$$

Note that the set of inequality constraints can be expressed:

$$\mathbf{Ax} = \begin{bmatrix} \mathbf{M} \\ --- \\ \mathbf{N} \end{bmatrix}\mathbf{x} = \begin{bmatrix} \nabla_{\mathbf{x}} g_1 \\ \nabla_{\mathbf{x}} g_2 \\ --- \\ \vdots \end{bmatrix}\mathbf{x} \geq \mathbf{b}$$

Linear Programming

Summary

- Linear Programs (LP) have the form:

$$\min_{\mathbf{x}} \quad \mathbf{c}^{T}\,\mathbf{x}$$

$$\text{subject to :}$$

$$\mathbf{A}\,\mathbf{x} \leq \mathbf{b}$$

$$0 \leq \mathbf{x} \leq \mathbf{x}_{\text{upper}}$$

- the solution of a well-posed LP is uniquely determined by the intersection of the active constraints.
- most LPs are solved using the SIMPLEX method.
- commercial computer codes used the Revised SIMPLEX algorithm (more efficient).
- optimization studies include:
  - solution to the nominal optimization problem,
  - a sensitivity study to determine how uncertainty in the assumed problem parameter values affect the optimum.

Interior Point Methods

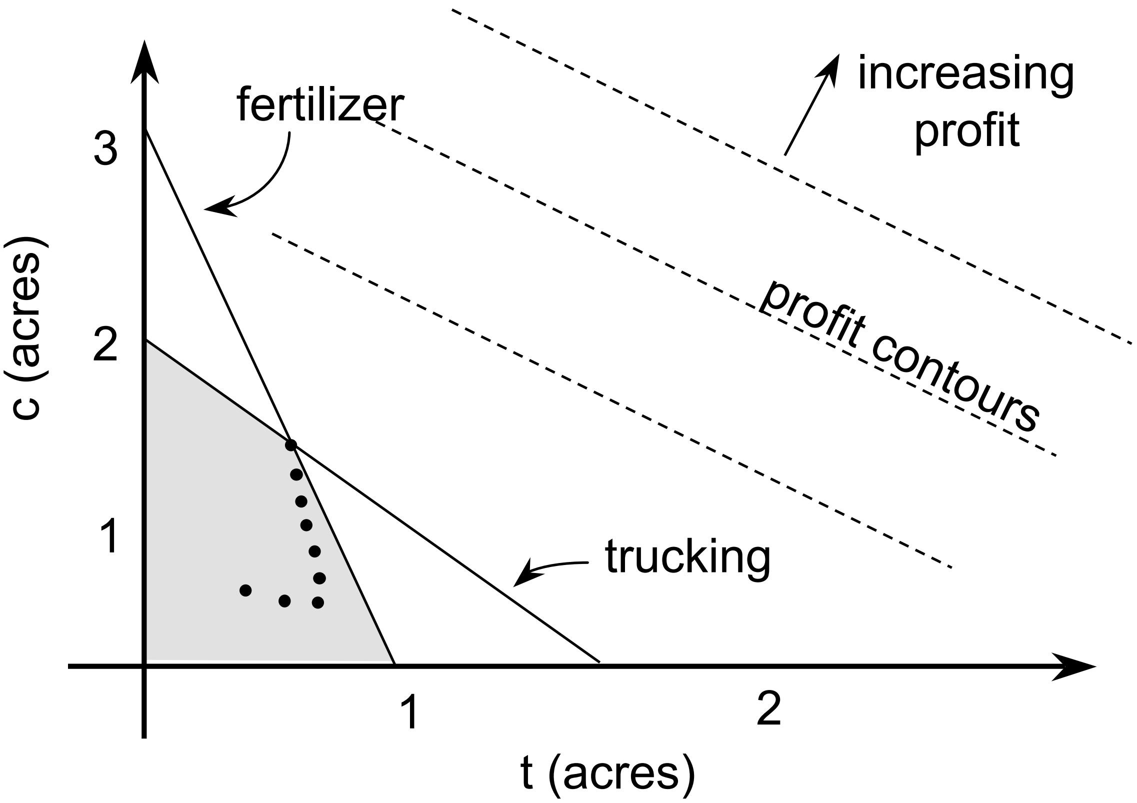
Recall that the SIMPLEX method solved the LP problem by "walking" around the boundary of the feasible region. The algorithm moved from vertex to vertex on the boundary.

Interior Point methods move through the feasible region toward the optimum. This is accomplished by:

1) Converting the LP problem into an unconstrained problem using "Barrier Functions" for the constraints at each new point (iterate),
2) Solving the optimality conditions for the resulting unconstrained problem,
3) Repeating until converged.

This is a very active research area and some exciting developments have occurred in the last two decades.

Interior Point Methods



Linear Programming Problems

1. Kirkman Brothers Ice Cream Parlours sell three different flavours of Dairy Sweet ice milk: chocolate, vanilla, and banana. Due to extremely hot weather and a high demand for its products, Kirkman has run short of its supply of ingredients: milk, sugar and cream. Hence, Kirkman will not be able to fill all of the orders its has from its retail outlets (the ice cream parlours). Due to these circumstances, Kirkman has to make the best amounts of the three flavours given the restricted supply of the basic ingredients. The company will then ration the ice milk to its retail outlets.

   Kirkman has collected the following data on profitability of the various flavours and amounts of each ingredient required for each flavour:

flavour	milk (gal/gal)	Sugar (lbs/gal)	cream (gal/gal)	Profit ($/gal)
chocolate	0.45	0.50	0.10	$1.00
vanilla	0.50	0.40	0.15	$0.90
banana	0.40	0.40	0.20	$0.95

   Kirkman has 180 gal. of milk, 150 lbs. of sugar and 60 gal. of cream available.

Linear Programming Problems

Continued

a) What mix of ice cream flavours will maximize the company's profit. Report your complete solution.
b) Using your solution determine the effects of changes in the availability of milk, sugar and cream on the profit.
c) What is the most effective way to increase your profit?

Workshop Problems - Linear Programming

---

2. A jeweler makes rings, earrings, pins, and necklaces. She wishes to work no more than 40 hours per week. It takes her 2 hours to make a ring, 2 hours to make a pair of earrings, 1 hour to make a pin and 4 hours to make a necklace. She estimates that she can sell no more than 10 rings, 10 pairs of earrings, 15 pins and 3 necklaces in a single week. The jeweler charges $50 for a ring, $80 for a pair of earrings, $25 for a pin, and $200 for a necklace. She would like to determine how many rings, pairs of earrings, pins and necklaces she should make each week in order to produce the largest possible gross earnings.

   a) What mix of jewelry will maximize the company's profit. Report your complete solution.
   b) Using your solution determine the effects of changes in the availability of number of work hours per week, number of hours to required to produce each type of jewelry and the demand for each type of jewelry on the profit.
   c) What is the most effective way to increase your profit?
   d) Comment on whether you think it was appropriate to solve this problem with a conventional LP algorithm.

Then x^* is an inflection point. Otherwise, if the first higher-order, non-zero derivative is even: