School of Computer Science and En
University of New South W

10. LINEAR PROGRAMMING

Linear Programming problems - Example 1

Problem:

- You are given a list of food sources $f_1, f_2, \ldots, f_n$;

Linear Programming problems - Example 1

Problem:

- You are given a list of food sources $f_1, f_2, \ldots, f_n$;
- for each source f_i you are given:

Linear Programming problems - Example 1

Problem:

- You are given a list of food sources $f_1, f_2, \ldots, f_n$;
- for each source f_i you are given:
  -

Linear Programming problems - Example 1

Problem:

- You are given a list of food sources $f_1, f_2, \ldots, f_n$;
- for each source f_i you are given:
  -
  -

Linear Programming problems - Example 1

Problem:

- You are given a list of food sources $f_1, f_2, \ldots, f_n$;
- for each source f_i you are given:
  -
  -
  - ent f_i.

Linear Programming problems - Example 1

Problem:

- You are given a list of food sources $f_1, f_2, \ldots, f_n$;
- for each source f_i you are given:
  -
  -
  -
- Your task: to find a combination of quantities of foo

Linear Programming problems - Example 1

Problem:

- You are given a list of food sources $f_1, f_2, \ldots, f_n$;
- for each source f_i you are given:
  -
  -
  -
- Your task: to find a combination of quantities of foo
  - the total number of calories in all of the chosen f
    recommended daily value of 2000 calories

Linear Programming problems - Example 1

Problem:

- You are given a list of food sources $f_1, f_2, \ldots, f_n$;
- for each source f_i you are given:
  -
  -
  -
- Your task: to find a combination of quantities of foo
  - the total number of calories in all of the chosen f
    recommended daily value of 2000 calories
  - the total intake of each vitamin V_j
    daily intake of w_j milligrams for all $1 \leq j \leq 13$;

Linear Programming problems - Example 1

Problem:

- You are given a list of food sources $f_1, f_2, \ldots, f_n$;
- for each source f_i you are given:
  -
  -
  -
- Your task: to find a combination of quantities of foo
  - the total number of calories in all of the chosen f
    recommended daily value of 2000 calories
  - the total intake of each vitamin V_j
    daily intake of w_j milligrams for all $1 \leq j \leq 13$;
  - the price of all food per day is as low as possible.

Linear Programming problems - Example 1 cont.

- To obtain the corresponding constraints let us assume that we take x_i grams of each food source f_i for $1 \leq i \leq n$. Then:

Linear Programming problems - Example 1 cont.

- To obtain the corresponding constraints let us assume that we take x_i grams of each food source f_i for $1 \leq i \leq n$. Then:
  - the total number of calories must satisfy

$$\sum_{i=1}^{n} x_i c_i = 2000;$$

Linear Programming problems - Example 1 cont.

- To obtain the corresponding constraints let us assume that we take x_i grams of each food source f_i for $1 \leq i \leq n$. Then:
  - the total number of calories must satisfy

$$\sum_{i=1}^{n} x_i c_i = 2000;$$

Linear Programming problems - Example 1 cont.

- To obtain the corresponding constraints let us assume that we take x_i grams of each food source f_i for $1 \leq i \leq n$. Then:
  - the total number of calories must satisfy

$$\sum_{i=1}^{n} x_i c_i = 2000;$$

  - an implicit assumption is that all the quanti
non-negative numbers,

$$x_i \geq 0,$$

Linear Programming problems - Example 1 cont.

- To obtain the corresponding constraints let us assume that we take x_i grams of each food source f_i for $1 \leq i \leq n$. Then:
  - the total number of calories must satisfy

$$\sum_{i=1}^{n} x_i c_i = 2000;$$

  - an implicit assumption is that all the quanti
non-negative numbers,

$$x_i \geq 0,$$

- Our goal is to minimise the objective function which is the total cost

$$y = \sum_{i=1}^{n} x_i p_i.$$

Linear Programming problems - Example 1 cont.

- To obtain the corresponding constraints let us assume that we take x_i grams of each food source f_i for $1 \leq i \leq n$. Then:
  - the total number of calories must satisfy

$$\sum_{i=1}^{n} x_i c_i = 2000;$$

  - an implicit assumption is that all the quanti
non-negative numbers,

$$x_i \geq 0,$$

- Our goal is to minimise the objective function which is the total cost

$$y = \sum_{i=1}^{n} x_i p_i.$$

- Note that all constraints and the objective function, are **linear**.

Linear Programming problems - Example 2

Problem:

- Assume now that you are politician and you want to make certain promises to the electorate which will ensure that your party will win in the forthcoming elections.

Linear Programming problems - Example 2

Problem:

- Assume now that you are politician and you want to make certain promises to the electorate which will ensure that your party will win in the forthcoming elections.
- You can promise that you will build

Linear Programming problems - Example 2

Problem:

- Assume now that you are politician and you want to make certain promises to the electorate which will ensure that your party will win in the forthcoming elections.
- You can promise that you will build
  - a certain number of bridges, each 3 billion a piece;

Linear Programming problems - Example 2

Problem:

- Assume now that you are politician and you want to make certain promises to the electorate which will ensure that your party will win in the forthcoming elections.
- You can promise that you will build
  - a certain number of bridges, each 3 billion a piece;
  -

Linear Programming problems - Example 2

Problem:

- Assume now that you are politician and you want to make certain promises to the electorate which will ensure that your party will win in the forthcoming elections.
- You can promise that you will build
  - a certain number of bridges, each 3 billion a piece;
  -
  -

Linear Programming problems - Example 2

Problem:

- Assume now that you are politician and you want to make certain promises to the electorate which will ensure that your party will win in the forthcoming elections.
- You can promise that you will build
  - a certain number of bridges, each 3 billion a piece;
  -
  -
- You

Linear Programming problems - Example 2

Problem:

- Assume now that you are politician and you want to make certain promises to the electorate which will ensure that your party will win in the forthcoming elections.
- You can promise that you will build
  - a certain number of bridges, each 3 billion a piece;
  -
  -
- You
  -
  suburban votes and 9% of rural votes;

Linear Programming problems - Example 2

Problem:

- Assume now that you are politician and you want to make certain promises to the electorate which will ensure that your party will win in the forthcoming elections.
- You can promise that you will build
  - a certain number of bridges, each 3 billion a piece;
  -
  -
- You
  -
    suburban votes and 9% of rural votes;
  - each rural airport you promise brings you no suburban votes and 15% of rural votes;

Linear Programming problems - Example 2

Problem:

- Assume now that you are politician and you want to make certain promises to the electorate which will ensure that your party will win in the forthcoming elections.
- You can promise that you will build
  - a certain number of bridges, each 3 billion a piece;
  -
  -
- You
  -
    suburban votes and 9% of rural votes;
  - each rural airport you promise brings you no suburban votes and 16% of rural votes;
  - each olympic swimming pool promised bri
    3% of suburban votes and no rural votes.

Linear Programming problems - Example 2

Problem:

- Assume now that you are politician and you want to make certain promises to the electorate which will ensure that your party will win in the forthcoming elections.
- You can promise that you will build
  - a certain number of bridges, each 3 billion a piece;
  -
  -
- You
  -
    suburban votes and 9% of rural votes;
  - each rural airport you promise brings you no suburban votes and 16% of rural votes;
  - each olympic swimming pool promised bri
    3% of suburban votes and no rural votes.
- In order to win, you have to get at least 51% of each of the city, suburban and rural votes.

Linear Programming problems - Example 2

Problem:

- Assume now that you are politician and you want to make certain promises to the electorate which will ensure that your party will win in the forthcoming elections.
- You can promise that you will build
  - a certain number of bridges, each 3 billion a piece;
  -
  -
- You
  -
    suburban votes and 9% of rural votes;
  - each rural airport you promise brings you no suburban votes and 16% of rural votes;
  - each olympic swimming pool promised bri
    3% of suburban votes and no rural votes.
- In order to win, you have to get at least 51% of each of the city, suburban and rural votes.
- You wish to win the election by cleverly making a promise that **appears** that it will blow as small hole in the budget as possible, i.e., that the total cost of your promises is as low as possible.

Linear Programming problems - Example 2

- We can let the number of bridges to be built be x_b, number of airports x_a and the number of swimming pools x_p.

Linear Programming problems - Example 2

- We can let the number of bridges to be built be x_b, number of airports x_a and the number of swimming pools x_p.
- We now see that the problem amounts to minimising the objective $y = 3x_b + 2x_a + x_p$ while making sure that the following constraints are satisfied:

- We can let the number of bridges to be built be x_b, number of airports x_a and the number of swimming pools x_p.
- We now see that the problem amounts to minimising the objective $y = 3x_b + 2x_a + x_p$, while making sure that the following constraints are satisfied:

$$0.05x_b \qquad + 0.12x_p \quad 0.51 \quad \text{(securing majority of city votes)}$$

0.0 s)

0.0

- We can let the number of bridges to be built be x_b, number of airports x_a and the number of swimming pools x_p.
- We now see that the problem amounts to minimising the objective $y = 3x_b + 2x_a + x_p$, while making sure that the following constraints are satisfied:

$0.05x_b \quad + 0.12x_p \quad 0.51$ (securing majority of city votes)

0.0 ... s)

0.0

- However, there is a very significant difference wit

- We can let the number of bridges to be built be x_b, number of airports x_a and the number of swimming pools x_p.
- We now see that the problem amounts to minimising the objective $y = 3x_b + 2x_a + x_p$, while making sure that the following constraints are satisfied:

$$0.05x_b \quad + 0.12x_p \quad 0.51 \quad \text{(securing majority of city votes)}$$

0.0 ... s)

0.0

- However, there is a very significant difference wit
  - you can eat 1.56 grams of chocolate, but

- We can let the number of bridges to be built be x_b, number of airports x_a and the number of swimming pools x_p.
- We now see that the problem amounts to minimising the objective $y = 3x_b + 2x_a + x_p$, while making sure that the following constraints are satisfied:

$$0.05x_b \qquad + 0.12x_p \quad 0.51 \quad \text{(securing majority of city votes)}$$

0.0 ... s)

0.0

- However, there is a very significant difference wit
  - you can eat 1.56 grams of chocolate, but
  - you cannot promise to build 1.56 bridges, 2. swimming pools!

- We can let the number of bridges to be built be x_b, number of airports x_a and the number of swimming pools x_p.
- We now see that the problem amounts to minimising the objective $y = 3x_b + 2x_a + x_p$, while making sure that the following constraints are satisfied:

$0.05x_b \quad + 0.12x_p \quad 0.51$ (securing majority of city votes)

0.0 ... s)

0.0

- However, there is a very significant difference wit
  - you can eat 1.56 grams of chocolate, but
  - you cannot promise to build 1.56 bridges, 2. swimming pools!
- The second example is an example of an **Integer Linear Programming problem**, which requires all the solutions to be integers.

Linear Programming problems - Example 2

- We can let the number of bridges to be built be x_b, number of airports x_a and the number of swimming pools x_p.
- We now see that the problem amounts to minimising the objective $y = 3x_b + 2x_a + x_p$, while making sure that the following constraints are satisfied:

$0.05x_b \quad + 0.12x_p \quad 0.51$ (securing majority of city votes)

0.0 ... s)

0.0

- However, there is a very significant difference wit
  - you can eat 1.56 grams of chocolate, but
  - you cannot promise to build 1.56 bridges, 2. swimming pools!

- The second example is an example of an **Integer Linear Programming problem**, which requires all the solutions to be integers.
- Such problems are MUCH harder to solve than the "plain" Linear Programming problems whose solutions can be real numbers.

Linear Programming problems

- In the **standard form** the *objective* to be maximised is given by

$$\sum_{j=1}^{n} c_j x_j$$

Linear Programming problems

- In the **standard form** the *objective* to be maximised is given by

$$\sum_{j=1}^{n} c_j x_j \tag{1}$$

- the

$$x_j \geq 0, \tag{2}$$

Linear Programming problems

- In the **standard form** the *objective* to be maximised is given by

$$\sum_{j=1}^{n} c_j x_j \tag{1}$$

- the

$$x_j \geq 0, \tag{2}$$

- Let the boldface $\mathbf{x}$ represent a (column) vector, $\mathbf{x} = \langle x_1 \dots x_n \rangle^\mathsf{T}$.

Linear Programming problems

- In the **standard form** the *objective* to be maximised is given by

$$\sum_{j=1}^{n} c_j x_j \tag{2}$$

- the

$$x_j \geq 0,$$

- Let the boldface $\mathbf{x}$ represent a (column) vector, $\mathbf{x} = \langle x_1 \dots x_n \rangle^\mathsf{T}$.
- To get a more compact representation of linear programs we introduce a partial ordering on vectors $\mathbf{x} \in \mathbf{R}^n$ by $\mathbf{x} \leq \mathbf{y}$ if and only if the corresponding inequalities hold coordinate-wise, i.e., if and only if $x_j \leq y_j$ for all $1 \leq j \leq n$.

- Letting $\mathbf{c} = \langle c_1 \ldots c_n \rangle^{\mathsf{T}} \in \mathbf{R}^n$ and $\mathbf{b} = \langle b_1 \ldots b_m \rangle^{\mathsf{T}} \in \mathbf{R}^m$, and letting A be the matrix $A = (a_{ij})$ of size $m \times n$, we get that the above problem can be formulated simply as:

$$\mathbf{x}$$

- Letting $\mathbf{c} = \langle c_1 \ldots c_n \rangle^\intercal \in \mathbf{R}^n$ and $\mathbf{b} = \langle b_1 \ldots b_m \rangle^\intercal \in \mathbf{R}^m$, and letting A be the matrix $A \equiv (a_{ij})$ of size $m \times n$, we get that the above problem can be formulated simply as:

  and

$$\mathbf{x}$$

- Thus, to specify a Linear Programming optimisation problem we have to provide a triplet $(A, \mathbf{b}, \mathbf{c})$;

Linear Programming

- Letting $\mathbf{c} = \langle c_1 \ldots c_n \rangle^{\mathsf{T}} \in \mathbf{R}^n$ and $\mathbf{b} = \langle b_1 \ldots b_m \rangle^{\mathsf{T}} \in \mathbf{R}^m$, and letting A be the matrix $A \equiv (a_{ij})$ of size $m \times n$, we get that the above problem can be formulated simply as:
  -
  -

  and

$$\mathbf{x}$$

- Thus, to specify a Linear Programming optimisation we have to provide a triplet $(A, \mathbf{b}, \mathbf{c})$;
- This is the usual form which is accepted by most standard LP solvers.

- The value of the objective for any value of the variables which makes the constraints satisfied is called a *feasible solution* of the LP problem.

Linear Programming

- The value of the objective for any value of the variables which makes the constraints satisfied is called a *feasible solution* of the LP problem.
- Equality constraints of the form $\sum_{i=1}^{n} a_{ij}x_i = b_j$ can be replaced by two inequalities: $\sum_{i=1}^{n} a_{ij}x_i \geq b_j$ and $\sum_{i=1}^{n} a_{ij}x_i \leq b_j$; thus, we can assume that all constraints are inequalities.

Linear Programming

- The value of the objective for any value of the variables which makes the constraints satisfied is called a *feasible solution* of the LP problem.
- Equality constraints of the form $\sum_{i=1}^{n} a_{ij}x_i = b_j$ can be replaced by two inequalities: $\sum_{i=1}^{n} a_{ij}x_i \geq b_j$ and $\sum_{i=1}^{n} a_{ij}x_i \leq b_j$; thus, we can assume that all constraints are inequalities.
- In ge
does
vari

Linear Programming

- The value of the objective for any value of the variables which makes the constraints satisfied is called a *feasible solution* of the LP problem.
- Equality constraints of the form $\sum_{i=1}^{n} a_{ij}x_i = b_j$ can be replaced by two inequalities: $\sum_{i=1}^{n} a_{ij}x_i \geq b_j$ and $\sum_{i=1}^{n} a_{ij}x_i \leq b_j$; thus, we can assume that all constraints are inequalities.
- In ge
does
vari
- Ho
vari

Linear Programming

- The value of the objective for any value of the variables which makes the constraints satisfied is called a *feasible solution* of the LP problem.
- Equality constraints of the form $\sum_{i=1}^{n} a_{ij}x_i = b_j$ can be replaced by two inequalities: $\sum_{i=1}^{n} a_{ij}x_i \geq b_j$ and $\sum_{i=1}^{n} a_{ij}x_i \leq b_j$; thus, we can assume that all constraints are inequalities.
- In ge
does
vari
- Ho
vari
- This poses no problem, because each occurrence o
variable x_j can be replaced by the expression $x_j^* - x_j^*$ are new variables satisfying the constraints

Linear Programming

- The value of the objective for any value of the variables which makes the constraints satisfied is called a *feasible solution* of the LP problem.
- Equality constraints of the form $\sum_{i=1}^{n} a_{ij}x_i = b_j$ can be replaced by two inequalities: $\sum_{i=1}^{n} a_{ij}x_i \geq b_j$ and $\sum_{i=1}^{n} a_{ij}x_i \leq b_j$; thus, we can assume that all constraints are inequalities.
- In ge
does
vari
- Ho
vari
- This poses no problem, because each occurrence o
variable x_j can be replaced by the expression $x_j^* - x_j^*$ are new variables satisfying the constraints
- If $\mathbf{x} = (x_1, \ldots, x_n)$ is a vector, we let $|\mathbf{x}|$ $|x_1|, \ldots, |x_n|$ problems are naturally translated into constraints of the form $|A\mathbf{x}| \leq \mathbf{b}$. This also poses no problem because we can replace such constraints with two linear constraints: $A\mathbf{x} \leq \mathbf{b}$ and $-A\mathbf{x} \leq \mathbf{b}$ because $|x| \leq y$ if and only if $x \leq y$ and $-x \leq y$.

Linear Programming - Standard Form

- Standard Form: maximize $\mathbf{c}^T\mathbf{x}$ subject to $A\mathbf{x} \leq \mathbf{b}$ and $\mathbf{x} \geq \mathbf{0}$.

Linear Programming - Standard Form

- Standard Form: maximize $\mathbf{c}^T\mathbf{x}$ subject to $A\mathbf{x} \leq \mathbf{b}$ and $\mathbf{x} \geq \mathbf{0}$.
- Any vector $\mathbf{x}$ which satisfies the two constraints is called a *feasible solution*, regardless of what the corresponding objective value $\mathbf{c}^T\mathbf{x}$ might be.

Linear Programming - Standard Form

- Standard Form: maximize $\mathbf{c}^T\mathbf{x}$ subject to $A\mathbf{x} \leq \mathbf{b}$ and $\mathbf{x} \geq \mathbf{0}$.
- Any vector $\mathbf{x}$ which satisfies the two constraints is called a *feasible solution*, regardless of what the corresponding objective value $\mathbf{c}^T\mathbf{x}$ might be.
- As an example, let us consider the following optimisation problem:

Linear Programming - Standard Form

- Standard Form: maximize $\mathbf{c}^T\mathbf{x}$ subject to $A\mathbf{x} \leq \mathbf{b}$ and $\mathbf{x} \geq \mathbf{0}$.
- Any vector $\mathbf{x}$ which satisfies the two constraints is called a *feasible solution*, regardless of what the corresponding objective value $\mathbf{c}^T\mathbf{x}$ might be.
- As an example, let us consider the following optimisation problem:

$$\text{maximize} \quad z(x_1, x_2, x_3) = 3x_1 + x_2 + 2x_3 \tag{3}$$

subj

$$\leq 30 \tag{4}$$

$$\leq 24 \tag{5}$$

Linear Programming - Standard Form

- Standard Form: maximize $\mathbf{c}^T\mathbf{x}$ subject to $A\mathbf{x} \leq \mathbf{b}$ and $\mathbf{x} \geq \mathbf{0}$.
- Any vector $\mathbf{x}$ which satisfies the two constraints is called a *feasible solution*, regardless of what the corresponding objective value $\mathbf{c}^T\mathbf{x}$ might be.
- As an example, let us consider the following optimisation problem:

$$\text{maximize} \quad z(x_1, x_2, x_3) = 3x_1 + x_2 + 2x_3 \tag{3}$$

subj

$$\leq 30 \tag{4}$$

$$\leq 24 \tag{5}$$

- How large can the value of the objective $+ 2x_3$ be, without violating the constraints?

Linear Programming - Standard Form

- Standard Form: maximize $\mathbf{c}^T\mathbf{x}$ subject to $A\mathbf{x} \leq \mathbf{b}$ and $\mathbf{x} \geq \mathbf{0}$.
- Any vector $\mathbf{x}$ which satisfies the two constraints is called a *feasible solution*, regardless of what the corresponding objective value $\mathbf{c}^T\mathbf{x}$ might be.
- As an example, let us consider the following optimisation problem:

maximize $z(x_1, x_2, x_3) = 3x_1 + x_2 + 2x_3$ (3)

subj

≤ 30 (4)

≤ 24 (5)

$+ 2x_3$

- How large can the value of the objective be, without violating the constraints?
- If we add inequalities (4) and (5), we get

$$3x_1 + 3x_2 + 8x_3 \leq 54 \quad (8)$$

Linear Programming - Standard Form

- Standard Form: maximize $\mathbf{c}^T\mathbf{x}$ subject to $A\mathbf{x} \leq \mathbf{b}$ and $\mathbf{x} \geq \mathbf{0}$.
- Any vector $\mathbf{x}$ which satisfies the two constraints is called a *feasible solution*, regardless of what the corresponding objective value $\mathbf{c}^T\mathbf{x}$ might be.
- As an example, let us consider the following optimisation problem:

maximize $z(x_1, x_2, x_3) = 3x_1 + x_2 + 2x_3$ (3)

subj

≤ 30 (4)

≤ 24 (5)

- How large can the value of the objective $+ 2x_3$ be, without violating the constraints?
- If we add inequalities (4) and (5), we get

$$3x_1 + 3x_2 + 8x_3 \leq 54 \quad (8)$$

- Since all variables are constrained to be non-negative, we are assured that

$$3x_1 + x_2 + 2x_3 \leq 3x_1 + 3x_2 + 8x_3 \leq 54$$

Linear Programming - Standard Form

$$\begin{aligned}
&\text{maximize:} && z(x_1, x_2, x_3) = 3x_1 + x_2 + 2x_3 && (3)\\
&\text{with constraints:} && x_1 + x_2 + 3x_3 \leq 30 && (4)\\
& && 2x_1 + 2x_2 + 5x_3 \leq 24 && (5)\\
& && 4x_1 + x_2 + 2x_3 \leq 36 && (6)
\end{aligned}$$

- Thu $z(x_1$

Linear Programming - Standard Form

$$\begin{aligned}
&\text{maximize:} && z(x_1, x_2, x_3) = 3x_1 + x_2 + 2x_3 && (3)\\
&\text{with constraints:} && x_1 + x_2 + 3x_3 \leq 30 && (4)\\
& && 2x_1 + 2x_2 + 5x_3 \leq 24 && (5)\\
& && 4x_1 + x_2 + 2x_3 \leq 36 && (6)
\end{aligned}$$

- Thu
  $z(x_1$

- Can
  $y_1, y_2, y_3 \geq 0$ to be used to for a linear combination o

$$\begin{aligned}
y_1(x_1 + x_2 + 3 \\
y_2(2x_1 + 2x_2 + 5 \\
y_3(4x_1 + x_2 + 2x_3) &\leq 36y_3
\end{aligned}$$

Linear Programming - Standard Form

$$\begin{aligned}
&\text{maximize:} && z(x_1, x_2, x_3) = 3x_1 + x_2 + 2x_3 && (3)\\
&\text{with constraints:} && x_1 + x_2 + 3x_3 \leq 30 && (4)\\
& && 2x_1 + 2x_2 + 5x_3 \leq 24 && (5)\\
& && 4x_1 + x_2 + 2x_3 \leq 36 && (6)
\end{aligned}$$

- Thu
  $z(x_1$

- Can
  $y_1, y_2, y_3 \geq 0$ to be used to for a linear combination o

$$\begin{aligned}
y_1(x_1 + x_2 + 3 \\
y_2(2x_1 + 2x_2 + 5 \\
y_3(4x_1 + x_2 + 2x_3) &\leq 36y_3
\end{aligned}$$

- Then, summing up all these inequalities and factoring, we get

$$x_1(y_1 + 2y_2 + 4y_3) + x_2(y_1 + 2y_2 + y_3) + x_3(3y_1 + 5y_2 + 2y_3) \leq 30y_1 + 24y_2 + 36y_3$$

Linear Programming - Standard Form

$$\text{maximize:} \quad z(x_1, x_2, x_3) = 3x_1 + x_2 + 2x_3 \tag{3}$$

$$\text{with constraints:} \quad x_1 + x_2 + 3x_3 \leq 30 \tag{4}$$

$$2x_1 + 2x_2 + 5x_3 \leq 24 \tag{5}$$

$$4x_1 + x_2 + 2x_3 \leq 36 \tag{6}$$

$$x_1, x_2, x_3 \geq 0 \tag{7}$$

- So w

$x_1(y_1 + 2y_2$ $1 + 24y_2 + 36y_3$ (9)

Linear Programming - Standard Form

$$\begin{aligned}
&\text{maximize:} && z(x_1, x_2, x_3) = 3x_1 + x_2 + 2x_3 && (3)\\
&\text{with constraints:} && x_1 + x_2 + 3x_3 \leq 30 && (4)\\
& && 2x_1 + 2x_2 + 5x_3 \leq 24 && (5)\\
& && 4x_1 + x_2 + 2x_3 \leq 36 && (6)\\
& && x_1, x_2, x_3 \geq 0 && (7)
\end{aligned}$$

- So w

$x_1(y_1 + 2y_2$ $1 + 24y_2 + 36y_3$ (9)

- If we c
  and y_3 so that:

$$\begin{aligned}
y_1 + 2y_2 + 4 \\
y_1 + 2y_2 + \\
3y_1 + 5y_2 + 2
\end{aligned}$$

y_1, y_2

then

$$3x_3 + x_2 + 2x_3 \leq x_1(y_1 + 2y_2 + 4y_3) + x_2(y_1 + 2y_2 + y_3) + x_3(3y_1 + 5y_2 + 2y_3)$$

Linear Programming - Standard Form

maximize: $z(x_1, x_2, x_3) = 3x_1 + x_2 + 2x_3$ (3)

with constraints:

$$x_1 + x_2 + 3x_3 \leq 30 \quad (4)$$

$$2x_1 + 2x_2 + 5x_3 \leq 24 \quad (5)$$

$$4x_1 + x_2 + 2x_3 \leq 36 \quad (6)$$

$$x_1, x_2, x_3 \geq 0 \quad (7)$$

- So w

$x_1(y_1 + 2y_2$ $1 + 24y_2 + 36y_3$ (9)

- If we c
  and y_3 so that:

y_1, y_2

$$y_1 + 2y_2 + 4$$
$$y_1 + 2y_2 +$$
$$3y_1 + 5y_2 + 2$$

then

$$3x_3 + x_2 + 2x_3 \leq x_1(y_1 + 2y_2 + 4y_3) + x_2(y_1 + 2y_2 + y_3) + x_3(3y_1 + 5y_2 + 2y_3)$$

Combining this with (9) we get:

$$30y_1 + 24y_2 + 36y_3 \geq 3x_1 + x_2 + 2x_3 = z(x_1, x_2, x_3)$$

Linear Programming - Standard Form

- Consequently, in order to find as tight upper bound for our objective $z(x_1, x_2, x_3)$ of the problem P:

$$\begin{aligned} &\text{maximize:} && z(x_1, x_2, x_3) = 3x_1 + x_2 + 2x_3 && (3)\\ &\text{with constraints:} && x_1 + x_2 + 3x_3 \leq 30 && (4)\\ & && 2x_1 + 2x_2 + 5x_3 \leq 24 && (5) \end{aligned}$$

Linear Programming - Standard Form

- Consequently, in order to find as tight upper bound for our objective $z(x_1, x_2, x_3)$ of the problem P:

$$\text{maximize:} \quad z(x_1, x_2, x_3) = 3x_1 + x_2 + 2x_3 \tag{3}$$

$$\text{with constraints:} \quad x_1 + x_2 + 3x_3 \leq 30 \tag{4}$$

$$2x_1 + 2x_2 + 5x_3 \leq 24 \tag{5}$$

we h

$$\text{minimise:} \quad z^*(y_1, y_2, y_3) = 30 \tag{10}$$

with constraints:

$$y_1 \quad y_2 \quad y_3 \geq$$

$$y_1, y_2, y_3 \geq 0 \tag{14}$$

Linear Programming - Standard Form

- Consequently, in order to find as tight upper bound for our objective $z(x_1, x_2, x_3)$ of the problem P:

$$\text{maximize:} \quad z(x_1, x_2, x_3) = 3x_1 + x_2 + 2x_3 \tag{3}$$

$$\text{with constraints:} \quad x_1 + x_2 + 3x_3 \leq 30 \tag{4}$$

$$2x_1 + 2x_2 + 5x_3 \leq 24 \tag{5}$$

we h

$$\text{minimise:} \quad z^*(y_1, y_2, y_3) = 30 \tag{10}$$

with constraints:

$$y_1 \quad y_2 \quad y_3 \geq$$

$$y_1, y_2, y_3 \geq 0 \tag{14}$$

then $z^*(y_1, y_2, y_3) = 30y_1 + 24y_2 + 36y_3 \geq 3x_1 + x_2 + 2x_3 = z(x_1, x_2, x_3)$ will be a tight upper bound for $z(x_1, x_2, x_3)$

- The new problem P^* is called the *dual problem* for the problem P.

Linear Programming - Standard Form

- Let us now repeat the whole procedure with P^* in place of P, i.e., let us find the dual program $(P^*)^*$ of P^*.

Linear Programming - Standard Form

- Let us now repeat the whole procedure with P^* in place of P, i.e., let us find the dual program $(P^*)^*$ of P^*.
- We are now looking for $z_1, z_2, z_3 \geq 0$ to multiply inequalities (11)-(13) and obtain

$$z_1(y_1 - 2y_2 + 4y_3) \geq 3z_1$$
$$z_2(y_1 + 2y_2 + y_3) \geq z_2$$

Linear Programming - Standard Form

- Let us now repeat the whole procedure with P^* in place of P, i.e., let us find the dual program $(P^*)^*$ of P^*.
- We are now looking for $z_1, z_2, z_3 \geq 0$ to multiply inequalities (11)-(13) and obtain

$$z_1(y_1 - 3y_2 + 4y_3) \geq 3z_1$$
$$z_2(y_1 + 2y_2 + y_3) \geq z_2$$

- Su

$$y_1(z_1 + z \quad + x_2 + 2z_3 \tag{15}$$

Linear Programming - Standard Form

- Let us now repeat the whole procedure with P^* in place of P, i.e., let us find the dual program $(P^*)^*$ of P^*.
- We are now looking for $z_1, z_2, z_3 \geq 0$ to multiply inequalities (11)-(13) and obtain

$$z_1(y_1 - 2y_2 + 4y_3) \geq 3z_1$$
$$z_2(y_1 + 2y_2 + y_3) \geq z_2$$

- Su

$$y_1(z_1 + z \quad + x_2 + 2z_3 \tag{15}$$

- If we choose multipliers z_1, z_2, z_3 so that

$$z_1 + z_2 + 3 \tag{16}$$
$$2z_1 + 2z_2 + 5 \tag{17}$$
$$4z_1 + z_2 + 2z_3 \leq \tag{18}$$

Linear Programming - Standard Form

- Let us now repeat the whole procedure with P^* in place of P, i.e., let us find the dual program $(P^*)^*$ of P^*.
- We are now looking for $z_1, z_2, z_3 \geq 0$ to multiply inequalities (11)-(13) and obtain

$$z_1(y_1 - 2y_2 + 4y_3) \geq 3z_1$$
$$z_2(y_1 + 2y_2 + y_3) \geq z_2$$

- Su

$$y_1(z_1 + z \quad + x_2 + 2z_3 \tag{15}$$

- If we choose multipliers z_1, z_2, z_3 so that

$$z_1 + z_2 + 3 \tag{16}$$
$$2z_1 + 2z_2 + 5 \tag{17}$$
$$4z_1 + z_2 + 2z_3 \leq \tag{18}$$

we will have:

$$y_1(z_1 + z_2 + 3z_3) + y_2(2z_1 + 2z_2 + 5z_3) + y_3(4z_1 + z_1 + 2z_3) \leq 30y_1 + 24y_2 + 36y_3$$

Linear Programming - Standard Form

- Let us now repeat the whole procedure with P^* in place of P, i.e., let us find the dual program $(P^*)^*$ of P^*.
- We are now looking for $z_1, z_2, z_3 \geq 0$ to multiply inequalities (11)-(13) and obtain

$$z_1(y_1 - 2y_2 + 4y_3) \geq 3z_1$$
$$z_2(y_1 + 2y_2 + y_3) \geq z_2$$

- Su

$$y_1(z_1 + z \quad + x_2 + 2z_3 \tag{15}$$

- If we choose multipliers z_1, z_2, z_3 so that

$$z_1 + z_2 + 3 \tag{16}$$
$$2z_1 + 2z_2 + 5 \tag{17}$$
$$4z_1 + z_2 + 2z_3 \leq \tag{18}$$

we will have:

$$y_1(z_1 + z_2 + 3z_3) + y_2(2z_1 + 2z_2 + 5z_3) + y_3(4z_1 + z_1 + 2z_3) \leq 30y_1 + 24y_2 + 36y_3$$

- Combining this with (15) we get

$$3z_1 + z_2 + 2z_3 \leq 30y_1 + 24y_2 + 36y_3$$

- Consequently, finding the dual program $(P^*)^*$ of P^* amounts to maximising the objective $3z_1 + z_2 + 2z_3$ subject to the constraints

$$z_1 + z_2 + 3z_3 \leq 30$$
$$2z_1 + 2z_2 + 5z_3 \leq 24$$

Linear Programming - Standard Form

- Consequently, finding the dual program $(P^*)^*$ of P^* amounts to maximising the objective $3z_1 + z_2 + 2z_3$ subject to the constraints

$$z_1 + z_2 + 3z_3 \leq 30$$
$$2z_1 + 2z_2 + 5z_3 \leq 24$$

- But n xactly our starting program P. Thus, the dual program $(P^*)^*$ for program P^* is just P itself, i.e., $(P^*)^* = P$.

Linear Programming - Standard Form

- Consequently, finding the dual program $(P^*)^*$ of P^* amounts to maximising the objective $3z_1 + z_2 + 2z_3$ subject to the constraints

$$z_1 + z_2 + 3z_3 \leq 30$$
$$2z_1 + 2z_2 + 5z_3 \leq 24$$

- But n xactly our starting program P. Thus, the dual program $(P^*)^*$ for program P^* is just P itself, i.e., $(P^*)^* = P$.
- So, at the first sight, looking for the multipliers much, because it only reduced a maximisation pr hard minimisation problem.

Linear Programming - Standard Form

- Consequently, finding the dual program $(P^*)^*$ of P^* amounts to maximising the objective $3z_1 + z_2 + 2z_3$ subject to the constraints

$$z_1 + z_2 + 3z_3 \leq 30$$
$$2z_1 + 2z_2 + 5z_3 \leq 24$$

- But n xactly our starting program P. Thus, the dual program $(P^*)^*$ for program P^* is just P itself, i.e., $(P^*)^* = P$.
- So, at the first sight, looking for the multipliers much, because it only reduced a maximisation pr hard minimisation problem.
- It is now useful to remember how we proved that the Ford - Fulkerson Max Flow algorithm in fact produces a **maximal flow**, by showing that it terminates only when we reach the capacity of a **minimal cut**.

Linear Programming - primal/dual problem forms

- The original, *primal* Linear Program P and its *dual* Linear Program can be easily described in the most general case:

$$P: \text{ maximize } z(\mathbf{x}) = \sum_{j=1}^{n} c_j x_j,$$

$$\text{subject to the constraints } \sum_{j=1}^{n} a_{ij} x_j \leq b_i; \quad 1 \leq i \leq m$$

$$\text{subject to the constraints } \sum_{i=1}^{m} a_{ij} y_i \geq c_j; \quad 1 \leq j \leq n$$

$$y_1, \ldots, y_m \geq 0,$$

Linear Programming - primal/dual problem forms

- The original, *primal* Linear Program P and its *dual* Linear Program can be easily described in the most general case:

$$P: \text{ maximize } z(\mathbf{x}) = \sum_{j=1}^{n} c_j x_j,$$

$$\text{subject to the constraints } \sum_{j=1}^{n} a_{ij} x_j \leq b_i; \quad 1 \leq i \leq m$$

$$\text{subject to the constraints } \sum_{i=1}^{m} a_{ij} y_i \geq c_j; \quad 1 \leq j \leq n$$

$$y_1, \ldots, y_m \geq 0,$$

or, in matrix form,

$$P: \text{ maximize } z(\mathbf{x}) = \mathbf{c}^{\mathsf{T}}\mathbf{x}, \text{ subject to the constraints } A\mathbf{x} \leq \mathbf{b} \text{ and } \mathbf{x} \geq 0;$$

$$P^*: \text{ minimize } z^*(\mathbf{y}) = \mathbf{b}^{\mathsf{T}}\mathbf{y}, \text{ subject to the constraints } A^{\mathsf{T}}\mathbf{y} \geq \mathbf{c} \text{ and } \mathbf{y} \geq 0.$$

Weak Duality Theorem

- Recall that any vector $\mathbf{x}$ which satisfies the two constraints, $A\mathbf{x} \leq \mathbf{b}$ and $\mathbf{x} \geq 0$ is called a *feasible solution*, regardless of what the corresponding objective value $\mathbf{c}^\mathsf{T}\mathbf{x}$ might be.

Weak Duality Theorem

- Recall that any vector $\mathbf{x}$ which satisfies the two constraints, $A\mathbf{x} \leq \mathbf{b}$ and $\mathbf{x} \geq 0$ is called a *feasible solution*, regardless of what the corresponding objective value $\mathbf{c}^\mathsf{T}\mathbf{x}$ might be.
- **Theorem** If $x = \langle x_1 \ldots x_n \rangle$ is any basic feasible solution for P and $y = \langle y_1 \ldots y_m \rangle$ is any basic feasible solution for P^*, then:

$$\sum_{n} \qquad \sum_{n}$$

Weak Duality Theorem

- Recall that any vector $\mathbf{x}$ which satisfies the two constraints, $A\mathbf{x} \leq \mathbf{b}$ and $\mathbf{x} \geq 0$ is called a *feasible solution*, regardless of what the corresponding objective value $\mathbf{c}^\mathsf{T}\mathbf{x}$ might be.
- **Theorem** If $x = \langle x_1 \ldots x_n \rangle$ is any basic feasible solution for P and $y = \langle y_1 \ldots y_m \rangle$ is any basic feasible solution for P^*, then:

$$\sum_{j=1}^{n} c_j x_j \leq \sum_{j=1}^{n} \left(\sum_{i=1}^{m} a_{ij} y_i \right) x_j = \sum_{i}^{n}$$

Pro respectively,
we ca P^* P to obtain

$$z(x) \equiv z^*(y)$$

Weak Duality Theorem

- Recall that any vector $\mathbf{x}$ which satisfies the two constraints, $A\mathbf{x} \leq \mathbf{b}$ and $\mathbf{x} \geq 0$ is called a *feasible solution*, regardless of what the corresponding objective value $\mathbf{c}^\mathsf{T}\mathbf{x}$ might be.
- **Theorem** If $x = \langle x_1 \ldots x_n \rangle$ is any basic feasible solution for P and $y = \langle y_1 \ldots y_m \rangle$ is any basic feasible solution for P^*, then:

$$\sum_{j=1}^{n} c_j x_j \leq \sum_{j=1}^{n} \left(\sum_{i=1}^{m} a_{ij} y_i \right) x_j = \sum_{i}$$

Pro respectively,
we ca P^* P to obtain

$$z(x) \equiv \sum_{j=1}^{n} c_j x_j \leq \sum_{j=1}^{n} \left(\sum_{i=1}^{m} a_{ij} y_i \right) x_j = \equiv z^*(y)$$

- Thus, the value of (the objective of P^* for) any feasible solution of P^* is an upper bound for the set of all values of (the objective of P for) all feasible solutions of P, and

Weak Duality Theorem

- Recall that any vector $\mathbf{x}$ which satisfies the two constraints, $A\mathbf{x} \leq \mathbf{b}$ and $\mathbf{x} \geq 0$ is called a *feasible solution*, regardless of what the corresponding objective value $\mathbf{c}^\mathsf{T}\mathbf{x}$ might be.
- **Theorem** If $x = \langle x_1 \ldots x_n \rangle$ is any basic feasible solution for P and $y = \langle y_1 \ldots y_m \rangle$ is any basic feasible solution for P^*, then:

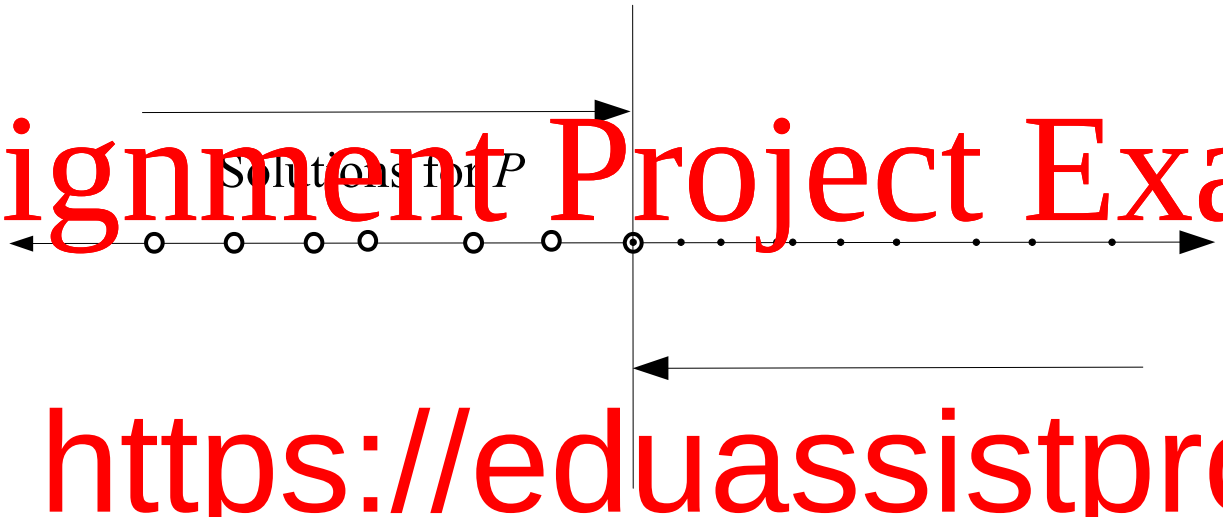
$$\sum_{j=1}^{n} c_j x_j \leq \sum_{i=1}^{m} b_i y_i$$

Proof Assume P and P^* are in standard form, respectively; we can always convert P^* and P to obtain

$$z(x) = \sum_{j=1}^{n} c_j x_j \leq \sum_{j=1}^{n} \left(\sum_{i=1}^{m} a_{ij} y_i \right) x_j = \sum_{i} \equiv z^*(y)$$

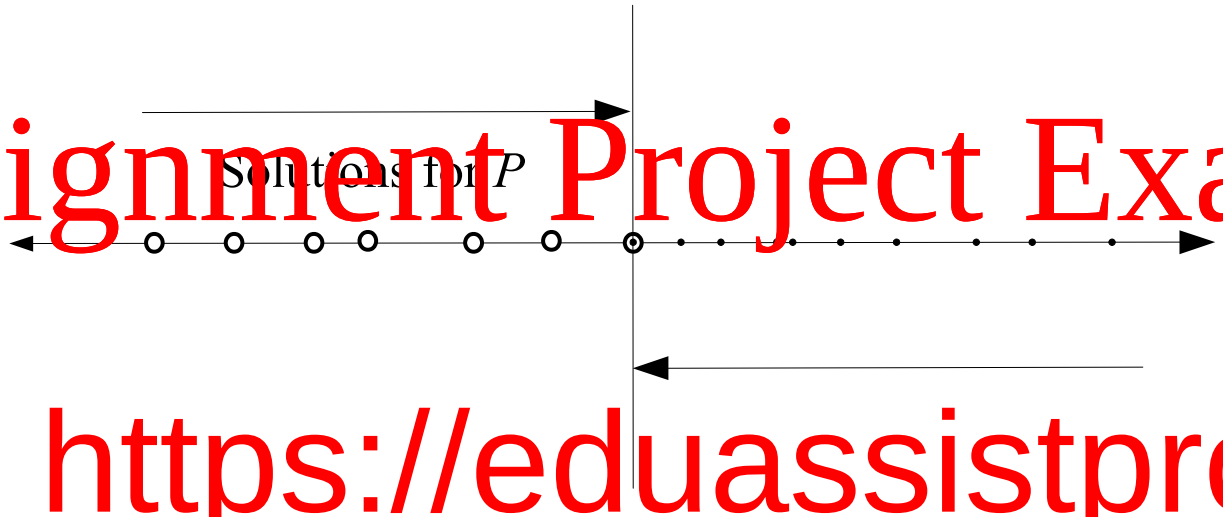
- Thus, the value of (the objective of P^* for) any feasible solution of P^* is an upper bound for the set of all values of (the objective of P for) all feasible solutions of P, and
- every feasible solution of P is a lower bound for the set of feasible solutions for P^*.

Weak Duality Theorem



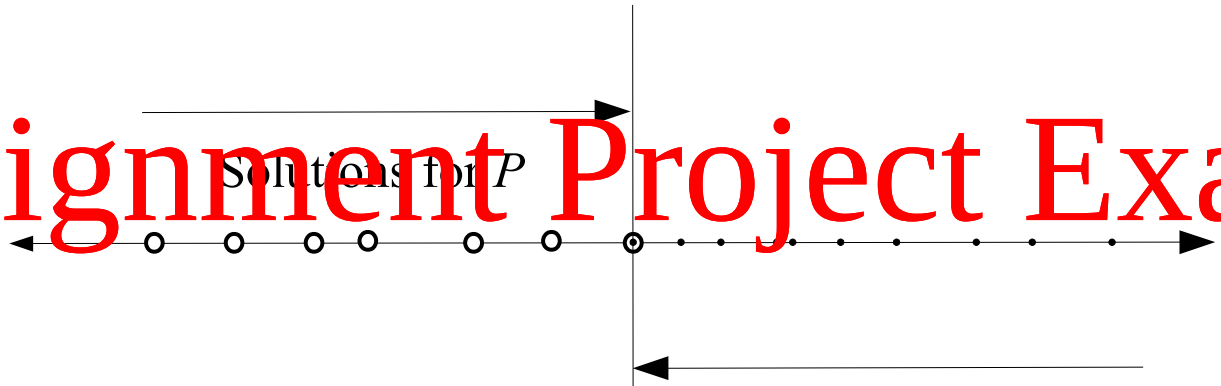
- Thus, if we find a feasible solution for P which is equal to a feasible solution to P^*, such solution must be the maxi
objective of P and the minimal feasible value P^*.

Weak Duality Theorem



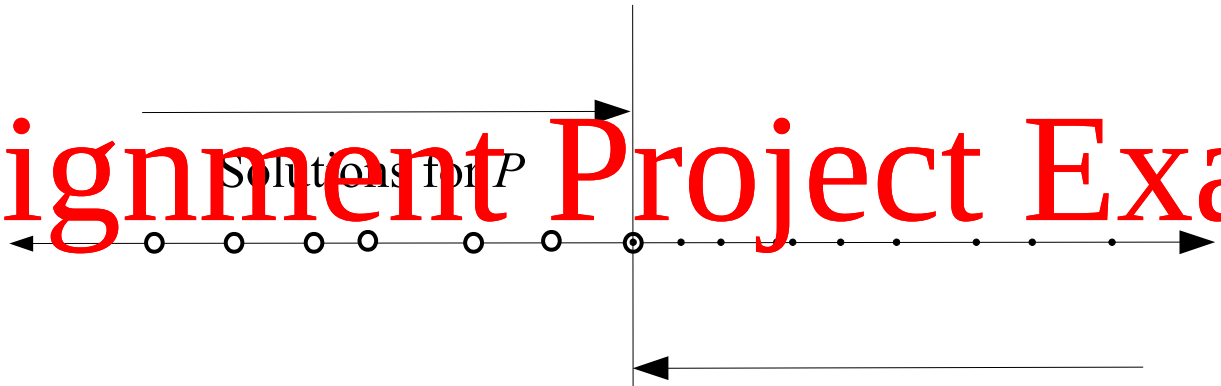
- Thus, if we find a feasible solution for P which is equal to a feasible solution to P^*, such solution must be the maxi
objective of P and the minimal feasible value P^*.
- If we use a search procedure to find an optimal solutio now when to stop: when such a value is also a feasible soluti .

Weak Duality Theorem



- Thus, if we find a feasible solution for P which is equal to a feasible solution to P^*, such solution must be the maxi
objective of P and the minimal feasible value P^*.
- If we use a search procedure to find an optimal solutio now when to stop: when such a value is also a feasible soluti .
- This is why the most commonly used LP solving method, the SIMPLEX method, produces optimal solution for P, because it stops at a value of the primal objective which is also a value of the dual objective.

Weak Duality Theorem



- Thus, if we find a feasible solution for P which is equal to a feasible solution to P^*, such solution must be the maxi objective of P and the minimal feasible value P^*.
- If we use a search procedure to find an optimal soluti now when to stop: when such a value is also a feasible soluti .
- This is why the most commonly used LP solving method, the SIMPLEX method, produces optimal solution for P, because it stops at a value of the primal objective which is also a value of the dual objective.
- See the Lecture Notes for the details and an example of how the SIMPLEX algorithm runs.

PUZZLE!!

Five siste
is playing c
Helen, th