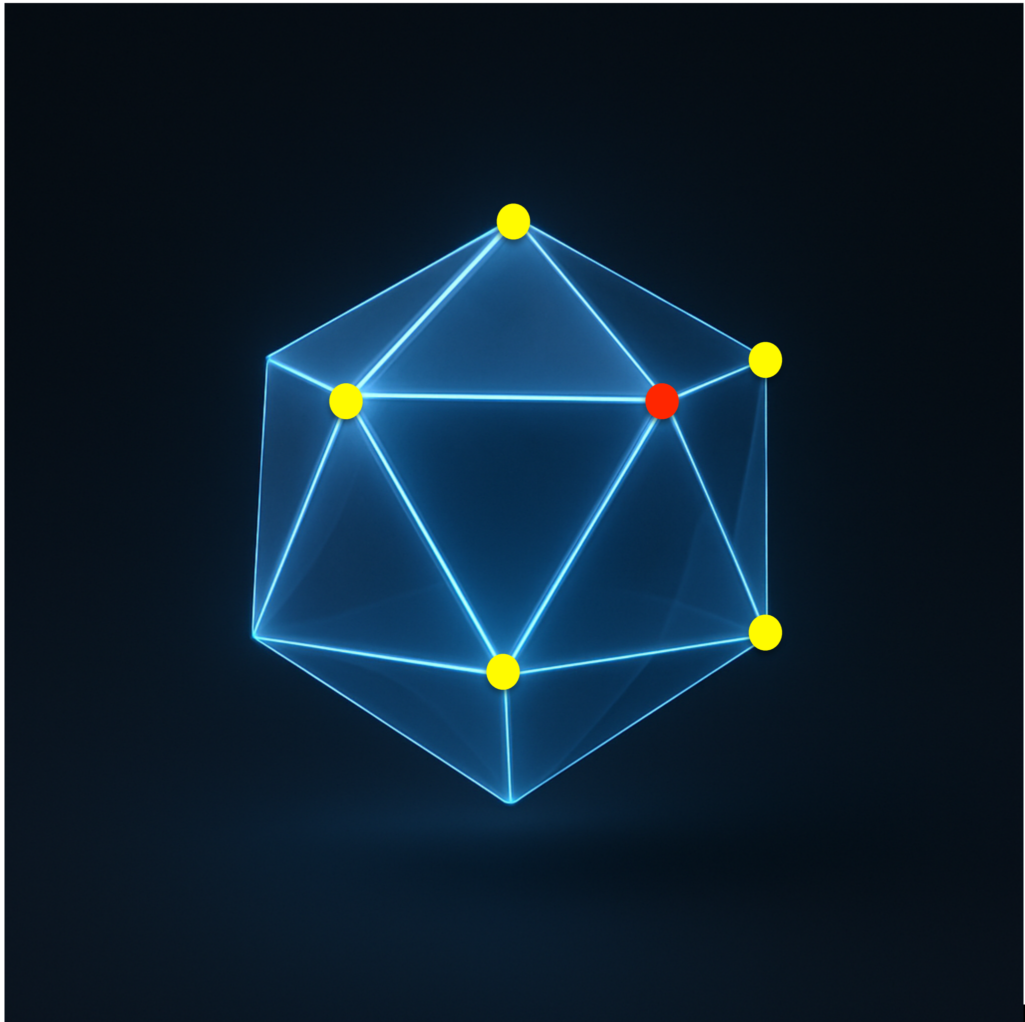
Simplex Tableaux Method



Deepak Ajwani

School of Computer Science

Recap

- Linear programs
- When 2 decision variables, can identify optimal solutions with help of graphical method.
- Feasible region is usually a polygon (or polytope, if we have more than 2 decision variables).
- Vertices correspond to feasible solutions (basic feasible solutions)
- Optimal solution always found at one of those.
- Today we look at the *tableaux method*.

Simplex Algorithm

Theorem 1: At least one of the points where the objective value is minimal is a vertex

Proof: Let x^* be the optimal solution of a minimisation LP problem. Since each point in a polytope is a convex combination of the vertices $v_1, v_2, \ldots, v_t$, we have $x^* = \lambda_1 v_1 + \lambda_2 v_2 + \ldots + \lambda_t v_t$ such that $\lambda_1 + \lambda_2 + \ldots + \lambda_t = 1.$

The objective function value at optimality can be expressed as $cx^* = c(\lambda_1 v_1 + \lambda_2 v_2 + \ldots + \lambda_t v_t)$
$= \lambda_1(cv_1) + \lambda_2(cv_2) + \ldots + \lambda_t(cv_t)$

If the minimum value x^* is not at a vertex, then $cx^* < cv_i \quad \forall i : 1 \leq i \leq t$

It follows that

$$
\begin{aligned}
cx^* = &\quad c(\lambda_1 v_1) + \ldots + c(\lambda_t v_t) \\
= &\quad \lambda_1(cv_1) + \ldots + \lambda_t(cv_t) \\
> &\quad \lambda_1(cx^*) + \ldots + \lambda_t(cx^*) \\
> &\quad (\lambda_1 \ldots + \lambda_t)(cx^*) \\
> &\quad cx^*
\end{aligned}
$$

This is a contradiction. Hence, it must be the case that $x^* = v_i$ for some $1 \leq i \leq t$

Simplex Algorithm

- So, we can only focus on vertices and ignore the internal points
- For any set of linear constraints, there are only ever *finitely many* of these vertices
- That's what makes LP a *combinatorial* optimisation problem. (Solution space is *essentially* finite)

Simplex Algorithm

- So, we can only focus on vertices and ignore the internal points
- For any set of linear constraints, there are only ever *finitely many* of these vertices
- That's what makes LP a *combinatorial* optimisation problem. (Solution space is *essentially* finite)
- But the number of vertices can still be very large, so finding the one with the smallest objective value by enumerating all is computationally expensive

Simplex Algorithm

- **Observation 2**: If a vertex is not optimal, we can find a neighbouring vertex that has a better or same objective value as itself

Simplex Algorithm

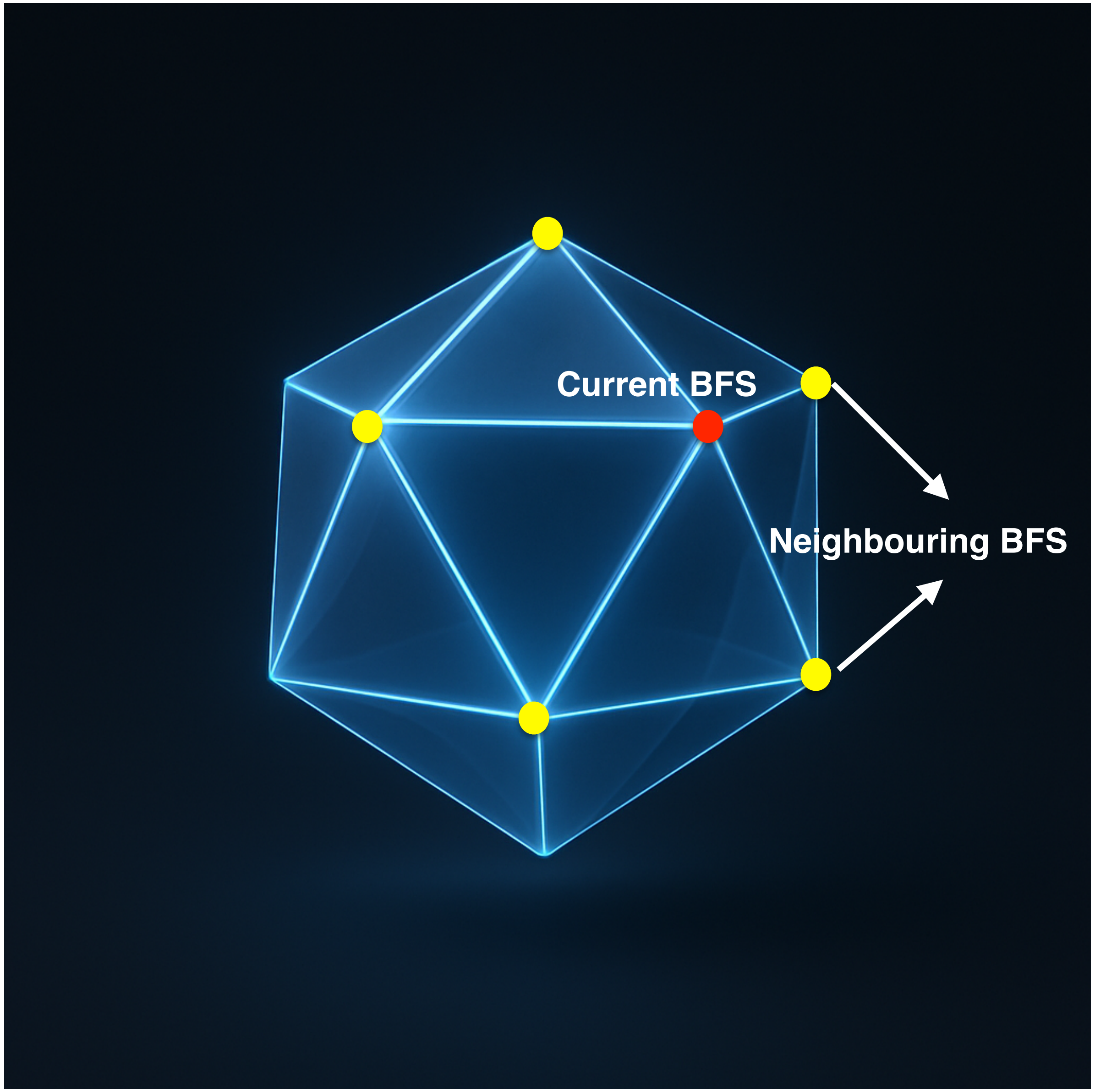
- **Observation 1**: At least one of the points where the objective value is minimal is a vertex
- **Observation 2**: If a vertex is not optimal, we can find a neighbouring vertex that has a better or same objective value as itself

- Start from a vertex and explore other vertices using local search
- So, what should our local search neighbourhood be?
  - Neighbouring vertices: Take the neighbourhood of a vertex to be all other vertices that can be reached by travelling along a single edge of the polygon/polytope
  - **The main insights of the Simplex method is that this neighbourhood is exact**

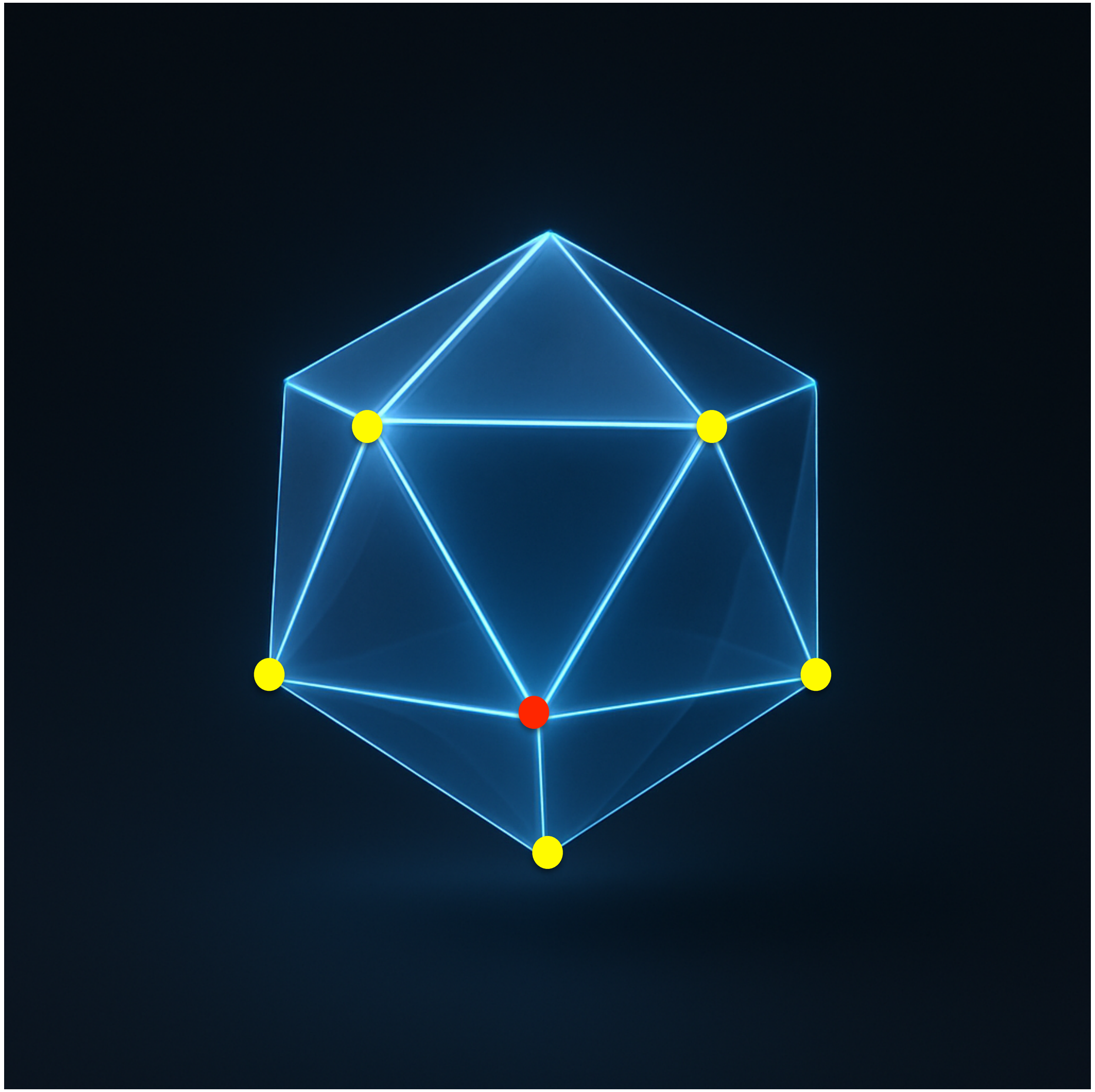
Simplex Algorithm

1. An optimal solution is at a vertex
2. A vertex is a basic feasible solution (BFS)
3. You can move from one BFS to a neighbouring BFS
4. You can detect whether a BFS is optimal
5. From any BFS, you can move to a BFS with a better cost

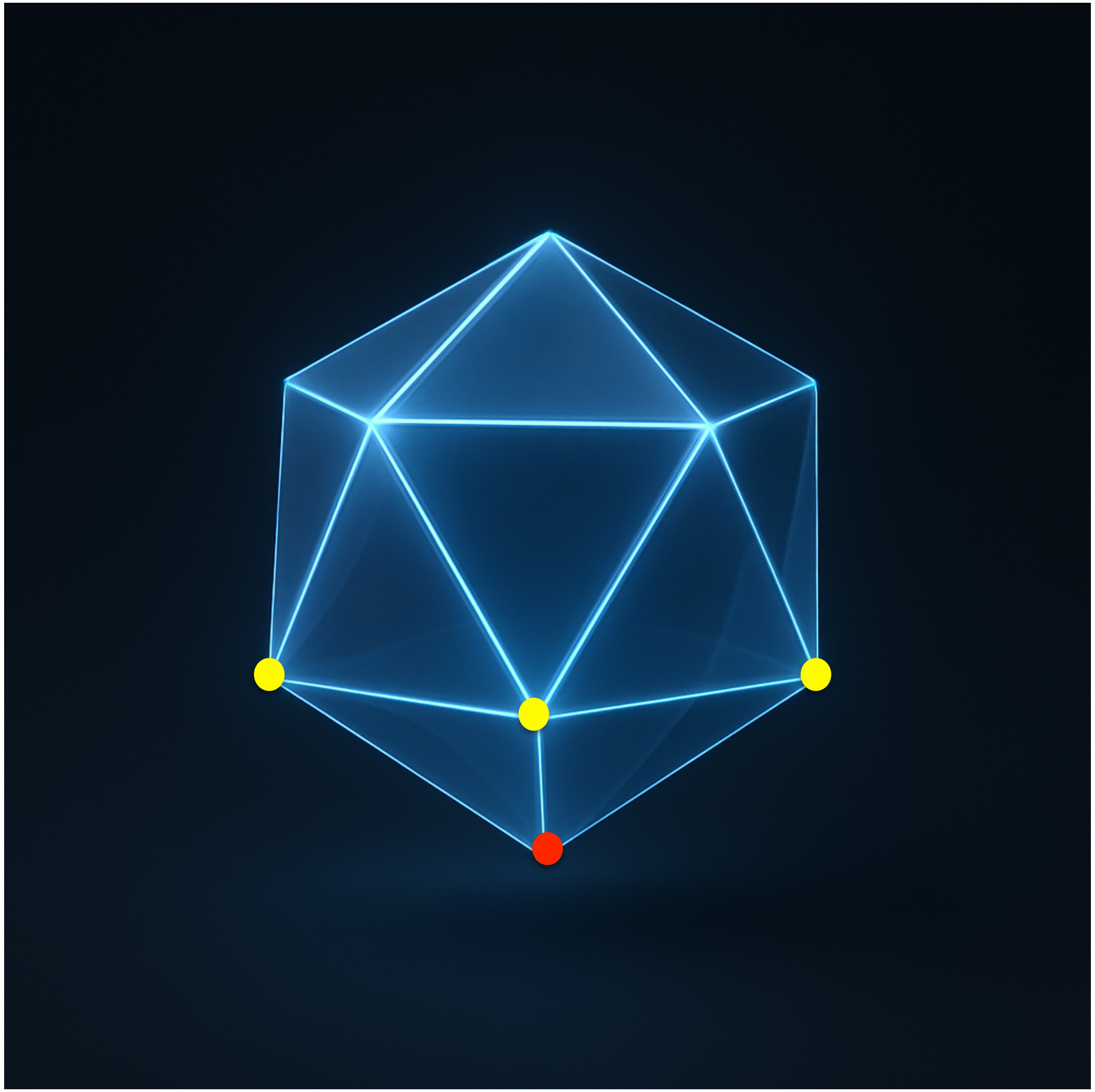
Simplex Algorithm



Simplex Algorithm



Simplex Algorithm



Simplex Algorithm

- Simplex algorithm is an extremely useful local search algorithm for solving linear programmes
    - It skips the interior region and jumps from vertex to vertex
    - It finds the global optimal solution

LP Problem for Today

- We will consider maximisation problems with constraints of the type $Ax \leq b$, where $x \geq 0$ and $b \geq 0$
  - For these constraints, origin is in the feasible region
- But what if the origin is not in the feasible region? How do you find the first feasible solution? [**Next Lecture**]

Simplex method: Tableaux version

Let's work through the same example LP from last class, but now using the tableaux method.

$$\textit{Maximise } 350x_1 + 300x_2$$

Subject to constraints:

$$x_1 + x_2 \leq 200$$

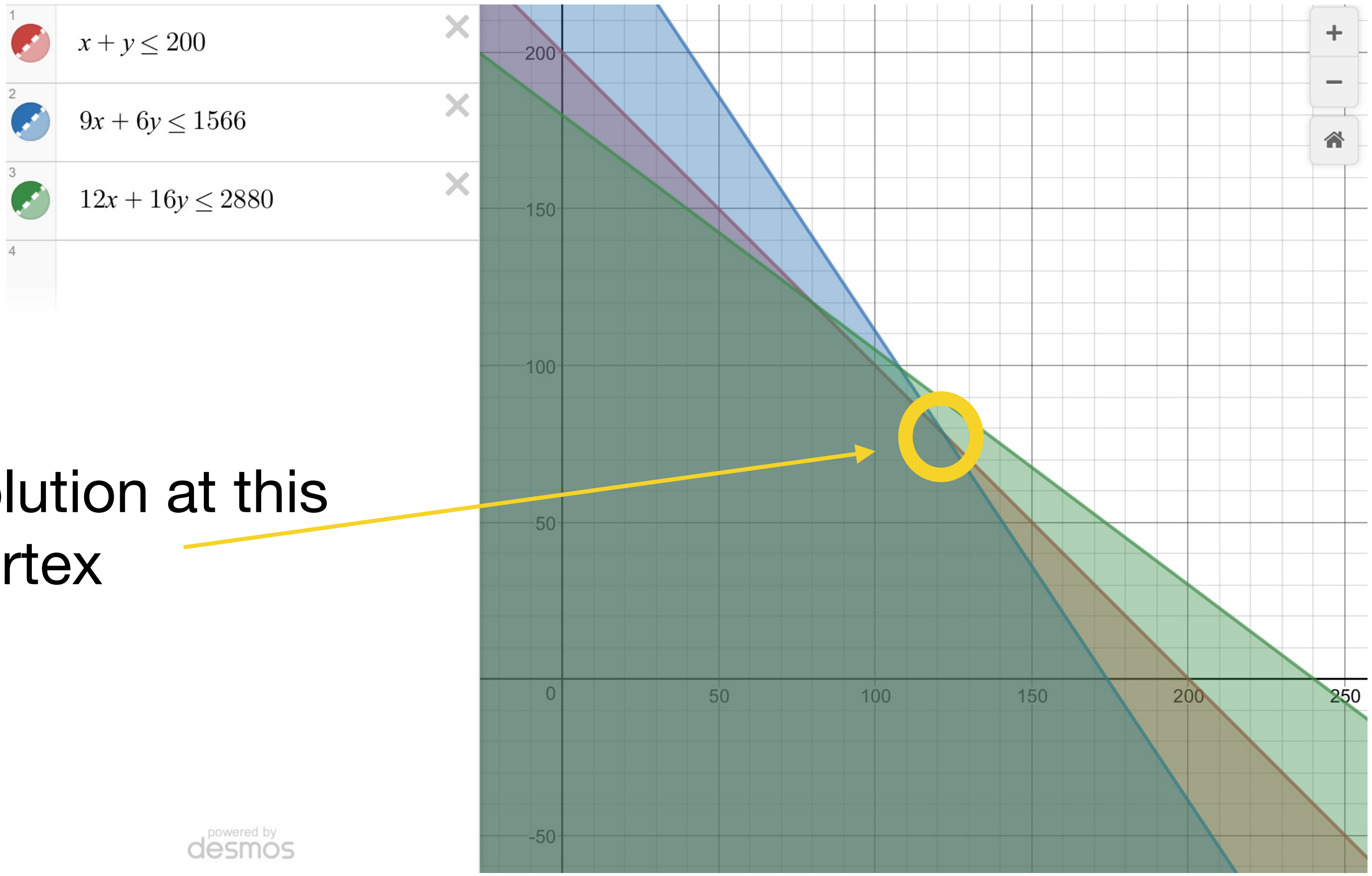
$$9x_1 + 6x_2 \leq 1566$$

$$12x_1 + 16x_2 \leq 2880$$

$$x_1 \geq 0,\ x_2 \geq 0$$

Feasible space

Recall the feasible space for these constraints (darkest area)



Optimal solution at this vertex

The procedure

1. Pre-processing steps (Convert LP into the right form.)

2. Create *initial tableau*

3. While the bottom row contains a negative number, do:

    a. Choose *pivot cell* in the tableau
    b. Do *row reduction* so pivot entry is 1 and all other entries in its column are 0

4. Optimal solution is the basic feasible solution corresponding to the (basis of the) final tableau.

The procedure

1. **Pre-processing steps (Convert LP into the right form.)**

2. Create *initial tableau*

3. While the bottom row contains a negative number, do:

   a. Choose *pivot cell* in the tableau
   b. Do *row reduction* so pivot entry is 1 and all other entries in its column are 0

4. Optimal solution is the basic feasible solution corresponding to the (basis of the) final tableau.

Pre-processing steps

First thing: tableaux method requires LP to be written in *standard form*. So, introduce some *slack variables* to the constraints:

$$
\begin{array}{rcrcrcrcrcr}
x_1 & + & x_2 & + & s_1 & & & & & = & 200 \\
9x_1 & + & 6x_2 & & & + & s_2 & & & = & 1566 \\
12x_1 & + & 16x_2 & & & & & + & s_3 & = & 2880
\end{array}
$$

$$x_1,\ x_2,\ s_1,\ s_2,\ s_3 \geq 0$$

Pre-processing steps

- *Second thing*: introduce variable *z* for the objective function.

Maximise z such that

$$z = 350x_1 + 300x_2$$

- Re-arrange this equation so x_1, x_2 on left hand side:

Maximise z such that

$$-350x_1 - 300x_2 + z = 0$$

Pre-processing steps

Our LP now looks like this:

Choose values for $x_1,\ x_2,\ s_1,\ s_2,\ s_3,\ z$ such that z is maximal and

$$-350x_1 - 300x_2 + z = 0$$

Subject to constraints

$$\begin{array}{rcrcrcrcrcr}
x_1 & + & x_2 & + & s_1 & & & & & = & 200 \\
9x_1 & + & 6x_2 & & & + & s_2 & + & & = & 1566 \\
12x_1 & + & 16x_2 & & & & & + & s_3 & = & 2880
\end{array}$$

$$x_1,\ x_2,\ s_1,\ s_2,\ s_3 \geq 0$$

The procedure

1. Pre-processing steps (Convert LP for into the right form.)

2. **Create *initial tableau***

3. While the bottom row contains a negative number, do:

   a. Choose *pivot cell* in the tableau
   b. Do *row reduction* so pivot entry is 1 and all other entries in its column are 0

4. Optimal solution is the basic feasible solution corresponding to the (basis of the) final tableau.

Initial tableau

We now convert this LP into our *initial tableau:*

x_1	x_2	s_1	s_2	s_3	z	
1	1	1	0	0	0	200
9	6	0	1	0	0	1566
12	16	0	0	1	0	2880
-350	-300	0	0	0	1	0

First 3 rows correspond to the 3 constraints

Bottom row corresponds to the objective function

Basis in a tableau

x_1	x_2	s_1	s_2	s_3	z	
1	1	1	0	0	0	200
9	6	0	1	0	0	1566
12	16	0	0	1	0	2880
-350	-300	0	0	0	1	0

Note the presence of 4 columns, each with a “1” and 3 zeros, and such that the “1” appears in a different row each time.

Any set of columns displaying this pattern form what’s called a *basis.* Every basis gives us a basic feasible solution (i.e., a vertex) to the LP.

Basic feasible solution corresponding to a basis

x_1	x_2	s_1	s_2	s_3	z	
1	1	1	0	0	0	200
9	6	0	1	0	0	1566
12	16	0	0	1	0	2880
-350	-300	0	0	0	1	0

We can read off the basic feasible solution corresponding to a basis as follows:

1. If the variable's column does **not** appear in the basis then set it equal to 0.
2. If the variable's column **does** appear in the basis then set it equal to that value in the final column appearing in the row corresponding to its "1".

Basic feasible solution corresponding to a basis

x_1	x_2	s_1	s_2	s_3	z	
1	1	1	0	0	0	200
9	6	0	1	0	0	1566
12	16	0	0	1	0	2880
-350	-300	0	0	0	1	0

Basic feasible solution corresponding to the above basis is $x_1 = x_2 = 0,\ s_1 = 200,\ s_2 = 1566,\ s_3 = 2880,\ z = 0.$

Basic feasible solution corresponding to a basis

$$x_1 = x_2 = 0,\ s_1 = 200,\ s_2 = 1566,\ s_3 = 2880,\ z = 0.$$

So we're currently at this point

The procedure

1. Pre-processing steps (Convert LP for into the right form.)

2. Create *initial tableau*

3. **While the bottom row contains a negative number, do:**

   a. Choose *pivot cell* in the tableau
   b. Do *row reduction* so pivot entry is 1 and all other entries in its column are 0

4. Optimal solution is the basic feasible solution corresponding to the (basis of the) final tableau.

Termination condition of the Tableaux method

x_1	x_2	s_1	s_2	s_3	z	
1	1	1	0	0	0	200
9	6	0	1	0	0	1566
12	16	0	0	1	0	2880
-350	-300	0	0	0	1	0

If the bottom row has **no** negative numbers in it, then it means *z* cannot be increased any further. Then the method ends and our current basic feasible solution is the optimal solution to the LP.

If there are still negative numbers there (as in this example), then we must perform an iteration of the method to move to a better vertex.

The procedure

1. Pre-processing steps (Convert LP for into the right form.)

2. Create *initial tableau*

3. While the bottom row contains a negative number, do:

    a. **Choose *pivot cell* in the tableau**

    b. Do *row reduction* so pivot entry is 1 and all other entries in its column are 0

4. Optimal solution is the basic feasible solution corresponding to the (basis of the) final tableau.

Choosing pivot cell

To identify the pivot cell we need to identify

1. the pivot *column,* followed by
2. the pivot *row*.

Choosing pivot column

x_1	x_2	s_1	s_2	s_3	z	
1	1	1	0	0	0	200
9	6	0	1	0	0	1566
12	16	0	0	1	0	2880
-350	-300	0	0	0	1	0

Choose a column with the *smallest* value in the bottom row. (I.e., the *"most negative"*)

Choosing pivot row

x_1	x_2	s_1	s_2	s_3	z	
1	1	1	0	0	0	200
9	6	0	1	0	0	1566
12	16	0	0	1	0	2880
-350	-300	0	0	0	1	0

For each row, divide the number in the *last* column by the number in the pivot column. **(Disregard a row if the number in the pivot column is less than or equal to 0).** Choose that row which gives the smallest value.

Choosing pivot row

x_1	x_2	s_1	s_2	s_3	z	
1	1	1	0	0	0	200
9	6	0	1	0	0	1566
12	16	0	0	1	0	2880
-350	-300	0	0	0	1	0

In this example we have 200/1 (row 1), 1566/9 (row 2), 2880/12 (row 3). The smallest of these is 1566/9, so the pivot row is row 2

Choosing pivot row

x_1	x_2	s_1	s_2	s_3	z	
1	1	1	0	0	0	200
9	6	0	1	0	0	1566
12	16	0	0	1	0	2880
-350	-300	0	0	0	1	0

In this example we have 200/1 (row 1), 1566/9 (row 2), 2880/12 (row 3). The smallest of these is 1566/9, so the pivot row is row 2

Choosing pivot row

x_1	x_2	s_1	s_2	s_3	z	
1	1	1	0	0	0	200
9	6	0	1	0	0	1566
12	16	0	0	1	0	2880
-350	-300	0	0	0	1	0

Combining the pivot column with the pivot row gives us the pivot cell.

The procedure

1. Pre-processing steps (Convert LP for into the right form.)

2. Create *initial tableau*

3. While the bottom row contains a negative number, do:

    a. Choose *pivot cell* in the tableau

    b. Do *row reduction* so pivot entry is 1 and all other entries in its column are 0

4. Optimal solution is the basic feasible solution corresponding to the (basis of the) final tableau.

Obtaining "1" in pivot cell

	x_1	x_2	s_1	s_2	s_3	z	
R1	1	1	1	0	0	0	200
R2	9	6	0	1	0	0	1566
R3	12	16	0	0	1	0	2880
R4	-350	-300	0	0	0	1	0

To obtain a "1" in the pivot cell, divide the entire pivot row by the number in the pivot cell.

In this example we replace row $R2$ with $R2 \times \frac{1}{9}$.

Obtaining “1” in pivot cell

	x_1	x_2	s_1	s_2	s_3	z	
R1	1	1	1	0	0	0	200
R2	1	2/3	0	1/9	0	0	174
R3	12	16	0	0	1	0	2880
R4	-350	-300	0	0	0	1	0

To obtain a “1” in the pivot cell, divide the entire pivot row by the number in the pivot cell.

In this example we replace row $R2$ with $R2 \times \frac{1}{9}$.

Obtaining "0"s elsewhere in pivot column

	x_1	x_2	s_1	s_2	s_3	z	
R1	1	1	1	0	0	0	200
R2	1	2/3	0	1/9	0	0	174
R3	12	16	0	0	1	0	2880
R4	-350	-300	0	0	0	1	0

To obtain zeros elsewhere in the pivot column, for each other row we subtract an appropriate multiple of the (new) pivot row.

In this example we replace $R1$ with $R1 - (1 \times R2)$, replace $R3$ with $R3 - (12 \times R2)$, and replace $R4$ with $R4 - (-350 \times R2)$.

(In general, replace j^{th} row Rj with $Rj - (a_i \times R2)$, where a_j is the j^{th} entry in the pivot column.)

Obtaining “0”s elsewhere in pivot column

	x_1	x_2	s_1	s_2	s_3	z	
R1	0	1/3	1	-1/9	0	0	26
R2	1	2/3	0	1/9	0	0	174
R3	0	8	0	-4/3	1	0	792
R4	0	-200/3	0	350/9	0	1	60900

To obtain zeros elsewhere in the pivot column, for each other row we subtract an appropriate multiple of the (new) pivot row.

In this example we replace $R1$ with $R1 - (1 \times R2)$, replace $R3$ with $R3 - (12 \times R2)$, and replace $R4$ with $R4 - (-350 \times R2)$.

(In general, replace j^{th} row Rj with $Rj - (a_i \times R2)$, where a_j is the j^{th} entry in the pivot column.)

We’ve moved to a new basis

	x_1	x_2	s_1	s_2	s_3	z	
$R1$	0	1/3	1	-1/9	0	0	26
$R2$	1	2/3	0	1/9	0	0	174
$R3$	0	8	0	-4/3	1	0	792
$R4$	0	-200/3	0	350/9	0	1	60900

The corresponding basic feasible solution is
$x_1 = 174$, $x_2 = 0$, $s_1 = 26$, $s_2 = 0$, $s_3 = 792$, $z = 60900$.

We've moved to a new basis

$$x_1 = 174,\ x_2 = 0,\ s_1 = 26,\ s_2 = 0,\ s_3 = 792,\ z = 60900$$

So we've now moved to this vertex

Now check termination condition

	x_1	x_2	s_1	s_2	s_3	z	
R1	0	1/3	1	-1/9	0	0	26
R2	1	2/3	0	1/9	0	0	174
R3	0	8	0	-4/3	1	0	792
R4	0	-200/3	0	350/9	0	1	60900

We still have negative values in the bottom row, so the termination condition is not met.

The procedure

1. Pre-processing steps (Convert LP for into the right form.)

2. Create *initial tableau*

3. **While the bottom row contains a negative number, do:**

   a. Choose *pivot cell* in the tableau
   b. Do *row reduction* so pivot entry is 1 and all other entries in its column are 0

4. Optimal solution is the basic feasible solution corresponding to the (basis of the) final tableau.

Second Iteration

	x_1	x_2	s_1	s_2	s_3	z	
R1	0	1/3	1	-1/9	0	0	26
R2	1	2/3	0	1/9	0	0	174
R3	0	8	0	-4/3	1	0	792
R4	0	-200/3	0	350/9	0	1	60900

Choose a column with the *smallest* value in the bottom row. (I.e., the *"most negative"*)

Second Iteration

	x_1	x_2	s_1	s_2	s_3	z	
R1	0	1/3	1	-1/9	0	0	26
R2	1	2/3	0	1/9	0	0	174
R3	0	8	0	-4/3	1	0	792
R4	0	-200/3	0	350/9	0	1	60900

For each row, divide the number in the *last* column by the number in the pivot column. **(Disregard a row if the number in the pivot column is less than or equal to 0).** Choose that row which gives the smallest value.

Second Iteration

	x_1	x_2	s_1	s_2	s_3	z	
R1	0	1/3	1	-1/9	0	0	26
R2	1	2/3	0	1/9	0	0	174
R3	0	8	0	-4/3	1	0	792
R4	0	-200/3	0	350/9	0	1	60900

In this example we have 26/(1/3) = 78 (row 1), 174/(2/3) = 261 (row 2), 792/8 = 99 (row 3). The smallest of these is 78, so the pivot row is row 1

Second Iteration

x_1	x_2	s_1	s_2	s_3	z	
0	1/3	1	-1/9	0	0	26
1	2/3	0	1/9	0	0	174
0	8	0	-4/3	1	0	792
0	-200/3	0	350/9	0	1	60900

Combining the pivot column with the pivot row gives us the pivot cell.

The procedure

1. Pre-processing steps (Convert LP for into the right form.)

2. Create *initial tableau*

3. While the bottom row contains a negative number, do:

   a. Choose *pivot cell* in the tableau
   b. Do *row reduction* so pivot entry is 1 and all other entries in its column are 0

4. Optimal solution is the basic feasible solution corresponding to the (basis of the) final tableau.

Second Iteration

	x_1	x_2	s_1	s_2	s_3	Z	
$R1$	0	1/3	1	-1/9	0	0	26
$R2$	1	2/3	0	1/9	0	0	174
$R3$	0	8	0	-4/3	1	0	792
$R4$	0	-200/3	0	350/9	0	1	60900

To obtain a "1" in the pivot cell, divide the entire pivot row by the number in the pivot cell.

In this example we replace row $R1$ with $R1 \times 3$.

Second Iteration

	x_1	x_2	s_1	s_2	s_3	Z	
R1	0	1	3	-1/3	0	0	78
R2	1	2/3	0	1/9	0	0	174
R3	0	8	0	-4/3	1	0	792
R4	0	-200/3	0	350/9	0	1	60900

To obtain a "1" in the pivot cell, divide the entire pivot row by the number in the pivot cell.

In this example we replace row $R1$ with $R1 \times 3$.

Second Iteration

	x_1	x_2	s_1	s_2	s_3	z	
$R1$	0	1	3	-1/3	0	0	78
$R2$	1	2/3	0	1/9	0	0	174
$R3$	0	8	0	-4/3	1	0	792
$R4$	0	-200/3	0	350/9	0	1	60900

To obtain zeros elsewhere in the pivot column, for each other row we subtract an appropriate multiple of the (new) pivot row.

In this example we replace $R2$ with $R2 - ((2/3) \times R1)$, replace $R3$ with $R3 - (8 \times R1)$, and replace $R4$ with $R4 + ((200/3) \times R1)$.

(In general, replace j^{th} row Rj with $Rj - (a_i \times R1)$, where a_j is the j^{th} entry in the pivot column.)

Second Iteration

	x_1	x_2	s_1	s_2	s_3	z	
R1	0	1	3	-1/3	0	0	78
R2	1	0	-2	1/3	0	0	122
R3	0	0	-24	4/3	1	0	168
R4	0	0	200	150/9	0	1	66100

To obtain zeros elsewhere in the pivot column, for each other row we subtract an appropriate multiple of the (new) pivot row.

In this example we replace $R2$ with $R2 - ((2/3) \times R1)$, replace $R3$ with $R3 - (8 \times R1)$, and replace $R4$ with $R4 + ((200/3) \times R1)$.

(In general, replace j^{th} row Rj with $Rj - (a_j \times R1)$, where a_j is the j^{th} entry in the pivot column.)

Second Iteration

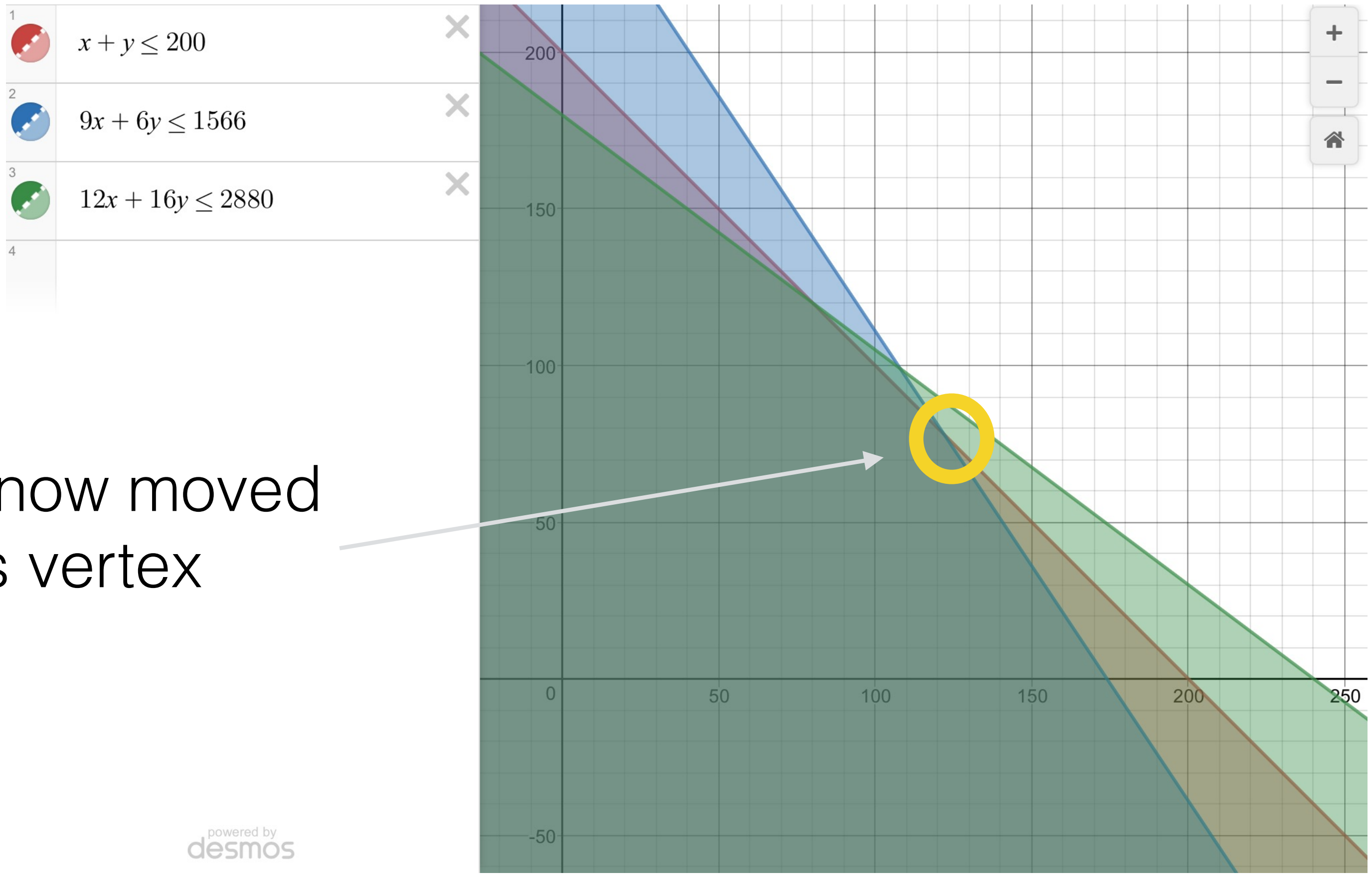
	x_1	x_2	s_1	s_2	s_3	z	
$R1$	0	1	3	-1/3	0	0	78
$R2$	1	0	-2	1/3	0	0	122
$R3$	0	0	-24	4/3	1	0	168
$R4$	0	0	200	150/9	0	1	66100

The corresponding basic feasible solution is
$x_1 = 122$, $x_2 = 78$, $s_1 = 0$, $s_2 = 0$, $s_3 = 168$, $z = 66100$.
No more negative values in the bottom row, so the method terminates here.

We've moved to a new basis

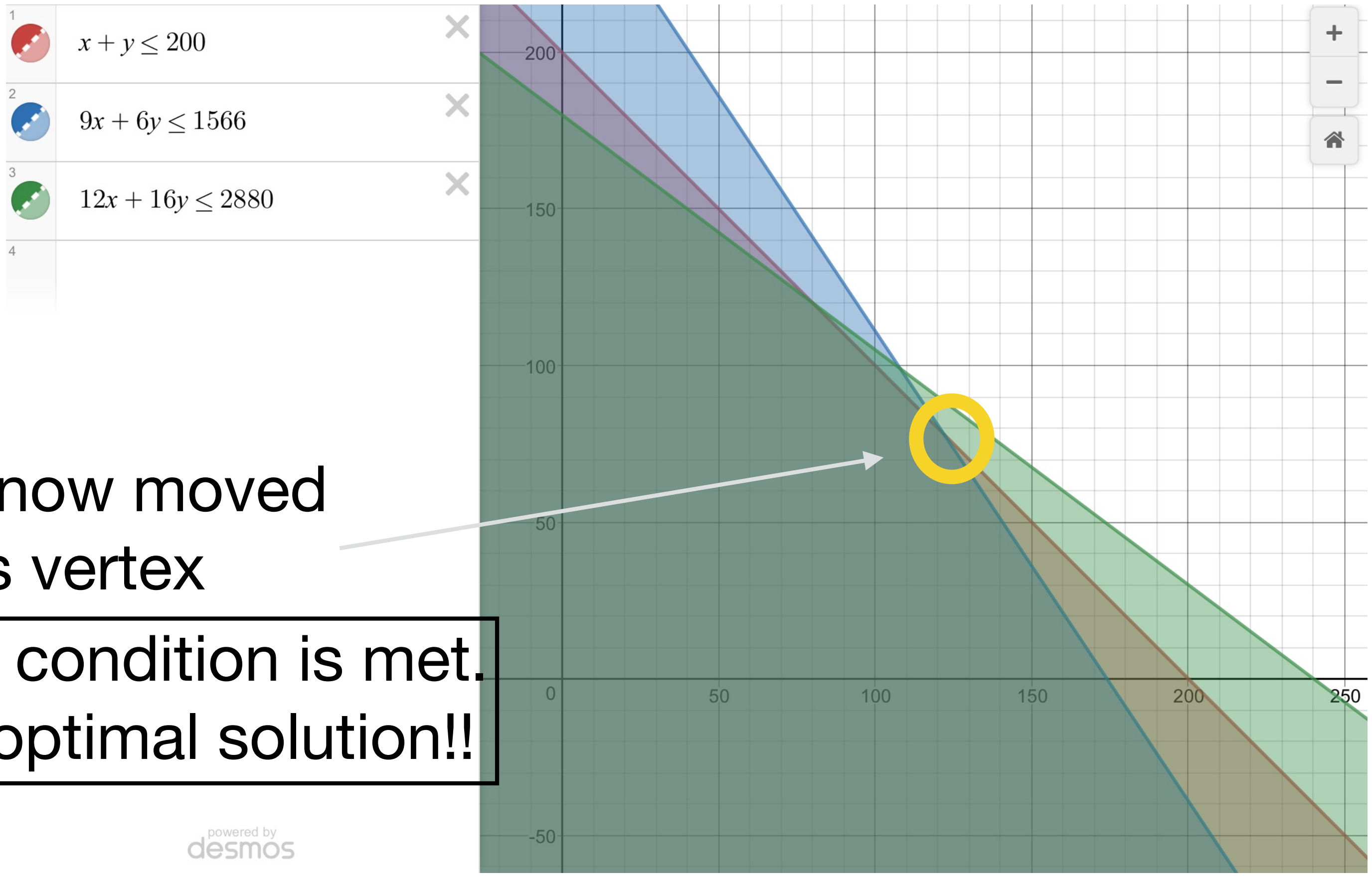
$x_1 = 122,\ x_2 = 78,\ s_1 = s_2 = 0,\ s_3 = 168,\ z = 66100$

So we've now moved to this vertex



The New Basis is Optimal

$$x_1 = 122,\ x_2 = 78,\ s_1 = s_2 = 0,\ s_3 = 168,\ z = 66100$$



So we've now moved to this vertex

Termination condition is met. This is the optimal solution!!